Load Aware Congestion Detection Technique in Internet

Ajay Shankar Singh[1], Dr. E G Rajan[2] and Dr. Manish Prateek[3]

[1] University Of Petroleum & Energy Studies
Dehradun-248007, India

[2] Professor Of Signal Processing
Chairman, Pentagram Group Of Companies
Hyderabad – 500 028, India

[3] Professor & Head, Dept. of CSE-COES
University Of Petroleum & Energy Studies
Dehradun-248007, India

Abstract

The Internet includes various ranges of applications such as multimedia, voice and data but it endures rigorous network congestion. Further, congestion detection is an essential preventive measure must be taken to provide congestion-free Internet. To stipulate proficient technique for congestion detection, in this paper, we propose a load aware congestion detection technique in the Internet. To be aware of congestion in the network, the router estimates the traffic load based on average packet arrival rate and queue occupancy rate. By comparing traffic load with threshold values, congestion notification bit is marked in packets. Before transmitting congestion information to the source, the router differentiates dedicated and shared link by means of link correlation factor. For dedicated link, congestion information is directed only to the corresponding source. Just in case of shared link, congestion information is transmitted to all sources along that link. We show the performance of our technique through simulation results. Our technique efficiently detects and prevents the congestion in the Internet.

Keyboards: *Congestion Detection, Congestion Control, Internet Congestion, Congestion Avoidance, Natural Logic, Fuzzy Logic*

1. Introduction

1.1 Internet Congestion

The Internet architecture shows prodigious example for scalability over the past fifty years. [1] Delay that occurs between users and network resources are remain unnoticed in communication networks. In the framework of the Internet, end- end congestion control algorithms are greatly interfered by delay that occurs in the network. The communication delays are heterogeneous in nature because the forward delay is different from feedback delay. The delay between user and resource is defined as forward delay. [2]

The overload of datagrams at switching points endures more delay and it initiates the circumstance of congestion in the Internet. [3] In other words, congestion in internet takes place, when the requested resource exceeds the available resource. Congestion incurs congestion collapse, delay in data delivery, data loss and waste of resources as a result of packet dropping. [4]

In general, bottlenecks that are endures by data are classified into two types based on the place it occurs. The classification is server side bottleneck and intermediate links and nodes bottleneck.

Server side bottleneck arise when a huge number of clients are connected to a server concurrently. The problem of handling simultaneous connection is known as concurrency. After certain limit is achieved, unless and until old connection is terminated, no new connection can be established. [3]

Mostly, intermediate links and nodes congestion happens in the Internet in the peak hours of Internet usage. During this time, the intermediate links and nodes controls packet of different connections. [3]

1.2 Causes of Congestion

Multiple causes can trigger congestion in the network. Increase in the number of users and instantaneous bandwidth demands leads to congestion. Further, multihop nature of the network is also one of the reasons for congestion. [3] In addition, overload and heterogeneity reasons congestion. [5] Finally, ‘any- to- any’ and ‘any-

to- many' traffic profiles (i.e) frequent topology changes initiates network congestion.

Network performance and Quality of Service (QoS) are degraded by network congestion. [6] When the network fails to provide end- to- end congestion control it lead to congestion collapse. In the Internet, congestion collapse is the outcome of congested links and this situation drops the sending packets rather than forwarding. [7]

Congestion control is a mechanism, which is used to rectify and control the issue of congestion. One of the congestion control schemes is drop tail scheme. When the buffer is full, this scheme drops the packets that are in queue tail. Another familiar mechanism is Active Queue Management (AQM) scheme. It is a reactive mechanism and it drops packets probabilistically before queue fills and when buffer overflow it responds to congestion. [1]

1.3 Congestion Detection

In congestion control mechanism, the prominent role is played by a congestion detection scheme. The congestion detection scheme must be accurate and efficient. Further congestion detection technique must meet low cost in terms of energy and computation complexity. [8] In existing Internet, only after the packet is dropped, the congestion is detected by the TCP transport protocol. [9]

In Internet, packet drop indicates that the network is already congested. Consequently, additional sources may also suffer from packet drops. As a result of packet drops, substantial number of packets must be retransmitted; other sources may regulate their congestion window size to alleviate congestion, these consequences may lead to the under utilization of network. The above-mentioned drawbacks could be eliminated if the network congestion is detected proactively. [18]

In standard TCP protocol, packet drop is the indicator of congested network. The packet drops can be discovered by sending triple-acknowledgement (triple-ack) message from the receiver and by packet timeout error. [18]

1.4 Problem Identification

Congestion incurs congestion collapse, delay in data delivery, data loss and waste of resources as a result of packet dropping and retransmission of packets. All these drawbacks of the congestion could be eliminated if the network congestion is detected preemptively. Further, there are enormous number of technique exists in the literature for congestion control. However, except some standard queuing mechanism, it is hard to find works for congestion detection in the Internet. To keep away the above-mentioned problems, in this paper we propose a load aware congestion detection technique in Internet.

