SHODH SAMAGAM

ISSN : 2581-6918 (Online), 2582-1792 (PRINT)

A Research study to develop Congestion Control Mechanism Using Fuzzy Logic

Aparna Tiwari, Department of Computer Science
Agrasen Mahavidyalaya, Purani Basti, Raipur, Chhattisgarh, INDIA

ORIGINAL ARTICLE

Author
Aparna Tiwari

shodhsamagam1@gmail.com

Received on	: 16/03/2024
Revised on	: -----
Accepted on	: 17/05/2024
Overall Similarity	: 10% on 09/05/2024

Plagiarism Checker X - Report
Originality Assessment
Overall Similarity: **10%**
Date: May 9, 2024
Statistics: 507 words Plagiarized / 4866 Total words
Remarks: Low similarity detected, check with your supervisor if changes are required.

ABSTRACT

As the use of internet & network are growing very fast and high speed is recommended, the problem of congestion becomes critical issue. There are several congestion control mechanism has been developed, but still there is no any single algorithm that can resolve all problems of congestion control on network. Congestion is a problem which effects on losses, throughput (wastage of bandwidth used for retransmission & unnecessary processing), delay & other performance metrics. Congestion control is a process which keeps the flow of load below than the network capacity. The main objective of this research is to develop a fuzzy based congestion control algorithm which can able to detect congestion prior to its occurrence & accordingly perform action in order to avoid congestion. A congestion control algorithm has been proposed using fuzzy logic. In the proposed algorithm taking two parameters as input the difference between the number of incoming packets and number of outgoing packets and available space in buffer, then calculate the Congestion gradient factor using the fuzzy rule set. According the value of congestion gradient factor necessary action is performed prior to the occurrence of congestion. Using conventional system, it is very difficult to solve the problem of congestion control as it unable to define various conditions of network. TCP uses end-to-end congestion control rather than the network supported congestion control. TCP/IP does not provide any services to prevent the buffer of intermediate routers from getting overflow. The proposed system can be helpful to identify intermediate routers that can be congested, by estimating the processing load and available buffer *space & also perform necessary action to avoid congestion. The proposed algorithm is capable to perform well in wired/wireless network especially in mobile ad-hoc network where each node can work as router.*

KEY WORDS

Congestion, Congestion control, Fuzzy logic, TCP/IP Performance metrics.

INTRODUCTION

As highway is congested when it is overloaded with traffic in the form of vehicles, similarly Network gets congested when it is overloaded with data. In networking, congestion occurs on shared networks when, multiple users contend for access to the same resources (bandwidth, buffers, and queues) or in simple language when the traffic flowing through a network exceeds its maximum capacity, because routers receive packets enough than their capacity to forward them, one of these two things may happen in case of congestion: The subnet may prevent additional packets to enter in congested region until pre-present packets are processed, or The congested routers can abandon queued packets to provide the room for new arriving packets. Without congestion control, a sending node, may continue transmitting packets that may be dropped later due to congestion collapse, and extra processing time & bandwidth is needed for retransmission, that will cause to increase the unnecessary load on network and network get congested.

Congestion handling can be broadly divided into the following:

- **Congestion Recovery:** Restores the operating state of the network when demand exceeds capacity.
- **Congestion Avoidance:** Anticipate congestion and avoid it so that congestion never occurs.

The meaning of congestion is that the load in the network is much more than the resource i.e. router can handle. In this situation, solution can be either to decrease the load or increase the number of resources.

To increase resources below mechanism can be used:

1. By increase the transmission power of a satellite we can increase the size of bandwidth.
2. Congestion is caused by slow links.
3. Use multiple routers to split-up the traffic rather than a single best route.
4. Use of dial-up links.

However large buffer space cannot solve the problem of congestion control, as queues and delays can get so long that when the packet come out from the router, most of them have already timed out and need to retransmit. A moderated buffer size should be used which is compatible with the packet lifetime or may be less than the packet lifetime, as no one packet reside in the queue which is already timed out. High speed links can also not work in this situation as the high speed networks get connected via low speed links and congestion at the point of interconnection become a problem. Even high speed links can help to control the congestion and can improve performance with one effective congestion control algorithm, that can detect the traffic and manage before it get congested. With high speed links, the arrival rate to the first router became much higher than the rate of departure, leading to long delay, buffer overflows and packet losses.

