An Analysis of Congestion Control Protocols with Performance Comparison of Conventional Traffic Network with Real-Time Video to Very High-Speed Networks

Sunil Kumar Kushwaha
Department of Computer Science and Engineering, Medi-Caps University, Indore (M.P.)
sunilietkushwaha@gmail.com
Dr.Suresh K.Jain
Department of Computer Science and Engineering, Medi-Caps University, Indore (M.P.)
suresh.jan@rediffmail.com

ABSTRACT

As the number of Internet users and apps has rapidly increased, especially after the corona pandemic, network congestion has grown increasingly, which seems to be a serious concern. In today's scenario most of the traffic is real time video rather than text. Congestion has been a big problem since the day of inception of the network and will remain forever. Congestion control for TCP has been developed to guarantee Internet reliability with fairness and effective bandwidth distribution. This research paper presents a brief survey and comparison of various network congestion control algorithms during the previous 25 years for normal as well as high speed network management protocols. It introduces the metrics for judging the best congestion management technique with aforementioned protocols have been compared and analyzed, by using parameters such as stability, fairness, efficiency, throughput, delay, dropping, performance, bandwidth-utilization, responsiveness etc. For the proper management of congestion control, a number of models and analysis techniques are also provided, including the nonlinear model used for minimizing, the control theory, fuzzy and the optimization approach using neural network. This essay seeks to assess and contrast the various control algorithms such as Westwood+, New Reno, BIC TCP, CUBIC, FAST, High-Speed TCP, Layered TCP, Scalable TCP, and XCP, and vegas using MATLAB simulations, to measure the performance. After doing considerable comparative study on congestion control challenges, the protocols for end-to-end communication as well as intermediate nodes were generalized. A few unresolved issues have been discussed for future implementation and as a challenge in forthcoming days. The limitations of all protocols designed for conventional data, real time high speed networks as well as wireless have also been studied.

Keywords - **Congestion, Diffserv , High-speed Networks, Performances, Real time Video, Wireless Networks**

I. INTRODUCTION

The problem of congestion is ubiquitous and will remain forever. Since the day of inception, the problem of congestion started and is still active. A number of methodologies and ideas was proposed and few has been implemented and tested under different scenarios but no solution is perfect. A best solution is high resources availability in term of bandwidth, router buffer and processing speed, but it may leads to overprovisioning. Since its introduction into the TCP protocol in 1988, congestion control algorithms have played a critical role in averting congestion collapse on the modern Internet. The throughput of the network may become zero and the path time may increase significantly during congestion. A congestion control strategy aids in the network's recovery from a congested state, implements network congestion control with a minimum loss ratio and little latency, and maximizes the utilization of available resources.

Van Jacobson proposed an algorithm for congestion control in the last 1980s [1], which is named as Tahoe TCP, and further with mild modification, renamed and further known as Reno TCP, and subsequent variations detailed in [3]–[5], which makes foundation for the majority of the stability of the computer internet. Following the end-to-end or process to process design philosophy, the Van Jacobson proposed congestion control algorithm that has been quite successful in preventing congestion collapse on the Internet [18]– [20]. There is a difficulty in matching the input rate of TCP connections with the bandwidth available in the network as well as the comparison to the maximum size of congestion window (*cwnd)* while using slow start threshold (*ssthresh*)by using these two variables. All such algorithms applied for congestion control uses the Additive-Increase/Multiplicative-Decrease (AIMD) to develops a pattern, which multiplicatively increases the c_{wnd} to snag the available resources in terms of bandwidth and abruptly decreases the c_{wnd} when network maximum threshold capacity is reached. There are multiple ways through which network may experience congestion that took place in which TCP segment loss is one of the primary signs as well as three acknowledgement of same number is received. Although AIMD algorithms guarantee network stability, they do not ensure equitable resource distribution. [1], [6],[7], [21]. The algorithm proposed by Van Jacobson algorithm was first developed, later on, a number of end-to-end congestion management algorithms have been developed to enhance the stability of network, equitable allocation of bandwidth, with resource utilisation in very high-speed networks, real-time video as well as audio, and wireless networks.[2,3,4,9,12,14,20,32,33,39]. In actuality, wireless networks, the most popular TCP is not feasible as losses resulting from radio channel issues are considerable and cause an unnecessarily low transmission rate and hence the said topic is not covered during the discussion.

The existing implemented protocols for congestion control in TCP/IP differentiated service use a DSCP value to differentiate the different packets. All such different behavioral packets were assigned multiple size buffer to get stored in the queue of router. A protocol is designed in such a manner that high priority packets were picked more in number compared to low priority packets. These high packets were forwarded too quickly and thus services were prioritized. However whenever congestion took place by measuring the parameters an unanimous decision takes place for dropping the packet irrespective of type. To operate well over wireless lines, TCP needs additional protocols for Data link layer as dependable link-layer or split-connections method [2,10,21,33,34].

