ISSN No. 2278 -3091
Volume 1, No.2, May – June 2012
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at www.warse.org/ijatcse/info.html

Delay based End to End Internet Congestion Control using Natural Logic

Ajay Shankar Singh
University Of Petroleum & Energy Studies, Dehradun-248007, India
ajay290672@gmail.Com
Dr. E G Rajan
Professor Of Signal Processing
Chairman, Pentagram Group Of Companies, Hyderabad – 500 028, India
Dr. Manish Prateek
Professor & Head, CIT-COES, University Of Petroleum & Energy Studies
Dehradun-248007, India
PSVS Sridhar
University Of Petroleum & Energy Studies
Dehradun-248007, India

ABSTRACT

The problem of network congestion management is a major issue and a high priority, especially given the growing size, demand, and speed (bandwidth) of the increasingly integrated services networks. Delay-based algorithms become the preferred approach for end-to-end congestion control as networks scale up in capacity. Their advantage is small at low speed but decisive at high speed. This paper describes new mechanisms for intelligent end to end internet congestion control (E2IC2) by means of natural logic systems based on delay feedback.

Keyboards: Congestion Control, Internet Congestion, Adaptive Congestion Avoidance, Natural Logic, Fuzzy Logic

1. INTRODUCTION

Internet is undoubtedly one of the most popular and revolutionary technology to be widely distributed in the past two decades. His impact and influence can be seen all over the world, especially in the first world, but increasingly in developing countries, many of whom see it as having an economic leveling effect.

The network itself is really only provides a (usually) a moderately reliable signaling and routing system for the transfer of small blocks (packets) of data from one computer to another. Packets may be lost or reordered without notice.

The goal of congestion control mechanism to just use the network as efficiently as possible, that is, to achieve maximum throughput while maintaining a low coefficient of loss and low

Hosts wishing to communicate on the network must work within these constraints. Communicating hosts must provide for themselves any greater level of reliability they need. This design is based on the “end-to-end” principal which puts the main, because much of the communication protocol operations as possible at the endpoints [1].

There are two very often, "Transport Protocols", running on end hosts. UDP just sends packets and does not provide additional reliability. If the user doesn’t get a response to a request, they are usually expected to request again if they so choose. TCP by contrast ensures that arrival of all packets at their destination is confirmed, retransmitting any which are lost; ensures that any reordering of the packets is corrected, keeps different communication sessions from interfering and attempts to send at the highest rate it can without causing excessive packet loss due to congestion in the network [1]. This last responsibility of TCP is called “congestion control” and was added in the late eighties in response to several instances of congestion collapse on the early internet.

In subsequent years, TCP’s standard congestion control (Tahoe, Reno) has evolved in small ways, but not a major redesign took place. Various problems have been shown, particularly in its achieved throughput on large bandwidth-delay product (BDP) networks and its propensity to cause a long delay on the network due to long queues available [4].

latency. Accumulation should be avoided because it leads to an increase in the queue and growth leads to delay and loss, so the "congestion avoidance" term is sometimes used.

1.1 Defining Congestion

Roughly speaking, everyone would agree that "congestion" is the state of network overload. Nevertheless, it is not sufficient to accurately characterize precisely and how long the network is congested. In queuing theory, traffic congestion is said to occur if the arrival rate exceeds the service rate of the system in time [5].

Congestion, or traffic intensity, can be measured as the ratio of the arrival rate to service rate.

Congestion occurs when the resource requirements exceed the capacity. Excess packets can not be transferred over the connection, there are only two things that this device can do: packet buffer or reduce them. Standard Internet routers usually place the excess packets in the buffer, which roughly works as a basic FIFO ("first in, first out") queue and drop packets only when the queue is full [6]. It is shown in Figure 1.

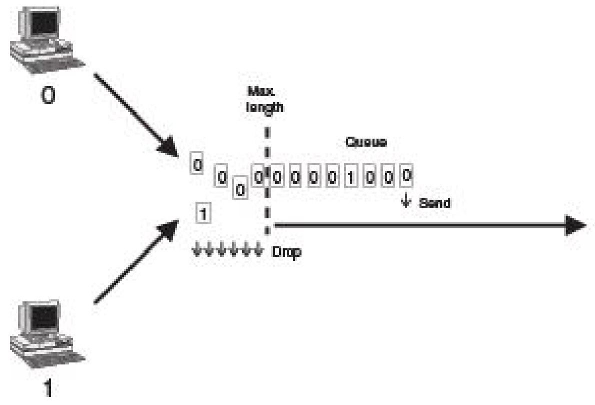


Figure 1: Packet Drop Due to Overflow of Queue Buffer

It would seem that the reservation of sufficient buffer for the long line is a good choice because it increases the likelihood of placement of traffic spikes. However, there are major challenges is to keep packets in the queue adds considerable delay, depending on the length of the queue.

