Network Congestion and Quality of Service Analysis Using OPNET

A Thesis
Submitted to the College of Information Engineering
of Nahrain University in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
In
Information Engineering

By
Furat Nidhal Tawfeeq
(B.Sc. 2005)

RabeaAl-Awal **1430**
March **2009**

بِسم الله الرَّحمَنِ الرَّحيمِ

وَعلَّمَكَ مَالَم تَكُن تَعلَمُ
وَكَانَ فَضلُ اللهِ عَلَيكَ عَظِيماً

صَدقَ اللهُ العَظيم

النساء 113

شكر وتقدير

سجدة شكر لله العظيم الذي مكنني من ان انجز هذا البحث المتواضع, والحمد لله على ما تقدم وما تأخر . . والصلاة والسلام على سيد الخلق نبينا محمد (صلى الله عليه وسلم).

بوفاء وامتنان عميقين يسعدني ان ارفع الى استاذي الفاضل الاستاذ الدكتور (محمد احمد عبدالله) المشرف على رسالتي جزيل شكري وخالص تقديري لما ابداه من جهد علمي كبير تجسد في توجيه البحث وتحقيقه عسى ان تفي كلماتي هذه بما له من فضل كبير علي.

والشكر المملوء بكل عطر زكي والذي يفوح بالحب والحنان الى والدي, والدتي وخطيبتي لما ابدوا من مساعده طيله فتره دراستي.

ومسك الختام هذا الشكر اتقدم به الى كل من اعانني ولو باليسير في انجاز هذا البحث.

فرات نضال توفيق

2009

ABSTRACT

Congestion management features operate to control congestion once it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, and then determine some methods of prioritizing it onto an output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on the network performance. The work presented in this thesis deals with important issue that is the quality of service (QoS) techniques, which can be integrated to enhance the operation of the network. The quality of service is the collective effect on service performance, which determines the degree of satisfaction of a user of the service.

In this thesis, packet schedulers (FIFO, WFQ, CQ and PQ) and traffic policing algorithms (CAR) were implemented and evaluated under different applications (file transfer, E-mail, voice and video conferencing) with different priorities (Background, Standard, Excellent Effort and Streaming multimedia).

Computer simulation was performed to study and verify the above mechanisms in the performance enhancement using the OPNET simulator; the output result which depends on some statistics that describe the behavior of the network such as end-to-end delay, delay variation, packet loss, etc were collected.

According to the results, we found that relatively the weighted fair queuing is the appropriate type of queuing by offering dynamic, fair queuing that divides bandwidth across queues of traffic based on weights, also this results will be more accepted when commit access rate is implement together with WFQ, so the later will be use the ability of the CAR to manage rate

traffic by using access bandwidth management which implements by using rate limit statements.

Table of Contents

List of Abbreviations

ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
CAR	Committed Access Rate
CIR	Committed Information Rate
CPU	Central Processing Unit
CQ	Custom Queuing
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DEC	Digital Equipment Corporation
Diffserv	Differentiated services
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FIFO	First In First Out
FQ	Fair Queuing
IEEE	Institute of Electrical and Electronics Engineers
IntServ	Integrated Services
ITU	International Telecommunication Unit
MAN	Metropolitan Area Network
OPNET	Optimized Network Engineering Tool
PAN	Personal Area Network
PPP	Point-to-Point Protocol
PQ	Priority Queuing
QoS	Quality of Service
ToS	Type of Service
TTL	Time To Live

VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
WAN	Wide Area Network
WFQ	Weighted Fair Queuing

List of Symbols

C_v	Coefficient of variation
B_C	Normal burst size
B_E	Extended burst size
d	Delay for one packet
D	Average delay
D_A	The number of tokens the stream currently borrowed
D_C	The sum of the D_A of all packets sent since the last time a packet was dropped
d_i	Summation of all delays
N	Total number of all packets
n	Counter
NL	The number of packets lost during the same time period
J	Jitter
ΔT	Time sample
τ	Interval time between two packets

List of Tables

List of Figures

CHAPTER ONE

INTRODUCTION

1.1 Background

When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors). However, as traffic increases too far, the routers are no longer able to cope and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely and almost no packets are delivered. Figure (1.1) shows the relationship between packet sent and packet received for perfect, desirable and congestion state, so from this figure once can consider the congestion state, which occurs when packet sent increase but the packet delivered decrease. Congestion can be brought on by several factors. If all of a sudden, streams of packets begin arriving on three or four lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packet will be lost. Adding more memory may help up to a point, but if routers have an infinite amount of memory, congestion become worse, not better because by the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent. All these packets will be dutifully forwarded to the next router, increasing the load for all the ways to the destination.

Slow processors can also cause congestion, if the routers' CPU (Central Processing Unit) is slow at performing the required tasks of them (queuing buffers, updating tables, etc), queues can build up, even though there are excess line capacity. Similarly, low-bandwidth lines can also cause congestion. Upgrading the lines but not changing the processors, or vice versa, often helps a little but frequently just shifts the bottleneck. Also, upgrading part but not all of the system often just moves the bottleneck somewhere else. The real problem is frequently a mismatch between parts of the system; this will persist until all the components are in balance [1].

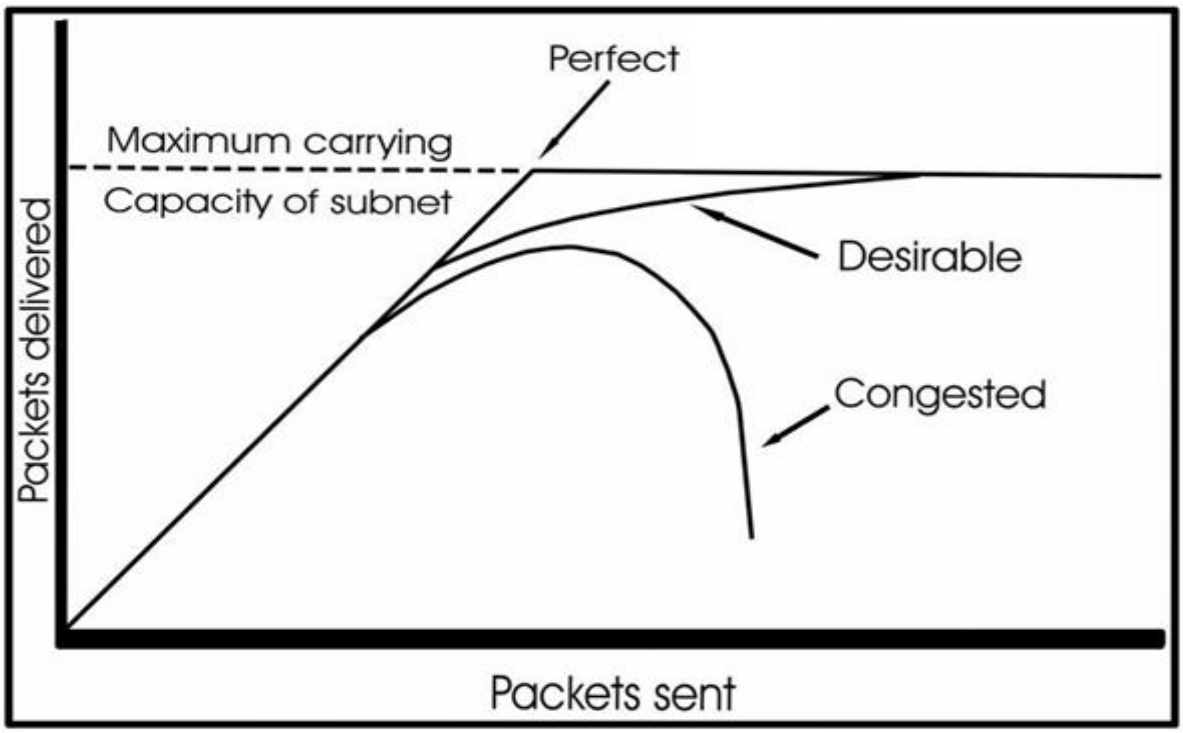

Figure (1.1): Relationship between packets sent and packet delivered [1]

The goal of congestion control mechanisms is simply to use the network as efficiently as possible, that is, attain the highest possible throughput while maintaining a low loss ratio and small delay. Congestion must be avoided because it leads to queue growth and this leads to delay and loss. Implementing congestion control is expensive, and its direct or indirect benefit to whoever implements it is unclear. Thus, reasonable first step towards alleviating these issues are by promoting a Quality of Service (QoS) oriented view of congestion control [2]. Communication networks face a variety of quality-of-service demands. The main motivation for a QoS unit in a data network port processor is to control access to available bandwidth and to regulate traffic. Traffic regulation is always necessary in WAN (Wide Area Network) in order to avoid congestion. Voice and video transmissions over IP must be able to request a higher degree of assurance from the network. A

network that can provide these various levels of services requires a more complex structure. So QoS goals are:

1. Keep queues short.
2. Utilize as much as possible of the available capacity.

The benefits of the first goal are low loss and low delay. The benefit of the second goal is high throughput; ideally, a congestion control mechanism would yield the highest possible throughput that does not lead to increased delay or loss [3].

Queuing schemes provide predictable network service by providing dedicated bandwidth, controlled jitter and latency, and improved packet loss characteristics. The basic idea is to pre-allocate resources (processor and buffer space) for sensitive data. Each of the following schemes require customized configuration of output interface queues [4]

1. Priority Queuing (PQ) assures that during congestion the highest priority data does not get delayed by lower priority traffic. However, lower priority traffic can experience significant delays. PQ is designed for environments that focus on mission critical data, excluding or delaying less critical traffic during periods of congestion.

2. Custom Queuing (CQ) assigns a certain percentage of the bandwidth to each queue to assure predictable throughput for other queues. It is designed for environments that need to guarantee a minimal level of service to all traffic.

3. Weighted Fair Queuing (WFQ) allocates a percentage of the output bandwidth equal to the design weight of each traffic class during periods of congestion.

1.2 OPNET Simulator

OPNET (Optimized Network Engineering Tool) is a comprehensive software environment for modeling, simulating, and analyzing the performance of communications networks, computer systems and applications, and distributed systems. It provides the user with a well built and detailed graphical user interface giving him the advantage to create difficult network topologies with different node characteristics and apply various simulations on it [5].

OPNET Modeler is based on a series of hierarchical editors that directly parallel the structure of real networks, equipment, and protocols which are [6]:

- The Project Editor: graphically represents the topology of a communications network. Networks consist of node and link objects, configurable via dialog boxes. Drag and drop nodes and links from the editor's object palettes to build the network, or use import and rapid object deployment features. Use objects from OPNET's extensive Model Library, or customize palettes to contain own node and link models. The Project Editor provides geographical context, with physical characteristics reflected appropriately in simulation of both wired and mobile/wireless networks. Use the protocol menu to quickly configure protocols and activate protocol specific views.
- The Node Editor: captures the architecture of a network device or system by depicting the flow of data between functional elements, called "modules". Each module can generate, send, and receive packets from other modules to perform its function within the node. Modules typically represent applications, protocol layers, algorithms and physical resources, such as buffers, ports, and buses.

- The Process Editor: used to create process models which control the underlying functionality of the node models created in the Node Editor one can use the Process Editor.

Network simulation provides a means of testing proposed changes prior to placing them into effect, performing what-if analysis concerning the reliability of key components in a network and the effects of losing a component, planning for future growth, and initial design of a proposed network. The costs associated with the building and operating of a network make simulation a viable option in making choices in the building, modification, and performance analysis of a network [7].

OPNET has been widely used to test new protocols and applications in a networked environment since its inception in 1987. It is also used by network equipment manufacturers to evaluate the performance of newly developed products prior to manufacturing. OPNET is structured into a number of hierarchical modeling layers (similar to communications protocols). A detail of a modeling editor is hidden from its higher editors. This allows users to concentrate on a specific modeling problem, and frees them from the unnecessary details of lower layers. OPNET can be run in either UNIX or Windows NT/Windows 2000. OPNET's built-in model libraries contain most popular network protocols and network devices, with new protocols and products added to the libraries each year. OPNET also models a wide range of network equipment (e.g. routers, switches, links, etc.) manufactured by leading network equipment manufacturers in the world. OPNET offers network modelers with great freedom and convenience in simulating networks they want to study. For example, a network can be built by selecting the desired node models from the Network Palette and connecting them with appropriate link models, just like the way the networks were built in real life [8].

1.3 Literature Survey

The following are the result of the survey made during the work in this thesis which is related to the thesis subject:

Lommerdal and Soder, in 2003 [9] described an approach for modeling different types of congestion management methods. A basic price calculation method and a new congestion management identification method are described. Case studies have been simulated, comparing counter trading and implicit auctioning. The focus is on what economic impact of these methods on the behavior of various actors in a re-regulated power market.

Cholvi and Laderas, in 2005 [10] presented a system model that, depending on the current network congestion, makes nodes to establish link connections, so the resulting topologies tend to a star when congestion is small and to random topologies when congestion becomes relevant, so a model can be easily implemented in practice and therefore, may be relevant in areas as the topology management of peer-to-peer networks.

Singh and Gupte, in 2005 [11] studied network traffic dynamics in a two-dimensional communication network with regular nodes and hubs by using Manhattan distance. Also discuss strategies to manipulate hub capacity and hub connections to relieve congestion and define a coefficient of betweenness centrality (CBC), a direct measure of network traffic, which is useful for identifying hubs that are most likely to cause congestion.

Binbin, in 2006 [12] studied the Bulk Data Transfer Scheduling (BDTS) problem, by finding the optimal bandwidth allocation profile for each task to minimize the overall network congestion. He show that the multi-interval scheduling, which divides the active window of a task into multiple intervals and assigns bandwidth value independently in each of them, is both

sufficient and necessary to attain the optimality in BDTS. Specifically, he shows that BDTS can be solved in polynomial time as a Maximum Concurrent Flow Problem. The optimal solution attained is in the form of multi-interval scheduling with the number of intervals upper-bounded.

Kleinebecker and Skelley, in 2006 [13] shows that the current model of Quality of Service (QoS) is Diffserv (Differentiated services). In this model packets are marked according to the type of service needed. The carriers distinguish multiple traffic classes to users and charge QoS fees for premium traffic classes. Users will pick up the appropriate traffic class needed for their application. The service level agreements (SLA) cover bandwidth, round-trip delay, packet loss rate and possibly jitter. The agreements are loose but sufficient for most applications.

Singh, Prabhan and Francis, in 2007 [14] presented MPAT (Multi-Probe Aggregate TCP); the first truly scalable algorithm for fairly providing differential services to TCP flows that share a bottleneck link. Unlike known schemes, our approach preserves the cumulative fair share of the aggregated flows even where the number of flows in the aggregate is large.

Chaintreau, Baccelli and Diot, in 2008 [15] studied the impact of random queuing delays stemming from traffic variability, on the performance of a multicast session. With a simple analytical model, they analyze the throughput degradation within a multicast tree under TCP-like congestion and flow control. They use methods based on stochastic comparison and on the theory of extremes (Lai and Robbins' notion of maximal characteristics) to prove various properties of the throughput.

Tang and Wan, in 2008 [16] introduced the mode of PJM FTR auction market used in congestion management and proposes an integrated process of congestion management including both contract and spot markets. As an analyzing result, this paper points out that FTR (Financial Transmission

Right) used in congestion management is more appropriate to the deregulated environment, which can hedge congestion charges, show the accurate signals, and so on.

1.4 Aim of the Research

The purpose of this thesis is to design a MAN that connect four of the major governorates in Iraq (Baghdad, Mosul, Basrah, Anbar) and try to solve the congestion problem occurs among these governorates by using the methods of congestion management and types of queue. The results among these different types will be compared to find the appropriate way.

1.5 Thesis Layout

This thesis is organized in five chapters which can be summarized as follows:

Chapter 2 Describes Networks technologies, internet protocol and method of information transmission, also describe the term of quality of service and their parameters

Chapter 3 Describes some methods which try to manage the congestion like types of queuing and CAR technique

Chapter 4 Simulates a proposal design of wide area network and view the effect of the congestion and try to manage this problem by using several techniques.

Chapter 5 Includes the conclusions drawn from this work followed by suggestions for future work.

CHAPTER TWO

NETWORK PROTOCOLS & QUALITY OF SERVICE

2.1 Introduction

This chapter gives a brief description about networks technology; also describe the Internet Protocol and the terms quality of service and their parameters and who much these parameters affect on the applications.

2.2 Network Technologies

Data signals are transmitted from one device to another using one or more types of transmission media, including twisted-pair cable, coaxial cable and fiber-optic cable. A message to be transmitted is the basic unit of network communications. A message may consist of one or more cells, frames or packets which are the elemental units for network communications [17]. Networks can be classified through their scale (size), as shown in Table (2.1) [1].

Table (2.1): Classification of interconnected processors by scale

Processor distance	Processors located in same example	Network Technology
1 m	Square meter	Personal Area Network
10 m	Room	Local Area Network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan Area Network
100 km	Country	Wide Area Network
1000 km	Continent	

2.2.1 Personal Area Network (PAN)

Personal area networks are the networks that are meant for one person, for example a wireless network connecting a computer with its mouse, keyboard, and printer is a personal network [1].

2.2.2 Local Area Network (LAN)

A local area network (LAN) is a communication system that allows a number of independent devices to communicate directly with each other in a limited geographic area such as a single office building, a warehouse or a campus. LANs are standardized by three architectural structures: Ethernet, token ring and fiber distributed data interface (FDDI) as discussed below [18]:

a) Ethernet: is a LAN standard originally developed by Xerox and later extended by a joint venture between Digital Equipment Corporation (DEC), Intel Corporation and Xerox. The access mechanism used in an Ethernet is called Carrier Sense Multiple Access with Collision Detection (CSMA/CD). In CSMA/CD, before a station transmits data, it must check the medium where any other station is currently using the medium. If no other station is transmitting, the station can send its data. If two or more stations send data at the same time, it may result in a collision. Therefore, all stations should continuously check the medium to detect any collision. If a collision occurs, all stations ignore the data receive.
b) Token Ring: a LAN standard originally developed by IBM^{TM} uses a logical ring topology. The access method used by CSMA/CD may result in collisions. Therefore, stations may attempt to send data many

times before a transmission captures a perfect link. This redundancy can create delays of indeterminable length if traffic is heavy. There is no way to predict either the occurrence of collisions or the delays produced by multiple stations attempting to capture the link at the same time. Token ring resolves this uncertainty by making stations take turns in sending data. As an access method, the token is passed from station to station in sequence until it encounters a station with data to send. The station to be sent data waits for the token. The station then captures the token and sends its data frame. This data frame proceeds around the ring and each station regenerates the frame. Each intermediate station examines the destination address, finds that the frame is addressed to another station, and relays it to its neighboring station. The intended recipient recognizes its own address, copies the message, checks for errors and changes four bits in the last byte of the frame to indicate that the address has been recognized and the frame copied. The full packet then continues around the ring until it returns to the station that sent it.

c) Fiber Distributed Data Interface (FDDI): is a LAN protocol standardized by ANSI (American National Standards Institute) and ITU (International Telecommunication Unit). It supports data rates of 100 Mbps and provides a high-speed alternative to Ethernet and token ring. The access method in FDDI is also called token passing. In a token ring network, a station can send only one frame each time it captures the token. In FDDI, the token passing mechanism is slightly different in that access is limited by time. Each station keeps a timer which shows when the token should leave the station. If a station receives the token earlier than the designated time, it can keep the token and send data until the scheduled leaving time.

2.2.3 Metropolitan Area Network (MAN)

A metropolitan Area Network covers a city. The best-known example of a MAN is the cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over-the-air television reception. In these early systems, a large antenna was placed on top of a nearby hill and signal was then piped to the subscribers' houses [1].

2.2.4 Wide Area Network (WAN)

A WAN provides long-distance transmission of data, voice, image and video information over large geographical areas that may comprise a country, a continent or even the world. In contrast to LANs (which depend on their own hardware for transmission), WANs can utilize public, leased or private communication devices, usually in combination [18].

a) PPP: The Point-to-Point Protocol (PPP) is designed to handle the transfer of data using either asynchronous modem links or high-speed synchronous leased lines.
b) X.25: is widely used, as the packet switching protocol provided for use in a WAN. It was developed by the ITU-T in 1976. X.25 is an interface between data terminal equipment and data circuit terminating equipment for terminal operations at packet mode on a public data network. X.25 defines how a packet mode terminal can be connected to a packet network for the exchange of data. It describes the procedures necessary for establishing connection, data exchange, acknowledgement, flow control and data control.
c) Frame Relay: is a WAN protocol designed in response to X.25

deficiencies. X.25 provides extensive error-checking and flow control. Packets are checked for accuracy at each station to which they are routed. Each station keeps a copy of the original frame until it receives confirmation from the next station that the frame has arrived intact.

d) Asynchronous Transfer Mode (ATM): is a revolutionary idea for restructuring the infrastructure of data communication. It is designed to support the transmission of data, voice and video through a high data-rate transmission medium such as fiber-optic cable. ATM is a protocol for transferring cells, where cell is a small data unit of 53 bytes long, made of a 5-byte header and a 48-byte payload. The header contains a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI). These two identifiers are used to route the cell through the network to the final destination. An ATM network is a connection-oriented cell switching network. This means that the unit of data is not a packet as in a packet switching network, or a frame as in a frame relay, but a cell. However, ATM, like X.25 and frame relay, is a connection-oriented network, which means that before two systems can communicate, they must make a connection.

WAN contains a collection of machines intended for running user (i.e., application) programs. These machines are called hosts. These hosts are connected by a communication subnet, or just subnet for short. The hosts are owned by the customers (e.g., people's personal computers), whereas the communication subnet is typically owned and operated by a telephone company or Internet service

provider. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener. In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements. Transmission lines move bits between machines, they can be made of copper wire, optical fiber, or even radio links. Switching elements are specialized computers that connect three or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line on which to forward them. These switching computers have been called by various names; the most name widely used is router. Figure (2.1) shows an example of network; each host is frequently connected to a LAN on which a router is present, although in some cases a host can be connected directly to a router.

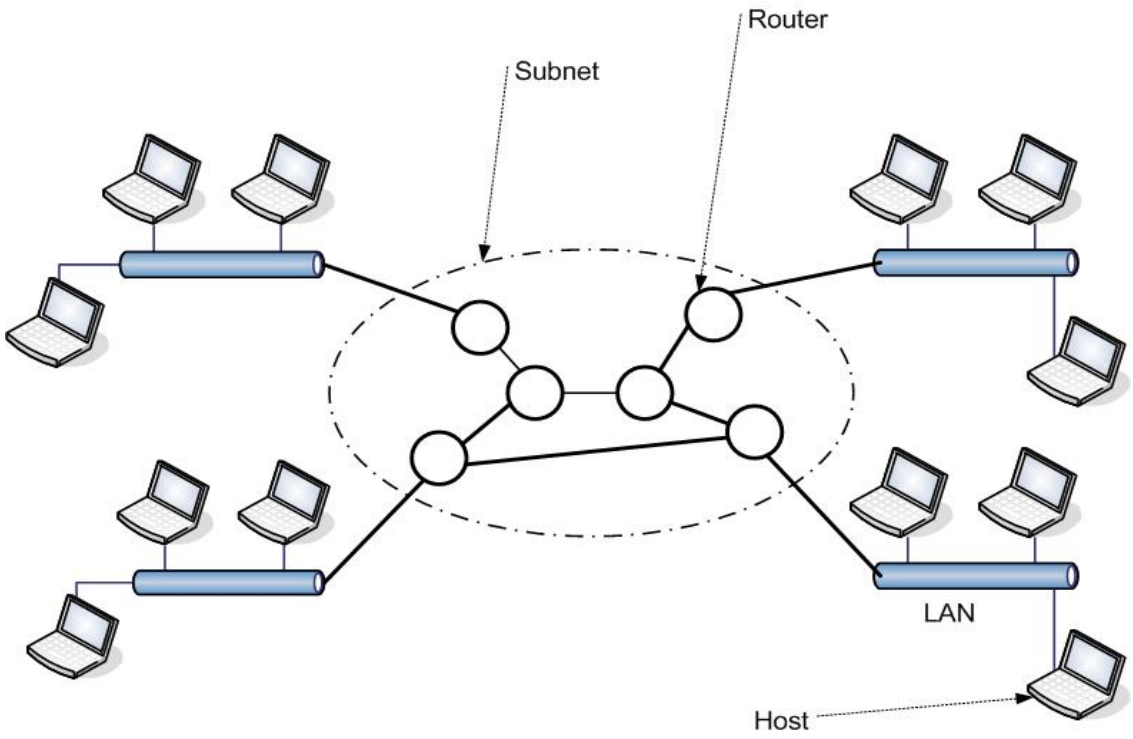

Figure (2.1): Relation between hosts on LANs and the subnet

In most WANs, the network contains numerous transmission lines, each one connecting a pair of routers. If two routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. A subnet organized according to this principle is called a store-and-forward or packet-switched subnet. Nearly all wide area networks (except those using satellites) have store-and-forward subnets [1].

2.3 IP Datagram Format

The IP datagram header is a minimum of 20 bytes long as shown in Figure (2.2), where [19]:

Version**:** Keep track on which version of the protocol the datagram belongs to, which may be 4, 5 or 6 (the current is 4).

Header Length**:** Internet Header Length is the length of the Internet header in 32 bit words, and thus points to the beginning of the data.

ToS with IP Precedence Bits**:** Type of Service tells how the datagram should be handled. The first 3 bits are the priority bits. Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above certain precedence at time of high load). The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput. More details about ToS will be described in chapter three.

Total Length**:** Is the total length of the datagram, header and data.

Identification**:** A unique number assigned by the sender to aid in reassembling a fragmented datagram. Fragments of a datagram will have the same identification number.

Flags**:** Specifies whether fragmentation should occur.

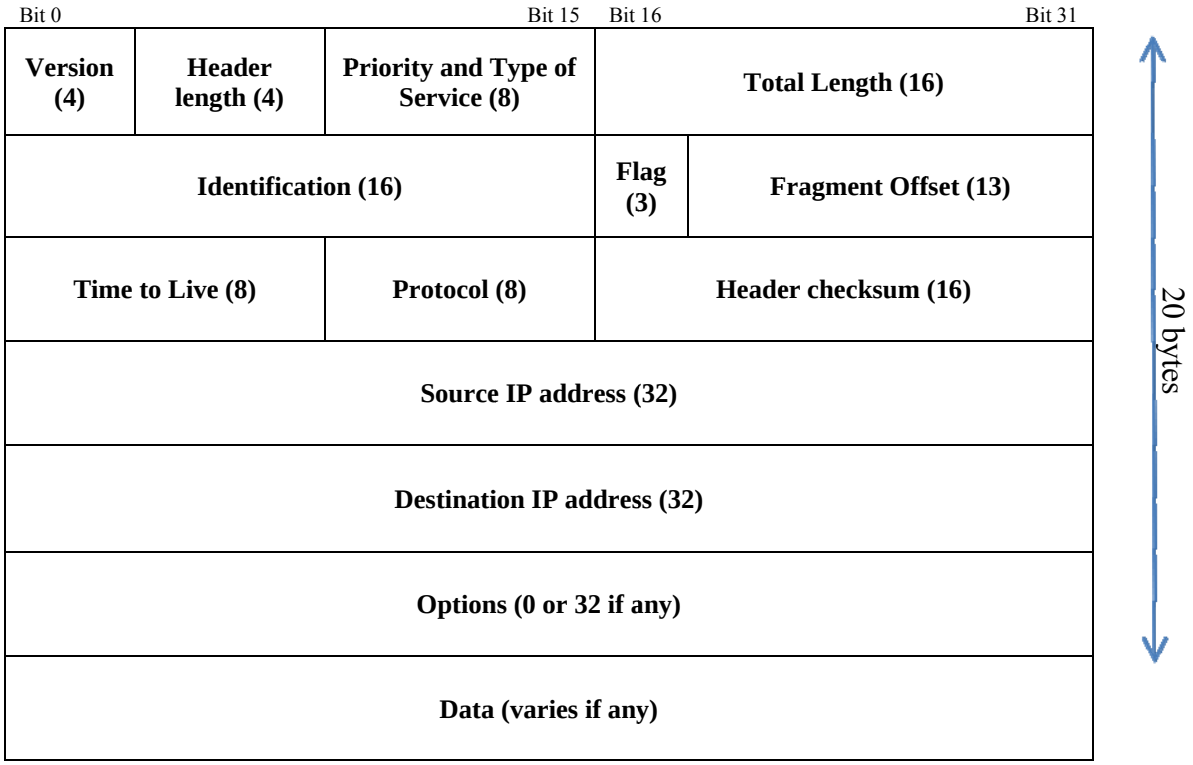

Figure (2.2): IP Datagram format [30]

Fragment Offset: This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

Time to Live (TTL): This field indicates the maximum time the datagram is allowed to remain in the Internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in Internet header processing. The time is measured in units of seconds, but since every module

that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Protocol**:** Indicates the higher level protocol to which IP should deliver the data in this datagram.

Header Checksum: A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the Internet header is processed.

Source IP Address**:** The 32-bit IP address of the host sending this datagram.

Destination IP Address**:** The 32-bit IP address of the destination host for this datagram.

Options (Variable length): The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

Padding (variable size): The Internet header padding is used to ensure that the Internet header ends on a 32 bit boundary. The padding is zero.

Data**:** The data contained in the datagram is passed to a higher level protocol, as specified in the protocol field.

2.4 Methods of Information Transmission [20]

There are many different network protocols and several protocols can be available even on a single layer. Especially with lower-layer protocols, this part will distinguish between the types of transmission that they facilitate, whether they provide connection-oriented or connection-less services, if the protocol uses virtual circuits, and so on, also distinguish between synchronous, packet, and asynchronous transmission.

