Sudan University of Science and Technology

College of Engineering

School of Electronics Engineering

IMPLEMENTATION OF AVQ BASED ALGORITHM FOR CONGESTION AVOIDANCE IN WIRELESS LAN

تجنب الازدحام في الشبكات اللاسلكيه بتطبيق خوارزميه ال avq

A Research Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honors) in Electronics Engineering

Prepared By:

1. **Asma Abd elazeez Albadwey Almobarak**
2. **Amel Abd alellah Haj ali ahmed**
3. **Sumia Babikir Mohammed Awad**
4. **Firdous Alsiddeg Mohammed noor**

Supervisor:

Dr. Salaheldin Mohamed Ibrahim Edam

October 2015

بسم الله الرحمن الرحيم

قَالَ تَعَالَىٰ:

﴿ إِنَّ فِي خَلْقِ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ وَٱخْتِلَٰفِ ٱلَّيْلِ وَٱلنَّهَارِ لَءَايَٰتٍ
لِّأُو۟لِى ٱلْأَلْبَٰبِ ﴿١٩٠﴾ ٱلَّذِينَ يَذْكُرُونَ ٱللَّهَ قِيَٰمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ
وَيَتَفَكَّرُونَ فِى خَلْقِ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَٰذَا بَٰطِلًا
سُبْحَٰنَكَ فَقِنَا عَذَابَ ٱلنَّارِ ﴿١٩١﴾ ﴾

صدق الله العظيم

سورة آل عمران: الآيات (١٩٠ ــ ١٩١)

Dedication

Dedication to our mothers…

With warmth and faith...

Dedication to our fathers…

With love and respect …

Dedication to all our teachers …

In whom we believe so much …

Dedication to our friends...

Whom we cherish their friendship

Dedication to our special people

Who mean so much to us…

Acknowledgment

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this research.

We would like to express our special thank of gratitude to our supervisor, DrSalaheldinEdam, who gave us a golden opportunity to do this wonderful project under his supervisingwhich helped us in doing a lot of research and we came to know about so many new things we really thankful to them.And special thanks for Dr. AbuAagla head master of the school of electronic engineering.

In addition we would also like to thanks our parents, families and friends who help us a lot in finalize.g this project within the limited time frame.

Abstract

The IEEE 802.11 wireless local area networking (WLAN) standard defines one of the most widely deployed wireless technologies in the world. The popularity of wireless networking is driven by the ubiquity of portable mobile hand-held devices, and the convenience of untethered communications. WLAN are being designed for fast transmission of large amounts of data, for which Congestion Control Algorithms (CCAs) are very important. Active Queue Management (AQM) with Explicit Congestion Notification (ECN), packets generated by different data sources are marked at the network's gateways. In other algorithms, packets are dropped to avoid and control congestion at gateways.

The purpose of this research is to cover some of congestion control techniques in WLAN Network and focuses on some of the recent active queue management (AQM) algorithms that are widely used in many applications; RED, BLUE, DECbit, drop tail, and AVQ schemes. In addition to implement these algorithms using matlab simulator and compare the efficiency of each algorithm of them with each other in term of packet loss, throughput and delay and analyze the results and referring it as a comparative study. This research also contain aside of optimize and modify AVQ algorithm that reduce the congestion.

المستخلص

الشبكة المحلية اللاسلكية (IEEE 802.11 (WLAN تعتبر إحدى أكثر التقنيات اللاسلكية المستخدمة في العالم. السبب وراء هذا الإنتشار هو التوسع في استخدام الاجهزة المحمولة hand-held devices وسهولة الاتصالات الغير مقيدة بأسلاك التوصيل. تم تصميم الشبكات المحليه اللاسلكية حتى توائم سرعات ارسال عالية وأحجام كبيرة للمعلومات المحمولة لذلك لابد من وجود طرق لتنظيم الارسال والتحكم في التدفق. لذلك تم تصميم خوارزميات التحكم في الإزدحام(AQM.(CCAs ترجع إشارة الإزدحام مع كل الحزم المولدة من مصادر المعلمومات المتفرقة في الشبكة. أما الخوارزميات الأخرى تقوم بإسقاط الحزم للتحكم وتجنب الإزدحام في الشبكة.

الهدف من هذا البحث هو تغطية بعض التقنيات التي تدخل في عملية التحكم في الإزدحام في الشبكات المحلية اللاسلكية و التركيز على خوارزميات active queue management للتحكم في الإزدحام التي تستخدم في كثير من التطبيقات وهي : خوارزمية RED، وخوارزمية BLUE، وخوارزمية DECbit، وخوارزمية drop tailو خوارزمية AVQ. تم استخدام برنامج matlab لتنفيذ المحاكاة الخاصة بالخوارزميات و مقارنة النتائج من حيث الفعالية لكل خوارزمية من حيث فقدان الحزم و التأخير ، بالاضافة الى هذه الخوارميات تم اقتراح خوارزمية AVQلتقليل الازدحام والتحكم به.

Table of content:

List of Table

List of Figures

LIST OF ABBREVIATIONS

AP	Access Point
AQM	Active Queue Management
AVQ	Adaptive Virtual Queue
BSA	Basic Service Area
BSS	Basic Service Set
CCAs	Congestion Control Algorithms
CoDel	Controlled delay
DT	Drop Tail
ECN	Explicit Congestion Notification
ESS	Extended Service Set
FIFO	First-In-First Out
GKVQ	Gibbens-Kelly Virtual Queue
MSDU	MAC Service Data Unit
MQL	Mean Queue Length

NICs	Network Interface Cards
QoS	Quality of Service
RED	Random Early Detection
RED-PD	RED with Preferential Dropping
REM	Random Exponential Marking
RRED	Robust random early detection
RSFB	Resilient Stochastic Fair Blue
SFB	Blue and Stochastic Fair Blue
TCP	Transmission Control Protocol
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network

Chapter One

Introduction

Chapter One: Introduction

1.1 Preface:

Congestion in a network may occur if the load on the network (the number of packets sent to the network) is greater than the capacity of the network (the number of packets a network can handle). Congestion reduces the performance of a network [1].

Congestion control is a method used for monitoring the process of regulating the total amount of data entering the network. Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity. Congestion control techniques can be broadly classified two broad categories: Open-Loop and close-loop [1]. Congestion control mechanism is one of the key that keeps any network efficient and reliable for the users.

Active Queue Management (AQM) is one such mechanism which provides better control. It works at the router for controlling the number of packets in the router's buffer by actively discarding an arriving packet [2].

Adaptive Virtual Queue (AVQ) is designed that results in low-loss, low-delay and high utilization operation at the link. AVQ algorithm maintains a virtual queue whose capacity is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect the new packets in the real queue to reflect the new arrival. Packets in the real queue are marked/dropped when the virtual buffer overflows[3].

1.2Problem Statement:

Congestion in a network that may occur if the load on the network greater than the capacity of the network, During congestion large amounts of packet experience delay or even be dropped due to the queue overflow , thus degradation of the throughput and large packet loss rate. Congestion may lead to a decreased efficiency and reliability of the whole network.

1.3Proposed Solution:

We propose to study the congestion control algorithms and apply the AVQ algorithm to see the effect of the network.

1.4 Aim and Objectives:

The aim of this project is to study the congestion control mechanisms. The Objectives are to simulate the AVQ algorithm to validate and evaluate the results, and enhance the results by changing the parameter.

1.5 Methodology:

Collecting information was the first step in attempting to study and investigate the wireless local area network, in order to apply an analysis in order to study the parameters used for Enhancement of congestion control in WLAN and the adaptive virtual queue (AVQ) was chosen. The system will be implemented on Matlab simulation program.

1.6 Chapters Layout:

This research consists of five chapters and is divided as the following:

Chapter two: represents a literature review of congestion control.

Chapter three: the methodology was written.

Chapter four: represents the results and discussion.

Chapter five: the conclusion and recommendation are included along with the references.

Chapter Two

Congestion Control

Chapter Two: Congestion Control

2.1 Background:

Network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. Typical effects include queuing delay, packet loss or the blocking of new connections. A consequence of the latter two effects is that an incremental increase in offered load leads either only to a small increase in network throughput, or to an actual reduction in network throughput [1].

Network protocols which use aggressive retransmissions to compensate for packet loss tend to keep systems in a state of network congestion, even after the initial load has been reduced to a level which would not normally have induced network congestion. Thus, networks using these protocols can exhibit two stable states under the same level of load. The stable state with low throughput is known as congestive collapse.

Modern networks use congestion control and congestion avoidance techniques to try to avoid congestion collapse. These include: exponential back off in protocols such as 802.11 CSMA/CA and the original Ethernet, window reduction in TCP, and fair queuing in devices such as routers. Another method to avoid the negative effects of network congestion is implementing priority schemes, so that some packets are transmitted with higher priority than others. Priority schemes do not solve network congestion by themselves, but they help to alleviate the effects of congestion for some services. A third method to avoid network congestion is the explicit allocation of network resources to specific flows [4].

