Chapter 1
Introduction

1.1 Background
Today's computer networks such as the Internet form an essential component of modern civilization. They are responsible for a variety of services in our society and their importance is unquestioned and unparalleled. In order to get maximum benefit at the least cost, these networks should be well designed and operated in an optimized manner. Though most computer networks work well under light load, congestion problems start to occur under heavy load. Therefore avoiding and controlling congestion is one of the most critical issues in present and future networks. This dissertation presents active network based approach router algorithms for congestion avoidance and control.

1.2 Congestion Control Problem
Congestion can be defined as a network state in which the total demand for resources, e.g. bandwidth, among the competing users exceeds the available capacity, leading to packet or information loss and resultantly requiring packet retransmissions. At the time of congestion in a computer network there will be a simultaneous increase in queuing delay, packet loss and number of packet re-transmissions. Thus, congestion will ultimately lead to a drop in a network's throughput which is highly undesirable. [1-5]

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. The congestive collapse is a network condition when little or no useful communication is happening due to congestion. 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).

Modern networks use congestion control and network congestion avoidance [6] techniques to try to avoid congestion collapse such as original Ethernet, window reduction in TCP, and fair queuing in devices such as routers.

The classic problem of congestion control has been described in [7][8], where a computer network is required to be designed to have the operating point at knee (associated with number of users \( nk \)) for congestion avoidance, between knee and cliff (associated with number of users \( nn \)) for congestion control, as shown in Figure 1.1. Where the total load (bits/s) handled by network at knee and cliff points is represented by \( l(nk) \) and \( l(nn) \), respectively, with \( l(nk) < l(nn) \).

![Figure 1.1 Illustration of Congestion Avoidance and Control problem. Where \( l(ni) \) is the load (bits/s) generated by \( ni \) number of users.](image)

TCP's congestion control mechanism is well studied and is effective in short delay path. TCP (and its variants) are capable of detecting congestion and adjust its sending rate accordingly. Slow start algorithm is used to control the congestion by TCP, can cause performance degradation on high delay links.

For high delay link, round trip time (rtt) is large. The rate at which the cwnd increases depends on the round trip time (rtt) for the TCP segments. In other words TCP, throughput is inversely proportional to the rtt. As the separation between the sender
and receiving host increases, rtts increase resulting in reduced TCP throughput. Also for longer connection paths there is a high probability for packet loss.

Present TCP congestion control schemes are employed at end hosts. These are based on feedback (ACKs).

1.3 TCP and congestion control - literature survey

Congestion control is a major performance/design issue in computer networks and in the Internet. Unprecedented growth of internet and excessive and innovative use of internet with poorly implemented TCP, opening multiple TCP connections aggressively, bandwidth hungry traffic flow are some of the causes of congestion and further resulting in congestion collapse. This has been elaborated by researchers like Floyd, Nagle, Fall, Jacobson [6, 9-17]

TCP itself has incorporated some of the congestion control algorithms like slow start, congestion avoidance, Fast retransmit, Fast Recovery, AIMD in various variants [7]. They have their own limitations. Different TCP variants; use above algorithms for congestion control. Although there are many TCP variants some of the commonly used TCP variants are Tahoe, Reno, New-Reno, SACK, Vegas. [17-24]

The successful performance and robustness of the present Internet is mainly due to TCP which has undergone number of changes in its primitive design, in an evolutionary process, over the last three decades. Presently, couple of basic congestion control algorithms, namely slow start, congestion avoidance, fast retransmission and fast recovery, are essential part and parcel of almost every existing and newly proposed versions of TCP's. These congestion control algorithms of TCP are end-to-end based and do not require any support of routers for their functioning.

All TCPs detect congestion either by detection of packet loss or after receiving duplicate acknowledgement and resort to AIMD mechanism to overcome it. Packet dropping wastes the network resources therefore another method called explicit congestion notification (ECN) is used to mark the packet instead of dropping packets. [25-43]
TCP Reno Connections with shorter round trip time can update congestion window sizes more quickly. But TCP Reno is biased against connections with longer delays. While a source does not detect any congestion, it continues to increase its window size by one during one round trip time. The analysis suggests that TCP Reno's bias against connections with larger round trip time is a fundamental consequence of its congestion window dynamics [22, 44, and 45]. One solution to this problem is to use higher buffer size. However, using buffers of large capacity will increase the response time and thus consequently slowing the system. Also, increased queuing delays will harm Quality of Service [46].

Most of the TCP connections are used for short data transfer [47-51]. For short TCP connections the steady models of TCP are in [27, 37, 45, and 47]. For quantitative analysis mathematical models are proposed in [27]. Others have done work in direction are [22, 27, 37, 52-54].

Transport protocol for high bandwidth delay product network. : Present TCP based flow show poor performance in high bandwidth delay product environment. Internet user wish to achieve good throughput employ multiple TCP connections in parallel. However this approach does not work well in moderate or high congestion [55, 56].

Explicit Congestion Notification (XCP) addresses the problem of low performance by TCP networks. XCP requires the participation of routers. It achieves high link utilization and reduces wastage of bandwidth by separating the mechanisms for efficiency and fairness. A XCP based router has two parts namely: efficiency controller and fairness controller, [57, 58]. The purpose of the former is to maximize the link utilization and the latter is to ensure fairness.

High speed TCP for Gigabit network uses large congestion windows. The concept of high speed TCP has been extended for scalable TCP. It has been implemented in the Linux 2.4.19 OS kernel [59, 60].