Our paper is organized as follows; section-2 includes the related work, our proposed technique is described in section-3, section-4 contains simulation results and section-5 concludes the paper.

2. Related Work

K. Srinivas and A. A. Chari [10] have proposed an energy efficient congestion detection and control scheme called ECDC. Their ECDC detects the degree of congestion at relay hop level with maximum accuracy. Their proposed mechanism is separated from other activities of the MAC layer like link reliability analysis and buffer size analysis. Their congestion controlling mechanism operates in two ways. First, congestion is detected energy efficiently and secondly, it utilizes the resources optimally.

Michele C. Weigle et al. [11] have presented a new end-to- end congestion detection and reaction mechanism for TCP. Their approach utilizes one way transmit times of TCP. Their proposed design is known as Sync- TCP, it places timestamps in TCP headers to measure variations of one- way transit times of TCP segments. Their mechanism enhances the changes in forward-path queuing delay to detect congestion. Further, this measurement is used to transmit early congestion notification.

Lingu Su and Jennifer C. Hou et al. [12] have introduced an active queue management scheme termed as Average Rate Early Detection (ARED) for Internet congestion control. Their ARED scheme calculates the average packet enqueue rate as a congestion indicator and signals the end hosts of incipient congestion. They have developed this scheme with the objective to reduce the packet loss ratio and to improve the link utilization. They have also enhanced differentiated service architecture at core routers to implement queue management mechanism. They also run down DiffServ architecture in the context of assured services. (AS)

Pascal Anelli et al. [13] have proposed Implicit Congestion Notification (ICN) algorithm to discover the congestion events that occurs in the network. Their ICN algorithm is implemented independently inside the Transport Congestion Events Detection (TCED) framework. Their TCED framework can be used either at the border of an autonomous system or conjointly within a TCP stack as a sub layer. Further, their algorithm is based on two feasible assumptions as, a delay or a time stamp to validate a loss following retransmission and the acknowledgement path.

C.N. Nyirenda and D.S. Dawoud [14] have proposed two online self-learning and organization structures for Fuzzy Logic Congestion Detection (FLCD) algorithm. This FLCD algorithm is used for congestion detection, it joins the best characteristics of both traditional Active Queue Management (AQM), and fuzzy logic based AQM algorithms. Their proposed online structures enable the FLCD algorithm to learn the system conditions and to adjust the fuzzy rule base in accordance with prevailing conditions. Their proposed structures include an RTT based sampling mechanism and a self-learning and adaptation mechanism.

Ivan D. Barrera et al. [15] have explored a statistical technique called Maximum Likelihood Ratio Test (MLRT) to be applied with Active Queue Management (AQM). Their proposed statistical technique is used to detect the circumstance of congestion. It provides the relationship between the observed marking state, the observed queue occupancy and likelihood of congestion. Further, it gives additional information of the level of congestion of the router's outbound link. Their technique either uses binary point of detection or reactive schemes that wait for the router's queue to be congested before reacting.

3. Network Aware Congestion Detection in Internet

3.1 Overview

In this paper, we propose a load aware congestion detection technique in Internet. As per our technique, the router periodically calculates packet arrival rate and queue occupancy. Based on these factors, traffic load of a router is evaluated. The estimated traffic load is compared with two predefined threshold values as min_{th} and $max_{th.}$ When the traffic load exceeds the value of max_{th} , it is dropped by the router. Packets that have values between min_{th} and max_{th} are marked with ECN bit. Simultaneously, the router estimates cumulative value of link queuing delay (LQ_D). It further estimates link correlation factor to determine whether the link is shared or dedicated. For shared link, the congestion information is sent to all sources in the link. In the case of dedicated link, the congestion information is sent only to the corresponding source.

3.2 Estimation of Metrics

3.2.1 Average Packet Arrival Rate Estimation

Average packet arrival rate is calculated using Exponential Weighted Moving Average (EWMA) algorithm. Let A_R be the arrival rate of packets. It is the reciprocal of packets inter arrival time (IA), which is the time interval of two consecutive packets. The arrival rate of packet i is estimated as follows, [16]

$$A_R = 1/IA \quad (1)$$

Where, IA represents inter arrival time of packets and it can be measured as below,

$$IA_i = (1- \lambda)\, IA_L + \lambda\, (A_n - A_{n-1}) \quad (2)$$

In the above equation (2), IA_L denotes the inter arrival time of finally estimated packet. A_{n-1} and A_n are the arrival time of two consecutive last packets. λ is used as weighting factor and takes value between 0 and 1. In our approach, it is set to 0.7.

