ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Application-Aware Data Center Network Bandwidth Utilization: the case of ethio telecom

by

Zerihun Mamo

Supervised by

Mesfin kifle (PhD.)

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Telecom Engineering

November, 2018

Addis Ababa, Ethiopia

Abstract

The existing ethio telecom data center network (DCN) is the traditional model which provides only the Best Effort (BE) traffic delivery service on a layer three links with the same priorities for all applications traffic. A diversity of applications is running on the data centers, the scarcity in-network bandwidth of data centers become the performance bottleneck for the integration of enterprise systems. The most important point is that the existing network has not a dynamic bandwidth management strategy, which lacks of flexible bandwidth utilization among different types of applications. To guarantee the network performance for system integration, an efficient in-network bandwidth management should be considered.

In this thesis work, a QoS model for an aggregated applications traffic in the DCN with a constrained bandwidth and with a consideration of the business criticality of the applications has been designed. The model nearly supports guarantees of QoS to real-time traffic without reserved bandwidth, and it assured forwarding high priority class traffic for business and mission critical applications. The design approach is based on the Internet Engineering Task Force (IETF) Differentiated Service(DS) Per-Hop-Behavior (PHB) group. Different types of Active queue management(AQM) and packet scheduler algorithms have been compared on the proposed design to minimize the packet loss and delay of each aggregated traffic class. In addition, the dynamic Benefit weighted scheduling (DB-WS) algorithm is modified to adapt with our solution, the algorithm dynamically allocated bandwidth based on the average queue length of each service class.

Extensive simulation results have shown that our proposed design is capable of improving the bandwidth utilization as well as providing desirable network performance for real-time and business-critical applications on a bottleneck link and also reduce the total packet loss by 1.34 %.

Keywords: Quality of Service (QoS), Differentiated Service (DiffServ), Differentiated Service Code Point (DSCP), Expedited Forwarding (EF), Assured Forwarding (AF), constrained bandwidth, Active queue management(AQM), bandwidth allocation

Acknowledgements

I would like to take this opportunity to make an attempt to express my gratitude towards all those people who have played an important role in some way or another to help me achieve whatever little I have.

First and foremost, I would like to offer my sincerest gratitude to ethio telecom for giving me the scholarship which helped me to proceed in my academic carrier and I'm genuinely grateful for the opportunity.

I would like to express my thanks to my advisor Dr. Mesfin Kifle for his encouragement and sound judgement, and provide valuable suggestions which helped in overcoming many obstacles and keeping the work on the right track under his thoughtful guidance.

Finally, I am grateful for my best friend Mr. Befekadu Worku for his valuable time and cooperation during the study of the current data center in the company. I am also obliged to all my friends who have always supported me directly or indirectly, to continue my work.

Contents

List of Figures

List of Tables

Abbreviations

AF	Assured Forwarding
AQM	Active Queue Management
BE	Best Effort
DB-WS	Dynamic Benefit Weighted Scheduling
DCN	Data Center Network
DiffServ	Differentiated Service
DPI	Deep Packet Inspection
DS	Differentiated Service
DSCP	Differentiated Service Code Point
EF	Expedited Forwarding
FCFS	First Come First Serve
FIFO	First in First Out
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet engineering task force
IntServ	Integrated Service
IP	Internet Protocol
ISP	Internet Service Provider
ITU	International Telecommunication Union
MF	Multi Filed
ML	Machine Learning
PHB	Per Hop Behavior

P2P	Point-to-Point
PQM	Passive Queue Management
PS	Priority Scheduling
QoS	Quality of Service
RED	Random Early Detection
RFC	Request for Comments
RSVP	Resource Reservation Protocol
SDN	Software-Defined Networking
SP	Strict Priority
ToR	Top of the Rack
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

1. Introduction

1.1. Background

In telecom industry, the telecom systems typically consist of numerous technologies, protocols, applications, and hardware's which are distributed across a network. Oftentimes the network is heterogeneous and is composed of diverse devices and operating systems. According to the enterprise systems become increasingly distributed and heterogeneous across multiple organizational and geographical boundaries, the interoperability and interconnectivity among those systems to exchange information have a huge influence on customer's experiences. In recent years, huge enterprises have a central or distributed data centers to host their systems. There is also a strong demand to integrate distributed systems in more than one data center in order to increase enterprises' competitiveness. Now a day, data centers have a big role on the telecom business day to day activities and performance, and their number and size also growing quickly.

In ethio telecom, there are five data centers which are located in different location and they contain more than several hundred's computing systems, such as servers, storage, network and other computing device. As a traditional data center network (DCN), it has a tree based interconnect topology, utilizing copper and optical cable as the wired links between the three levels of hierarchy namely core, aggregation and access layer. Some major issues of the current topology are scalability, undefined oversubscription ratio and only the uncontrolled best effort traffic delivery service model is functional. The best-effort service is working very well with data application like file transfer, mail delivery and web browsing. However, it does not provide guarantee for real-time applications in terms of end-to-end packet delay, and packet loss and it is not given priority for a business and mission critical applications traffic when inadequate resources are available [19].

Today's data center networking does not have a feature to directly exchange information between applications and the network. Due to the varying needs of the mixed applications running on top of the network, it might be leading to the scarcity of resources (bandwidth), its consequence to a poor user's quality experience of the applications in certain situations. To overcome this limitation, IETF provides an integrated service (IntServ) QoS models to guaranteed network resource per applications request. However, its scalability drawback emerged for a differentiated service (DiffServ) QoS model which assured forwarding for aggregated traffic class. In order to get good Quality of Service from the network, we need to control the traffic flow and manage the available

network resources in the data center based on the applications communication behavior and also we must well understand the relationship among application aware networking, QoS models and bandwidth management. [29]

In this research, an approach to solve the bandwidth-sharing problem on a bottleneck link based on the business criticality of the application and its networking behavior. Our approach allows applications traffic to categorize into multiple classes and controlled each traffic class queue independently on the routing device. In the rest of this Chapter, the research problem is defining with some exploration. The objective of the research and its scope are also set in this chapter and then the methodology used to achieve our objectives and the contribution of the research are described. Finally, the layout of the whole structure of the thesis is described.

1.2. Statement of the problem

In data center network, there are several hundred computing devices or servers with diversity of applications which are interchanged information each other's to carry company's day-to-day business activity. The scarcity of the resource or bandwidth in the data center network is one of the cause for the degrading of the performance of the applications and its integration to each other. One of the finding on the assessment of the ethio telecom enterprise system integration problems was the data center poor network performance and it had been the cause of the seamless integration failure among the company enterprise systems or applications.

The partial data collected from the company's ManageEngin monitoring system shows the incidence of a bottleneck on layer three links, especially at core layer devices interfaces which is connected to the backbone wide area network (WAN) and also the I2000 monitoring system event log shows there is a network interruption and communication failure between integrated enterprise applications. To overcome those problems temporarily, the company applies QoS method at the access network of the Intranet and also scheduling high bandwidth intensive job to run at night. However, these solutions do not consider the network congestion problem at aggregate and core layer of the DCN.

Applications in the data center have different network requirement and communication behavior. In addition, the applications criticality level for the company's business is also different. So, it needs differential traffic treatment for applications traffic. For example, the interactive application

which influence on the quality of the customer experience or the real time traffic which are not tolerate delay may need high level priority when congestion or bottleneck occurs.

In recent years, the emerging of the software-defined networking (SDN) and virtual network functionality (VNF) technology, it makes easy for the researcher to bring application-aware network solution [5,6,7]. However, most of the study conduct to solve intrusion attack by using application level traffic identification technique or to provide QoS for the Internet traffic on SDN enable networks. Even if some researcher study on resource management in DCN, their solution lay on the reservation of bandwidth to a particular application such as Hadoop solution which reduced the completion time of each Map reduced job [18]. However, the network in our data center is a traditional one (not support SDN) with the existence of diverse application types which need different network resources. So, our study of the new solution for all real time and critical application in the DCN is absolutely essential.

1.3. Objective

1.3.1. General Objective

The main objective of this thesis is to design application aware network bandwidth utilization solution on shared data center layer three links with a constrain of available bandwidth.

1.3.2. Specific Objective

The specific objectives of the research are: -

- To review state-of-the-art in application aware networking and bandwidth utilization technique.
- To categorize DCN traffic according to its applications network requirement (delay, packet loss and throughput).
- To design applications traffic identification and classification method for each traffic enter in the DCN.
- To compare and select queue management algorithm per each applications traffic class.
- To compare and select dynamic bandwidth allocation algorithm suitable for application aware bandwidth utilization.

- To design bandwidth utilization and traffic control solution based on the current DCN capability, applications communication behavior and companies networking rule and policy with a constrained bandwidth bottleneck.
- To validate the proposed solution by using an open source simulator.

1.4. Methodology

1.4.1. Literature Review

Different types of literatures (i.e. IEEE papers, journals, textbooks and public documents) reviewing to understand data center network and its characteristics: bandwidth allocation algorithm, traffic control technique and quality of service, and also to understand how other researchers solves related problems.

1.4.2. Data Collection and Analysis

Gathering and analysis of a high and low-level network design document, network access rules and policy, operational procedure, data center network device (such as switch, firewall, router, etc.) vendor manual and default device configuration. In addition, the link usage and the device stats are collected from existing network monitoring system.

1.4.3. Design Solution

Our design solution is based on the data collected from the company (topology, applications or systems network requirements, and traffic property) and the lesson from the related problem solved by other researcher. In this part we include:

- Design the quality of service architecture based on the Internet engineering task force (IETF) standard and suitable to our solution.
- Design different aggregate traffic class based on the application network requirement (delay, packet loss and throughput).
- Design traffic control and dynamic bandwidth allocation solution based on the network requirement of each traffic class which has the ability to prioritize critical application traffic and provide some level of guarantee for applications which needs real time communication.

1.4.4. Evaluation

As a Proof of concept, the OMNET++ simulator software is used and the evaluation of the solution performed on a similar network topology. The evaluation will focus on multiple queue traffic control and dynamic bandwidth allocation with different aggregated applications priority level. Then the performance metrics for measuring and comparing the results in different scenarios are defined.

1.5. Scope and Limitation of the Thesis

The scope of this thesis is to design an application aware bandwidth utilization solution for server to server data center networking at the forwarding plane which include the classification of applications in multiple traffic class, the comparison of different queueing and scheduling algorithm with adaptation of the delay, and loss requirement of those applications traffic with a constrain of available bandwidth at layer three network. This study based on the current data center applications communication behavior and network capability.

This research does not include the VPN connectivity among the distributed DCNs which belong to the wide area networks (WAN), and the management method of the shared network resources at the control plane. In addition, the preventive of network congestion or bottleneck problem is not the main focus of this thesis. Instead, it brings a solution which gives priority for business critical and real-time applications traffic over non-critical and delay tolerant applications when those problems happened for a short period of time.

1.6. Contribution of the Thesis

The contributions of this thesis are the following:

- A comprehensive background presents about the DCN, the basic concept of traffic control, the IETF QoS architecture and the dynamic resource allocation of the Internet. It also delivers an extensive literature survey on application-aware networking and the relationship with the networks QoS and bandwidth management.
- Propose low delay and packet loss queue management and bandwidth allocation technique and algorithm for multiple aggregated traffic classes.
- The combination of automatic queue management and adaptive weighting scheduler algorithm based on the applications aggregate traffic classes behavior.

1.7. Thesis Layout

The reminder of this thesis organized as follows:

Chapter 2 Presents the theoretical and technical backgrounds. An overview of data center network, its topology and traffic properties, followed by the Internet traffic controlling technique such as traffic classification, conditioning and dynamic resource allocation. We also present the IETF QoS standard models. The related Works in Chapter 3, we start by introducing the problem domain and then introduce, and analyze key research paper on our research domain.

The design of application aware bandwidth utilization in DCNs is described in Chapter 4: We propose a heuristic QoS architecture with appropriate traffic control or scheduling algorithm which has a capability to prioritize resource provisioning for business critical application traffics.

In Chapter 5, the evaluation methods and its metrics are defined. Then the output of simulation validated by analyzing the result of each compared algorithm with different test scenario.

Finally, chapter 6 comes with the conclusions of the study and point out some direction for future work. The references and appendixes are also attached at the end of the thesis.

2. Data Center Networking

2.1. Overview

A data center refers to any large, dedicated cluster of computers that is owned and operated by a single organization [1]. Data centers of various sizes are being built and employed for a diverse set of purposes. On the one hand, large universities and private enterprises are increasingly consolidating their IT services within on-site data centers containing a few hundred to a few thousand servers. On the other hand, large online service providers, such as Google, Microsoft, and Amazon, are rapidly building geographically diverse cloud data centers, often containing more than 10K servers, to offer a variety of cloud-based services such as Email, Web servers, storage, search, gaming, and Instant Messaging. Conventional DCs are modeled as a multi-layer hierarchical network with thousands network node [2].

Data Center Networks (DCN), are built typically based on the three tier, hierarchical, tree based architecture. It has a core layer at the root of the tree which is responsible to high speed switching, an aggregation layer in the middle and edge or access layer at the leaves of tree. There is an existing multiple redundant paths among two hosts in the network to guaranty packet delivery in case of device failures. Today's data center network contains thousands of compute nodes with significant network bandwidth requirements and its topology plays a significant role in determining the level of failure resiliency, ease of incremental expansion, communication bandwidth and latency.

2.2. Topology

Network topology refers to the physical and logical layout of a network, it defines the way different nodes are placed and interconnected with each other. The physical topology emphasizes the physical layout of the connected devices and nodes, while the logical topology focus on to the logical layout of a network and describe how the data is transferred between these nodes. Most of physical datacenter topologies designed to overcome the challenge of scalability, agility, fault tolerance, aggregate bandwidth, automated naming and address allocation, and backward compatibility. Some of the data center network topologies that have been proposed over the time are Dcell which is server-centric hybrid DCN architecture where one server is directly connected to many other servers, Bcube, MDcube, Scafida, HCN & BCN, Jellyfish, F10, facebook FAT Tree and Fat Tree which allows for high bisection bandwidth using a large number of less expensive switches allowing support for a large number of hosts at much less cost [11, 12].