Queues should generally be kept short.

A network is said to be congested from the perspective of a user if the service quality noticed by the user decreases because of an increase in network load.

The purpose of congestion control mechanisms simply use the network as efficiently as possible, that is, to achieve maximum throughput while maintaining low loss factor and low latency [5]. Congestion should be avoided because it leads to an increase in queues and queue growth leads to delay and loss, so the congestion, the term "avoidance. Figure 2 shows technical difeerence between concongestion avoidance and control. Congestion avoidance is proactive but congestion control terrn is reactive.

.

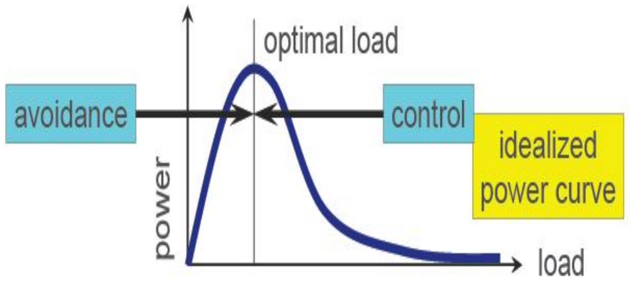


Figure 2 : Logical Difference Between Congestion Avoidance and Congestion Control

Some fashion to replace the 'congestion control' with terms such as "high-performance network", "high speed data transmission and so on for the past few years. Do not let this confuse people - the same goal with slightly different environmental conditions [2].

On a side note, moving to the congestion to the access channel does not mean that it will disappear if the network is used in a careless manner, the queue can still grow and increase in latency and packet loss can still occur.The heterogeneity of link speeds from one end of the path traversed by multiple boundaries provider may also be a source of congestion[8].

2. CONTROLLING CONGESTION: DESIGN CONSIDERATIONS

Traffic can be controlled at the sender and at the intermediate nodes; performance measurements can be taken by intermediate nodes and by the receiver. Let us call members of the first group controls and members of the second group measuring points. Then, at least one control and one measuring point must participate in any congestion control scheme that involves feedback[9].

Congestion can be sensed (or predicted) by:
1. packet loss sensed by
    - the queue as an overflow,
    - destination (through sequence numbers) and acknowledged to a user [3],
    - sender due to a lack of acknowledgment (timeout mechanism) to indicate loss.
2. packet delay
    - can be inferred by the queue size,
    - observed by the destination and acknowledged to a user (e.g. using time stamps in the packet headers),
    - observed by the sender, for example by a packet probe to measure Round Trip Time (RTT).
3. loss of throughput
    - observed by the sender queue size (waiting time in queue).

2.1 Classification Of Congestion Control Algorithms

There are many ways to classify congestion control algorithms:

• The feedback received from the network: Loss, delay, one-bit or multi-bit explicit signals.

• The additional ability to deploy on the current Internet:

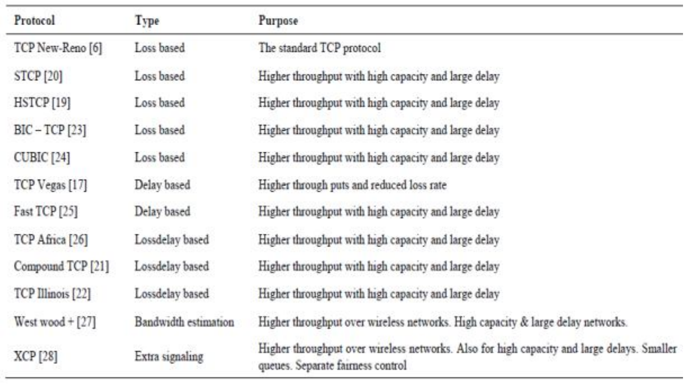
- Only sender needs modification,
- the sender and receiver need to be modified,
- only the router needs to be modified,
- sender, receiver and routers need to be modified.

• The aspect of performance, it aims to improve:

- high bandwidth delay product networks,
- loss of links,
- fairness,
- the advantage of short flows,
- variable-rate links.

• In terms of fairness, it uses: max-min, proportional, "minimum potential delay".

