Chapter - 1

INTRODUCTION
INTRODUCTION

1.1 General

Mankind depends on using information, which originates from a variety of forms like speech, written and printed documents etc. The inventions of telegraph by Wheatstone and Morse in 1837 and later in 1876, telephone by Alexander Graham Bell led to the far-reaching developments in transferring information from end to end. The information can be processed, stored and transported and technologies have been developed to perform all these functions more commonly converting into electrical signals and hence transmitting the same over a distance known as 'Telecommunications'. The twentieth century witnessed for the key technology of information gathering, processing, and distributing electronically, this led to development of telephone networks, computer industry and satellite communications.

Communication Technology is attracting a widespread attention by every individual in the modern world. Over the last few decades, there has been a remarkable achievement in the field of communication and communication networks all over the world. This includes the home banking, videophone, voice recognition, cell phones, messaging, internet services etc. Many organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As the ability to gather, process, and
distribute information grows, the demand for more sophisticated information processing grows even faster. This is achieved by transmission and reception of information's through the process of switching technology [1].

Switching technology has acquired a good momentum to provide such dedicated line for communication of voice and data. Communication field, hence switching technology is changing everyday to meet challenges and it is biggest revolution in these days. The scope of the issue is analysis of high-performance switches, and their influence on the operation and functionality of network infrastructures. Most widely used switching systems are circuit switching and packet switching [2].

The existing circuit switching techniques are ideal for communication that requires data to be transmitted in real time where the time cost is involved in establishing a connection. In this technique, Time Division Multiplexing and Frequency Division Multiplexing (TDM and FDM) resources remain allocated even if no data is flowing on a circuit. This major issue in circuit switching causes wasting of link capacity when a circuit does not carry as much traffic as the allocation permits [3].

The Packet switching technique is the now-dominant communications paradigm [4] which is used to optimize the use of bandwidth available in the network, to minimize the transmission latency and to increase robustness of communication. Packet switching mechanism determines how network resources are allocated for data transmission, i.e., how and when the input channels are connected to the output channel selected by the routing algorithm as shown in Fig. 1.1. It is the actual mechanism that removes data from input channels and places them on output channels [5].
Packet switching refers to protocols in which messages are broken up into small packets before they are sent. Each packet is transmitted individually across the net. The packets may even follow different routes to the destination. Thus, each packet has header information which enables to route the packet to its destination as shown in Fig. 1.2. At the destination the packets are reassembled into the original message. Due to its superior flexibility and resource usage, the majority of today’s networks are based on packet switching technology. In view of this, in the present work more emphasis is given on the packet switching techniques.
As the Internet evolves into a global commercial infrastructure, it is expected to support a surplus of new applications such as Internet Protocol (IP) telephony, interactive TV, and e-commerce, causes transient congestion leads to low throughput, high latency and high packet loss in packet switching technique. As a consequence, there is a need to provide more powerful services for transient congestion to achieve high throughput (total output traffic rate/input traffic rate), low latency and less packet loss and there has been an intense research towards achieving this goal [6-7].

1.2 Congestion

Congestion is the aggregate demand for bandwidth and exceeds the available capacity of a link. A network is considered congested when too many packets try to access the same router’s buffer, resulting in an amount of packets being dropped. During congestion, actions need to be taken by both the transmission protocols and the network routers in order to avoid a congestion collapse and furthermore to ensure network stability, throughput and fair resource allocation to network users [8].

The Basic causes of Congestion are:

- The input traffic demand exceeds the capacity of the network.
- In typical packet switching network, this can occur quite easily when:
  - Output links are slower than inputs.
  - Multiple traffic sources competing for same output link at the same time.
- High queuing delays.
- Congestion collapse.

There are two approaches of Handling Congestion:

- Congestion Control
- Congestion Avoidance
1.3 Difference between Congestion Control and Congestion Avoidance

- Congestion Control plays after the network is overloaded, whereas Congestion Avoidance plays before the network is overloaded.

- A strategy to reduce the impact of overload in a network is called Congestion Control. It is primarily concerned with controlling the traffic to reduce overload on the network. A Congestion Avoidance strategy will have two components: the network must be able to signal the transport endpoints that congestion is occurring and the endpoints must have a policy that decreases utilization if this signal is received and increases utilization if the signal is not received.

- Congestion control detects when the network has entered the early stages of congestion collapse and recovers the network, but not before some packet loss has occurred. Congestion avoidance keeps the network operating just on the cusp of collapse, but does not drive it into this state.