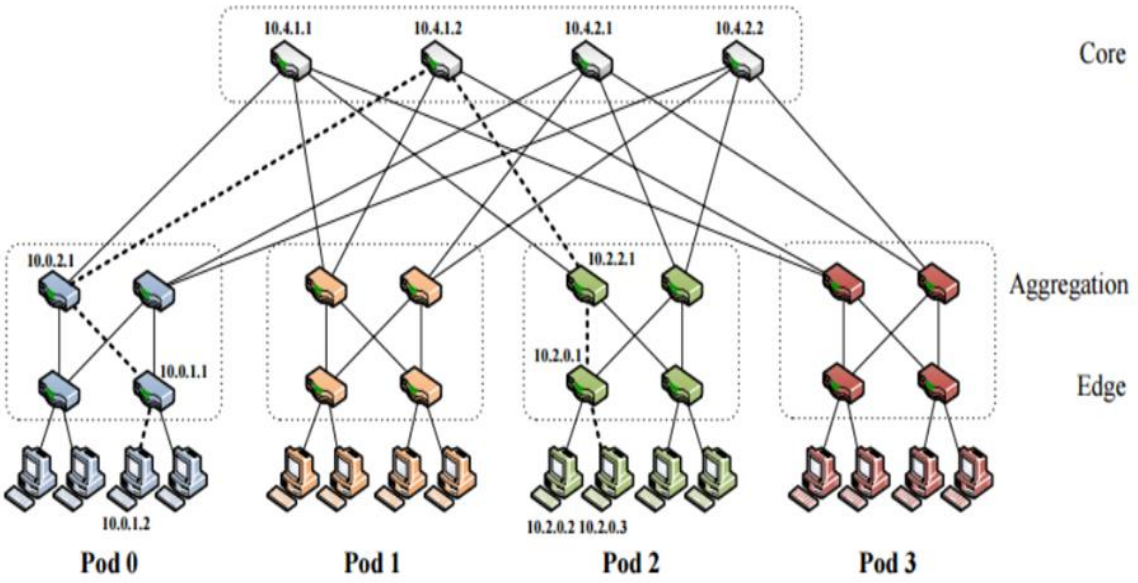

Figure 1: Google fat tree topology

2.3. Traffic Property

In recent years, various sizes of data centers have been built by enterprises to run a variety of applications. These applications have dissimilar communication pattern and need different traffic treatment. Before applied any traffic control mechanism, we must understand the behavior of the traffic in the data center. First, there are large numbers of short flows in datacenter networks. Second, short congestion periods are common across many links. Third, the datacenter traffic has significant variability. These unique characteristics make datacenter traffic remarkably different from any other network traffic. Hence, a practical bandwidth allocation should not only achieve the basic requirements, but also dynamically adapt to the traffic patterns in datacenters [13].

To identify the data centers traffic characteristics using a top-down analysis, starting with the applications run in each data center, the application placement which influence the traffic characteristics then drilling down to the applications' send and receive patterns and the resulting link-level and network-level performance. A better understanding of these issues can lead to a variety of advancements, including traffic engineering mechanisms to improve available capacity and reduce loss rates within data centers, mechanisms for improved quality-of-service, and even techniques for managing other crucial data center resources.

2.4. Bandwidth and Bandwidth Management

In data communication, Bandwidth is the capacity of the data transfer of the communication line, typically measured in number of bits per second [9]. The actual amount of data transfer through the connection link which is called the throughput, it is less than the capacity of the link, and also Bandwidth is inversely proportional to latency, the amount of time the packet takes to travel from one point on a network to another. At the current time, bandwidth is a finite and important network resource, even for local access area (LAN) networks. So, it is very important to manage the bandwidth, to Save it and to use it efficiently by implementing bandwidth management.

Bandwidth management is the method of assessing and controlling the communications (traffic, packets) on a network link, to avoid filling the link to capacity or overfilling the link, which leading to a network congestion and poor performance of the application. It is another important way to ensure the QoS by considering that there is a tight relationship between bandwidth and QoS. It is easier to get good QoS if the bandwidth is large and bandwidth management may support traffic control. When the bandwidth is abundant, traffic control will be easier, because there is no need to use complicated traffic control methods. There are a lot of bandwidth management mechanisms and techniques such as traffic shaping (rate limiting), Scheduling algorithms, Congestion avoidance and Bandwidth reservation protocols.

2.5. Traffic Control Technique

2.5.1. Classification

Network Traffic Classification is important to control network traffic flow and apply QoS in a data center network. It is very essential for Internet Service Providers (ISPs) and commercial enterprises to manage the overall performance of a network. Traffic classification is the first step to identity and classify unknown network traffic. Network Traffic Classification plays a very vital role in network security and management, such as Intrusion Detection and Quality of Service (QoS) [14]. In the last two decades, researchers proposed many network traffic classification techniques. The main three traffic classification techniques are Port-based Technique, Payload Based Technique and Machine Learning (ML) techniques [14,15, 16].

A) Port-based Technique

Port-based traffic classification is combined in most hosts, networking devices and software. In this technique, a classification of network applications is performed using the well-known ports number registered by the Internet Assigned Number Authority (IANA). It is an efficient and effective approach to identify the traffic if the application protocols using their standard port numbers. Some well-known application port from IANA as shown in Table 1 [39].

Table 1: known applications ports

Assigned Port	Application
20	FTP Data
21	FTP
22	SSH
23	Telnet
25	SMTP
53	DNS
66	Sql-net
80	HTTP
110	POP3
123	NTP
161	SNMP

Since the rise of peer-to-peer (P2P) applications such as eDonkey, Kazaa, BitTorrent, OpenNap & WinMx, Gnutella etc. that tend to use arbitrary port numbers in order to avoid detection and filtering, the port-based approach has been proved to become gradually inaccurate. Thus, this technique does not provide good classification accuracy results. Moreover, this technique fails due to the new application using dynamic port number to escape being detected.

B) Payload-Based Technique

In this method, a portion of the captured packet payload data is examined using deep packet inspection technique (DPI) for matching with predetermined signatures of the application. This is proposed to overcome the inefficiency of port-based traffic classification for P2P application which use dynamic port number to identity traffic in a network. However, this technique also has some problems. The first problem in this technique is that it needs a very expensive hardware for pattern searching in a payload and it also introduced additional delay. The second problem in this technique is that it does not work in encrypted network application traffic. Finally, this approach needs continuous update of signature pattern of new applications.

C) Machine Learning (ML) Technique

This technique is based on data set (Labeled Data Set). In this technique, a machine learning classifier is trained as input and then using the trained sample prediction, unknown classes are classified. There are two main areas in machine learning technique: the supervised and unsupervised learning technique.

Supervised learning technique needs a complete labeled data set to classify unknown classes. It means that the supervised learning technique trains, the model with some labeled data set and then it will produce prediction output in new data samples. Below are the two figures which are discussed in details

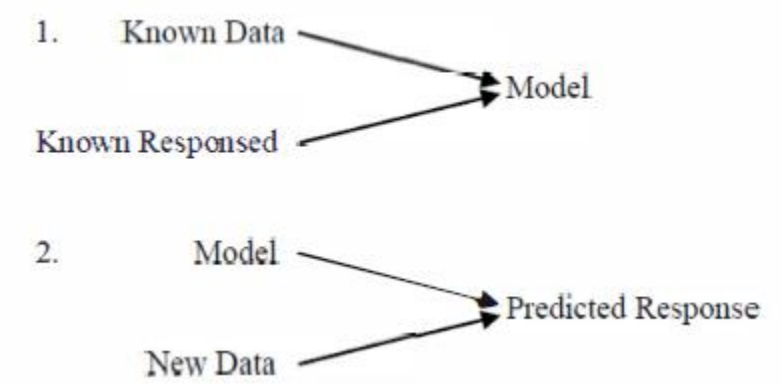

Figure 2: Method description by math works [14]

Unsupervised technique is also called a cluster technique. In this method, there is no need of complete labeled data sets. Unsupervised is a type of machine learning. Thus the result output of machine learning training does not identity or classify instances in predefined classes.

2.5.2. Traffic Shaping

We can improve network performance by making sure that it adapts to required policy rules and profile [3]. Traffic shaping can be done either at the host, or in the network by the edge device. Traffic shaping at the host can be implemented using a buffer and a rate controller. The main issues are the rate control mechanism, shaper delay and delay variation, and the shaper buffer size at the server. The rate controller fixes the outgoing data rate which should be consistent with the bandwidth available from the network. The shaper needs a large buffer for accumulating the incoming burst Stream. However, if the outgoing rate of the shaper is low, a large shaper buffer may result in long delay variation. Therefore, there exists a trade-off between the buffer size, shaper delay, and outgoing rate of the shaper [23].

Traffic shaping limits the data transmission to specific configured rate. As mentioned, the rate of transfer depends on these three components that constitute the token bucket: burst size, mean rate, measurement interval. The mean rate is equal to the burst size divided by the interval. When traffic shaping is enabled, the bit rate of the interface will not exceed the mean rate over any integral multiple of the interval. in other words, during every interval, a maximum burst size can be transmitted. Within the interval, however, the bit rate may be faster than the mean rate at any given time [24].

2.5.3. Traffic Scheduling

Traffic scheduling disciplines such as rate-controlled strict priority and weighted fair queuing scheduling allows individual connections to obtain guarantees on bandwidth, delay, and delay jitter. Packets from different service or applications should be scheduled according to one of these factors. These sources should reserve enough resources to meet their performance requirements. The basic purpose of the scheduler is to allocate the shared resources among different types of incoming packets. Based on the type of the scheduling algorithm and the traffic characteristics, certain network performance can be computed, which can then be used by the network to provide end-to-end QoS guarantees. The QoS guarantees provided by the network are greatly impacted by the nature of the scheduling mechanism. There are so many network scheduler, some of them are

strict priority, Round Robin, weighted round robin, and Bounded Arbitration Algorithm (BAA) etc. [20].

2.5.4. Buffer Management

Queue management is a mechanism used to in the current Internet to prevent network congestion by provisioning the router and deciding to drop packet at a given congestion period of time. It can be used to control the buffer allocated and decide to dropped packets based on their precedence. The queuing discipline affects the delay experienced by determining how long a packet waits to be forwarded. There are various queuing disciplines such as Drop Tail, random early detection (RED), adaptive RED (ARED) is a simple queue management algorithm used by network controller in a network equipment to decide when to drop packets [34,35].

Drop Tail is a Passive Queue Management (PQM) algorithm which only control a maximum length for each queue at router. In Drop Tail, the traffic is not differentiated to decide which packet to drop first or when, it only uses first in first out algorithm and the newly arriving packets are dropped until the queue has enough room to accept incoming traffic. The Drop Tail buffer management forces to choose large buffer for high utilization or small for low delay requirement. The other known buffer management algorithm is the RED algorithm; it manages the queue in a more active manner by randomly with based on the minimum and maximum threshold values. RED monitors the average queue size, and checks whether it lies between some minimum threshold and maximum threshold. If it does, then arriving packet is dropped or marked with a probability p value which is an increasing function of the average queue size. If the average queue size exceeds the maximum threshold, all the packet arrived will be discarded. In order to improve the fairness and constancy, several improved algorithms have been developed, including Weighted-RED, Adaptive-RED, RED with In/Out (RIO) [34, 37,38].

2.6. Quality of Service

2.6.1. Overview

Quality of Service (QoS): as ITU manual [25], QoS is “the ability to segment traffic or differentiate between traffic types in order for the network to treat certain traffic differently from others”, and in the ISO definition, quality is defined as “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs” (ISO 8402). In order to get smooth systems integration and applications performance improvement, the network must apply appropriate QoS

technique. Quality of Service (QoS) implies the statistical performance guarantee of a network system. It may be defined by some parameters such as average packet loss, average delay, average jitter (delay variation) and average throughput. It needs to control the network traffic and to manage the available resources. QoS is now a goal for building and managing the networks [26, 30]. To support Internet Qos IETF group has define two types of architectural model: differentiated services, and integrated services. Intserv provides end-to-end guaranteed or controlled load service on a per flow basis, while Diffserv provides a service differentiation traffic among aggregate traffic class [21].

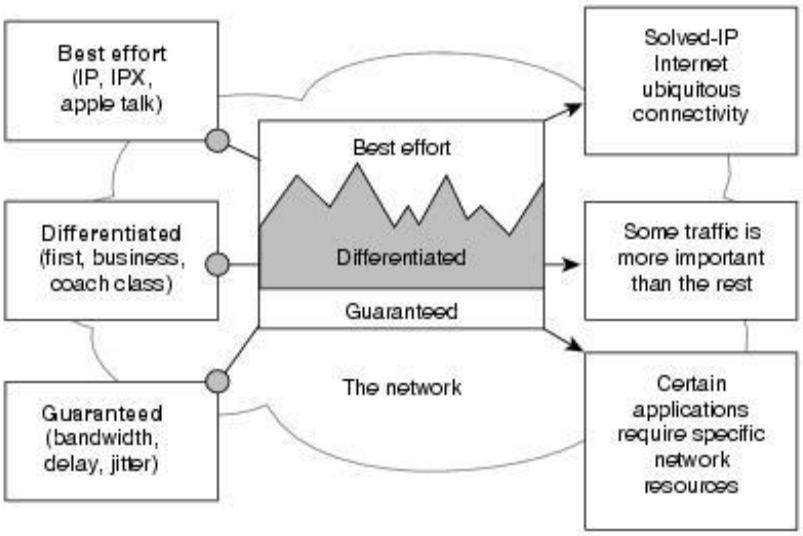

Figure 3: The three levels of end-to-end QoS [40].

2.6.2. Integrated Service (IntServ)

The Integrated Services (Intserv) architectural model provides a means for the delivery of end-to-end Quality of Service (QoS) to applications that need guaranteed or control services over a heterogeneous network. To support this end-to-end model, the Intserv architecture must be supported over a wide variety of different types of network elements [26]. IntServ expresses in three types of services:

- Guaranteed Quality Service: For applications with rigid end to end delay bounds provides this type of guaranteed service.
- Controlled Load Service: For applications which are not need Guaranteed Quality Services but requiring higher qualities than normal best effort services.
- Best Effort Service: Services provided for best delivery as of today's Internet.

IntServ architecture uses Resource Reservation Protocol (RSVP) as its standard protocol to achieve its end-to-end signaling. In RSVP, resources are reserved at each router along the path between sender and receiver through explicit signaling for a flow that demands QoS. It is a means communication for applications to request the network a reserved bandwidth. It is used by hosts to obtain specific qualities of service from the network for particular application data streams or flows, it is also used by routers to deliver quality-of-service (QoS) requests to all nodes along the path of the flows and to establish and maintain state to provide the requested service. The Source and Dentation hosts in the RSVP are exchange the PATH and RESV message, the RSVP source sends a PATH message. When it is received by the destination, if it wants to make a reservation for the particular RSVP flow, it responds with a RESV message and it traverses the reverse path back to the sender. Otherwise, a RESV ERROR message is delivered and is sent back to the receiver. RSVP uses the routing table in routers to determine routes to the appropriate destinations. When a receiver sends a RESV message, the message tells how the reservation has to be treated in relationship to the sender, this set of options are classified as Distinct and shared reservation styles which also control the sender with another option Explicit and wildcard [22].

2.6.3. Differentiate Service (DiffServ)

In the differentiated services architecture, the traffic classification and conditioning is done at the edge of a network, and assigned to different behavior aggregates class which are identified by a single DiffServ code point. Within the core of the network, packets are forwarded according to the per-hop behavior associated with the DiffServ code point [28]. It is a multiple service model that can satisfy different QoS requirements. However, unlike the integrated service model, an application using differentiated service does not explicitly signa1 the router before sending data. For differentiated service, the network tries to deliver a particular kind of service based on the QoS specified by each packet. This specification can occur in different ways. For example, using the IP Precedence bit settings in the packets or source and destination addresses. The network uses the

QoS specification to classify, shape, and police traffic, and to perform intelligent queuing. A forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate group or per host behavior (PHB) group which are defined in terms of behavior characteristics relevant to service provisioning policies, the two PHB group are: -

- Assured Forwarding (AF) PHB: it is a means for offering different levels of forwarding assurances for IP packets received from a customer DS domain. In general, four AF classes are defined, where each AF class is in each DS node allocated a certain amount of forwarding resources (buffer space and bandwidth). However, you can use the other class if you need.
- Expedited Forwarding (EF) PHB: it is providing a certain configured rate guaranteed end-to-end QoS for low delay, low jitter and low loss services for aggregate flow.