By offering a system of congestion detection before packet losses, the idea generated by Vegas TCP was the initial effort to minimize from the TCP's loss-driven approach [9]. Vegas TCP determines network congestion by calculating the difference between the expected rate and the actual rate (c_{wnd}/RTT_{min}), where RTT is the minimum measured round trip time and RTT_{min} is the round trip time. In particular, the c_{wnd} is additively increased if the difference is less than a threshold, and additively decreased if the difference is more than another threshold; In the event if the difference is equal to or greater than [9], the c_{wnd} is kept constant. Although Vegas TCP keeps the network stable, [11] claims that it is unable to capture when it uses its own bandwidth bargaining with Reno-like algorithms that frequently exceed the maximum capacity of the network queue.

Westwood TCP proposed a new idea for end-to-end calculation of estimated bandwidth which further forms the foundation of the new congestion control [12]. Looking into depth of Westwood TCP which counts and differentiate the flow of replied ACKs to estimate the available bandwidth, and following congestion, which adaptively sets the c_{wnd} and the ss_{thresh} by considering the predicted bandwidth. The first limitation which generates is that bandwidth estimation method doesn't work properly when there is ACK compression [41]. A significantly altered variation of the bandwidth estimate approach has been provided in [14] order to address the consequences of ACK compression. The term "Westwood+" refers to the original Westwood method plus an improved bandwidth prediction as well as estimation.

Additionally, it has been shown analytically in [14] that Westwood+ is more equal than Reno in terms of bandwidth distribution and friendlier to Reno TCP. The objective of preparing this research paper is to contrast Westwood+, RED, New Reno, BRED, BIC, TCP, CUBIC, GRED, FAST, High-Speed TCP, Layered TCP, Scalable TCP, and XCP, and Vegas. A better version of Reno, called New Reno, prevents the c_{wnd} from being reduced several times, when multiple segments from the data generated from single window are lost [3]. The most popular used congestion control protocol for Internet, New Reno TCP, has been taken into consideration [27]. Vegas TCP has been taken into consideration since it also suggests, like Westwood+, a novel method of regulating the congestion window based on RTT measurements of the network's congestion situation. Additionally, Vegas TCP offers the fundamental concepts for the fresh Fast TCP congestion control method that was just put forth by Caltech researchers [39]. According to the authors, "Fast TCP is sort of a high-speed version of Vegas" [40]. Fast TCP is still in the testing stage as of this paper, and neither the authors' kernel code nor their ns-2 implementation have been made public. It might receive every Vegas' drawbacks that has been described in this research, most notably the inability to secure bandwidth when Reno traffic is present or when reverse traffic is present, according to RTT measures to infer congestion.

In actuality, typical congestion control used for TCP is not completely effective for high-speed networks, and it is difficult to build a high-speed alternative to TCP. It has been discovered through comparison that several of the high-speed protocols now in use have convergence and stability issues. In order to solve these issues, an unique Congestion Control Algorithm Coupling Logistic TCP (CLTCP) is designed using a population ecology model. Like XCP and Max_{Net}, it consumes bandwidth pre-assignment as its foundation. The pre-assignment rate factor in the routers is determined using data on the capacities of router(R_{pc}) total incoming traffic(T_{inctr}), and queue length(q_l). Theoretical study and simulation findings demonstrate that, regardless of round trip time, CLTCP offers high utilization, fair bandwidth allocation, robust stability, and fast convergence.

Typically, when congestion control algorithms converge, it is examined how long it takes a transport control system to transmit from the starting state to the steady state. Efficiency convergence and fairness convergence are the two facets of this problem [15]. It is contemplating the possibilities that a new flow will use all of the link's available bandwidth as soon as it joins the network.

The consumers' expectations for the size, functionality, and performance of the Internet have increased as a result of its development. Given the network architecture, finding efficient flow management measures and control techniques will determine how to ensure the stability and efficiency of the network functioning in the manner we anticipate. The technique's developed has main objective is to implement the scheduling, queue management, congestion control, and other essential management at the proper levels and granularities scheduling, and flow forming, among other things, that are likely to show serious congestion and cannot be relieved. Therefore, assuming other QoS measures are effective as planned, congestion control should be the initial focus.

Understanding and grasping the present and upcoming demand of Internet behavior requires an in-depth understanding of how to manage the busty flow dynamic characteristics of network model. To sum up the current development, we must first conduct a detailed analysis on the dynamics model of network systems.

This analysis is divided into four section as section-1 covers the analysis of existing algorithms, section-2 covers the performance parameters on which comparison may be carried out, section -3 brings the equation on which the algorithm were designed and the parameter on which ones existence is maintained. The new proposed algorithm which suggests a better and optimum utilization of resources is finally placed in Section 4.

SECTION-I
Analysis of Existing Algorithms

The Additive Increase Multiplicative Decrease (AIMD) method [1] [10] which is widely used in TCP states that if the steady-state throughput of a TCP flow is P, then the period of efficiency convergence and the period of convergence TCP's fairness is O(P) [24], i.e., the AIMD algorithm. It implies TCP and converges linearly to efficiency and fairness. It will converge efficiency over a long period of time, and make justice in super-fast networks. TCP therefore tries to a slow-start algorithm to increase convergence in its beginning stage. However, the rate of convergence in congestion avoidance phase still moves slowly.