Table 1: Variants of TCP Congestion Control Implementation



Protocol	Type	Purpose
TCP New-Reno [6]	Loss based	The standard TCP protocol
STCP [20]	Loss based	Higher throughput with high capacity and large delay
HSTCP [19]	Loss based	Higher throughput with high capacity and large delay
BIC – TCP [23]	Loss based	Higher throughput with high capacity and large delay
CUBIC [24]	Loss based	Higher throughput with high capacity and large delay
TCP Vegas [17]	Delay based	Higher through puts and reduced loss rate
Fast TCP [25]	Delay based	Higher throughput with high capacity and large delay
TCP Africa [26]	Lossdelay based	Higher throughput with high capacity and large delay
Compound TCP [21]	Lossdelay based	Higher throughput with high capacity and large delay
TCP Illinois [22]	Lossdelay based	Higher throughput with high capacity and large delay
West wood + [27]	Bandwidth estimation	Higher throughput over wireless networks. High capacity & large delay networks.
XCP [28]	Extra signaling	Higher throughput over wireless networks. Also for high capacity and large delays. Smaller queues. Separate fairness control

2.2 Rate-Based Congestion Control Scheme

The receiver estimates the new sending rate . The estimated new sending rate is sent as a feedback to the sender. The sender then performs rate adjustment based on this estimated new rate. Table 1 shows different variants of TCP congestion control implementation. End to End rate based congestion control techniques is shown in Figure 3.

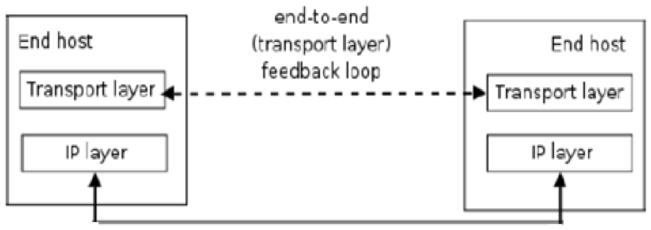


Figure 3: End to End Feedback Loop in Internet Traffic Regulation

2.3 Delay

As a packet travels from one node (host or router) to the subsequent node (host or router) along this path, the packet suffers from several different types of delays at *each* node along the path. The most important of these delays are the nodal processing delay, queuing delay, transmission delay and propagation delay; together, these delays accumulate to give a total nodal delay[14].

The time required to examine the packet's header and determine where to direct the packet is part of the processing delay. The processing delay can also include other factors, such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream router to router A. After this nodal processing, the router directs the packet to the queue that precedes the link to router B. At the queue, the packet experiences a queuing delay as it waits to be transmitted onto the link. The queuing delay of a specific packet will depend on the number of other, earlier-arriving packets that are queued and waiting for transmission across the link; the delay of a given packet can vary significantly from packet to packet. If the queue is empty and no other packet is currently being transmitted, then our packet's queuing delay is zero. On the other hand, if the traffic is heavy and many other packets are also waiting to be transmitted, the queuing delay will be long [15].

The propagation delay is the distance between two routers divided by the propagation speed. That is, the propagation delay is *d/s*, where *d* is the distance between router A and router B and *s* is the propagation speed of the link.

Newcomers to the field of computer networking sometimes have difficulty understanding the difference between transmission delay and propagation delay. The difference is subtle but important. The transmission delay is the amount of time required for the router to push out the packet; it is a function of the packet's length and the transmission rate of the link, but has nothing to do with the distance between the two routers. The propagation delay, on the other hand, is the time it takes a bit to propagate from one router to the next; it is a function of the distance between the two routers, but has nothing to do with the packet's length or the transmission rate of the link[17].

Delay = Fixed Component (Transmission at node, Propagation delay) + Variable Component (processing and queueing delays at node).

If we let d_{proc}, d_{queue}, d_{trans} and d_{prop} denote the processing, queuing, transmission and propagation delays, then the total nodal delay is given by

$$d_{\text{nodal}} = dproc + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} .$$

2.4 Network timer

Round-trip time (RTT) time is the time required for a packet to travel from the testing host to a remote computer that receives the packet and retransmits it back to the source. The One-Way Delay (OWD) value is calculated between two synchronized points A and B of an IP network, and it is the time in seconds that a packet spends in travelling across the IP network from A to B [18].

The one-way delay (OWD) of a stream of packets is the sum of the path delay dp and the queuing delay dq. Queuing delay is a function of the network load or congestion level. Maximum queuing delay occurs when the buffer at the bottleneck link is full and packets start being dropped: full blown congestion [7]. Figure 4 shows relationship between Input rate and one way delay in networks.

The OWD of packets is at a minimum when the sending rate is less than the available network bandwidth and OWD starts increasing above the minimum when the sending rate exceeds the available bandwidth at the bottleneck link. A further increase in the sending rate results in an increase in OWD until a maximum is reached, which is a function of the buffer size at the bottleneck link[19].