In this Chapter, the data center network describes with several types of topology and traffic characteristics. The bandwidth is one of the main scare resource in the DCN, specially at the distributed and core layer of the network links where the traffic flows from many switches on the top of the rack (ToR) become aggregated. To manage precisely the scarce network resources, it is necessary to implement some traffic control techniques and such techniques covers the traffic classification, shaping and scheduling methods. The two IETF QoS model also discussed in this chapter, these model standardized the traffic control technique which helps to share scarce network resource by reserving or controlling the resource to the single application traffic flow or aggregated applications traffic flows.

3. Related Work

Data Center networking provide an essential role to communicate a mixed variety type of computing devices. These computing device share network resource to exchange information in the underlying Data Centers network. It has been observed that when a massive traffic existed on a limited resource, there is a possibility the network become congested, and if there is not a mechanism to control the traffic and not prioritize critical applications traffic, it become the cause to interruption of a process run on integrated systems and the reason to worst application performance.

Many researches [5,6,7,8] have done to address the problem of bandwidth utilization according to the requirement of the application in the data center with a number of new mechanisms to efficiently share data center network among applications. In this chapter, we try to explore such mechanisms and state of the art works, and address theirs advantage and disadvantage.

The paper on [5], the authors starting by assuming that each application can be differentiated, and can be classified in the corresponding application type and focus on the designing of an efficient dynamic bandwidth management framework in a shared Data Center link to provide the in-network bandwidth guarantees for applications in all switches (or nodes) along the routing path and improve the bandwidth efficiency. First they formulate a max-min bandwidth constraint model to reserved bandwidth for each application type and enforced to a minimal value and a maximum value by considering the node Bandwidth capacity and molded the Bandwidth consume and availability matrix for each node. Moreover, when the application used less than the minimal reserved bandwidth, the unused reserved Bandwidth will be borrowed to the other application which request it and update the Bandwidth consumed and available matrix on the node. They also propose bandwidth-return-preempt (BRP) bandwidth management model by combined the Borrow, Return and Preemption algorithms, and design a BRP method to allocate bandwidth to each incoming request, to return after used and it allow the request from an application which has higher priority to preempt the bandwidth from low-priority application. Finally, they conduct comprehensive simulations to evaluate their proposed BRP method by measuring Bandwidth utilization performance and request success rate. They achieve a higher bandwidth utilization rate than the fair allocation method and they showed clearly that the BRP method can achieve a higher request success rate across all nodes. However, they used shortest-path-first algorithm to select the

route which might be a reason for congestion if there is a massive traffic with the same source and destination address.

Most of the recent proposed solution based on emerging technology such as network virtualization and SDN which have centralized and holistic view of the network. In [6], they argue the current SDN controller implementations are not designed to be fully application-aware and their aim are to consider one step further that higher scalability would be achieved if they map applications to network resources under the application-aware SDN circumstance and the resource management database in the management plane should record the decision made by the control plane for applications. They propose the network devices to make up a resource pool in SDN networks and the controller to consumes resources in the pool by deploying flow table entries to network devices. They introduce another concept naming path, to represent different connections running on the same route and create the relationship among application, route and path by tripartite graph G= (v^1, v^2, v^3, E). Where, vertices in V1, V2 and V3 represent applications, routes and paths respectively.

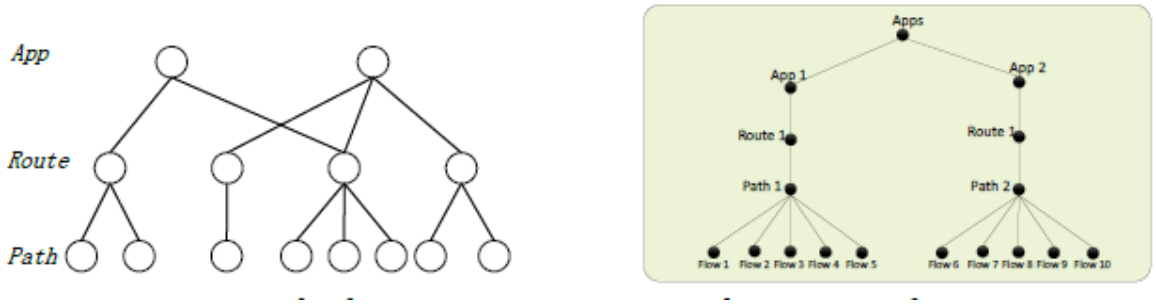

Figure 4: Tripartite graph of mapping and implementation of mapping [6]

They use OpenDaylight controller. The mapping module maintains the proposed tree structure and provides APIs to manipulate data in the tree. Finally, they try to identifying the gap between the current SDN design concept, and true application-aware networking resource management and showed the mapping model could bring performance enhancement in time cost.

As multidisciplinary European research project NEPHELE in [7], they proposed six main applications and controller core services, together with the REST APIs and there are also three potential architectural models design for the control plane architecture. The heart of their research is the SDN controller, which hosts the necessary DCN core services and programming applications required for the seamless operation of the whole NEPHELE architecture. The objective of the framework is to orchestrate the application network requirements to the SDN controller and to provide a monitoring service that can compare the real time performances to what has been decided, promised or expected by the applications initially. The outcome of their research which has enhanced the network capabilities to collect the application requirements and perform resource allocations and optimizations accordingly.

In other study [8], they propose an application aware traffic engineering (TE) solution that cooperates with deep packet inspection (DPI) which could help to identify the application traffic type. The proposed system contains the Traffic Scheduler which is responsible to routing decisions, Utilization Monitor to collects and computes the underlying network statistics, and Classifier that provides an SDN controller with application aware on incoming traffic and flow treatment after traffic identification performed by DPI on the data plane. They designed a system based on SDN concept of control and data plane separation to improve Quality of Service (QoS) of certain network traffic efficiently, and the SDN application controller handles the traffic as a network flow and performs flow treatment depend on the traffic identification performed by DPI on the data plane.

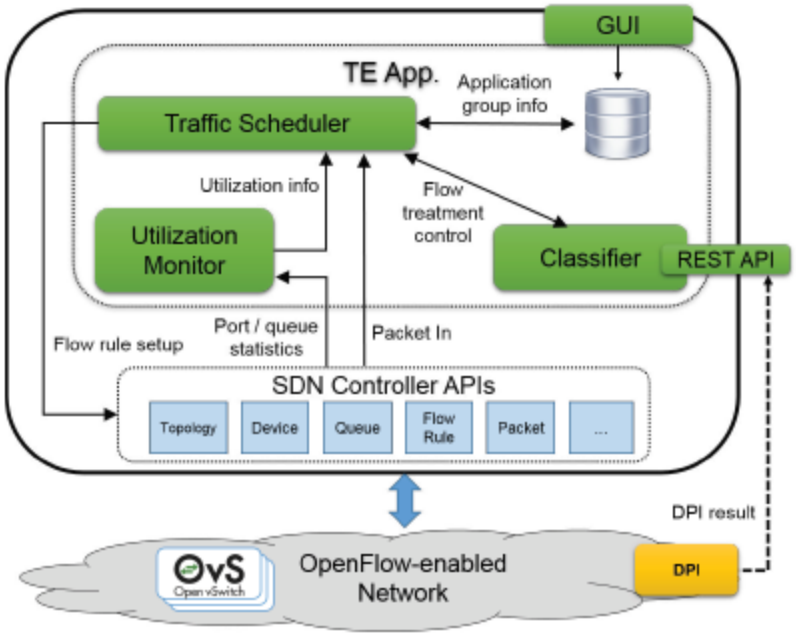

Figure 5: System design [8]

They also proposed multi-queue approach in a switch port as a type of bandwidth guarantee queue that avoids starvation of lower priority queues in congestion time. For paths with the same hop count, port based algorithm considers current capacity of each egress port to achieve overall load balancing. The following diagram shows the algorithm how unknown traffics handles and temporary flow rules install to prevent delay and packet drop until the traffic classified and permeant flow rule installed.

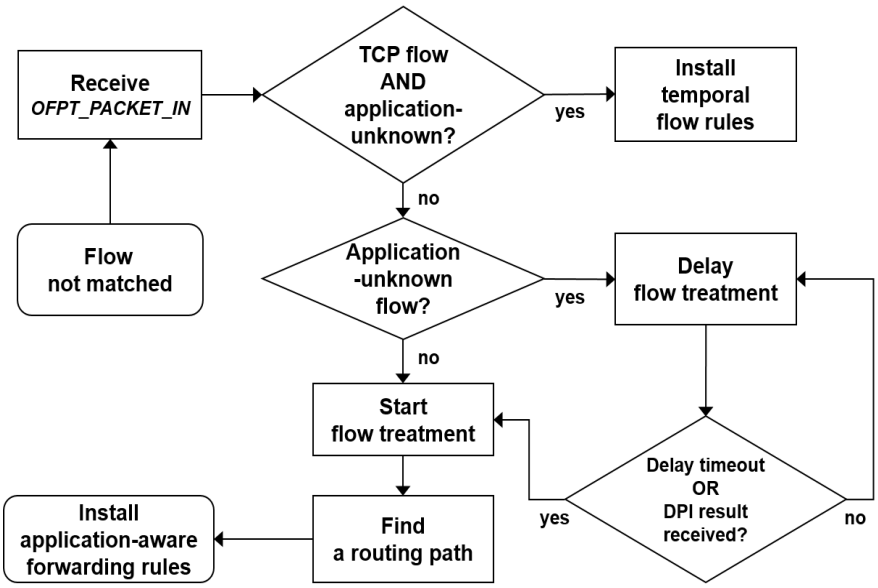

Figure 6: Overall process [8]

They validate their design using Mininet simulation testbed, had been used iPerf traffic generator tool and YouTube as a source of application traffic with three scenarios: baseline approach without TE, TE by link utilization, and TE by both link utilization and using DPI results. the total throughput increased with congestion due to path sharing in first scenario. In second scenario, the traffic could flow through different paths for link utilization and the throughput increase for all traffic. Moreover, YouTube traffic throughput increased because of YouTube traffic was given higher priority than iPerf traffic. As a result, YouTube traffic in scenario 3 increased dramatically. However, the time required to install a flow rule for the first time was higher and to implement the DPI in Data center for server to server communication is difficult because of the traffic has multiple entrance point to the network.

The big data applications [17] [18] have a high amount of data transfer in the data center network, when the default MapReduce application running on the traditional IP networks, it can cause a network bottlenecks. Both study based on SDN solution and aim to accelerate the big data Hadoop application job completion time. They design Hadoop application to send information about the size of the data and other network communication pattern to the SDN controller periodically. This way, the Hadoop controller has the network bandwidth requirements of every shuffling flows and can pass this information to the SDN controller. The application-aware SDN routing [17] system component which include FatTreeManager: which gather information about Fat Tree network

topology, LinkLoadingMonitor: to keeps monitoring the load of all links in the network and provides the loading information to Routing Component, MapReduceManager: to maintains two tables to record the shuffling information, and the state of path allocation and the "RoutingComponet" which allocation the routing path are designed. their Application-Aware SDN routing scheme outperform by 20% than equal cost multi path protocol and outperforms the Spanning Tree scheme and the Floodlight OpenFlow controller by up to 70%. The paper on [18] also address the network congestion problem caused by identified major traffic patterns of various MapReduce workloads and they demonstrate a concept of AAN using SDN implementation and MapReduce performance improvement. However, both research [17] [18] address the network performance issue on a big data Hadoop application which is unlike to our enterprise application communication characteristics.

Finally, the overview of this chapter shows that almost all researches done on reservation resources to guarantee a network bandwidth for a particular application in the Internet. The emerging of the SDN technology and its programmability, protocol independency easily manageability features helps network administrator to write a program which improves end to end QoS, multi-path load balancing and application-aware routing protocol selection. The steps followed by all the solution have similarity, it includes traffic identification, classification and scheduling based of the application profile. Some of the drawbacks are the extra processing time and complexity introduce by the DPI traffic identification method, and the wastage of resources may happen if the application is not fully used the reserved capacity. For finalizing, the provisioning of guaranteed bandwidth by reserving for a particular application is thought as on the concept of IETF IntServ model, this model has a scalability problem to reserve the resource for a large number of applications and its complexity introduced by the request and reservation signaling process may be a headache for large networks. Our solution follows the DiffServ QoS model which provide different treatment for aggregated application traffic.

4. The Proposed Solution

In this chapter, an application-aware network bandwidth utilization solution model is proposed. The proposed solution is intended to avoid packet loss and delay for real-time or business critical applications traffic when a bandwidth bottleneck happened at the shared links in DCN. The model is based on the QoS architectures mentioned in chapter 2. The solution model design has two parts, the first part proposed the QoS architecture and in the second part we offer the bandwidth utilization model. The bandwidth utilization modeling includes the comparison of different algorithms which fits to the proposed solution.

In most of the researches conducted on application-aware networking, it is well-thought-of for the Internet service provider (ISP) resource provisioning. Those thinking are based on reserving bandwidth for specific real-time application or providing different level of bandwidth capacity per affordable price. Whatever the solutions the principle is laying on the two IETF QoS architecture models, the Integrated or differentiated QoS, the IETF also combined the two models to provide end-to-end solution by implementing IntServ at the edge network and DiffServ at the core network parts [26]. Among these the most scalable QoS mechanism, the DiffServ model are suitable for the DCN, it has been chosen for providing good QoS in multi-level classified applications traffic. In addition, the DiffServ model is working without modification of the existing applications, and it doesn't depend on any signaling protocol and it should be avoiding micro flow state within core nodes [28, 29].

4.1. Proposed QoS architecture

The proposed QoS model is based on the differentiated services architecture, where traffic entering a network is classified and possibly conditioned at the edge or boundaries of the network, and treat according to the behavior of the assigned aggregates class. Our architecture defines in a single DS domain; the DS domain is including all networks under the same network administration, whatever it is the Edge or Core devise in the DCN. The Edge network devices include the server farm layer three switch or the devices that use as a gateway for the servers which hosts the applications. The Edge nodes act both as a DS ingress node and as a DS egress node for different directions of traffic flows. A single domain DiffServ architecture which illustrates the QoS applied parts with the edge and core devices is displayed in Figure 7.