3.2.2 Assessment of Queue Occupancy

In network router, queue occupancy is estimated using Gaussian approximation of sum of congestion windows. Queue occupancy (QO) of flow i at time t is measured as follows, [15]

$$QO_i(t) = CW_i(t) - \overline{RTT_i} \times P_i - \sigma_i \quad (3)$$

Where, $QO_i(t)$ represents number of packets occupied in the queue at time t of flow i, *CW* denotes the congestion window, number of packets that are currently in the link is symbolized by $\overline{RTT} \times P_i$ and σ_i is number of packets dropped.

3.2.3 Traffic Load Evaluation

Traffic load is deliberated by multiplying queue occupancy rate and packet arrival rate. The traffic load of a router is measured as below,

$$L = A_R \times QO_i(t) \quad (4)$$

Whereas, L denotes traffic load of the router, A_R is the average packet arrival rate and $QO_i(t)$ represents queue occupancy rate of the router.

3.2.4 Queuing Delay Estimation

Queuing delay is the waiting time of packets in the buffer of routers before transmission. The Q_D depend on the details of the switching fabric in routers. Q_D is typically stochastic in nature due to the interference of probe packet with other IP packets on the path.

For each received packet P_r, the One Way Delay (OWD) is given by,

$$OWD = R_t - S_t \quad (5)$$

Where R_t is the receive time of the current packet and S_t is the time the packet was sent. The queuing delay over the network path, Q_D is computed from the measured delay and the minimum delay as,

$$Q_D = OWD - OWDmin \quad (6)$$

An exponentially weighted average of the queuing delay for the i^{th} received packet is formed by,

$$(AvgQ_D)_i = (1 - \phi) * (avgQ_D)_{i-1} + \phi * (Q_D)_i \quad (7)$$

Where, $\phi \leq 1$ is the forgetting constant. In simulations, ϕ was set to 0.1.

A Delay Factor (DF) is computed from the average queuing delay and the maximum queuing delay,

$$DF = \frac{(AvgQ_D)_i}{(MaxQ_D)} \quad (8)$$

Where, DF ranges between [0, 1] with 0 indicating no incipient congestion, 1 indicating full blown congestion, with shades of incipient congestion between 0 and 1. It is difficult to determine the $AvgQ_D$ and to check whether it is increasing or not, given that background cross traffic can cause the queuing delay to fluctuate around the average [17]

The cumulative queuing delay of packets along the link is defined as,

$$LQ(L) = \sum_{i=1}^{n} Q_{D(i)} \quad (9)$$

3.3 Congestion Detection in Internet

During transmission of data in the Internet, the router periodically calculates packet arrival rate (A_R) and queue occupancy (QO) of any pair of source (S) and destination (d) as per equation (1) and (3) respectively. Using A_R and QO measurements, the router estimates the traffic load of each path using equation (4).

After the calculation of traffic load, the router detects the congestion of paths as follows,

Let L be the traffic load of the path

Let min_{th} be the minimum threshold value of traffic load

Let max_{th} be the maximum threshold value of traffic load

Let CNB be the congestion notification bit

If (L < min_{th})

Then

The packet is transmitted to the destination

Else if (L > max_{th})

Then

The packet is dropped by the router

Else if ($min_{th} \geq$ L $\leq max_{th}$)

The packet is marked with CNB

End if

End if

End if

In the above algorithm, our technique defines two threshold values as min_{th} and max_{th} according to the traffic load in the network. The estimated traffic load L is compared with min_{th} and max_{th} values. If the L value is greater than min_{th} then the packet is allowed by the router to be transmitted in the network. The packet will be dropped when the L value exceeds $max_{th.}$ In the condition, when L value is between min_{th} and max_{th} , the router marks the packet with congestion notification bit. To mark packets with congestion notification bit, the router use Explicit Congestion Notification (ECN) bits. This strategy utilizes unused bits in IP headers. The following table-1 represents the format of ECN bits.

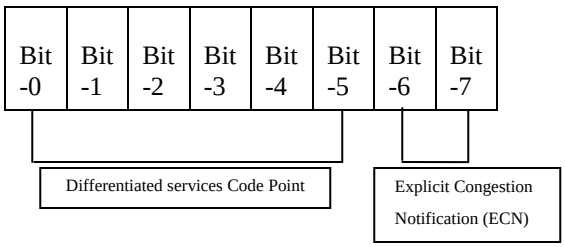

Table-1 Format of ECN Fields

When the router detects the congestion in the path, it marks Bit-6 and Bit-7 with binary 1, 1 value. In the absence of congestion, the bits are marked either 1, 0 or 0, 1 value, which means the host supports ECN strategy.