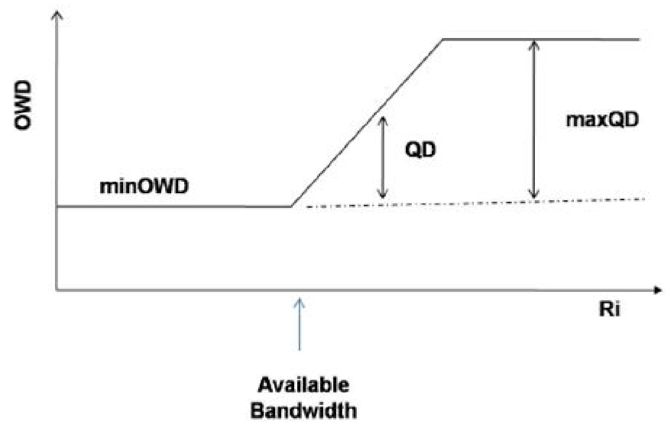


Figure 4: Relationship Between Input Rate and OWD

2.5 Delay Factor (D_f) Calculation

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

$$OWD = R_t - S_t \text{ --- } (1)$$

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

$$Q_D = OWD - OWDmin$$

An exponentially weighted average of the queuing delay for the ith received packet is formed by,

$$(AvgQ_D)_i = (1 - \phi) * (avgQ_D)_{i-1} + \phi * (Q_D)_i \text{ --- } (2)$$

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,

$$DF = \frac{(AvgQ_D)_i}{(MaxQ_D)} \text{ ---- } (3)$$

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 whether it is increasing or not, given that background cross traffic can cause the queuing delay to fluctuate around the average [8].

2.6 Increasing (ITR) / Decreasing(DTR) Trend Analysis

We determine the average queuing delay ($AvgQ_D$) from equation 3 and whether it is increasing or not, given that background cross traffic can cause the queuing delay to fluctuate around the average [17]. A trend analysis method is used to compute a trend value which is fuzzified by the (E_2IC_2) to determine whether the queuing delay is increasing (ITR) or decreasing (DTR) [16].

Linguistic Input Variables and Resulting Output		
Delay Factor (D_f)	I/D trend	Bit Flow Rate (Ctrl)

VeryLow (VL)
Low (L)
Medium(M)
High (H)
VeryHigh(VH)
	Increasing(Incr)
	Decreasing(Decr)
		Decrease Extremely High(DEH)
		Decrease High(DH)
		Decrease Medium(DM)
		Decrease Low(DL)
		Zero
		Increase Low(IL)
		Increase Medium(IM)
		Increase High(IH)
		Increase Extremely High(IEH)

Table 2: Linguistic Variables for (E_2IC_2) System

3. NATURAL LOGIC

Natural Logic Controls may be viewed as alternative, non-conventional way of designing feedback controls where it is convenient and effective to build a control algorithm without relying on formal models of the controlled system and control theoretic tools. The control algorithm is encapsulated as a set of commonsense rules. NLCs have been applied successfully to the task of controlling systems for which analytical models are not easily obtainable or the model itself, if available, is too complex and highly nonlinear.

3.1 Membership Functions and Linguistic Variables

Definition of membership functions and linguistic variables are the first steps in system designing. Each linguistic variable contains terms which are interpretation of technical figures. In our work we have used experimental triangular membership functions for coding and evaluation simplicity. The input linguistic variables are Delay Factor (Df) and Increase/decrease trend of congestion. The output linguistic variable is “Bit Flow Rate (Ctrl)” which regulate the source flow. Figure 5 shows a natural logic congestion control which takes delay factor (Df) and increasing /decreasing trend to determine the bit flow rate (Ctrl) which are calculated at receiver end but implemented by source node. In our paper control decision is taken on the base of Table 2. Figure 6 & 7 shows membership functions of variables according to natural logic and Figure 8 shows membership function of decision based on input variables.

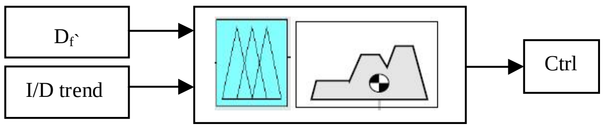


Figure 5 : High-Level View Of The Proposed (E_2IC_2) System

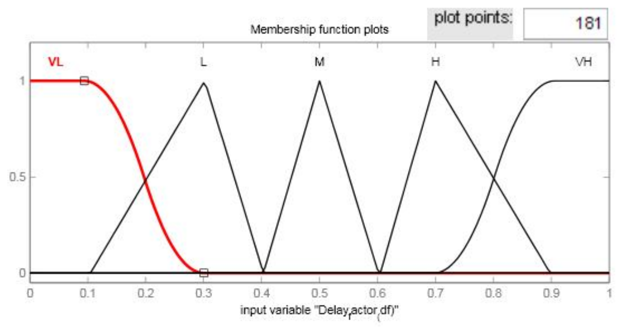


Figure 6 : The membership functions for the measured Delay Factor (D_f)