- The purpose of a congestion control scheme [9] is to detect the fact that the network has reached the cliff, resulting in packet losses, and to reduce the load so that the network can recover to an uncongested state. Congestion avoidance, on the other hand, operates the network at a highly desirable point, which is at the knee of the response time (or delay) curve. This is the point beyond which the increase in throughput is small, but the response time increases rapidly with load. This enables the network to reduce significantly the probability of packet loss and prevents the possibility of serious congestion developing and impacting user performance in the network.
Examples of Congestion Control Algorithms are Leaky bucket, token bucket, RSVP. Examples of Congestion Avoidance algorithms are Drop Tail, RED, DRR, FQ, SFQ, and CSFQ.

All the techniques mentioned above, explains the efforts made by congestion control and congestion avoidance techniques. Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottleneck, and it is achieved through packet dropping. In view of these, it is essential to avoid the congestion instead of considering more effort for the congestion control [10]. Hence the present work is focused on the congestion avoidance techniques. Some of the congestion avoidance mechanisms are:

- Drop Tail (DT)
- Random Early Detection (RED)
- Fair Queuing (FQ)
- Stochastic Fair Queuing (SFQ)
- Core-Stateless Fair Queuing (CSFQ).

1.3.1 Drop Tail

Drop Tail is typically the default congestion avoidance mechanism, and it is First-In First-Out (FIFO) default congestion management queuing method. There is no priority assigned to traffic flows in Tail Drop. Drop Tail treats all traffic equally and does not differentiate between classes of service. Once the output queue or buffer on an outbound interface is full, tail drop simply drops any additional packets until the buffer is no longer at maximum capacity.
The drawback to utilizing Tail Drop occurs when multiple Transmission Control Protocol (TCP) sessions are affected by the packet discarding. Due to the send/acknowledgement behavior of TCP, the TCP senders will slow their transmission rate by half when they discover a drop. This is known as TCP slow start. When drops occur during multiple, simultaneous TCP flows, the resulting slow-start state will decelerate traffic severely, which underutilizes the available bandwidth. The traffic flows then increase their transmit rate until the outbound queue limit is attained again and prompts another slow-start state. This wave of traffic flow is known as global synchronization, a poor use of available bandwidth [11].

1.3.2 Random Early Detection

In Random Early Detection (RED) packets are dropped at random when the queue increases, which provides early congestion indication to flows which can then gracefully discards the packets before the buffer is overloaded. RED maintains two buffer thresholds. When the buffer occupancy is smaller than the first threshold, no packet is dropped, when the buffer occupancy is larger than the second threshold all packets dropping probability increases linearly with buffer occupancy. This algorithm greatly alleviates the problem of segregation. By keeping the average buffer occupancy small, it reduces the delays of most packets [12].

1.3.3 Fair Queuing

Demers, Keshav and Shenker proposed a mechanism called Fair Queuing (FQ) [13]. It uses Nagle's idea [30] of creating separate queues for the packets from each individual source, generally forwarding packets from different sources in Round Robin fashion [71],
and selectively dropping packets from high-volume sources in order to fairly allocate bandwidth among sources. If a source sends packets too quickly, it makes the length of its own queue grow and more packets in the queue will be dropped. This is because in per-flow queuing, packets belonging to different flows are isolated from each other and one flow cannot have much impact on another [14].

1.3.4 Stochastic Fair Queuing

A proposal called Stochastic Fair Queuing (SFQ) has been made to economize the computations by hashing, which is used to map packets to corresponding queues, and using fewer queues [15]. Normally, one queue is required for every possible flow through the router. But based on the assumption that, at any particular time the number of active flows is much less than the total number of possible flows through the router. SFQ doesn't have a separate queue for every flow, and flows that hash into the same bucket are treated equivalently, as a result, those flows are treated unfairly.

1.3.5 Core-Stateless Fair Queuing

The Core-Stateless Fair Queuing (CSFQ) was proposed [16] for achieving reasonably fair bandwidth allocations while reducing the cost. It works by establishing an island of routers (a contiguous region of the network) that implement the protocol. The core of the island ("core" routers of the network) adopts a protocol that does not maintain, "state" (such as a record of the volume) for each flow, thereby allowing it to employ a fast, simple queuing algorithm. Such per-flow states still need to be maintained at the borders of the island ("edge" routers of the network), which then need to communicate rate estimates for the various flows to the core routers. In this method the core routers are not slowed down by flow-specific computations. Edge routers estimate flow's arrival rates by
exponential averaging based on per flow information and insert them into the packet labels, the flow’s arrival rates are updated at each router along the path based only on aggregate information at that router.