2.2 Literature Review:

A wireless local area network (WLAN) is a wireless computer network that links two or more devices using a wireless distribution method (often spread-spectrum or OFDM radio) within a limited area such as a home, school, computer laboratory, or office building. This gives users the ability to move around within a local coverage area and still be connected to the network, and can provide a connection to the wider Internet. Most modern WLANs are based on IEEE 802.11 standards, marketed under the Wi-Fi brand name [5].

2.2.1 IEEE 802.11 architecture:

IEEE 802.11 is a basic standard for Wireless Local Area Network (WLAN) communication. The IEEE first developed in 1997. The IEEE designed 802.11 to support medium-range, higher data rate applications, such as Ethernet networks, and to address mobile and portable stations. The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS) [6].

2.2.1.1 Basic Service Set:

The basic service set (BSS) is the fundamental building block of the IEEE 802.11 architecture. A BSS is defined as a group of stations that are under the direct control of a single coordination function. The geographical area covered by the BSS is known as the basic service area (BSA), which is analogous to a cell in a cellular communications network. Conceptually, all stations in a BSS can communicate directly with all other stations in a BSS. However, transmission medium degradations due to multipath fading or interference from nearby BSSs reusing the same Physical-layer characteristics can cause some stations to appear “hidden” from other stations [7].

The BSS without an AP is a stand-alone network and can't send data to other BSSs. It is called an ad hoc architecture. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network [8].

a. Ad Hoc Network:

In ad hoc mode, the wireless network is relatively simple and consists of 802.11 network interface cards (NICs). The networked computers communicate directly with one another without the use of an access point (Fig 1) [9].

There are some drawbacks for ad hoc networks. First, it is much more difficult and complex to perform routing in ad hoc networks because of frequent changes in the network topology due to host mobility. Second it is more difficult to control or coordinate proper operation of an ad hoc network, since each wireless host may have its own algorithms to perform activities such as time synchronization, power management, and packet scheduling. In an infrastructure network, these algorithms are often implemented in and thus harmonized by the base stations or access points [10].

b. Infrastructure Network:

The infrastructure is a network with an Access Point (AP), in which all stations must be associated with an AP to access the network. Station communicates with each other through the AP (Fig 1).

The access point performs the conversion of 802.11 packets to 802.3 Ethernet LAN packets. Data packets traveling from the LAN to a wireless client are converted by the access point into radio signals and transmitted out into the environment [9].

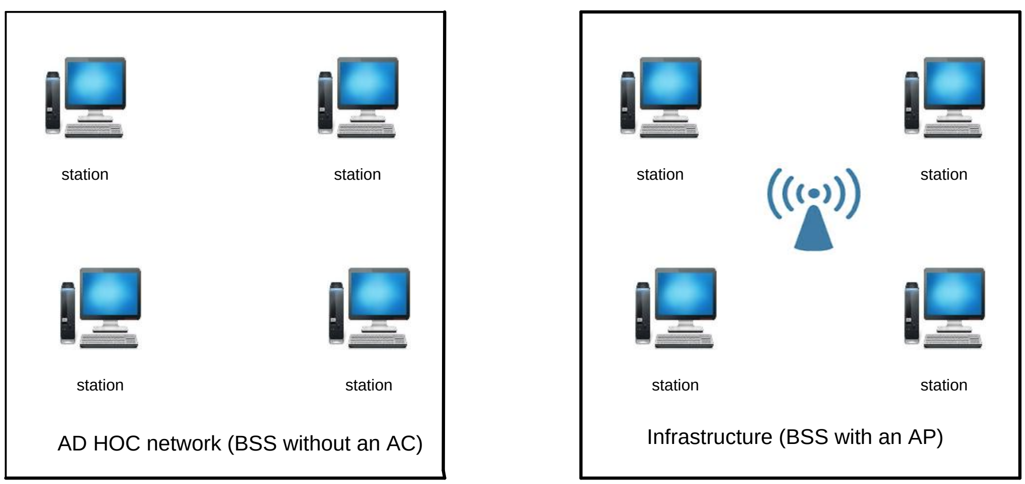


Figure (2.1) Basic service sets (BSSs)

2.2.1.2 Extended Service Set:

BSSs can create coverage in small offices and homes, but they cannot provide network coverage to larger areas. 802.11 allow wireless networks of arbitrarily large size to be created by linking BSSs into an extended service set (ESS). An ESS is created by chaining BSSs together with a backbone network. 802.11 do not specify a particular backbone technology; it requires only that the backbone provide a specified set of services [11].

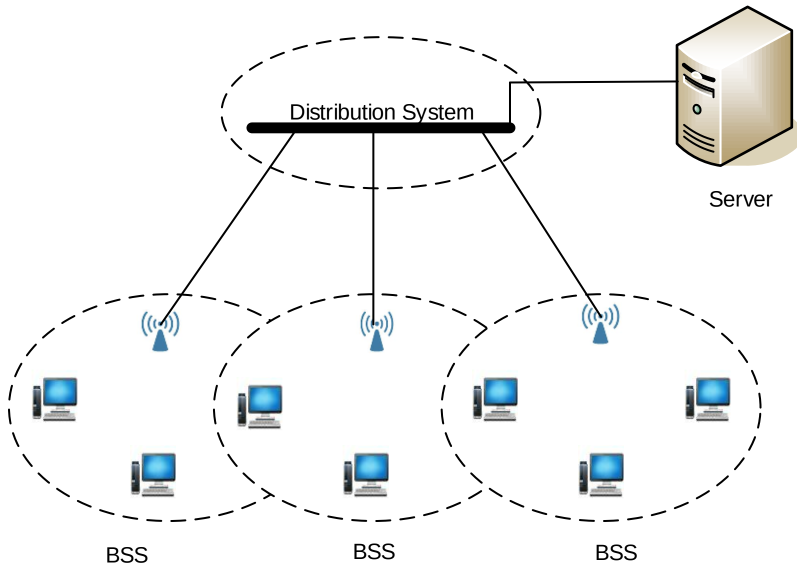


Figure (2.2) Extended service sets (ESSs)

2.2.2 Network Services:

One way to define a network technology is to define the services it offers and allow equipment vendors to implement those services in whatever way they see fit. 802.11 provide nine services. Only three of the services are used for moving data; the remaining six are management operations that allow the network to keep track of the mobile nodes and deliver frames accordingly.

2.2.2.1 Distribution:

This service is used by mobile stations in an infrastructure network every time they send data. Once a frame has been accepted by an access point, it uses the distribution service to deliver the frame to its destination. Any communication that uses an access point travels through the distribution service, including communications between two mobile stations associated with the same access point.

2.2.2.2 Integration:

Integration is a service provided by the distribution system; it allows the connection of the distribution system to a non-IEEE 802.11network. The integration function is specific to the distribution system used and therefore is not specified by 802.11, except in terms of the services it must offer.

2.2.2.3Association:

Delivery of frames to mobile stations is made possible because mobile stations register, or associate, with access points. The distribution system can then use the registration information to determine which access point to use for any mobile station.

2.2.2.4Reassociation:

Reassociations are initiated by mobile stations when signal conditions indicate that a different association would be beneficial; they are never initiated by the access point. After the reassociation is complete, the distribution system updates its location records to reflect the reach ability of the mobile station through a different access point.

2.2.2.5 Disassociation:

To terminate an existing association, stations may use the disassociation service. When stations invoke the disassociation service, any mobility data stored in the distribution system is removed. Once disassociation is complete, it is as if the station is no longer attached to the network. Disassociation is a polite task to do during the station shutdown process.

2.2.2.6Authentication:

Authentication is a necessary prerequisite to association because only authenticated users are authorized to use the network.

2.2.2.7 Deauthentication:

Deauthentication terminates an authenticated relationship. Because authentication is needed before network use is authorized, a side effect of Deauthentication is termination of any current association.

2.2.2.8 Privacy:

802.11 provide an optional privacy service called Wired Equivalent Privacy (WEP). WEP is not ironclad security in fact; it has been proven recently that breaking WEP is easily within the capabilities of any laptop its purpose is to provide roughly equivalent privacy to a wired network by encrypting frames as they travel across the 802.11 air interface.

2.2.2.9 MSDU delivery:

Networks are not much use without the ability to get the data to the recipient. Stations provide the MAC Service Data Unit (MSDU) delivery service, which is responsible for getting the data to the actual endpoint [12].

2.2.3 Wireless networks advantages:

Wireless networks offer several advantages over fixed (or "wired") networks:

2.2.3.1 Mobility:

Users move, but data is usually stored centrally. Enabling users to access data while they are in motion can lead to large productivity gains.