3.4 Handling Shared and Dedicated Link

3.4.1 Link Correlation Detection

In Internet, more than two paths can share a same link. Path that uses the single link is termed as dedicated link and conversely, link is known as shared when it is shared by more than two paths. This phenomenon of link is defined as link correlation. In our technique, we detect link correlation using crosslink factor. The picturization of dedicated link and shared link are shown in Figure-1 and 2 respectively.

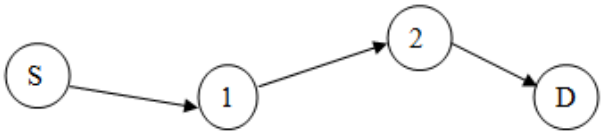

Figure-1 Dedicated Link

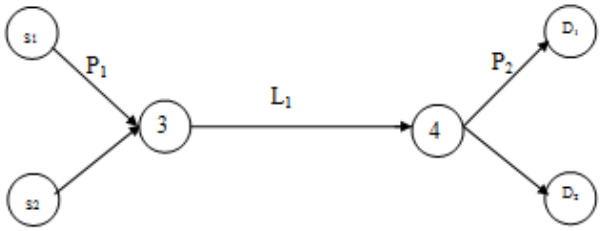

Figure-2 Shared Link

Consider P1 as a path that connects source S_1 and destination D_1 and P2 as a path between source S_2 and destination D_2. LQ_{p1} and LQ_{p2} be the queueing delays of path P1 and P2, respectively (estimated as per section 3.2.4).

Now, the link correlation factor (LCRF) of path P1 and P2 is measured as follows

$$LCRF_{P1,P2} = \frac{\sum_{i=1}^{n}\left(\mathrm{LQp1}-\overline{\mathrm{LQ}}\right)\left(LQp2-\overline{LQ}\right)}{\sqrt{\sum_{i=1}^{n}(LQp1-\overline{LQ})^2 \bullet \sum_{i=1}^{n}(LQp1-\overline{LQ})^2}} \quad \text{---------- (9)}$$

We take queuing delay as a measuring factor for discovering link correlation. By observing queuing delay of paths, LCRF shows high link correlation when the paths share more congested links. On the other hand, LCRF exhibits low link correlation when the paths do not share any congested links. In simple, the value of link correlation (LCRF=0) will be zero in the presence shared link and the value will be one (LCRF=1) if the link is a dedicated.

When the router detects congestion in the network, it estimates LCRF value.

If (LCRF = 0)

Then

The link is shared link

Congestion information is transmitted to all the corresponding sources of the link

Else if (LCRF = 1)

Then

The link is dedicated link

Congestion information is sent to only to the corresponding source

End if

End if

If LCRF value is one, the router decides the link as dedicated link. Then, it sends the congestion information to the corresponding source. Conversely, if LCRF value is zero, then the router transmit the congestion information to all sources in the shared link.

Algorithm

Step-1

The router calculates the packet arrival rate (A_R) (as per equation-1)

Step-2

The router calculates queue occupancy (QO) (as per equation-3)

Step-3

The router calculates traffic load (L)

(3.1) If ($L < min_{th}$), then packets are transmitted

(3.2) If ($L > max_{th}$), packets are dropped by the router
(3.3) If ($min_{th} \geq L \leq max_{th}$), packets are marked with CNB

Step-4

In the case of (condition 3.3), the router calculates the cumulative value of queuing delay (LQ_P) (as per section 3.2.4)

Step-5

Link Correlation Factor (LCRF) is estimated by the router.

(5.1) If (LCRF=0), the link is shared link

(5.2) If (LCRF=1), the link is dedicated link

Step-6

For shared link, the router sends congestion information to all sources in the link

Step-7

For dedicated link, the router transmits congestion information only to the corresponding source.

4. Simulation Results

4. 1. Simulation Model and Parameters

In this section, we examine the performance of our Load-Aware Congestion Detection Technique (LCDT) with an extensive simulation study based upon the ns-2 network simulator [19]. We compare our results with the Fuzzy Logic Congestion Detection (FLCD) [14] approach. The topology used in the simulation is depicted in Figure 3. The packet size is 512 bytes and packet sending rate is varied from 2 to 10Mb. The link bandwidth and link delay is set as 10Mb and 10ms respectively. The bottleneck bandwidth for the links (0, 1), (0, 2) and (1, 3) is set as 2 Mb initially. There are 3 UDP flows and 3 TCP flows.