1.4 Literature Survey

Routers are the place where congestion occurs, so it would make sense to try and do smarter things at the routers with regard to congestion avoidance [17]. Several methods for Congestion avoidance in computer networks have been discussed in the literature. Following is the outline for briefly surveyed literature.

Computer networks is implemented using a variety of protocol stack architectures, computer busses and the combination of media and protocol layers [18-19]. Larry Peterson and Tennun baum et al (2000) explained, a computer network is a group of computers that are connected to each other for the purpose of communication [1, 20]. Distributed communication networks and resource sharing in distributed communication environment is explained by Paul Baran (1964)[21] and Leonard Kleinrode (2002) [2].

Thomas E. Anderson et al (1993) explains, current technology trends make it possible to build communication networks that can support high performance distributed computing. This paper describes issues in the design of a prototype switch for an arbitrary topology point-to-point network with link speeds of up to one Gigabit [22] and Terabit [23] per second.

Sven A M et al (2002) has explained the importance of switching technology to control the network traffic. It has been reported that, the elastic traffic include available bit rate traffic in Asynchronous Transfer Mode (ATM) networks and TCP/IP traffic networks

Cheng-Shang Chang et al (2001) have reported crossbar switches, due to their scalability, these switches have received a lot of attention recently [25]. The concept of separating the common memory block into multiple parallel sub memories shared among input/output ports explained by S X Wei et al (1992). They considered a scheduling algorithm for unicast cells to obtain optimal delay throughput performance [26]. Victor yau et al (1999) has reported the multichannel telecommunication networks, switching systems, and processor memory interconnects [27]. Xike Li et al (2008) presents that, Input-Queued cell switching architectures are commonly utilized in Internet routers as they offer pragmatic scalability while requiring moderate memory bandwidth [28].


G. Bajko’ I.Moldovan et al [32] proposed a TCP flow control technique, based on controlling the flow of acknowledgement (Ack) packets according to the level of congestion at the routers. In this way packet loss can be significantly reduced and even eliminated in the network, thus improving the good put of TCP connections and assuring a high fairness among them.

K.K.Ramakrishan et al [33] explains an algorithm for congestion avoidance in networks. The scheme uses a minimal amount of feedback from the network to the users, who adjust the amount of traffic allowed into the network. According to Dan Duchamp [34],
work on congestion control has always been focused on end-to-end transport layer techniques. The reason is that there are only two layers of protocol in the TCP/IP architecture, and IP is limited to being a dumb routing layer.

Michele C. Weigle et al [35] present an end-to-end alternative to AQM - new congestion detection and reaction mechanism for TCP based on measurements of one-way transit times of TCP segments within a TCP connection. Ahsan habib and Bharat Bhargava [36] use network tomography an edge-to-edge mechanism to infer per-link internal characteristics of a domain to identify unresponsive flows that cause packet drops in other flows. They have designed an algorithm to dynamically regulate unresponsive flows.

Asfand-E-Y et al [37] describes various approaches to congestion control of Internet traffic and focuses mainly on one of the congestion control approaches known as Random Early Detection (RED) congestion control mechanism for Active Queue Management (AQM) scheme at routers. Sally Floyd and Van Jacobson [38] present Random Early Detection (RED) gateways for congestion avoidance in packet switched networks. The gateway detects incipient congestion by computing the average queue size. The gateway could notify connections of congestion either by dropping packets arriving at the gateway or by setting a bit in packet headers.

Sharad Agarwal et al [39] explore that Random Early Detection (RED) queues are becoming ubiquitous as they offer higher throughput, lower latency and more fairness than FIFO drop tail queues with a relatively low implementation cost. By increasing the probability of dropping a packet as the queue size increases, RED queues attempt to detect imminent congestion and avoid it by informing flows about it.
Tigist Alemu's [40] work focuses on an adaptive approach of RED namely ARED (Adaptive RED) that performs a constant tuning of RED parameters according to the traffic load. ARED requires no hypothesis on the type of traffic, which diminishes its dependency on the scenario parameters such as the bandwidth, the round-trip time and the number of active connections.

