Controlling Packet Loss at the Network Edges by Using Tokens

B.Suryanarayana[1], K. Bhargav Kiran[2]

[1]Research Scholar (PG), Dept of Computer Science and Engineering, Vishnu Institute of Engineering, Bhimavaram, India

[2]Assistant professor, Dept of Computer Science and Engineering, Vishnu Institute of Engineering, Bhimavaram, India

E-mail- Surya0530@gmail.com

Abstract— The Internet accommodates simultaneous audio, video and data traffic. It requires the Internet to guarantee the packet loss which at its turn depends very much on congestion controls. A series of protocols have been introduced to supplement the insufficient TCP mechanism controlling the network congestion's. CSFQ was designed as an open-loop controller to provide the fair best effort service for supervising the per-flow bandwidth consumption and has become helpless when the P2P flows started to dominate the traffic of the Internet. Token-Based Congestion Control (TBCC) is based on a closed-loop congestion control principles, which restricts token resources consumed by an end-user and provides the fair best effort service with O(1) complexity. As Self-Verifying Re-feedback and CSFQ, it experiences a heavy load by policing inter-domain traffic for lack of trusts. In this paper, 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 input and output traffic at the inter-domain link with O(1) complexity. STLCC produce a congestion index is pushes the packet loss to the network edge and improves the network performance. At last, 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.

Keywords—*TCP, Tokens, Network, Congestion Control Algorithm, Addressing, Formatting, Buffering, Sequencing, Flow Control, Error Control, Qos, Random Early Detection (RED).*

INTRODUCTION

Modern IP network services provide for the simultaneous digital transmission of video, voice and data. These services require congestion control protocols and algorithms which can solve the packet loss parameter can be kept under control. Congestion control is the cornerstones of packet switching networks. It should prevent congestion collapse it provide fairness to competing flows and optimize transport performance indexes such as throughput, loss and delay. The literature abounds in papers on this subject; there are papers on high-level models of the flow of packets through the network, and on specific network architectures.

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. 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.

Core-Stateless Fair Queuing (CSFQ) [3] set up an open- loop control system at the network layer, it inserts the label of the flow arrival rate onto the packet header at edge routers and drops the packet at core routers based on the rate label if congestion happens. CSFQ is first to achieve approximate fair bandwidth allocation among flows with O(1) complexity at core routers.

According to Cache Logic report, P2P traffic was 60% of all the Internet traffic in 2004, of which Bit-Torrent [4] was responsible for about 30% of the above, although the report generated quite a lot of discussions around the real numbers. In networks with P2P traffic, CSFQ can provide fairness to competing flow, but unfortunately it is not what end-users and operators really want. Token-Based Congestion Control (TBCC) [5] restricts the total token resource consumed by an end-user. So, no matter how many connections the end-user has set up, it cannot obtain extra bandwidth resources when TBCC is used.

In this paper a new and better mechanism for congestion control with application to Packet Loss in networks with P2P traffic was 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. This token is read by the path routers and interpreted as its value will give a measure of the congestion [2] especially at the edge router. Based on the token number the edge router at the source, 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 domain. 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.

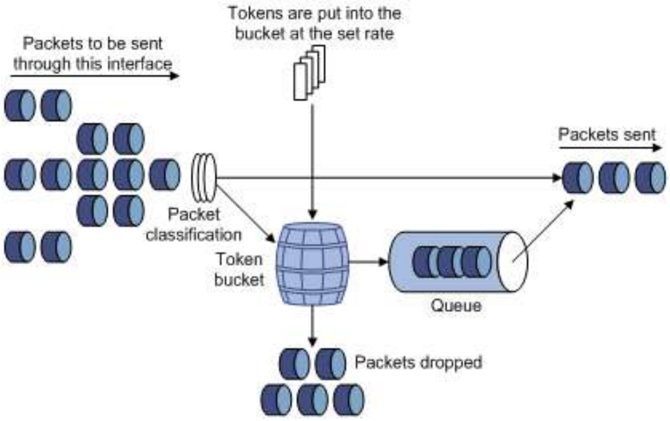

Tokens are put into the bucket at the set rate
Packets to be sent through this interface
Packets sent
Packet classification
Token bucket
Queue
Packets dropped

Fig 1. Architecture

2. RELATED WORK

The basic idea of peer- to- peer network is to have peers participate in an application level overlay network and operate as both A number of approaches for queue management at Internet gateways have been studied previously. Droptail gateways are used almost universally in the current Internet because of their simplicity. A droptail gateway drops an incoming packet only when the buffer becomes full, thus the providing congestion notifications to protocols like TCP. While simple to implement, it distributes losses among the flows arbitrarily [5]. Often results in the bursts losses from a single TCP connection, reducing its window sharply. Thus, the flow rate and consequently throughput for that flow drops. Tail dropping also results in multiple connections simultaneously suffering from losses leading to global synchronization [6]. Random early detection (RED) addresses some [11][12] of the drawbacks of droptail gateways. The RED gateway drops incoming packets with a dynamically computed probability when the exponential weighted moving average queue size avg q exceeds a threshold. In [6], the author does per-flow accounting maintaining only a single queue. It is suggest changes to the RED algorithm to ensure fairness and to penalize the misbehaving flow. It puts a maximum limit on the number of packets a flow can have in the queue.