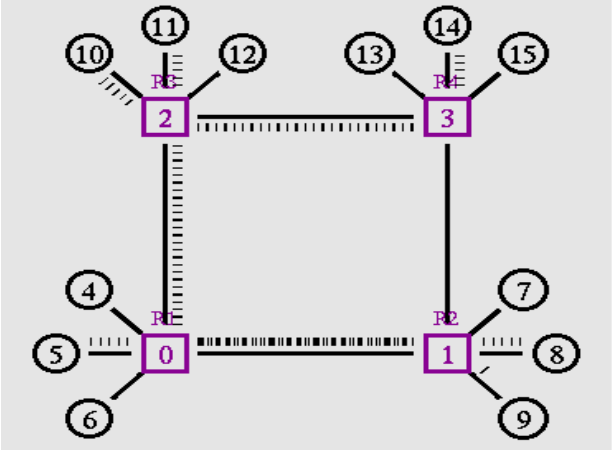

Figure 3: Simulation Topology

4. 2. Performance Metrics

In the simulation experiments, we vary the bottleneck bandwidth, traffic rate and time. We measure the following metrics

- Packet Loss
- Throughput
- Delay

The results are described in the next section.

4. 3. Results

A. Effect of Varying Bottleneck Bandwidth

In our first experiment, we vary the bottleneck bandwidth for the links as 2Mb, 4Mb… 8Mb.

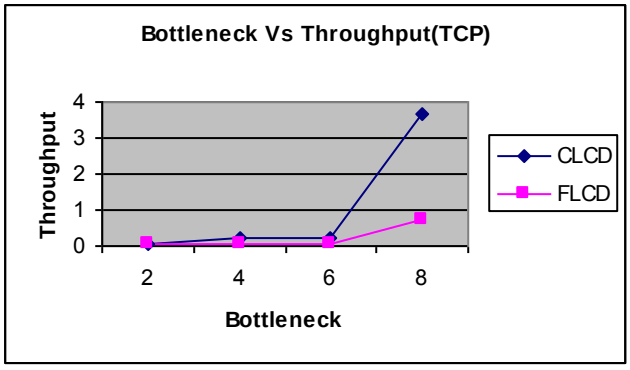

Fig 3: Bottleneck Vs TCP-Throughput

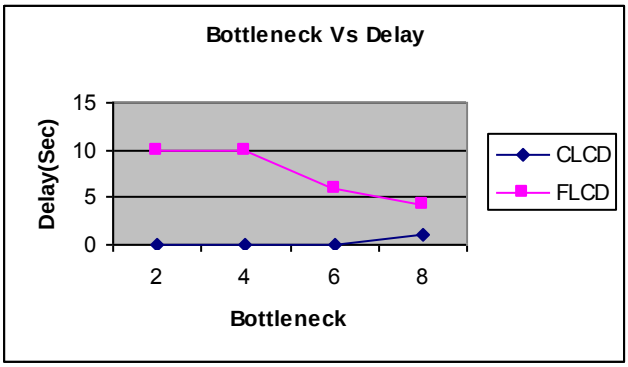

Fig 4: Bottleneck Vs Delay

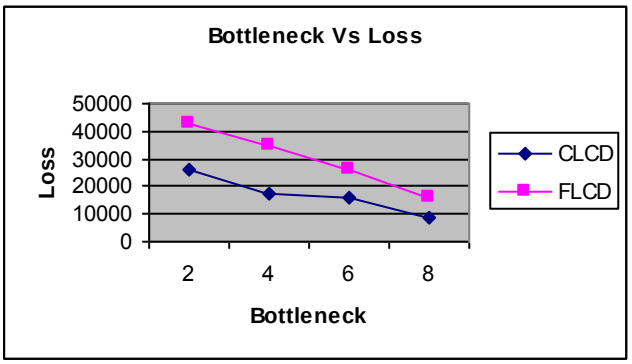

Fig 5: Bottleneck Vs Loss

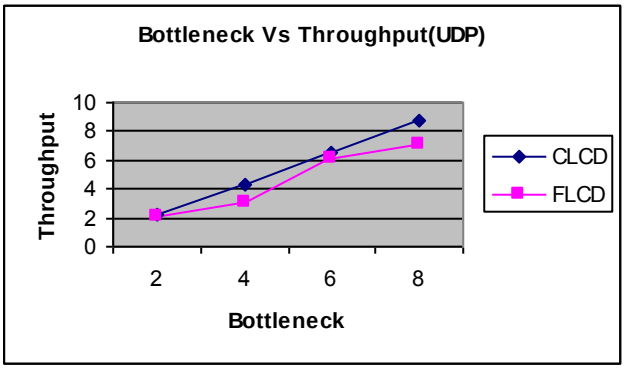

Fig 6: Bottleneck Vs UDP-Throughput

From Figure 3, we can see that the TCP-throughput of our proposed FBRCLCD is higher than the existing FLCD protocol.