T.J.Otta et al [41] describes a mechanism called SRED (Stabilized Random Early Detection). SRED pre-emptively discard packets with a load-dependent probability when a buffer in a router in the Internet or an Intrant seems congested. SRED has an additional feature that over a wide range of load levels help it stabilize its buffer occupation at a level independent of the number of active connection.

Ramakrishna Gummadi et al [42] find that revised version of Adaptive RED, which can be implemented as a simple extension within RED routers, removes the sensitivity to parameters that affect RED's performance and can reliably achieve a specified target average queue length in a wide variety of traffic scenarios.

Wu-chang et al [43] proposed a fundamentally different active queue management algorithm called BLUE. It uses packet loss and link idle events to manage congestion. Using simulation and controlled experiments, BLUE is shown to perform significantly better than RED both in terms of packet loss rates and buffer size requirements in the network.

that TCP is believed to be largely responsible for preventing congestion collapse while the Internet has undergone dramatic growth in the last decade. Indeed, numerous measurements have consistently shown that more than 90% of the traffic on the current internet is still TCP packets, which are fortunately congestion controlled [45].

Sundar Iyer et al (2008) explained that, Packet Switches, regardless of their architecture, require packet buffers. The general architecture presented them can be used to build high bandwidth packet buffers for any traffic arrival pattern or packet scheduling algorithm. The scheme uses a number of dynamic RAMs in parallel, all controlled from a single address bus. But in the systems for which the number of queues is too large, this technique is inapplicable [46].

R.L.Cruz et al (2008) addresses that, advances in communication link technology have increased the potential bit rates between nodes in packet switched networks. In modern packet switches, technology limitations may introduce switch configuration delays that are non-negligible compared with the time required to transmit a single packet [47].

Ion Stoica et al (1998) explains, Router mechanisms designed to achieve fair bandwidth allocations, like Fair Queuing, have many desirable properties for congestion control in the Internet. However, such mechanisms usually need to maintain state, manage buffers, and/or perform packet scheduling on a per flow basis, and this complexity may prevent them from being cost-effectively implemented and widely deployed. They proposed an architecture that significantly reduces this implementation complexity yet still achieves approximately fair bandwidth allocations, and they call the scheme Core-Stateless Fair Queuing [16].
Nabeshima. M (2002) reports that, in stateless core (SCORE) networks, edge routers maintain per-flow state while core routers do not. Core Stateless Fair Queuing (CSFQ) has been proposed for approximating the operation of per-flow queuing techniques in SCORE networks. However, the packet dropping probability offered by CSFQ suits only for UDP flows. Thus, CSFQ cannot achieve fair bandwidth allocation for TCP flows [48].

Stoica. I et al (2002) themselves argue that, Previously (Ion Stoica et al 1998), a class of solutions including core-stateless fair queuing, rainbow fair queuing, and Diffserv have been proposed to address the scalability concerns that have plagued stateful architectures such as Intserv and fair queuing. However, despite some desirable properties, these solutions still have serious scalability, robustness, and deployment problems. Their scalability, suffers from the fact that the core cannot transcend trust boundaries, and so the high-speed routers on these boundaries must maintain per flow or per aggregate state. The lack of robustness is because a single malfunctioning edge or core router could severely impact the performance of the entire network [49].

Cristel Pelsser et al (2002) explained Core Stateless Fair Queuing (CSFQ) is a scalable mechanism to provide per-flow fairness in high-speed networks in that it does not need to maintain per-flow state in the core routers. This is possible because the state for each flow is encoded as special labels inside each packet. In this paper, they propose and evaluate by simulations, two improvements to CSFQ. They show that CSFQ does not provide a fair service when some links are not congested. Second, they propose an algorithm to allow CSFQ to provide a service with a minimum guaranteed bandwidth and evaluate its performance with TCP traffic [50].
Peng et al (2005) explains that, the fair bandwidth allocations can isolate flows and protect well-behaved flows from ill-behaved ones. CSFQ (Core Stateless Fair Queuing) achieves the approximate fairness by dropping the extra packets beyond the fair share bandwidth at the routers. A heuristic method is used to estimate the fair share in CSFQ. Furthermore, they explained that SRED (Stabilized RED) uses a probabilistic method based on a Zombie list to estimate the number of flows at the router. In this paper, they took the probabilistic idea from SRED and apply it to CSFQ to estimate the fair share without using the Zombie list [51].