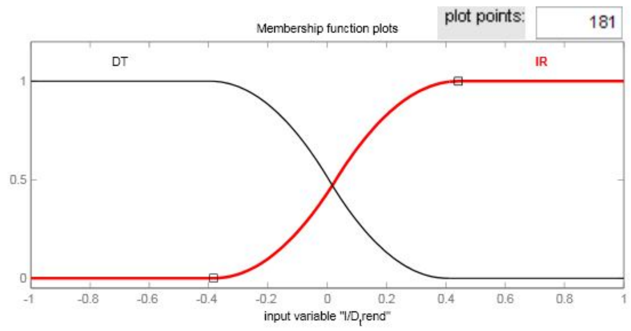


Figure 7 : The membership functions for the Increase/decrease trend

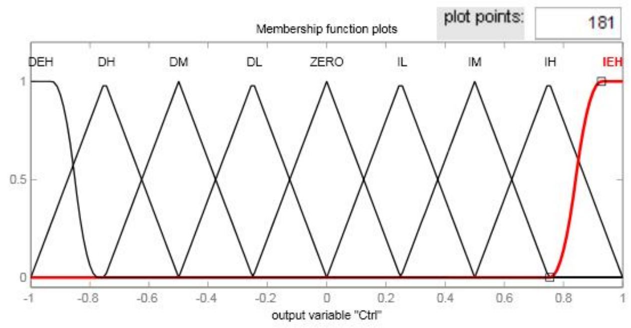


Figure 8 : The membership functions for the OUTPUT “Bit Flow Rate (Ctrl)”

3.2 Natural Logic Rule Base

The second step in designing a Natural logic system is the creation of a natural logic rule base which supplies the knowledge of the system [15]. To build the rule base, we need to present some standard methods. A natural logic rule is an IF-THEN rule.

*Rule definition***:** A membership function which characterizes a set A in x can be implemented easily using conditional statements. If an antecedent in a natural logic statement is true to some degree of membership then the consequent is also true to that same degree.

*Rule structure***:** If antecedent then consequent

The rule: If one variable is low and one variable is high then output will be benevolent else it will be malevolent.

On applying a set of natural logic rules based on the linguistic values of its attributes a case or an object can be classified in a natural logic classification system. The rule is applied to the number given by the antecedent. This rule has a weight which is numbered between 0 and 1. Initially the antecedent is evaluated which fuzzifies the input and applies any necessary natural logic operators. Then the result is applied to the consequent which is known as inference. A set of natural logic rules related to specific classification problem need to be found for building a natural logic classification system. This is considered as the most difficult task.

We will now describe our methodology for natural logic approach to control the rate in the network. The two most important variables in controlling the rate are *Delay Factor (D_f) and Decrease/Increase trend.* With Natural Logic , we assign grade values to our three variables. Our Natural Logic set therefore consists of three Natural Logic variables[12].

Natural Logic set = { D_f ,I/D trend}

Natural Logic implements human experiences and preferences via membership functions and Natural Logic rules. In this work, the Natural Logic if-then rules consider the parameters: *Delay Factor (D_f) and Decrease/Increase trend.*

1. If (Delay_factor_(df) is VL) and (I/D_trend is IR) then (Ctrl is IM)
2. If (Delay_factor_(df) is VL) and (I/D_trend is DT) then (Ctrl is IEH)
3. If (Delay_factor_(df) is L) and (I/D_trend is IR) then (Ctrl is IL)
4. If (Delay_factor_(df) is L) and (I/D_trend is DT) then (Ctrl is IM)
5. If (Delay_factor_(df) is M) and (I/D_trend is IR) then (Ctrl is DL)
6. If (Delay_factor_(df) is M) and (I/D_trend is DT) then (Ctrl is IL)
7. If (Delay_factor_(df) is H) and (I/D_trend is IR) then (Ctrl is DH)
8. If (Delay_factor_(df) is H) and (I/D_trend is DT) then (Ctrl is DL)
9. If (Delay_factor_(df) is VH) and (I/D_trend is IR) then (Ctrl is DEH)
10. If (Delay_factor_(df) is VH) and (I/D_trend is DT) then (Ctrl is DM)