A few novel transport protocols that have been developed in response to these issues are listed as

(i) . The High Speed Transmission Control Protocol (HSTCP) [4],
(ii) Sensor Transmission Control Protocol (STCP) [12]
(iii) eXplicit Control Protocol (XCP) [11],
(iv) Exponential Max-Min Kelly Control (EMKC) [24],
(v) Variable-structure Congestion Control Protocol (VCP) [23],
(vi) Explicit Virtual Load Feedback - Transmission Control Protocol (EVLF-TCP) [9]

By employing more aggressive increasing and more cautious decreasing algorithms than the AIMD method, HSTCP and STCP increase convergence at the expense of a greater loss ratio. The RTT unfairness issue with HSTCP and STCP is made worse by this method than it is with TCP. In order to effectively distribute network resources to the end system, EMKC, VCP, and EVLF-TCP explicitly transmit back router state information such as the packet received to packet forwarded, the packets placed in the ingress port named, and virtual load factor. However, some other methods such as EMKC, VCP, and EVLF-TCP only enhance exponential convergence to efficiency, or the convergence to efficiency from O(P) to O(l*nP*). They continue to maintain their consolidation to fairness as O(P). Contrarily, XCP enhances consolidation by directly allocating bandwidth to each flow through the router, resulting in linking to efficiency and fairness that are both 0(1), or continuous consolidation. According to some study, in multi-congested gateway networks, XCP can be unstable and occasionally fail to meet Max-Min fairness.

We think the main factor is that XCP fails to provide good stability as well as fairness in complicated topologies because it is too sensitive to network traffic.

Stability is a crucial prerequisite for congestion control methods. Whenever there is a wide variation in terms of bandwidth, flow rate, and round-trip duration, the congestion control method must be stable and flexible. Numerous studies on the rationality of transport protocols have revealed stability criteria for various layered protocols, including:

1. RED application for fluid-based analysis of an AQM router network supporting TCP flows.
2. Control theoretic evaluation of RED, [18].
3. A state feedback control method for ecn-enabled tcp connections queue stabilisation [6]
4. Global stability of internet congestion controllers. - By R. Srikant and S. Deb. [2]
5. In regards to the internet's end-to-end congestion control's stability. G. Vinnicombe [20]
6. Hétérogenous feedback delays for distributed congestion control. Written by L. Massoulie. [17]

The network settings as well as the congestion control algorithm's control parameters determine whether or not these protocols are stable. Therefore, the network characteristics have limitations on the stability of these protocols. These protocols may be unstable if the parameters defined for networks are not in an area that satisfies these stability criteria.

Therefore, it is crucial to make more potential transport protocols by giving strength to converge and minimise or completely do away with the impact of network characteristics on the stability of the transport protocol.

Congestion control algorithms for Differentiated Service Network

Binary Increase Congestion Control (BIC): Utilising the additive increase and binary search increase the size of window which uses control methodologies, the BIC [1] and [18] congestion control approach reduces traffic congestion. When the congestion window is large, additive rise with a huge increase in size or exponential increment provides both strong scaling and square RTT unfairness. During brief instances of congestion, increased binary search encourages TCP friendliness.

Increased binary search: The system gives information via packet loss on when the instantaneous transmitting rate (or window) exceeds the available bandwidth named resources available here, where congestion control is considered as a finding problem. The starting points for this search are the current minimum window size W_{min} and maximum window size W_{max}. W_{min} is the window size immediately after the fast recovery, while W_{max} is normally the window size shortly before the most recent fast recovery (i.e., the time when the most recent packet loss occurred). Repeatedly calculating the midpoint between W_{max} and W_{min}, setting the current window size to that midpoint, and watching for feedback in the form of packet losses are the steps of the method. Based on this feedback, if there is a packet loss, the midpoint is assumed to be the new W_{max}, and if not, the new W_{min}.

The cycle is repeated until the minimum increment (S_{min}), or the difference between W_{max} and W_{min}, is reached. This method, known as binary search increase, enables bandwidth probing to start off aggressively when the current window size and the goal window size are far apart, and it allows bandwidth probing to become less aggressive as the current window size approaches the target window size. The increment at the saturation point for the other scalable protocols, on the other hand, tends to increase so that it is the highest in the current epoch (defined as the interval between two consecutive loss events). Typically, the amount of the increase just before the loss is inversely correlated with the number of lost packets. Thus, increasing binary search can decrease packet loss.

The primary benefit of binary search is the concave response function it offers, which completes the additive climb described below.

Additive Increase: A binary search increase is jointly added with an additive increase approach to provide quick merger and RTT-fairness. Increasing the window size directly to the midpoint may put too much strain on the network if the distance from the present minimum to the midpoint is too large.

In binary search increase, the window size is expanded by S_{max} until the distance from the current window size to the goal is less than S_{max}, at which point the window expands directly to the target. S_{max} is a predetermined maximum step, as a result, the approach increases the window, first linearly and then exponentially, following a significant window decrease. Under wide windows, binary increase approaches pure additive growth when combined with a multiplicative drop technique. This is so that there will be a longer additive rise period since a larger window causes a larger reduction in multiplicative decrease. The window size approaches pure binary search increase, i.e., a shorter additive increase time, when it is small.