Besides it also maintains the per flow queue use. Drop or accept decision for an incoming packet is then based on the average queue length and the state of that flows. It also keeps track of the flows which consistently violate the limit requirement by maintaining a per-flow variable called as strike and penalizes those flows which have a high value for strike. It is intended that this variable will becomes high for non- adaptive flows and so they will be penalized aggressively. It has been shown through simulations [7] that FRED fails to ensure the fairness in many cases. CHOKE [8] is an extension to RED protocols. It does not maintain any per flow state and works on the good heuristic that a flow sending at a high rate is likely to have more packets in the queue during the time of the congestion. It decides to drop a packet during congestion if in a random toss, it finds another packet of the same flow. In [9], the authors establish how rate guarantees can be provided by simply using buffer management. They show that the buffer management approach is indeed capable of providing reasonably accurate rate guarantees and the fair distribution of excess resources.

3. Core Stateless Fair Queuing

In the proposed work, a model called the Terminal Dependent Congestion Control case which is a best-effort service in the Internet that was originally designed for a cooperative environment which is the congestion control but still it is mainly dependent on the TCP congestion control algorithm at terminal, supplemented with load shedding at[13][14] congestion links is shown in Figure 2.

In high speed network Core Stateless Fair Queuing (CSFQ) is enhanced to fairness set up an open- loop control system at the network layer, which insert the label of the flow arrival rate onto the packet header at edge routers and drops the packet at core routers based on the rate label if congestion happens. At the core routers CSFQ is the first to achieve approximate fair bandwidth allocation among flows with O (1) complexity.

CSFQ can provide fairness to competing flows in the networks with P2P traffic, but unfortunately it is not what end-users really want. By an end user Token Based Congestion Control (TBCC) restricts the total token resource consumed. It cannot obtain extra bandwidth resources when TBCC is used so no matter how many connections the end user has set up. The Self Verifying CSFQ tries to expand the CSFQ across the domain border. It randomly selects a flow,[15] then re-estimates the flow's rate, and the checks whether the re-estimated rate is consistent with the label on the flow's packet. Consequently Self-Verifying CSFQ will put a heavy load on the border router and makes the weighted CSFQ null as well as avoid.

The congestion control architecture re-feedback, which aims to provides the fixed cost to end-users and bulk inter-domain congestion charging to network operator. Re-feedback not only demands very high level complexity to identify the malignant end user, but it is difficult to provide the fixed congestion charging to the inter domain interconnection with low complexity. There are three types of inter domain interconnection polices: the Internet Exchange Points[16], the private peering and the transits. In the private peering polices, the Sender Keep All (SKA) peering arrangements are those in which the traffic is exchanged between two domains without mutual charges. As Re-feedback is based on the congestion charges to the peer domain, it is difficult for re-feedback to support the requirements of SKA.

The modules of the proposed work are:

- NETWORK CONGESTION
- STABLE TOKEN LIMIT CONGESTION CONTROL (STLCC) TOKEN
- CORE ROUTER
- EDGE ROUTER

Network Congestion: Congestion occurs when the number of packets being transmitted through the network crosses the packet handling capacity of the networks. Congestion control aims to keep number of packets below the level at which performance falls off dramatically.

Stable Token Limit Congestion Control (STLCC): STLCC is able to shape output and input traffic at the inter domain link with O(1) complexity. STLCC produce a congestion index, pushes the packet loss to network edge and improves the overall network performance. To solve the oscillation problems, the Stable Token-Limited Congestion Control (STLCC) is also introduced. It integrate the algorithms of TLCC and XCP [10] altogether. In STLCC, output rate of the sender is controlled using the algorithm of XCP, there is almost no packet lost at the congested link. At the same time, the edge router allocates all the access token resources to the incoming flow equally. When congestion happens, the incoming token rate increases at the core router, and the congestion level of the congested link will also increased as well. Thus STLCC can measure the congestion level analytically, and then allocates network resources according to the