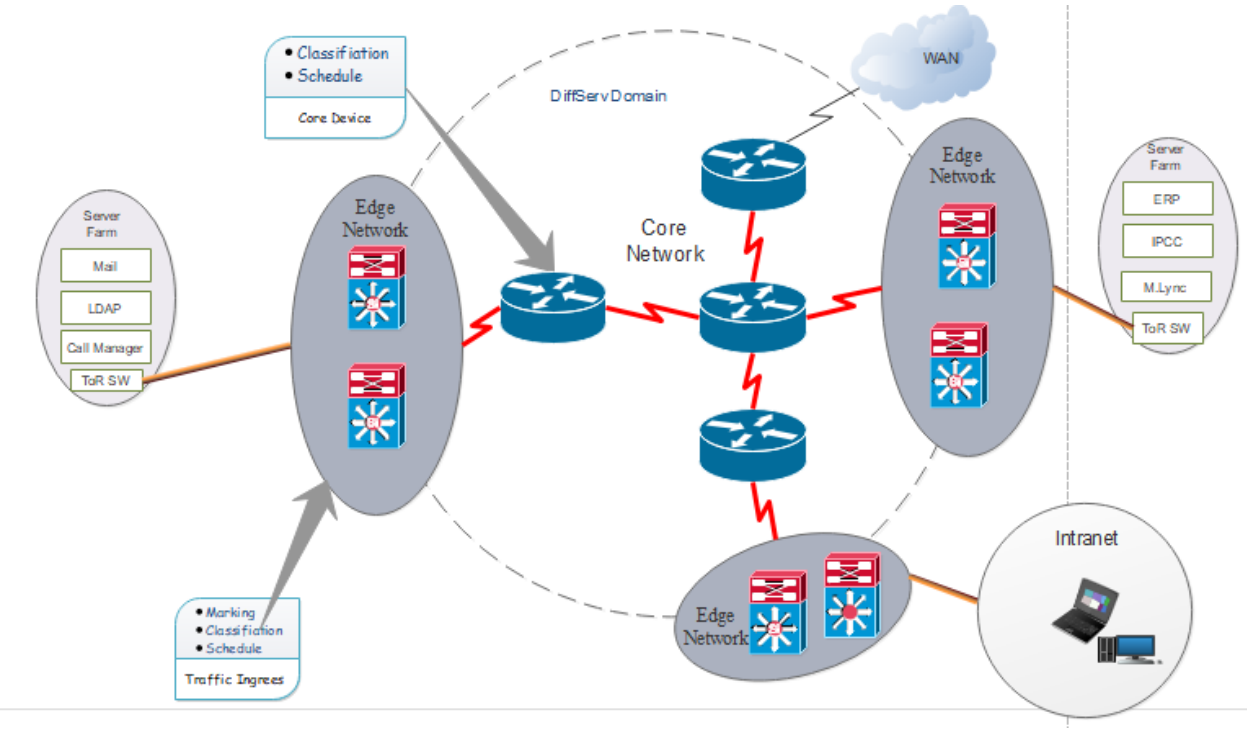

Figure 7: DS single domain QoS illustration architecture

4.1.1. Edge Network Part

All server connected to the network through the ToR switches, these layer two devices have high data rate links per port which dedicates for each system. So, the distribution layer devices, where exists between ToR switches and core devices, or a server farm gateway on the actual network in the data center considered as an Edge Network which is an ingress point for traffic enter to the DS domain. The main functions of QoS applied at the edge devices are:

- Identifying traffics by using multi-field (MF) classifier. Packets identified based on one or more header fields value, such as source address, destination address, protocol ID, source port and destination port numbers. Most of application-aware network used deep packet inspection (DPI) to classify traffic by application signature in the payload of the packet or by implemented ML technique. However, in our case all applications running on the server in DC are known easily by source or destination IP address and port number without introduced additional burden by deeply inspection of the packet payload.

- Marking packets with a code point that reflects the desired level of service. The output of the MF classifier packet header set with a 6 bit DSCP value. The service class to DSCP code mapping is described in Section 4.2.1.
- Queuing and scheduling the packet according to the DSCP value of the header.
- For EF class, the traffic shaping or rate control applied to limit the maximum resource utilization. However, the traffic conditioning at the edge is not cover in this work.

4.1.2. Core Network Part

The core network contains the routers or multi-layer switches node which are only connected to other network device, not servers. Core devices handle the aggregate flows. The main responsibilities of core devices are:

- Classify the packets using multi-behavior classifier by examining the header of the packet DSCP value that marked at the edge network.
- Queuing and scheduling the output of classifier packets.

4.2. Proposed Bandwidth Utilization Model

The proposed application-aware network bandwidth utilization solution mainly focusses on to prioritize resource allocation to real time and business critical applications with a constrain bandwidth. In our case, the scarcity of bandwidth between two nodes that lead to bottleneck problem is a motivation to this thesis. The main objective is not to prevent the issue or its causes, instead of that we want to provide the solution which answers the questions how to handle resource sharing on this type of incident happening. In our design we show the key parts of the proposed solution in Figure 8.

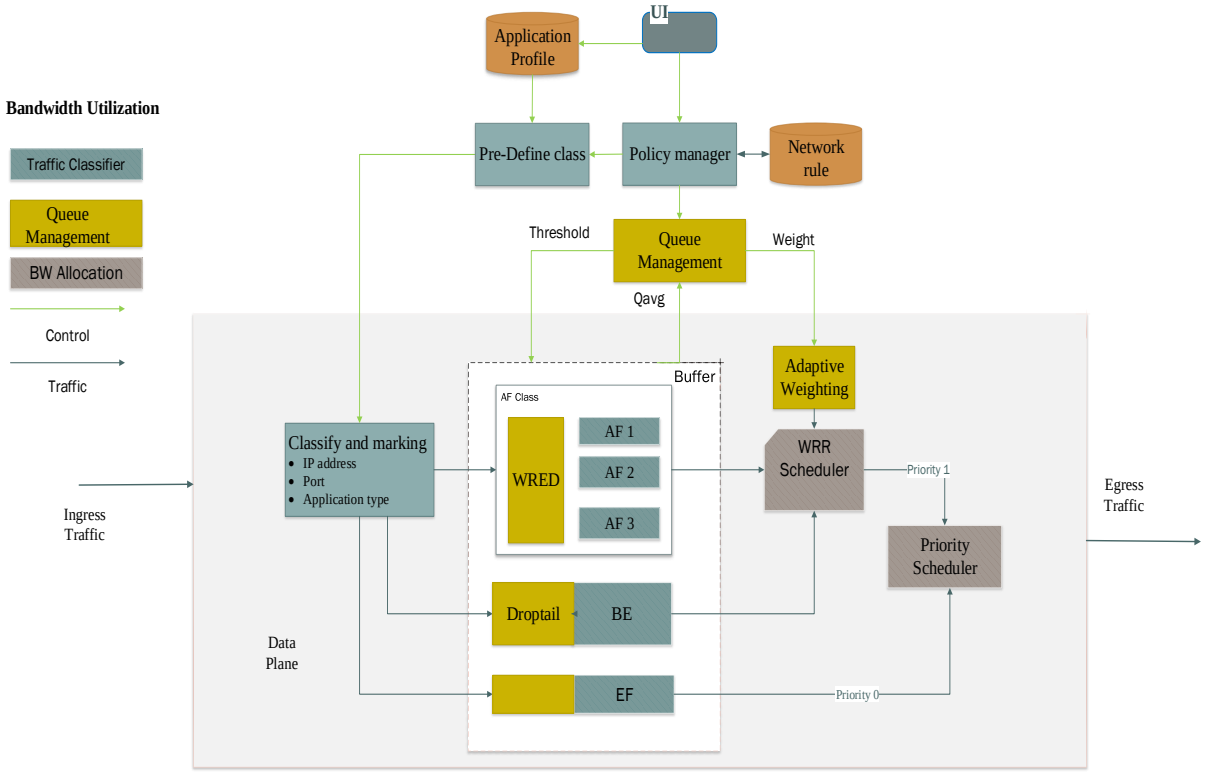

Figure 8: Application aware bandwidth utilization design model

In the suggested application aware bandwidth utilization design model shown in the above diagram, the solution has been divided in three main parts according to the main functionality of the PHB characteristics of QoS and which have direct relationship to the resource utilization of a router or node. Those main parts of the solutions are:

1) Application traffic Classification
2) Buffer (queue) management
3) Bandwidth allocation

4.2.1. Applications Traffic Classification

There are so many types of application exist in the data center, and those applications traffic categorized as per their interconnection behavior or network requirement such as delay, jitter and packet loss. In addition, there are also grouping based on the company policy such as business critical and mission critical applications. In the proposed design the application profile, policy manager and pre-define class are the module which help us to classify traffic per aggregated application group.

Application profile: - it contains all applications information in the DC which includes the physical IP address of the server, the virtual IP address of the application or services (if it uses), the port number (if it uses static), the maximum delay and loss occur during transmission.

Policy manager: - the company network access policy (access list, access map), traffic control and traffic classification rule are managed by this module.

Pre-define class: - An independent IP packet forwarding class with it DSCP codepoint is defined. Within each class, an IP packet is assigned one of two different levels of drop precedence.

Classify and Marking: - At edge device the multi filed (MF) classification is used and then assign the DSCP value to the packet according to the corresponding class. On the core device the packet that already mark at the edge classified using aggregated behavior classifier. The classified packets queue on the output interface buffer and treat based on the priority level of the class..

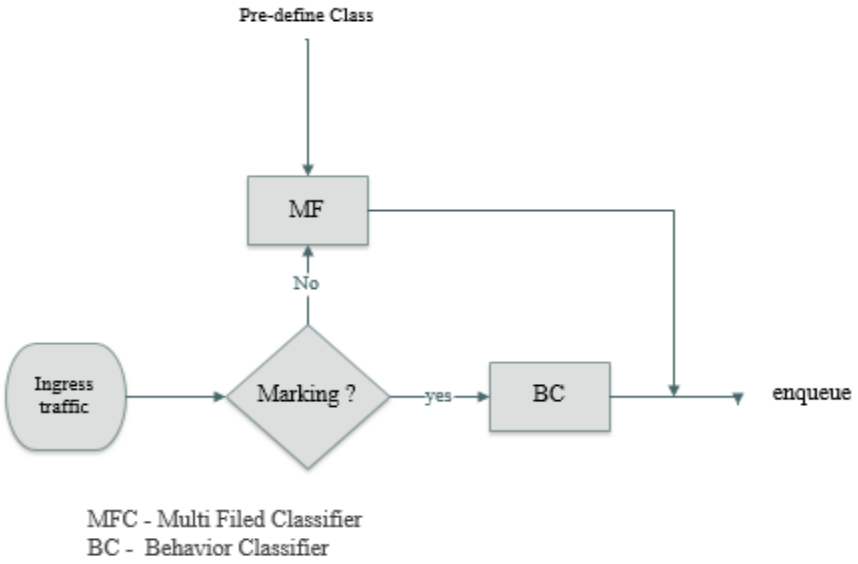

Figure 9: Traffic Classifier

First, the application profile database created and it contains each application information with its network requirement. Secondly, we classified those applications in to three groups based on their bandwidth requirement types, those are the application which wants nearly guaranteed service, the applications which needs controlled service and the default one for best effort delivery. In the pre-define class module based on the information collected from the application profile data store and

the rule from the policy manager, define the service class which is used by the MF classifier to mark the undefined traffic.

The AF service contains three general use class with two level drop precedence (high and low). Applications which do not need guaranteed bandwidth and a real-time communications assigned to these classes according to their behavior. The drop precedent level within each class categorized into critical and non-critical applications based on the company business or mission criticality.

On Table 2, we show all applications category, the corresponding DiffServ class and DSCP value assigned to each class.

Table 2: Application category with DSCP code

Application	Service	DS Class	DSCP Code
Real time	Guaranteed	EF	101110
Signaling and network control	Signaling	AF3x	011010
	Network control		011110
Interactive Data	Critical	AF2x	010010
			010100
Transactional Data	Critical	AF1x	001010
			001100
Default		BE	000000

The DSCP code is an eight-bit value inserted in the packet header and only six bit is used for identification of the class or traffic type the other two bit used for congestion notification, the mapping of the code to DS class is based on the IETF recommendation [43].

4.2.2. Queue Management

In multiple class traffic handling, the queue management controls the packet queue time on the buffer based on the acceptable delay requirement of the aggregated class. Each node implements some queuing discipline that manages how packets are buffered until it will be forwarded. Various queuing disciplines can be used to control the buffer memory and decide to dropped packets based on their instruction. One of our objectives is to compare queue management algorithms based on

the previous research result and the feature of the algorithm and align it to our aggregated behavior class. One algorithm may have good performance to all class or different algorithm might be chosen. We compared the Drop Tail, RED and WRED queuing discipline with different scenarios per aggregated service class validate with a simulator in the next chapter.

The default queuing algorithm in most of the nodes is FIFO queuing with Drop Tail discipline. An increasing packet flow will lead to buffer overflow, and the Drop Tail algorithm ultimately drop new arriving packets [34]. Some researcher suggests Drop Tail algorithm to achieve a high forwarding rate, low packet losses, and high bulk throughput. However, the algorithm neither differentiate packets to be dropped nor early detect congestion. For congestion avoidance, the RED algorithm early detection mechanism is suitable. In addition, the RED algorithm is more manageable by configuring with its parameter such as maximum, and minimum threshold value, maximum drop probability and its constant weight [34] [35].

The RED parameter configures with different threshold value per class depend on the acceptable delay and packet loss of the applications. The probability of the packet drops P_X and the average queue length Q_{avg} are calculated using equation 1 and 2 [36, 37].

$$Q_{avg} = \{(1-w_q)\, Q_{avgp} + qw_q, \quad if\ q > 0\} \qquad (1)$$

$$\{(1-w_q)^{m}\, Q_{avgp}, \ otherwise\}$$

where Q_{avgp} is the previous value of the average queue length, q is the current value of the queue length and w_q is the queue constant weight. The average queue length depends on the constant weight value w_q, the current queue length (q) and the previous average queue length (Q_{avgp}). The only configurable parameter in this equation is Wq, but we have not any intention to control the average queue length by readjusting this constant value. Instead we used Q_{avg} as a metric point to mark the packets that will be dropped and to readjust the WRR scheduler weight parameter to configure the next Q_{avg} length. The packet dropping probability is zero when Q_{avg} is less than the minimum threshold as per equation 2, so that we are considering the minimum threshold point as the desired delay of the class that can be tolerable up to the maximum threshold and then we can take it also as the maximum time the packet to be late.

$$Px = \{0, \quad \text{if } Qavg < Tthr(min)\}$$

$$\{Pmax*(Qavg-Tthr(min)/(Tthr(max)-Tthr(min), \quad Tthr(min) < Qavg < Tthr(max)\} \quad (2)$$

$$\{1, \quad \text{if } Qavg >= Tthr(max)\}$$

Px is temporary packet drop probability varies from 0 to Pmax, the maximum drop probability. Tthr (min) and Tthr (max) are the minimum and maximum threshold values. Pmax, the maximum dropping probability of a packet.

The proposed queue management algorithm alignment for the EF, AF, and BE PHBs service are described as follows:

EF PHB service: the recommended codepoint for this PHB is 101110 [42]. The IETF recommended queue length is one or two packet sizes, and so, the desired delay and loss of this class is almost zero. It is difficult to configure the RED thresholds and control the queue length of the class. Therefore, the default Drop Tail algorithm for FIFO queue is recommended.

AF PHB service: our solution only includes three of IETF AF service classes and we select AF1, AF2 and AF3 from those service class. Each class has two drop precedent level with its own codepoint and threshold value. The treatment of the application traffic in these aggregated class depend on the assigned threshold value for each sub aggregated class. It helps to avoid packet drops from a business critical system before a non-critical systems packet drop. The queue length of these classes controlled by active queue management scenario. The RED algorithm, one of the active queue management disciplines, and it can be controlling the packet drop by configuring different minimum and maximum threshold value per class. So we recommend WRED to the AF classes. The threshold parameter for each class is planned as shown on Table 3. The threshold parameters sets as a variable which can be configured to get the desired and acceptable results according to the class requirement. Where Thrmin is the minimum number of packets, Thrmax is the maximum number of packets and Pmax is the maximum drop probability assigned to the specific aggregated traffic. The first digit of the variable represents the AF class of service, and the second digit represents the sub aggregated traffic behavior per class which classify the critical and non-critical systems which have the same networking behavior.