Sungwoo Park et al discusses a new algorithm called CSFQ-, which maintains the overall structure of CSFQ yet achieves higher degree of fairness. First they tested a hypothetical algorithm called CSFQ+ and set their direction in order to improve CFSQ. Then they test three ideas to improve CSFQ: discrete averaging of accepted track rates, tangent-based linear interpolation in computing the estimate fair share, and fast drop of estimate flow rates. From experimental results, they analyze CSFQ- as the algorithm resulting from the combination of the latter two ideas. They also found that an algorithm which works poorly (or well) in hypothetical scenarios may turn out to be highly effective in real networks. They stated that, though CSFQ achieves a much higher degree of fairness in bandwidth allocation compared with other routing mechanisms with similar implementation complexity, still there are several possibilities for improving CSFQ [52].

The self-similar nature of network traffic has been convincingly demonstrated in a number of research studies as shown in Table 1, and if traffic is ignored, this property affects the performance of networks resulting in higher delays and lower throughput.
Table 1.1. Performance of current congestion avoidance algorithms

<table>
<thead>
<tr>
<th>No.</th>
<th>Algorithm</th>
<th>Main Strengths</th>
<th>Main weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drop Tail (DT) [11, 53]</td>
<td>Simplicity; no State information is needed</td>
<td>Lacks of QoS; no fairness; global synchronization problems; biased for bursty traffic</td>
</tr>
<tr>
<td>2</td>
<td>DECbit [54-55]</td>
<td>Simple; distributed; optimized; low overhead; congestion feedback by marking packets; dynamic; provides good fairness</td>
<td>Use simple averaging; biased against bursty traffic</td>
</tr>
<tr>
<td>3</td>
<td>RED &amp; Variants [12,40, 56-60]</td>
<td>Simple; fair; EMWA; AQM; unbiased for bursty traffic</td>
<td>Sensitive to the parameters settings</td>
</tr>
<tr>
<td>4</td>
<td>Proportional Integral (PI) control [61-62]</td>
<td>Simple; fast; robust; AQM; less queue oscillations</td>
<td>Estimation and setting of constant</td>
</tr>
<tr>
<td>5</td>
<td>CHOKe [63-64]</td>
<td>Simple; stateless and easy to implement</td>
<td>Fairness and scalability problems</td>
</tr>
<tr>
<td>6</td>
<td>Virtual Queue (VQ) [65-66]</td>
<td>High link utilization</td>
<td>Fixed &amp; DT type of VQ</td>
</tr>
<tr>
<td>7</td>
<td>BLUE &amp; SRED [43, 67]</td>
<td>Low packet loss rate and less buffer needed</td>
<td>Not scalable</td>
</tr>
<tr>
<td>8</td>
<td>Random Exponential Marking (REM) [68-69]</td>
<td>Low packet loss; high link utilization; scalable and less delay</td>
<td>Based on global parameter; lacks QoS</td>
</tr>
<tr>
<td>9</td>
<td>FQ &amp; DRR [13, 70]</td>
<td>Bound on delay</td>
<td>Expensive to implement</td>
</tr>
<tr>
<td>10</td>
<td>SFQ [15, 37,71]</td>
<td>Reduced look up cost</td>
<td>Complicated; incomplete fairness; more queues</td>
</tr>
<tr>
<td>11</td>
<td>Class Based Queue (CBQ) [72]</td>
<td>Better management of gateway resources</td>
<td>Modified Ethernet; no traffic control</td>
</tr>
<tr>
<td>12</td>
<td>CSFQ [16, 48-52]</td>
<td>Fairness</td>
<td>Extra field in packet header</td>
</tr>
</tbody>
</table>
1.5 Problem Statement in Packet Switching Technique

Studies on congestion avoidance, which outline measures for controlling network traffics in order to prevent, avoid, or recover from network congestion, have long been considered significant for the future development of network communications. A large number of various congestion avoidance schemes have been proposed, and few mechanisms have been implemented in real networks. However, despite years of research efforts, the problem of network congestion control remains a critical issue and a high priority, especially given the prospective of the continually growing speed and size of future networks.

In packet switching technique it is revealed that if the network traffic is ignored, this leads to the congestion in network. Congestion avoidance in packet switching networks became a high priority in network design and research due to ever growing network bandwidth and intensive network applications. Congestion in a packet switching network is a state in which performance degrades due to the saturation of network resources such as communication links, processor cycles, and memory buffers. Adverse effects resulting from such congestion include the long delay of message delivery, waste of system resources, and possible network collapse, when all communication in the entire network ceases. Network congestion, like traffic jams in big cities, are becoming real threats to the growth of network interconnections and communication applications [73-74].