2.4.1 Synchronous Transmission

Synchronous transmission is needed when it is necessary to provide a stable (guaranteed) bandwidth, for example, in audio and video. If the source does not use the provided bandwidth it remains unused. Synchronous transmission uses frames that are of fixed length and are transmitted at constant speeds. In synchronous transmission; the guaranteed bandwidth is established by dividing the transmitted frames into slots as shown in Figure (2.3)

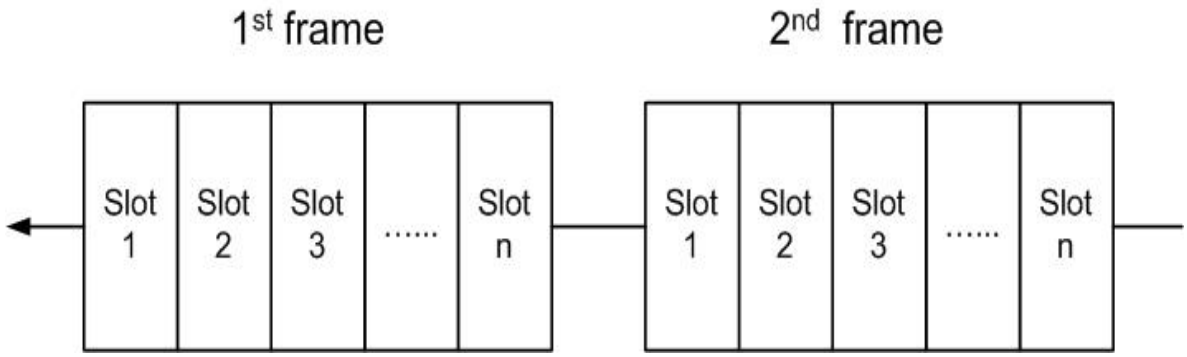

Figure (2.3): Frames divided into slots in synchronous transmission

2.4.2 Packet Transmission

Packet transmission is especially valuable for transferring data. Packets usually carry data of variable size. One packet always carries data of one particular application (of one connection). It is not possible to guarantee bandwidth, because the packets are of various lengths. On the other hand, bandwidth can be used more effectively because if one application does not transmit data, then other applications can use the bandwidth instead as shown in Figure (2.4).

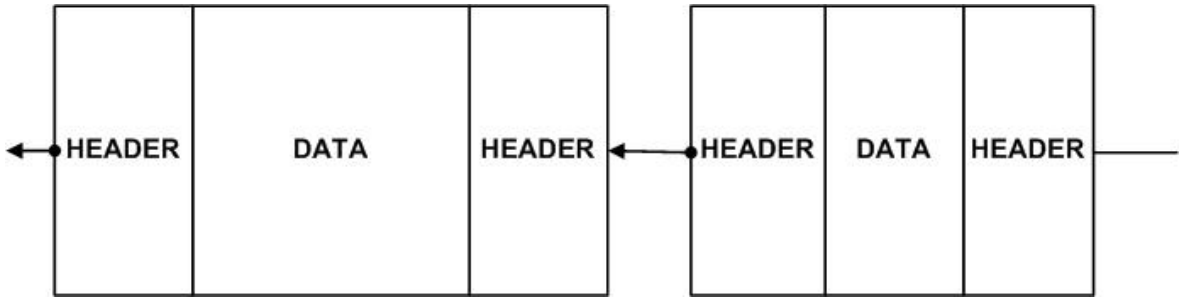

Figure (2.4): Packet data transmission

2.4.3 Asynchronous Transmission

Asynchronous transmission is used in the ATM protocol. This transmission type combines features of packet transmission with features of synchronous transmission. Similarly to synchronous transmission, in asynchronous transmission, the data are transmitted in packets that are rather small, but are all of the same size; these packets are called cells. Similarly to packet transmission, data for one application (one connection) is transmitted in one cell. All cells have the same length as shown in Figure (2.5).

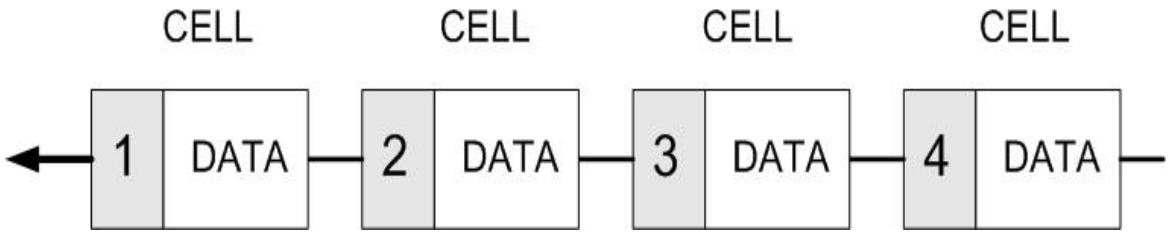

Figure (2.5): Asynchronous data transfer

2.4.4 Virtual Circuit

Some network protocols create virtual circuits in networks. A virtual circuit is conducted through the network and all packets of a particular connection go via the circuit. If the circuit gets interrupted anywhere, then the connection is interrupted, a new circuit is established, and data transmission continues. In Figure (2.6), a virtual circuit between nodes A and D is established via nodes B, F, and G. All packets must go through this circuit. Datagrams can be transmitted via the virtual circuit in two ways:

a) The circuit does not guarantee the datagram's delivery to its destination
b) The virtual circuit can establish a connection and guarantee the data delivery, i.e., the data packets transmitted are numbered and the destination confirms their reception.

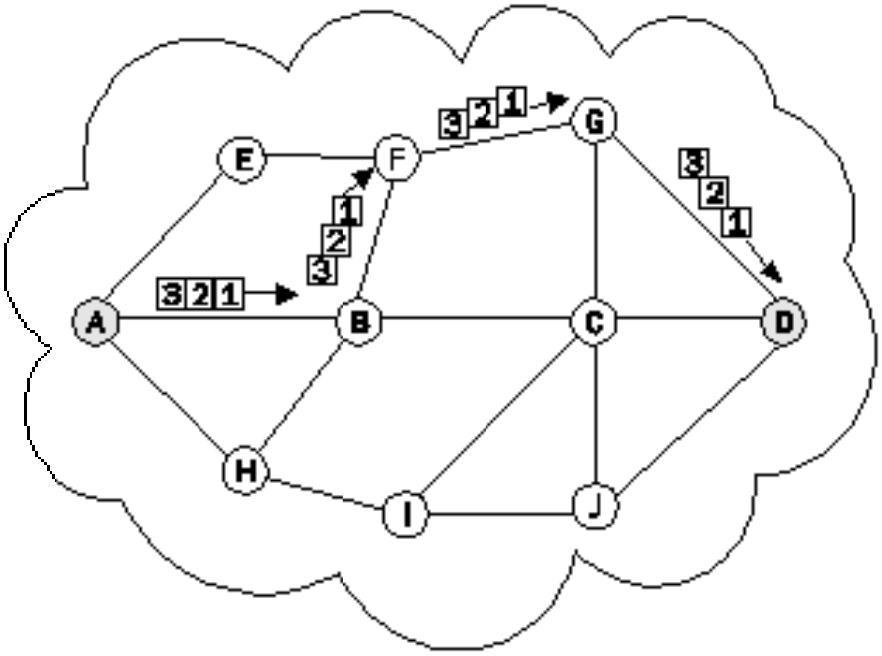

Figure (2.6): Virtual circuit between node A and D

The advantage of virtual circuits is that they are first established (using signalization) and then the data is inserted only into the established circuit. Each packet does not have to carry the globally unique address of the destination (complete routing information) in its header. It only needs the circuit ID.

2.5 Quality of Service (QoS)

Bandwidth is an important subject, more and more people are using the Internet for private and business reasons. The amount of data that must be transmitted through the Internet increases exponential, new applications such as RealAudio, Real Video, Internet Phone software and videoconferencing systems need a lot more bandwidth than the applications that were used in the early years of the Internet, this means that without any bandwidth control, the quality of these real-time streams depends on the bandwidth that is just available. Low bandwidth or better unstable bandwidth causes bad quality in

real-time transmissions, for instance dropouts and hangs [21]. The quality of a service (QoS) is a vague term and can encompass a number of attributes. To some people a good service is one with a low end-to-end delay and high bandwidth, to some people a good service is an extremely reliable one with very few packet drops, while others would enjoy a predictable service regardless of the bandwidth or the end-to-end delay [22]. Various QoS characteristics are divided in two groups, technology- and user-based QoS parameters. Technology-based parameters include delay, response time, jitter, data rate, and loss rate and more. User-based QoS parameters are more subjective and include categories, such as perceived QoS, the visual quality of a streaming video, cost per unit time or per unit of data, and also security, while a user browsing the web and watching public news broadcast would be more interested in the quality of the picture, rather than its security. A user connecting remotely to a corporate network would be most interested in the security of the connection, and less interested in costs [23].

Two different approaches to control the service quality given by internetwork have been suggested, namely IntServ and DiffServ, they stand for Integrated Services and Differentiated Services, respectively. The main difference between them is in the way they control the service given. Generally, IntServ handles the service level by means of admission control and reserved bandwidths for packet streams, while DiffServ handles the service quality in a packet-by-packet manner, using tags in the packet headers [24].

2.5.1 Importance of QoS

In particular, QoS features provide improved and more predictable network service by providing the following services [25]:

- Supporting dedicated bandwidth
- Improving loss characteristics (reduce Packet dropping)
- Avoiding and managing network congestion
- Shaping network traffic
- Setting traffic priorities across the network (arrange data flow)

2.5.2 Quality of Service Parameters

QoS deployment intends to provide a connection with certain performance bounds from the network. Bandwidth, packet delay and jitter, and packet loss are the common measures used to characterize a connection's performance within a network, which are described in the following sections [1].

- **Bandwidth**

The term bandwidth is used to describe the rated throughput capacity of a given medium, protocol, or connection. It effectively describes the "size of the pipe" required for the application to communicate over the network. Generally, a connection requiring guaranteed service has certain bandwidth requirements and wants the network to allocate a minimum bandwidth specifically for it [26].

- **Packet Delay and Jitter**

The network is considered to be *ideal* if it transmits each bit of information with a constant delay equal to the delay of light propagation in a physical medium and the link bandwidth is finite; so the information source transmits packets into the network within certain finite time intervals rather

than instantly. The result of transmitting packets by such an ideal network is illustrated in Figure (2.7), the upper axis shows the times of packet transmission into the network from the source node and the lower axis shows the times of packet arrival to the destination node. In other words, the upper axis shows the offered load of the network, and the lower one demonstrates the results of transmitting this traffic through the network.

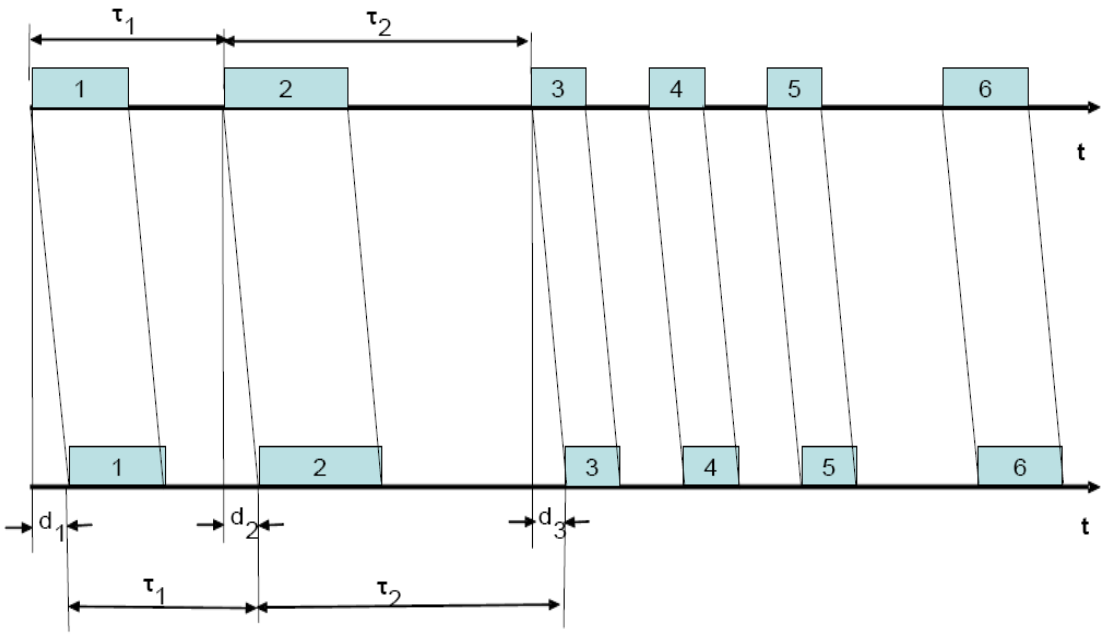

Figure (2.7): Packet transmission in an ideal network where *d* is the delay and τ is delay variation

It can be seen from this illustration, the ideal network:

- Delivers all packets to the destination node, losing or distorting none of them
- Delivers all packets in the same order as they were sent
- Delivers all packets with the minimum delay (*d1* = *d2*, and so on)

It is important that all intervals between adjacent packets remain unchanged. For instance, the interval between the first packet and the second packet was equal to *τ1* when they were sent; this value remained the same when these packets arrived to the destination node. Reliable delivery of all packets with minimum delay and at a constant inter packet interval would satisfy any network user, no matter what kind of traffic is transmitted through the network (Web-service or IP-telephony traffic).

On other hand the *non ideal* case which is the real case, a variable delay are exists, as shown in Figure (2.8).

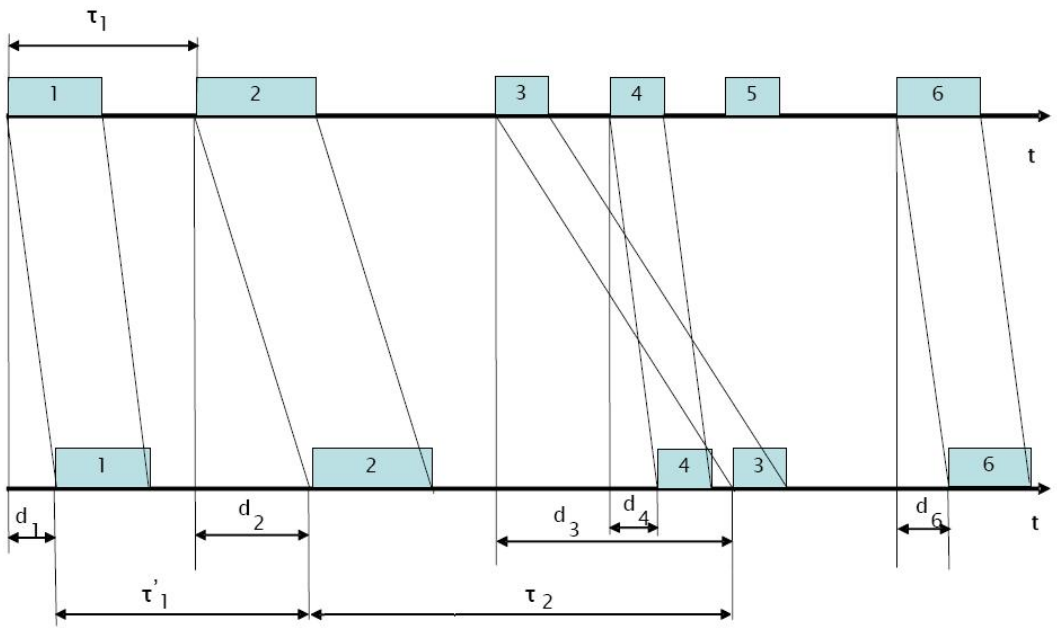

Figure (2.8): Packet transmission in a real network

The characteristics of such system are:

- Packets are delivered to the destination node with variable delays. This is a generic property of the packet-switched networks. The random nature of the queuing process results in variable delays; at the same time, delays in the delivery of some packets might be considerable tens

of times more than the average delay value ($d1 \neq d2 \neq d3$, and so on). Variable delays result in uneven inter packet intervals, so that, for example, $\tau 1 \neq \tau 2$. Thus, the time correlation between adjacent packet changes, and this, in turn, may result in critical failures of some applications. For instance, during digital voice transmission, the unevenness of inter packet intervals results in significant voice distortions.

- Packets can be delivered to the destination node in an order different from the order in which they were sent (packet reordering effect).
- Packets can be lost or corrupted. The latter situation is equivalent to packet loss, since most protocols cannot restore damaged data. In most cases, protocols are only able to detect that data corruption occurred, using the Frame Check Sequence (FCS) which is on the IP header.
- The average rate of information flow at the input of the destination node can differ from the average rate of the information flow sent into the network by the source node. This is because of packet losses rather than packet delays.

Obviously, the set of values of the transmission times of each individual packet provides a comprehensive characteristic of the traffic transmission quality. However, this characteristic of network performance is too bulky and redundant.

From above the delay or End-to-end latency refers to the total transit time for packets in a data stream to arrive at the remote endpoint, so the average delay (D) is the summation of all delays (di), divided by the total number of all measurements (N), which can be calculated by equation (2.1) [27].

$$D = \sum_{i=1}^{N} d_i \Big/ N \qquad \ldots (2.1)$$

The variation in the packet arrival times is called jitter; the mathematical term for this value is standard deviation. Packet jitter is important because it estimates the maximum delays between packet receptions at the receiver against individual packet delay. A receiver, depending on the application, can offset the jitter by adding a receive buffer that could store packets up to the jitter bound. Playback applications that send continuous information stream including applications such as interactive voice calls, videoconferencing, and distribution all into this category [28]. Equation (2.2) shows the calculation of jitter (J).

$$J = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d_i - D)^2} \qquad \ldots (2.2)$$

Both average delay and jitter are measured in seconds. Obviously, if all *di* delay values are equal, then $D = d_i$ and $J = 0$ (i.e., there is no jitter).The ratio of jitter to average delay is called Coefficient of variation (C_v), the calculation of C_v is shown in equation (2.3).

$$C_v = J/D \qquad \ldots (2.3)$$

Coefficient of variation characterizes the traffic without relating it to absolute values of the time scale. Ideal synchronous flow (stream) will always have a standard deviation of 0. If the coefficient of variation is equal to 1, then the traffic is bursty [27].

- **Packet Loss**

This characteristic is defined as a ratio of the number of lost packets to the total number of transmitted packets as shown in equation (2.4):

$$Loss\ Packets\ ratio = N_L / N \quad \ldots (2.4)$$

Where N equals the total number of packets transmitted during a specific time period, and N_L equals the number of packets lost during the same time period [27]. Packet drops at network congestion points and corrupted packets on the transmission wire cause packet loss, also packet drops generally occur at congestion points when incoming packets far exceed the queue size limit at the output queue. They also occur due to insufficient input buffers on packet arrival. Packet loss is generally specified as a fraction of packets lost while transmitting a certain number of packets over some time interval [26].

2.5.3 Applications and QoS flow parameters

A stream of packets from a source to a destination is called a flow. In a connection-oriented network, all the packets belonging to a flow follow the same route, while in a connectionless network, they may follow different routes. The needs of each flow can be characterized by four primary parameters: reliability (packet loss), delay, jitter and bandwidth which are described before. Together these determine the QoS flow requirement. Several common applications and the stringency of their requirements are listed in Table (2.2) [1].

Table (2.2): Stringent of the QoS requirements [1]

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low

The first four applications have stringent requirements on reliability. No bits may be delivered incorrectly. This goal is usually achieved by checksumming each packet and verifying the checksum at the destination. If a packet is damaged in transit, it is not acknowledged and will be retransmitted eventually. This strategy gives high reliability.

The three final (audio/video) applications can tolerate errors, so no checksums are computed or verified. File transfer applications, including e-mail and video, are not delay sensitive. If all packets are delayed uniformly by a few seconds, no harm is done. Interactive applications, such as Web surfing and remote login, are more delay sensitive. Real-time applications, such as telephony and video conferencing have strict delay requirements. If all the words

in a telephone call are each delayed by exactly 2 seconds, the users will find the connection unacceptable. On the other hand, playing audio or video files from a server does not require low delay. The first three applications are not sensitive to the packets arriving with irregular time intervals between them. Remote login is somewhat sensitive to that, since characters on the screen will appear in little bursts if the connection suffers much jitter. Video and especially audio are extremely sensitive to jitter. If a user is watching a video over the network and the frames are all delayed by exactly 2 seconds, no harm is done. But if the transmission time varies randomly between 1 and 2 seconds, the result will be terrible. For audio, a jitter of even a few milliseconds is clearly audible. Finally, the applications differ in their bandwidth needs, with e-mail and remote login not needing much [1]. The International Telecommunication Union (ITU) considers network delay for voice applications in recommendation; this recommendation defines three bands of one way delay as shown in Table (2.3) [28].

Video communication is a new application for most IP networks, unlike e-mail or typical database applications, video conferencing required limits on the total end-to-end delay and jitter, which are result in packet loss and responsible for degrading audio and video quality and usability. The overall delay budget for a one-way video or voice conversation must be approximately 150-200 milliseconds [29]. Four applications will be used in this thesis which are: file transfer, Email, voice and video conferencing.

Table (2.3): Delay specifications

Range in Milliseconds	Description
0-150	Acceptable for most user applications
150-400	Acceptable provided that administrators are aware of the transmission time and the impact it has on the transmission quality of user applications
Above 400	Unacceptable for general network planning purposes. However, it is recognized that in some exceptional cases this limit is exceeded

2.5.4 Type of Service (ToS)

The IP header Type of Service (ToS) field consists of 3 bit IP precedence bits and followed by 4 ToS bits and an unused bit, as shown in Figure (2.9).

Figure (2.9): TOS bytes in IP header, where (P2-P0): IP Precedence, (T3-T0): ToS and CU: Currently Unused [30]

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above certain precedence at time of high load) [19]. The IP Precedence field in the packet's IP header is used to indicate the relative priority with which a particular packet should be handled. Apart from IP Precedence, the ToS byte contains ToS bits. ToS bits were designed to contain values indicating how each packet should be handled in a network, but this particular field is never used much in the real world [30]. Tables (2.4) describe the IP precedence values and names. Table (2.5) shows the different values that the ToS can handle.

Table (2.4): IP Precedence values and names

IP Precedence Value	IP Precedence Bits	IP Precedence Names
0	000	Routine
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash Override
5	101	Critical
6	110	Internetwork Control
7	111	Network Control

Table (2.5): Values for Type of Service (how to treat the packet)

Number of bit	Value	
	0	**1**
bit 3	Normal Delay	Low Delay
bit 4	Normal Throughput	High Throughput
bit 5	Normal Reliability	High Reliability
bit 6	Normal monetary cost	Low monetary cost

2.5.5 QoS Router Building Blocks

Services are constructed from somewhat independent logical building blocks. Depending on their specific instantiation and combination, numerous types of QoS architectures can be formed. An overview of the block types in routers is shown in Figure (2.10), which combined:

1. Packet classification: If any kind of service is to be provided, packets must first be classified according to header properties. For instance, in order to reserve bandwidth for a particular end-to-end data flow, it is necessary to distinguish the IP addresses of the sender and receiver as well as ports and the protocol number. Such packet detection is difficult because of mechanisms like packet fragmentation, header compression and encryption. ToS octets for IP header fit in this category [31].
2. Meter: A meter monitors traffic characteristics and provides information to other blocks.

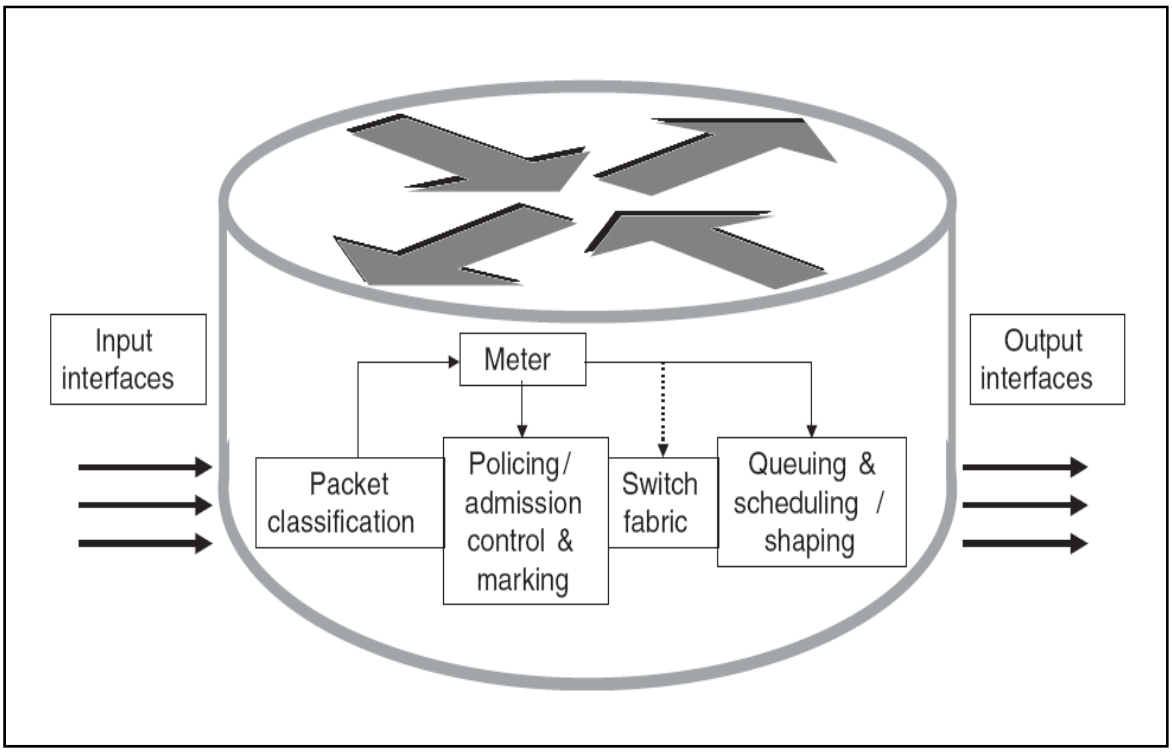

Figure (2.10): A generic QoS router

3. Policing: Under certain circumstances, packets are policed (dropped); the reason for doing so is to enforce conforming behavior. For each input flow, in each switch there is an appropriate set of QoS parameters known as traffic profiles. Policing assumes the check of correspondence of each input flow to the parameters of its profile. There are algorithms that allow this check to be accomplished automatically at the same rate at which packets arrive to the input interface of the switch, if profile parameters are violated (if the burst size or average rate are exceeded), the packets of such a flow are discarded or marked [22].
4. Admission control: When each flow is offered to a router, it has to decide based on its capacity and how many commitments it has already made for other flows, whether to admit or reject the flow. This decision is not simple matter of comparing the (bandwidth, buffers, CPU cycles)

requested by the flow with the router's excess capacity in those three dimensions [1].

5. Marking: Marking of packets facilitates their detection; this is usually done by changing something in the header. This means that, instead of carrying out the expensive multi-field classification process, packets can later be classified by simply looking at one header entry. This operation can be carried out by the router that marked the packet, but it could just as well be another router in the same domain [31].
6. Switch(ing) fabric: The switch fabric is the logical block where routing table lookups are performed and it is decided where a packet will be sent [31].
7. Queuing: This block represents queuing methods of all kinds, for example standard FIFO queuing or active queue management [31], these methods will be described with details in the next section.
8. Scheduling: If a router is handling multiple flows, there is a danger that one flow will hog too much of its capacity and starve all the other flows. Processing packets in the order of their arrival means that an aggressive sender can capture most of the capacity of the routers its packets traverse, reducing the quality of service for others [1]. Scheduling decides when a packet should be removed from which queue, the simplest form of such a mechanism is a round robin strategy, but there are more complex variants; one example is Fair Queuing (FQ), which emulates bitwise interleaving of packets from each queue. Also, there is its weighted Fair Queuing (WFQ), which makes it possible to hierarchically divide the bandwidth of a link [31]. Scheduling dictates the temporal characteristics of packet departures from each queue typically at the output interface toward the next router or host, but potentially at other queuing points within a router.

Traditional routers have only a single queue per output link interface. So, the scheduling task is simple pull packets out of the queue as fast as the underlying link can transmit them [32].

9. Shaping: Mainly, this function is used to smooth the traffic burst to make the output flow more uniform than the flow at the device input. Burst smoothening helps to decrease queues in network devices that will process the traffic after the given device. It is also expedient to use it for restoring the time relationships of applications traffic working with streams, such as voice applications [27]. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets [22].

CHAPTER THREE

CONGESTION MANAGEMENT

3.1 Introduction

This chapter presented many methods and ways that will be used in this thesis; which all try to achieve good quality of service to manage the high traffic and congestion problem that decrease the performance of the network.

3.2 Congestion Management [25]

Congestion management features allow controlling congestion by determining the order in which packets are sent out to an interface based on priorities assigned to those packets. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. The congestion management QoS feature offers four types of queuing protocols; each one allows to specify creation of a different number of queues, affording greater or lesser degrees of differentiation of traffic, and to specify the order in which that traffic is sent. During periods with light traffic, that is, when no congestion exists, packets are sent out the interface as soon as they arrive. During periods of transmit congestion at the outgoing interface, packets arrive faster than the interface can send them. If congestion management features were used, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled for transmission according to their assigned priority and the queuing mechanism configured for the interface. The router determines the order of packet transmission by controlling which packets are placed in which queue and how queues are serviced with respect to each other. Congestion management features operate to control congestion once it occurs. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, and then determine some method of prioritizing it onto an

output link. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance.

3.2.1 Importance

The goal of congestion control mechanisms is simply to use the network as efficiently as possible by attain the highest possible throughput with low loss ratio and small delay. These days, networks are often over provisioned, and the underlying question has shifted from ‘how to eliminate congestion’ to ‘how to efficiently use all the available capacity. Efficiently using the network means answering both these questions at the same time; this is what good congestion control mechanisms do [31]. Heterogeneous networks include many different protocols used by applications, giving rise to the need to prioritize traffic in order to satisfy time-critical applications while still addressing the needs of less time-dependent applications, such as file transfer. Different types of traffic sharing a data path through the network can interact with one another in ways that affect their application performance. If the network is designed to support different traffic types that share a single data path between routers, congestion management techniques must be used to ensure fairness of treatment across the various traffic types [24].

2.3.2 Types of Queuing [25]

There are four types of queuing, which constitute the congestion management QoS features which are FIFO (First-In-First-Out), WFQ (Weight Fair Queuing), CQ (Custom Queuing) and PQ (Priority Queuing).

a) **FIFO**: In its simplest form, which also known as first-come, first-served (FCFS) queuing, involves buffering and forwarding of packets in the order of arrival. FIFO embodies no concept of priority or classes

of traffic and consequently makes no decision about packet priority. There is only one queue, and all packets are treated equally. Packets are sent out an interface in the order in which they arrive. When FIFO is used, ill-behaved sources can consume all the bandwidth, bursty sources can cause delays in time-sensitive or important traffic, and important traffic can be dropped because less important traffic fills the queue. FIFO, which is the fastest method of queuing, is effective for large links that have little delay and minimal congestion.

b) **WFQ**: Weighted Fair Queuing is a dynamic scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priority, or weights, to identified traffic to classify traffic into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. WFQ is a flow-based algorithm that simultaneously schedules interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows. WFQ gives concurrent file transfers balanced use of link capacity; that is, when multiple file transfers occur, the transfers are given comparable bandwidth. Figure (3.1) shows how WFQ works. WFQ provides traffic priority management, which dynamically sorts traffic into messages that make up a conversation. WFQ breaks up the train of packets within a conversation to ensure that bandwidth is shared fairly between individual conversations and that low-volume traffic is transferred in a timely fashion. WFQ places packets of the various conversations in the fair queues before transmission. The order of removal from the fair queues is determined by the virtual time of the delivery of the last bit of each arriving packet.