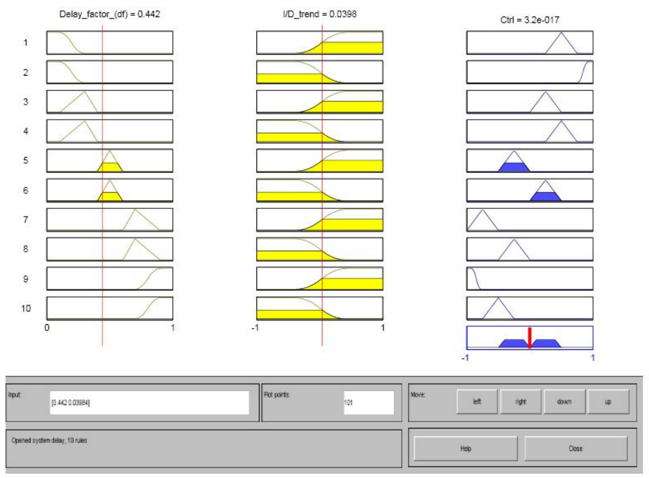

Delay_factor_(df) = 0.442
I/D_trend = 0.0398
Ctrl = 3.2e-017
Input: [0.442 0.03984]
Plot points: 101
Opened system delay, 10 rules

Figure 9 : The membership functions for the OUTPUT “Bit Flow Rate (Ctrl)”

3.3 Bit Flow Rate (Ctrl):

The E_2IC_2 employs a simple Mamdani inference model and center-of-areas defuzzification method [11]. Figure 9 shows membership functions for the OUTPUT “Bit Flow Rate (Ctrl) based on Mamdani center-of-areas defuzzification method. Figure 10 shows decision surface of the natural logic inference engine using MATLAB fuzzy logic tools software.

$$\text{Ctrl} = \frac{\sum_{i=1}^{M} s_i K_i}{\sum_{i=1}^{M} K_i}$$

Eqn. maps the input to the output of the control. For input bitrate *Rin*, the target output bitrate is *Rout* is given by:

Rout = (1 + *Ctrl*) * *Rin*

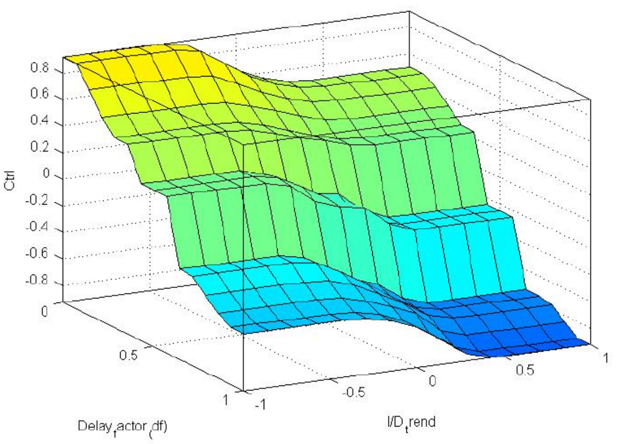


Figure 10 : Decision Surface of The Natural Logic Inference Engine.

4. SIMULATION RESULTS

4.1 Simulation Model and Parameters

In this section, we examine the performance of our end to end internet congestion control (E_2IC_2) with an extensive simulation study based upon the ns-2 network simulator [21]. We compare our results with Adaptive Delay-based Congestion Control (DBC_2) approach[10]. The topology used in the simulation is depicted in Figure 11. 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.

In our experiment, we vary the bottleneck bandwidth for the links as 2Mb, 4Mb… 8Mb in order to calculate the throughput, delay and packet loss.

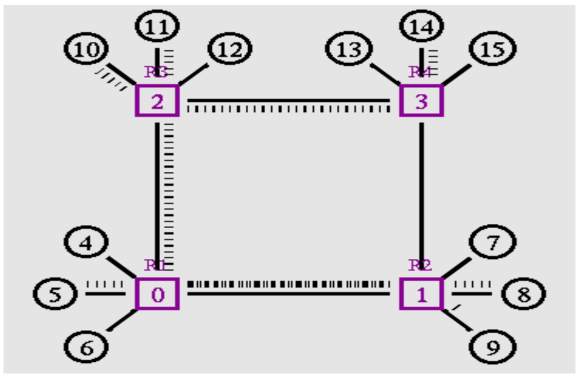


Figure 11: Simulation Topology

4.2 Performance Metrics

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

- Throughput
- Delay
- Packet Loss

The results are described in the next section.