2.2.3.2 Ease and speed of deployment:

Many areas are difficult to wire for traditional wired LANs. Older buildings are often a problem; running cable through the walls of an older stone building to which the blueprints have been lost can be a challenge. In many places, historic preservation laws make it difficult to carry out new LAN installations in older buildings. Even in modern facilities, contracting for cable installation can be expensive and time-consuming.

2.2.3.3 Flexibility:

No cables mean no recabling. Wireless networks allow users to quickly form amorphous, small group networks for a meeting, and wireless networking makes moving between cubicles and offices a snap. Expansion with wireless networks is easy because the network medium is already everywhere. There are no cables to pull, connect, or trip over. Flexibility is the big selling point for the "hot spot" market, composed mainly of hotels, airports, train stations, libraries, and cafes.

2.2.3.4 Cost:

In some cases, costs can be reduced by using wireless technology. As an example, 802.11-equipment can be used to create a wireless bridge between two buildings. Setting up a wireless bridge requires some initial capital cost in terms of outdoor equipment, access points, and wireless interfaces. After the initial capital expenditure, however, an 802.11-based, line-of-sight network will have only a negligible recurring monthly operating cost. Over time, point-to-point wireless links are far cheaper than leasing capacity from the telephone company [13].

2.2.4 Network congestion capacity:

A fundamental problem is that all network resources are limited, including router processing time and link throughput.

For example:

- A wireless LAN is easily filled by a single personal computer
- Even on fast computer networks (e.g. Gigabit Ethernet), the backbone can easily be congested by a small number of servers and client PCs
- The aggregate transmission from P2P networks have no problem filling an uplink or some other network bottleneck

- Denial-of-service attacks by botnets are capable of filling even the largest Internet backbone network links, generating large-scale network congestion
- In telephone networks (particularly mobile phones), a mass call event can overwhelm digital telephone circuits [14].

2.2.5 Congestive collapse:

Congestive collapse (or congestion collapse) is a condition that a packet-switched computer network can reach, when little or no useful communication is happening due to congestion. Congestion collapse generally occurs at "choke points" in the network, where the total incoming traffic to a node exceeds the outgoing bandwidth. Connection points between a local area network and a wide area network are the most likely choke points.

When a network is in such a condition, it has settled (under overload) into a stable state where traffic demand is high but little useful throughput is available, and there are high levels of packet delay and loss (caused by routers discarding packets because their output queues are too full) and general quality of service is extremely poor.

Congestion collapse was identified as a possible problem as far back as 1984, for example in RFC 896, dated January 6, 1984. It was first observed on the early Internet in October 1986, when the NSF net phase-I backbone dropped three orders of magnitude from its capacity of 32 kbit/s to 40 bit/s, and this continued to occur until end nodes started implementing Van Jacobson's congestion control between 1987 and 1988.

When more packets were sent than could be handled by intermediate routers, the intermediate routers discarded many packets, expecting the end points of the network to retransmit the information. However, early TCP implementations had very bad retransmission behavior. When this packet loss occurred, the end points sent extra packets that repeated the information lost, doubling the data rate sent, exactly the opposite of what should be done during congestion. This pushed the entire network into a 'congestion collapse' where most packets were lost and the resultant throughput was negligible [15].

2.3 Congestion Control:

Congestion control concerns controlling traffic entry into a telecommunications network, so as to avoid congestive collapse by attempting to avoid oversubscription of any of the processing or link capabilities of the intermediate nodes and networks and taking resource reducing steps, such as reducing the rate of sending packets. It should not be confused with flow control, which prevents the sender from overwhelming the receiver.

2.3.1 TCP/IP congestion avoidance

The TCP congestion avoidance algorithm is the primary basis for congestion control in the Internet.

Problems occur when many concurrent TCP flows are experiencing port queue buffer tail-drops. Then TCP's automatic congestion avoidance is not enough. All flows that experience port queue buffer tail-drop will begin a TCP retrain at the same moment this is called TCP global synchronization.

2.3.2 Active queue management (AQM)

Active queue management (AQM) is the reorder or drop of network packets inside a transmit buffer that is associated with a network interface controller (NIC). This task is performed by the network scheduler, which for this purpose uses various algorithms described below [16].

2.3.2 .1 Drop Tail:

A traditional Drop Tail (DT) queue management mechanism drops the packets that arrive when the buffer is full [17]. It works on first-in-first out (FIFO) based queue of limited size, which simply drops any incoming packets when the queue becomes full [18]. Its main advantages are simplicity, suitability to heterogeneity and its decentralized nature. However this approach has some serious disadvantages, such as lack of fairness, no protection against the misbehaving or non responsive flows and no relative Quality of Service [19].

The traditional technique for managing router queue lengths is to set a maximum length (in terms of packets) for each queue, accept packets for the queue until the maximum length is reached, then reject (drop) subsequent incoming packets until the queue decreases because a packet from the queue has been transmitted. This technique is known as ‘drop tail’, since the packet that arrived most recently (i.e., the one on the tail of the queue) is dropped when the queue is full. This method has served the Internet well for years, but it has two important drawbacks

1. Lock-Out: In some situations drop tail allows a single connection or a few flows to monopolize queue space, preventing other connections from getting room in the queue. This "lock-out" phenomenon is often the result of synchronization or other timing effects.

2. Full Queues: The drop tail discipline allows queues to maintain a full (or, almost full) status for long periods of time, since tail drop signals congestion (via a packet drop) only when the queue has become full. It is important to reduce the steady state queue size, and this is perhaps queue management's most important goal. In short, drop tail is effectively ‘no management'. As the demand on networks increased, the amount of data being passed through links and hubs could no longer be unmanaged [20].

2.3.2.2 DECbitAlgorithm:

DECbit is one of the earliest examples used to control the congestion at routers. The bit in packet header to control congestion in this mechanism is known as congestion indication bit and it is used to provide feedback to the sources for controlling flow of traffic accordingly. In this mechanism, routers set congestion indication bit in arriving packet headers when mean queue length (MQL) exceeds value of 1 [18].

The main disadvantages of this scheme are averaging queue size for fairly short periods of time and no difference between congestion detection and indication [19].

2.3.2.3 Random Early Detection Algorithm:

Random early detection algorithm (RED) solves congestion problem in packet-switched networks. RED can prevent congestion by controlling the average queue size. The strategy it adopted is dropping packets timely and ensuring that there will be always a buffer available for an incoming packet [21]. This drop probability is depending on running average queue size (qa). If the average queue size is between minimum threshold (*minth*) and maximum threshold (*maxth*) the packet is marked or dropped with some probability, pa. If qa > *maxth*, then the packet is dropped. If qa<*maxth*, then the packet is forwarded through [18]. Longer-lived congestion is reflected by an increase in the computed average queue size and result in randomized feedback to some of theconnections to decrease their windows [12]. One of RED's main weaknesses is that the average queue size varies with the level of congestion and with the parameter settings. A second, related weakness of RED is that the throughput is also sensitive to the traffic load and to RED parameters. A badly configured RED will not do better than DT [19].

2.3.2.4 BLUE Algorithm:

The BLUE algorithm resolves some of the problems of RED by employing the use of a hybrid flow control scheme along with a queue size congestion measuring scheme. It uses flow and queue events to modify the congestion notification rate . This rate is adjusted by two factors: packet loss from queue congestion and link utilization or underutilization . A key difference the BLUE algorithm has from RED is that uses packet loss than the average queue length [20].

BLUE maintains a single probability, pm, which it uses to drop packets when they are queued. If the queue is continually dropping packets due to buffer overflow, BLUE increments pm, thus increasing the rate at which it sends back congestion notification. Conversely, if the queue becomes empty or if the link is idle, BLUE decreases its marking probability. This effectively allows BLUE to “learn” the correct rate it needs to send back congestion notification. The typical parameters of BLUE are (d1, d2) and freeze time. The first is freeze-time. This parameter determines the minimum time interval between two successive updates of pm. This allows the changes in the marking probability to take effect before the value is updated again. The other parameters used, (d1and d2), determine the amount by which Pm is incremented when the queue overflows or is decremented when the link is idle [22].

Here are some problems with BLUE:

• BLUE uses a hashing function to discover the non-responsive flows. This depends on the assumption that non-responsive flows would not be very large in number. We know that this is true but there could be cases when this assumption does not hold true.

• When we have large number of non-responsive flows, they could pollute the bins and TCP flows could be mistaken to be non-responsive, resulting in needlessly penalizing them.

• One solution to this could be that we change the hash functions during regular intervals of time. This would map some responsive flows to unpolluted bins.

• Another problem is that once a flow is marked, it is tainted forever. If later the flow restrains itself, BLUE still tries to reduce its sending rate through packet drops [20].

2.3.2.5 Adaptive Virtual Queue:

The motivation behind the AVQ algorithm is to design an AQM scheme that results in a low-loss, low-delay and high utilization operation at the link and whose buffer size is equal to the buffer size of the real queue.