Congestion control required not only to prevent congestion collapse in the network but also to improve the network utilization. With increasing in speed of network the problem of congestion becomes a critical issue. It is very difficult to control congestion using conventional system. Fuzzy logic based congestion controller can be helpful to estimate the performance of network and adjust the traffic without any loss.

In this research I have proposed a congestion control algorithm which is based on fuzzy logic that calculate the congestion gradient factor on the basis of Available buffer space & difference between the number of incoming

& outgoing packets with in a particular router. On the basis of congestion gradient factor it perform the necessary action to avoid the congestion.

TCP Congestion Control

TCP (Transmission Control Protocol) provides congestion control services to application layer in Connection oriented network. It uses open loop congestion control to prevent congestion and closed loop congestion control to remove the congestion in network when it occurred. The basic idea of TCP congestion control is that each sender have to transmit just the right amount of data to keep the network resources utilized but not overloaded. To avoid network congestion and receiver buffer overflow, the maximum amount of data that the TCP sender can transmit at any time is the minimum of the advertised window[1] and the congestion window[2]. The TCP congestion control algorithm dynamically adjust the congestion window according to the network state.

This algorithm has 4 phases:

1. Slow start,
2. Congestion avoidance,
3. Fast-retransmit and
4. Fast recovery.

In first phase the size of congestion window grows exponentially on the receiving of each ACK. The reason is to fill an empty pipe as early as possible for the better utilization of network. Slow start stops when the congestion window reaches a value specified as the congestion threshold, which is initially set to 65,535 bytes. At this point the next phase (Congestion Avoidance) takes over. In this phase network in running close to full utilization. In this phase the congestion window size increase linearly rather than exponentially to avoid congestion. The congestion window stops increasing when TCP detects that the congestion occurs. At this point the congestion threshold is first set to one-half of the current window size (the minimum of the congestion window & the advertised window, but at least two segments). Next the congestion window is set to one maximum sized segment. Then the algorithm restarts with slow start phase. The basic assumption of this algorithm is that a segment loss is due to congestion rather than errors. It is quite valid in wired network where the percentage of segment loss due to error is less than 1 percent. However, the assumption may not be valid in a wireless network where transmission errors can be relatively high.

TCP Variants

In order to improve the performance of the network, several mitigation techniques have been suggested over standard TCP Version:

- **TCP Tahoe:** TCP Tahoe is one of the first TCP implementation for congestion control which was suggested by Van Jacobson. It included slow start, Congestion avoidance and fast retransmits. In TCP Tahoe, RTO (Retransmission TimeOut) is an indication of congestion and enter congestion avoidance phase[17]. It then sets the congestion window (cwnd) to 1 and slow start thrash (ssthresh) to cwnd/2. The problem with Tahoe is that it takes complete timeout time to detect packet loss and sometimes even longer because of the coarse grain timeout.
- **TCP Reno:** TCP-Reno uses fast retransmit and fast recovery to handle packet losses. It retransmit the lost segment Whenever 3 duplicate ACKs received, without waiting for timeout and enters fast recovery. In fast recovery, ssthresh & cwnd is set to half of the value of the current cwnd. TCP- Reno cannot effectively detect multiple packet loss within the same window.
- **TCP New Reno:** TCP New Reno is slight modification of TCP-Reno that can improve performance during fast recovery and Fast Retransmit mode. It can detect multiple losses within the same window. In phase of Fast Recovery, when a partial ACK is received which acknowledge some segments but

not all of the outstanding segments at the start of Fast Recovery period. In TCP Reno, Partial ACKs lead the sender out from Fast Recovery resulting in a timeout in case of multiple segment losses. In New Reno Partial ACKs don't take out TCP from Fast Recovery. Instead it assumes that the segment immediately after the most recently acknowledged has been lost and thus lost segment retransmitted. It remains in Fast Recovery until all of the data outstanding when Fast Recovery began has been acknowledged. TCP New Reno does not wait for a retransmission timeout and continues to retransmit lost segments every time it receives a partial ACK therefore it is more suited than TCP Reno to the mobile wireless environment where packet losses may occur in bursts. The New Reno suffers from, is that to sense packet loss it need one RTT for each packet. After receiving the ACK for first retransmitted segment than only we can understand which other segment was lost.