Advantages: The BIC TCP effectively probes the available bandwidth using a binary increment method [1], [18], [28]. BIC TCP does not exacerbate the traditional TCP's RTT fairness problem while achieving high throughput [1].

CUBIC: A New High-Speed TCP Variant that is TCP-Friendly an improved form of Binary Increase Congestion Control (BIC) is CUBIC [13, 14]. It increases the TCP neighborliness and RTT fairness of the BIC window control while also making it simpler.

The exponential growth function of the BIC may still find it overly forceful for TCP, especially in short RTT or low speed networks, despite the fact that BIC provides rather decent scalability, fairness, and stability in the current high speed environments [13], [26], and [32]. The various separate window control phases make it very difficult to analyse the procedure.

The growth of window function in CUBIC is an exponential function, which resembles the growth function in BIC in terms of shape. CUBIC is intended to augment and simplify BIC's window control.

An improved form of Binary Increase Congestion Control (BIC) is CUBIC [13, 14]. Upon window reduction, the growth function of CUBIC with the origin at W_{max} expands very quickly, but as it approaches W_{max} its growth decreases. The window increment almost disappears at W_{max}.

Above that, CUBIC begins to probe for additional bandwidth, and as it moves away from W_{max}, the window's development picks up speed. While the rapid expansion away from W_{max} ensures the protocol's scalability, the gradual growth around W_{max} improves the stability of the protocol and raises network utilisation. The cubic function makes guarantee that competing flows of the same protocol are treated fairly within it. Due to the fact that all competing flows with various RTTs would have the same to following a synchronised packet loss, linear RTT fairness is ensured (notice that TCP and BIC provide square RTT fairness in terms of throughput ratio).

Window increments are constrained to be no greater than *Smax* per second in order to further improve fairness and stability.

Due to the fact that BIC raises the window in an additive manner when the window increment per RTT exceeds a certain threshold, this feature ensures that the window continues to grow linearly when it is far from *Wmax*.

While under short RTTs the linear increment per RTT is less but remains constant in real time, it is ensured in CUBIC that the window is growing linearly and in real time dependent.

CUBIC has a bigger area of the TCP friendly region than HSTCP, which is TCP friendly when the loss rate is greater than 0.01%. Furthermore, regardless of loss rates, CUBIC is far more TCP friendly than HSTCP when the RTT is very tiny.

Benefits: CUBIC keeps the scalability and stability of BIC while improving its fairness characteristics [13]. The RTT fairness of the growth function is ensured since various RTT flows will continue to increase their windows at the same rate [13]. Both queueing latency and packet loss are addressed by FAST [15].

BaseRTT - The overall amount of packets held at routers along the flow's path that are balanced is determined by BaseRTT. BaseRTT is a positive protocol parameter. According to the mean RTT and mean queueing delay provided by the estimate component, FAST periodically updates the congestion window under typical network conditions.

Fast conceptualises the network as a collection of finitely-capable resources, such as transmission lines, computing units, memory, etc. These resources are collectively marked as "links" in the model. The network is shared by several distinct unicast flows, recognized by their sources.

Benefits: Using the multi-bit information that the queuing delay provides, FAST TCP directly calculates the congestion window. It is claimed that this technique outperforms the 1-bit flag loss indication used in standard, Scalable, and High Speed TCP [8], [23], which is supported by a collection of unicast flows that can be identified by their sources, in terms of responsiveness, stability, and fairness. One significant change from those is the protocol model's.

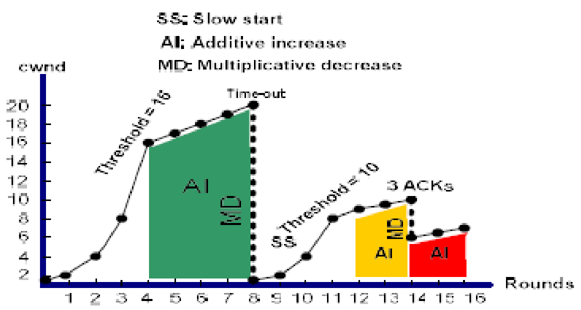

Fig:1 Additive Increase/Multiplicative Decrease (AIMD)

Some more models suggested are summarized as below:-

1. RED[5] is the first algorithm proposed by Flyod S.Jacobson in year 1993 , which emphasizes on random early detection instead of previous proposed algorithms in which action carried out was after incident took place i.e. traditional tail drop method. It uses minimum and

maximum threshold in order to foresee overcrowding before buffer overflows.
2. GRED[6] an another method proposed by Flyod in 2000 by calculating active queue length instead of active length. The rules for deleting packets have changed since it now employs three thresholds: a minimum, a maximum, and a double maximum (2*max). Performance analysis clearly shows that GRED is better than RED.