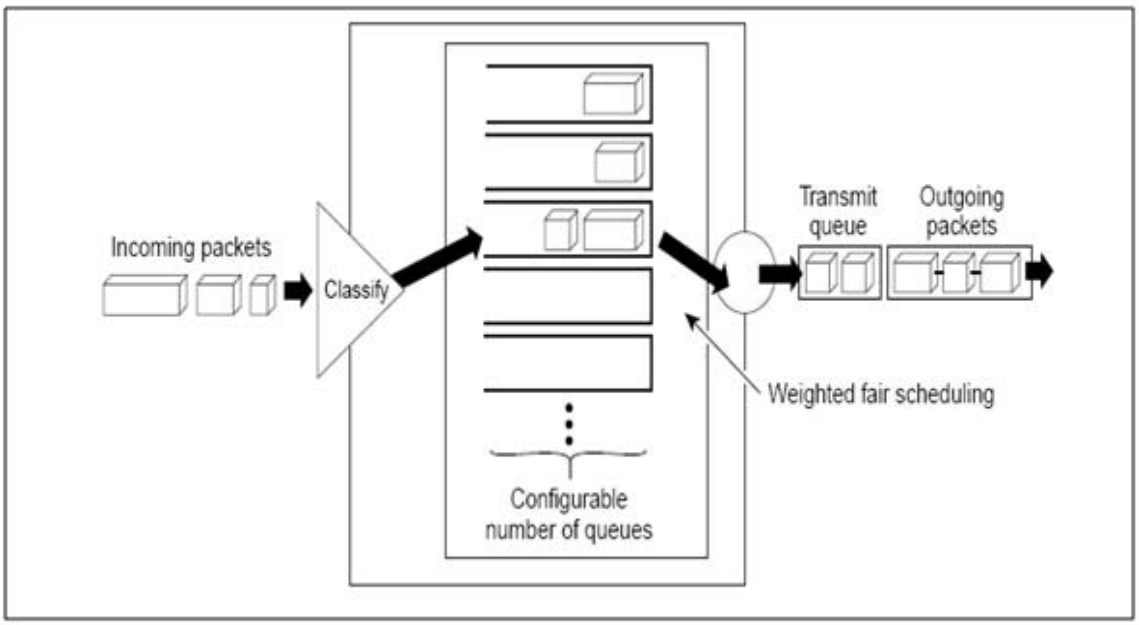

Figure (3.1): Weighted Fair Queuing process

WFQ can manage duplex data streams, such as those between pairs of applications, and simplex data streams such as voice or video. WFQ provides the solution for situations in which it is desirable to provide consistent response time to heavy and light network users alike without adding excessive bandwidth. WFQ automatically adapts to changing network traffic conditions. The restrictions of this type of queue are that the WFQ is not supported with tunneling and encryption because these features modify the packet content information required by WFQ for classification. Although WFQ automatically adapts to changing network traffic conditions, it does not offer the degree of precision control over bandwidth allocation that CQ offer. WFQ is IP precedence-aware. It can detect higher priority packets marked with precedence by the IP Forwarder and can schedule them faster,

providing superior response time for this traffic. Thus, as the precedence increases, WFQ allocates more bandwidth to the conversation during periods of congestion. WFQ assigns a weight to each flow, which determines the transmit order for queued packets. In this scheme, lower weights are served first.

c) **CQ**: Custom Queuing allow to specify a certain number of bytes to forward from a queue each time the queue is serviced, thereby allowing you to share the network resources among applications with specific minimum bandwidth or latency requirements. A maximum number of packets in each queue can be specified. CQ handles traffic by specifying the number of packets or bytes to be serviced for each class of traffic. It services the queues by cycling through them in round-robin fashion, sending the portion of allocated bandwidth for each queue before moving to the next queue. If one queue is empty, the router will send packets from the next queue that has packets ready to send. When CQ is enabled on an interface, the system maintains 17 output queues for that interface, queues 1 through 16 can be specified. Associated with each output queue is a configurable byte count, which specifies how many bytes of data the system should deliver from the current queue before it moves on to the next queue. Queue number 0 is a system queue; it is emptied before any of the queues numbered 1 through 16 are processed. The system queues high priority to packet like keep alive and signaling packets. Other traffic cannot be configured to use this queue. CQ ensures that no application or specified group of applications achieves more than a predetermined proportion of overall capacity when the line is under stress. Like PQ (will be described later), CQ is statically configured and does not

automatically adapt to changing network conditions. Figure (3.2) shows how CQ behaves.

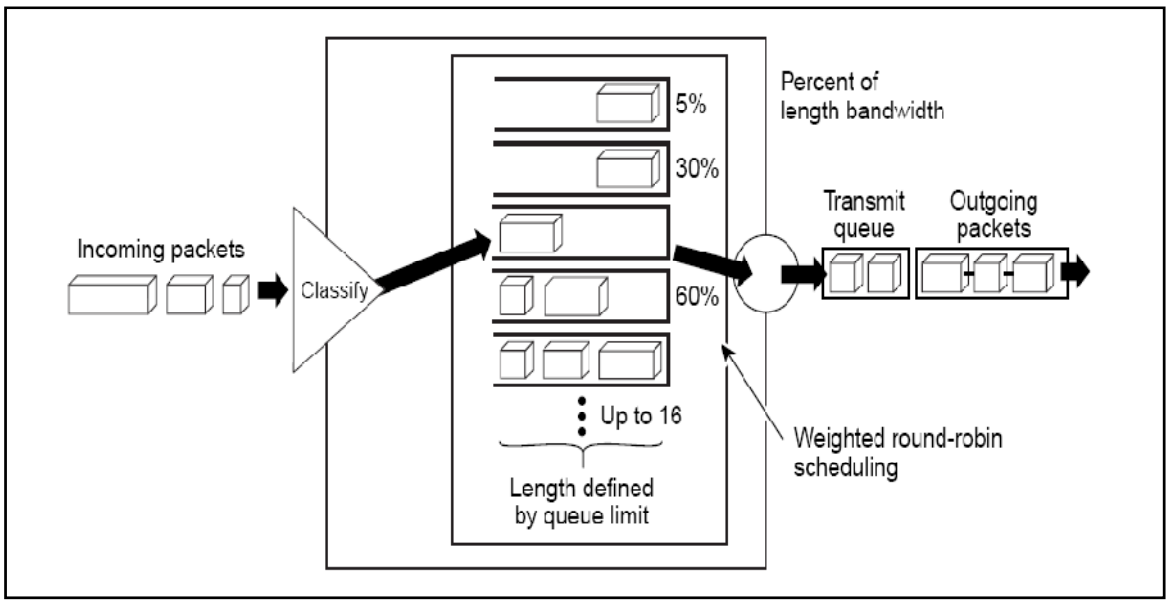

Figure (3.2): Custom queuing behaviors

The restrictions of this method are that the CQ is statically configured and does not adapt to changing network conditions. With CQ enabled, the system takes longer to switch packets than FIFO because the packets are classified by the processor card.

d) **PQ**: Priority queuing allow to define how traffic is prioritized in the network. Four traffic priorities can be configured, a series of filters based on packet characteristics can be defined to cause the router to place traffic into these four queues; the queue with the highest priority is serviced first until it is empty, then the lower queues are serviced in sequence. During transmission, PQ gives priority queues absolute preferential treatment over low priority queues; important traffic, given the highest priority, always takes precedence over less important traffic.

Packets are classified based on user-specified criteria and placed into one of the four output queues (high, medium, normal, and low) based on the assigned priority. Packets that are not classified by priority fall into the normal queue. Figure (3.3) illustrates this process.

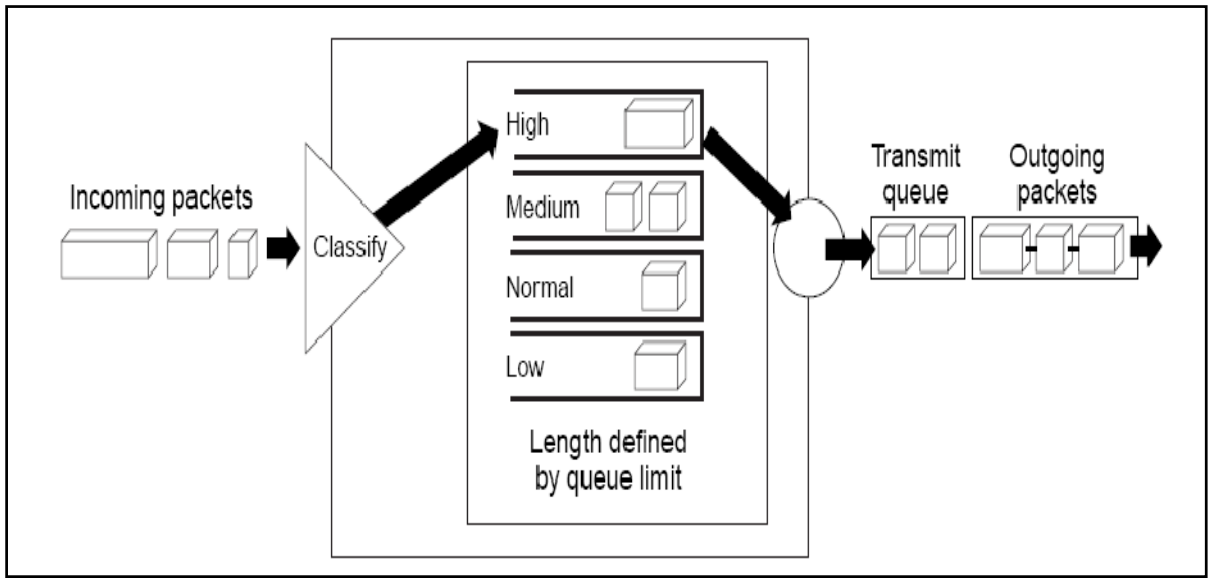

Figure (3.3): Priority queuing process

When a packet is to be sent out an interface, the priority queues on that interface are scanned for packets in descending order of priority. The high priority queue is scanned first, then the medium priority queue, and so on. The packet at the head of the highest queue is chosen for transmission. This procedure is repeated every time a packet is to be sent. The maximum length of a queue is defined by the length limit. When a queue is longer than the queue limit, all additional packets are dropped. A priority list is a set of rules that describe how packets should be assigned to priority queues. A priority list might also describe a default priority or the queue size limits of the various priority queues. Packets can be classified by the following criteria: Protocol or sub protocol type, Incoming interface, Packet size, Fragments and Access

list. PQ provides absolute preferential treatment to high priority traffic, ensuring that mission-critical traffic traversing various WAN links gets priority treatment. In addition, PQ provides a faster response time than do other methods of queuing. Priority output queuing can be enabled for any interface; it is best used for low bandwidth, congested serial interfaces. The restrictions of PQ are lower priority traffic never being sent because lower priority traffic is often denied bandwidth in favor of higher priority traffic, and PQ introduces extra overhead that is acceptable for slow interfaces, but may not be acceptable for higher speed interfaces such as Ethernet. With PQ enabled, the system takes longer to switch packets because the packets are classified by the processor card and PQ uses a static configuration and does not adapt to changing network conditions, also PQ is not supported on any tunnels.

3.3 Some Techniques to Achieve Good Quality of Service

Quality of service requirements must be achieved to produced a good QoS, some of these techniques were explain previously like traffic shaping, admission control and packet scheduling, etc. But there is two algorithms which deals directly with the internal queue for each interface, which are “The Leaky Bucket Algorithm” and “The Token Bucket Algorithm” that will be described below.

3.3.1 The Leaky Bucket Algorithm [1]

Imagine a bucket with a small hole in the bottom. No matter the rate at which water enters the bucket, the outflow is at a constant rate (p), also this constant rate is zero when the bucket is empty. Also, once the bucket is full, any additional water entering it spills over the sides and is lost. The same idea can be applied to packets, as shown in Figure (3.4).

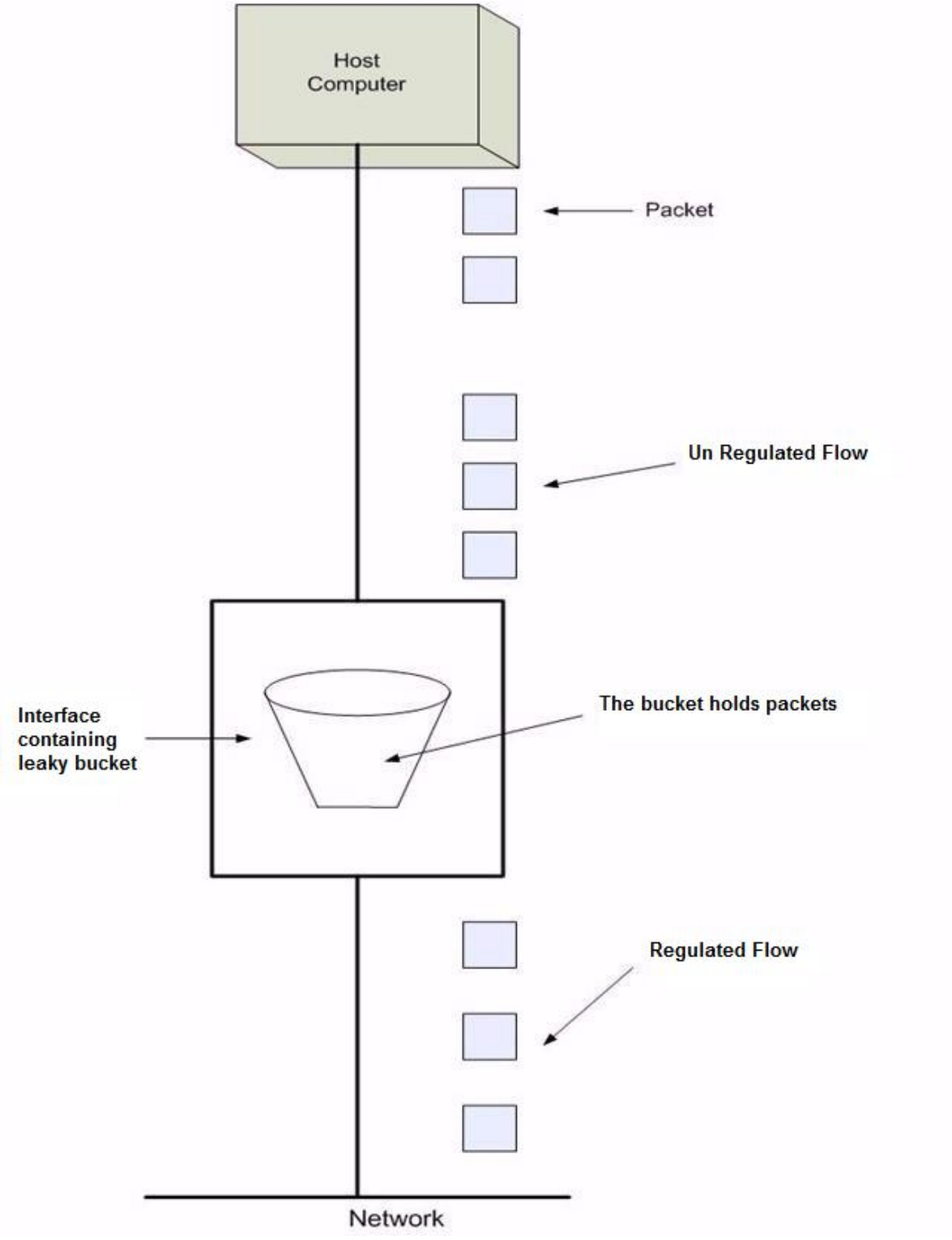

Figure (3.4): A leaky bucket with packets [1]

Conceptually, each host is connected to the network by an interface containing a lucky bucket, that is, a finite internal queue. If a packet arrives

at the queue when it is full, the packet is discarded. In other words, if one or more processes within the host try to send a packet when the maximum number is already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulated by the host operating system. This algorithm is called the *leaky bucket algorithm***.**

In fact, it is nothing other than a single-server queuing system with constant service time. The host is allowed to put one packet per clock tick onto the network. Again this can be enforced by the interface card or by the operating system. This mechanism turns an uneven flow of packets from the user processes inside the host into even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion. Implementing the original leaky bucket algorithm is easy. The leaky bucket consists of a finite queue. When a packet arrives, if there is room on the queue it is appended to the queue; otherwise, it is discarded. At every clock tick, one packet is transmitted (unless the queue is empty). The byte-counting leaky bucket is implemented almost the same way. At each tick, a counter is initialized to *n.* If the first packet on the queue has fewer s than the current value of the counter, it is transmitted, and the counter is decremented by that number of bytes. Additional packets may also be sent, as long as the counter is high enough. When the counter drops below the length of next packet on the queue, transmission stops until the next tick, at which time the residual byte count is reset and the flow can continue.

3.3.2 The Token Bucket Algorithm

The leaky bucket algorithm enforces a rigid output pattern at the average rate no matter how bursty the traffic is. For many applications, it is better to allow the output to speed up somewhat when large bursts arrive, so a more

flexible algorithm is needed, preferably one that never loses data. One such algorithm is the *token bucket algorithm*. In this algorithm, the leaky bucket holds token, generated by a clock at the rate of one token every ΔT sec. Figure (3.5-a) shows a bucket holding three tokens, with five packets waiting to be transmitted, for a packet to be transmitted, it must capture and destroy one token, while in Figure (3.5-b), three of the five packets have gotten through, but the other two are stuck waiting for two more tokens to be generated.

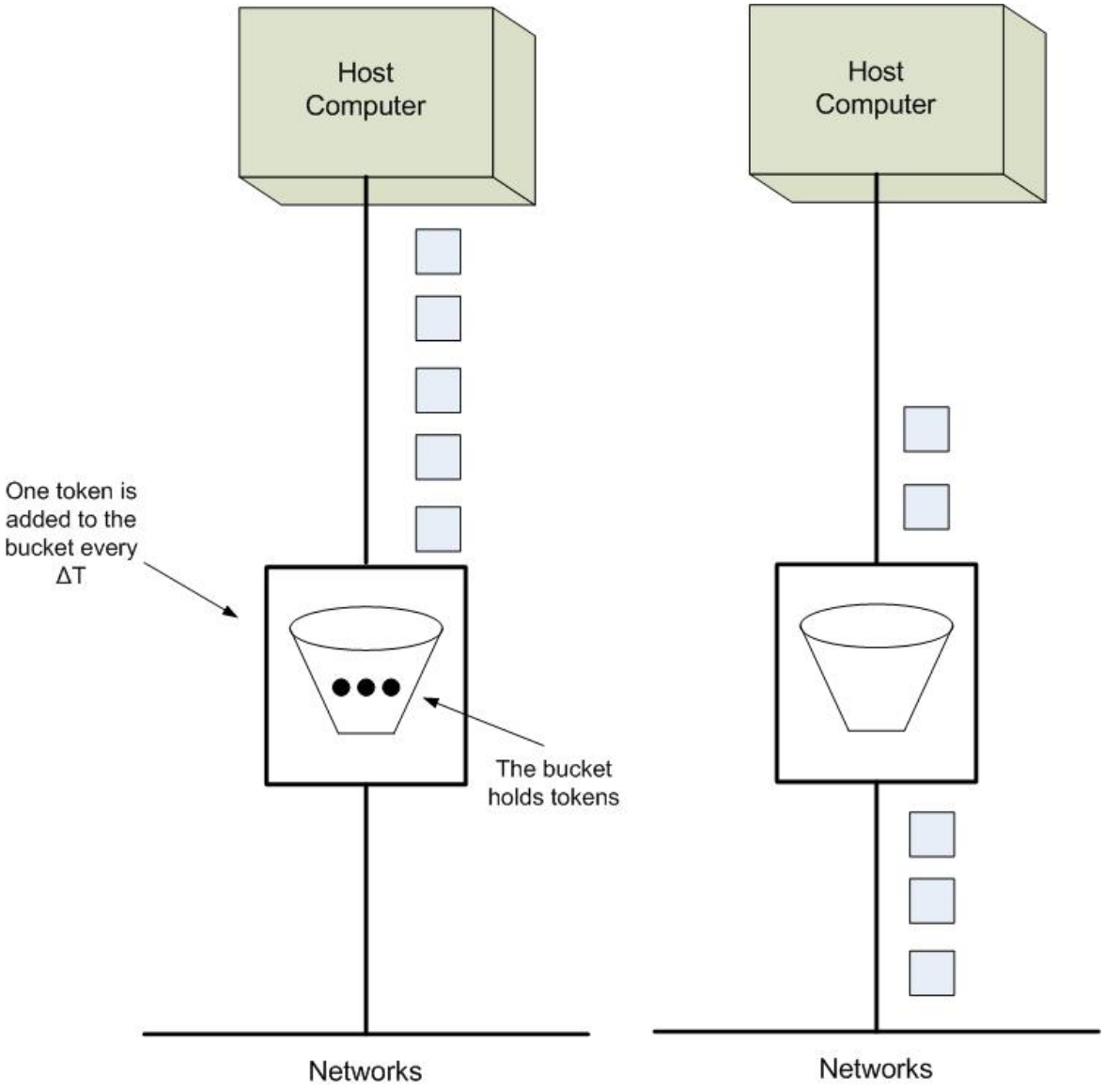

Figure (3.5): The token bucket algorithm. (a) Before. (b) After. [1]

The token bucket algorithm provides a different kind of traffic shaping than that of the leaky bucket algorithm. The leaky bucket algorithm does not allow idle hosts to save up permission to send large bursts later. The token algorithm does allow saving, up to the maximum size of the bucket, *n.* This property means that bursts of up to *n* packets can be sent at once, allowing some burstiness in the output stream and giving faster response to sudden bursts of input. Another difference between the two algorithms is that the token bucket algorithm throws away tokens (i.e., transmission capacity) when the bucket fills up but never discards packets. In contrast, the leaky bucket algorithm discards packet when the bucket fills up. The leaky bucket and token bucket algorithms can also be used to smooth traffic between routers, as well as to regulate host output; also one clear difference is that a token bucket regulating a host can make the host stop sending when the rules say it must. Telling a router to stop sending while its input keeps pouring in may result on lost data [1].

Token bucket has three key parameters [26]:

1) Mean rate or Committed Information Rate (CIR), in bits per second: On average, traffic does not exceed CIR.
2) Conformed burst size (B_C), in number of bytes: This is the amount of traffic allowed to exceed the token bucket on an instantaneous basis. It is also occasionally referred to as normal burst size.
3) Extended burst size (B_E), in number of bytes: This is the bonus round. It allows a reduced percentage of traffic to conform between the conform burst and extended burst which will be explain later.

3.3.3 Traffic Rate Management

To offer QoS in a network, traffic entering the service provider network needs to be policed on the network boundary routers to make sure the traffic rate stays within the service limit. Even if a few routers at the network boundary start sending more traffic than what the network core is provisioned to handle, the increased traffic load leads to network congestion. The degraded performance in the network makes it impossible to deliver QoS for all the network traffic. Traffic policing functions using the CAR (Committed Access Rate) feature manage traffic rate by treat traffic when tokens are exhausted. The concept of tokens comes from the token bucket scheme [26]. The traffic policing or rate-limiting function is provided by CAR, which have two functions:

1) Packet Classification, using IP Precedence and QoS group setting
2) Access Bandwidth Management, through rate limiting

As a traffic policing function, CAR doesn't buffer or smooth traffic and might drop packets when the allowed bursting capability is exceeded. CAR is implemented as a list of rate-limit statements. These statements can be applied to both the output and input traffic on an interface [31].

Each rate-limit statement is made up of three elements. The rate-limit statements are processed, as shown in Figure (3.6), which are [26]:

A. The traffic matching specification defines what packets are of interest for CAR. Each rate-limit statement is checked sequentially for a match. When a match occurs, the token bucket is evaluated. If the action is a continue action, it resumes looking for matches in subsequent rate

limits. Note that you can define a match specification to match every packet.

B. A token bucket is used as a means of measuring traffic. The size of the token bucket (the maximum number of tokens it can hold) is equal to the conformed burst (B_C). For each packet for which the CAR limit is applied, tokens are removed from the bucket in accordance to the packet's byte size.

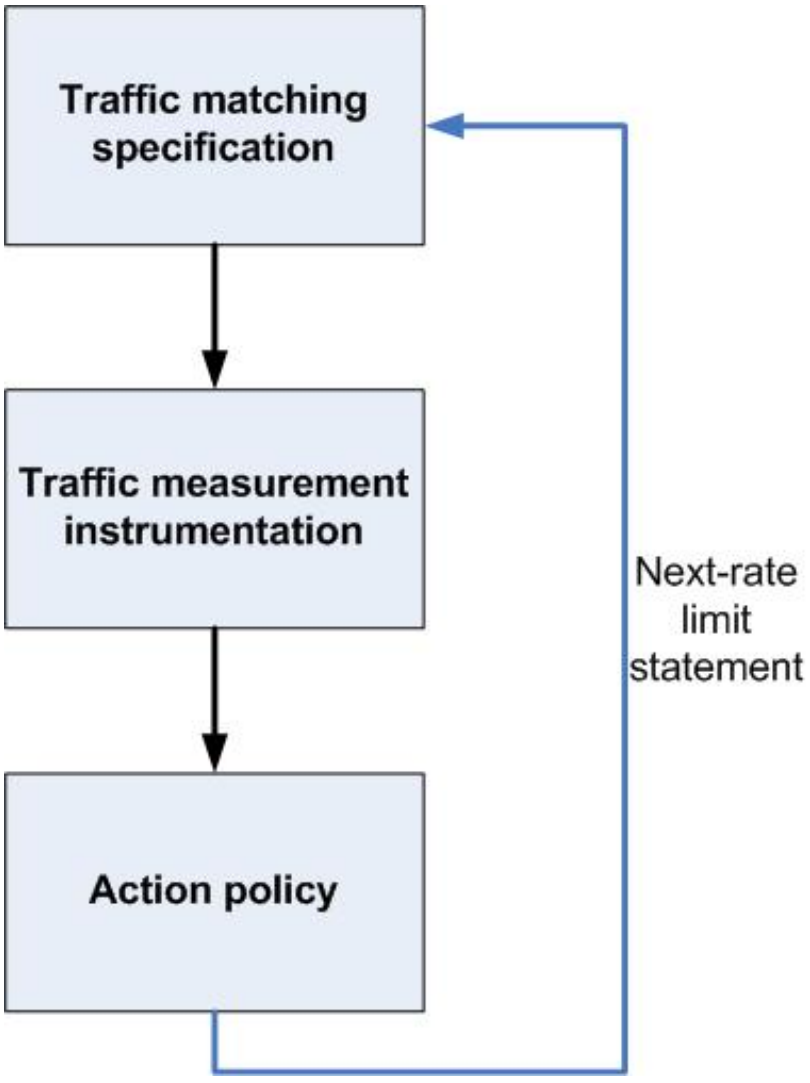

Figure (3.6): The Evaluation Flow of Rate-Limit Statements

When enough tokens are available to service a packet, the packet is transmitted. If a packet arrives and the number of available tokens in the bucket is less than the packet's byte size, however, the extended burst (B_E) comes into play. Consider the following two cases:

1. Standard token bucket, where $B_E = B_C$

   A standard token bucket has no extended burst capability, and its extended burst is equal to its B_C. In this case, you drop the packet when tokens are unavailable.

2. Token bucket with extended burst capability, where $B_E > B_C$

   A token bucket with an extended burst capability allows a stream to borrow more tokens, unlike the standard token bucket scheme.