➢ **TCP Vegas:** Vegas is fundamentally different from other TCP variants as it doesn't wait for loss to trigger congestion window reduction. Vegas read & record the system clock when the segment is sent & ACK arrives and do the RTT calculation to check the difference between the current time and the time stamp recorded. If the difference is greater than the timeout value then Vegas retransmit the segment without having to wait for 3 duplicate ACKs.

It calculates the expected throughput as:

Expected = WindowSize / BaseRTT

Where the WindowSize is the size of the current congestion window. If congestion exists the actual throughput will be less than the expected throughput. The difference between the expected and actual throughput is maintained by the variable Diff, If Diff< α , a Linear increase of cwnd takes place in the next RTT; else if Diff > β, cwnd is linearly decreased in the next RTT. The factors α and β (usually set to 2 and 4) represent too little and too much data in network, respectively. This is the Vegas congestion Avoidance scheme12. In Reno, it is possible to reduce the congestion window size more than once when loss occurs during one RTT interval. In contrast Vegas only decrease the congestion window if the retransmitted segment was previously sent after the last shrink. Vegas detect losses much sooner than Reno.

TCP SACK

When the TCP Segment has been sent & the retransmission timer expires, the sender is forced to resend the segment and it is assumed that the segment has been lost. It may be possible that later these segment arrived at the receiver then receiver has no way to inform sender that it has received other segments because of the requirement of the ACK only in contiguous manner. Ideally the sender should retransmit only the lost segment(s) while the receiver continues to queue the later segments. TCP SACK is a technique that can help to reduce such type of unnecessary retransmission. Sack algorithm allow a TCP receiver to acknowledge out of order segments selectively and the sender retransmit the only the missing data segments instead of sending all unacknowledged segments. The major problem with SACK is that we will need to implement selective acknowledgement which is not easy task.

TCP Westwood

TCP Westwood improves the performance of TCP Reno in wired as well as wireless networks. There are two variants of TCP-Westwood, one is based on TCP Reno and other is based on TCP New-Reno. TCP WestwoodNR relies on end to end bandwidth estimation to discriminate the cause of packet loss, which is a major problem in TCP-New Reno.

Existing Algorithms

Drop Tail as the name suggests drops packets from tail of the full queue. it is one of the simplest and most widely used congestion control algorithm in the current internet routers, which drops packets from tail of the full queue. It works on FIFO based queue of limited size which simply drops packets when queue becomes full. Advantages included its simplicity, suitability to heterogeneity, its decentralized nature and better link utilization. Disadvantages are lake of fairness, no protection against the non responsive flows (Flows where the sending rate is not reduced even after the congestion indicated) and no relative Quality of Service.

Droptail-TCP Reno routers have Two major drawbacks are its lockout behavior (monopolizing of available bandwidth by a single or a few sources which is usually the result of global synchronization) and the full queue phenomena. Full queue is a serious problem when which refers to the situation when queue becomes full for long period of time, which results large end-to-end delays. Some other scheme like Random drop on full queue solves the problem of lock-out. But unable to solve the problem of Full Queue. One of the possible solution of this is to detect congestion earlier and then accordingly acknowledge the sources about the congestion before queue gets overflow.

One of the mechanism adopt this strategy is Active Queue Management (AQM), it detect & reacts before buffer overflow occurs. It drops packet actively to notify the traffic sources to slow down its transmission.

RED (Random Early Detection) is mainly proposed to be mainly used in the implementation of AQM. RED Signals incipient congestion to TCP by dropping packets before the queue becomes full with the drop probability depending on running average queue size (q_a). If the average queue size is between the minimum threshold (min_{th}) & maximum threshold(max_{th}) then the packet is marked/dropped with a linearly increasing probability depending on average queue size .if the average queue size exceeds the maximum threshold then every packet is dropped.

If $q_a > max_{th}$, then the packet is dropped. If $q_a < min_{th}$ then the packet is forwarded through. RED uses randomized approach to solve both the lockout and full queues problems. Though RED is better solution than drop tail but it fails for different network condition due to selection of congestion indicators.