The AVQ algorithm maintains a virtual queue whose capacity (called *virtual capacity*) is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect the new arrival. Packets in the real queue are marked/dropped when the virtual buffer over flow.

Features of the AVQ scheme:

1. The implementation complexity of the AVQ scheme is comparable to RED. RED performs averaging of the queue length, dropping probability computation and random number generation to make drop decisions. We replace these with the virtual capacity calculation in AVQ.
2. AVQ is a primarily a rate-based marking, as opposed to queue length or average queue length based marking.

3. Regulate utilization this is more robust to the presence of extremely short flows or variability in the number of long flows in the network.

4. There are two parameters that have to be chosen to implement AVQ: Both the parameters and determine the stability of the AVQ algorithm and we provide a simple design rule to choose these parameters [23].

CHAPTER THREE

ADAPTIVE VIRTUAL QUEUE

Chapter Three: Adaptive Virtual Queue

3.1 Methodology:

Adaptive virtual Queue schemes have been recently proposed for Active Queue Management (AQM) in Internet routers. A particular consideration scheme was chosen, Adaptive Virtual Queue (AVQ), and a study of its properties: its stability in the presence of feedback delays, its ability to maintain small queue lengths, and its robustness in the presence of extremely short flows (the so-called web mice). Using a linearized model of the system dynamics, we present a simple rule to design the parameters of the AVQ algorithm. We then compare its performance through simulation with several well-known AQM schemes such as RED, Proportional Integral (PI) controller, and a no adaptive virtual queue algorithm. With a view toward implementation, we show that AVQ can be implemented as a simple token bucket using only a few lines of code.

3.2 Active Queue Management:

In Internet routers, **active queue management** (**AQM**) is the intelligent drop of network packets inside a buffer associated with a network interface controller (NIC), when that buffer becomes full or gets close to becoming full, often with the larger goal of reducing network congestion. This task is performed by the network scheduler, which for this purpose uses various algorithms such as random early detection (RED), Explicit Congestion Notification (ECN), or controlled delay (CoDel)[24].

3.2.1 Queue Management:

An Internet router typically maintains a set of queues, one per interface that holds packets scheduled to go out on that interface. Historically, such queues use a *drop-tail* discipline: a packet is put onto the queue if the queue is shorter than its maximum size (measured in packets or in bytes), and dropped otherwise [25].

Active queue disciplines drop or mark packets before the queue is full. Typically, they operate by maintaining one or more drop/mark probabilities, and probabilistically dropping or marking packets even when the queue is short [26].

3.2.2 Benefits of AQM:

Drop-tail queues have a tendency to penalize bursty flows, and to cause global synchronization between flows. By dropping packets probabilistically, AQM disciplines typically avoid both of these issues [27].

By providing endpoints with congestion indication before the queue is full, AQM disciplines are able to maintain a shorter queue length than drop-tail queues, which combats buffer bloat and reduces network latency [28].

3.2.3 Drawbacks of AQM:

Early AQM disciplines require careful tuning of their parameters in order to provide good performance. Modern AQM disciplines (ARED, Blue, PI) are self-tuning, and can be run with their default parameters in most circumstances. [29]

For AQM systems that drop packets (rather than using ECN marking), the result seems counter-intuitive to many network engineers: "Why should I drop perfectly good packets when I still have free buffer space?" What they fail to think about is that the packets will have to be dropped once the buffer is full. (Full buffers cause excessive latency)[30].

3.2.4 Active Queue Management Algorithms:

- ❖ Random early detection (RED).
- ❖ Random Exponential Marking (REM).
- ❖ Blue and Stochastic Fair Blue (SFB).
- ❖ PI controller.
- ❖ Robust random early detection (RRED).
- ❖ RSFB: a Resilient Stochastic Fair Blue algorithm against spoofing DDoS attacks.
- ❖ RED with Preferential Dropping (RED-PD).
- ❖ Controlled Delay (CoDel).

3.3 Random early detection:

Random early detection (RED), also known as randomearly discard or random early drop is an active queue management algorithm.

It is also a congestion avoidance algorithm. In the conventional tail drop algorithm, a router or other network component buffers as many packets as it can, and simply drops the ones it cannot buffer. If buffers are constantly full, the network is congested. Tail drop distributes buffer space unfairly among traffic flows [31].

Tail drop can also lead to TCP global synchronization as all TCP connections "hold back" simultaneously, and then step forward simultaneously. Networks become under-utilized and flooded by turns. RED addresses these issues[32].

3.3.1 Operation:

RED monitors the average queue size and drops (or marks when used in conjunction with ECN) packets based on statistical probabilities. If the buffer is almost empty, all incoming packets are accepted. As the queue grows, the probability for dropping an incoming packet grows too. When the buffer is full, the probability has reached 1 and all incoming packets are dropped.

RED is more fair than tail drop, in the sense that it does not possess a bias against bursty traffic that uses only a small portion of the bandwidth. The more a host transmits, the more likely it is that its packets are dropped as the probability of a host's packet being dropped is proportional to the amount of data it has in a queue. Early detection helps avoid TCP global synchronization[31].

3.3.2 Problems with Classic RED:

RED one of the most well known AQMs is difficult to configure and does not provide significant performance gains given the complexity required for proper configuration. Recent variants of RED, such as Adaptive-RED are designed to provide more robust RED performance under a wider-range of traffic conditions but have not yet been evaluated. Pure RED does not accommodate quality of service (QoS) differentiation.[Analysis of Active Queue Management].

3.4 PID controller:

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) commonly used in industrial control systems. A PID controller continuously calculates an "error value" as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error over time by adjustment of control variable such as the position of a control valve, a damper, or the power supplied to a heating element, to a new value determined by a weighted sum:

$$u(t) = K_p e(t) + K_i \int_0^t e(T)\, dT + K_d \frac{de}{dt} \quad \ldots\ldots\ldots\ldots(3.1)$$

WhereK_p,K_i andK_d, all non-negative, denote the coefficients for the proportional, integral, and derivative terms, respectively (sometimes denoted *P*, *I*, and *D*). In this model, *P* accounts for present values of the error (e.g. if the error is large and positive, the control output will also be large and positive), *I* accounts for past values of the error (e.g. if the output is not sufficient to reduce the size of the error, error will accumulate over time, causing the controller to apply stronger output), and *D* accounts for predicted future values of the error, based on its current rate of change [33].

As a PID controller relies only on the measured process variable, not on knowledge of the underlying process, it is a broadly useful controller [2]. By tuning the three parameters of the model, one can design a PID controller for specific process requirements [34]. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point, and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability [35].

Some applications may require using only one or two terms to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value due to the control action [36].

The early history of automatic process control the PID controller was implemented as a mechanical device. These mechanical controllers used a lever, spring and a mass and were often energized by compressed air. These pneumatic controllers were once the industry standard[37].

Electronic analog controllers can be made from a solid-state or tube amplifier, a capacitor and a resistor. Electronic analog PID control loops were often found within more complex electronic systems, for example, the head positioning of a disk drive, the power conditioning of a power supply, or even the movement-detection circuit of a modern seismometer. Nowadays, electronic controllers have largely been replaced by digital controllers implemented with microcontrollers or FPGAs. However, analog PID controllers are still used in niche applications requiring high-bandwidth and low noise performance, such as laser diode controllers [38].

3.5 AVQ Algorithm:

In the modern day Internet, there has been a strong demand for QoS and fairness among flows. As a result, in addition to the sources, the links are also forced to play an active role in congestion control and avoidance. Random Early Discard (RED) was originally proposed to achieve fairness among sources with different burstiness and to control queue lengths. RED allows for dropping packets overflow. Another form of congestion notification that has been discussed since the advent of RED is Explicit Congestion Notification (ECN). ECN has been proposed to allow links to help in congestion control by notifying user when it detects an onset of congestion. The links on detecting incipient congestion set a bit in the packet header that notifies the user that a link on its route is experiencing congestion. The user then reacts to the mark as if a packet has been lost. Thus, the link avoids dropping the packet (thereby enhancing good put) and still manages to convey Congestion information to the user [3].

To provide ECN marks or drop packets to provide fairness and control queue lengths, the routers have to select packets intelligently in a manner that conveys information about the current state of the network to the users. Algorithms which the routers employ to convey such information are called Active Queue Management (AQM) schemes. An AQM scheme might mark or drop packets depending on the policy at the router. In this research, we use the term "marking" more generally to refer to any action taken by the router to notify the user of incipient congestion. The action can, in reality, be ECN-type marking or dropping (as in RED) depending upon the policy set for the router. As in earlier work on studying AQM schemes [13, 6, 5], this distinction is blurred in the mathematical analysis to allow for the development of simple design rules for the choice of AQM parameters. However, our simulations consider marking and dropping schemes separately. Designing robust AQM schemes have been a very active research area in the Internet community [39], [40].