Here two terms must be explained to describe the concerns of borrowing; these two terms are actual debt (D_A) and compounded debt (D_C). D_A is the number of tokens the stream currently borrowed. This is reduced at regular intervals, determined by the configured committed rate by the accumulation of tokens. Say you borrow 100 tokens for each of the three packets you send after the last packet drop. The D_A is 100, 200, and 300 after the first, second, and third packets are sent, respectively. D_C is the sum of the D_A of all packets sent since the last time a packet was dropped. Unlike D_A, which is an actual count of the borrowed tokens since the last packet drop, D_C is the sum of the actual debts for all the packets that borrowed tokens since the last CAR packet drop. Say, as in the previous example, you borrow 100 tokens for each of the three packets you send after the last packet drop. D_C equals 100, 300 (= 100 + 200), and 600 (= 100 + 200 + 300) after the first, second, and third packets are sent, respectively. Note that for the first packet that needs to borrow tokens after a packet drop, D_C is equal to D_A. The

D_C value is set to zero after a packet is dropped, and the next packet that needs to borrow has a new value computed, which is equal to the D_A. In the example, if the fourth packet gets dropped, the next packet that needs to borrow tokens (for example, 100) has its $D_C = D_A = 400$ (= 300 + 100). Unlike D_C, D_A is not forgiven after a packet drop. If D_A is greater than the extended limit, all packets are dropped until D_A is reduced through accumulation of tokens. If a packet arrives and needs to borrow some tokens, a comparison is made between B_E and D_C . If D_C is greater than B_E , the packet is dropped and D_C is set to zero. Otherwise, the packet is sent, and D_A is incremented by the number of tokens borrowed and D_C with the newly computed D_A value. If a packet is dropped because the number of available tokens exceeds the packet size (in bytes), tokens will not be removed from the bucket (for example, dropped packets do not count against any rate or burst limits). It is important to note that CIR is a rate expressed in bytes per second. The bursts are expressed in bytes. A burst counter counts the current burst size. The burst counter can either be less than or greater than the B_C. When the burst counter exceeds B_C, the burst counter equals B_C + D_A. When a packet arrives, the burst counter is evaluated, as shown in Figure (3.7). For cases when the burst counter value is between B_C and B_E, the exceed action probability can be represented as in equation (3.1):

$$Excced\ action\ probability = (Burst\ counter - B_C)/(B_E - B_C) \quad \ldots (3.1)$$

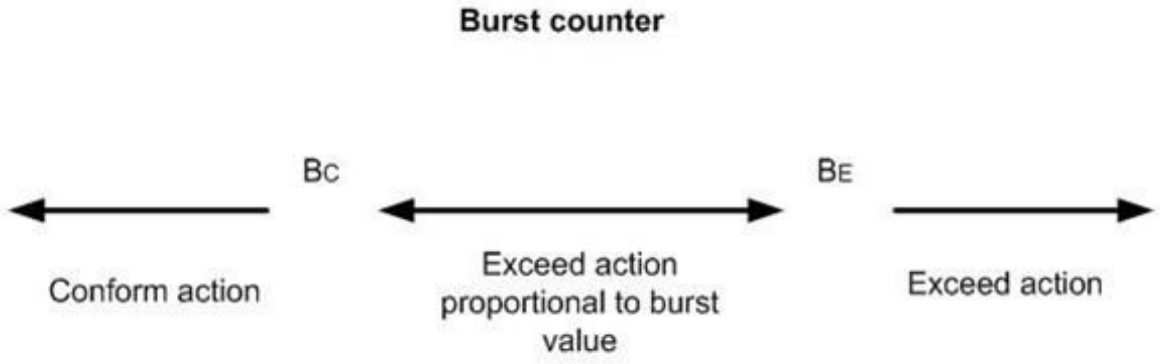

Figure (3.7): Action Based on the Burst Counter Value

Based on this approximation, the Commit access rate packet drop probability is shown in Figure (3.8)

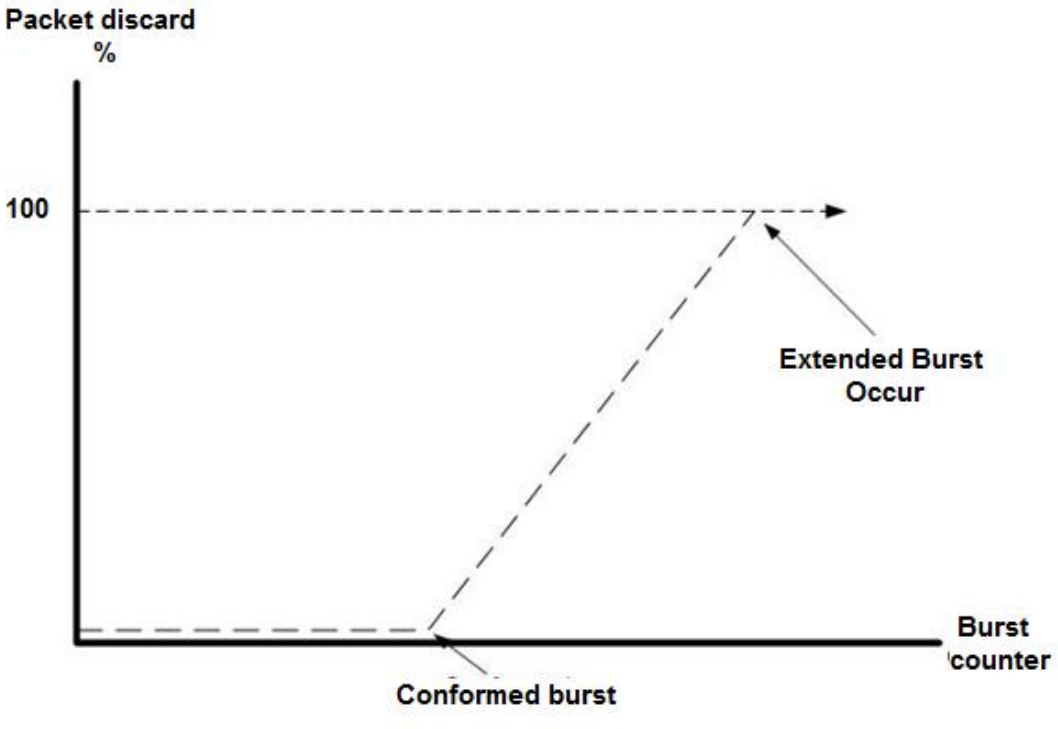

Figure (3.8): CAR packet drop probability

C. Separate action policies can be defined for conforming and exceeding traffic for each rate-limit statement. A *conform-action* or *exceed-action* could be one of the following:

- Transmit
- Drop
- Continue (go to next rate limit statement in the list)
- Set precedence and transmit
- Set precedence and continue

CHAPTER FOUR

WAN SIMULATION ENHANCEMENTS AND EVALUATION

4.1 Introduction

This chapter will present the design and simulation results for some important techniques that must be existence to manage the problem of congestion in a wide area network. These techniques include quality of service congestion management (types of scheduling) and committed access rate (CAR).

In this thesis, two approaches of investigation were carried out. The first implements the QoS technique. This approach consists of four scenarios; each scenario will implement a four application; which are File Transfer, E-mail (Electronic mail), Voice and Video conferencing, also there is a scenario which are none of these techniques are used. The type of services (ToS) used is background, standard, excellent effort and streaming multimedia. The second approach implements the CAR algorithm.

4.2 Networks Topology and Components

Before doing the simulation, there is a brief description about the designed network. The main network (Iraq map) for this simulation is shown in Figure (4.1), and the sub-networks are as shown in Figures (4.2-a-b-c-e) which are Baghdad, Basrah, Mosul and Anbar governorates respectively.

The components which are consisting on the above figures are:

A. **Server**: The server model represents a server node with server applications running over TCP/IP and UDP/IP (one server located in Baghdad).
B. **Client**: this represents a workstation with client-server applications running over TCP/IP and UDP/IP (the design used 68 clients).
C. **Router**: the main purpose for this component is the "routing", so any IP packets arriving on any interface are routed to the appropriate output

interface based on their destination IP address. The Routing Information Protocol (RIP) or the Open Shortest Path First (OSPF) protocol may be used to dynamically and automatically create the gateway's routing tables and select routes in an adaptive manner. The numbers of routers which are used are 4 routers, every one located in governorate.

D. **Switch**: The switch implements the Spanning Tree algorithm in order to ensure a loop free network topology. Packets are received and processed by the switch based on the current configuration of the spanning tree, each switch in the design connect several clients to the governorate's router.

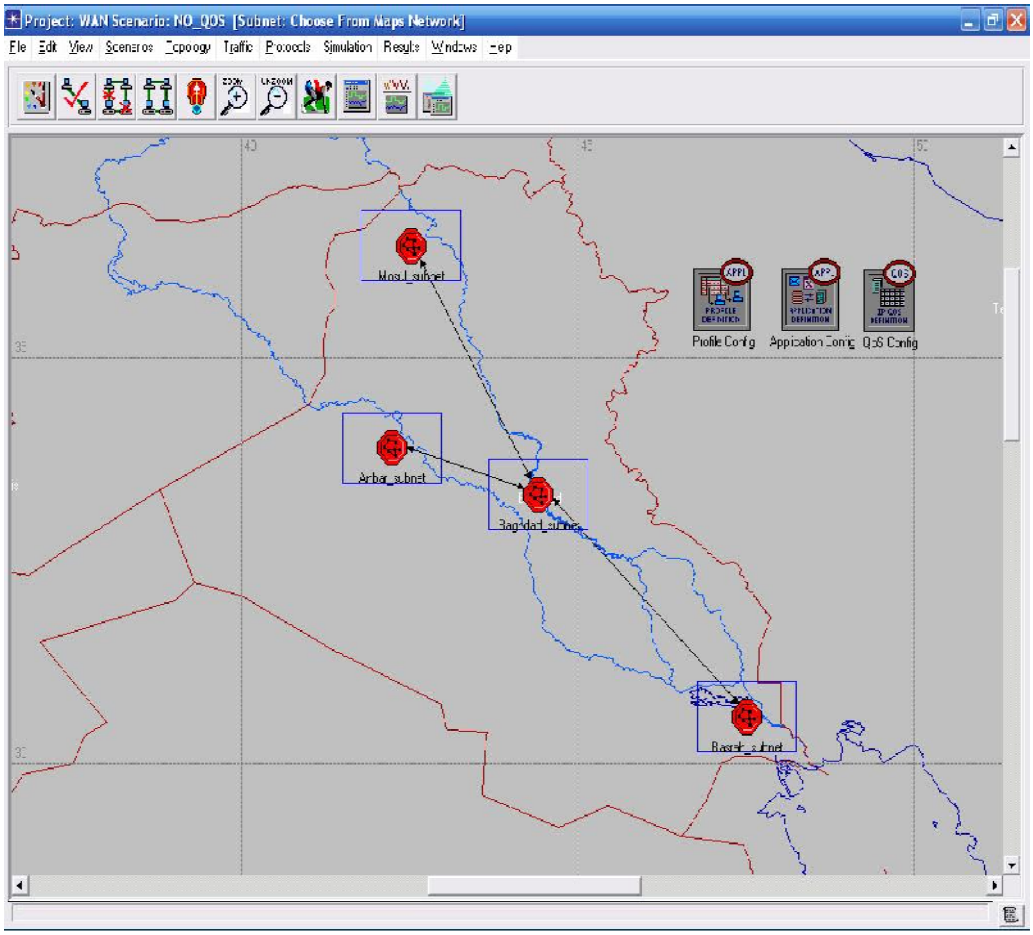

Figure (4.1): Main Network

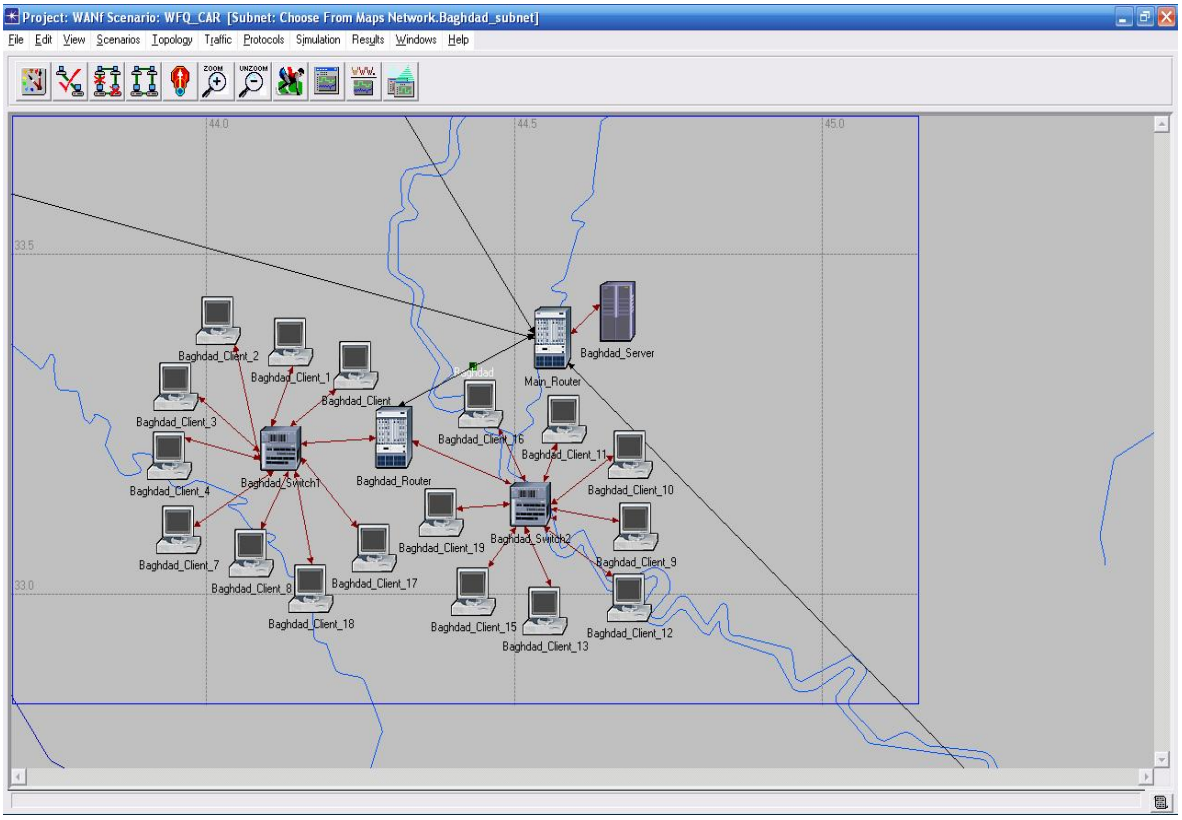

Figure (4.2-a): Baghdad Subnet

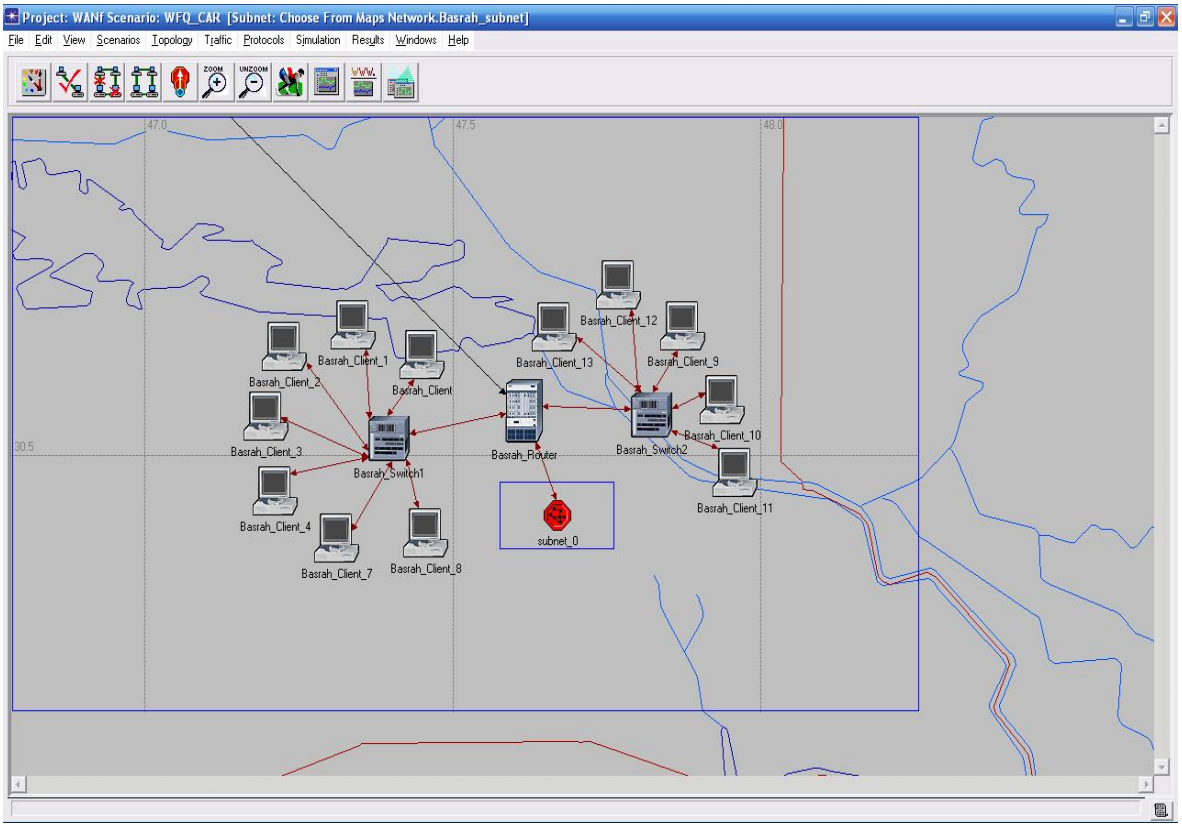

Figure (4.2-b): Basrah Subnet

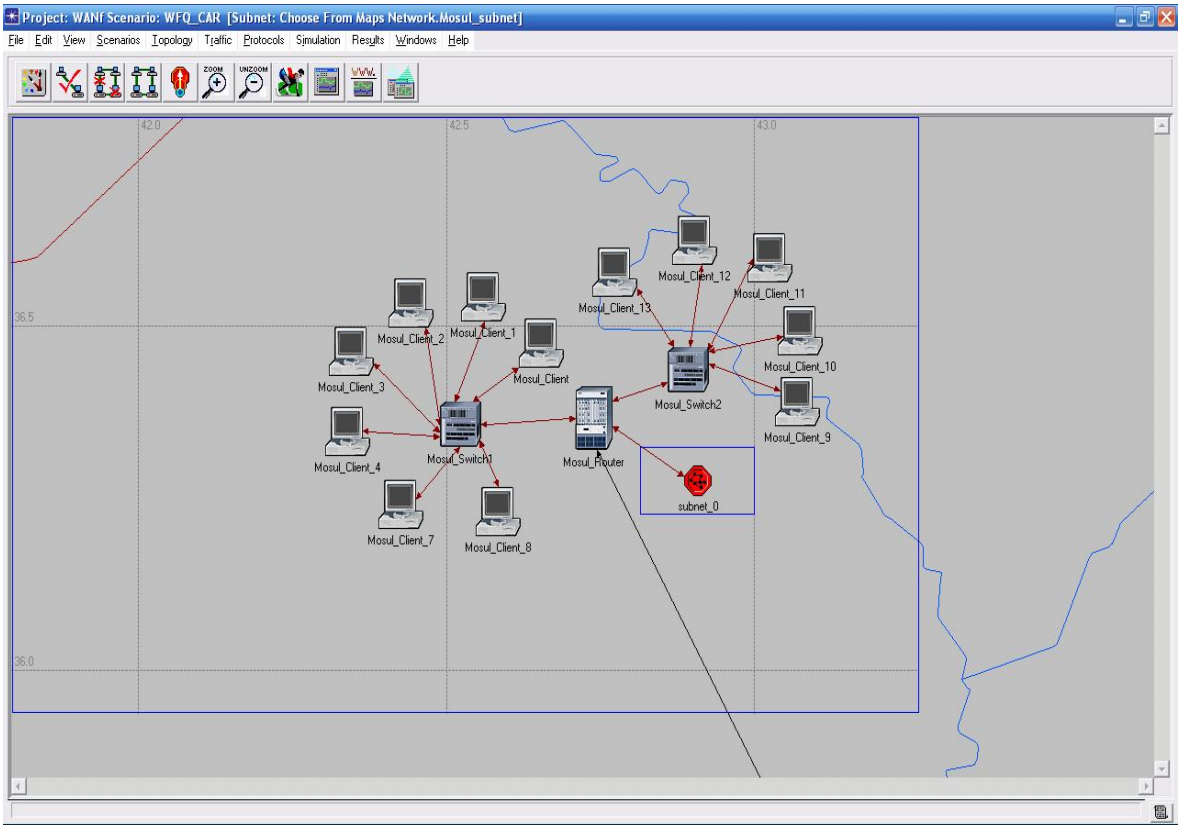

Figure (4.2-c): Mosul Subnet

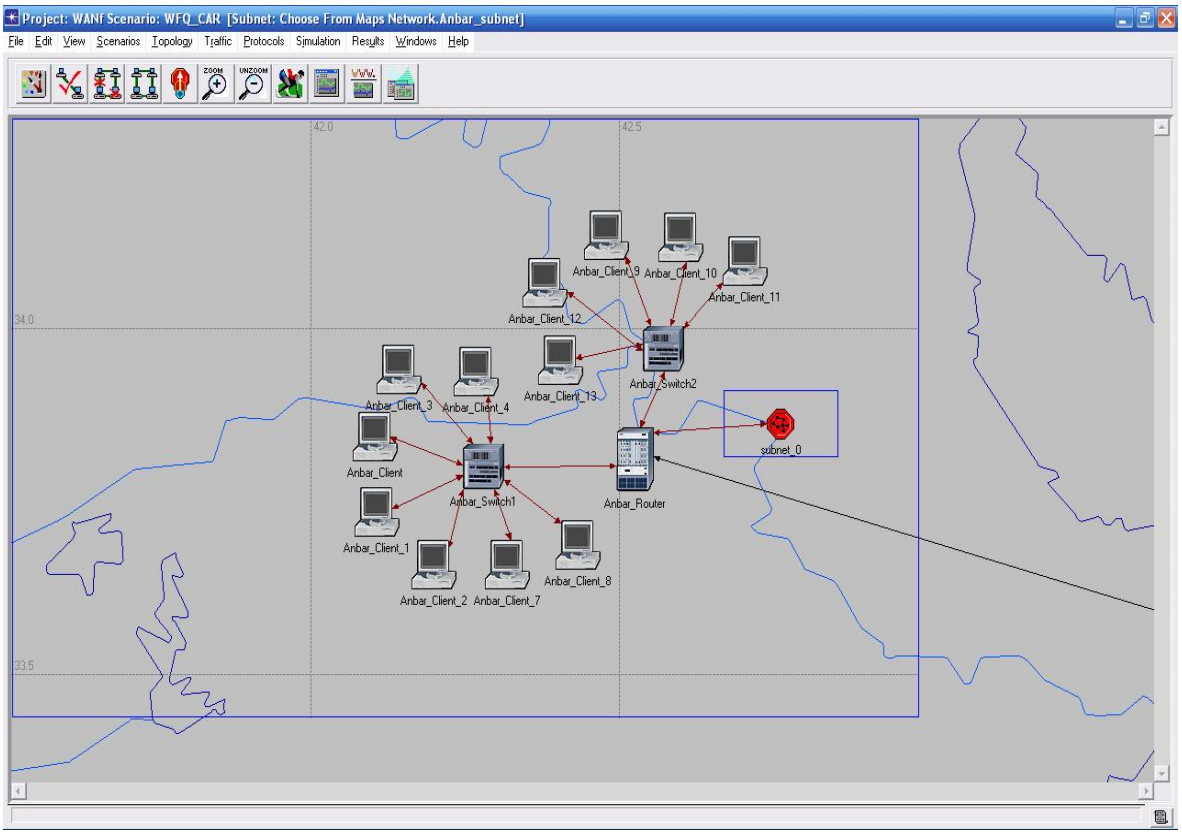

Figure (4.2-d): Anbar Subnet

The server for File Transfer and E-mail applications is located in Baghdad subnet. The main router which connects all the routers with each other is located in Baghdad subnet.

4.3 Simulation Sequences

The first approach consist of five scenarios with four applications, these scenarios will execute their method in all routers for the network, except the scenario one; while this scenario did not run any methods. The table (4.1) shows the list of scenarios and application for approach one.

Table (4.1): Simulation sequence for approach one

Application type	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
E-mail	No QoS	FIFO at All router	WFQ at all router	CQ at all router	PQ at all router
File Transfer	No QoS	FIFO at All router	WFQ at all router	CQ at all router	PQ at all router
Voice	No QoS	FIFO at All router	WFQ at all router	CQ at all router	PQ at all router
Video Conferencing	No QoS	FIFO at All router	WFQ at all router	CQ at all router	PQ at all router

For example, from the table above second scenario implements FIFO method on the all routers and run the applications (E-mail, file transfer, voice and video conferencing).

The first approach illustrates the benefit of using ToS (types of service) in the network, and studies the effect of using these mechanisms on some statistics such as end–to–end delay, delay variation, throughput and shows how QoS differentiates between the traffic based on their (ToS)

The second approach implements the CAR algorithm. According to the results founded by first approach, the second will implements with the appropriate queuing type.

4.4 Network Assumptions

Four (ToS) are implemented on the clients; the priority of these (ToS) are shown in Table (4.2).

Table (4.2): The priority of the type of service

Standard > Background
Excellent effort > Standard
Streaming multimedia > Excellent effort

These four priorities are distributed on the applications as shown in Figure (4.3). Each subnet clients has a combination of applications with their type of service, taking into consideration that each client shall used all the four applications. Section (A.3 Utilities) in Appendix A page A-3 describes who to make Application definition table. Also table (4.3) shows the most important parameters for the applications used in this design.

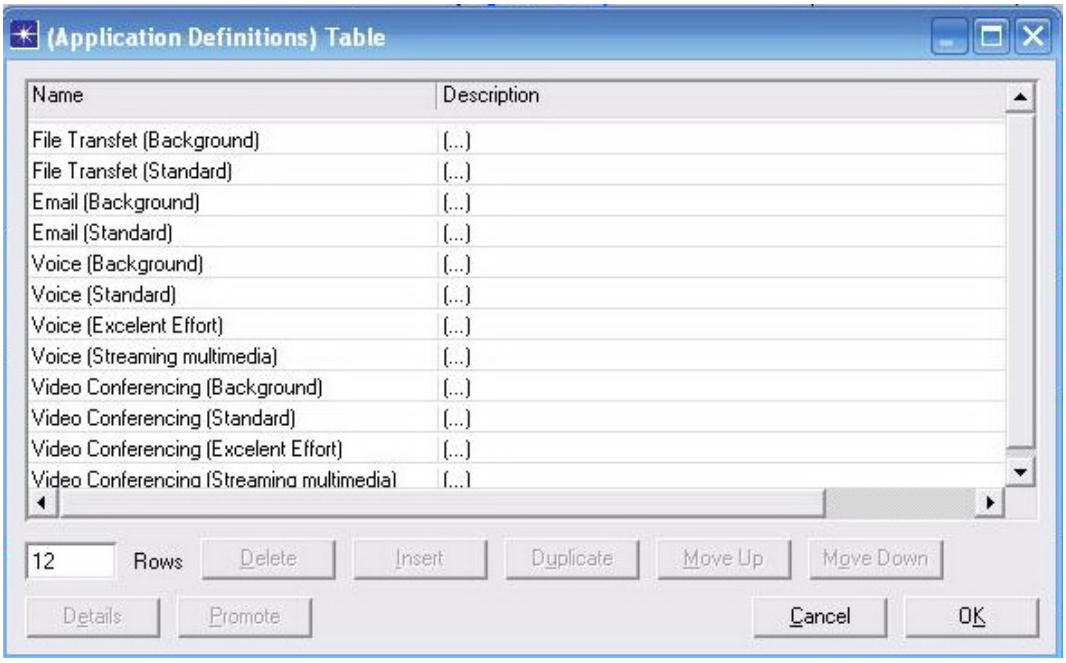

Figure (4.3): Applications with their type of service

Table (4.3): Applications parameters

Application	Parameter	Value or type
FTP	File size	20000 (byte)
Email	Email size	8000 (byte)
Voice	Voice Encoder Scheme	G.711
	Voice Frame per packet	7
Video Conferencing	Frame inter arrival time information	30 frames/second
	Frame size information	30000 (byte)

The links between the governorates (routers) is PPP_E3 which have throughput equal to 32 MB, while the links between the clients and switch inside the governorates is 100BaseT which have throughput equal to 100 MB.

The governorates have their weights when they communicate with each other or with the server. In case of video conferencing and voice, Baghdad

connections have the higher weights among the others. Table (4.4) shows the weight that is given to each connection between the governorates.

Table (4.4): Connection's weights between the governorates

	Baghdad	**Basrah**	**Mosul**	**Anbar**
Baghdad		50	30	20
Basrah	50		30	20
Mosul	50	30		20
Anbar	50	30	20	

To understanding the meaning of these weights, figure (4.4) shows an example for three packets with different weights, on the outgoing interface the packet between Baghdad and Basrah served first because it's have the higher weight comparing with the remaining.

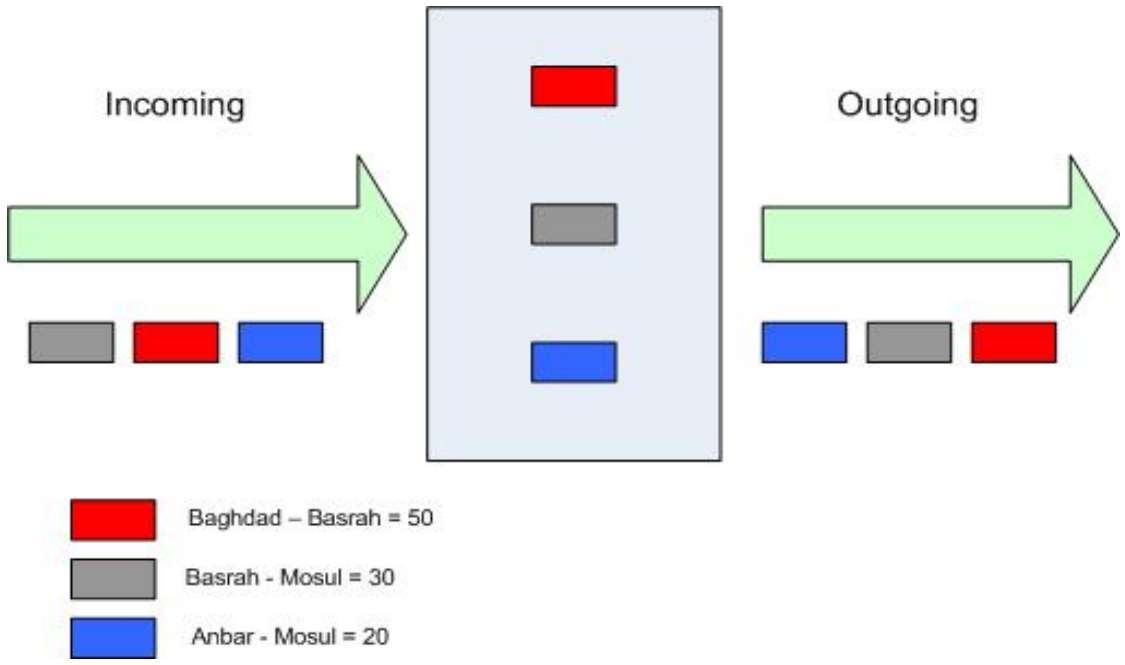

Figure (4.4): An example shows who packet weight works

On the other hand, for File transfer and E-mail applications where the connection is between the client and the server; the weight for Baghdad, Basrah, Mosul and Anbar clients are 10, 8, 6 and 4 respectively to the server.

Clients in the network get the benefit of many known services. Profiles in OPNET identify the clients specific work behavior. Four profiles are programmed according to their locations as shown in Figure (4.5). Section (A.3 Utilities) in Appendix A page A-5 describes who to make Profile configuration table.

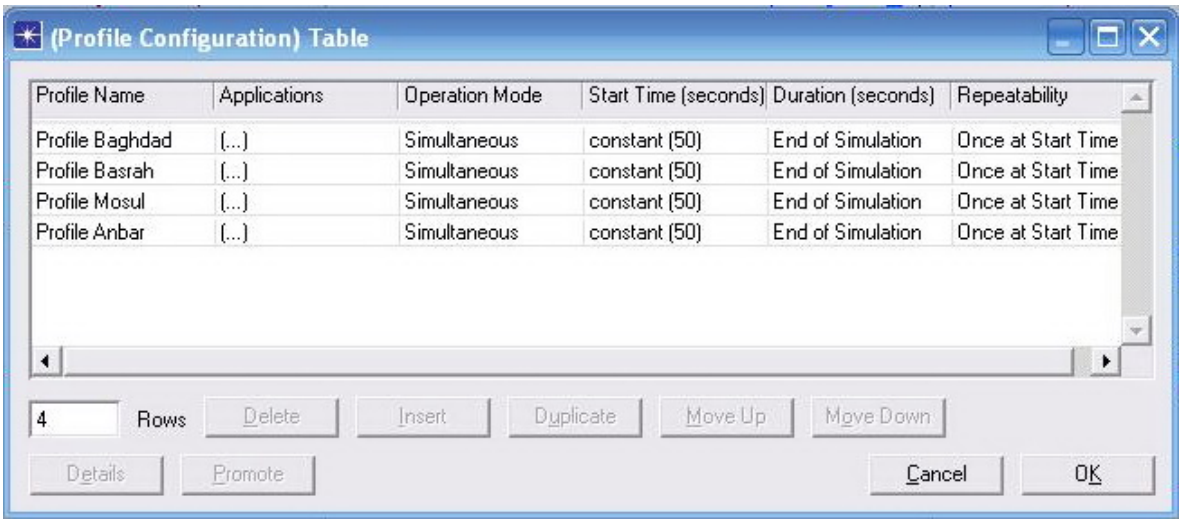

Figure (4.5): Profiles for the governorates

4.5 Network Simulation and Evaluations

1. Approach One

As shown in Table (4.1), approach one implements four types of scheduling with four types of applications. It is better to compare these types of scheduling for a single application to show their effects clearly.