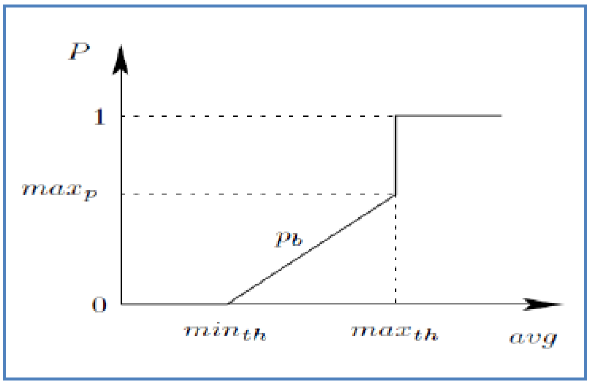

Fig.2 RED Algorithm

3. Effective RED (ERED) utilize q and aql to prognosticate and calculate dropping probability. The model was proposed by Korukoglu[7] in 2009 in which dropping rules says that if min< -aql < max and q>min then the packets are dropped similar as of RED. However if aql<min and q<1.75 \times max then the packets are dropped on higher probability.
4. 4. Bakilizi et al.'s DGRED [8] model, which utilize the dynamic max and double-max thresholds to maintain the active queue length between the lowest and maximum thresholds and at a predetermined target value known as target $a_{ql}(ta_{ql})$, uses another model. According to performance analysis, DGRED performs better than RED and GRED.
5. To come out from abovesaid problems, Ott et al. set up an predescribed stabilized RED(SRED)[9] which utilizes the active queue q and number of flows to detect and minimize the congestion. To prevent congestion, the packets are dropped depending on a probability determined by the number of flows..
6. An another method in which active queue(q), instead of average queue is proposed by Jamali et al. in 2014[16]. The number of flows decide the dropping of packets. The response is to the current queue rather than to long term and avoid congestion.
7. In order to come out from old traditional method as mentioned from 1 to 6 , Chrysotomou et al. in 2000 proposed Fuzzy Explicit Marking(FEM). In order to avoid Internet traffic having a linear pattern, and making amendment in the corresponding parameters are not sufficient to make stable and robust method. In order to deal with non-linear systems,
8. A more efficient congestion control method named as FLRED[19] is based on a fuzzy logic network congestion control method. FLRED fared better than RED and effective RED (ERED) in situations of high congestion by reducing packet load and latency. An another method proposed in name of [12] Fuzzy logic is used by Intel's rate controller, which is comparable to FEM and uses it to get beyond the linearity and parametrization issues that RED and its derivatives had. It forecasts congestion and computes probability drop using active q. According to the simulation performed in this manner, Intel Rate Controller is more reliable and efficient since it increases throughput, link utilisation, and minimises delay.
9. Woodward et al. [18] suggested the REDFL, a version of RED that incorporates fuzzy logic into the RED algorithm. Packet loss was a new congestion indication added by REDFL which was the only indication used by RED, a_{ql}. As a result, REDFL leverages Packet loss and a_{ql} as linguistic input variables for a fuzzy logic system. This procedure lessens RED's complex parameter dependence and settings. The goodput of the mentioned fuzzy logic system is the calculation at which packets are dropped in an effort to alleviate congestion. In contrast to REDFL, which simulates discrete-time queues, RED is an algorithm that uses a different approach. The outcomes highlighted that REDFL shows signific improvement on RED based on parameters such as throughput, average queue length, packet loss rate, and likelihood of dropping of packet.
10. E. Jamhoura et al. [3] offer a protocol for developing a predictor using fuzzy to simulate a differentiated services (DiffServ) node with two separate queues—one for voice over IP (VoIP) traffic and the other for remaining connection-oriented as well as connection-less data traffic. They construct the fuzzy membership functions and apply a model using fuzzy to generate network traffic controllers based on extending the existing queue models..
11. A fuzzy based controller for traffic differential services was proposed by M. Yaghmaee et al. Their fuzzy scheduler is based on the waiting fair queue system, in which the fuzzy controller modifies the importance of each queue. A two input, with output fuzzy controller is used by the authors to dynamically adjust the committed interface rate. The findings are superior than non-fuzzy systems in terms of performance.
12. S. Shalinie et al.'s research team [5] uses a fuzzy set to define the system's input and output for regulating queue size. Two inputs—instantaneous bursty data flow and the instantaneous bandwidth available are used in the proposed model. This model's output is the parameter for queue size.

13. Fuzzy logic-based Adaptive Drop Tail, According to A. Mishra [7], significantly improves congestion control without requiring any further parameterization or tuning. The results demonstrate that, when compared to conventional Drop Tail methods, their suggested Adaptive Drop Tail Fuzzy Logic controller has lower packet loss. The simulation is carried out and further cultivated to make a balance between adaptive buffer space whenever an unexpected shift in overloading occurs, preventing Internet the memory of router from getting fully consumed when overloading took place.

SECTION 2
Examining the shortcomings of these Protocols.

Researchers [3, [4], [6], [17], [21], [23], [26], [28], [30], [31], and [32] have demonstrated that the protocols designed to allow collaborative communication, real-time data, as well as high-speed data transmission via multiple gigabit lines display some inefficiencies under a variety of networking environment and network traffic scenarios.

Making an analysis of network in terms of performance for the protocols developed for conventional to acknowledgement and further to very high-speed networks specified and defined above, we are taking into account the following measures and researching their deficiencies.