4.3 Results

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

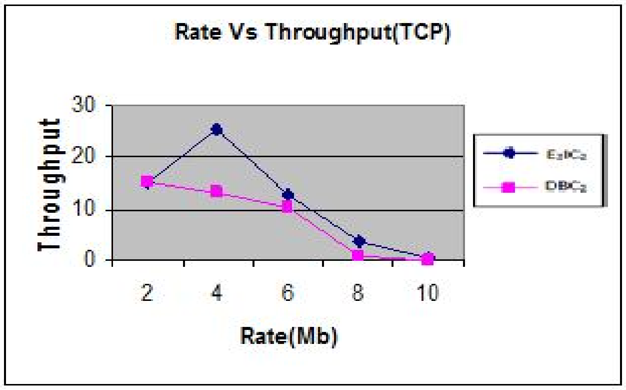


Figure 12: Rate Vs TCP-Throughput

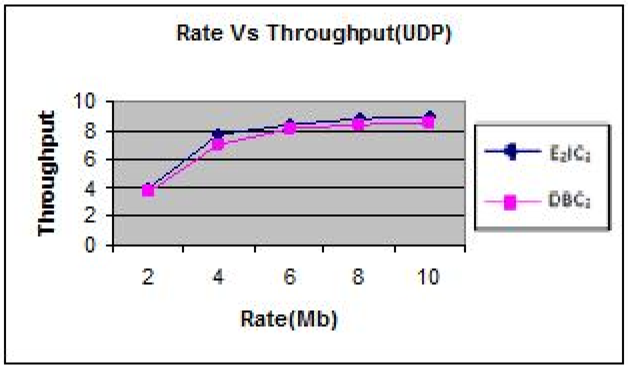


Figure 13: Rate Vs UDP-Throughput

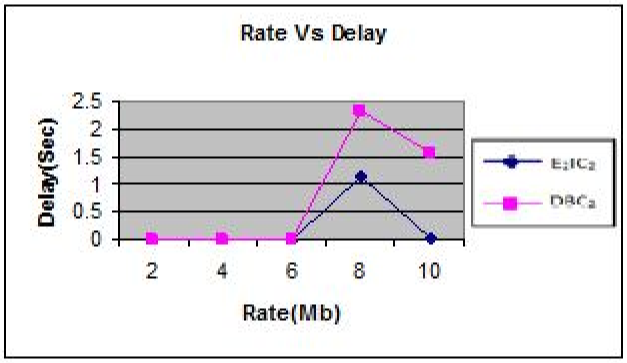


Figure 14: Rate Vs Delay

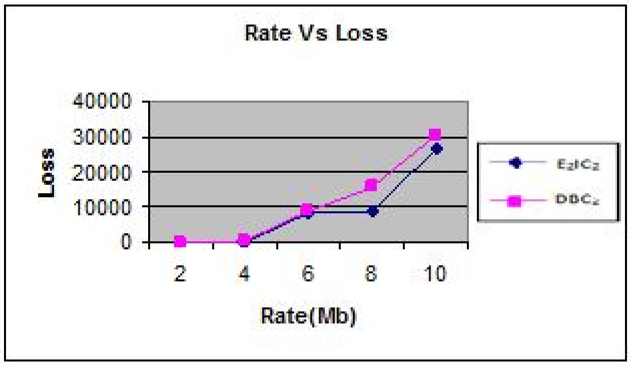


Figure 15: Rate Vs Loss

From Figure 12, we can see that the TCP-throughput of our proposed E_2IC_2 is higher than the existing DBC_2 protocol.

From Figure 13, we can see that the UDP-throughput of our proposed E_2IC_2 higher than the existing DBC_2 protocol.

From Figure 14, we can see that the delay of our proposed E_2IC_2 is less than the existing DBC_2 protocol.
.
From Figure 15, we can see that the packet loss of our proposed E_2IC_2 is less than the existing DBC_2 protocol.

5. CONCLUSION

In conclusion, there is a real problem in the area of congestion in communication networks, especially those that support video, voice and data simultaneously. Computational Intelligence techniques are expected to play a central role, especially in the large scale, geographically distributed network systems.Under severe congestion condition, the TCP congestion control algorithm goes into Slowstart mode i.e. all the sources have to drastically reduce their packet transmission rate to one packet and then again they slowly increase their transmission rate through Additive Increase & Multiplicative Decrease (AIMD).

From the measurement study, it is apparent that packet loss is not a accurate indicator of congestion because packet loss can occur for many factors such as background traffic type (broadly 'bursty' or uniform) across the tight link on a streaming path. With suitable instrumentation it is possible to detect packet-by-packet delay, which this study indicates is a more reliable way of detecting the congestion level.