- **E-mail Traffic**

The first application is the E-mail which is applied to five scenarios as mentioned in Table (4.1) and the results are collected and a comparison is then made among them.

The first statistic to deal with is the traffic sent (bytes) as shown in Figure (4.6), where the E-mail traffic sent is the average number of bytes per second submitted to the transport layers by all E-mail applications in the network.

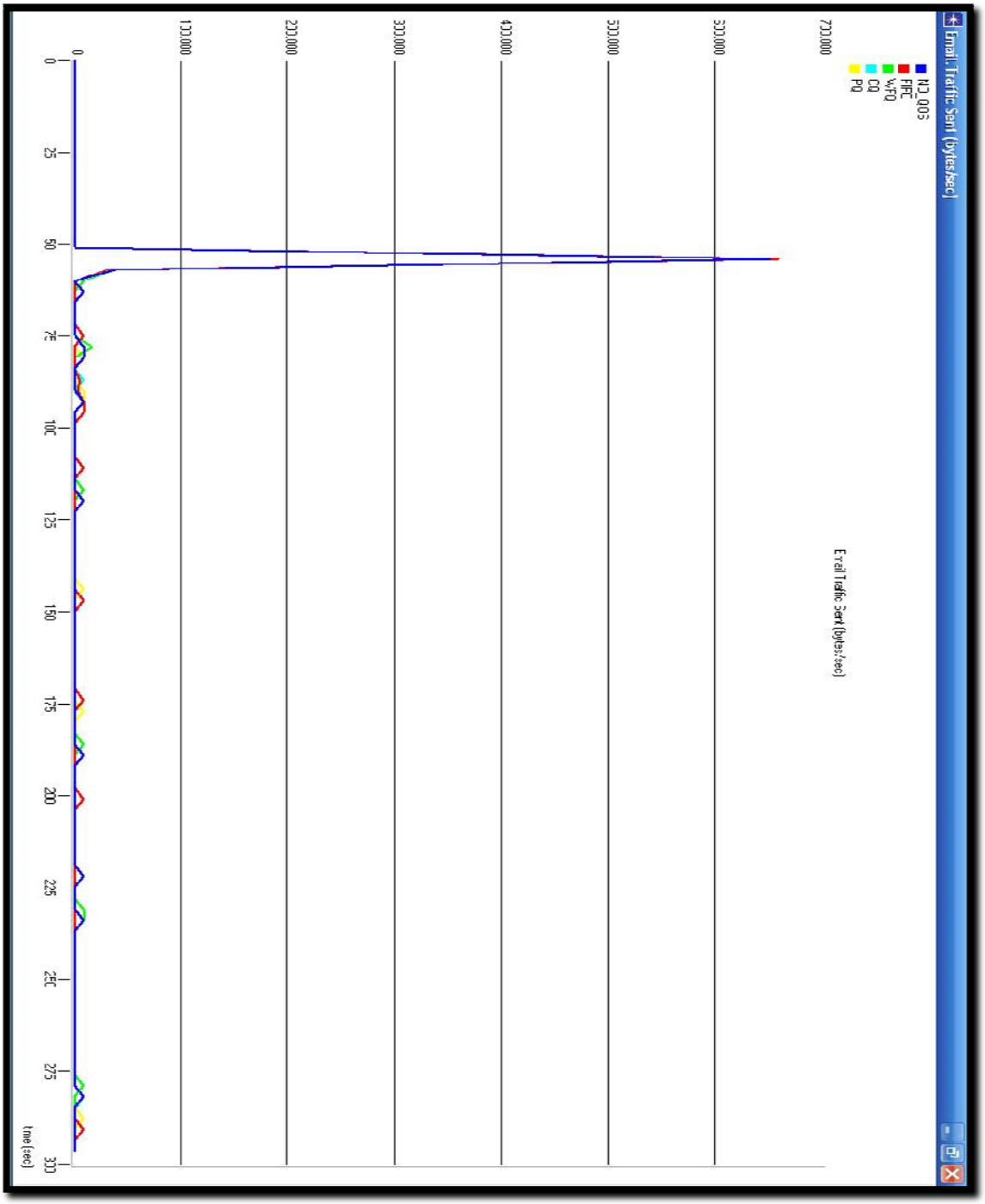

Figure (4.6): Approach One E-mail traffic sent (bytes/second)

The second statistic covers the E-mail traffic received (bytes/second). The result is shown in Figure (4.7). The E-mail traffic received is the average number of bytes per second forwarded to all E-mail applications by the transport layers in the network. Because the traffic type is light (E-mail), the

results seem to be the same, although there are a very minor differences between them, but from these two figures there are differences between traffic sent and received which is due to packets loss, this losses happened because of the priority of this application which is less than the priority of the other applications (like voice and video).

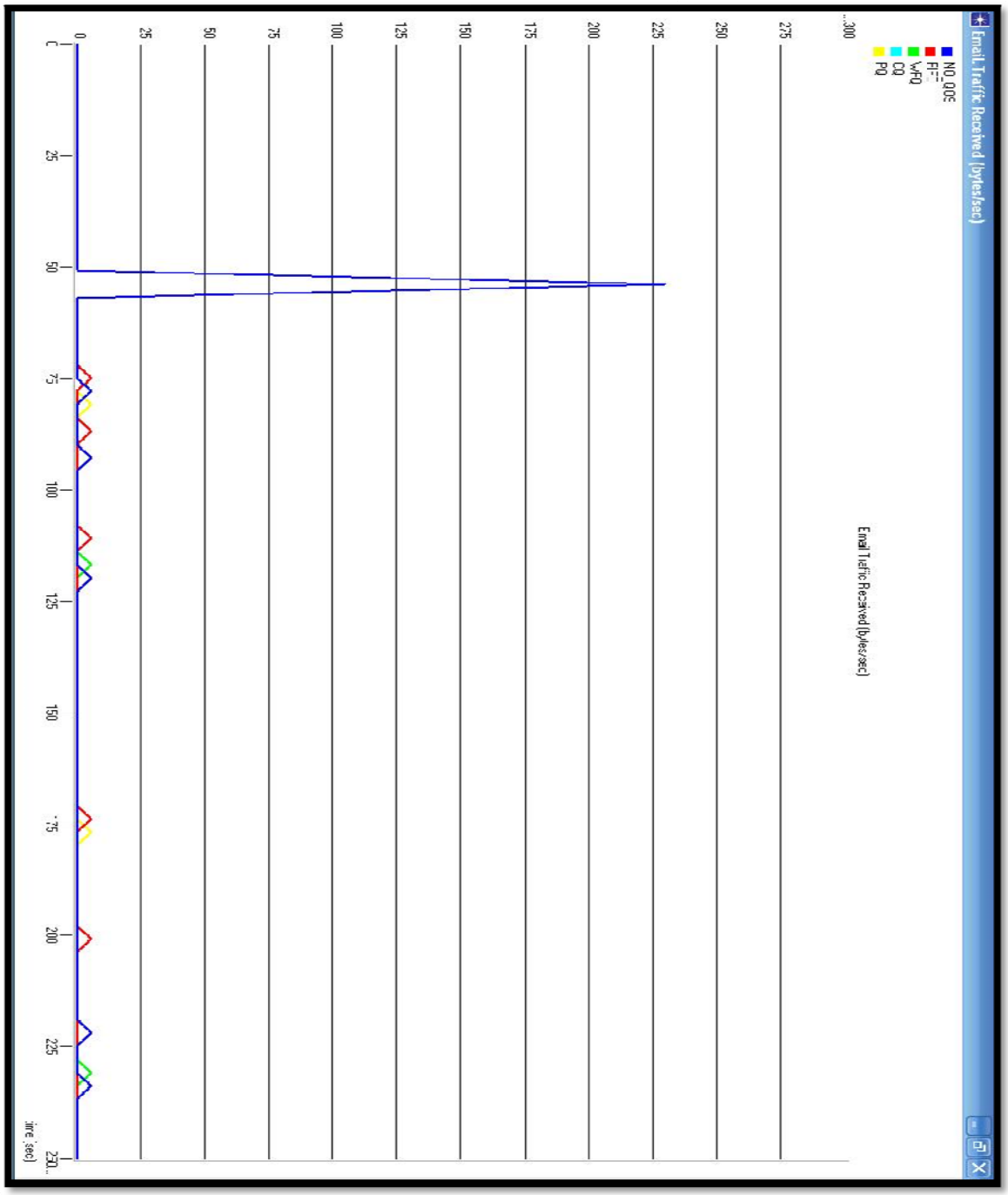

Figure (4.7): Approach One E-mail traffic received (bytes/second)

- **File Transfer**

The second application is the File Transfer which is applied to five scenarios as mentioned in Table (4.1). The results are collected and a comparison is made among them. The first statistic to deals with is the traffic sent as shown in Figure (4.8), which is the average bytes per second submitted to the transport layers by all FTP applications in the network.

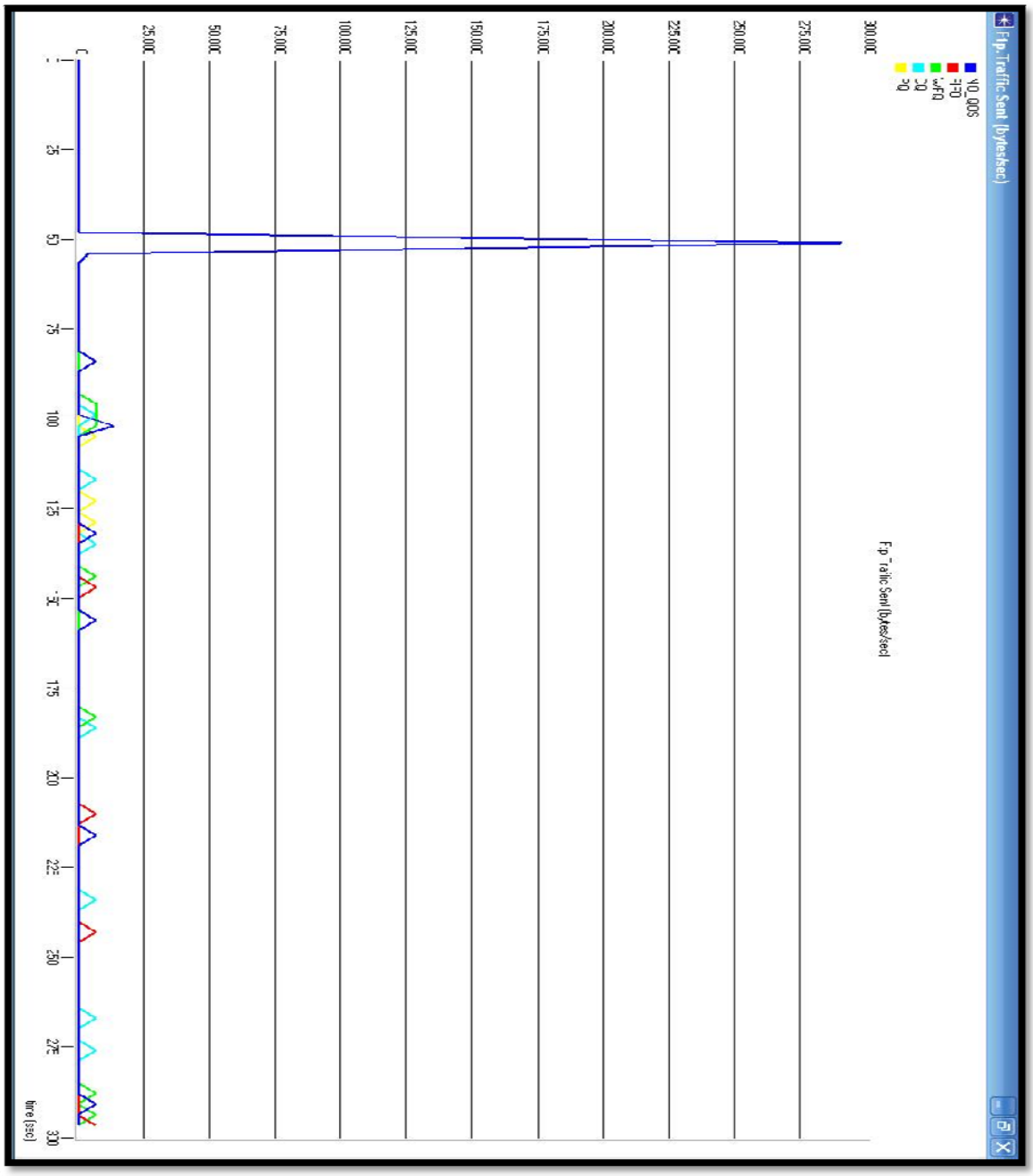

Figure (4.8): Approach One FTP traffic sent (bytes/second)

The second statistic to deal with is the FTP traffic received. The result is shown in Figure (4.9). The FTP traffic received is the average bytes per second forwarded to all FTP applications by the transport layers in the network. As in the Email traffic, the FTP traffic received is different from traffic sent but with less packet drop than the Email, this happened because the weight priority connection between the clients and the server download or upload files is more than the weight priority for Email connections between clients and server.

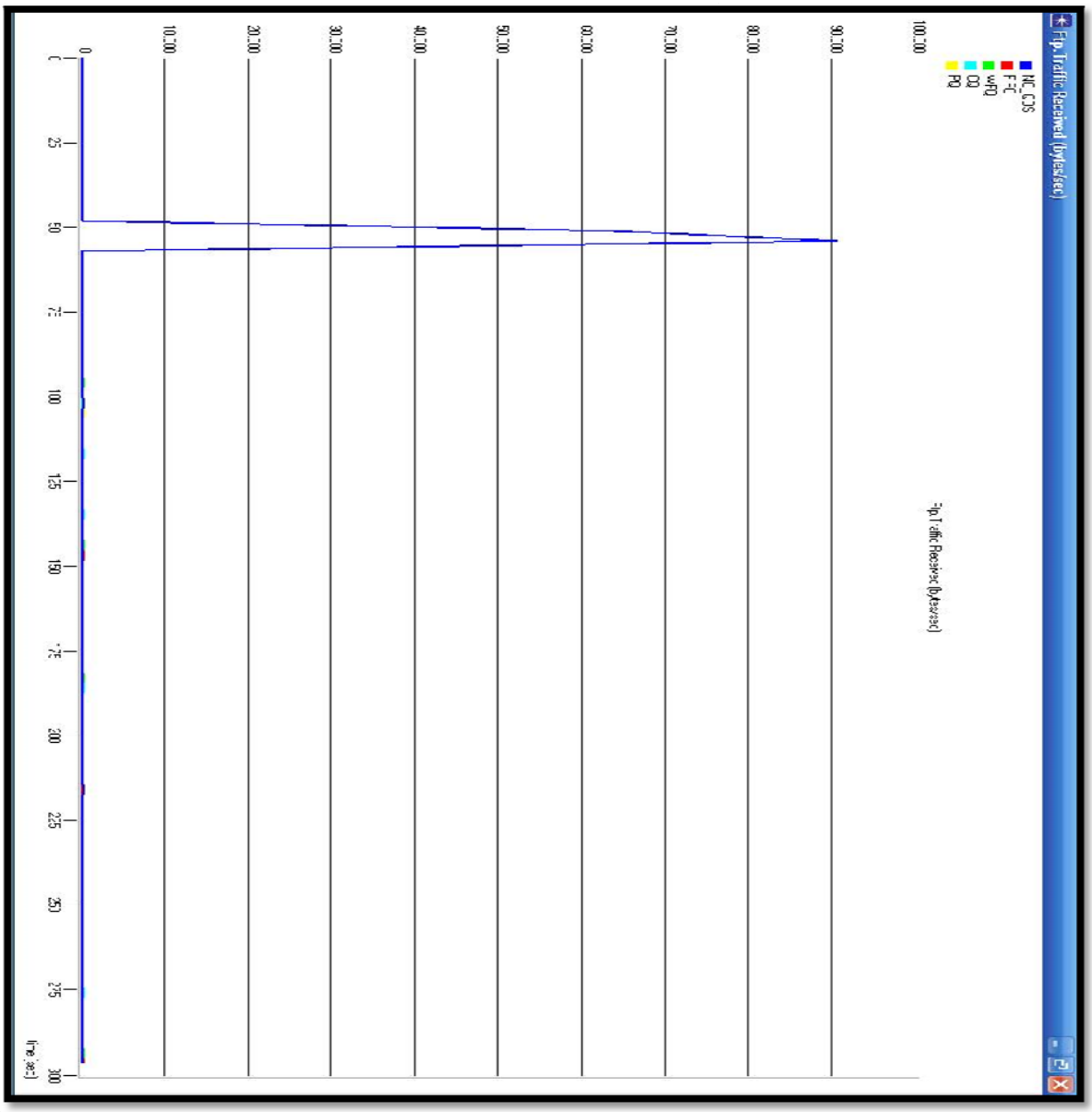

Figure (4.9): Approach One FTP traffic received (bytes/second)

- **Voice Traffic**

The third application is the voice. It is also applied to five scenarios, with four statistics. The first statistic is the traffic sent, which is the average number of bytes per second submitted to the transport layers by all voice applications in the network. The result is shown in Figure (4.10).

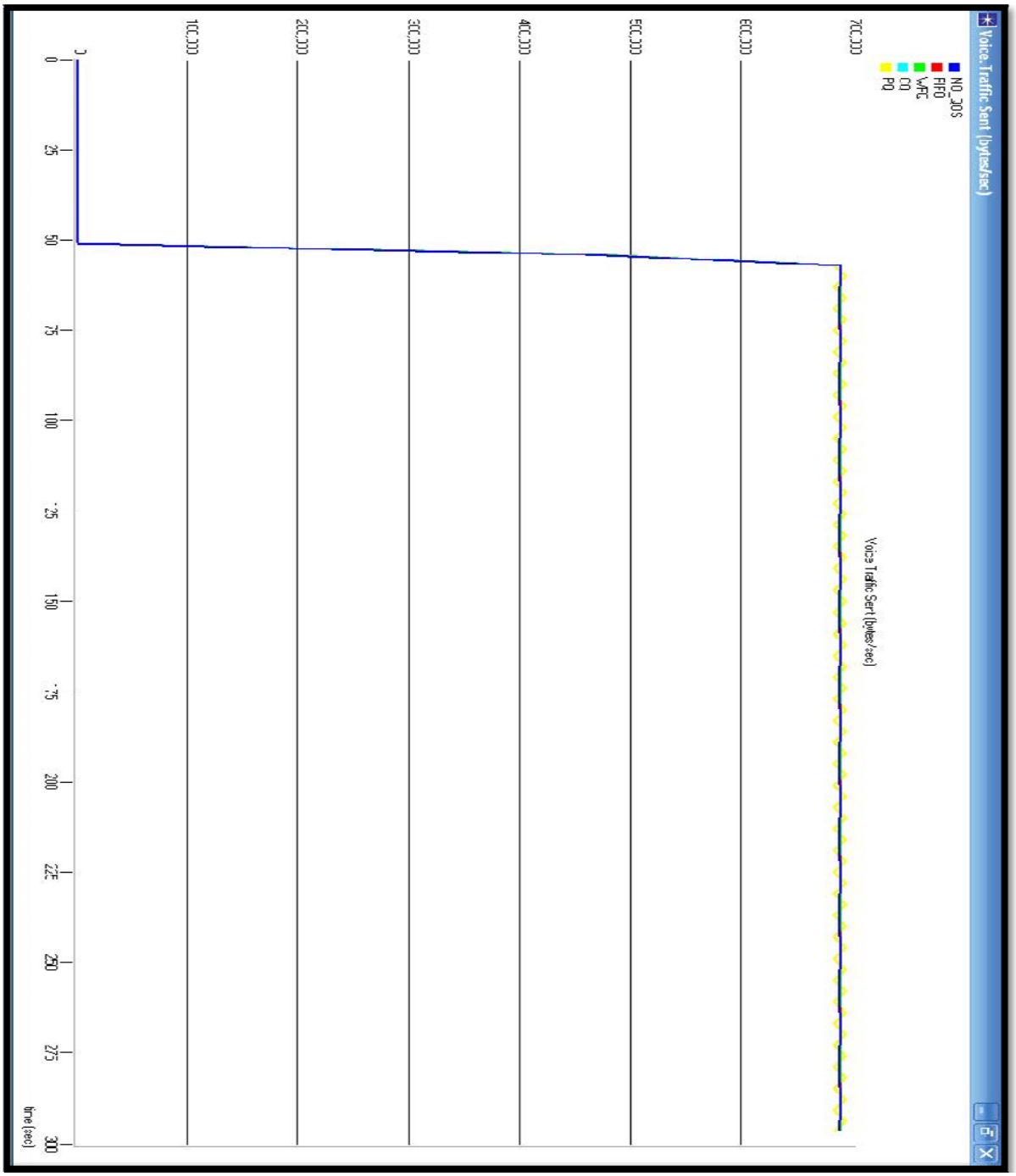

Figure (4.10): Approach One Voice traffic sent (bytes/second)

The second statistic is the traffic received, which is the average number of bytes per second forwarded to all voice applications by the transport layers in the network. Figure (4.11) shows the voice traffic received.

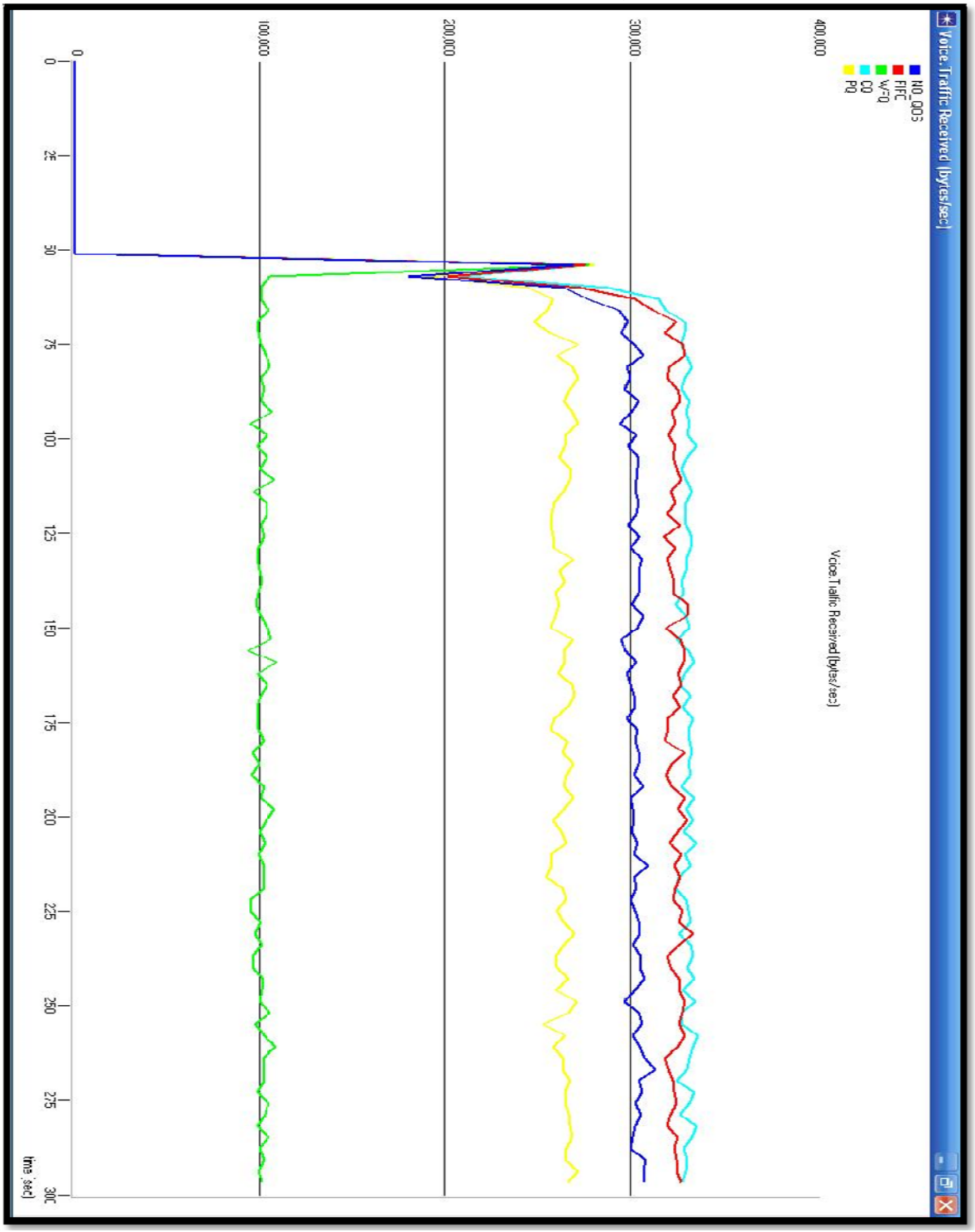

Figure (4.11): Approach One Voice traffic received (bytes/second)

Figures (4.9) and (4.10) show that there are no different in voice traffic sent, but the differences are exists in voice traffic received which is due to congestion. Also figure (4.10) shows the different actions for each type of queuing methods related to the way which deal with it. The third and fourth statistics deals with delay concepts, which are packet end-to-end delay and packet delay variation respectively. Figure (4.12) and (4.13) shows voice packet end-to-end delay and voice packet delay variation respectively.

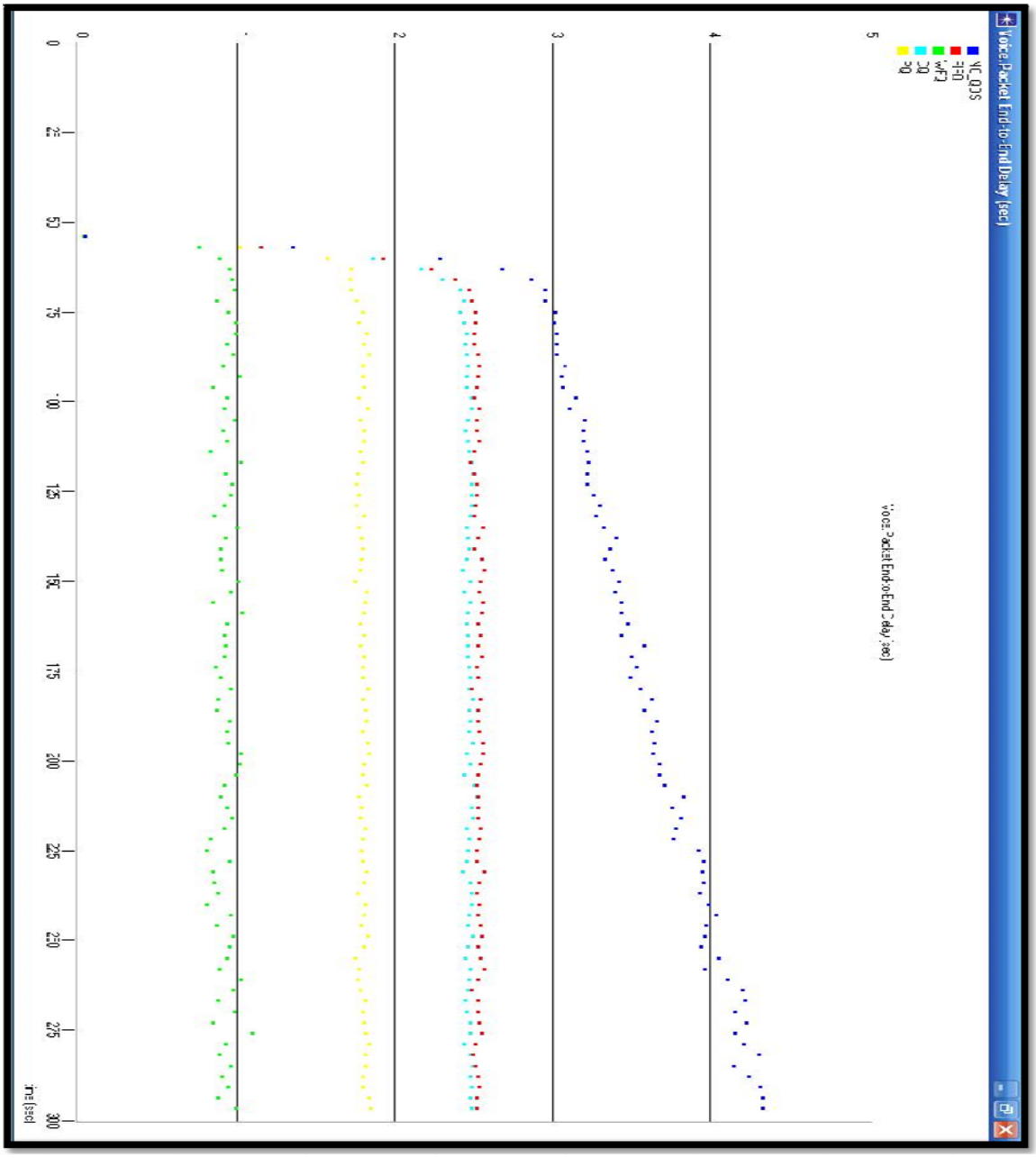

Figure (4.12): Approach One Voice packet end-to-end delay (second)

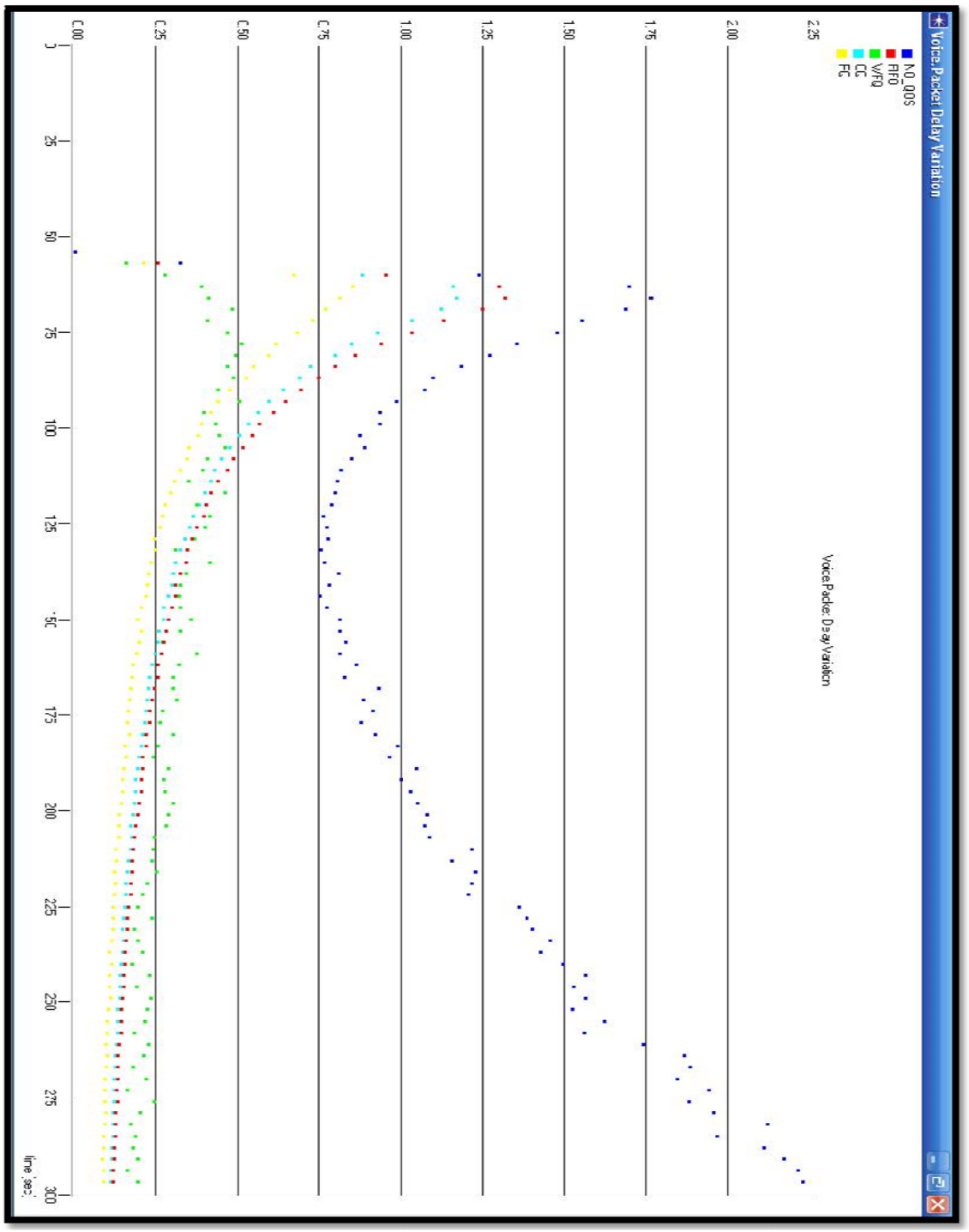

Figure (4.13): Approach One Voice packet delay variation

Figure (4.12) shows that packet delay increased quickly in first scenario because it does not implement any type of queuing and the routers buffer

increase which this leads to packets delay. Figure (4.13) shows the delay variation which is a very important parameter in communication. Here, the variation is large when no quality of service is applied, while other scenarios produced excellent variation because the delay is approximately constant.