Fast AQM scalable TCP (FAST) has been introduced for high speed long latency networks in 2004 [61]. It separates the congestion control algorithm of TCP into four components, which are data control, window control, burstiness control and estimation. All these have been described in [55-61].
Traditionally performance of TCP type protocols is mainly determined by throughput and fairness [19, 62, 63, and 64]. However, other metrics such as link utilization and loss rate are also important in effective evaluation of TCP. These metrics are described in the [64-68].

We tried to cover major points from the exhaustive literature available on TCP and congestion control. Thus this section is the basis to understand various ways adopted for congestion control using end to end principle

1.4 Problem Statement

All the congestion control schemes employed by TCP and its variants are incorporated at end hosts. Different schemes have their own advantages and disadvantages [2].

For large bandwidth delay product they are hardly effective. These schemes are designed keeping in mind the end-to-end system design principle [69] which is there in existing for more than 25 years.

Some schemes like Active Queue Management (AQM) [70] are employed at router but these schemes only work related to managing the queue at the router by dropping the packets. Lot of efforts have been put in by various researchers for improving the performance by such schemes but these schemes stick to end to end argument.

It is a proven fact that end-to-end implementation can help robustness, scalability, ease of deployment and the provision of appropriate service. However in recent time, end-to-end argument has been challenged by advent of firewalls, caches, NAT and network QOS etc [71].

The central thrust in this dissertation is departure from end-to-end argument and tries to implement by exploring the potential advantages offered by the active networking in improving the throughput in high delay bandwidth network.

The reasons for deviating from end-to-end argument are as follows [71, 72]
1 Congestion is a phenomenon of network. Multiple endpoints share network and offers excessive traffic. It is the responsibility of the network to isolate the endpoints. Or in other words, even though the network is responsible for controlling congestion, it has no reason to trust that endpoint will co-operate in controlling congestion.

2 It is inappropriate for certain networks to implement congestion control at end point because it leads to an unnecessary performance penalty. e.g. slow start algorithm unnecessarily impedes sources that are transmitting on optical circuits (which does not congest). Assumption of packet loss indicates congestion is invalid for wireless networks in which appreciable loss may also occur due to noise. For traffic local to a LAN, congestion control is provided by MAC protocol.

3 Transport layer can only detect possible presence of congestion by observing packet loss. By no way it cannot detect that the congestion is imminent. Schemes such as RED may signal imminent congestion, but they do so by unnecessarily discarding traffic for which the network has already spent resources for delivering.

4. End points that implement congestion control separately must independently re-learn the network state, leading to excessively cautious behavior. Finally, while the end point may know how it would like to adapt to congestion, it is the network that knows when and where adaptation is needed [72] and should be responsible for ensuring that adaptation occurs.

Thus, congestion control is one function that is not well suited to end-to-end implementation.

Above points have motivated us to deviate from end end argument, make us strongly believe to explore possibility of reducing congestion and to address the issue of improving the throughput by adopting the active network based approach.

1.5 Proposed Approach

For above reasons we have chosen a path to implement additional congestion control function at the intermediate system (e.g. router). Traditionally routers only forward packets.
Traditional data networks passively transport bits from one end system to another. Ideally, the user data is transferred opaquely, i.e., the network is insensitive to the bits it carries and they are transferred between end systems without modification. The role of computation within such networks is extremely limited, e.g., header processing in packet-switched networks and signaling in connection-oriented networks.

Active Networks break with tradition by allowing the network to perform customized computations on the user data. For example, a user of an active network could send a customized compression program to a node within the network (e.g., a router) and request that the node execute that program when processing their packets. With the availability of high end hardware (such as high speed processor, fast memory at cheaper rate, network processor), it is possible to implement this.

In this dissertation we have shown that intermediate system can participate in controlling congestion and help in improving the throughput. This is possible by incorporating software service in the router which will ultimately prove effective in improving throughput. Router can provide execution environment for data packets containing code (i.e active packet).

This dissertation explores and investigates congestion related issues. It analyzes present congestion control algorithms in routers where routers become active router to provide execution environment for code contained in the packets. It compares the results with standard TCP throughput. Simulation experiments are performed using variety of traffic and scenarios.

1.6 The outline of the thesis

The outline of the thesis is as follows:
In Chapter 2 detail implementation of simulation based comparison of TCP variants namely Tahoe, Reno, and SACK is carried out.
Chapter 3 introduces active networking concepts and related work for throughput improvement
Active node based Congestion Control (ACC) is used to improve network performance through Router assisted dynamic congestion control. ACC system is an
active network based system, which improves the networks performance with large delay-bandwidth product. ACC system can be deployed at various active nodes in the network. Chapter 4 deals with details of ACC system, its simulation and comparison with standard system.

Performance Enhancement Proxy (PEP) is used to improve TCP performance by deploying it generally in satellite network Chapter 5 presents detailed of simulation of PEP and its comparison with standard TCP in wired network environment.

Proxy Transport Service (PTS) is active networks based service, which improves the TCP bulk data throughput for TCP connections with large rtt. PTS splits the TCP connection between sender and receiver in two separate TCP connections. Chapter 6 gives details of Proxy Transport Service (PTS) which is host initiated service. TCP throughput of PTS is compared with that of standard TCP and TCP Vegas with the aid of simulation. The details of PTS are given in Appendix C

Three schemes based on active network based approach like active node based congestion control, PEP and PTS, their performance is compared with standard TCP and results are indicated in chapter 7.