CHOKe Algorithm, Whenever the new packet arrived in congested gateway router a packet is drawn at random from FIFO queue, and drawn packet is compared with the arriving packet . if both belongs to the same route then both are dropped else the packet that is chosen randomly is kept integral and the new arrived packet is admitted into the buffer with a probability depending on the level of congestion.

The computation of probability is same as in RED . However, this algorithm not present well if the number of flows is huge as compared to buffer space.

Virtual Queue Algorithm, In this algorithm a virtual queue is maintained in link with the same arrival rate as the real queue, but the capacity of the virtual queue is less than the capacity of real queue . When the packets are dropped in virtual queue then all packets already enqueued in the real queue an all arriving packets are marked until virtual queue becomes empty again.

Adaptive Virtual Queue Algorithm, It is same as the Virtual Queue Algorithm but the size of the virtual queue in this algorithm is same as that of real queue. At the arrival of each new packet the capacity of virtual queue is updated. However it does not follow the varying traffic pattern at flow in network. It is also FIFO based methodology.

A need for an effective algorithm arises in complex networks which can observe the traffic level able to take a prior step in order to avoid the congestion.

Releted Work

In Literature review researcher observe that there are many researchers who have worked on Congestion control. It shows that at present there is no any single algorithm that can resolve every problem of congestion control.

In, Author surveys various congestion control algorithms and evaluate their characteristic. It concluded that there is no single algorithm that can resolve all of the problems of congestion control on the network.

C. Socrates et al. presents a survey on available congestion control algorithm for packet switched network and an overview of category of congestion control. It also conclude that there is no single algorithm that can resolve every problems of congestion control on the computer network.

Shakeel Ahmad et al. concentrates on comparative study of different congestion control schemes based on some key performance metrics (i.e. Throughput, Mean Queue Length, Packet loss probability, Link Utilization, End-to End Delay).

Dina Katab et al. Presents a simple definition for congestion based on supply and demand of resources. Discussed a number of feedback mechanisms, for e g Feedback Messages, Feedback in routing messages, rejecting further traffic, probe packets. However these schemes are not desirable, as feedback is sent only during low load, & absence of feedback automatically indicate a high load. Various policies of different layers of network has discussed which affect the performance of network such as packet queuing policy at network layer affect the resource allocation among users. Various characteristics of network based on its fundamental architecture has been suggested to implement for better congestion control.

Abdulkadir Ibrahim et al. Presents a modified Fast recovery algorithm to improve the performance of the TCP New Reno. This modified algorithm with TCP New Reno improves its performance against both the throughput and packet delay.

KP Vijay et al. Proposed a new congestion avoidance mechanism coupled with authenticated mode of data transfer, which relies on the exchange of feedback between routers at the border of a network in order to detect and restrict unresponsive traffic flows before they enter the network and to transmit data securely by employing cryptographic technique. This mechanism ensure that packets do not enter in network faster than its capacity to proceed, hence packet is transferred without loss.

The different behavior of Wired & Wireless network has been discussed in [10]. In wired network CSMA/CD (Carrier sense multiple access with collision detection) keeps on listening the channel while transmitting. By comparing the transmitted signal & sensed signal, it detect collision and aborts its transmission immediately. On the other hand wireless network rely on CSMA/CA(carrier sense multiple access with collision avoidance), the transmitter can not transmit & listen channel simultaneously and channel condition is also different at the transmitter & receiver. Because of these, the transmitter first transmit the packets and then wait for ACK, if ACK doesn't come within the timeout duration the collision assumed and packet is retransmitted . The channel is wasted for such unnecessary retransmission. To avoid this problem CSMA-CN a new technique is proposed, in which transmitter uses two antennas, one for transmission and other for just listening. CSMA-CN is an attempt to approx. CSMA/CA in wireless networks. It thus improve performance at the cost of additional antenna at transmitter and correlation at both the ends.

An enhanced algorithm, called Fuzzy –AQM is suggested using fuzzy logic system to achieve the benefits of AQM in [13]. Uncertainty associated with queue congestion estimation and lack of mathematical model for estimating the time to start dropping incoming packets makes the Fuzzy –AQM algorithm best choice.