Table 3: AF class with WRED per class threshold parameters

Traffic Class	Queue Algorithm	T(min)	T(max)	P(max)
AF1	RED	Thrmin11	Thrmax11	Pmax11
		Thrmin12	Thrmax12	Pmax21
AF2	RED	Thrmin21	Thrmax21	Pmax21
		Thrmin22	Thrmax22	Pmax21
AF3	RED	Thrmin31	Thrmax31	Pmax31
		Thrmin32	Thrmax32	Pmax31

BE Class: is the default class. The traffic which are not assigned to the EF or AF service class are set to this class by default. The main target of the BE service is to deliver of low priority applications traffic as much as possible without concerning the delay of the packet. The Drop Tail queue algorithm is recommended for this class.

4.2.3. Bandwidth Allocation

The main objective of application-aware scheduling is to answer the question which applications traffic must have priority to transmit and which one's packet must be dropped before critical applications traffic affected when a bottleneck incident happens. The differentiated traffic class shares the buffer to wait until it gets the access of the link and there must be a scheduling rule to allocated a transmission time to each traffic class. There are so many scheduler discipline; the default one without any traffic control is first come first serve (FCFS) which serve packets based on their arrival time. For our case we reviewed the previous researchers work, and then we had a preliminary comparison test between the SP and WRR and proposed the combination of the two scheduling algorithm according to the requirement of the aggregated application traffic class as shown in figure 10.

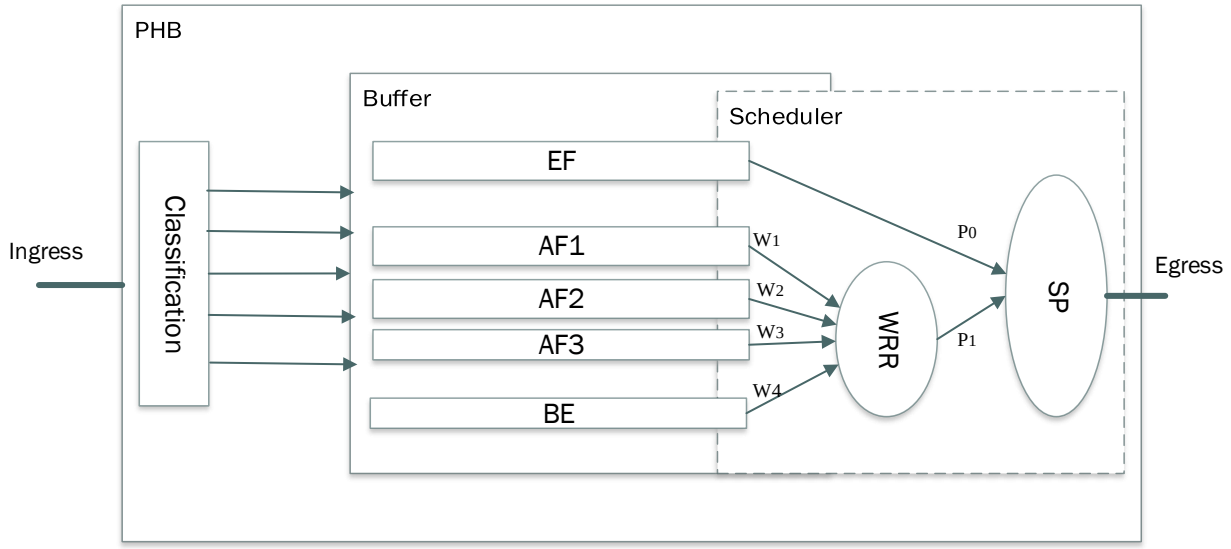

Figure 10: PHB queuing and scheduling logical design

SP Scheduler: The SP scheduling using p0 and p1 priority level to allocated bandwidth for aggregate traffic class. The EF class has high priority than the other, to limit the aggregate bandwidth assign to the EF, we also recommended an input rate control as a traffic conditioner at the edge node.

WRR Scheduler: The AF and BE service classes which have low priority on the SP scheduler shared the unused bandwidth by the EF service class. This allocates bandwidth to AF and BE service class using WRR scheduler. It means that the total capacity (C) minus the consumed bandwidth by the EF class allocated to the reaming class according to their assigned weight. The WRR scheduler configured with parameters w_1, w_2, w_3 and w_4 to share the transmitting time based on the weight value assigned for each aggregated traffic class. The weight assigned to each queue or service class are calculated by considering the minimum delay and packet loss requirements of the applications in the aggregated service class. For optimal allocation of the bandwidth to each class, we propose to adjust the weight parameter proportional to the desired average queue length of each differentiate class.

An adaptive weight scheduling algorithm has better performance in terms of effectively using resource sharing time among the service classes. As an example, if the queue length of the class is less than the minimum threshold value readjusts the weight to minimum allocation for this class and increase the weight of other class or the default BE service class. For our proposed solution, which combine the SP and WRR scheduling algorithm, some researcher on [41] propose a new

dynamic benefit weighted scheduling (DB-WS) algorithm in DiffServ network which provides guaranteed EF service and also ensuring minimal loss in AF and BE service class by improving the WRR scheduling algorithm.

The algorithm [41] retains the sum of EF, AF and BE weight as one where one refers to the total bandwidth of the link through which packets are transferred. The DB-WS algorithm based on WRR scheduler for all traffic service class and the total bandwidth of the link is proportionally divided into EF, AF and BE service traffic according to the weights calculated. However, in our case we combine the SP and WRR scheduler with a little modification on the DB-WS algorithm. The EF service class packets strictly prioritize on the other class packets and we used the DB-WS scheduling idea for the AF and BE service class. The algorithm also used the IETF queue length recommendation for EF class as a threshold value to decide the dynamic weight allocation but on the AF service class doesn't define queue length requirements. So we use the RED queue algorithm threshold variable, and then perform several experiments to assign optimal threshold value which provide maximum performance. Equation 3 used to assign the maximum and minimum bandwidth allocation.

$$\begin{cases} W(min), & \text{if } Qavg < Tthr(min) \\ PAF*(Qavg-Qavgp/(Tthr(max)-Tthr(min)), & Tthr(min) < Qavg < Tthr(max) \\ W(max), & \text{if } Qavg >= Tthr(max) \end{cases} \quad (3)$$

W(min) minimum weight, W(max) maximum weight and PAF is the priority level of AF service class. We take the threshold value as a decision point to assign minimum and maximum weight for dynamic bandwidth allocation.

The algorithm DB-WS also modified as follows:

While (packet arrive)
{classify packet and store in the corresponding buffer
Calculate EF weight
Store EF weight
For each AF_J {
Assign min threshold, max threshold and P (max) on buffer
Compare AF_J average queue length

Compare AF_J queue weight

Store queue weight and average queue length of AF_J calculated at this time in variables

J++

}

Compare BE queue Weight

Schedule packet using SP + WRR scheduler (dynamic bandwidth allocation)

} end

5. Evaluation

In Chapter 4, a DiffServ based DCN bandwidth utilization model for aggregate applications traffic in IP networks has been designed and adapted with per class queuing and scheduling algorithm. The different types of buffer and bandwidth resource allocation algorithm compared based on the minimum packet delay and loss of the aggregated application traffic. In this chapter we evaluate the logical bandwidth utilization of a node or router with a constrain available resource on an OMNeT++ Simulator.

5.1. Experimental Simulator Setup

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. The simulator is open source and the component modules are written in the C++ programming language. It also assembled into larger components and models using a high-level language (NED). It has extensive GUI support, and due to its modular architecture, and the capability to add a new or modifying the existing module easily makes it suitable for the purpose of this thesis evaluation [31].

The topology on Figure 11 reflects a simple logical network in an actual DCN environment. The link bandwidth capacity on this simulation test set to 1:1000 proportional ratio from the actual link capacity (which is 1GB) in the DCN and configured to be 1Mbs Ethernet access links in the simulation where the servers connected to the edge node; four Edge router on the distribution layer which connected the access layer to the core layer with 1Mbs point-to-point (P2P) connection and the core layer link capacity between the two core router have been configured with different load value. The data rate and propagation delay over the link is shown on Table 4.

Table 4: Link information

Links	Link type	Delay	Data rate
LAN	Ethernet	0.1micro	1Mbs
Edge	P2P	2ms	1Mbs
Core	P2P	2ms	Configurable

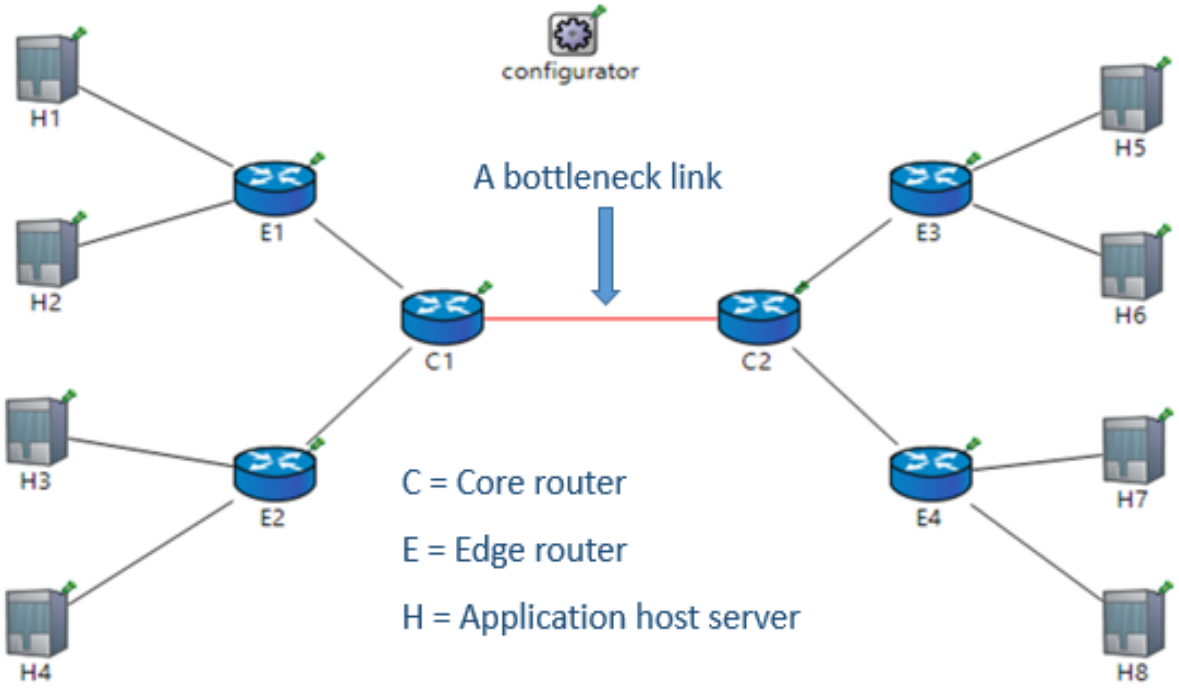

Figure 11: Testing topology

Eight servers access the network through the edge routers E1, E2, E3 and E4. Servers H1, H2, H3 and H4 hosts the application which generate the traffic and servers H5, H6, H7 and H8 sink the applications traffic. The test scenario only applied on one direction traffic flow. The servers access link and the edge link which connected the edge router to the core router bandwidth capacity set too high to eliminate any drop introduce by those links. The load on the link between the core routers C1 and C2 set to full load to create bottleneck problem. The application generated by the host server with basic parameters are shown on Table 5.

Table 5: Applications information

Applications	Servers	Sink server	Packet size	Interval (ms)	Start time	Burst traffic (sec)	Sim time (s)
App0	H1- H4	H5- H8	500B	20	uniform(1s,2s)	Exponential (0.352)	1200
App1	H1- H4	H5- H8	500B	40	uniform(1s,2s)	0	1200
App2	H1	H5 and H6	500B	30	uniform(1s,2s)	0	1200

5.2. Performance Metrics

The experiment conducted on the traffic flows without any traffic control technique, and with DiffServ QoS applied to an aggregated application traffic using different testing scenario to compare different types of queue management and bandwidth allocation algorithms. The result indicates various levels of network performance; theses performance must be measured to deciding how effective the algorithm which is selected by comparing on multi class application flows. A One Way Delay and Packet Loss Metric were used for this purpose [32] [33].

Some measurement assumptions have been taken. Those are:

- The sent packet, packet loss and end to end delay are taken from the measured statistics on server only. This gives a proper estimate of end-to-end delay for the applications.
- The packet loss considered the difference between the number of packet sent at the traffic generated servers and the number of packet received at the sink servers.
- The end-to-end packet delay statistics taken form the receiving servers.
- The queue delay and packet loss at the edge routers and the second core router C_2 are approximately zero.

The sent packet, packet loss, throughput and end-to-end delay are used to express the validation results. The description of these parameters are:

Sent Packet: It is the number of packets generated and sent by the application per specific time interval. For simplicity the size of the packet set to be similar and in some results the number of packet in count changed to kilobyte per second.

Received packet: It is the total number of packets received at the sink servers. We consider it as the throughput of network.

Packet Loss: It is the number of packets lost on the transmission. We consider the packets are lost during the congestion time at the core layer only and no packet is lost anywhere else, then it would be the number of packets dropped at the core router C_1.

End to end delay: It is the total time taken by the packet to reach its destination. It is the combination of the processing time at each node on the path, the propagation time on the links

between those nodes and queue time at the edge and core node. The processing time at each node and the queueing time at edge node are considered to zero.

Not all parameters are representing in the output test result report, the packet loss may present in no of packet or Kilobit per second. The sample test report table format shown on Table 6.

Table 6: Performance metrics

Service class	Delay (sec)	Packet loss (count/Kb/s)	Loss rate (%)
EF	Queue time	Per class	Loss/Sent packet
AF1	Queue time	Per class	>>
AF2	Queue time	Per class	>>
AF3	Queue time	Per class	>>
BE	Queue time	Per class	>>
Application	End-to-End	Per application	>>

5.3. Experiment

5.3.1. Best Effort Service

In the first test scenario, the test performed without applying any QoS or traffic control method. Packets served based on its arrival time as FCFS. The two applications App0 and App1 running on four servers to generate eight applications traffic and send data through the network, the test scenario takes from 300 up to 1200 second simulation time. The output data on Table 7 shows the number of packet lost and delay on the bottleneck link. The end-to-end delay statistics taken from the receiver servers are shows for each applications.

Table 7: Packet loss and end-to-end delay without QoS applied.