Some AQM schemes that have been proposed include RED a virtual queue based scheme where the virtual capacity is adapted, ,Blue, Proportional Integral (PI) controller , REM, a virtual queue based AQM scheme (which we refer to as the Gibbens-Kelly Virtual Queue, or the GKVQ scheme) among others. While most of the AQM schemes proposed detect congestion based on the queue lengths at the link (e.g., RED),some AQM schemes detect congestion based on the arrival rate of the packets at the link (e.g., virtual queue-based schemes) and some use a combination of both (e.g., I). Also, most of the AQM schemes involve adapting the marking probability (as noted before we use the term marking to refer to both marking and dropping) in some way or the other. An important question is how fast should one adapt while maintaining the stability of the system? Here the system refers jointly to the TCP congestion controllers operating at the edges of the network and the AQM schemes operating in the interior of the network. Adapting too fast might make the system respond well to changing network conditions, but it might lead to large oscillatory behavior or in the worst case even instability.

- Active role of links in congestion control in addition to sources to achieve QoS and Fairness
- Explicit Congestion Notification (ECN) used to notify congestion to users.
- Algorithms used to identify packets to drop or mark are called ACTIVE QUEUE MANAGEMENT (AQM) schemes.
- Some of AQM RED, SRED, BLUE, Proportional Integral (PI), REM, Virtual Queue Based AQM scheme (also called as Gibbens-Kelly Virtual Queue (GKVQ)

Router maintains virtual queue whose capacity is less than or equal to capacity of link ($\check{C}$).

Buffer size of virtual queue is equal to real queue buffer size.

At arrival of each packet virtual capacity is calculated

$\check{C} = \alpha (\gamma C - \lambda)$…………………(3.2)

α is smoothing factor, γ is desired utilization and λ is arrival rate.

3.5.1 Parameters of adaptive virtual queue:

B = buffer size (physical queue size)

s = the arrival time of the previous packet

t = the current time (i.e., the arrival time of the current packet)

b = the size of the current packet in bytes

VQ = the current size of the *virtual queue* in bytes

3.5.2 AVQ Features:

- AVQ implementation complexity is comparable to RED.
- AVQ is rate-based marking as opposed to average queue length marking.
- AVQ regulates link utilization instead of queue length unlike in any AQM.
- Two parameters are chosen to implement AVQ (γ desired utilization and α damping factor) [3], [40], [41].

3.6 Comparison between different AQM schemes:

3.6.1 Packet Losses and Link Utilization:

The losses incurred byall the schemes are shown in Figure [3.1] as a function ofthe number of FTP flows. The A VQ scheme has fewerlosses than any other scheme.Figure [3.2] shows the utilization of the link for some of the AQMschemes. RED also results in a poor utilization of the link. Our observation has been that when increasing the utilization of RED (by tuning its parameters),the packet losses at the link also increases. PI has a utilization of0.99 as the queue is always non-empty. For the AVQ scheme, we required a desired utilization of 0.98 and we can see that the AVQ scheme tracks the desiredutilization quite well. Thus, the main conclusion from this experiment is that the AVQ achieves low loss with high utilization. [3]

3.6.2 Responsiveness to changing network conditions:

The objective of this section is to measure the response of eachAQM scheme when the number of flows is increased. Twenty new FTP users are added every 100s till the total number of FTPconnections reach 180 and the average queuelength over every 100s is computed. Schemes that have along transient period will have an increasing average queuelength as new users are added before the scheme is able toconverge. The average queue length (over each 100 secondinterval) of each scheme as the number of users increase isshown in Figure [3.3]. We see from the figure that PI has higher average queue lengths than the desired queue length. On the other hand, AVQ and RED havesmaller queue sizes. This is due to the fact that PI apparently has a long transient period and new usersare added before the queue length converges. The average queue length over each 100s interval is used to capturepersistent transients in this experiment for studying the responsiveness of the AQM schemes to load changes. This figure shows that AVQ is responsive to changes innetwork load and is able to maintain a small queue lengtheven when the network load keeps increasing. [3]

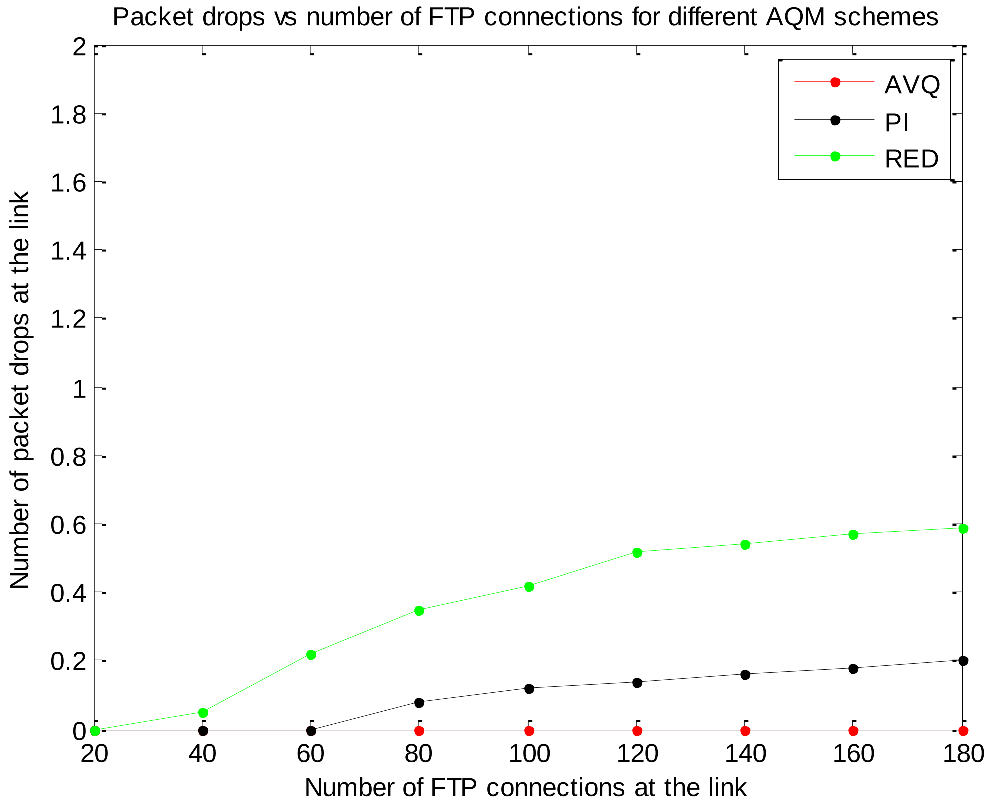


Figure 3.1 Packet Drop at the link vs. FTP Connections

Table 3.1 illustrate the number ofPacket Drop at the link vs. FTP Connections

no of FTB CONNECTION	Dropped (AVQ)	Dropped (PI)	Dropped (RED)
20	0	0	0
40	0	0	0.05
60	0	0	0.22
80	0	0.08	0.35
100	0	0.12	0.42
120	0	0.14	0.52
140	0	0.16	0.54
160	0	0.18	0.57
180	0	0.2	0.59

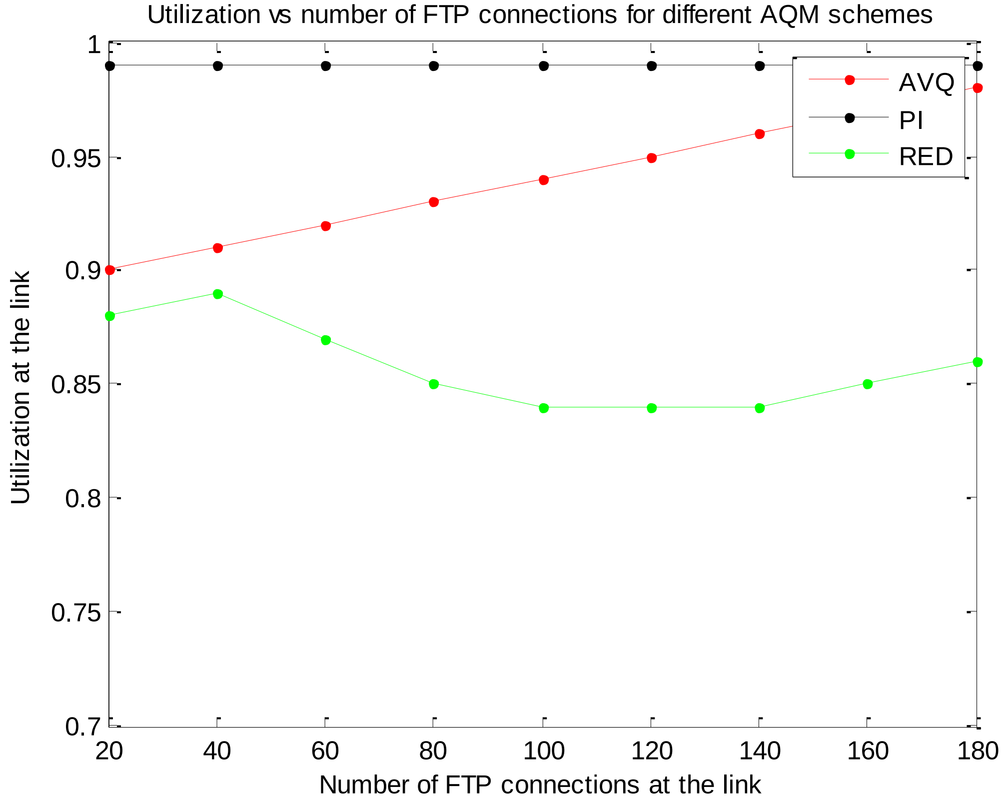


Figure3.2 Utilization at the link vs. FTP Connections

Table 3.2 illustrate the Utilization at the link vs. FTP Connections

No of FTP connection	Link Utilization(AVQ)	Link Utilization(PI)	Link Utilization(RED)
20	0.9	0.99	0.88
40	0.91	0.99	0.89
60	0.92	0.99	0.87
80	0.93	0.99	0.85
100	0.94	0.99	0.84
120	0.95	0.99	0.84
140	0.96	0.99	0.84
160	0.97	0.99	0.85
180	0.98	0.99	0.86