1. Fairness [38], [40]
2. Throughput [38], [40]
3. Bandwidth utilization [38],
4. Stability [40]
5. Responsiveness[39]
6. Packet drop[32,39.41]

BIC-TCP: If a packet drops early, BIC-TCP may experience problems during its initial climb because the maximum link capacity was not properly predicted [30]. Despite employing a quick Convergence approach, BIC-TCP might not be able to fairly allocate available resources to contending traffic [30], [42], [43], and [45].

CUBIC: It is discovered that CUBIC takes a long time to increase its c_{wnd} to fully occupy the connection. Due to the inaccurate calculation of the maximum link capacity, it suffers significantly, more so than BIC TCP, by dropping packets too early on its initial stage.[30], [42].

No matter the type of background traffic, FAST encounters inequality and precariousness in networks with a least size of available buffer or a lengthy delay. [26]. The main drawback of **FAST TCP** is that its performance might be impacted by reverse traffic and that its control parameter "" needs to be manually set up [26]. When the router queue size is less than "" [30], [42], [43], FAST performs poorly.

HSTCP: Its compatibility with typical TCP flows has been found to be lacking [31]. While HSTCP's fairness is good regardless of the background traffic types, it trades stability for fairness because of the higher degree of global loss synchronisation that the protocol induces among competing flows, which lowers link utilisation.

LTCP: Studies are being done to determine its inefficiencies.

STCP: According to [11], [29], the fairness equilibrium may not be reached using the Scalable TCP. Scalable TCP may result in a greater RTT bias. [1].

XCP: To maintain a consistent understanding of the network state, XCP [4] is highly dependent on the returned ACK packets. This reliance, when used as it is on dynamic, highly high-speed networks, significantly affects the XCP performances in the event of ACK losses on the backward track, which makes XCP exceedingly unstable. XCP must maintain RTT for each connection. requires router involvement, deployment may be challenging, unfair to connections with longer RTT, and malicious The header can be fabricated by the sender, and this can lead to inaccurate feedback calculations [22], [42], [43], [44], and [45].

Induce a reduced link utilisation due to increased global loss synchronisation between competing loads.

SECTION 3
PERFORMANCE ANALYSIS

Differential equations can be used to define the system's worldwide asymptotic stability and predict when convergence will happen. The right CLTCP parameters can be selected by doing an analysis of the effect of performance control parameters.

Stability

3.1) No matter the capacity of the constriction, the number of flows, or the round-trip time, the differential equations of the network's whole control model describe a system that is globally asymptotically stable. It indicates that even when a control system's parameters jitter, the system can continue to function normally. The scheme must function in an unreliable (random) environment in order to be robust. Therefore, plans that assume a specific distribution (exponential) for service times or plans that only function for deterministic ~~service times were disregarded~~.

$$S = \frac{1}{x_i}\sqrt{\frac{1}{m-1}\sum_{k=1}^{m}(x_i(k) - \bar{x}_i)^2}$$

Eq.-(1)

3.2) Equity. Let the number of users sharing a single resources is n, fairness requires that individual user receive the equal amount of the resource (Unless the consumer requested a lower percentage than what was fair). Using the following equation from Reference [2], fairness can be measured even if throughput is not perfectly equal.

$$E = \frac{(\sum_{i=1}^{n} x_i)^2}{n\sum_{i=1}^{n} x_i^2}$$

Eq.-(2)

3.3) Efficiency. When the distribution of resources is Pareto optimum, no user can achieve better satisfaction through a different distribution of resources without also lowering the satisfaction of the other users. Therefore, Pareto optimum will ensure that congestion control algorithms are valid. Power can be used to measure how effectively resources are allocated in Reference [3].

$\eta = Goodput/Response\ Time$

Goodput= sent_data – retransmission_data/transfer time

Eq.-(3)

3.4). Scalability. With regard to bandwidth, access, user count, and other factors, the internet has advanced quite quickly. The physical expansion, such as from a local area network (LAN) to WAN and further to a wireless network link(wifi), must be supported by a congestion management protocol. Additionally, it must be able to receive a variety of expansion possibilities.

3.5) Realistic. The technology requirements are all included in the feasibility. Due to the complexity of the hardware and software systems used by the Internet, congestion management protocols must incorporate a number of distinct technologies. For the congestion control protocol to be practical, it must also be sufficiently simple and simple to implement.

When an optimal protocol for congestion control is created, the evaluation index mentioned above was significant and required. To meet the evaluation index in practise, we often employ several control mechanisms dependent on the controlled object.