In, A Fuzzy inference system implementation for Drop tail, ADT-FL is proposed which regulates the queue size of router buffer based on prevailing traffic condition and available link bandwidth preventing the router buffers from becoming full when congestion occurs. It take two inputs, Traffic Intensity and Available bandwidth and the output of this model is Queue size parameter.

A fuzzy based congestion control scheme has been developed in. The fuzzy input parameters such as delay, buffer size and flow rate for each packet is considered to produce a congestion gradient factor. Depending on congestion gradient factor the flow rate is maintained same or reduced.

Yogini Bazaz et al., proposed a mechanism to control the congestion in streaming media application by using fuzzy logic. The inputs used in this system are TCPresponse, change rate and Available bandwidth. Based on these input parameters six rules are defined, which sets the send Rate as output parameter.

A modification of TCP Westwood congestion control algorithm by using fuzzy controller to enhance its performance has been proposed in. The proposed functional system consist of three inputs RTT (Round Trip Time), Ratio (Ratio of number of timeout to the number of dupacks), and diff_ratio (The ratio the time difference between two consecutive timeouts to the current estimate of retransmission timer's timeout interval). There are two outputs ssthresh (slow start threshold) and cwnd (congestion window) which specify the new phase of TCP to trigger after the packet loss event. In case of congestion it simply behaves like original TCP Reno. However after detecting a probable non-congestion, it does not throttle its transmission rate too much and continue to transmit thus does not remain unutilized network capacity in the presence of random bit errors.

Proposed Algorithms

Using conventional system it is difficult to define various condition of congestion control, as it contains only two truth values (true or false). A problem where a lot of uncertainty and variation in parameters a Fuzzy inference system can have a good option.

Fuzzy Inference system is a Rule based system which implements a nonlinear mapping between input & output parameters. It mainly consist four elements- Fuzzyfication, Defuzzyfication, Inference mechanism & Rule Base.

In Fuzzification process a crisp input (a real number) is converted into the linguistic value such as low, High etc by fuzzy sets through membership function. In FLS a single crisp input can have more than one linguistic value depends on membership function. Defuzzyfication is a process used to convert fuzzy outputs to a crisp value using Rule base. The Rule base can be defined as a set of rules and the Inference mechanism is that which emulates the decision making process.

In proposed work, there are two input parameters and one output parameter. Inputs are the Available buffer space ($Avail_{BS}$) and the Differentiation in number of incoming packets & the number of outgoing packets (Diff) and the output is Congestion Gradient factor (CG_{factor}).

There are 3 linguistic values for each input which are low, moderate and high. And on the basis of inputs the CG_{factor} calculated with a set of fuzzy rules.

Output function CG_{factor} is classified into five linguistic variables as CG_{factor} ={VeryLow, Low, Medium, High, Very High}.Once the CG_{factor} is determined from inference rule the defuzzification is performed to obtain the final value of CG_{factor}.

Table: Fuzzy Rule Base for CG_{factor}

Diff	Available buffer space ($Avail_{BS}$)		
	Low	Moderate	High
Low	Low	Low	Very Low
Moderate	High	Medium	Low
High	Very High	High	Medium

Algorithm 1

$Avail_{BS}$ = Available buffer space, Diff = number of incoming packet – number of outgoing packets, CG_{factor} = Congestion gradient factor, Min_{th} = Minimum threshold and Max_{th} = Maximum threshold.

```
Start
Step 1 : Get the current value of $Avail_{BS}$ and Diff .
Step 2 : Call Algorithm 2; // to compute the value of $CG_{factor}$
Step 3 : If $CG_{factor}$ is less than the $Min_{th}$ then forward all incoming packets through.
Step 4 : If $CG_{factor}$ is in between of $Min_{th}$ and $Max_{th}$ then mark all enqueued packets and all incoming
         packets to notify the sender to decelerate.
Step 5 : If $CG_{factor}$ is more than the $Max_{th}$ then all arriving packets are dropped.
Step 6 : goto step 2;
Stop.
```