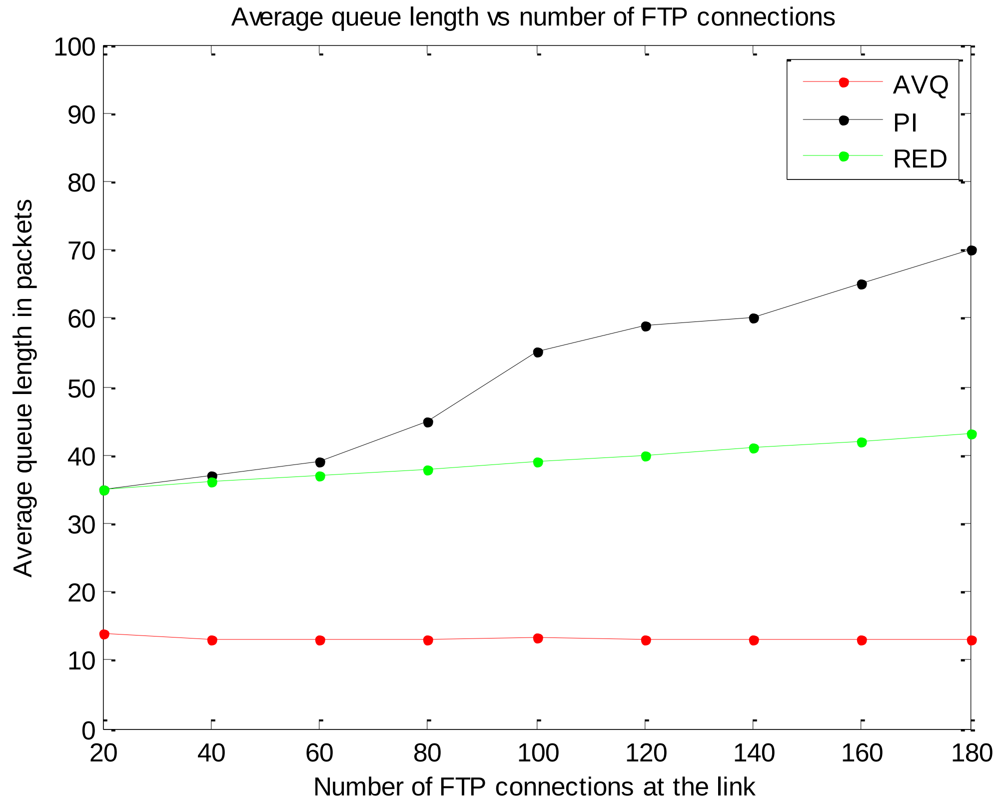


Figure 3.3:Average queue length vs. FTP Connections.

Table 3.3 illustrates the Average queue length vs. FTP Connections.

No of FTB connection	Average queue length(AVQ)	Average queue length(PI)	Average queue length(RED)
20	14	35	35
40	13	36	37
60	13.1	37	39
80	13	38	45
100	13.2	39	55
120	13.1	40	59
140	13	41	60
160	13	42	65
180	13.01	43	70

CHAPTERFOUR

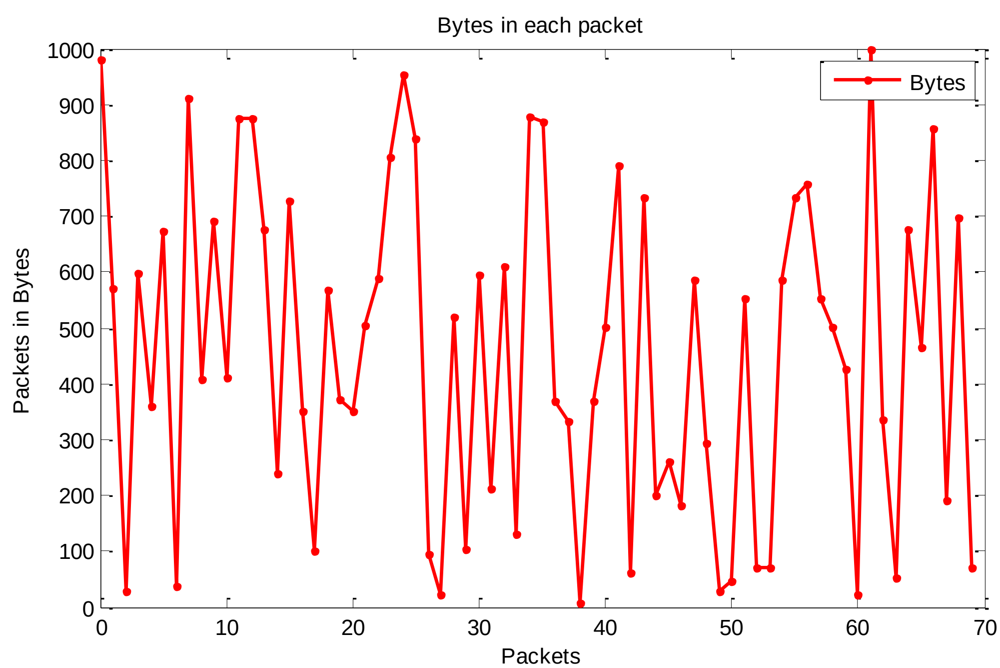
RESULT & DISCUSSION

Chapter Four: Result & Discussion

4.1 AVQ simulation:

4.1.1 Bytes in Packets:

The following figure illustrate the bytes per each packet inserted in the queue the x axis illustrates the packets it was configured 150 packet and the y axes illustrates the packets in bytes.



Figure(4.1) AVQ number of bytes in each received packet

4.1.2 Total number of packets:

The following figure illustrates the packets along with the size in bar chart so the end user can see the difference in size.

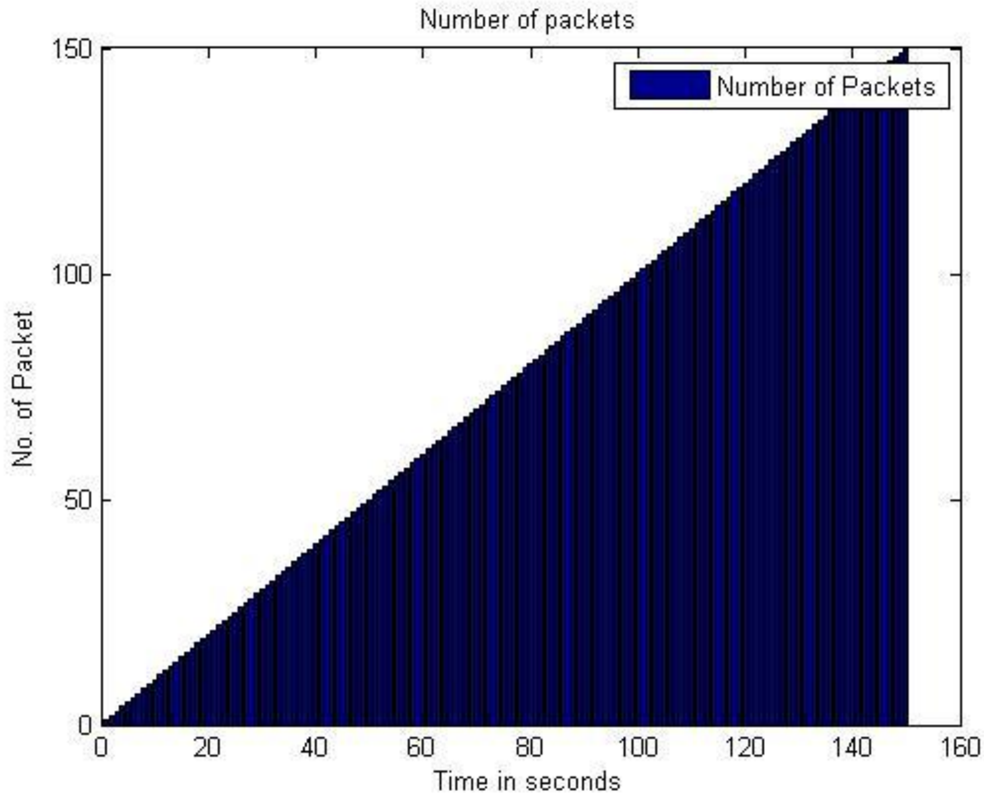


Figure (4.2) noof packets

4.1.3 AVQ with fixed packet number and different bandwidth values:

Table (4.1) illustrate the simulation of AVQ using a different values of Bandwidth and fixed number of packets it was found that increasing bandwidth reduces the packet loss and the bandwidth was set to 100 and increased each time by 100 to reach 5000 and the optimal value of bandwidth during the simulation was found 4000 with a packet loss equals zero.