Applications ID	Sent Packet (Kbs)	Receive Packet (Kbs)	Drop Packet (Kbs)	Drop Packet (%)	end-to-end Delay (sec)
H1.App[0]	8.667	7.955	0.712	8.216	0.444
H1.App[1]	12.484	12.206	0.278	2.223	0.452
H2.App[0]	8.922	8.218	0.704	7.888	0.426
H2.App[1]	12.481	12.079	0.402	3.218	0.454
H3.App[0]	8.908	8.325	0.582	6.535	0.454
H3.App[1]	12.481	12.025	0.456	3.656	0.423
H4.App[0]	9.217	8.533	0.684	7.423	0.420
H4.App[1]	12.481	11.692	0.789	6.323	0.426

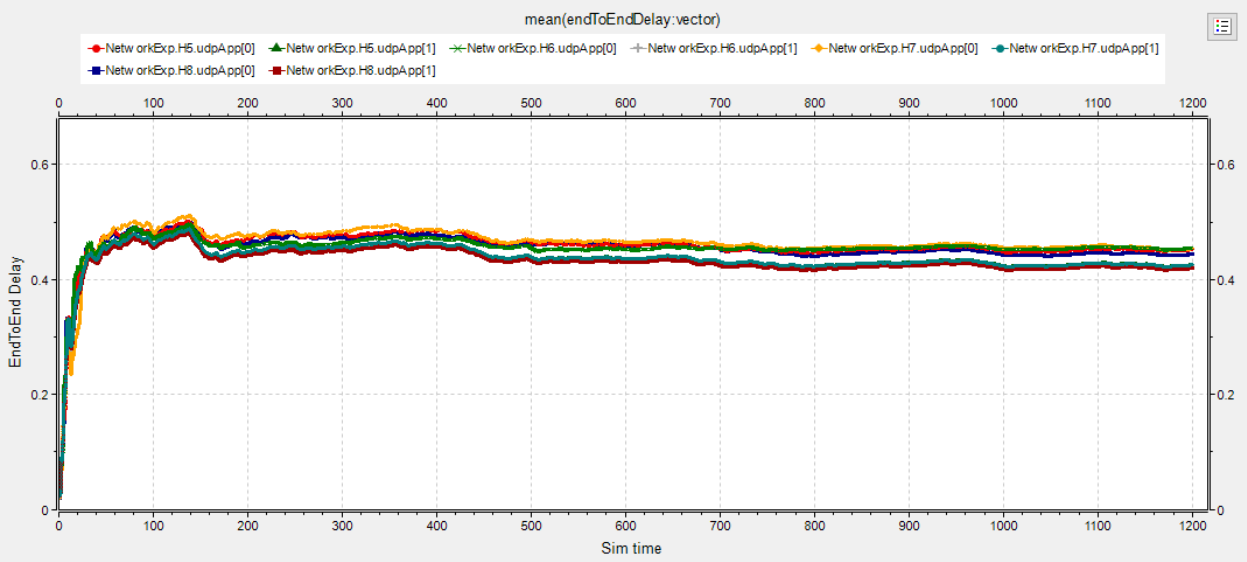

Figure 12: End-to-End delay without QoS applied

As shown the data on Table 7, the packet loss is dispersed to all applications traffic and the end-to-end delay of the packets transmitted through the network almost similar. So, because of any traffic control mechanism does not applied to the network the congestion problem affected all types of applications traffic.

5.3.2. Differentiate Service Network

For the rest of the experiment another application App2 added on server1. The comparison of different type of logical queue management and traffic scheduling technique is done within a DS architecture. The test conducts with many scenarios which includes the comparison of the SP and WRR scheduling, the combining of both scheduling technique with Drop Tail and RED algorithm and the WRED algorithm with different threshold value for each class queue. The DiffServ QoS applied on each node at the edge and core layer. On the edge layer multi field classifier and marker are used. The MF classifier identify the traffic sender host by its source hostname and differentiate the application type by its destination port, and then mark the header of the packet DSCP value according to the applications traffic type category which belongs to the class.

(a) Scenario 1: Strict Priority

The scheduler used for both the edge and core nodes is SP scheduling discipline. It assigns different priority level for each class traffic based on the behavior of the applications. For this test scenario, we use a Drop Tail algorithm as the queue management for all traffic class until the SP scheduler allow to forward packet according to the assigned priority level for each class. Figure 13 and Table 8 shows the logical diagram and priority level of SP scheduler.

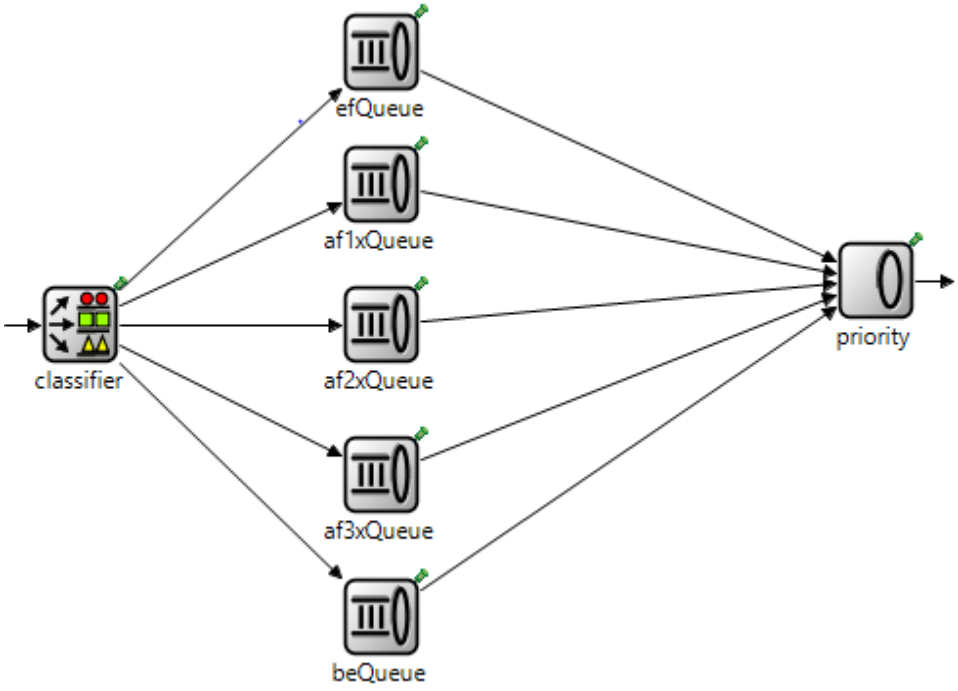

Figure 13: Bandwidth allocation with SP scheduling algorithm

Table 8: Class with priority level.

Class	Priority
EF	0
AF1	1
AF2	2
AF3	3
BE	4

The application App2 running on server H1 map to the EF class, the applications App0 and App1 running on H1, H2 and H3 mapped to AF1, AF2 and AF3 respectively and the applications App0 and App1 running on server H4 are set to the default class with a minimum priority. The packet loss and end to end delay of the test is shown on Table 9, and the average queue time of each class in graph shown on Figure 14.

Table 9: Packet loss and end-to-end delay with SP scheduler.

Applications	Sent PK (Kbyte/s)	Throughput (Kbyte/s)	Packet Loss (Kbyte/s)	Packet Loss (%)	End-to-end delay (sec)
H1.App[0]	9.012	9.012	0.000	0.000	0.024
H1.App[1]	12.191	12.191	0.000	0.000	0.023
H1.APP[2]	16.253	16.253	0.000	0.000	0.023
H2.App[0]	8.792	8.792	0.000	0.000	0.057
H2.App[1]	12.191	12.191	0.000	0.000	0.039
H3.App[0]	9.041	8.194	0.847	9.366	1.294
H3.App[1]	12.192	11.512	0.679	5.570	1.204
H4.App[0]	8.945	0.517	8.428	94.218	28.905
H4.App[1]	12.193	1.139	11.054	90.662	29.457

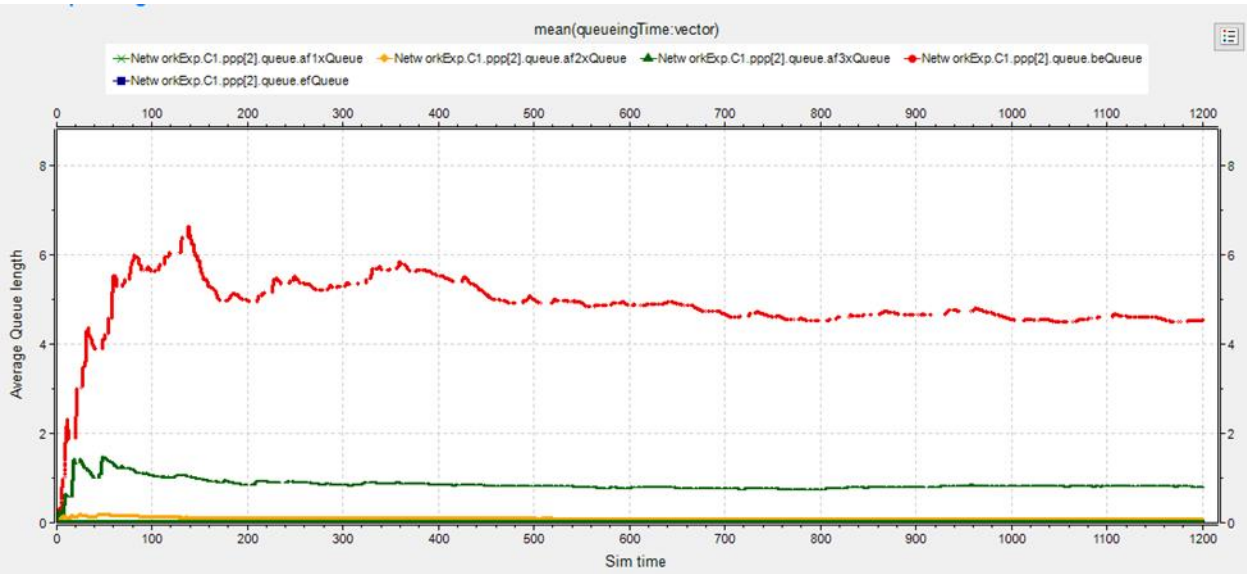

Figure 14: Average queue time graph of SP scheduler.

As shown in the data on the table the default class BE traffic has dropped more than 95%. However, the high priority class applications which are running on server H1 and H2 have not any packet loss with minimum end-to-end delay for these high priority class.

(b) Scenario 2: SP with WRR

In this test scenario performed by combining the SP scheduling algorithm and WRR scheduling algorithm with Drop Tail queue management algorithm. This experiment conducts within two test case, in the first test case the BE service class with lowest priority than the other service class and forward traffic if the other class have no packet to forward, in the second test case the BE service

class shared the bandwidth based on the weight assign to the class. The SP scheduler priority level and the WRR scheduler weigh are show on Table 10.

Table 10: Application class mapping with SP priority level and WRR weight

Class	Application	Priority (case 1)	Priority (case 2)	Weight (Case 1)	Weight (case 2)
EF	H.1.App[2]	0	0	Not apply	Not apply
AF1	H.1.App[0] H.1.App[1]	1	1	10	10
AF2	H.2.App[0] H.2.App[1]	1	1	9	9
AF3	H.3.App[0] H.3.App[1]	1	1	8	8
BE	H.4.App[0] H.4.App[1]	2	1	Not apply	1

The logical design of the two case are shown in Figure 15, in case 1 the applications map to the AF service class shared the bandwidth that is not used by the applications in EF service class according to the weight assigned to each class and the applications in the BE service class only have a chance to transmit packets if any other service class have no any packets to transmit. However, in case two the applications in BE service class have a chance to share the bandwidth with the applications in the other class with a minimal sharing time.

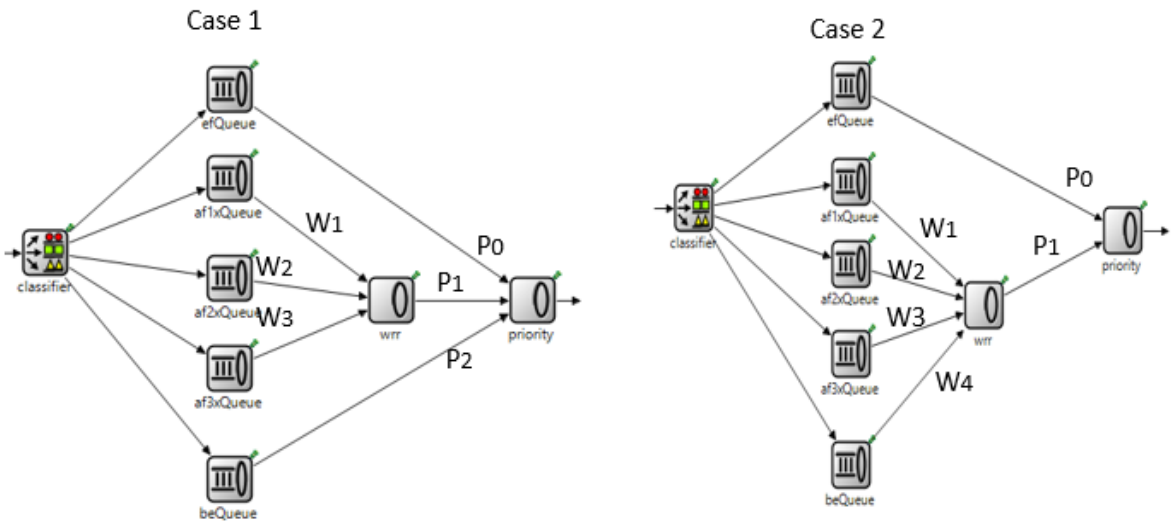

Figure 15: BE class scheduling options

Table 11: BE service class performance result with different options

Applications	Class	Case 1				Case 2			
		Sent Packet (Kbs)	Packet Drop (Kbs)	Packet Drop (%)	End-to-end delay (sec)	Sent Packet (Kbs)	Packet Drop :(Kbs)	Packet Drop (%)	end-to-end delay : sec
H1.App[0]	AF1	9.012	0.009	0.095	0.414	9.012	0.071	0.790	0.649
H1.App[1]	AF1	12.191	0.004	0.037	0.319	12.191	0.043	0.350	0.532
H1.APP[2]	EF	16.253	0.000	0.000	0.023	16.253	0.000	0.000	0.023
H2.App[0]	AF2	8.792	0.094	1.074	0.676	8.792	0.313	3.564	1.039
H2.App[1]	AF2	12.191	0.060	0.494	0.565	12.191	0.186	1.522	0.920
H3.App[0]	AF3	9.041	0.533	5.891	1.183	9.041	1.208	13.358	1.808
H3.App[1]	AF3	12.192	0.281	2.303	1.082	12.192	0.673	5.524	1.721
H4.App[0]	BE	8.945	8.578	95.901	39.466	8.945	8.204	91.712	17.717
H4.App[1]	BE	12.193	11.403	93.519	43.171	12.193	10.230	83.901	18.145
Total		100.809	20.962			100.809	20.927		

As shown in the data on Table 11, the total packet loss of BE class in case 1 is 19.981 Kb/s which mean 94.527% of the total sent traffic by applications in the class and in case 2, the total packet loss of the BE class is 18.433 Kb/s which mean 87.215% of the total sent traffic by applications in the class. So the result shows the packet loss on BE class decrease by 7.312% because we share 2.632% of the time assigned to the AF class, the overall packet loss decrease by 0.035%. However,

the delay on the high priority class increase significantly and we need to adjust the queuing delay by changing the queue algorithm of the high priority class to RED algorithm.