From figure 4, we can see that the delay of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 5, we can see that the packet loss of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 6, we can see that the UDP-throughput of our proposed FBRCLCD higher than the existing FLCD protocol.

B. Based on Rate

In our second experiment we vary the rate as 2,4,6,8 and 10 Mb.

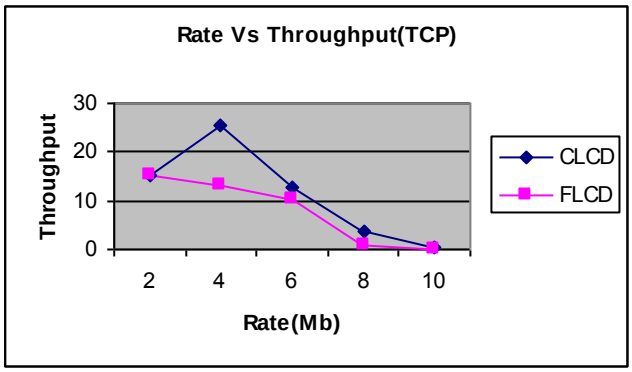

Fig 7: Rate Vs TCP-Throughput

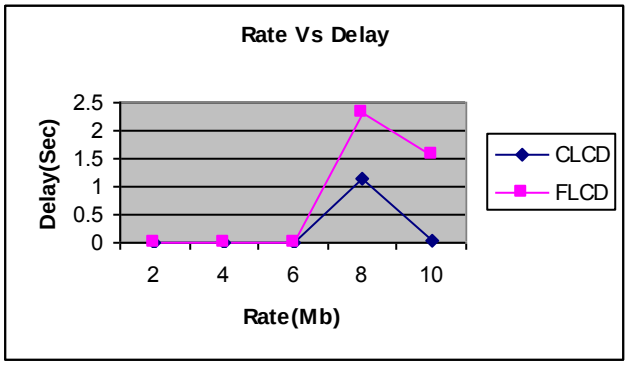

Fig 8: Rate Vs Delay

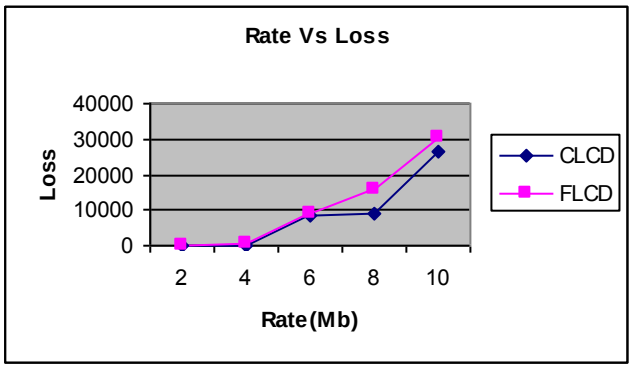

Fig 9: Rate Vs Loss

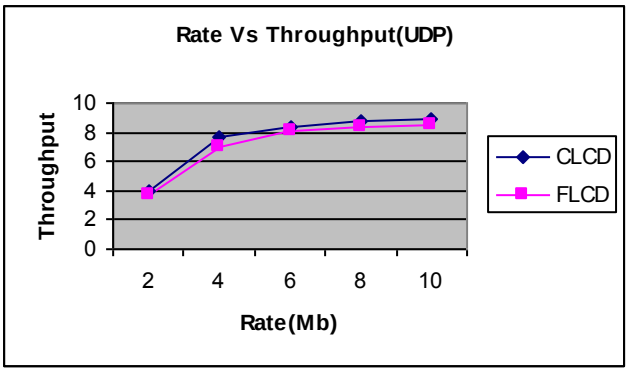

Fig 10: Rate Vs UDP-Throughput

From Figure7, we can see that the TCP-throughput of our proposed FBRCLCD is higher than the existing FLCD protocol.

From figure 8, we can see that the delay of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 9, we can see that the packet loss of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 10, we can see that the UDP-throughput of our proposed FBRCLCD higher than the existing FLCD protocol.

C. Based on Time

In our third experiment we vary the time as 2,4,6,8 and 10.