Table 4.1 Packet Drops in different bandwidth configurations and fixed noof packets

No.	No. of Packets	Bandwidth	Drops
1	150	100	148
2	150	200	145
3	150	300	143
4	150	400	132
5	150	500	127
6	150	600	130
7	150	700	130
8	150	800	112
9	150	900	115
10	150	1000	117
11	150	1100	98
12	150	1200	109
13	150	1300	101
14	150	1400	99
15	150	1500	94
16	150	1600	94
17	150	1700	91
18	150	1800	91
19	150	1900	76

20	150	2000	74
21	150	2100	77
22	150	2200	83
23	150	2300	62
24	150	2400	55
25	150	2500	61
26	150	2600	50
27	150	2700	53
28	150	2800	41
29	150	2900	36
30	150	3000	53
31	150	3100	41
32	150	3200	21
33	150	3300	27
34	150	3400	21
35	150	3500	19
36	150	3600	20
37	150	3700	15
38	150	3800	8
39	150	3900	3
40	150	4000	0

The following figure illustrate the packet drop in different bandwidth configuration, it was found that the increasing of bandwidth decrease the packet drop and the optimum value of bandwidth was found near 4000 in the simulation while the maximum packet was set to 150 packet.

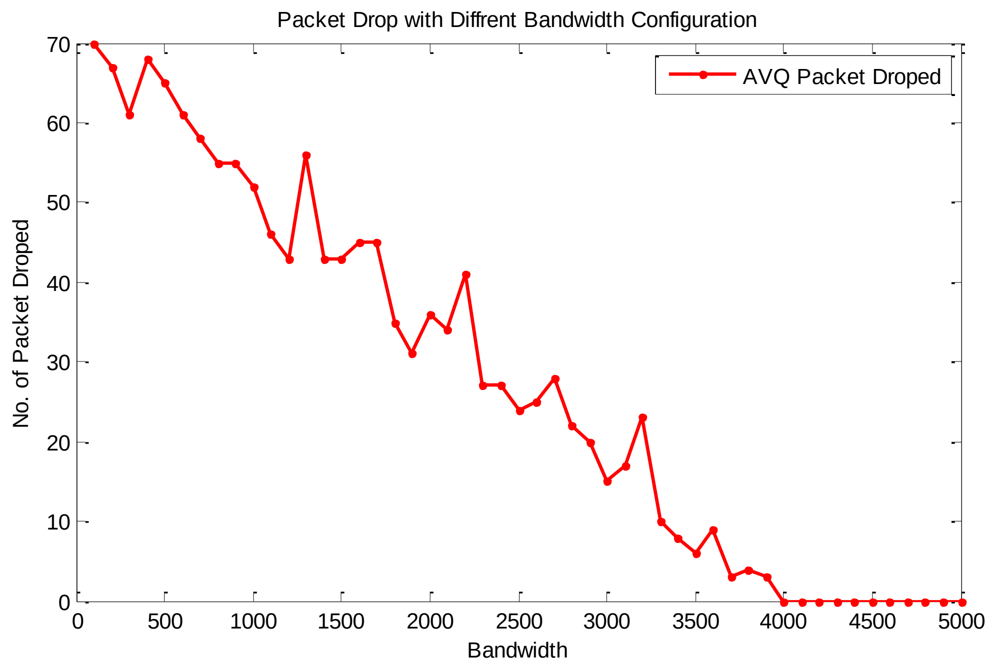


Figure (4.3) bandwidth vs. packet drop

4.1.4 AVQ with fixed Bandwidth and different packet number in the queue:

Table 4.2 illustrate the simulation of AVQ using a different values of packets and fixed bandwidth it was found that increasing packet increases the drop of packet or packet loss and the packets was set to 100 and increased each time by 100 to reach 5000 and the optimal value of bandwidth during the simulation was found 4000 with a packet loss equals zero.

Table 4.2 illustrates the drops while fixed bandwidth and variable packets and the total drops of packets.

No.	No. of Packets	Bandwidth	Drops
1	100	100	0
2	200	100	150
3	300	100	266.6667
4	400	100	375
5	500	100	480
6	600	100	583.3333
7	700	100	685.7143
8	800	100	787.5
9	900	100	888.8889
10	1000	100	990
11	1100	100	1090.909
12	1200	100	1191.667
13	1300	100	1292.308
14	1400	100	1392.857
15	1500	100	1493.333
16	1600	100	1593.75
17	1700	100	1694.118
18	1800	100	1794.444
19	1900	100	1894.737
20	2000	100	1995

21	2100	100	2095.238
22	2200	100	2195.455
23	2300	100	2295.652
24	2400	100	2395.833
25	2500	100	2496
26	2600	100	2596.154
27	2700	100	2696.296
28	2800	100	2796.429
29	2900	100	2896.552
30	3000	100	2996.667
31	3100	100	3096.774
32	3200	100	3196.875
33	3300	100	3296.97
34	3400	100	3397.059
35	3500	100	3497.143
36	3600	100	3597.222
37	3700	100	3697.297
38	3800	100	3797.368
39	3900	100	3897.436
40	4000	100	3997.5

The following figure (4.4) illustrates a fixed bandwidth while running the simulation code and a verity of packet number stated from 100 packets to 5000 packet with 500 gaps in each loop, it was found that increasing the packet number into the queue with a fixed bandwidth increase the drop.

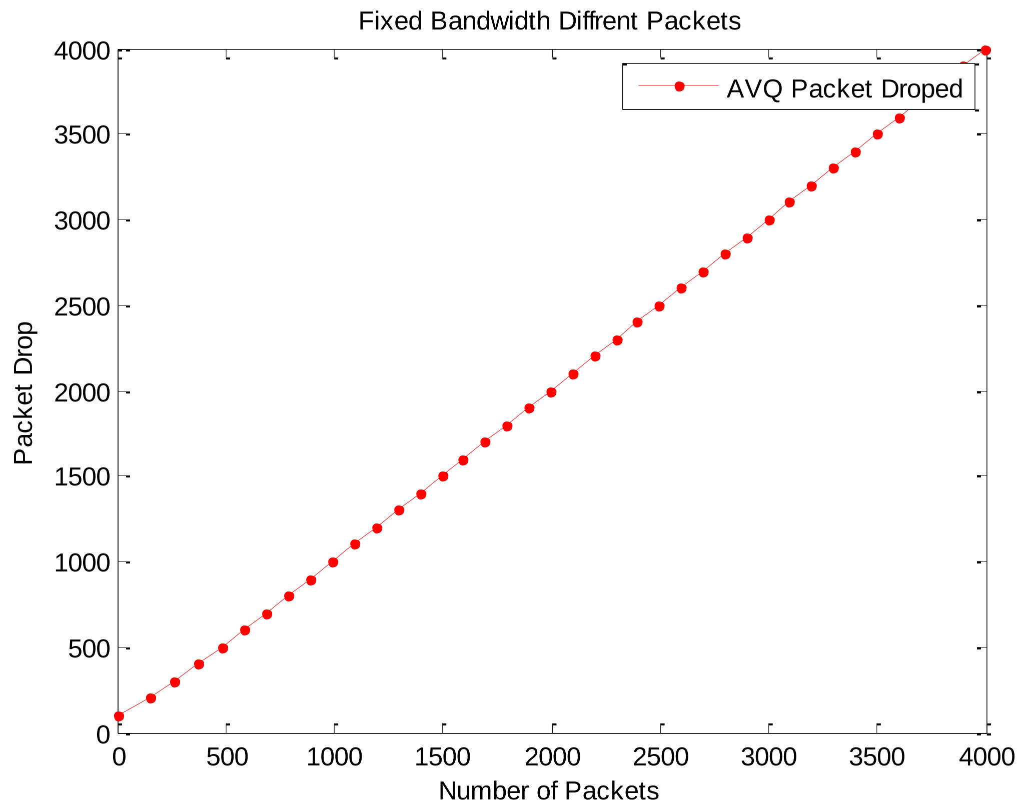


Figure (4.4) fixed bandwidth different packet in the queue

We show through simulations that the AVQ controller out performs a number of other well-known AQM schemes in terms of losses, utilization and average queue length. In particular, we show that AVQ is able to maintain a small average queue length at high utilizations with minimal loss at the routers.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

Chapter Five: Conclusion and Recommendations

5.1 Conclusion:

After study and implementation of the active Queue Management using adaptive virtual queue which can be defined as the process of anticipating congestion at the router level in an attempt to reduce the queue size and increase throughput. This can be done only if the routers can communicate network congestion states to the end users by either dropping or marking packets. This paper designs and analyzes an algorithm called the Adaptive Virtual Queue (AVQ), which addresses the problem of how to mark or drop packets in an intelligent manner so the that the current state of the network can be conveyed to the end users.