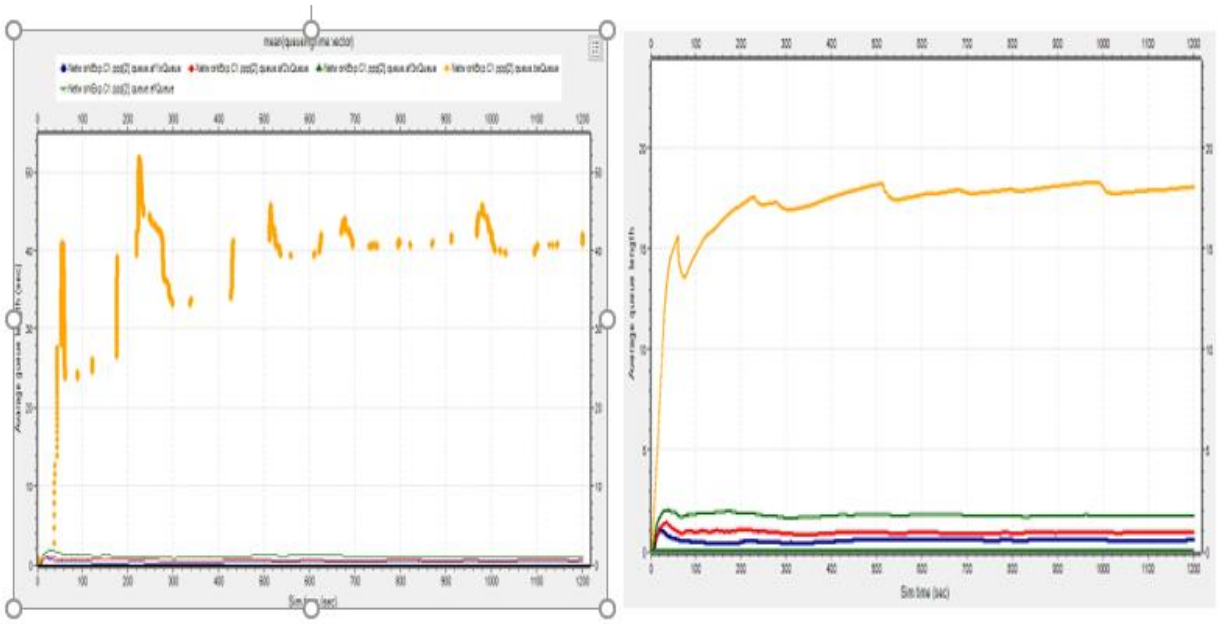

Figure 16: BE queue length with different option

(c) Scenario3: RED Queue Management

In the previous test scenario, case 2 has a better result which minimize packet loss and it prevents starvation of the lower class applications. So, we use this logical model for further test scenario to minimize the packet loss and delay of the aggregate application traffic class. The RED queue management algorithm apply for the AF service class and the other service class using Drop Tail Queue management algorithm. The value of the SP scheduler priority level and the WRR scheduler weight using to this test are show on Table 12.

Table 12: SP and WRR scheduling algorithm with Drop-Tail and RED Queue algorithm

Aggregate class	Drop algorithm	Applications	SP priority level	WRR weight
EF	Drop Tail	H.1.App[2]	0	Not apply
	RED (Low)	H.1.App[0]		
AF1	RED (High)	H.1.App[1]	1	10
	RED (Low)	H.2.App[0]		
AF2	RED (High)	H.2.App[1]	1	9
AF3	RED (Low)	H.3.App[0]		
	RED (High)	H.3.App[1]	1	8
		H.4.App[0]		
BE	Drop Tail	H.4.App[1]	1	1

The output which shows on Table 13, the packet loss percentage of the applications in the same class has difference value. As an example, application App0 has less packet loss percentage than application App1 for each class. The result shows the RED algorithm capability to select among applications with in the same class and helps to select which applications traffic drop first when a bottleneck link occurred.

Table 13: Packet loss and end-to-end delay of scenario 3 test.

Applications	Sent Packet (Kb/s)	Throughput (Kb/s)	Packet Loss (Kb/s)	Packet Loss (%)	end-to-end delay (sec)
H1.App[0]	9.096	9.08	0.017	0.183	0.431
H1.App[1]	12.191	11.923	0.268	2.2	0.341
H1.APP[2]	16.253	16.253	0	0	0.023
H2.App[0]	8.536	8.536	0	0	0.617
H2.App[1]	12.191	11.674	0.517	4.239	0.55
H3.App[0]	8.712	8.643	0.069	0.789	0.942
H3.App[1]	12.192	10.553	1.638	13.437	0.909
H4.App[0]	8.912	0.892	8.02	89.988	14.863
H4.App[1]	12.193	2.262	9.93	81.445	15.374
Total	100.276	79.816	20.459		

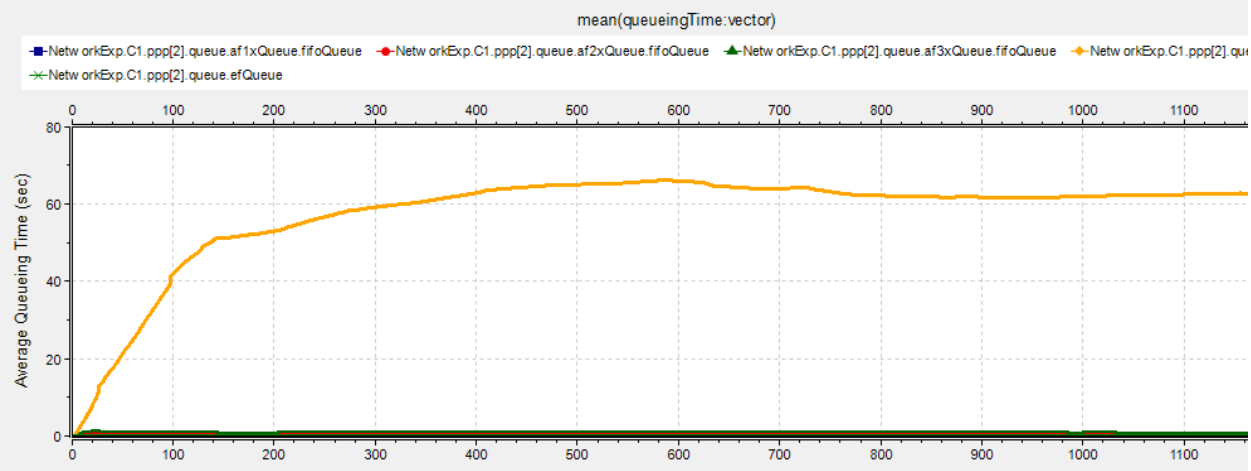

Figure 17: the queueing time of each class with RED algorithm for AF class

(d) Scenario4: WRED

In this test scenario, we evaluate the RED buffer management algorithm with different threshold value per class. The buffer allocated based on the service class characteristics, delay sensitive high priority class has low buffer size and for the class which needs high throughput assigned high buffer space. For this test we set the maximum buffering size of the class in terms of number of packet, such as for EF = 5, AF = 200 and BF = 1000 packets. The minimum and maximum threshold values for each AF service class are sets. The threshold value also varies for each dropping precedent level per class. The maximum drop precedent for each critical and non-critical are sets 0.3 and 0.6 respectively as shown Table 14.

Table 14: Per class threshold value of WRED

AF Class	Min TH1 T1, T2, T3, T4	Min TH2 T1, T2, T3, T4	Max TH1 T1, T2, T3, T4	Max TH2 T1, T2, T3, T4	MaxPr1	MaxPr2
AF1	30, 35, 40, 45	50, 55, 60, 65	60, 65, 70,75	100, 105, 110, 115	0.3	0.6
AF2	50, 55, 50 55	60, 65, 70, 75	100, 105, 110, 115	120, 125, 130, 135	0.3	0.6
AF3	60, 65, 70, 75	80, 85, 90, 95	120, 125, 130, 135	150, 155, 160, 165	0.3	0.6

Table 15: Packet loss with different threshold value per class

Applications	Packet Loss: (Kb/s)				Packet Loss (%)			
	T1	T2	T3	T4	T1	T2	T3	T4
H1.App[0]	0.000	0.001	0.000	0.000	0.000	0.014	0.000	0.000
H1.App[1]	0.205	0.251	0.236	0.179	1.686	2.059	1.933	1.472
H1.APP[2]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
H2.App[0]	0.060	0.033	0.085	0.052	0.684	0.376	0.988	0.581
H2.App[1]	0.563	0.435	0.424	0.439	4.619	3.565	3.481	3.601
H3.App[0]	0.148	0.161	0.133	0.274	1.686	1.832	1.505	2.949
H3.App[1]	1.398	1.265	1.573	1.851	11.465	10.380	12.903	15.186
H4.App[0]	8.218	7.552	7.677	8.203	90.785	90.193	89.570	91.016
H4.App[1]	9.989	9.883	9.831	10.059	81.926	81.058	80.631	82.503
Total Loss	20.582	19.581	19.960	21.058				

The objective of this test scenario was to show the RED algorithm with different threshold per class value which are considered as weighted RED. The data on Table 15 shows the result by adjusting the minimum and maximum threshold value and it minimize packet loss in different forwarding classes and the total packet loss also decrease as compared to the default RED algorithm that have the same value for all forwarding classes. Figure 18 shows the average queueing time of each class, the queue length of the default BE class increased and become constant after 500s simulation time.

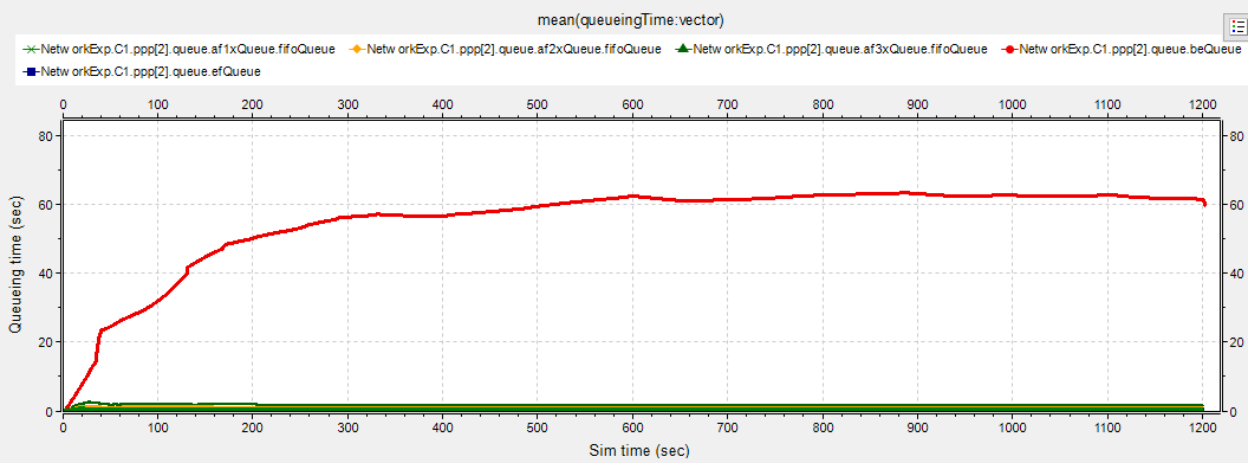

Figure 18: Queueing Time and mean value of different class with WRED algorithm

5.4. Discussion

Most researches shows MF traffic classification is not enough to identify all types of applications traffic, we agree the idea where unknown application traffic exists such as ISP access network. On paper [8] the authors use DPI to identify the application type of the traffic at the ingress point of the network. The DPI has a drawback; it takes high processing time with complexity to inspect the packet payload content for application signature [14, 16], most ISP used at the access network parts to apply QoS or to prevent the core network from intrusion attack. As we see at the data center network view, almost all systems or application are known and their information is documented and the packet can be easily identified by its source or destination IP address, port number or the combination of both IP address and port number on the header. Hence, for our proposed solution MF traffic classification technique is adequate to classify the traffic generated by the applications in the data center without introducing additional burden on the edge device, processing time delay and complexity.

Almost all the papers reviewed in chapter 3 are based on the network resource or bandwidth reservation for a particular application traffic. Their approach reflects the IETF IntServ QoS, and it has a scalability issue and it must be reserved the resource before the application sending traffic. It also has complexity, each node or device along the path need hop by hop signaling to reserve the require resource. When we come in DCN, so many applications exist in the data center, these applications have different types of network requirement and it need differential level of traffic treatment and controlling mechanism based on their behavior. To reserve resources to each of the real-time or business critical application is also very difficult in terms of request handling complexity. Because of this drawback, our solution categorizes these applications based on their behavior such as real-time, interactive, signaling and transactional. Therefore, the IETF DiffServ approach is suitable to map these application behaviors to different level traffic service class which provide nearly guaranteed and controlled bandwidth sharing technique.

On the other hand, most of the research in the related work is conducted on SDN-enable Network. In addition to the technology difference from the Traditional Network, the papers more concerned on the control plane of the network device. Only the researcher on [5] study to improve the minimum and maximum bandwidth reservation algorithm on the data plane of the network. Our DCN technology is a traditional network. Even if most research is done on centrally control the

resource at each node or network device for traditional network by implementing bandwidth broker but the changing of the technology to SDN is discouraged to conduct this research on the control plan. However, the algorithms in data plane (forwarding plane) is also applicable for both technology, and this is the reason we had been interested to tide our research to compare different types of queue management and packet scheduling algorithms of each node or router to improve application bandwidth utilization on bottleneck link existed.

In section 5.3.1, the experiment on a BE service, the test performed without any traffic control mechanism and the same priority level for all applications. The packets forward based on their arrival time which means first come first serve (FCFS). The output of the test shows on Table 7, the packet loss distributed to all application and the average end to end delay of the applications present on Figure 11 are almost similar. After different priority level set to each aggregated applications class in section 5.3.2, the packet loss and delay of the application also vary depend on the class they are existed. In strict priority algorithm in the first scenario of section 5.32, the packets loss in the higher class traffic is zero and the end to end delay of the applications approximately 33ms. However, the packet loss and end to end delay of applications in the lower class extraordinary high. The SP scheduling algorithm has a great quality of service for applications in the higher priority class with the expense of the applications in the lower priority class.

The experiment in scenario 2 of section 5.3.2 is combining the SP and WRR scheduling algorithm. It is minimizing the packet loss on the applications in the lower class by delaying the applications in the higher class with acceptable packet loss. The test performed with two cases by assigning the BE service class with lower priority than the other class and it has not a chance to forward packet from the BE class unless the other class queue is empty and there is no packet to forward at that time. But in case two the BE service class traffic share the bandwidth with the AF service class traffic with a weighted time sharing. In case 2 packet loss in the BE service class and the total packet loss of all traffic decrease. However, the packet loss on the applications in the higher service class significantly increases and also the rate of the packet loss of the application within the same class are randomly depend on the characteristics of Drop Tail queue management algorithm.

In scenario 3, the queue management algorithm of the AF service class altered to the RED algorithm. With assumption of critical and non-critical systems the test performed with two packet dropping precedence level per class. Each precedent configured with its own minimum and

maximum threshold value with maximum packet dropping precedence (Pmax). As the result shows on Table 13, application App (1) set to high Pmax value. So, more packets from application App 1 has been dropped in each AF service class. In addition, the overall packet losses significantly decrease.

In scenario 4, we are using weighted RED queue management algorithm by configured different min and max threshold value per class. The WRED queue management algorithm for AF service class provide flexibility to control the queue time at the core node, and base to dynamic bandwidth allocation by adapt with the average queue time of the class. The result on Table 15 shows the total packet loss decrease within some interval T1 up to T4. However, the packet loss rate per class is not uniform, it going up and down when the threshold value increase. The combination of the Drop Tail and WRED queue algorithm has less packet loss rate than the Drop Tail algorithm when the traffic near at full load. Figure 16 shows the line graph of the comparison queue management algorithm.