- **Video Conferencing Traffic**

The forth application is video conferencing which is also applied to all the scenarios like the other applications. Three statistics were measured to show the effects of QoS to this application. The first statistic covers the video traffic sent which is the average number of bytes per second submitted to the transport layers by all video conferencing applications in the network.

The second statistic shows the video traffic received which is the average number of bytes per second is forwarded to all video conferencing applications by the transport layers in the network. The third statistic covers the video packet end-to-end delay, which is the time taken to send a video application packet to a destination node application layer. This statistic records data from all the nodes in the network. Figures (4.14), (4.15) and (4.16) show these statistics respectively.

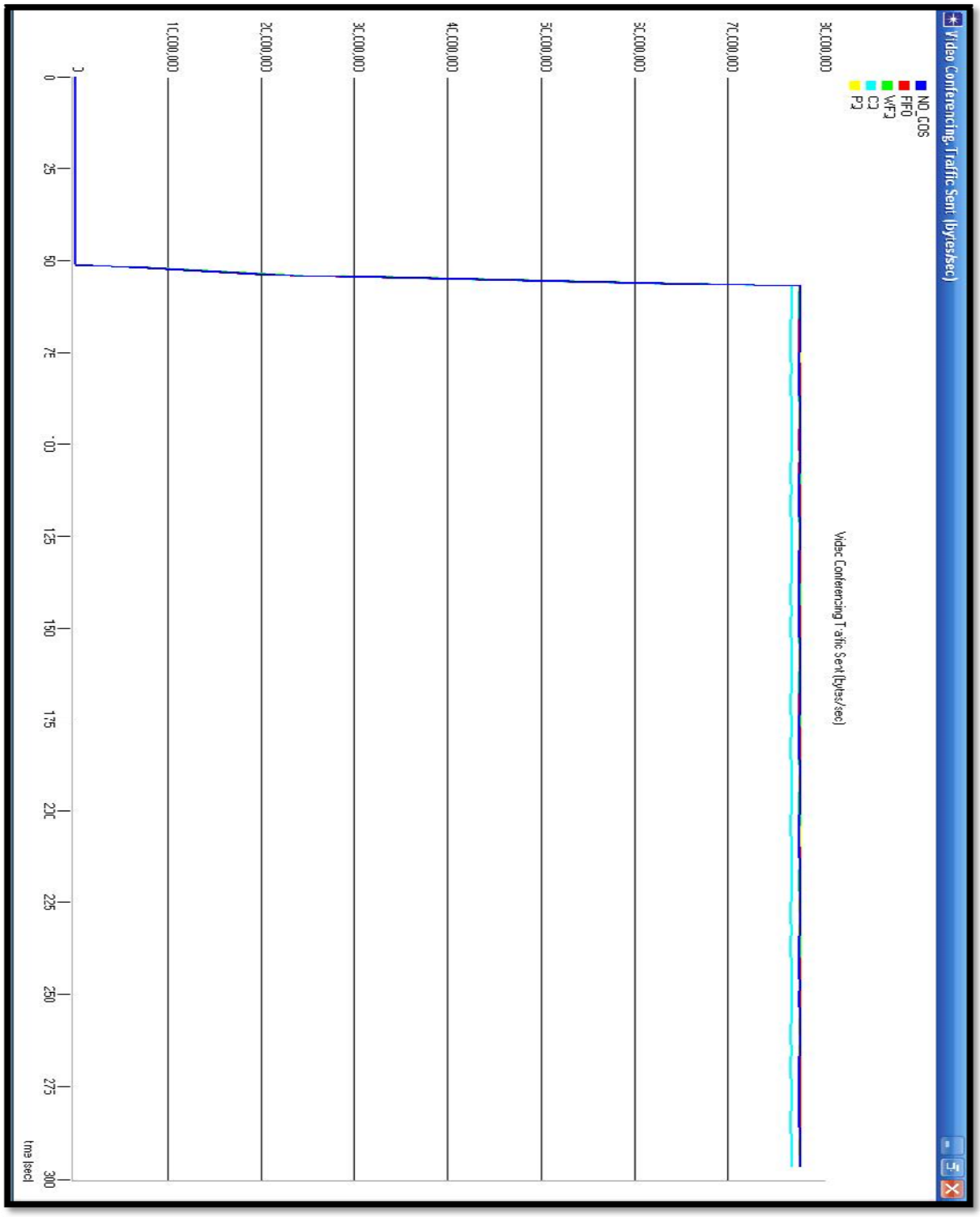

Figure (4.14): Approach One Video Conferencing traffic sent (bytes/second)

From figure above, traffic sent is the same for all scenarios which means that all send the same traffic (number of bytes).

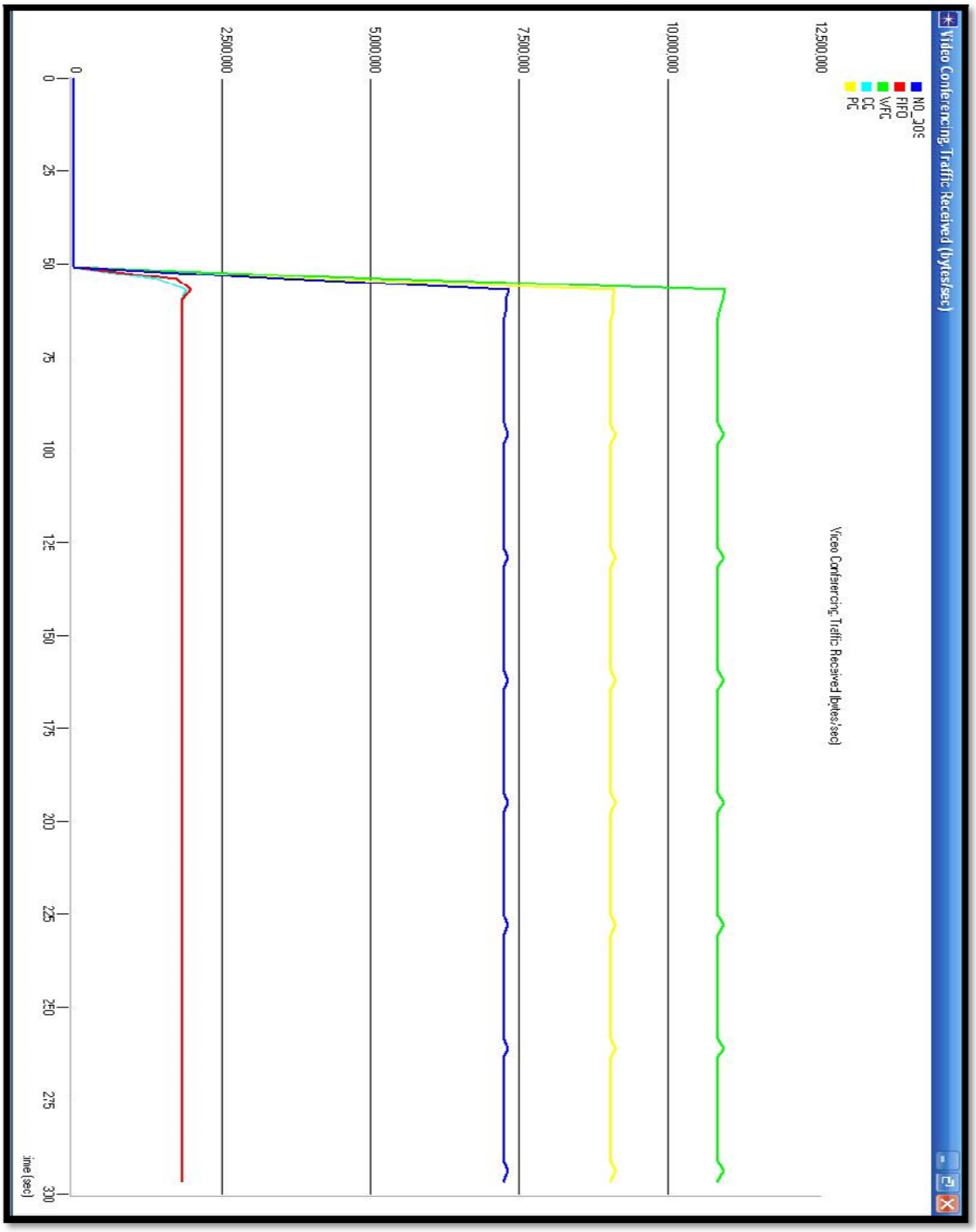

Figure (4.15): Approach One Video Conferencing traffic received (bytes/second)

Figure (4.15) show the different in video traffic received for all scenarios which is due to the way each type used to arrange packets in cause

of congestion. The differences between traffic sent and received represent the packets drop.

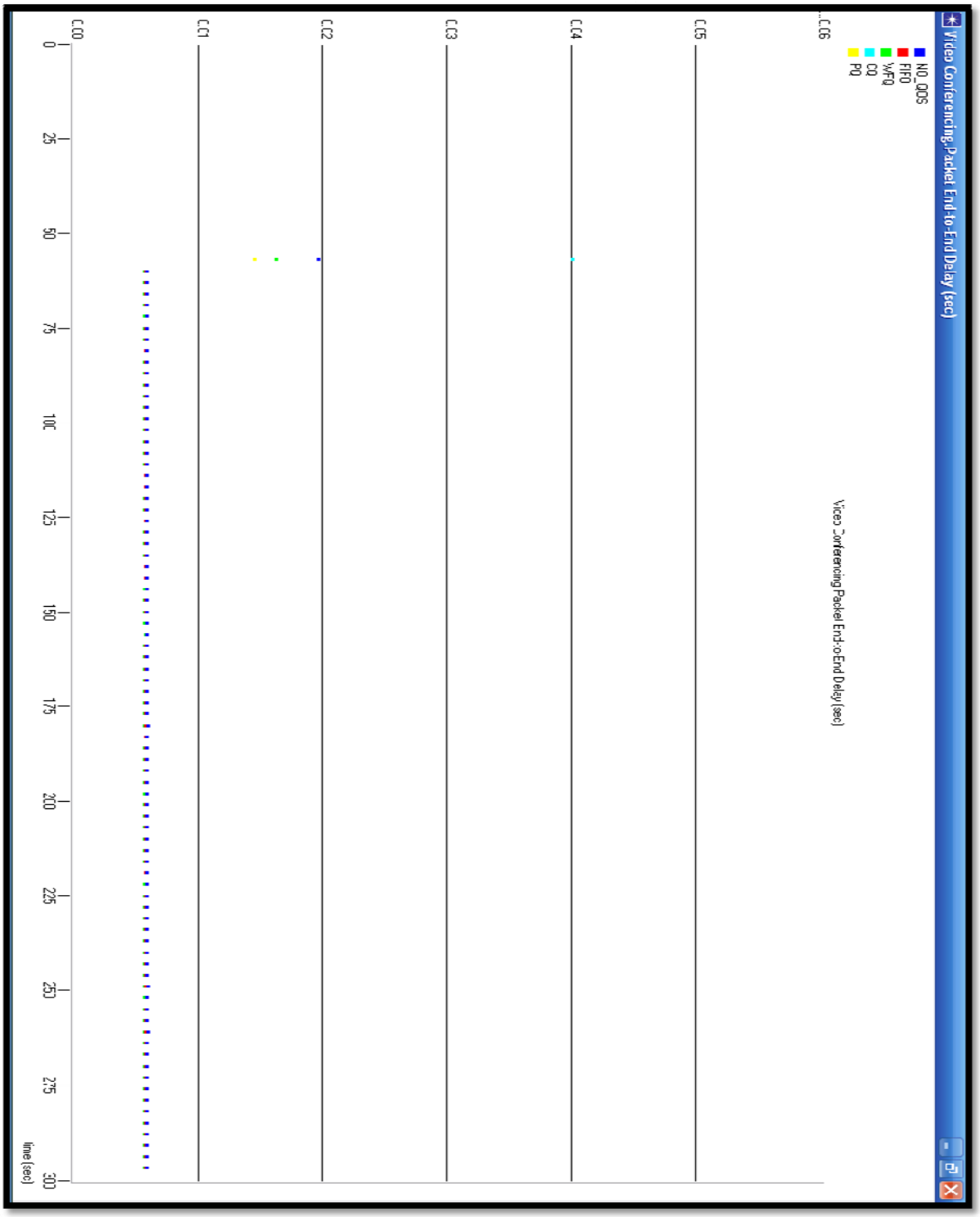

Figure (4.16): Approach One Video Conferencing packet end-to-end delay (second)

According to figure (4.16) results, video end to end delay is acceptable for all scenarios (less than 0.1) which related to the higher priority design for video conferencing, so the video packet will not stay long time in the router buffer.

- **IP Traffic Dropped**

When quality of service is implemented, traffic at the router increase. When queuing and buffer size at the router reaches the maximum amount, packets will be dropped. The IP traffic dropped is the number of IP datagram dropped by all nodes in the network across all IP interface. The amount of these dropped packets has an effect on the network efficiency. Figure (4.17) shows the IP traffic dropped of the whole system.

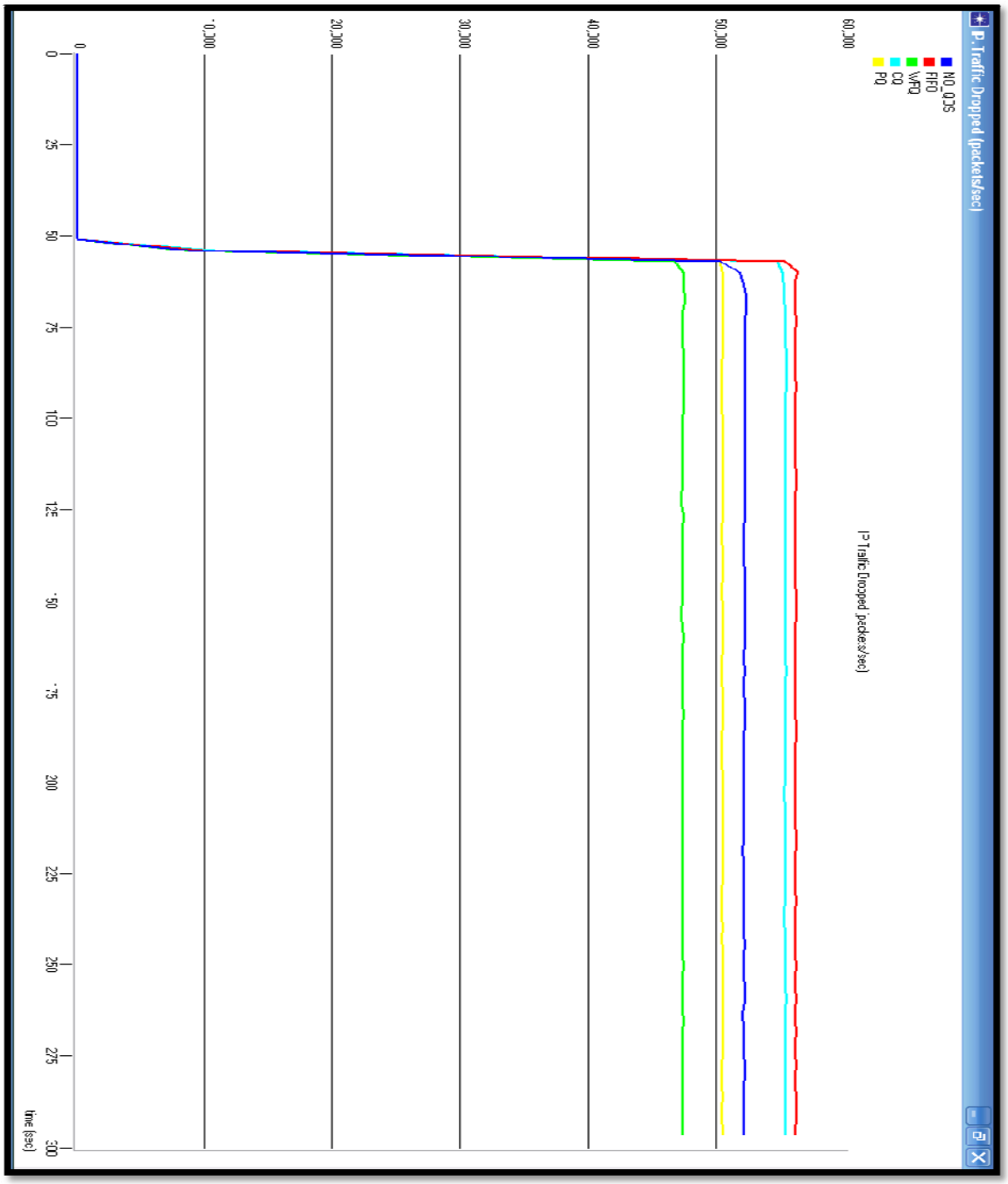

Figure (4.17): Approach One Global IP traffic dropped (packets/second)

From figure above, WFQ gave minimum packet dropped comparing with the others type of queuing, which is related to the design and the fair way that the WFQ used.

2. Approach Two

From results obtained from approach one, weight fair queuing is the best queuing disciplines comparing with the others by giving accepted delay and packets dropped, so approach two will implement WFQ with CAR algorithm in order to show the effect of it. The same is implemented here as approach one.

- **E-mail Traffic**

  Figure (4.18) shows a comparison between WFQ and WFQ with CAR for E-mail traffic sent, while Figure (4.19) shows a comparison between WFQ and WFQ (CAR) for E-mail traffic received. The two figures shows that there are little differences between them. This is due to the light traffic for E-mail traffic.

  From these two figures, WFQ with CAR can access bandwidth management through rate limiting which is clear in figure (4.19).

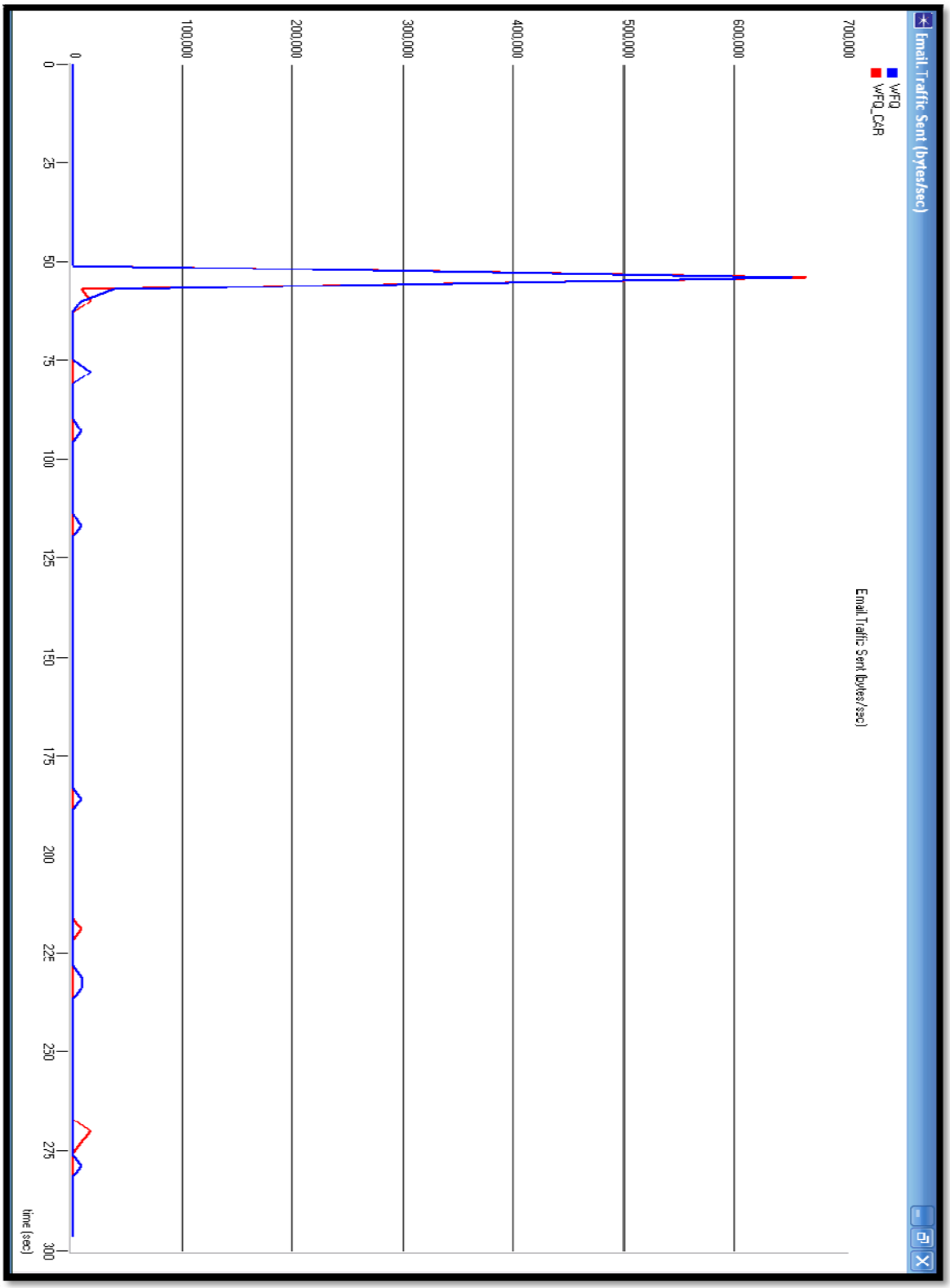

Figure (4.18): WFQ and WFQ (CAR) E-mail traffic sent (bytes/second)

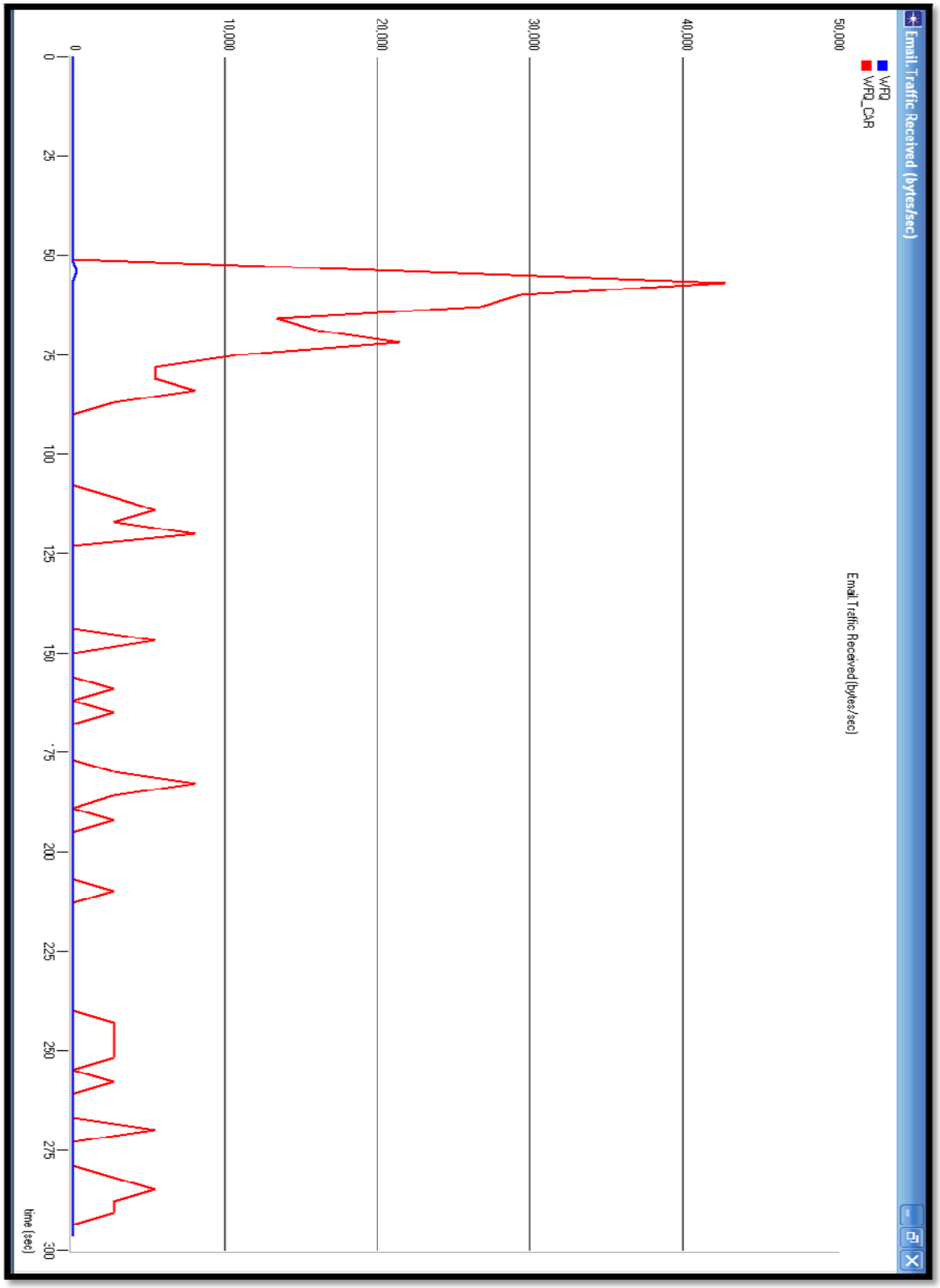

Figure (4.19): WFQ and WFQ (CAR) E-mail traffic received (bytes/second)

- **File Transfer Traffic**

Figures (4.20) and Figure (4.21) show a comparison between CAR and WFQ for FTP traffic sent and FTP traffic received respectively.

