IJDCST @Feb-March Issue- V-2, I-3, SW-05
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Congestion Control Using Minimum Cost Flow Model to Reduce Packet Loss

[1] **Mohammad Imran ,[2] G J Sunny Deol.**
[1]**Mtech, Dept. of CSE, Sri Mittapalli College of Engineering,Tummalapalem,Gutur.**
[2]**Assistant Professor , Sri Mittapalli College of Engineering,Tummalapalem,Gutur.**

Abstract: Modern IP network services provide for the simultaneous digital transmission of voice, video, and data. These services require congestion control protocols and algorithms which can solve the packet loss parameter can be kept under control. So we propose a novel technique Stable Token-Limited Congestion Control (STLCC) is introduced as new protocols which appends inter-domain congestion control to TBCC and make the congestion control system to be stable. STLCC is able to shape output and input traffic at the inter-domain link with O(1) complexity. STLCC produces a congestion index, pushes the packet loss to the network edge and improves the network performance. Finally, the simple version of STLCC is introduced. This version is deployable in the Internet without any IP protocols modifications and preserves also the packet datagram. We formulate end-to-end congestion control as a global optimization problem. Based on this formulation, a class of minimum cost flow control (MCFC) algorithms for adjusting session rates or window sizes are proposed. Significantly, we show that these algorithms can be implemented at the transport layer of an IP network and can provide certain fairness properties and user priority options without requiring non-FIFO switches. Our proposed work also implements network performance from congestion present in the network.

Index Terms: Congestion Control, Congestion-Index, MCFC, Session rates.

I. INTRODUCTION

Congestion control in packet networks has proven to be a difficult problem, in general. However, this problem is particularly challenging in the Internet, due to very limited degrees of network observability and controllability. In order to accommodate rapid growth and proliferation, the design of the IP protocol and the requirements placed on individual sub networks have been kept at a minimum. Consequently, the main form of congestion control possible in the current Internet is end-to-end control of user traffic at the transport layer. As exemplified by TCP [Jac88], this control must be exerted using only the limited network observation that sessions can locally make, based on their own performance. The prevalent form of service discipline in the Internet is FIFO queuing, and control approaches based on more sophisticated service disciplines are not easily applicable.

Despite this vast literature, congestion control in telecommunication networks struggles with two major problems that are not completely solved. The first one is the time-varying delay between the control point and the traffic sources. The second one is related to the possibility that the traffic sources do not follow the feedback signal. This latter may happen because some sources are silent as they have nothing to transmit. Congestion control of the best-effort service in the Internet was originally designed for a cooperative environment. It is still mainly dependent on the TCP congestion control algorithm at terminals, supplemented with load shedding [1] at congestion links. This model is called the Terminal Dependent Congestion Control case.

Performing global optimization in involving users and links scattered throughout the Internet is a seemingly infeasible task. Although it is well-known that network optimization problems can be solved using distributed computations, algorithms proposed for this purpose [Gal77, Gol79, GG80] have relied on the presence of sophisticated network layer protocols, a luxury not available in the Internet for end-to-end congestion control. A significant accomplishment of this paper is in showing how such optimization is indeed feasible in the Internet. We even show that the current TCP congestion control, after some medication, belongs to the class of optimal algorithms that we describe. We refer to these optimal algorithms, either collectively or individually, as the minimum cost flow control (MCFC) algorithm. Two versions of the MCFC algorithm, referred to as the coarse realization and the exact realization, are explored in this paper. The coarse realization is geared towards implementation in the current Internet. This version of the algorithm, like TCP, relies on the end-to-end packet loss observations made by each session as indication of network congestion.