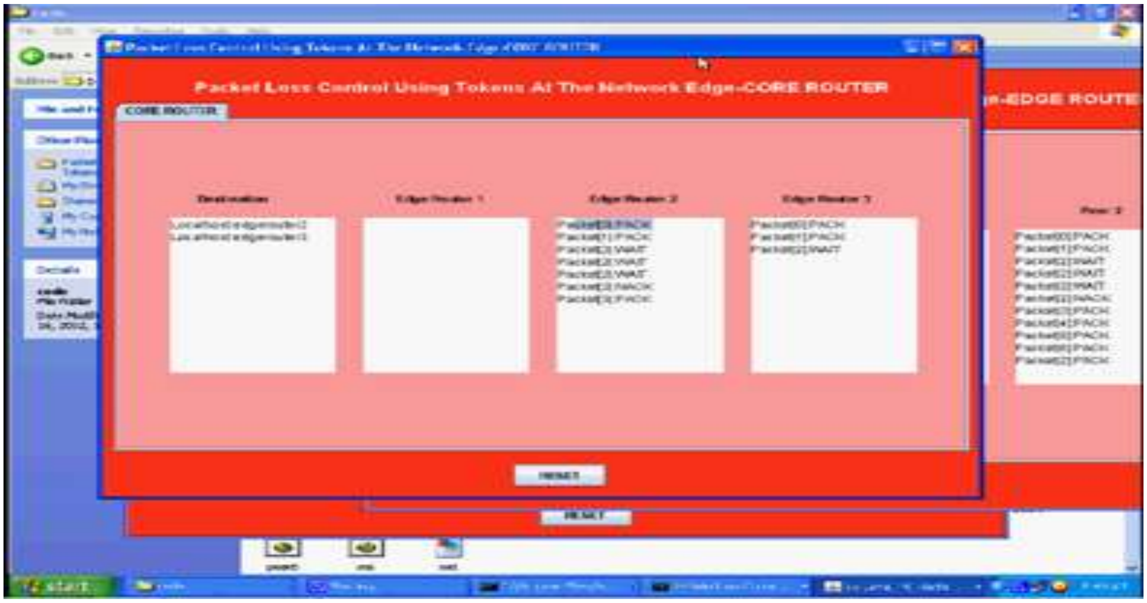
Token: A new and better mechanism for the congestion control with application to Packet Loss in networks with P2P traffic is proposed. In this method the edge and the core routers will write a measure of the quality of service guaranteed by the router by writing the digital number in the Option Field of the datagram of the packet. This is called as token. The token is read by the path routers and then interpreted as its value will give a measure of the congestion especially at the edge routers. Based on the token numbers, the edge router at the source, it reducing the congestion on the path.

Core Router: A core router is a router designed to operate in the Internet Backbone (or core). To fulfill this role, a router must be able to support multiple telecommunications interfaces of the highest speed in use in the core Internet and must be able to forward the IP packets at full speed on all of them. It must also supports the routing protocols being used in the backbone. A core router is distinct from the edge routers.

Edge Router: Edge routers sit at the edge of a backbone network and connect to the core routers. Then the token is read by the path routers and then interpret as its value will give a measure of the congestion especially at the edge routers. Based on the token number of the edge router at the source, it reducing the congestion on the path.

4. RESULTS

Packets of Edge Router:



Packet Loss Control Using Tokens At The Network Edge-CORE ROUTER

CORE ROUTER

Destination
Localhost:edgerouter2
Localhost:edgerouter3

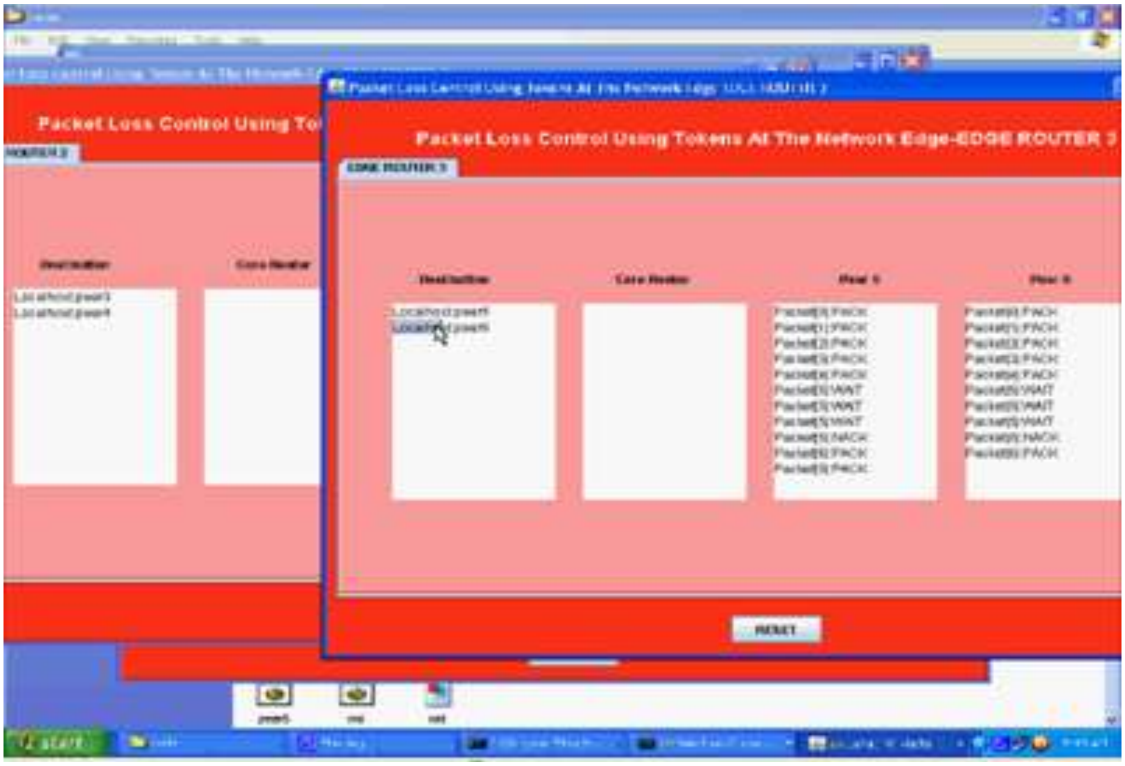
Edge Router 1

Edge Router 2
Packet[0]:PACK
Packet[1]:PACK
Packet[2]:WAIT
Packet[2]:WAIT
Packet[2]:WAIT
Packet[2]:NACK
Packet[3]:PACK

Edge Router 3
Packet[0]:PACK
Packet[1]:PACK
Packet[2]:WAIT

Peer 2
Packet[0]:PACK
Packet[1]:PACK
Packet[2]:WAIT
Packet[2]:WAIT
Packet[2]:WAIT
Packet[2]:NACK
Packet[3]:PACK
Packet[4]:PACK
Packet[5]:PACK
Packet[6]:PACK
Packet[7]:PACK

Edge Router3:



Packet Loss Control Using Tokens At The Network Edge-EDGE ROUTER 3

CORE ROUTER 3

Destination
Localhost:peer5
Localhost:peer6

Core Router

Peer 5
Packet[0]:PACK
Packet[1]:PACK
Packet[2]:PACK
Packet[3]:PACK
Packet[4]:PACK
Packet[5]:WAIT
Packet[5]:WAIT
Packet[5]:WAIT
Packet[5]:NACK
Packet[5]:PACK
Packet[6]:PACK

Peer 6
Packet[0]:PACK
Packet[1]:PACK
Packet[2]:PACK
Packet[3]:PACK
Packet[4]:PACK
Packet[5]:WAIT
Packet[5]:WAIT
Packet[5]:WAIT
Packet[5]:NACK
Packet[6]:PACK

CONCLUSION:

The architecture of Token-Based Congestion Control (TBCC), which provides fair bandwidth allocation to end-users in the same domain will be introduced. It evaluates two congestion control algorithms CSFQ and TBCC. In this STLCC is presented and the simulation is designed to demonstrate its validity. It presents the Unified Congestion Control Model which is the abstract model of STLCC, CSFQ and Re-feedback. Finally, conclusions will be given. To inter-connect two TBCC domains, then the inter-domain router is added to the TBCC system. To support the SKA arrangements, the inter-domain router should limit its output token rate to the rate of the other domains and police the incoming token rate from peer domains.

REFERENCES:

[1] Andrew S. Tanenbaum, Computer Networks, Prentice-Hall International, Inc.

[2] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance, ACM/IEEE Transactions on Networking, August 1993.

[3] Ion Stoica, Scott Shenker, Hui Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks", In Proc. of SIGCOMM, 1998.

[4] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In Proc. of SIGCOMM, 2004.

[5] Zhiqiang Shi, Token-based congestion control: Achieving fair resource allocations in P2P networks, Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference.

[6] I. Stoica, H. Zhang, S. Shenker ， Self-Verifying CSFQ, in Proceedings of INFOCOM, 2002.

[7] Bob Briscoe，Policing Congestion Response in an Internetwork using Refeedback， In Proc. ACM SIGGCOMM05, 2005.

[8] Bob Briscoe,Re-feedback:Freedom with Accountability for Causing Congestion in a Connectionless Internetwork, http://www.cs.ucl.ac.uk/staff/B.Briscoe/project s /e2ephd/ e2ephd_y9_cutdown_ appxs.pdf

[9] Zhiqiang Shi, Yuansong Qiao, Zhimei Wu, Congestion Control with the Fixed Cost at the Domain Border, Future Computer and Communication (ICFCC),2010.

[10] Dina Katabi, Mark Handley, and Charles Rohrs, "Internet Congestion Control for Future High Bandwidth-Delay Product Environments." ACM Sigcomm 2002, August 2002.

[11] 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.

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

[13] 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.

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

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