Algorithm 2

```
Start
Step 1 : initialize fuzzy controller with $Avail_{BS}$ and Diff.
Step 2 : find the Membership values for $Avail_{BS}$ and Diff using triangular rule.
Step 3 : Computer the $CG_{factor}$ by referring rule base.
Step 4 : Inform $CG_{factor}$ to congestion controller.
Step 5 : Return
Stop
```

Observations, Results and Discussion

From the study of previous researches it has been observed that A number of research work has been published on the congestion control mechanism, but still there is no any single algorithm that can be able to achieve all the goals of congestion control and avoid packet loss . Because of limited bandwidth and the unpredictable behavior of users the performance of network is uncertain. Fuzzy logic is best suited to work in this situation. The proposed algorithm is fuzzy based algorithm, which able to detect the congestion prior to its occurrence and perform the necessary action to avoid congestion.

Congestion is the main reason of performance decline. The problem of congestion occurs when the multiple users contend to access for the same resources i.e. Bandwidth, buffer space and queues. The main goal of this implementation is to prevent the routers to get congested. Proposed algorithm make the appropriate decision by calculating CG_{factor} on the basis of available buffer space and on time difference between number of incoming & outgoing packets.

Drop tail Policy is mostly used in recent network for Queue management, which drops all incoming packets when queue becomes full, without indicating to sources. Full queue & Lockout are the main problems of Drop tail. AQM actively dropping the packets to notify the traffic sources slow down its transmission rate. Both mechanisms drop packets to indicate congestion. RED (an implementation of AQM) sets minimum & maximum threshold values, if Average queue size >Min_{th} it starts to drop packets based on the probability of average queue size and if Average queue size >Max_{th} then drop every packet. RED is better solution than the Drop tail but still there is packet loss to indicate the congestion. In paper [14] the proposed mechanism adjusts the queue size based on traffic intensity and available bandwidth. But we already discussed that the large queue size is not a good solution of congestion control. Instead of these the proposed algorithm can perform better as it indicate the sender to decelerate before the occurrence of congestion.

TCP uses end-to-end congestion control rather than the network supported congestion control. TCP/IP does not provide any services to prevent the buffer of intermediate routers form getting overflow. The proposed system can be helpful to identify intermediate routers that can get congested, by estimating the processing load and available buffer space & also perform necessary action to avoid congestion. The proposed algorithm is capable to perform well in wired/wireless network especially in mobile ad-hoc network where each node can work as router.

CONCLUSION

In this paper, a new fuzzy based congestion control algorithm has been proposed. When the number of incoming packets are more than the forwarding capacity of routers, buffer space filled up actively & all incoming packets need to be dropped, because of unavailability of buffer space. At this point network professed to be congested. The proposed algorithm is able to detect the traffic load and available buffer space prior to network get congested and accordingly perform the necessary action to avoid packet loss. Using this algorithm if it is found that rate of incoming packets is higher than the capacity of network than it slow down the sender to protect the network from getting congested.

After a thorough study of the different available algorithms, it can be concluded that none of the perfect algorithms are available that protect the network from packet loss as well as increase network performance. With limited bandwidth and unpredictable behavior of user a fuzzy logic based algorithm can be a good option to protect the network from getting congested as the proposed algorithm.

Hence, the proposed algorithm can provide better performance than all existing algorithm as it can proactively identify the congestion prior to its occurrence and applied to protect the network from congestion & packet loss . The proposed algorithm is suitable with both environment wired and/or wireless networks. Unlike TCP it can also be used on intermediate routers to prevent from congestion.