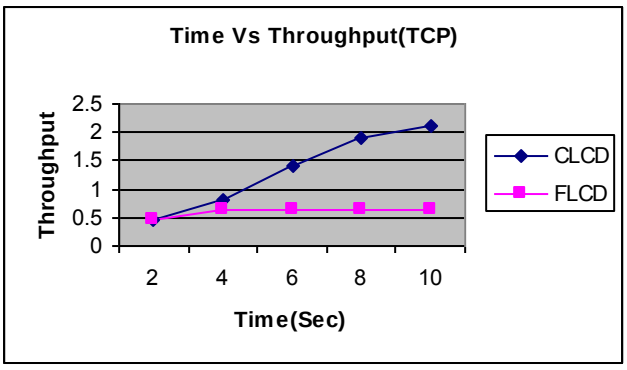

Fig 11: Time Vs Throughput

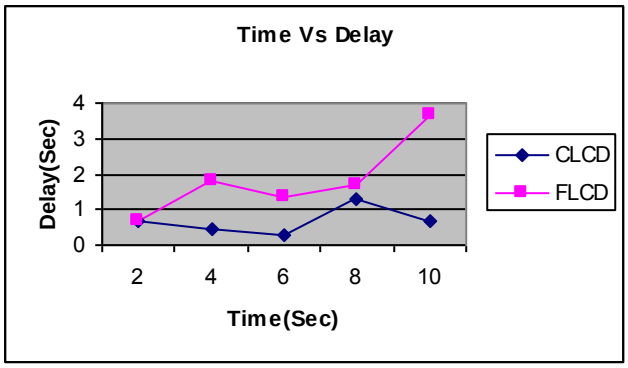

Fig 12: Time Vs Delay

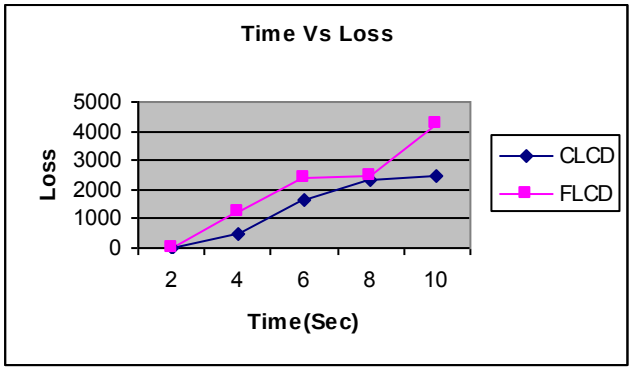

Fig 13: Time Vs Loss

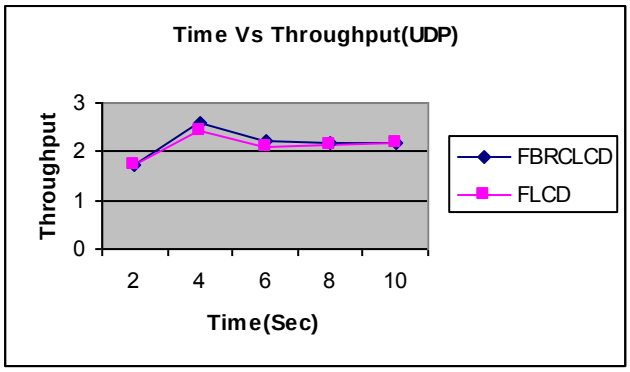

Fig 14: Time Vs Throughput

From Figure 11, we can see that the TCP-throughput of our proposed FBRCLCD is higher than the existing FLCD protocol.

From figure 12, we can see that the delay of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 13, we can see that the packet loss of our proposed FBRCLCD is less than the existing FLCD protocol.

From figure 14, we can see that the UDP-throughput of our proposed FBRCLCD higher than the existing FLCD protocol.

5. Conclusion

In this paper, we have proposed a load aware congestion detection technique for Internet. To detect congestion in the network, the router estimates the traffic load based on average packet arrival rate and queue occupancy rate. The estimated traffic load is compared with two predefined threshold values as min_{th} and $max_{th.}$ When the traffic load exceeds the value of max_{th} , it is dropped by the router. Packets that have values between min_{th} and max_{th} are marked with ECN bit. Simultaneously, the router estimates cumulative value of link queuing delay (LQ_D). Before transmitting congestion information to the source, the router differentiates dedicated and shared link using link correlation factor. For dedicated link, congestion information is directed only to the corresponding source. Just in case of shared link, congestion information is transmitted to all sources along that link. We have shown the performance of our technique through simulation results. Simulation results show that the proposed technique involves minimum delay and maximum throughput for TCP and UDP traffic.