The proposed algorithm The output(Ctrl) gives us a stringent passive measure of rate control level and will result in a perfect rate control for end to end congestion control for Internet. Thus our algorithm proves to be more effective in controlling the rate as we consider Delay Factor (Df) and its Decrease/Increase trend. parameters. We have shown the performance of our technique (E_2IC_2) through NS2 simulation results. Simulation results show that the proposed technique involves minimum packet loss , minimum delay and maximum throughput for TCP and UDP traffic till today.

ACKNOWLEDGEMENTS

We would like to thank to our colleague Prof. Anil Kumar, Mr. Vishal Kaushik, Mr. Neeraj Chugh of the UPES, Dehradun, India for having extended all the support and encouragement for carrying out this work. We wish to special thanks to Mr. RK Jaiswal and Mr. Prakash GL who help us to design controller in MATLAB and implemented our idea in NS2 simulator.
.

REFERENCES

1. www.hamilton.ie/net/gavin_thesis.pdf
2. www.fas.org/sgp/othergov/doe/lanl/pubs /00326726.pdf
3. www.scribd.com/doc/50122162/book -CI- congestion-control.
4. S. Floyd, M. Handley, J. Padhye, and J. Widmer. **Equation-Based Congestion Control for Unicast Applications**, http://www.aciri.org/tfrc/, June 2000.
5. Shakeel Ahmad. **Analysis Of Complex Queues With Finite Capacities**. PhD thesis, pp. 51-78, 2007.
6. P. E. McKenney**. Stochastic Fairness Queueing**. Proc. IEEE INFOCOM, 2:pp. 733– 740, June 1990.
7. N. Pekergin. **Stochastic Bounds on Delays of Fair Queueing Algorithms**. Technical Report PRISM, UVSQ 10, Universit'e de Versailles-St-Quentin, 1998.
8. H. Jung, S. Kim, H. Yeom, S. Kang, and L. Libman. **SNU Technical Report: Adaptive Delay-Based Congestion Control For High Bandwidth-Delay Product Networks**, 2010. [Online]. Available: http://dcslab.snu.ac.kr/acp/tr-dcs-0901.pdf
9. I. A. Qazi and T. Znati. **On The Design Of Load Factor Based Congestion Control Protocols For Next-Generation Networks,** in Proc. of IEEE INFOCOM, Phoenix, AZ, USA, Apr. 2008.
10. Hyungsoo Jung, Shin-gyu Kim, Heon Y. Yeom†, Sooyong Kang and Lavy Libman. **Adaptive Delay-based Congestion Control for High Bandwidth-Delay Product Networks,** IEEE INFOCOM,pp. 2885-2893, 2011.
11. M. H. Yaghmaee. **A Modified Random Early Detection Algorithm: Fuzzy Logic Logic Based Approach**, JCN, Journal of Communication Network, Vol.7, No. 3, pp. 337-352. September 2005.
12. M. H. Yaghmaee, M. Menhaj and H. Amintoosi. **A Fuzzy Logic Extension To The Blue Active Queue Management Algorithm**, IAEEE, Journal of Iranian Association of Electrical and Electronics Engineers, Vol. 1, No. 3, pp. 3-14, Winter 2005.
13. R.J. Gibben and F.P. Kelly. **Resource Pricing And Evolution Of Congestion Control**. Automatica, 1999.
14. Srisankar Kunniyur. **Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management**. ACM SIGCOMM, 2001.

15. De Marco, F. Postiglione and M. Longo. **Run-Time Adjusted Congestion Control For Multimedia: Experimental Results**, Journal of Interconnection Networks (JOIN), Vol. 5, No. 3, pp. 249-266, 2004.

16. M. Jain and C. Dovrolis. **Pathload: A Measurement Tool for End-to-End Available Bandwidth,** Passive and Active Measurements Workshop, March 2002.

17. M.C. Weigle, K. Jeffay, and F.D. Smith. **Delay-Based Early Congestion Detection And Adaptation In TCP: Impact On Web Performance,** Computer Communications, Vol. 28, No. 8, pp. 837-850, May 2005.

18. S. Floyd, M. Handley, J. Padhye, and J. Widmer. RFC 5348: **TCP Friendly Rate Control (TFRC): Protocol Specification**, Sep. 2008.

19. H. Jung, S. Kim, H. Yeom, and S. Kang. **TCP-GT: A New Approach To Congestion Control Based On Goodput And Throughput,** Journal of Communications and Networks, vol. 12, no. 5, pp. 499–509, Oct. 2010.

20. X. Huang, C. Lin, F. Ren, G. Yang, P. Ungsunan, and Y. Wang. **Improving The Convergence And Stability Of Congestion Control Algorithm,** In Proceedings of IEEE ICNP, Beijing, China, Oct. 2007.

21. Network Simulator, http://www.isi.edu/nsnam/ns