International Journal of Allied Practice, Research and Review
Website: www.ijaprr.com (ISSN 2350-1294)

An Introduction to ACC Mechanism for Congestion Control

Syed Nusrat[#1]

[#] *Department of Computer Science, SJJT University*
Churu-Bishau Road, Chudella, Jhunjhunu, Rajasthan, India

ABSTRACT – Active Congestion Control is another scheme to avoid congestion. It is not possible to remove complete congestion but we can minimize the congestion based on the properties required for congestion control. In this paper we have discussed another scheme for congestion control, names as Active Congestion Control.
Active Congestion Control (ACC) uses Active Networking (AN) technology to make feedback congestion control more responsive to network congestion. Current end to-end feedback congestion control systems detect and relieve congestion only at endpoints. ACC includes programs in each data packet that tell routers how to react to congestion without incurring the round trip delay that reduces feedback's effectiveness in wide area networks. The congested router also sends the new state of the congestion control algorithm to the endpoints to ensure that the distributed state becomes consistent. We discuss a model for extending feedback congestion control into an Active Network, apply that model to TCP congestion control, and present simulations that show that the resulting system exhibits up to 18% better throughput than TCP under busty traffic. In simulations without busty traffic, the systems behaved comparably.

Keywords: Networks; TCP; ACC; Congestion;

I. INTRODUCTION

Active Networking Congestion Control (ACC) is a system which uses Active Networking (AN) technology to greatly reduce the control delay that feedback congestion control systems exhibit. Each ACC packet contains either a program for a router or data used by such a program that enables a router to react to network congestion thereby avoiding the delay in communicating congestion information to endpoints.
We chip away at reproduction investigations of an ACC framework focused around TCP blockage control components. The reproductions contrast the dynamic framework with standard TCP blockage control in systems with and without well-proportioned cross movement. In reproductions without curvaceous movement, the frameworks carried on equivalently. At the point when full figured cross-activity is included, the dynamic framework indicates as much as an 18% throughput change. Input based blockage control frameworks don't scale well regarding system data transfer capacity and postponement in light of the fact that increments in either amount

decouple the clogging site and endpoint. The bigger the end-to-end postpone in a system, the more extended until the endpoint can establish that the system has ended up congested. The higher the data transfer capacity of the system, the bigger the measure of information the endpoint may send into a congested system in the time it takes the endpoint to discover the blockage. Bolot and Shankar have demonstrated that under input based clogging control, the span of blockage at the bottleneck of an association is straightforwardly identified with the transfer speed postponement item [1]. High Speed Networks are an illustration of a class of systems with an extensive transmission capacity delay item.

Dynamic Networking, the thought of reconstructing switches with information PACKETS [2], OFFERS A CHANCE TO address this weakness of criticism control. A has been proposed to proposed location numerous system issues, for instance, to design systems dynamically[3], or to perform application-particular assignments in the network[2], Work by Bhattacharjee has connected A thoughts to non-criticism ATM blockage control[4]. ACC applies those thoughts to input blockage control. ACC moves the endpoint clogging control calculations into the system where they can promptly respond to blockage. The current condition of the endpoint's input calculation is incorporated in every bundle. This state is little, a number or two, to keep for every bundle overhead low. At the point when a switch encounters blockage, e.g., it is compelled to dispose of a bundle, the switch computes the new window estimate that the endpoint would pick on the off chance that it had right away recognized the clogging. The switch then erases parcels that the endpoint would not have sent and educates the endpoint of its new state. Hence, ACC switches in a flash unsend bundles that would have delayed blockage in the system.

II. OBJECTIVES AND HYPOTHESIS

The ACC System - ACC system uses AN techniques to enable router participation in both congestion detection and congestion 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 congestion 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.

III. RESEARCH METHODOLOGY

ACC for TCP - As a test of the ACC principles outlined above, we have defined an active congestion control based on TCP. TCP contains a classic, well-understood feedback control system: the congestion avoidance mechanisms defined by Jacobson [5]. Endpoint sending rate is controlled by a sliding window which is advanced by packet acknowledgments. The size of the window is modulated in response to congestion along the connection's path [5].