References

[1] Steven Bauer and David Clark, "The Evolution of Internet Congestion",37th Research Conference on Communication, Information and Internet Policy, Arlington, VA, 2009.
[2] Tansu Alpcan and Tamer Basar, "Global Stability Analysis of an End-End Congestion Control Scheme for General Topology Networks with Delay", IEEE conference on Decision and control, 2003, pp-1092-1097.
[3] Kevin Curran, Derek Woods, Nadene McDermot and Colleen Bradley, "The effects of badly behaved routers on Internet Congestion", International Journal of Network Management, 2003, pp-83-94.
[4] Seungwan Ryu, Rump, C, and Chumming Qiao, "Advances in Internet Congestion Control", IEEE Communication Survey and Tutorials, vol-5, 2003, pp-28-39.
[5] Hari Balakrishnan, "End- End Congestion Control", Lecture 6, 1998.
[6] Chin-Fu Ku, Sao-Jie Chen, Jan-Ming Ho and Ray-I Chang, "Improving End-to-End Performance by Active Queue Management", IEEE 19th International Conference on Advanced Information Networking and Applications, Vol-2, 2005, pp-337-340.
[7] Sally Floyd and Kevin Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking, 1999.
[8] Chieh-Yih Wan, Shane B. Eisenman and Andrew T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", Proceedings of the ACM 1st international conference on Embedded networked sensor systems, 2003, pp- 266 – 279.
[9] Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, 1993, Vol-1, No-1.
[10] K. Srinivas and A. A. Chari, "ECDC: Energy Efficient Cross Layered Congestion Detection and Control Routing Protocol", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-2, May 2012.
[11] Michele C. Weiglea, Kevin Jeffayb and F. Donelson Smith, "Delay-based early congestion detection and adaptation in TCP: impact on web performance", Computer Communications, 2005, pp- 837–850.
[12] Ling Su and Jennifer C. Hou, "An Active Queue Management Scheme for Internet Congestion Control and its Application to Differentiated Services", Proceedings of IEEE Ninth International Conference on Computer Communications and Networks, 2000, pp- 62 – 68.
[13] Pascal Anelli, Emmanuel Lochin, Fanilo Harivelo and Dino Martin Lopez Pacheco, "Transport Congestion Events Detection (TCED): Towards Decorrelating Congestion Detection from TCP", Proceedings of the 2010 ACM Symposium on Applied Computing, (SAC '10) 2010, pp-663-669.
[14] C.N. Nyirenda and D.S. Dawoud, "Self-Organization in a Particle Swarm Optimized Fuzzy Logic Congestion Detection Mechanism for IP Networks", Scientia Iranica, International Journal of Science and Technology, Vol. 15, No. 6, December 2008, pp. 589-604.
[15] Ivan D. Barrera, Stephan Bohacek and Gonzalo Arce, "Statistical Detection of Congestion in Routers", IEEE Transactions on Signal Processing, pp- 957 – 968, Vol- 58, Issue-3, 2010.
[16] Wei Wu, Zhongzhao Zhang, Xuejun Sha and Chenguang He, "Auto Rate MAC Protocol based on Congestion Detection for Wireless Adhoc Networks", Inforamation Technology Journal 8, 2009.
[17] E Jammeh, M. Fleury and M. Ghanbari, "Delay and Loss Fuzzy Controlled Video over IP Networks", supported by the EPSRC, UK.
[18] Yury Puzis and Vasily Tarasov,"Effectiveness of ECN and RED Based Congestion Detection", Stony Brook University, 2008.
[19] Network Simulator: http:///www.isi.edu.ns/nam.

Ajay Shankar Singh has done his B. Tech and M. Tech degree in the field of Computer Science & Engineering from Kursk State Technical University, Russia, in 1997. His current areas of research are Congestion Control in High Speed Networks and Data Communications. His specializations include Bluetooth network ,Internet Technology , computer Networks, Cloud Computing & Virtualizations , Storage Technology, High Performance Computing & Availability and Open Source software.

Prof. Dr. E. G. Rajan is the Founder President of the Pentagram Research Centre (P) Ltd., Hyderabad, India. He is an Electronics Engineer and a Professor of Signal Processing having about 36 years of experience in teaching, research and administration. He received his Ph.D. degree in Electrical Engineering, (Signal and Image Processing), from Indian Institute of Technology (IIT), Kanpur, U.P. He received Distinguished Scientist and Man of the Millennium Award from. Who is Who Bibliographical Records, Cambridge, 2000. . He is the father of a novel paradigm Symbolic Computing In the Framework of Markov's Constructive Mathematical Logic. He has brought out more than 25 original concepts. He has guided more 26 Ph.D. scholar and published more than 320 contributed papers in reputed journals.

Dr. Manish Prateek is working as Professor and Head, Department of CSE at University of Petroleum and Energy Studies, Dehradun, India. He has done his B. Tech and M. Tech degree in the field of Computer Science from Kursk State Technical University, Russia, in 1996. He has also worked in the IT industry for 8years at different levels. He did his Ph. D from L. N. Mithila University, in the area of Manufacturing and Robotics. He is also recipient of the award "Distinguished Academician" for his achievements in Academics and Research & Development, in the year 2010.