This property of congestion affects the performance of networks resulting in low throughput and high latency. The main challenges to be considered to avoid congestion are:
Chapter 1: Introduction

1. throughput (total output traffic rate/input traffic rate)
2. packet delays (latency)
3. utilization of link capacity (average input traffic rate/maximum possible output traffic rate)
4. packet loss rate
5. amount of buffering
6. complexity of implementation

There is significant evidence that traffic control is an important aspect of effectively managing congestion, so that available bandwidth is utilized and degradation of the quality of higher priority connection is avoided. Thus one intuitively expects high throughput and less latency in networks under heavy traffic loads mainly from these challenges.

1.6 Motivation for research work

Rapid advances in computer technology have lead to the emergence and evolution of many new techniques for computer communication and networking. Network congestion has been well recognized as a resource sharing problem. In a packet switched network, resources are shared among all the hosts attached to it, including switch processors, communication channels, and buffer spaces. These three driving forces of data transmission in network communication can also be potential bottlenecks that cause congestion in the network. On one hand, networks need to serve all user requests for data transmission, which are often unpredictable and bursty with regard to transmission starting time, rate and size. On the other hand, any physical resource in the network has a
finite capacity, and must be managed for sharing among different transmissions. Consequently, network congestion will result if the resources in the network cannot meet all of the users' current demands.

A more formal and quantitative definition for network congestion is based on the performance behavior of a network. Fig. 1.3a shows the throughput-load relationship in a packet switching network without effective means of flow control. It is noticed that, if the load is small, network throughput generally keeps up with the increase of load until the offered load reaches the knee point, where the increase of the throughput becomes much slower than the increase of the load. If the load keeps increasing up to the capacity of the network, the queues on switching nodes will build up, potentially resulting in packets being dropped, and throughput will eventually arrive at its maximum and then decrease shapely to a low value. It is at this point the network is said to be congested [75].

Fig. 1.3b and 1.3c illustrates the relationship between the round trip delay, and the resource power with respect to the offered load. The delay (or response time) curve follows a similar pattern as the throughput curve. At first, the response time rises slowly with the load due to the fast increment of the throughput. Then after the knee point is reached, the delay curve jumps significantly while the throughput stays flat. Finally, the delay grows indefinitely when the network becomes congested. The resource power is defined as the ratio of the throughput to the response time. The resource power gets to maximum value at the knee point, where the average queue size is close to one, including the packet in service.
In order to maintain a network always in a healthy working condition, certain measures or mechanisms have to be provided to prevent the network from operating in the congested region for any significant period of time. Such mechanisms are generally referred to as the congestion avoidance of networks. The congestion avoidance in packet
switching networks may involve different components in a network, including the host machines of sources and destinations, as well as switching nodes.

The problem of congestion avoidance has long been considered as an important topic in R & D of computer networks. With the recent development of network technology and the growth of network intensive applications, the issue of congestion avoidance becomes even more essential. A great number of congestion avoidance algorithms and strategies have been proposed and developed, ranging from Drop Tail, Random Drop [76], Fair Queuing, SFQ, CSFQ, and so on.

All these algorithms vary in terms of their operating conditions, functional principles, and performance behaviors. Although a number of survey papers on a variety of congestion control algorithms have appeared in the literature [77-80], there is still no a systematic way to find the solution for congestion avoidance. A framework of research and performance evaluation of congestion avoidance algorithms on packet switching networks will help people understand the major features of existing algorithms and similarities and differences among various avoidance schemes, and formulate new congestion avoidance algorithms that can be better fit the characteristics of future network traffic [81].

1.7 Scope of the Research work

Modern telecommunication and computer networks, including are being designed for fast transmission of large amount of data for packet switching, for which Congestion avoidance algorithms are very important. Without proper congestion avoidance algorithm, congestion collapse of such network is a real possibility. Congestion avoidance in packet switching networks became a high priority in network design and
research due to ever growing network bandwidth and intensive network applications. Efforts have been made by several investigators to control network traffic effectively, so that available bandwidth is utilized and degradation of the quality of higher priority connection is avoided.

However, as the internet evolves into a global commercial infrastructure, there is a growing need to provide more powerful services. Because increasing link bandwidth demands faster nodal processing, especially of data plane traffic. Data processing ranges from routing table lookup to various classifications for firewalls, differentiated services and web switching. The traditional general purpose processor architecture is no longer sufficiently scalable for data communication networking.