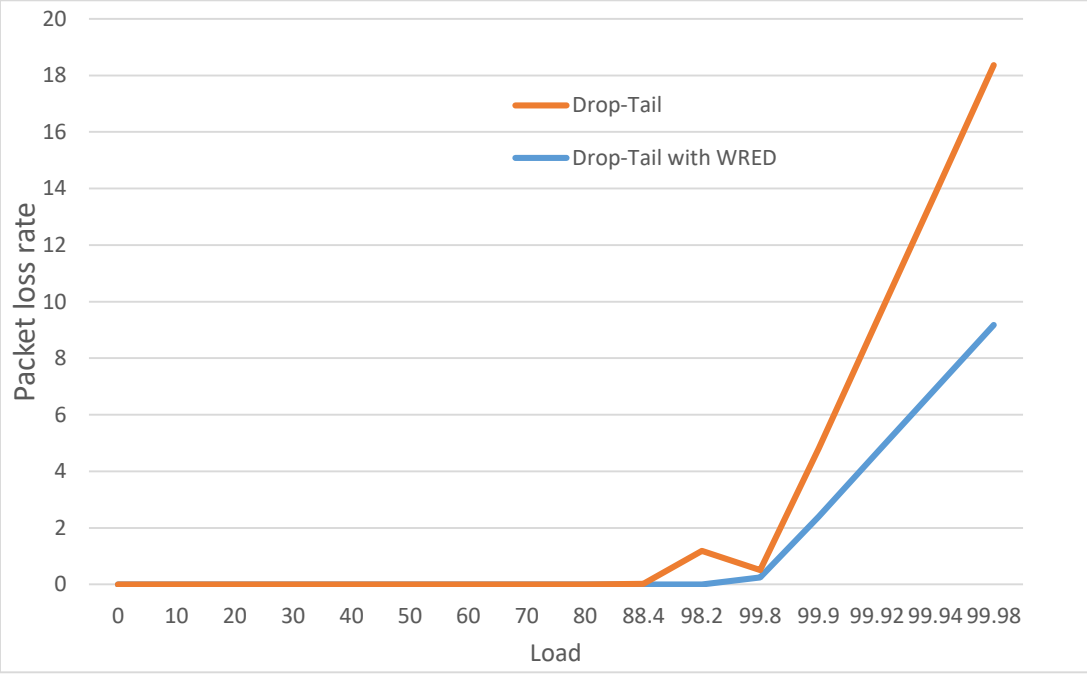

Figure 19: Load vs packet loss comparison

Finally, the experiment shows the combination of the two scheduling algorithm SP and WRR provide near guarantee service for applications in the EF service class and share the forwarding time for the other service class per its assigned weights. The EF service is a low delay and loss

class, so is not as such important to manage the queue length. However, for BE service class main objective is to forward traffic as much as possible with high delay tolerance. So the Drop-tale algorithm provides better performance for high throughput with delay tolerate applications. WRED algorithm for AF service class increase the performance of the network by reducing the total packet loss by 1.34% as shown in Figure 20. In addition, the WRED queue management algorithm has a capability of differentiated an aggregated applications group traffic and drop a packets with the same service class.

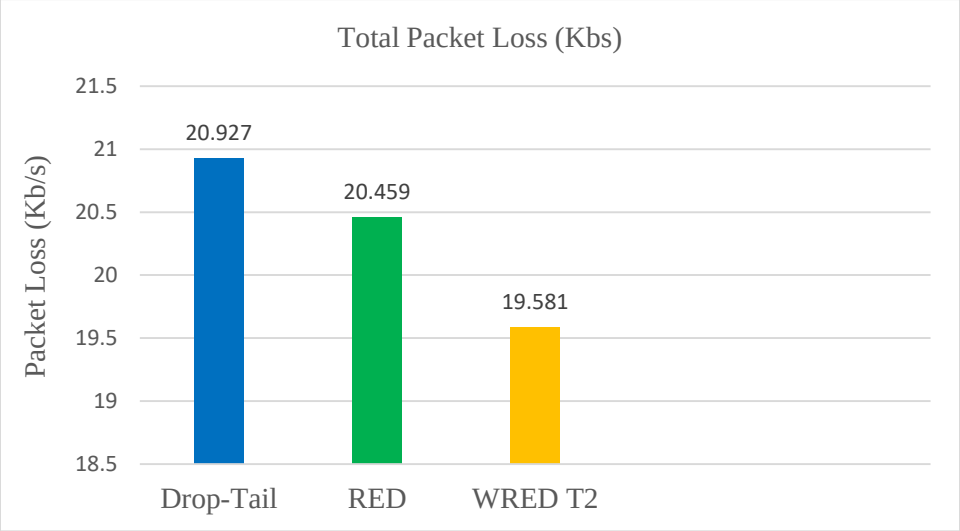

Figure 20: Comparison of different queue management algorithm

6. Conclusion and Future work

6.1. Conclusion

This thesis tries to addressed the performance degradation of critical application or system when a lack of available resource in the DCN. Many approaches have been proposed to solve this problem. One of the long time favorable approaches is the reservation of network bandwidth for a particular application. This approach has its own drawback, which means, the incident caused by burst traffic may end a short period of time, and if the reserved bandwidth not used by the application, it leads to wasting valuable resources. In the other hand, the existence of various types of applications in the DC triggered to the necessity of a differentiate level of traffic handling service.

To challenge this problem and provide a solution, we went through series of steps. First step was to figure out how to the other researches handle similar problem. We categorized the applications in the data center according to their behavior. The process of collecting information about the applications and it traffic characteristic in the data center is very hard problem, so we take the assumption that all applications includes one of the four main categories, such as real-time, signaling, interactive and transactional. Then we selected to follow the DiffServ QoS approach. In general, the DiffServ provide quality of service by differentiate aggregated application and assured forwarding high priority application traffics. The bandwidth utilization on a single node or router depend on the capacity of the link, the scheduling algorithm, the size of the buffer and it queuing management algorithm. We compare three queue management and three scheduling algorithm. In the queue management algorithm weighted RED has better performance for the AF service class and for BE service class Drop Tail algorithm has high throughput.

On the comparison the scheduler the SP algorithm has good performance in terms of low delay without packet loss for high priority class and the WRR class provide some level of fairness and it is configurable. The combination of the two algorithm provide both benefit of the strict priority for real time traffic and fairly share the rest resource to the other traffic class. So the proposed queue management algorithm base on the behavior of each class and the bandwidth allocation weighted as per the criticality of the service class.

After that, we modified the third DB-WS scheduling algorithm to adapt to our proposed design. The design of this algorithm is based on min and max bandwidth allocation depend on the change of the average queue length. We used simulation tools to study the performance of the proposed approach, and perform multiple experiments to compare the performance and effectiveness of our solution with limitation of dynamic weight allocated testing.

Based on the simulation results, we proved that preventing packet loss on critical applications traffic and by configuring per class threshold value we can control the desire and tolerable packet loss and delay. Bandwidth utilization viewpoint, the simulation results showed that the utilization of the critical applications is higher than other non-critical application in the same class.

6.2. Future work

The thesis work may be continued with various aspects. The scope of future work of work is as following

- This thesis work concentrates on improving the delay and throughput of the aggregated application on a bottleneck link. The performance of particular application can also be improved by implementing integrated service at the edge network and will provide the end to end QoS.
- The end to end bandwidth utilization also achieved by implementing QoS routing and load balancing. Application flows which need high throughput without bothering the delay can take a long alternative path.

Reference:

[1] Wu He, and Li Da Xu, “Integration of Distributed Enterprise Applications: A Survey”, IEEE Transactions on Industrial Informatics, Vol. 10, PP. 35 – 42, Feb. 2014.

[2] T. Benson, A. Akella and David A. Maltz, “Network Traffic Characteristics of Data Centers in the Wild”, 2010 ACM.

[3] Mohammad Noormohammadpour and Cauligi S. Raghavendra, “Datacenter Traffic Control: Understanding Techniques and Trade-offs”, 2017 IEEE.

[4] Thomas Zinner, Michael Jarschel, Andreas Blenk, Florian Wamser and Wolfgang Kellerer "Dynamic Application-Aware Resource Management Using Software-Defined Networking: Implementation Prospects and Challenges,” 2014 IEEE Network Operations and Management Symposium (NOMS).

[5] Renhai Xu, Wenxin Li, Keqiu Li and Heng Qi, “Towards Application-aware In-network Bandwidth Management in Data Centers”, 2016 IEEE TrustCom/BigDataSE/ISPA.

[6] Jiyang Liu, Liang Zhu, Weiqiang Sun and Weisheng Hu “Scalable Application-Aware Resource Management in Software Defined Networking,” 2015 17th International Conference on Transparent Optical Networks (ICTON). IEEE, 13 August 2015.

[7] Muzzamil Aziz, H. Amirreza Fazely, Giada Landi, Domenico Gallico, Kostas Christodoulopoulos and Philipp Wieder, “SDN-enabled Application-aware networking for Data center networks”, 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 06 February 2017.

[8] Seyeon Jeong, Doyoung Lee, Jonghwan Hyun, Jian Li and James Won-Ki Hong, “Application-aware Traffic Engineering in Software-Defined Network”, 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 02 November 2017.

[9] Flickenger, Rob (Ed.), " How to accelerate your Internet: a practical guide to bandwidth management and optimisation using open source software", INASP/ICTP, 2006.

[10] Brian Lebiednik, Aman Mangal and Niharika Tiwari, “A Survey and Evaluation of Data Center Network Topologies”, Georgia Institute of Technology, Atlanta, Georgia 2016.

[11] Mohammad Al-Fares, Alexander Loukissas and Amin Vahdat, “A Scalable, Commodity Data Center Network Architecture”, 2008, ACM 978-1-60558-175-0/08/08.

[12] Fan Yao, Jingxin Wu, Guru Venkataramani and Suresh Subramaniam, “A comparative analysis of data center network architectures”, 2014 IEEE International Conference on Communications (ICC). IEEE, 28 August 2014.

[13] Jian Guo1. Fangming Liu, Xiaomeng Huang, John C.S. Lui, Mi Hu, Qiao Gao and Hai Jin, "On Efficient Bandwidth Allocation for Traffic Variability in Datacenters", IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. IEEE, 08 July 2014.

[14] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, Foudil Abdessamia, “Network Traffic Classification Techniques and Comparative Analysis Using Machine Learning Algorithms”, 2016 2nd IEEE International Conference on Computer and Communications. IEEE, 11 May 2017.

[15] Zhenbiao Lin*, Xingyuan Chen and Yongwei Wang, “Application-Level Traffic Identification of Network Security Monitoring”, 2009 First International Workshop on Education Technology and Computer Science. IEEE, 26 May 2009.

[16] Yu Wang, “Automatic Network Traffic Classification”, Deakin University, May 2013.

[17] Li-Wei Cheng and Shie-Yuan Wang, “Application-Aware SDN Routing for Big Data Networking", 2015 IEEE Global Communications Conference (GLOBECOM). IEEE. February 2016.

[18] Shuai Zhao and Deep Medhi, “Application-Aware Network Design for Hadoop MapReduce Optimization Using Software- defined Networking”, IEEE Transactions on Network and Service Management. Vol. 14, PP. 804 – 816, 18 July 2017.

[19] A. Kebede, T. Abebe, W. Melesse, Z. Mamo, “Assessment of Enterprise Systems integration problems in ethio telecom”, Seminar paper 2017.

[20] Y. Qian, Z. Lu and Q. Dou, “QoS Scheduling for NoCs: Strict Priority Queueing versus Weighted Round Robin”, 2010 IEEE International Conference on Computer Design. IEEE, 29 November 2010.

[21] Ion Stoica., Hui Zhang, “Providing Guaranteed Services Without Per Flow Management”, Carnegie Mellon University Pittsburgh, 1999 ACM.

[22] Tim Szigeti, Christina Hatting, “End-to-End QoS Network Design”, cisco press, Nov 2004.

[23] S. Radhaknshnan, S.V. Radghsivan, a. Agrawaia, Design & Perfiormance Study of a Flexible Trafic Shaper for High Speed Nenvorks, Feb. 1997

[24] Edward W. Knightly and Jingyu Qiu, Meusurement-Based Admission control with Aggregare Trafic Envelopes, ECE Department, Rice University.

[25] “Quality of service regulation manual”, ITU 2017

[26] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine “A Framework for Integrated Services Operation over Diffserv Networks”, RFC 2998, Nov 2000.

[27] R. Braden, D. Clark, S. Shenker, “Integrated Services in the Internet Architecture: An Overview”, IETF RFC 1633, June 1994

[28] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, IETF RFC2475, December 1998.

[29] J. Harju and P. Kivimaki, “Co-operation and Comparison of DiffServ and IntServ: Performance Measurements”, Proceedings 25th Annual IEEE Conference on Local Computer Networks, LCN 2000.

[30] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, IETF RFC 2597, June 1999.

[31] https://www.omnetpp.org

[32] G. Almes, S. Kalidindi, M. Zekauskas and A. Morton, Ed., “A One-Way Delay Metric for IP Performance Metrics (IPPM)”, IETF RFC 7679, January 216.

[33] G. Almes, S. Kalidindi, M. Zekauskas and A. Morton, Ed., “A One-Way Loss Metric for IP Performance Metrics (IPPM)”, IETF RFC 7680, January 216.

[34] O. Yelbas and E. Germen “A New Approach to Estimate RED Parameters Using Global Congestion Notification”, 2011 International Conference on Network Computing and Information Security, 2011 IEEE.

[35] Omar Almomani, Osman Ghazali, Suhaidi Hassan, Shahrudin Awang Nor and Mohammad Madi, “Impact of Large Block FEC with Different Queue Sizes of Drop Tail and RED Queue Policy on Video Streaming Quality over Internet”, 2010 IEEE.

[36] Ashish Kumar, Ajay K Sharma, Arun Singh Dr. B R Ambedkar “Comparison and Analysis of Drop Tail and RED Queuing Methodology in PIM-DM Multicasting Network”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3816 – 3820.

[37] S. Patel, P. Gupta and G. Singh “Performance Measure of Drop Tail and RED Algorithm”, 2010 IEEE.

[38] Zhang Heying, Liu Baohong and Dou Wenhua, “Design of a Robust Active Queue Management Algorithm Based on Feedback Compensation”, *SIGCOMM’03,* August 2003 ACM.

[39] https://www.iana.org, “Service Name and Transport Protocol Port Number Registry”, Last Updated 2018-10-04.

[40] http://docwiki.cisco.com, “Quality of Service Networking with End-to-End QoS Levels”.

[41] Namratha B Gowda, Reema Sharma, Navin Kumar, Talabatulla Srinivas, “Dynamic Benefit-Weighted Scheduling Scheme in Multi Service Networks”, 2015 IEEE International Advance Computing Conference (IACC).

[42] B. Davie, A. Charny, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, “An Expedited Forwarding PHB”, IETF RFC 3246, March 2002.

Declaration

I certify that the work contained in the thesis is original and has been done by myself under the general supervision of my advisor. The work has not been submitted to in this or any other Institute for any degree or diploma. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Zerihun Mamo _____

Name Signature