3.6) Response: The responsiveness index [41] gauges how quickly convergence occurs when the network equilibrium

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

shifts when k = 1, or when new or existing flows enter or exit.
Let $x_i(k)$ represent the instantaneous average for the interval k–m:

$$\bar{x}_i(k) = \frac{1}{k}\sum_{t=1}^{k} x_t(k)$$
Eq.-(4)

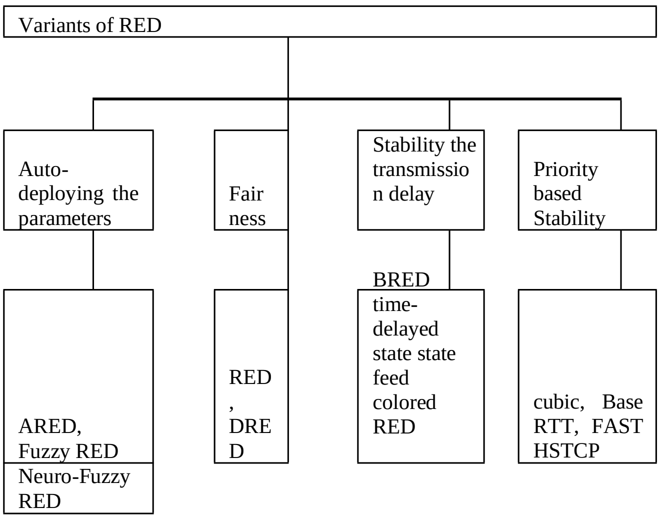

Fig.3 Variants of RED

SECTION-IV

In all of above method proposed and as mentioned used for controlling the congestion in TCP/IP Differentiated services use one common approach for dropping the packets in case maximum threshold achieved. A new multi-level matrix based tabular method is proposed which created a new paradigm in the field of congestion management. Instead of dropping the packets uniformly, a multilevel dropping approach i.e. there will be multilevel threshold level which will be achieved by using fuzzy logic.
To make it more elaborative, if packets belongs to real-time video communication maximum threshold will be at higher value compare to recorded video data.

Thus based on bandwidth allocated, instantaneous queue size and average queue size, the dropping probability is decided which will vary as per application mentioned in below table.

$$\begin{bmatrix} dr_{pz_{1axi}} & dr_{pz_{2axi}} & dr_{pz_{3axi}} \,..\, dr_{pz_{naxi}} \end{bmatrix}$$
$$\begin{bmatrix} Application1 \\ Application2 \\ Application3 \\ Applicationz \end{bmatrix}\begin{bmatrix} x_{11} & x_{12} & x_{13} \cdots\cdots x_{1n} \\ x_{21} & x_{22} & x_{23} \cdots\cdots x_{2n} \\ x_{31} & x_{32} & x_{33} \cdots\cdots x \\ x_{m1} & x_{m2} & x_{m3} \cdots\cdots x_{mn} \end{bmatrix}$$
Eq.-5

The use of neural network with three level makes the system self-adaptive and decision making which makes the protocol more effective and powerful for management of congestion.

Prospection

Taking details on the previously described topics, the specific research focuses that will be comparatively worthwhile for study can be presented in the paragraphs that follow.
A foundational component of congestion control is network model research, and efforts have been made to establish several nonlinear models, including the model of a dynamic system for discrete occurrences, the Chaos model, and others.

Researches carried out previously mostly overlooked some characteristics or employed local linearization techniques. When any uncertainty variables affect on, it should be a worthwhile problem to look for an acceptable theory to design the system's global dynamic features.

We should build new AQM mechanisms using intelligent decision-making technology. This will also be an issue that requires AQM research to resolve.
The feedback control method for TCP end-to-end is not well-suited to study the self-similar problem. In addition to this the price mechanism that was incorporated into the idea of congestion control is a crucial area for further research.
Most high-speed protocols generally do not compete fairly with other high-speed protocols [31].
The slower convergence times that result from the flows starting at different times compromise intra-protocol fairness [31]. S-TCP and FAST's performance is solely reliant on how they are operating; it is not affected by the competing traffic [31]. Even when it is up against another S-TCP flow, S-TCP is overly aggressive in its pursuit of bandwidth [31].

Criteria used in the comparative analysis Throughput/Goodput, Fairness, Stability, Utilisation of Bandwidth and Queuing Delays.

At the conclusion of this paper, in Appendix A, is a table of comparative analysis.

Conclusion

Starting from conventional approach proposed for congestion control to the method proposed for high speed network as well as real time video network each has its own cons and pros. None of the method proposed is efficient in all parameters.
Thus an efficient method using fuzzy-neuro needs to be implement which may work efficiently on all parameters by making a trade-off between almost all parameters. Hence a multi-level matrix-based dropping and forwarding mechanism has been proposed which works differently on different needs. Instead of dropping packets unanimously a differential equation needs to be calculate based on tabular matrix.
The matrix will calculate the maximum threshold value which will be different for different dscp value which may lead to improve overall performance. The traffic calculated in forthcoming days will be more on real-time video which created more complication in managing the congestion.

References

Journal Papers:

[1] Lisong Xu, Khaled Harfoush, and Injong Rhee "Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks", *IEEE, INFOCOM, 2004*.

[2] Sumitha Bhandarkar, Saurabh Jain and A. L. Narasimha: Reddy "LTCP: A Layered Congestion Control Protocol for Highspeed Networks" *ACM SIGCOMM Computer Communication Review,* Volume 36 , Issue 1, pp.: 41 – 50, 2006. http://portal.acm.org/citation.cfm?id=1111322.1111332

[3] D. Katabi, M. Handley, C. Rohrs. "Congestion Control for High Bandwidth-Delay Product Networks". *ACM SIGCOMM 2002*.

