Veermata Jijabai Technological Institute
Matunga, Mumbai – 400 019

Electrical Engineering Department

Synopsis on Dissertation Work

Name of the Candidate : Girish Prakash Bhole

Degree : Ph. D. (Tech) (Electrical Engineering)

Institute : Veermata Jijabai Technological Institute (VJTI)

Name of Guide : Dr. S A Patekar

Date of Registration : 13th December 2004

Registration Number : 8

Place of Research : Veermata Jijabai Technological Institute
Matunga, Mumbai – 400 019

Title of the Dissertation : Transport Control Protocol Congestion Control
in Active Networks

Guide
Dr. S A Patekar

Candidate
G P Bhole
SYNOPSIS
Transport Control Protocol Congestion Control in Active Networks

In last few years computer networks have experienced an explosive growth and with it the congestion problems. One of the potential problems of congestion is large feedback delay, which results due to increase in bandwidth delay product.

Though most networks perform well when lightly used, problems can appear when the network load increases. Congestion refers to a loss of network performance when a network is heavily loaded. Since congestion phenomena can cause data loss, large delays in data transmission, and a large variance in these delays. Therefore controlling or avoiding congestion is a critical problem in network management and design. This dissertation presents TCP congestion control using active network based approach for improving throughput in wide-area computer networks for large bandwidth delay product.


There are some non TCP congestion control techniques. Random Early Detection (RED) [5] is a notable enhancement to congestion control technique for inter-network routers. This algorithm manages router queues and drops packets based on a queue threshold. This in effect causes congestion control to be activated just prior to any network congestion event. Explicit Congestion Notification (ECN) [6] is a technique that just marks packets instead of dropping them as RED usually does. ECN avoids packets drops, particularly where the delay involves caused by retransmission needs to be avoided. Conventional approach of congestion control is strongly revealed on end-to-end argument.

The design of TCP was heavily influenced by the end-to-end argument. [7]
The key component of the end-to-end argument is in its method of handling congestion and network overload. The premise of the argument and fundamental to TCP design is that the end stations are responsible for controlling the rate of data flow. In this model, there are explicit signaling mechanisms in the network, which tell the end station how to transmit, when to transmit, when to speed up or when to slow down. The TCP software in each of the end station is responsible for answering these questions from implicit knowledge it obtains from the network or explicit knowledge it receives from the other TCP hosts.

In this dissertation we digressed from the end to end approach because of following reasons [8] [9]

Congestion is a phenomenon of network. Multiple endpoints share a network and offers excessive traffic. It is the responsibility of the network to isolate the end points or in other words, even though the network is responsible for controlling congestion, it has no reason to trust that an end point will cooperate in controlling congestion.

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 unnecessary impedes sources that are transmitting on optical circuits( which do not congest). Assumption of packet loss may also occur due to noise. For traffic local to LAN, congestion control is provided by MAC protocol.

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.

End points that implement congestion control separately must independently relearn 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, and should be responsible for ensuring that adaptation occurs.
Thus congestion control in particular is one function that is not well suited to end to end implementation. Therefore in this dissertation we take a fresh look at the congestion problem in computer network using active network based approach [10] [11].

Attempt has been made to show that intermediate system can participate in controlling congestion and help in improving throughput. This is done by incorporating software service in router which will perform computation on active packets. This active node shall perform tasks like generating premature acknowledgement, forward stored packets at the node buffer etc. The service could be transparent or host aware.

Performance of the network is evaluated by measuring the throughput for three schemes based on active network concept. Three schemes are Active node based Congestion control, Performance Enhancement Proxy and Proxy Transport service. In this thesis we have attempted to show that intermediate system can participate in controlling congestion and help in improving the throughput.

Active node based congestion control system:
Active node based Congestion Control (ACC) is used to improve network performance through Router assisted dynamic congestion control [12] for large bandwidth delay product. ACC system uses Active Network technique to enable router participation in both congestion detection and recovery. The feedback congestion control system is extended from the endpoints into the routers. Congestion is detected at the router, which also immediately begins reacting to congestion by changing the traffic that has already entered the network.

Locating both the congestion detection and congestion response at the router removes the feedback delay; the system is stable because changes made at the routers are propagated back to the endpoints. In a conventional feedback system, congestion relief must move from the endpoint to the congested as the endpoint sending rate is reduced; in ACC the congestion relief starts at the congested node and the change in state that sustains that relief propagates out to the endpoint. Simulation on considered
network shows that the active congestion control system shows improvement in throughput under busy traffic conditions.

Performance Enhancement Proxy [13]

The motivation behind this approach is TCP's performance degradation in a heterogeneous environment. PEP node is an active node located in the path between the sending and the receiving host at the edge of the Heterogeneous Network. The idea is that if a communication path consists of physical media that have very different characteristics, the end-to-end performance is optimized by isolating one type of physical medium from another and optimizing each separately. The PEP node acknowledges the sender's TCP segments even before they have reached the receiver, reducing the rttr (round trip time) on sender to PEP link, thereby increasing the sender's transmission rate. PEP node will in turn forward the segments to the receiver, buffering them until it receives acknowledgements from the actual receiver. PEP node provides buffer space for storing the unacknowledged segments. PEP node implements all the functionalities of a host TCP sender i.e., it will retransmit lost segments, perform flow control and respond to congestion. PEP is transparent service.

Proxy Transport service [14]
Proxy Transport Service (PTS) is an active networks based service which improves the TCP bulk data throughput for TCP connections with large average rttr. The rttr for TCP packets on a connection can be large when the connection spans multiple subnets, or when one of the node in route is congested with no alternate route is available to the destination or when propagation delay for one of the intermediate links is very high (e.g. a satellite link or a slow wireless link). Active nodes are programmable network elements which can perform programmable computations on user packets inside the network. PTS service is deployed at various active nodes in the network. PTS works by splitting the TCP connection between the sender and the receiver into two separate TCP connections:

Using simulation and analysis we have shown that PTS improves the TCP throughput for TCP connections having large average rttr. Also we have shown how PTS node
location, PTS service time and buffer space per session affect the PTS throughput. A scheme for connection establishment and termination using PTS is presented.

PTS is on-demand service i.e., a host must explicitly make a request for PTS service to the active node. This approach is different from similar scheme like PEP [13], wherein a proxy service is provided transparently. That is the host is not aware of the proxy service. The active node has to inspect all the incoming packets to determine which flows must be provided with a proxy service. This is not only an additional overload on the active node but also affect the best effort traffic significantly violating the end-to-end argument. In our design, the intelligence of determining whether a connection needs PTS service and choosing a PTS node resides in the host. If multiple PTS nodes are discovered the ranking function to determine the suitability of the PTS node is presented. We have proposed in this dissertation a rank based formula for locating PTS node.

Central thrust in this dissertation has been deviating from end-to-end system design principle. Improvement of throughput by controlling congestion is kept in focus throughout. Extensive simulation experiments are carried out using variety of traffic and scenarios and varying different parameters for the above referred schemes and compared with the standard TCP throughput. The analysis shows that the active network based approach effectively controls congestion, resulting in much improved throughput.

References:


