International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 10 Issue 1 –OCTOBER 2014.

A Tokenized Mechanism of Controlling Packet Loss over the Network

N.Sravanthi

Mtech-CSE, Prasad Engineering College, Vikas Nagar,Jangoan,Warangal District, Andhra Pradesh, India

rootas.sravanthi@gmail.com

Abstract --- The modern internet services are responsible for increasing network traffic by means of audio ,video and data traffic. It is necessary to generate internet applications which can control the packet losses and controls the congestion problem over the internet. A set of protocols are developed in alternative to inefficient TCP mechanism in controlling the data congestion over the internet. CSFQ algorithm was developed as an open loop controller that provides the best services for monitoring the per flow bandwidth consumption but failed when the p2p started dominating the traffic over the internet. Token-Based Congestion Control (TBCC) limits the token resources consumed by an end-user and provides the fair best effort service with O (I) complexity as it is based on a closed-loop congestion control principle. It experiences a heavy load by policing inter-domain traffic for lack of trust as it is self-verifying CSFQ and re-feedback. In this paper, we introduce a new protocol Stable Token-Limited Congestion Control (STLCC) which appends inter-domain congestion control to TBCC and make the congestion control system to be stable. STLCC produces a congestion index and pushes the packet loss to the network edge and improves the network performance. 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 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.

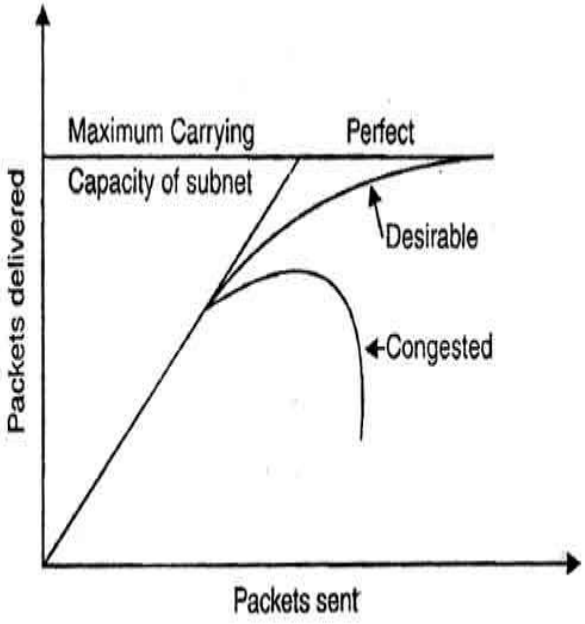
I. INTRODUCTION

Networked control systems (NCSs), in which nodes communicate over communication networks, have attracted lots of researchers' attention. Since networks-based control gives rise to many advantages including low cost, easy maintenance, and flexible system structure, the successful application of NCSs can be found in a wide range of areas such as industrial automation, intelligent transportation system, and smart grid. However, packet disordering and packet dropout inevitably exist in the transmission of signals. They are recognized to be two main causes for performance deterioration or even instability of NCSs; hence, considerable research has been done. So far, the majority of NCSs research has focused on controller design to provide sufficient stability conditions for NCSs with packet loss. A lot of effort has also been taken for modeling NCSs in presence of packet losses as asynchronous dynamic systems or Markovian jumping systems. With further study on packet loss, some effective compensation strategies for packet loss that occurred during communication are proposed to improve control performance of NCSs. Predictive control is a typical method with wich the control prediction generator provides a set of future control predictions to enable the closed-loop system to achieve the desired control performance leading to removing the effects of data dropout. Another typical compensation methodology for packet loss is observer-based state estimation. In addition, proposed a packet dropout-based compensation scheme, namely, the latest control signal is used for compensation if the ideal control input is missing. However, note that, in all of the aforementioned literature, packet disordering is not considered, but packet disordering and packet loss coexist in packets delivered network communication.

Packet disordering means that a packet sent earlier may arrive at the destination node later or vice versa. Packet disordering of NCSs has drawn an increasing attention. The packets that arrived late at control nodes were discarded, and stability and compensation control were investigated. However, the packet disordering is not described clearly. The sampling instants of received signals were compared to describe packet disordering, and stability analysis and synthesis were studied. Some literature using the similar method can be found in the existing reported results. Recently, proposed an active compensation for packet disordering; that is, the latest control actions applied to the plant are available by comparing the time stamps of packets. The so called compensation method has been also presented in, where the latest signals are executed by the plant by defining an operator constructing a mapping between the newest signals and packet displacement values. However, note that a situation where no new control actions arrive at the actuator may occur due to packet disordering and packet loss during a sampling interval. In this case, it is critical how to control the plant. To the best of the authors' knowledge, this problem has been not fully investigated to date, which motivates this work for proposing a new compensation scheme.

The specific problem addressed in this paper is the compensation control when the newest signal is not available for NCSs due to the packet disordering and packet loss. The highlighted method is that control inputs are determined by defining some operators associated with packet displacements; that is, the latest control input is chosen when no new signals arrive at the actuator during the sampling interval ; otherwise, the newest signal controls the plant. After that, a Markovian jumping model of NCSs is put forward. Stability analysis and the controller synthesis are thoroughly investigated and the adaptive controller design is obtained in terms of linear matrix inequalities (LMI). A numerical example is provided to demonstrate the effectiveness of the proposed approach.

Congestion is an important issue that can arise in packet switched network. Congestion is a situation in Communication Networks in which too many packets are present in a part of the subnet, performance degrades. Congestion in a network may occur when the load on the network (i.e. the number of packets sent to the network) is greater than the capacity of the network (i.e. the number of packets a network can handle.). In other words when too much traffic is offered, congestion sets in and performance degrades sharply.



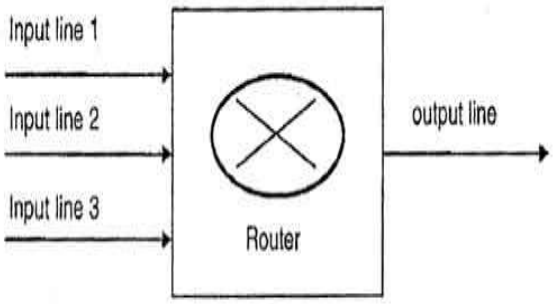
Concept of Congestion

II. CAUSING OF CONGESTION:

The various causes of congestion in a subnet are:

1.The input traffic rate exceeds the capacity of the output lines. If suddenly, a stream of packet start arriving on three or four input lines and all need the same output line. In this case, a queue will be built up. If there is insufficient memory to hold all the packets, the packet will be lost. Increasing the memory to unlimited size does not solve the problem. This is because, by the time packets reach front of the queue, they have already timed out (as they waited the queue). When timer goes off source transmits duplicate packet that are also added to the queue. Thus same packets are added again and again, increasing the load all the way to the destination.

2.The routers are too slow to perform bookkeeping tasks (queuing buffers, updating tables, etc.).



Data from three input lines at same time

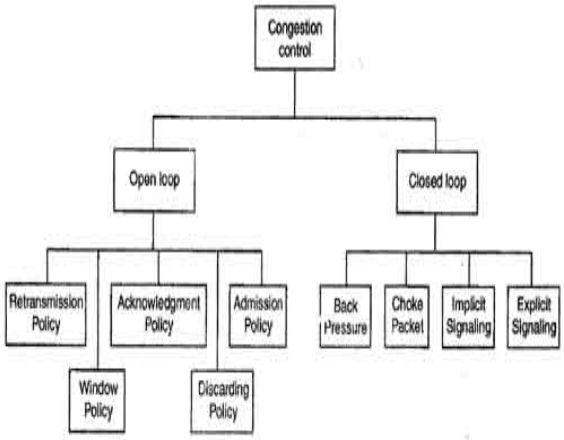
3. The routers' buffer is too limited.

4. Congestion in a subnet can occur if the processors are slow. Slow speed CPU at routers will perform the routine tasks such as queuing buffers, updating table etc slowly. As a result of this, queues are built up even though there is excess line capacity.

5. Congestion is also caused by slow links. This problem will be solved when high speed links are used. But it is not always the case. Sometimes increase in link bandwidth can further deteriorate the congestion problem as higher speed links may make the network more unbalanced. Congestion can make itself worse. If a route!" does not have free buffers, it start ignoring/discarding the newly arriving packets. When these packets are discarded, the sender may retransmit them after the timer goes off. Such packets are transmitted by the sender again and again until the source gets the acknowledgement of these packets. Therefore multiple transmissions of packets will force the congestion to take place at the sending end.

How to correct the Congestion Problem:

Congestion Control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.



Types of Congestion Control Methods

These two categories are:

1. Open loop
2. Closed loop

Open Loop Congestion Control

In this method, policies are used to prevent the congestion before it happens. Congestion control is handled either by the source or by the destination. The various methods used for open loop congestion control are:

1. Retransmission Policy

• The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted.

• However retransmission in general may increase the congestion in the network. But we need to implement good retransmission policy to prevent congestion.

• The retransmission policy and the retransmission timers need to be designed to optimize efficiency and at the same time prevent the congestion.

2. Window Policy

• To implement window policy, selective reject window method is used for congestion control.

• Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this duplication may make congestion worse.

• Selective reject method sends only the specific lost or damaged packets.

3. Acknowledgement Policy

• The acknowledgement policy imposed by the receiver may also affect congestion.

• If the receiver does not acknowledge every packet it receives it may slow down the sender and help prevent congestion.

• Acknowledgments also add to the traffic load on the network. Thus, by sending fewer acknowledgements we can reduce load on the network.

• To implement it, several approaches can be used:

1. A receiver may send an acknowledgement only if it has a packet to be sent.
2. A receiver may send an acknowledgement when a timer expires.
3. A receiver may also decide to acknowledge only *N* packets at a time.

4. Discarding Policy

• A router may discard less sensitive packets when congestion is likely to happen.

• Such a discarding policy may prevent congestion and at the same time may not harm the integrity of the transmission.

5. Admission Policy

• An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.

• Switches in a flow first check the resource requirement of a flow before admitting it to the network.

• A router can deny establishing a virtual circuit connection if there is congestion in the "network or if there is a possibility of future congestion.

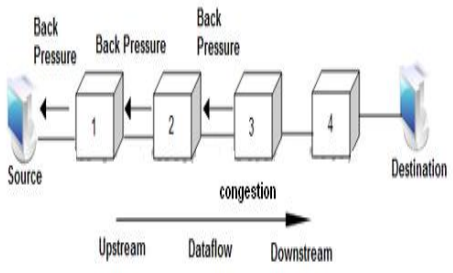
Closed Loop Congestion Control

• Closed loop congestion control mechanisms try to remove the congestion after it happens.

• The various methods used for closed loop congestion control are:

1. Backpressure

• Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow.



Backpressure Method

• The backpressure technique can be applied only to virtual circuit networks. In such virtual circuit each node knows the upstream node from which a data flow is coming.

• In this method of congestion control, the congested node stops receiving data from the immediate upstream node or nodes.

• This may cause the upstream node on nodes to become congested, and they, in turn, reject data from their upstream node or nodes.

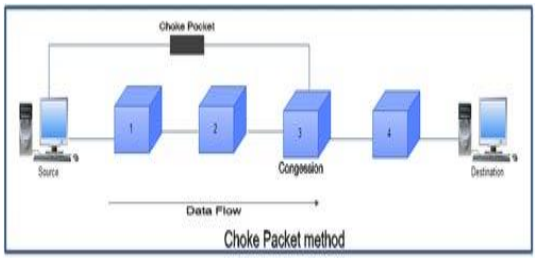
• As shown in fig node 3 is congested and it stops receiving packets and informs its upstream node 2 to slow down. Node 2 in turns may be congested and informs node 1 to slow down. Now node 1 may create congestion and informs the source node to slow down. In this way the congestion is alleviated. Thus, the pressure on node 3 is moved backward to the source to remove the congestion.

2. Choke Packet

• In this method of congestion control, congested router or node sends a special type of packet called choke packet to the source to inform it about the congestion.

• Here, congested node does not inform its upstream node about the congestion as in backpressure method.

• In choke packet method, congested node sends a warning directly to the source station *i.e.*the intermediate nodes through which the packet has traveled are not warned.



3. Implicit Signaling

• In implicit signaling, there is no communication between the congested node or nodes and the source.

• The source guesses that there is congestion somewhere in the network when it does not receive any acknowledgment. Therefore the delay in receiving an acknowledgment is interpreted as congestion in the network.

• On sensing this congestion, the source slows down.

• This type of congestion control policy is used by TCP.

4. Explicit Signaling

• In this method, the congested nodes explicitly send a signal to the source or destination to inform about the congestion.

• Explicit signaling is different from the choke packet method. In choke packed method, a separate packet is used for this purpose whereas in explicit signaling method, the signal is included in the packets that carry data .

• Explicit signaling can occur in either the forward direction or the backward direction .

• In backward signaling, a bit is set in a packet moving in the direction opposite to the congestion. This bit warns the source about the congestion and informs the source to slow down.

• In forward signaling, a bit is set in a packet moving in the direction of congestion. This bit warns the destination about the congestion. The receiver in this case uses policies such as slowing down the acknowledgements to remove the congestion.

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. Congestion control is therefore, the cornerstone of packet switching networks [28]. It should prevent congestion collapse, provide fairness to competing flows and optimize transport performance indexes such as throughput, delay and loss. 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 architecture. 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.

III. TOKEN-BASED CONGESTION CONTROL METHOD FOR THE INTERNET

This invention relates to a congestion control technology in a computer network, specifically, a congestion control method for a packet-switched network. Currently routers used in the Internet are weak in congestion control. Packet loss information is the only network congestion signal provided to the terminal. Network traffic is controlled by the terminal at the transport layer or the application layer. The networks are lack of appropriate policing ability. Recent breakpoint-resume and multi-threaded downloading technologies present challenges to the stability and the fairness of the Internet. Core-Stateless Fair Queuing (CSFQ) is a technique for enhancing congestion control at the network layer. In CSFQ, edge routers measure and label the flow rate in the packet header. Packets having the same source and destination IP addresses are usually classified into a same flow. Core routers adaptively adjust the parameter of the fair bandwidth according to the congestion state at the output port. The core router can calculate reception probability of a packet using the fair bandwidth and the flow rate in the packet header. The core routers adaptively adjust the parameter of the fair bandwidth according to the congestion state at the output port. The edge routers can therefore provide differentiated services using bandwidth-weighted flow rates based on the access bandwidth. The CSFQ technique, however, has several shortcomings: firstly, the technique is unable to provide fair services in the presence of Bit-Torrent (BT) downloading. A single BT downloading includes multiple destinations. CSFQ cannot properly restrict the total throughput of a BT download because CSFQ identifies a flow only by its source and destination addresses. A BT download can thus increase its own throughput by increasing the number of sessions at the

sacrifices of the throughputs of other traditional applications. BT downloads thus affect the fairness and the stability of the Internet. Moreover, due to lack of trust between different domains, flows are measured and re-labeled at the domain border, which becomes a performance bottleneck. Furthermore, for inter-domain measurements and re-mark rate of conversation, the bandwidth weights of the flows are inevitably lost. The bandwidth weight is in effect only inside its domain and cannot be credibly passed on to other domains, which limits the flexibility of CSFQ.

IV. 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. 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. Random early detection (RED) addresses some 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, 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 that FRED fails to ensure the fairness in many cases. CHOKE 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, 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 congestion links. 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, 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.

V. 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.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grants No. 60873258, and by Chinese National 973 Fundamental Research Program under Grants No.2007CB307106.

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/ projects/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/ IONESCU et al.: PACKET LOSS CONTROL USING TOKENS 1393