The window modulation algorithm in TCP is a classic linear increase/multiplicative decrease algorithm. When congestion is detected, the window is reduced to half its current size. When a full window of consecutive packets has been acknowledged without congestion being detected, the window is increased by one maximum-sized packet. We omit the discussion of the Slow-Start algorithm because the current work considers primarily steady state effects. An endpoint deduces that a connection's path is congested when it detects a packet loss on that connection. Such a loss can be detected by a retransmission timer expiring, or by receiving three consecutive acknowledgments of the same packet by the other endpoint. The ACC implementation based on TCP, called ACC TCP, follows the same algorithm, except that traffic modification begins at the congested router. When a router detects a packet loss, it calculates the correct window size for the endpoint from information it provides in each packet, and forwards a packet with the new window size to the endpoint [5]. Using TCP's window adjustment algorithm, the new window is half the old.

Then the router installs a filter at the previous router on the connection's path that deletes all packets from that endpoint until it begins to act on the router's feedback (or it has detected congestion itself).A more sophisticated ACC TCP implementation would install filters that deleted or delayed endpoint packets so that they would appear to the congested router to have been sent by a Slow-Starting endpoint. The current implementation uses the simpler filter that unsends the rest of the current endpoint's window, by discarding packets in flight. A version of ACC TCP that takes full advantage of AN to more fully edit traffic is the topic of future work [11].
Under ACC, the preferred method of congestion detection is notification by the congested router, but the system stills follow the TCP congestion control algorithm in the absence of that feedback. Because TCP only responds to one packet drop per round trip time, packets dropped by the filters will not cause endpoints to close their windows more quickly than they would in the face of a single packet drop. The reaction to congestion begins at the router with the packet filter installation.

This is in contrast to TCP with Explicit Congestion Notification (ECN)[6], which uses routers to notify endpoints of congestion, but applies the corrective action from the endpoint.
Thus ACC will react more quickly to congestion than TCP with ECN. TCP Reno [7] and TCP Vegas [8] are examples of variations on TCP congestion control that we do not implement here; there are others. They differ from the basic Jacobson TCP algorithm that we study by using different congestion detection mechanisms. We chose to extend the basic Jacobson TCP control algorithms to show that ACC can improve feedback performance. The ACC TCP dynamics are easier to see and understand in the context of the simpler algorithms. Extending our results to TCP Reno and TCP Vegas is future work.

Simulation Studies

Simulation studies of ACC TCP show that it increases endpoint's average throughput by up to 18% compared to standard TCP in the presence of busty traffic, and provides comparable performance in a stable network. The simulations were done using links with bandwidths between 1.5 and 10 Mb/s. The bandwidth-delay product was varied by increasing the delay on an uncongested link. All simulations were made using ns, a simulator produced by the University of California Berkeley, the Lawrence Berkeley National Labs, and the Virtual InterNet Testbed (VINT) project [9]. The simulator was extended to implement the ACC algorithms in the routers and endpoints, but not to allow full AN programmability. The modified simulator does not compile or interpret code from simulated packets. A Drop Tail router drops the last packet it receives when its buffer is full, like a glass overflowing; A RED router picks a random packet to discard[10]. A RED router also discards packets before it queue is full. The probability that an arriving packet causes a discard is proportional to the amount that the router's current queue length exceeds a configured minimum. All throughputs are based on the number of useful packets received by the destination endpoint. Retransmitted packets are not considered useful. We are primarily interested in showing that ACC supports high bandwidth-delay product networks. The simulations vary the link delay between 10 and 300 ms to increase the bandwidth-delay product. This range was selected because it is the high end of what the author perceives to be common round trip times in the Internet: the round trip time from the East coast of the US to the West is routinely 150 ms; the round trip time to a geosynchronous satellite is 250 ms. However, the important metric is the bandwidth delay product, replacing the 10 Mb/s link with a 100 Mb/sec link and varying the delay between 1 and 30 ms in the simulations will yield similar results (modulo adjusting the buffering at routers).

Stable Network Simulations

These simulations show that TCP and TCP ACC behave comparably in the absence of busty cross traffic. TCP ACC reduces the reaction time when the network changes state rapidly, but does not reduce the performance in the stable case [9].