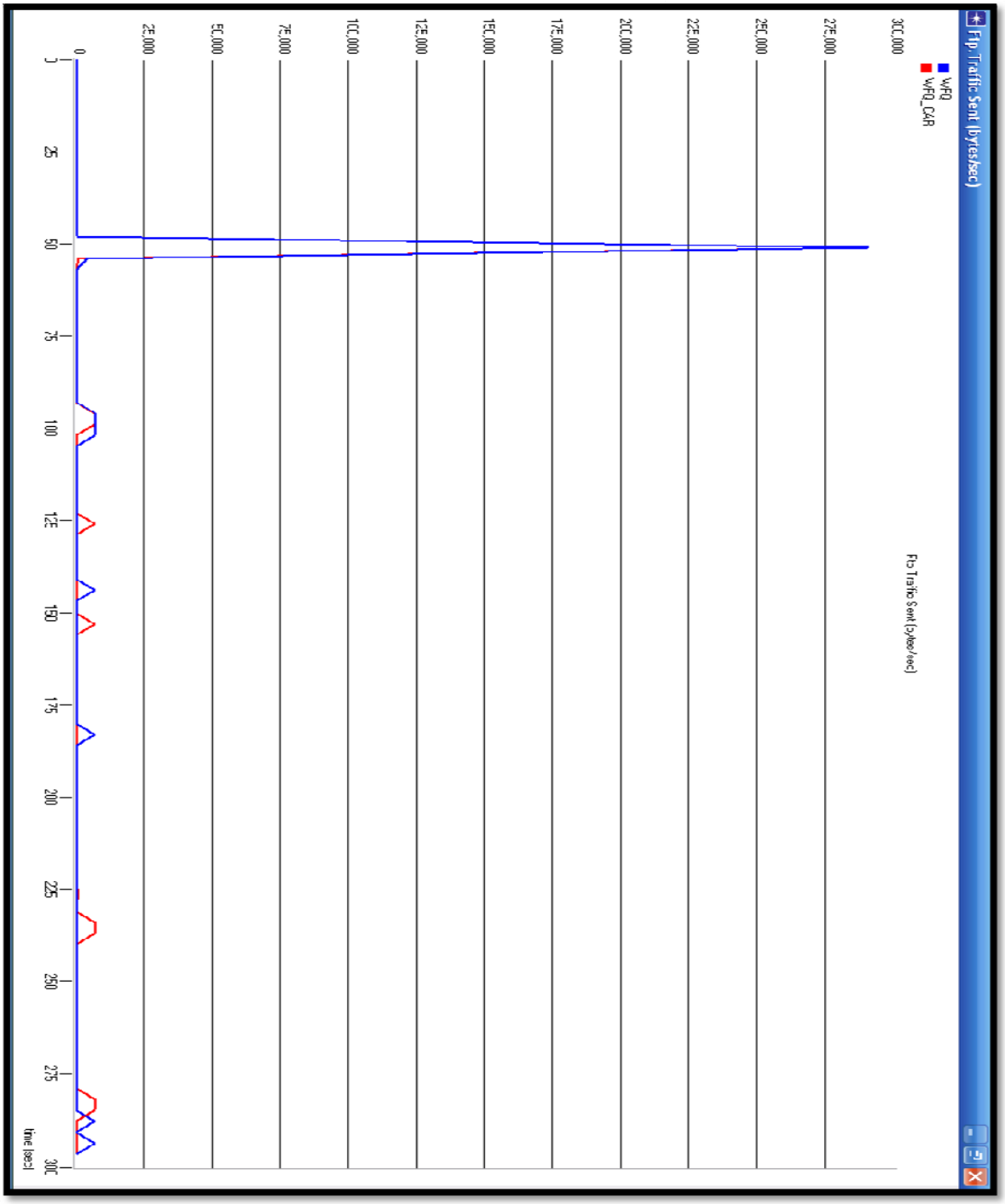

Figure (4.20): WFQ and WFQ (CAR) FTP traffic sent (bytes/second)

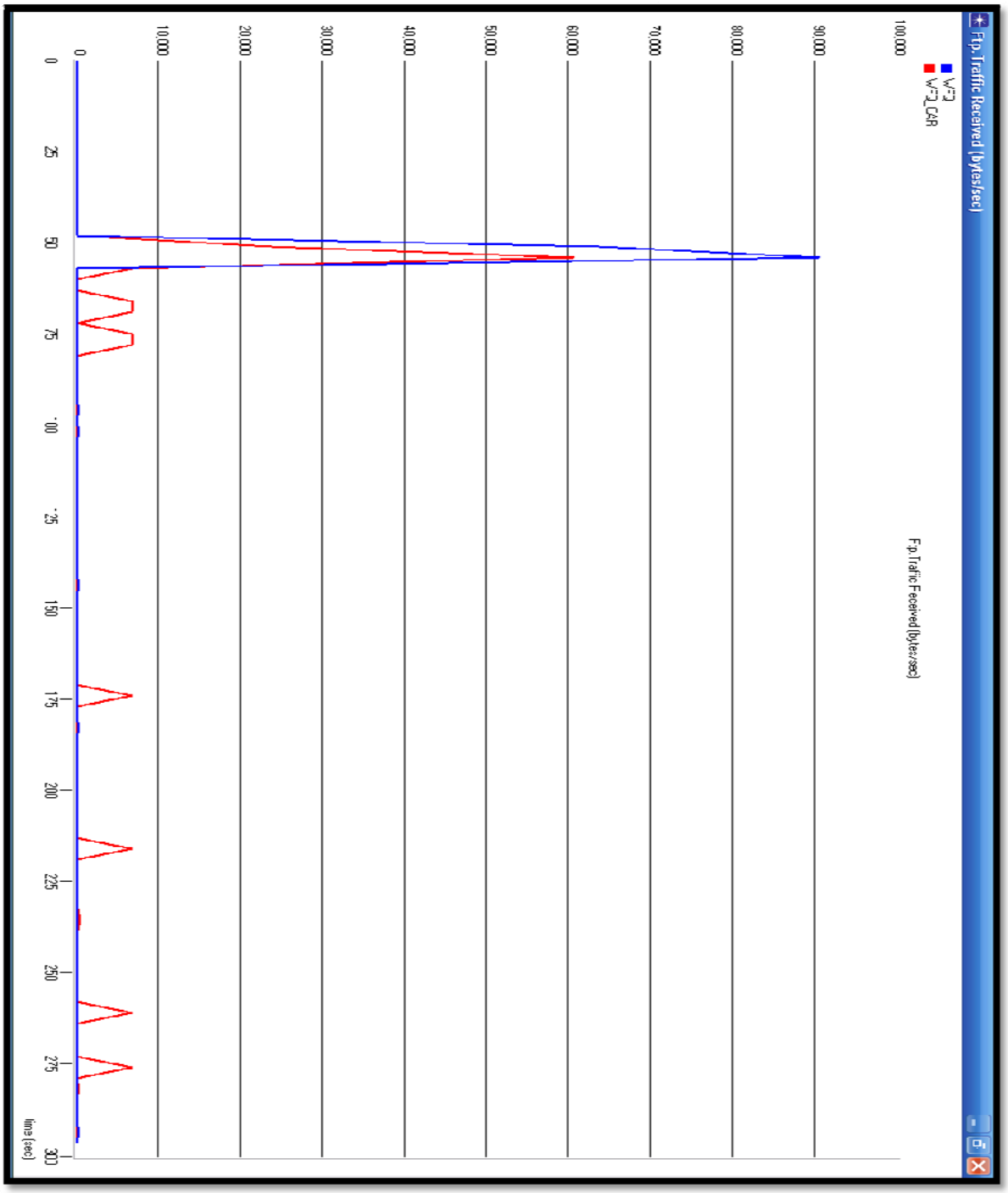

Figure (4.21): WFQ and WFQ (CAR) FTP traffic received (bytes/second)

Like the E-mail traffic, CAR file transfer traffic is almost the same as the WFQ, also there is no difference between traffic sent, while differences appear in traffic received which is the same reasons for Email traffic.

- **Voice Traffic**

Like approach one, approach two voice traffic contain four statistics, Figure (4.22) and (4.23) show a comparision between WFQ and WFQ with CAR Voice traffic sent and Voice trrafic received respectivily. These figures show that there are no difference in case of traffic sent but it will exist in case of traffic recrived related to the traffic policing which CAR algorithm used.

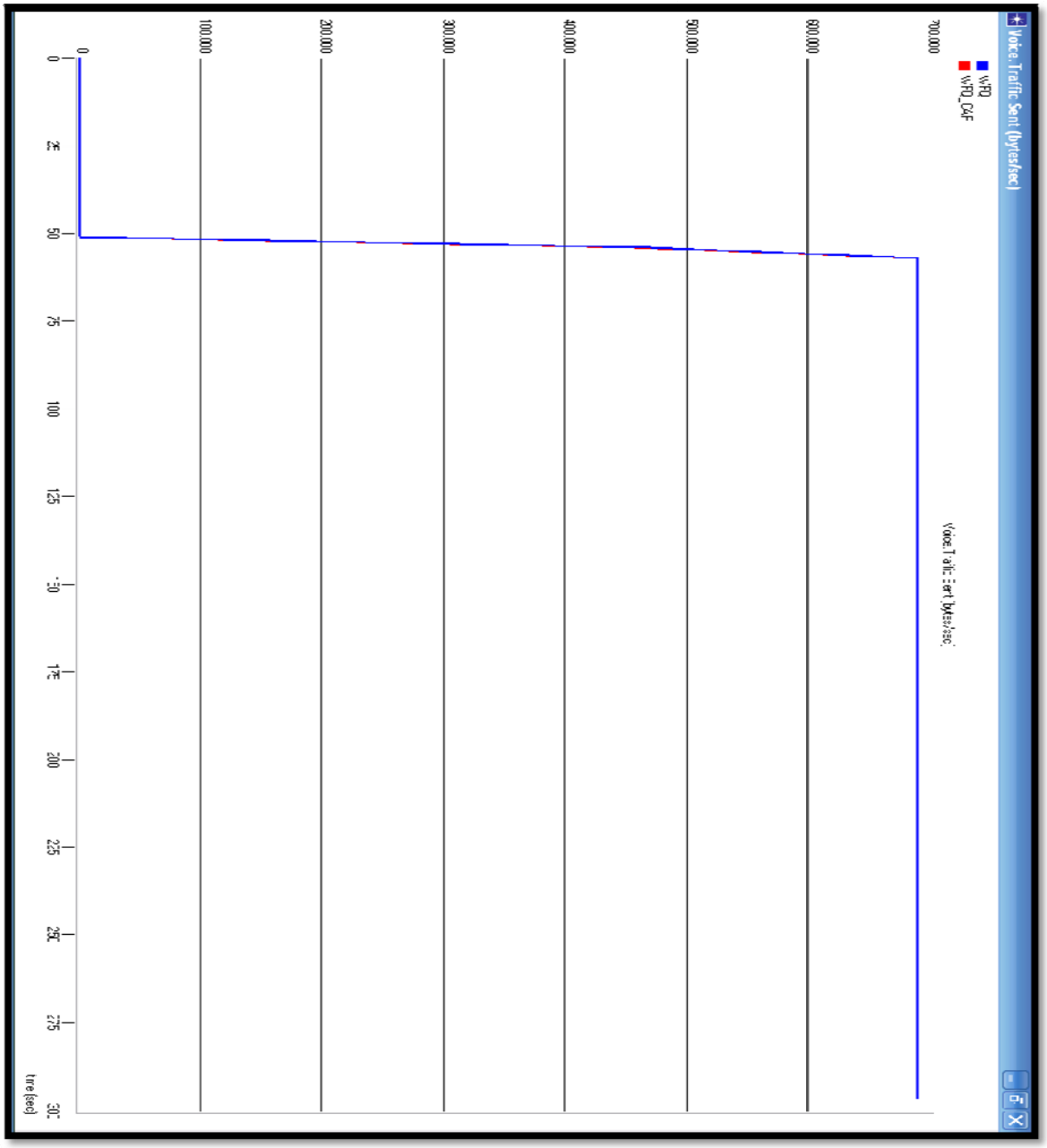

Figure (4.22): WFQ and WFQ (CAR) Voice traffic sent (bytes/second)

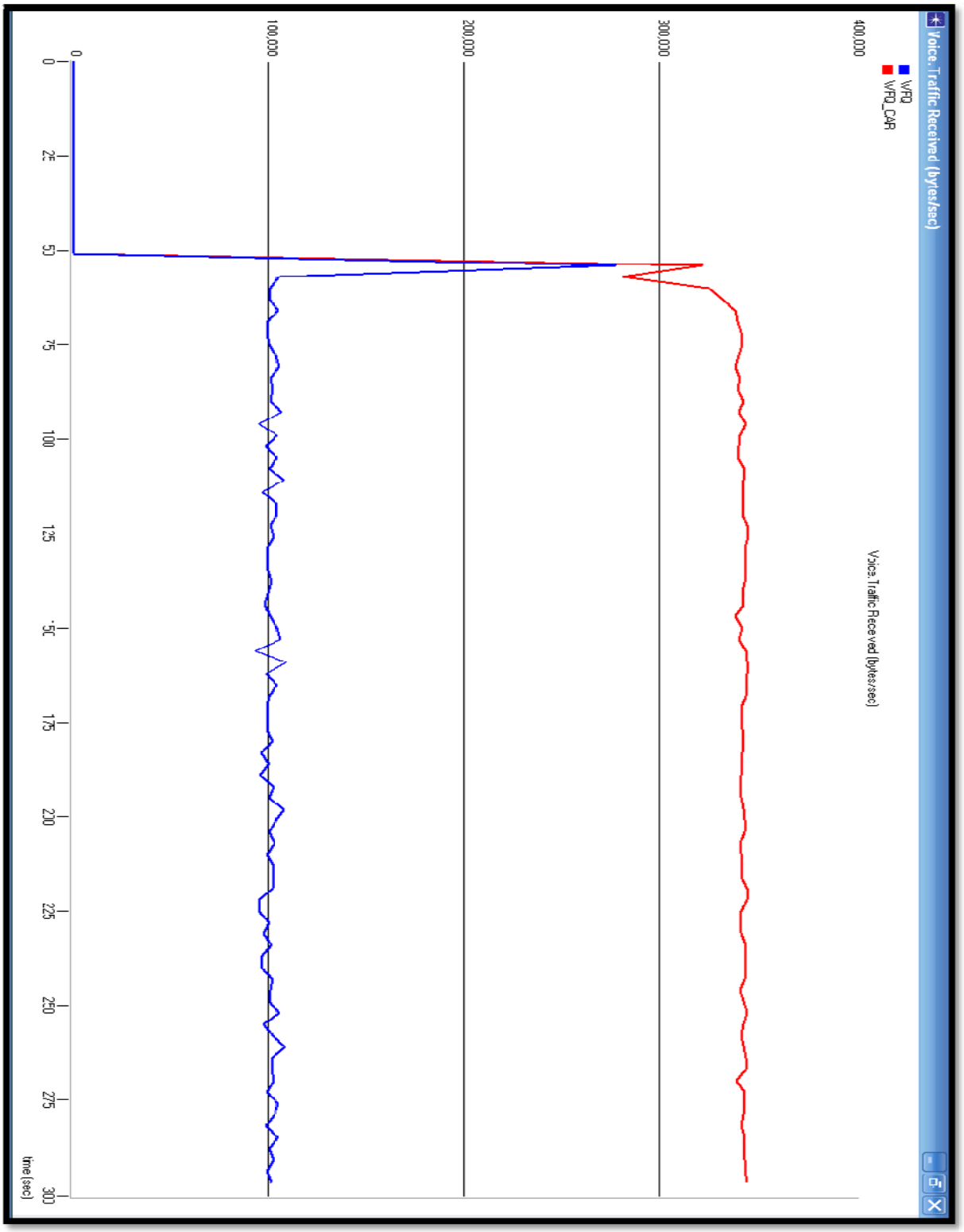

Figure (4.23): WFQ and WFQ (CAR) Voice traffic received (bytes/second)

Figure (4.24) shows the voice end to end delay for WFQ (CAR) and WFQ, from this figure one can illustrates that CAR algorithm gives a negligible packets delay related to the traffic limiting.

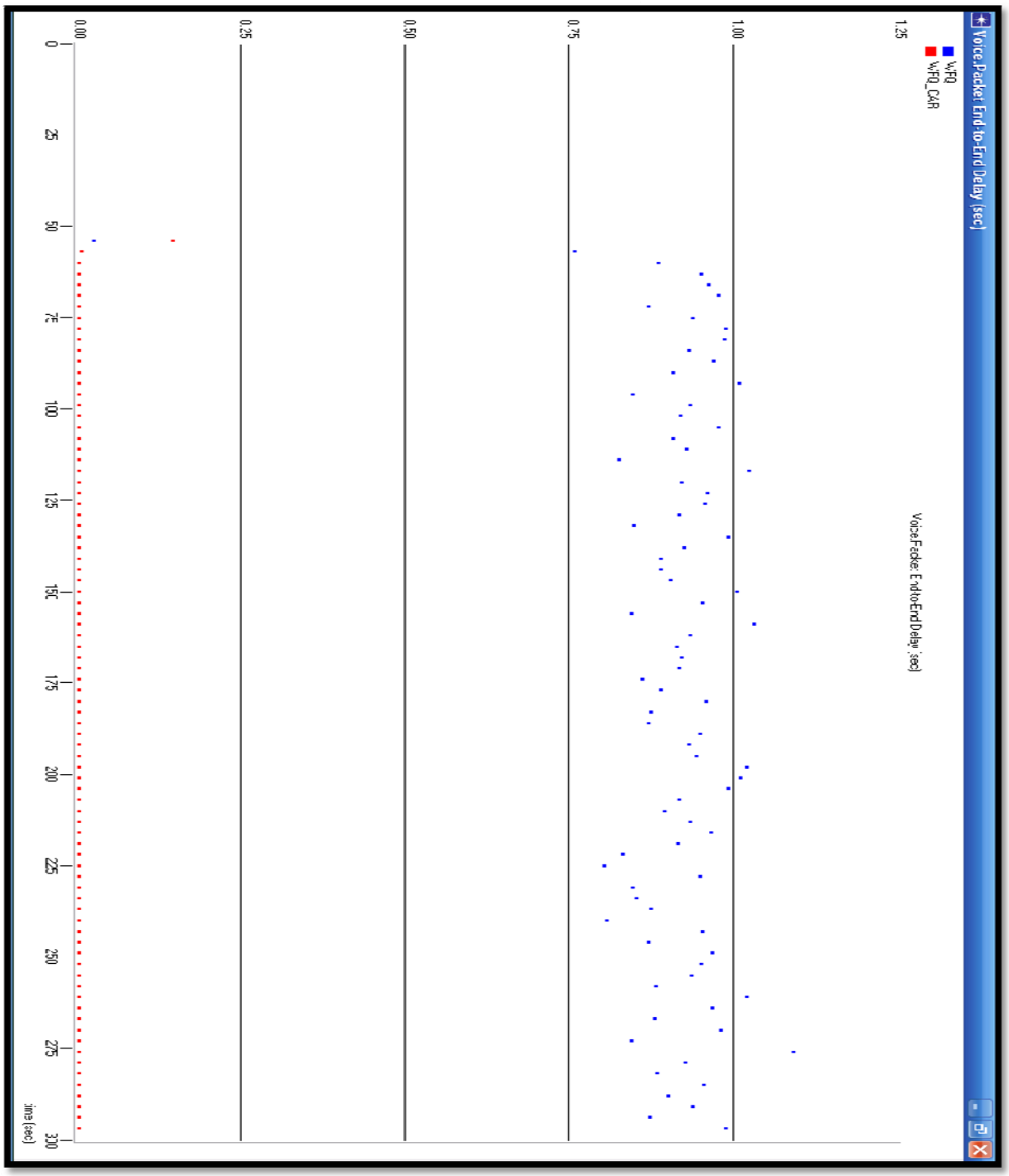

Figure (4.24): WFQ and WFQ (CAR) Voice packet end-to-end delay (second)

Figure (4.25) shows the voice packet delay variation for WFQ (CAR) and WFQ which is negligible related to the small variation in delay.

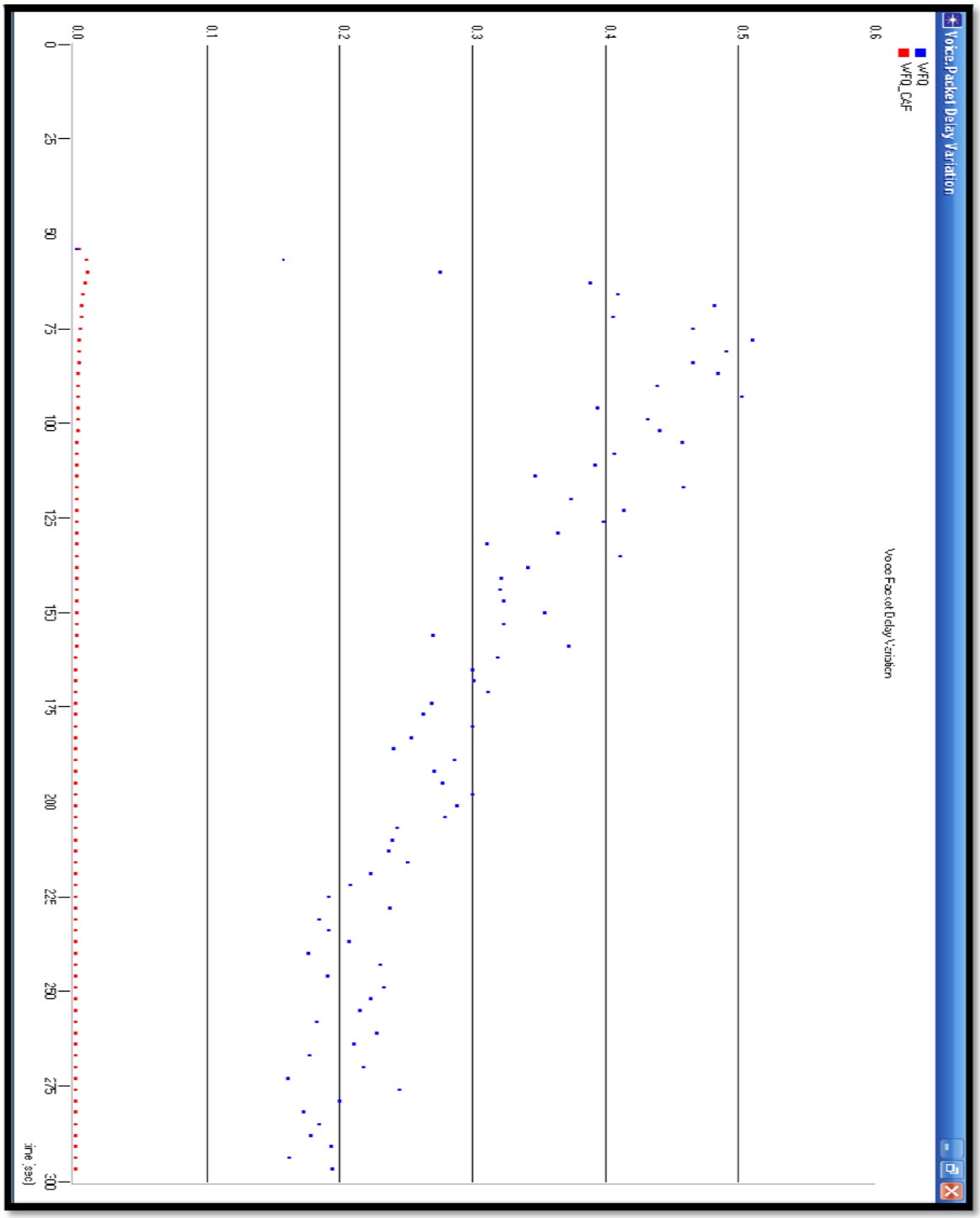

Figure (4.25): WFQ and WFQ (CAR) Voice packet Delay Variation

- **Video Conferencing Traffic**

The heaviest traffic application is the video conferencing. Figures (4.26), (4.27) and (4.28) below show the video traffic sent and video traffic received and video packet end to end delay respectively, figure (4.26) shows how CAR manage traffic rate when network are congested.

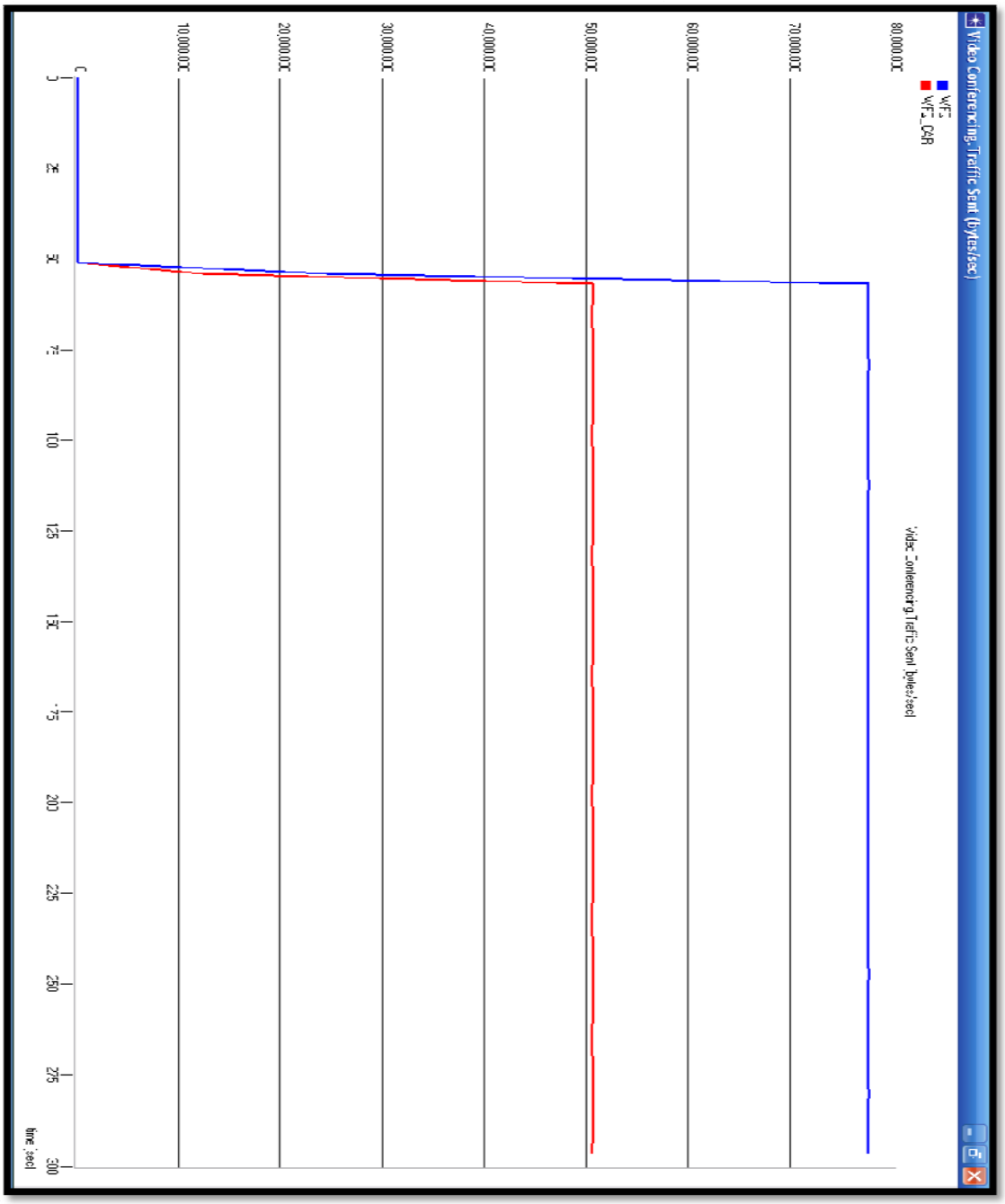

Figure (4.26): WFQ and WFQ (CAR) Video traffic sent (bytes/second)

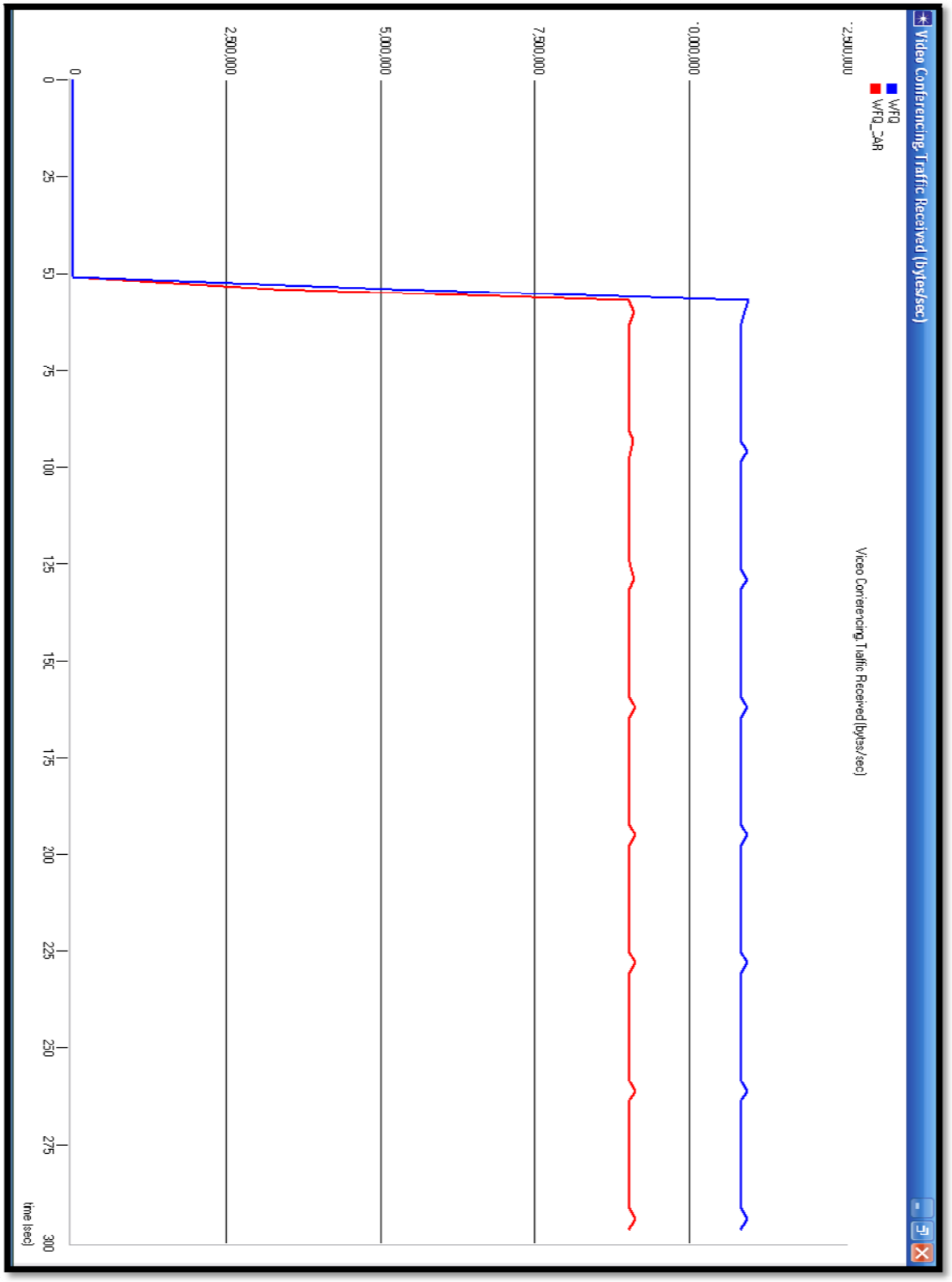

Figure (4.27): WFQ and WFQ (CAR) Video traffic received (bytes/second)

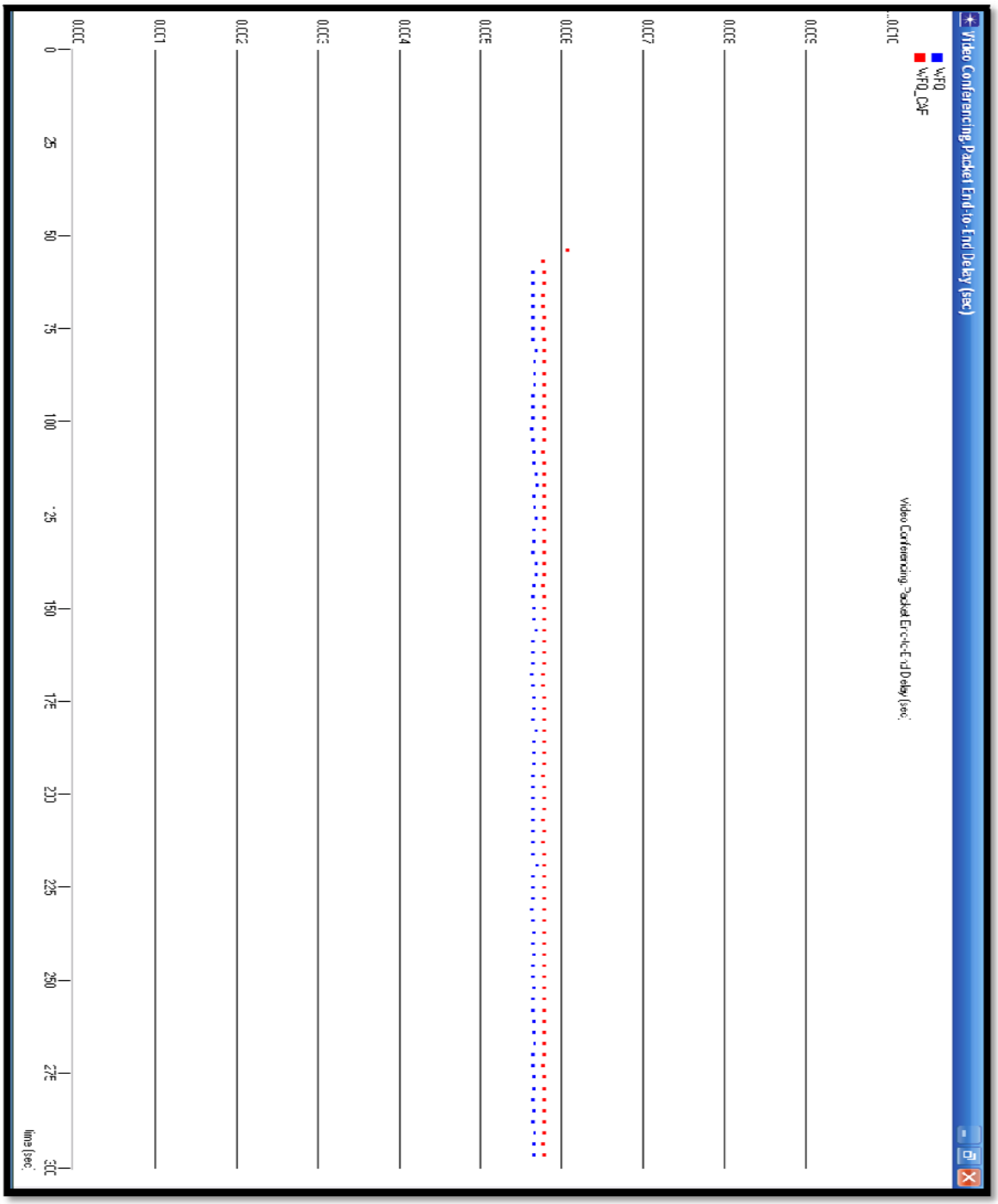

Figure (4.28): WFQ and WFQ (CAR) Video packet end-to-end delay (second)

Figure (4.28) above show, that WFQ with CAR has packet video delay in the accepted range which is also related to the traffic rate management.