II. BACKGROUND WORK

A new and better mechanism for congestion control with application to Packet Loss in networks with P2P traffic is proposed. In this new method the edge and the core routers will write a measure of the quality of service guaranteed by the router by writing a digital number in the Option Field of the datagram of the packet. This is called a token. The token is read by the path routers and interpreted as its value will give a measure of the congestion especially at the edge routers. Based on the token number the edge router at the source's edge point will shape the traffic generated by the source, thus reducing the congestion on the path. In Token-Limited Congestion Control (TLCC) [9], the inter-domain router restricts the total output token rate to peer domains. When the output token rate exceeds the threshold, TLCC will decreases the Token-Level of output packets, and then the output token rate will decrease. Similarly to CSFQ and TBCC, TLCC uses also the iterative algorithm to estimate the congestion level of its output link, and requires a long period of time to reach a stable state. With bad parameter configuration, TLCC may cause the traffic to fall into an oscillated process. The window size of TCP flows will always increase when acknowledge packets are received, and the congestion level will increase at the congested link. At congestion times many flows will lose their packets. Then, the link will be idle and the congestion level will decrease. The two steps may be repeated alternately, and then the congestion control system will never reach stability. To solve the oscillation problem, the Stable Token-Limited Congestion Control (STLCC) is introduced. It integrates the algorithms of TLCC and XCP [10] altogether. In STLCC, the output rate of the sender is controlled according to thealgorithm of XCP, so there is almost no packet lost at the congested link. At the same time, the edge router allocates all the access token resource to the incoming flows equally. When congestion happens, the incoming token rate increases at the core router, and then the congestion level of the congested link will also increase. Thus STLCC can measure the congestion level analytically, allocate network resources according to the access link, and further keep the congestion control system stable.

III. PROPOSED METHODOLOGY

In STLCC, the output rate of the sender is controlled according to the algorithm of XCP, so there is almost no packet lost at the congested link. At the same time, the edge router allocates all the access token resource to the incoming flows equally. When congestion happens, the incoming token rate increases at the core router, and then the congestion level of the congested link will also increase. Thus STLCC can measure the congestion level analytically, allocate network resources according to the access link, and further keep the congestion control system stable. The dynamics of a network congestion control strategy can span multiple time scales. On the fastest time scale,

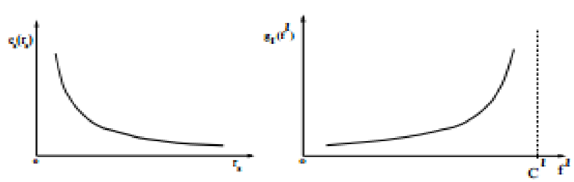

Figure 1. Cost of limiting the rate of session and congestion flow of process.

Congestion control should provide protection against sudden surges of traffic by quick reaction to buffer overloads. The reaction time in this type of control is, at best, in the order of one round-trip time since that is how fast news of

congestion can reach a source node and the response to it propagate back to the trouble spot. We refer to this type of congestion control as dynamic and to the corresponding time scale as short term. On a slower time scale, congestion

control could mean gradual but more steady reaction to the build-up of congestion, as perceived over a period involving tens or hundreds of round-trip times.

IV. REALIZATION OF THE MCFC ALGORITHM IN IP NETWORKS

In a network with a highly developed network layer, the task of computing congestion measures and distributing them to the corresponding sessions (or access points) can be performed by a specially designed network layer protocol in possible cooperation with the routing protocol. In the Internet or other IP networks, realization of the MCFC algorithm is more challenging since it should be done without explicit knowledge of the routing parameters and without expecting cooperation from the IP layer.

4.1 Exact Realization with Switch Cooperation.

Distributed execution of the MCFC algorithm (14) by various network sessions is possible if the sessions have a way of evaluating the corresponding congestion measures. The method we employ to relay congestion information to the sessions is both simple and concise. But more significantly, it relaxes the need for explicit knowledge about routing parameters, thereby enabling a realization of the algorithm in the Internet

4.2 Coarse Realization in the Current Internet.

We developa realizationfor theMCFC algorithm without using probe packets and requiring explicit congestion information from network switches In the absence of explicit congestion notification, the only observation a session can have about the network is through its own performance, i.e., the loss and delay of its own packets. We try to choose a form for the cost functions so that the resulting congestion measures can be best estimated through the available loss and delay information.

V. EXPERIMENTAL RESULTS

In this section,we discuss a limited set of simulations for the coarse realization of the MCFC algorithm. The goal of these simulations is to gain some understanding of the behaviorof the algorithm, rather than to provide a comprehensive study. In particular, no simulation results are provided for the exact realization of the algorithm. Two types of scenarios were developed for congestion control.