Cross Traffic

These simulations demonstrate that ACC TCP reacts better than endpoint TCP to uncontrolled cross traffic. This simulation includes 10 bulk traffic sources and 7 cross traffic sources. The cross traffic sources have exponentially distributed on and off periods with means of 2.5 seconds and send at 100 Kb/s during their on periods. Cross traffic uses UDP, which does not react to congestion. Each cross traffic source sends for an average of 2.5 seconds at 100 Kb/s and then is quiet for 2.5 seconds. The goal is to model busty cross traffic that is not responsive to congestion, for example, busty video sources [9].

IV. CONCLUSION

This work has defined how AN technology of ACC has been used for Congestion Control, we discuss the mechanism and architecture of ACC in this paper.

We assert that in a well implemented Active Networking architecture, having routers edit endpoint traffic inside the network can improve performance during congestion, and does not reduce performance in stable networks. We have demonstrated that ACC systems derived from TCP show markedly better performance in simulation than pure TCP systems. The systems behave equivalently in stable networks, but the ACC system reacts faster to congestion. ACC systems are superior in maximizing throughput in networks of busty traffic sources. This work is preliminary in the sense that it has only shown the benefits of augmenting an existing feedback system.
We intend to adapt other feedback congestion controls to the ACC model, such as Dynamic Time Windows [12] and more complicated TCP variants like Vegas. These different systems will allow us to evaluate how ACC performs relative to other congestion control methods.

V. ACKNOWLEDGEMENT

This research is part of our PhD work. I am doing PhD in Computer Science. This project will help to find the better protocol out of much protocol to control the congestion. Congestion control is very necessary for future high bandwidth-delay network.

VI. REFERANCES

1. J-C Bolot and A. Shankar, "Dynamical Behavior of Rate Based Flow Control Systems," *Computer Communications Review,* vol. 20, no. 2, ACM SIGCOMM (Apr. 1990).

2. David L. Tennenhouse and David J. Wetherall, "Towards an Active Network Architecture," *Computer Communication Review,* vol. 26, no. 2, pp. 5-18, ACM SIGCOMM (April 1996).

3. Y. Yemini and S. de Silva, "Towards Programmable Networks," *IFIP/IEEE International Workshop on Distributed Systems Operations and Management,* L'Aquila, Italy (October 1996)
.
4. Samrat Bhattacharjee, Ken Calvert, and Ellen Zegura, "On Active Networking and Congestion,"
Technical Report GIT-CC-96-02, College of Computing, Georgia Tech, Atlanta, GA (1996).

5. Van Jacobson, "Congestion Avoidance and Control," *Proc. SIGCOMM Symposium on Communications Architectures and Protocols,* pp. 314-329, ACM SIGCOMM, Stamford, CA (Aug 16-19 1988).

6. Sally Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review,* vol. 24, no. 5, pp. 10-23, ACM (October 1994).

7. Van Jacobson, “Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno,” *Proceedings of the Eighteenth Internet Engineering Task Force*, pp. 363-366, University of British Columbia, Vancouver (September 1990).

8. Lawrence S. Brakmo, Sean W. O’Malley, and L. Peterson, “TCP Vegas: New techniques for congestion detection and avoidance,” *Proceedings of ACM SIGCOMM Symposium on Communications Architectures and Protocols*, pp. 24-35, ACM, London, UK (August 1994).

9. Steve McCanne, Sally Floyd, and Kevin Fall, *UCB/LBL Network Simulator NS* (1996,), available electronically from http://www-mash.cs.berkeley.edu/ns/ns.html.

10. Sally Floyd and Van Jacobson, “Random Early Detection gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413 (August 1993).

11. S. Floyd and Van Jacobson, “Traffic Phase Effects in Packet Switched Gateways,” *Computer Communications Review*, vol. 21, no. 2, pp. 26-42, ACM SIGCOMM (Apr. 1991).

12. T. Faber, L. Landweber, and A. Mukherjee, “Dynamic Time Windows: Packet Admission Control with Feedback,” *Proc. ACM Symposium on Communications Architectures and Protocols*, pp.
124-135, ACM, Baltimore, MD (August 1720,1992).