- **IP Traffic Dropped**

As said previously, the IP traffic dropped for WFQ with committed access rate is less than with WFQ only; this is due to the CAR algorithm. Figure (4.29) shows a global IP traffic dropped for the whole system.

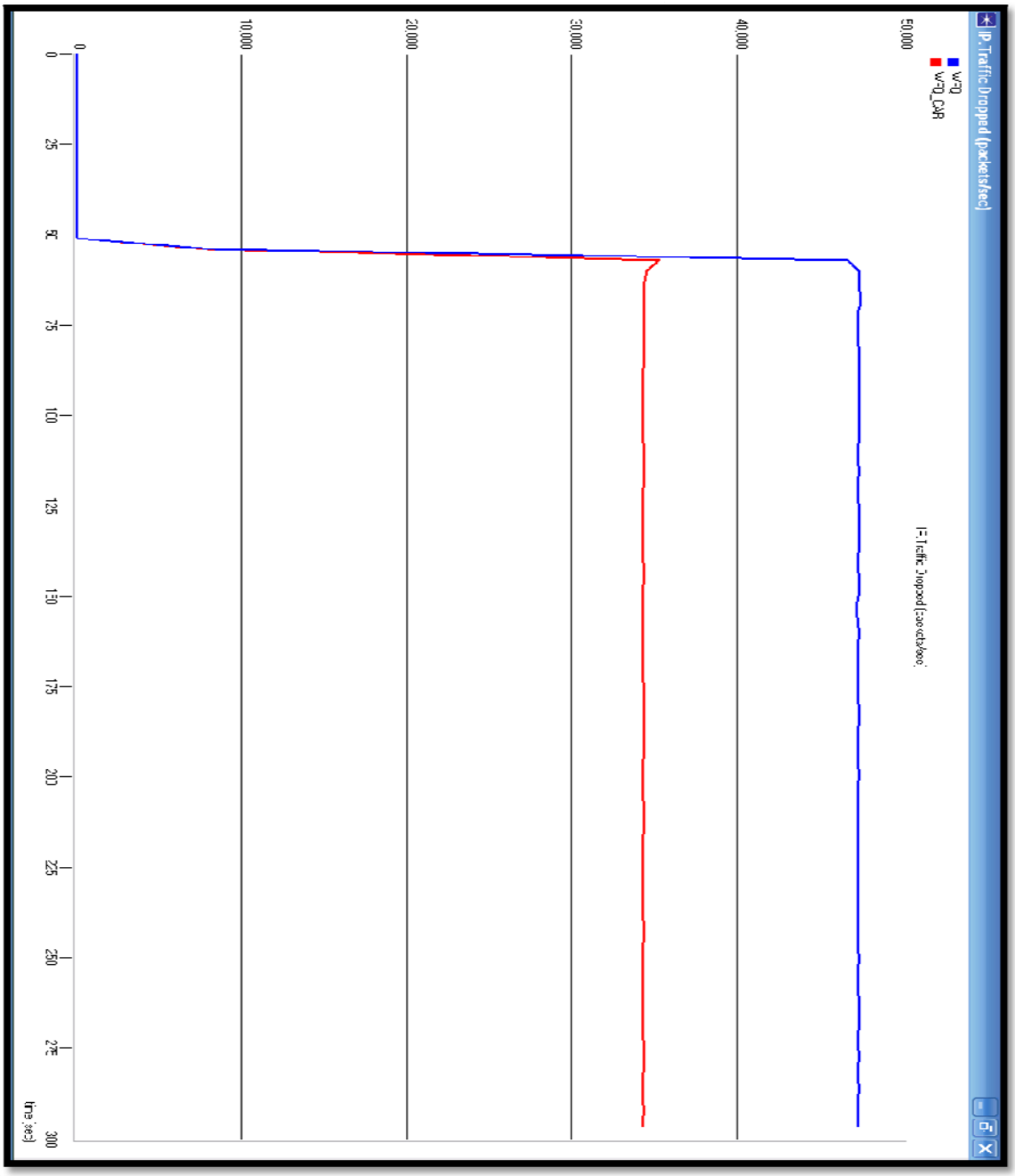

Figure (4.29): WFQ and WFQ (CAR) Global IP traffic dropped (packets/second)

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The following can be concluded from the work in this thesis:

1. A large number of information flows are carried across the network simultaneously. Each flow requires servicing according to specified QoS requirements. Each flow passes several network devices (routers and switches) along its route from the source node to the destination node. Quality of service can treat packets with different ways, which is used to improve the operation in the networks and to solve the congestion when happened. It does not allow the low priority traffic to delay the high priority traffic.
2. FIFO queuing performs no prioritization of data packets on user data traffic. It entails no concept of priority or classes of traffic. When FIFO is used, ill-behaved sources can consume available bandwidth, important traffic may be dropped because less important traffic fills the queue. From the result, FIFO gives acceptable results but it does not ensure that higher priority packets are treated first.
3. CQ guarantees some level of service to all traffic because of their ability to allocate bandwidth to all classes of traffic. The size of the queue can be defined by determining its configured packet-count capacity, thereby controlling bandwidth access. This type of scheduling can produce a very acceptable packet delay but with unacceptable packets dropped because it will treat packet depending to its weight.
4. PQ guarantees strict priority as it ensures that one type of traffic will be sent, possibly on the account of all others. For PQ, a low priority queue can be detrimentally affected. In the worst case, never allow to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high. The results show that this

type behaves similar to CQ with more insurance that higher priority packets serve first.

5. The Weighted Fair Queuing algorithm shows that it provides a certain minimum bandwidth to all classes of traffic, or at least to guarantee the observance of some requirements to delays. The results show that the WFQ is the best scheduling because it gives acceptable packets delay and dropping.
6. Committed Access Rate implements both classification services and policing through rate limiting. When this multifaceted feature is executed with WFQ, less packet delay and drop are found, this is related to the rate limit statement and traffic policing used by CAR algorithm which implement by access bandwidth management.
7. Finally using CAR with WFQ algorithm produced accepted packet delay for all applications and accepted packet delay by controlling the traffic sent to comfort with network capacity.

5.2 Future Works

Future work involves the following suggestions:

1. Security is important issues which must be establish on almost networks, this is done by encryption the data on the links.
2. Congestion problem in this thesis is managed but this problem can also be avoided by many algorithms like Random Early Detection and Weighted Random Early Detection, or by TCP congestion avoidance algorithms like Reno and Tahoe.

3. The compression schemes also can be used to improve throughput of the network. Many types of compression can be implemented like header compressions, packet compression and data compression
4. Some TCP parameters could be set to appropriate values to improve the performance of the network. These parameters include Maximum Segment Size, Receive Buffer, Receive Buffer Adjustment and Receive Buffer Usage Threshold.
5. Instead of using wired link, wireless can be established between the governorates, also give clients the property of mobility with known region.

.

References:

[1] A. S. Tanenbaum, “Computer Networks”, Fourth Edition, Prentice Hall PTR, New Jersey 2003.

[2] M. Welzl, "Network Congestion Control: Managing Internet Traffic", John Wiley & Sons Ltd, England, 2005.

[3] N. F. Mir, "Computer and Communication Networks", Prentice Hall, USA, 2006.

[4] OPNET IT Guru Academic Edition helps document Version-1997, OPNET technologies.

[5] P. Sebos, "Building a topology generator, a fault generator and a distributed platform for running simulations using OPNET as a simulation tool”, Columbia University, USA, 2005.

[6] OPNET helps documentation, http://www.opnet.com, June, 2007.

[7] A. Kim, "OPNET Tutorial", White paper, March 7, 2003.

[8] T. Bjornerback, "Analysis of a Network simulation tool for advanced courses in computer communication" M.Sc. Thesis, Department of computer engineering, Umea University, 2001.

[9] M. Lommerdal and L. Soder. “Simulation of Congestion Management methods” IEEE Bologna PowerTech Conference, Italy, June 2003.

[10] V. Cholvi, V. Laderas, L. Lopez and A. Fernandez, "Self-adapting network topologies in congested scenarios", Physical Review, The American Physical Society, 21 March, 2005.

[11] B, K. Singh and N Gupte, “Congestion and decongestion in a communication network”, Physical Review, The American Physical Society, 31 May, 2005.

[12] C. Binbin, "Scheduling deadline-constrained bulk data transfers to minimize network congestion", 28 October, 2006.

[13] B. Kleinebecker and C. Skelley, "A Proposal for Better Management of Internet QoS" Technology Futures, July 21, 2006.

[14] M. Singh, P. Pradhan and P. Francis, "MPAT: Aggregate TCP Congestion Management as a Building Block for Internet QoS" IBM™ T.J. Watson research center, France 2007.

[15] A. Chaintreau, F. Baccelli and C. Diot, "Impact of TCP-like Congestion Control on the Throughput of Multicast groups", White paper, 2008.

[16] Y. Tang, H Xu, and Q Wan, “Research on the Application of Financial Transmission Right in Congestion Management” IEEE, Nanjing China, April 2008.

[17] C. Hunt, “TCP/IP Network Administration”, Second Edition, O’Reilly & Associates, December 1997.

[18] M. Y. Rhee, “Internet Security”, John Wiley & Sons, Ltd, 2003.

[19] J. Postel, “Internet Protocol”, RFC 791, USC/Information Science Institute, September 1981.

[20] L. Dostalek and A. Kabelova, “Understanding TCP/IP”, PackT Publishing, 3 Aug 2006.

[21] M. W. Murhammer, O. Atakan, S. Bretz, L. R. Pugh, K. Suzuki and D. H. Wood, “TCP/IP Tutorial and Technical Overview”, IBM Corporation, International Technical Support Organization, Sixth Edition, October 1998.

[22] J. Manner, “Provision of Quality of Service in IP-based Mobile Access Networks”, PhD Thesis, Department of Computer Science, University of Helsinki, Finland, 2003.

[23] D. Chalmers, M. Sloman. “A survey of quality of service in mobile computing environments”, IEEE communications survey, April 1999.

[24] G. Huston, “Next steps for the IP QoS Architecture”, RFC 2990, November 2000.

[25] Cisco documentation, “Cisco IOS Quality of Service Solutions Configuration Guide Release 12.2”, Cisco Systems, Inc., 2006.

[26] S. Vegesna, “IP Quality of Service”, Cisco Press, USA, 2001.

[27] N. Olifer, V.Olifer, J. Wiley & Sons, “Computer Networks: Principles, Technologies and Protocols for Network Design”, John Wiley & Sons 2005.

[28] Cisco documentation, “Understanding Delay in Packet Voice Networks”, Cisco Systems, Inc, 3/10/2005.

[29] Video delay, www.vbrick.com/documentation/WhitePapers/ Streaming _Video Delay.pdf, January 2008,

[30] P. Almquist, “Type of Service in the Internet Protocol Suite”, RFC 1349, July 1992.

[31] M. Welzl, “Network Congestion Control Managing Internet Traffic”, John Wiley & Sons Ltd, 2005.

[32] G. Armitage, “Quality of Service in IP Networks”, New Riders Publishing, First Edition, April 07, 2000, Last updated on 11/30/2001

APPENDIX A

OPNET UTILITIES AND DEFINITIONS

A.1 OPNET Simulator

OPNET (Optimized Network Engineering Tool) provides a comprehensive development for the specification, simulation and performance analysis of communication networks. A large range of communication systems from a single WLAN to global Satellite networks can be supported. Discrete event simulations are used as the means of analyzing system performance and behavior. The key features of OPNET are summarized here as [6]:

1. Modeling and Simulation Cycle, OPNET provides powerful tools to assist user to go through three out of the five phases (as shown in Figure A.1) in design circle (i.e. the building of models, the execution of a simulation and analysis of the output data).
2. Hierarchical Modeling, OPNET employs a hierarchical structure to modeling. Each level of the hierarchy describes different aspects of the complete model being simulated.
3. Specialized in communication networks detailed models provide support for existing protocols and allow researchers and developers to either modify these existing models or develop new models of their own.
4. Automatic simulation generation OPNET models can be compiled into executable code.

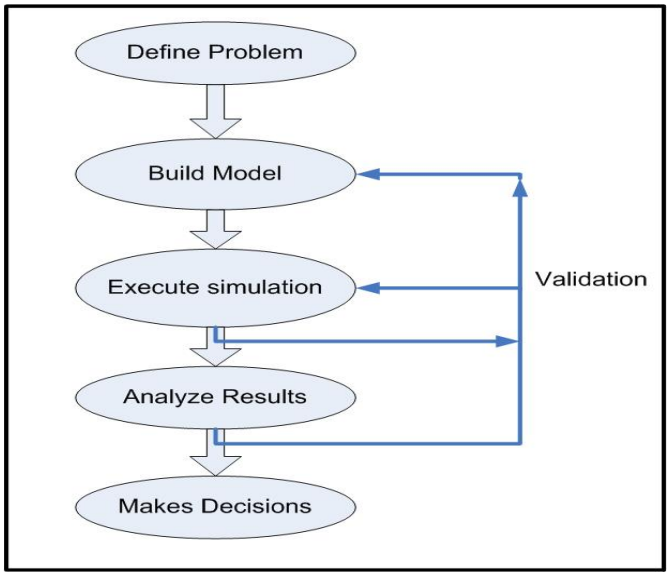

Figure (A.1): Modeling and simulation Cycle

A.2 The Toolbar

The toolbar buttons (icons) functions are as follows as shown in Figure (A-2):

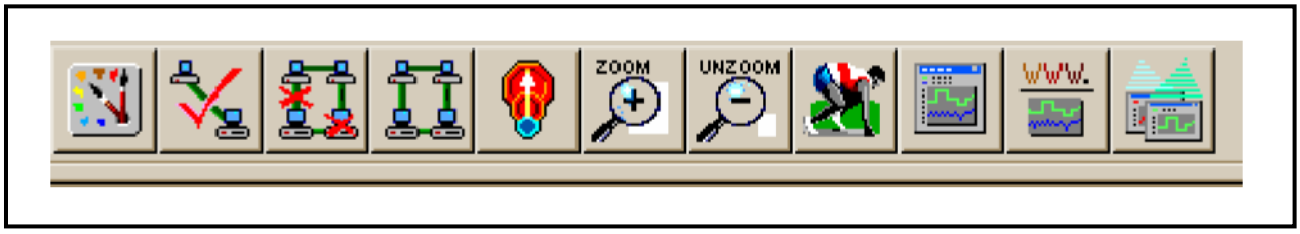

Figure (A.2): the toolbar

The left most icons are the “Pallet” this icon brings up a group of related objects to be used to populate the work space. The current pallet can be change to any other pallet as well as new pallet can be created of individual components as you wish. The second link is to verify the connections between objects. This verifies that all connections are valid. However this does not mean any higher level protocols are set correctly or not. Next to icons are used to fail or un-fail links.

The next icon is to move up in layer. The red octagon is used to indicate subnets. The arrow in the middle indicates moving up the layer. Next

to tools are to expand and compress the viewing of the work space. The next icon (man getting set to spring) is used to startup a simulation. Clicking this icon bring up the “Configure Simulation” menu. The configuration menu is used to configure various run parameters. Do not go crazy in setting the duration and update interval. Your simulation run may last excessively long time without adding any more information. Next icon is used to “View Results”. This brings up the configuration for bringing up the graphs. The standard means of expanding the listing is used [+] and [-]. The report with [+] indicates there is/are data underneath which can be used to generate graph(s). The bottom three pull-down selection menu is used to configure how the data is to be displayed. In general we shall use “Statistics Stacked”, “Time Average”, and “This scenario” as selection for displaying the graphs. Next icon is used to open most recently generated result published on the web. The last icon is used to hide and un-hide the graphs generated [11].

A.3 Utilities

There are some important utilities (objects) in OPNET that are used to simplify and facilitate the work with it. Some of these objects that are used in this thesis are Application Configuration, Profile Configuration and QoS Attribute Configuration [6]:

- Application Configuration Object**:** Specifies applications using available application types. You can specify a name and the corresponding description in the process of creating new applications. For example, "Web Browsing (Heavy HTTP)" indicates a web application performing heavy browsing using HTTP. The specified application name will be used while creating user profiles on the “Profile Configuration" object. The Application Configuration object is shown in figure (A-3).

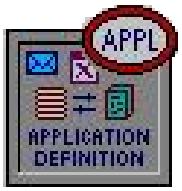

Figure (A.3): Application Object

The Application Definitions Table of the Application Configuration object is shown in figure (A-4).

The **Name field** is a symbolic name for the application. It is recommended that it used with very specific names that must identify the application. For example "My_web_browsing_app" or “Engineering_email" or “Sales_db_app" etc. Later on in the application definition, it will be associated (the "Name") with a specific application model such as "Email" or "FTP".

The **Description field** includes a table that specifies all the application parameters. Note that for a particular application name, it can only choose one application within the description table.

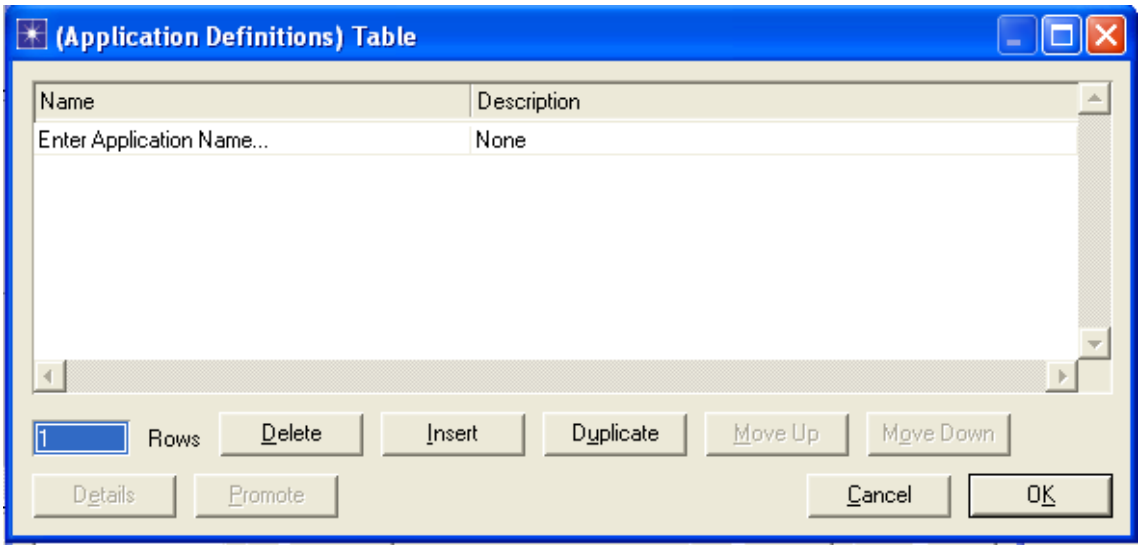

Figure (A.4): Application Definition Table

- Profile Configuration Object**:** The "Profile Configuration" node can be used to create user profiles. These user profiles can then be specified on different nodes in the network to generate application layer traffic. The applications defined in the "Application Configuration" objects are used by this object to configure profiles. Therefore, you must create applications using the "Application Configuration" object before using this object. You can specify the traffic patterns followed by the applications as well as the configured profiles on this object. The Profile Configuration object is shown in Figure (A-5).

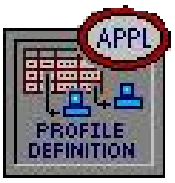

Figure (A.5): Profile Configuration object

The Profile Configuration Table of the Profile Configuration node is shown in figure (A-6). The following are the definition of each column:

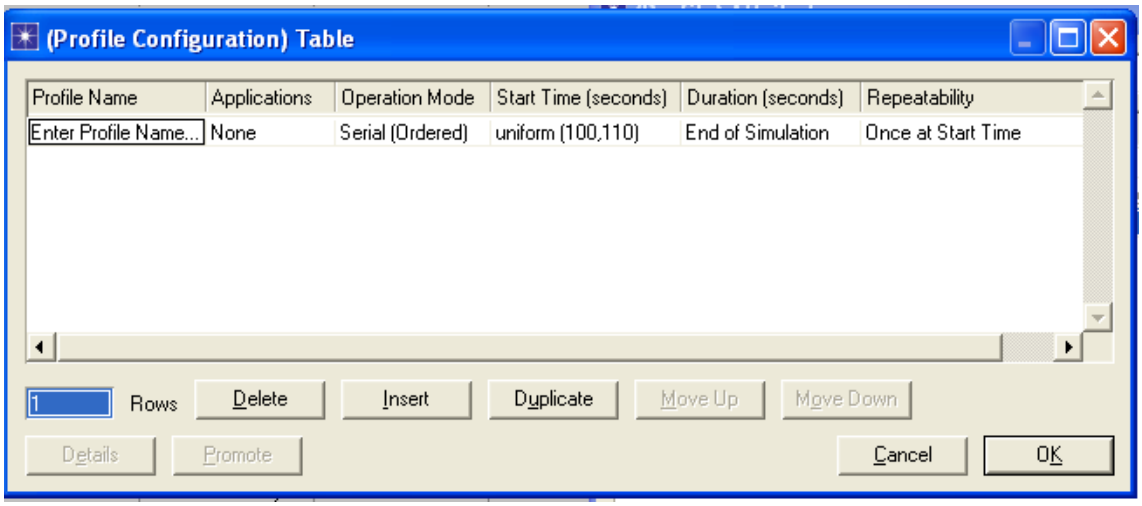

Figure (A.6): Profile Configuration Table

Profile Name: The symbolic name of the profile.

Application: The different applications that are executed within the profile and their characteristics.

Operation Mode: Defines how applications will start. Serial (Ordered) - they can start one after each other in an ordered manner (first row to last row). Serial (Random) - they can start one after each other in a random manner. Simultaneous- they can start all at the same time.

Start Time: Defines when during the simulation the profile session will start.

Duration: Defines the maximum amount of time allowed for the profile before it aborts. When set to 'End of Simulation' the profile is allowed to continue indefinitely. Profiles that repeat should not have their duration set to 'End of Simulation'.

Repeatability: Specifies the parameters used to repeat execution of the profile.

- QoS Attribute Configuration Object**:** Defines attribute configuration details for protocols supported at the IP layer. These specifications can be referenced by the individual nodes using symbolic names (character strings). Queuing Profiles defines different queuing profiles such as FIFO, WFQ, and CQ. The QoS Attribute object is shown in Figure (A-7).

Figure (A.7): QoS Attribute object

The Queuing Configuration Table of the QoS Attribute object is shown in figure (A-8). The following are the definition of each column:

Weight: Weights are attributed to each queue. The weight indicates the allocated bandwidth for the queue. A higher weight indicates larger allocated bandwidth and shorter delays.

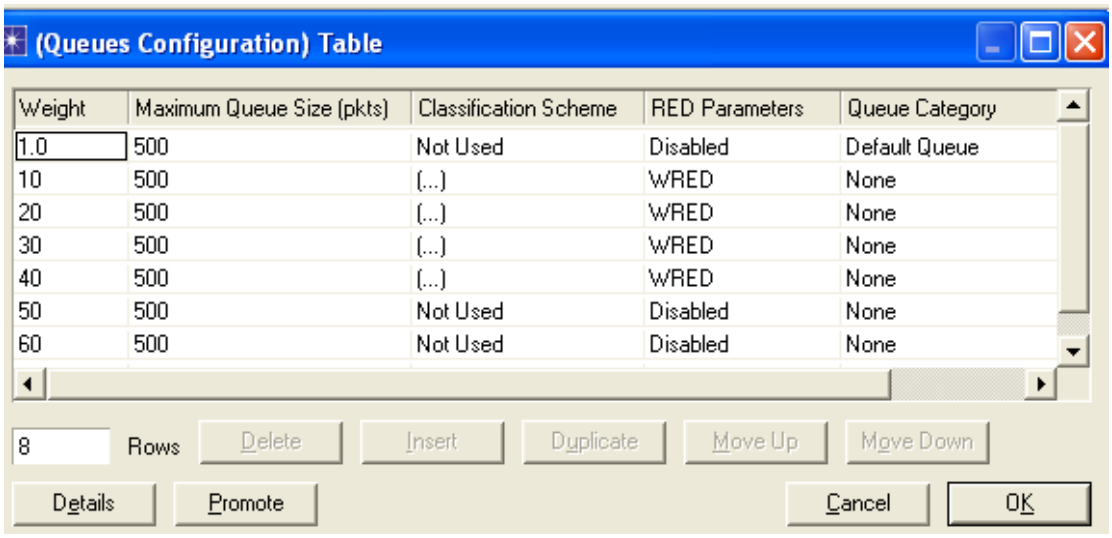

Figure (A-8): QoS Configuration Table

Maximum Queue Size: Maximum number of packets per queue. Used when the interface is congested (when the total number of buffered packets in all the queues is reached).

Classification Scheme: Defines the criteria for the packet in order to queue it. If no criterion matches a packet, this one is put in the queue configured as the "Default Queue".

RED Parameters: Set/Unset RED or WRED.

Queue Category: The Queue Category attribute determines whether the queue has the Default Queue and/or Low Latency Queue property.

الخلاصه

ادارة ظاهرة الاختناق تتم عن طرق السيطرة على الاختناق وقت حصوله. من الطرق التي تتعامل بها عناصر الشبكه مع التدفق الكثيف للمرور القادم هو باستعمال خوارزمية الانتظار في الصف لتراتبية المرور، ثم اتخاذ قرار بشأن بعض طرق منح الاولويات للوصله الخارجه. كل خوارزمية انتظار في الصف صممت لحل مشكله مرور معينه في الشبكه ويكون لها تأثير معين على أداء الشبكه. العمل المعروض في هذه الاطروحة يتعامل مع قضيه مهمه وهي تقنيات نوعية الخدمه، التي يمكن تطبيقها لتحسين عمل شبكات واسعة المساحه. نوعية الخدمه هي التأثير الكلي على أداء الخدمه، التي تحدد درجة رضاء مستخدم تلك الخدمه.

في هذه الاطروحه، تم تطبيق وتقييم انظمه جدولة الحزم (الاولويه في الخدمه للاول في الوصول، الوزن المنصف للانتظار في الصف، اولوية الانتظار في الصف و الانتظار في الصف حسب الطلب) و كذلك أستخدمت خوارزمية مراقبة المرور، بتطبيقات مختلفه (نقل البيانات، البريد الالكتروني، المحادثه الصوتيه و المحادثه الصوريه) مع اولويات مختلفه (خلفيه، المعيار، الجهد الممتاز والتحكم بتدفق الصوت والصوره)

نفذت المحاكات بالحاسبه للدراسة والتحقق من الاليات المشار اليها سابقا في تحسين الاداء بواسطة المحاكي (OPNET)،حسبت النتائج الخارجيه والتي تعتمد على بعض الاحصائيات التي تشرح سلوك الشبكه مثل التأخير بين طرفين، تغير التأخير و خسارة الرزم الخ.

طبقاً للنتائج المكتسبه من المحاكي (OPNET)، وجدنا انه نسبيا الوزن المنصف للانتظار في الصف هي النوع الملائم للجدولة وذلك لمنحهه ديناميكيه وعدالة في الصف والتي تقسم الموجه عبر طوابير المرور مستندة على الأوزان. كذلك هذه النتائج سوف تصبح اكثر قبول عندما يتنفذ التزام نسبة دخول مع الوزن المنصف للانتظار، هكذا الاخير سوف يستخدم القابليه للالتزام نسبة الدخول لاداره نسبه المرور باستخدام اداره مدخل الموجه والتي تتنفذ باستخدام جمل حصر النسبه.

تحليل اختناق الشبكه و جودة الخدمه بواسطه OPNET

اطروحة
مقدمة الى كلية هندسة المعلومات في جامعة النهرين كجزء من
متطلبات نيل درجة الماجستير في هندسة المعلومات

من قبل
فرات نضال توفيق
(بكلوريوس ٢٠٠٥ م)

ربيع الاول ١٤٢٩
اذار ٢٠٠٩