AVQ is a virtual queue that imitates a real queue. Packets can be enquired and dequeued, and they are dropped when the capacity of the virtual queue is exceeded. When a packet in the virtual queue is dropped, the corresponding packet in the real queue is either dropped or marked. The capacity of the virtual queue is less than that of the real queue, and that virtual capacity can be updated based on a desired utilization ratio (< 1), a value that represents the total flow in to the link, and a smoothing factor.

After running the simulation and obtain the results it was found that increasing the packets in the queue with a fixed bandwidth increases the packet drop while increasing the bandwidth the packet drop decreases.

The simulation code was written to handle both fixed packet number and variable and also fixed bandwidth and variable.

5.2 Recommendations:

1. We recommended that to apply more algorithms in order to detect the performance of each moreover the simulation can be done using other simulation programs such as Opnet Modular and Network Simulator II.
2. Apply more parameters to improve results.
3. Use high ultra-video traffic to detect the performance

References:

1. Nanoscale Communication Networks, Bush, S. F., ISBN 978-1-60807-003-9, Artech House, 2010.
2. *Spurgeon, Charles E. (2000). Ethernet the Definitive Guide. O'Reilly & Associates. ISBN 1-56592-660-9.*
3. Kunniyur, Srisankar S., and RayadurgamSrikant. "An adaptive virtual queue (AVQ) algorithm for active queue management." *Networking, IEEE/ACM Transactions on* 12.2 (2004): 286-299.
4. Socrates, C., PM Beulah Devamalar, and R. KannammaSridharan. "Congestion Control for Packet Switched Networks: A Survey."
5. Singh, Rakesh Kumar, and Neeraj Tiwari. "An Investigation on Wireless Mobile Network and Wireless LAN (Wi-Fi) for Performance Evaluation." *International Journal of Computer Applications* 126.6 (2015).
6. Banerji, Sourangsu, and Rahul Singha Chowdhury. "On IEEE 802.11: Wireless LAN Technology." *arXiv preprint arXiv:1307.2661* (2013).
7. Crow, Brian P., et al. "IEEE 802.11 wireless local area networks." *Communications Magazine, IEEE* 35.9 (1997): 116-126.
8. Zimányi, Esteban, Pedro Miguel Antunes, and Jean-Michel Dricot. "Design and Implementation of a Physical-Constrained Wireless Routing Algorithm in Linux."
9. Robotics, U. S. "Wireless LAN Networking White Paper." *IEEE Computer Society* (2009).
10. Singh, Raj Kumar, and Dr AK Jain. "Research issues in wireless networks." *International Journal of AdvancedResearch in Computer Science and Software Engineering* 2.4 (2012).

11. Najim, Salam A., Ibrahiem MM El Emary, and Samir M. Saied. "Performance Evaluation of Wireless IEEE 802.11 b used for E-learning Classroom Network." (2007).
12. Gast, Matthew. *802.11 wireless networks: The definitive guide*. Southeast University Press, 2006.
13. Connolly, Cornelia. *An Analysis of Queuing Mechanisms at Layer 2, for Wireless LAN's*. Diss. University of Limerick, 2002.
14. Afolabi, A. O., and S. O. Olabiyisi. "Application of Routing Algorithm to Congestion Control in GSM Network." *Proceedings of the World Congress on Engineering*. Vol. 2. 2012.
15. Priyadharshini, S., and J. SathyaPriya. "Counter Measure for Congestion in TCP/IP using Bandwidth Estimation."
16. Marsic, Ivan. "Computer networks: Performance and quality of service." (2010).
17. Matthew, Gast. "802.11 Wireless networks: The definitive guide." O'ReillyNetworking (2002).
18. Hashmi, Syed Shakeel. "Congestion Avoidance in IP Based CDMA Radio Access Network." International Journal of Wireless & Mobile Networks 3.1 (2011).
19. Weng, Chien-Erh, Ruei-Yi Li, and Chao-Hsing Hsu. "joint priority random early detection and ieee 802.11 e edcf to support qos in wlan." Published by International Journal of Innovative Computing, Information and Control Volume 8 (2012): 5425-5438.
20. Haider, Aun, et al. "Congestion control algorithms in high speed telecommunication networks." ORSNZ Conference Twenty Naught One. Vol. 30. 2003.

21. Behrouz, A. Forouzan, and C. F. Sophia. "Data Communications and Networking." McGraw Hill (2003).
22. Abdel-Jaber, Hussein, et al. "Performance investigations of some active queue management techniques using simulation." International Journal of New Computer Architectures and their Applications (IJNCAA) 2.1 (2012): 286-301.
23. Feng, Wu-chang, et al. "BLUE: A new class of active queue management algorithms." Ann Arbor 1001 (1999): 48105.
24. Key, Peter, et al. "Congestion pricing for congestion avoidance." *Microsoft Research, Cambridge, UK, Tech. Rep. MSR-TR-99-15* (1999).
25. Mahida, P. T., et al. "A Comparative Analysis of Queue Management Techniques using NS-2 Simulator." *International Journal of Computer Applications* 65.6 (2013).
26. Sun, Jinsheng, and Moshe Zukerman. "RaQ: A robust active queue management scheme based on rate and queue length." *Computer Communications* 30.8 (2007): 1731-1741.
27. Olawoyin, L., N. Faruk, and L. Akanbi. "Queue management in network performance analysis." *International Journal of Science and Technology* (2011).
28. Liu, Shao, Tamer Başar, and Ravi Srikant. "TCP-Illinois: A loss-and delay-based congestion control algorithm for high-speed networks." *Performance Evaluation* 65.6 (2008): 417-440.
29. Ryu, Seungwan, Christopher Rump, and ChunmingQiao. "Advances in active queue management (AQM) based TCP congestion control." *Telecommunication Systems* 25.3-4 (2004): 317-351.

30. Le, Long, et al. "The effects of active queue management and explicit congestion notification on web performance." *IEEE/ACM Transactions on Networking (TON)* 15.6 (2007): 1217-1230.
31. Patil, Ganesh, Sally McClean, and Gaurav Raina. "Drop tail and RED queue management with small buffers: stability and Hopf bifurcation." *ICTACT Journal on Communication Technology: Special Issue on Next Generation Wireless Networks and Applications* 2.2 (2011): 339-344.
32. Floyd, Sally, and Van Jacobson. "Random early detection gateways for congestion avoidance." *Networking, IEEE/ACM Transactions on* 1.4 (1993): 397-413.
33. Astrom, Karl J. "PID controllers: theory, design and tuning." *Instrument society of America* (1995).
34. Mange, J., S. Pace, and A. Dunn. "Optimization of PID Controller Parameters for Automated Ground Vehicle Control on Dynamic Terrain." *Proceedings of the International Conference on Scientific Computing (CSC)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.
35. Krass, Matt. "PID Control Theory." *Matt Krass* 2 (2006).
36. Åström, Karl Johan, and Tore Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.
37. Mishra, Anurag. *a study on pid controller design for systems with time delay*. Diss. National Institute of Technology Rourkela, 2011.
38. Gamasu, Ramesh, and Venkata Ramesh BabuJasti. "Robust Cohen-Coon PID Controller for Flexibility of Double Link Manipulator." *International Journal of Control and Automation* 7.1 (2014): 357-368.

39. Briscoe, Bob, Gabriele Corliano, and Ben Strulo. *How to Build a Virtual Queue from Two Leaky Buckets (and why one is not enough)*. BT, Technical Report TR-DES8-2011-001, Apr. 2012.[Online]. Available:

40. Kunniyur, Srisankar S., and RayadurgamSrikant. "An adaptive virtual queue (AVQ) algorithm for active queue management." *Networking, IEEE/ACM Transactions on* 12.2 (2004): 286-299.

41. Marbach, Peter, and Yiyi Lu. "Active queue management and scheduling for wireless networks: the single-cell case." *Proc. CISS*. 2006.