In the first experiment, sessions 4–10 are started and the network is allowed to reach a stable operating point, then session ✍ is activated at time t=1000 sec. Using n=50 and l=0.01,0.05,and 0.25, we observe the effect of

the step size in (50) and (51) on the stability and speed of convergence of the algorithm. We observe from Figure2 that as increases, session 1's window reaches its steady state value faster but the size of oscillations in the steady state increases.

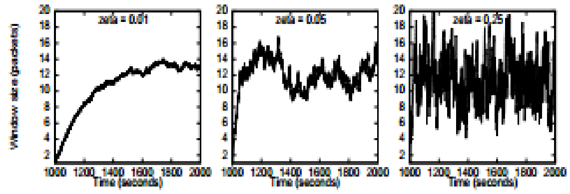

Figure 2. Experiment ✗ – Evolution ofthe window of session ✗ for different step sizes . Sessions 4–10 are active, but not shown.

In the second experiment, we have tried to combine the benefits of a large (fast rise to steady state) and a small (small oscillations). Hence we have used l=0.25 when a session is first activated and have switched to l1=0.01 at a later stage. The criterion that we have applied for this switching to takes place is the numberof losses experienced by a session. A threshold of 12 losses has been used in this simulation.

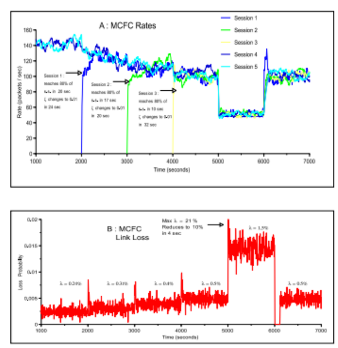

Figure 3. Rates of sessions (averaged over ✄ second intervals) for MCFC. .The loss probability of the link (averaged over ✄ second intervals) for MCFC.

The threshold value has to be chosen in a way such that the session's window reaches a given neighborhood of the steady state value before the switching takes place. It can be shown that, with Ab and Bm(Ws) chosen as in (50) and (51), the threshold, on the average, depends only on the initial value of and is independent of the final value of the window, the link capacity, or the buffer size.

VI. CONCLUSION

It integrates the algorithms of TLCC and XCP [10] altogether. In STLCC, the output rate of the sender is controlled according to the algorithm of XCP, so there is almost no packet lost at the congested link. At the same time, the edge router allocates all the access token resource to the incoming flows equally..We have developed a class of optimal algorithms for end-to-end congestion control at the transport layer of IP networks. The global optimization framework used for this purpose, allowed us to systematically address issues of fairness and user priority. Although the proposed algorithms do not require non-FIFO switches, we have shown that they can provide fair services to the users or help enforce certain priority options among them. These algorithms are realizable in both a coarse and an exact fashion, using implicit or explicit congestion information. Therefore, they facilitate an objective evaluation of the performance improvement that explicit congestion notification can bring to the Internet. Our proposed work also implements network performance in control flow process with low cost.

VII. REFERENCES

[1] Abhay K. Patekh, "A Generalized Processor Sharing Approach Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Trans. on Network, Vol. 1, No.3, June 1993.

[2] Sally Floyd, Van Jacobson, Link-sharing and Resource Management Models for Packet Networks, IEEE\ACM Transactions on Networking, Vol.3, No.4, 1995.

[3] John Nagle, RFC896 congestion collapse, January 1984. [14] Sally Floyd and Kevin Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, August 1999.

[5] V. Jacobson. "Congestion Avoidance and Control". SIGCOMM Symposium on Communications Architectures and Protocols, pages 314–329, 1988.

[6] http://www.isi.edu/nsnam/ns/

[7] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: The case of a single congested node," IEEE/ACM Trans. Networking, vol. 1, pp. 693–708, Dec. 1993.

[8] E. Altman, T. Basar, and R. Srikant, "Control methods for communica- tion networks," in Proc. 36th Conf. Decision and Control, San Diego, CA, 1997, pp. TA 31774–1809, TM3 2368– 2404, TP3 2903–2945.