REFERENCES

1. Abhale, Ashwini & Nagraj, Uma (2014) "A Reviewon Improving Performance by reducing Collision Overheads with CSMA-CN (Collision Notification)", *International Journal of Advance Foundation and Research in Computer(IJAFRC),* Volume 1, Issue 1, Jan 2014, ISSN No. : 2348-4853
2. Ahmad, Shakeel; Mustafa, Adli Bashir; Ahmad, Arjamand Bano and Hosam, (2009) Al-Sammarraie "Comparative Study of Congestion Control Techniques in High Speed Networks" *Internation Journal of Computer Science and information Security,* Vol. 6, No. 2.
3. Bazaz, Yogini; Kumar, Sudesh and Anand, Sanjay (2013) "Congestion Control Mechanism using Fuzzy Logic" *International Journal of Emerging Trends & Technology in Computer Science(IJETTCS),* Volume 2, Issue 2, March –April 2013 ISSN 2278-6856.
4. Computer Networks, 4th Edition, A.S. Tanenbaum, Prentice Hall of India, New Delhi.
5. Data and Computer Communication, 6th edition, William Stallings Pearson Edition, New Delhi.
6. Fahmy, Soniya and Karwa, Tapan prem "TCP Congestion Control : Overview and Survey of Ongoing Research" Purdue University, West Lafayette, IN 47907-1398.
7. Hemlata, Pandey, V. K. (2015) "Automatic Traffic Management and Congestion Avoidance", *Internation Journal of Advance Research in Computer Science and Management Studies,* Volume 3, Issue 4, April 2015. ISSN: 2321-7782 (online) available at www.ijarcsms.com
8. Ibrahim, Abdulkadir & Widiantoko, Ari (2016) "Evaluation of TCP RENO and TCP NEWRENO Using Congestion Control for Host to Host", *Internation Journal of Science Technology and Management,* Vol. No. 5, Issue No. 03, March 2016, ISSN 2394-1537.
9. Jain, Monal; Tomar, Deepak Singh & Tomar, Shiv Kuamr Singh (2015) "a Survey on TCP Congestion Control Schemes in Guided Media and Unguided Media Communication" *Internation Jouranal of Computer Applications* (0975-8887), Volume 118-No. 3, May 2015.
10. Katab, Dina; Handley, Mark & Rohrs, Charlie (2002) "Congestion Control for High Bandwidth-Delay Product Networks" SIGCOMM'02, August19-23.

11. Mishra, Akshay & Sinha, Nirmala (2016) “Congestion Control Issues & Trends”, Internation Jouranal of Advance Research in Computer Science and Software Engineering, Columе 6, Issue 4, April 2016. ISSN: 2277 128X. Available online at : www.ijarcsse.com, Assess on 20/02/2024.

12. Natsheh, Essam; Jantan, Adznan B.; Khatun, Sabira and, Subramaniam, Shamala (2007) “Fuzzy Active Queue Management for Congestion Control in Wireless Ad-Hoc” *The International Arab Journal of Information Technology,* vol 4, No. 1, January 2007.

13. Shalinie, S. Mercy; Preetha, G.; Mnidhya, S. Dina; Kiruthika Devi B. S. (2010) “Fuzzy AdaptiveTuning of Router Buffers for Congestion Control”, *International Journal of Advancements in Technology,* http://ijict.org, Vol 1, No 1, June 2010, ISSN 0976-4860, Assess on 22/02/2024.

14. Smitha, K; Anand, H. U. and Mallapur, J. D. (2011) “ FUZZY BASED CONGESTION CONTROL IN WIRELESS NETWORK”, *Internation Journal of Computer Science and Communication,* Vol 2, No. 2, July-December 2011.

15. Socrates, C.; Devamalar, P. M. Beuah & Sridharan, R. Kannamma (2014) “Congestion Control for Packet Switched Networks: A Survey” *International Journal of Scientific and Research publications,* Volume 4, Issue 12, December 2014.

16. Subramani, B. & Karthikeyan T. (2014) “A Review on Congestion Control”, *International Journal of Advanced Research in Computer and Communication engineering,* Vol. 3, Issue 1, January 2014.

17. Vijay, K.P. & Karthick, S.V. (2014) “Novel Implementation of Enhancing Reliability of Transmission on High Performance Network”, *InternationJournal of Advance Research in Computer Science and Management Studies,* Volume 2, Issue 1, January 2014, ISSN: 2321-7782(Online) available at www.ijarcsms.com, Assess on 27/02/2024.

18. Vijipriya, J. & Suppiah, S. (2016) “An Extended study on Newton Raphson Congestion Control” *Internation Journal of Advanced Research,* Volume 4, Issue 2, 574-581, ISSN 2320-5407.

19. Zainab T. Alisa, Sara Raad Qasim “A Fuzzy based TCP Congestion Control for Wired Networks”, *International Journal of Computer Applications* (0975-8887), Vol 89-No. 4, March 2014.

****