Hence current work proposes an evaluation of congestion avoidance algorithm in packet switching computer networks. Based on control theoretic concepts, congestion avoidance scheme is viewed as a control policy to achieve prescribed goals such as transmission latency and throughput in a distributed network environment.

Over the past decade, there has been intense research toward achieving high performance networks with low latency and high throughput. Several mechanisms have been proposed, and they can be classified into two basic types: random early detection (RED) [17-25] and fair queuing (FQ) [26-31].

In the present work, a set of criteria for control systems is used to evaluate various congestion avoidance algorithms. Performance evaluation of existing algorithms, not only provides a coherent framework for comparative studies of existing approaches, but also helps in research and development of new strategies for congestion avoidance.
1.7.1 Aim and Objectives

In the present research work, extensive and systematic studies were carried out on RED and FQ algorithms. The performances of these algorithms were evaluated and their limitations were observed. A new technique has also been developed to overcome the observed limitations.

The main objectives of the present investigation were to analyze high-performance switching techniques, and their influence on the operation and functionality of network infrastructures, thereby developing a new/optimum decision function for switching circuits to provide better communication. Also, the present study involves

1. Detailed studies on the challenges in existing switching techniques and their effects.
2. Suitable suggestions (proposals) to overcome the challenges and hence their trade-off with existing techniques.
3. An approach and solution to the best proposals.

To start with, extensive studies have been carried out on the analysis of RED and its variants, Fair Queuing (FQ) and Core Stateless Fair Queuing (CSFQ) routing algorithms. These algorithms were studied in detail and their performances were evaluated using a sophisticated package NS2 [82-83]. The parameters used to evaluate these algorithms are packet size, buffer capacity, the transmission time and the number of flows. In order to overcome the disadvantages of existing RED and CSFQ mechanisms, a slightly modified version of CSFQ was developed called as Enhanced CSFQ or ECSFQ. ECSFQ not only achieves high throughput and less packet loss compared to existing techniques, it also achieves fair bandwidth allocations to all flows when used with preventive congestion-
avoidance mechanisms. The proposed work introduces a load balancer which contains a Classifier and Scheduler in the core router.

A. Classifier

1. It classifies packets into either Aggressive or Normal flow.
2. It calculates percentage of time allocation for scheduler.
3. It also computes packet drop rate for both flows.

B. Scheduler

It processes the aggressive and normal flow queue based on percentage of time allocated for these queues.

Hence the present work would therefore be of great importance both from the point of study as well as reliable technological development to provide good Quality of Service in the field of Communication. The study would provide the key alternative design and process facility to make the switching device immune to communication background.

1.8 Organization of the Thesis

Chapter-1 embodies the general introduction of computer networking, Switching Techniques, congestion, Congestion control and congestion avoidance, literature survey on recent studies of congestion avoidance algorithms for packet switching technique, problems statement and motivation for the research work. This chapter also outlines the broad aim and scope of the work.

Chapter-2 describes methods of switching technologies – Circuit switching and Packet switching, advantages of packet switching over circuit switching, delays in packet
switching, the routers, queuing, buffering and the switching mechanism. An overview of Network Simulator NS2.

Chapter-3 deals with results and discussion of Random Early Detection algorithm (RED) and its variants. The performance of these algorithms is evaluated with the parameters such as packet drop rate and transmission time. The Network Simulator NS2 is used to implement the algorithms.

Chapter-4 includes results and discussion of the analysis and simulation of Core Stateless Fair Queuing (CSFQ) algorithm. The performance of this algorithm is compared with RED and FQ algorithms. The network Simulator NS2 is used to simulate these algorithms. The major parameters used to discuss these algorithms are packet size, buffer capacity, transmission time and the number of flows.

Chapter-5 incorporates results and discussion on analysis and simulation of new proposed technique called Enhanced Core Stateless Fair Queuing (ECSFQ) algorithm. The result of ECSFQ has been compared with RED and CSFQ algorithms. The prominent parameters used to compare this algorithm with RED and CSFQ are packet size, buffer capacity, transmission time and the number of flows. NS2 is used for simulations.

Chapter-6 outlines the main conclusions of the work. The conclusions drawn from the present investigations were summarized in this chapter. The suggestions for future studies in this area of research are also given at the end of the concluding chapter.