[4] B.Even, Y.Li, D.J.Leith "Evaluating the Performance of TCP Stacks for High-Speed Networks", *Hamilton Institute, Ireland, 2006*. www.hamilton.ie/net/pfldnet2006.pdf

[5] T. Cui and L. Andrew. "FAST TCP simulator module for ns-2", version 1.1, 2004. Available at http://www.cubinlab.ee. *mu.oz.au/ns2fasttcp*.

[6] Junsoo Lee, Stephan Bohacek, Jo˜ao P. Hespanha, Katia Obraczka "A Study of TCP Fairness in High-Speed Networks", *University of Southern California, white paper 2007.h*ttp://whitepapers.silicon.com/0,39024759,60300997p,00.htm.

[7] S. Floyd, S. Ratnasamy, and S. Shenker "Modifying TCP's congestion control for high speeds", May 2002.

[8] C. Jin, D. X. Wei, and S. H. Low, Hegde, S. FAST TCP: Motivation, Architecture, Algorithms, Performance". *IEEE/ACM Transactions on Networking*, Volume 14, Issue 6, pp. :1246 – 1259, Dec. 2006.

[9] C. Jin, D. X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh. "FAST TCP: From theory to experiments". *IEEE Network, 19(1) pp. 4– 11, January/February 2005*.

[11] D. Leigh, R. Shorten, et al. H-TCP ns-2 implementation, 2005.

[12] ns the Network Simulator. http://www.isi.edu/nsnam/ns/.

[14] I. Rhee and L. Xu. "Simulation code and scripts for CUBIC", 2005.*http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/cubic-script/script.htm*.

[15] C. Jin, D. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, H. Newman, F. Paganini, S. Ravot, and S. Singh, "FAST Kernel: Background theory and experimental results," in *First International Workshop on Protocols for Fast Long- Distance Networks*, February 2003..

[16] E. Souza and D. A. Agarwal. "A HighSpeed TCP study: Characteristics and deployment issues". *Technical Report LBNL-53215, 2003.*

[17] K. Tokuda, G. Hasegawa, and M. Murata. "Performance analysis of HighSpeed TCP and its improvements for high throughput and fairness against TCP Reno connections". *In High-Speed Networking Workshop (HSN), 2003.*

[19] S. Floyd. "Highspeed TCP for large congestion window", *Internet Draft draft-floyd-tcp-highspeed-01.txt*, February 2003.

[20] S. Floyd and K. Fall. "Promoting the use of end-to-end congestion control in the Internet". *IEEE/ACM Transactions on Networking, 7(4):458–472*, August 1999.

[21] Eric He, Michael Welzl and Pascale Vicat-Blanc Primet "A Survey of Transport Protocols other than Standard TCP" *Data Transport Research Group, GFD*-I.055, 2005.

[22] http://www.cs.ucla.edu/NRL/hpi/tcpw/tcpw_papers/part1- congestion-control.pdf

[23] Eric He, Rajkumar Kettimuthu, Sanjay Hegde, Michael Welzl, Jason Leigh, Chaoyue Xiong and Pascale Vicat- Blanc Primet, "Survey of Protocols and Mechanisms for Enhanced Transport over Long Fat Pipes", *Data Transport Research Group,2003/2004.http://www.globus.org/alliance/publications/papers/Survey.pdf*

[25] Sumitha Bhandarkar, Saurabh Jain and A. L. Narasimha Reddy, "Improving TCP Performance in High Bandwidth High RTT Links Using Layered Congestion Control", *International Workshop on Protocols for Fast Long-Distance Networks,* February 2022. http://whitepapers.silicon.com/0,39024759,60303395p,00.htm

[26] Sangtae Ha, Long Le, Injong Rhee and Lisong Xu "Impact of background traffic on performance of high-speed TCP variant protocols" *S. Ha et al., Computer Networks 51, pp*. 1748–1762, 2007.

[27] Sangtae Ha, Yusung Kim, Long Le, Injong Rhee, and Lisong Xu* "A step toward realistic evaluation of highspeed TCP protocols" , 2006. *http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/asteppaper. htm.*

[29] D. J. Leith and R. Shorten. "H-TCP Protocol for High- Speed Long Distance Networks". PFLDnet '04, Feb. 2004. *http://dsd.lbl.gov/DIDC/PFLDnet2004/talks/Leith-slides.pdf.*

[30] Pankaj Sharma, "Performance Analysis of High-Speed Transport Control Protocols", Master of Science (Computer Science) Thesis, Clemson University, August 2006. *http://www.cs.odu.edu/~mweigle/papers/sharmathesis06. pdf*

[31] Michele C. Weigle, Pankaj Sharma, and Jesse R. Freeman IV, "Performance of Competing High-Speed TCP Flows" Networking 2006. Networking Technologies, Services, and Protocols; *Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems, pp. 476-487,2022* http://www.cs.odu.edu/~mweigle/papers/networking06.pdf

[32] Yee-Ting Li, Douglas Leith and Robert N. Shorten, "Experimental Evaluation of TCP Protocols for High-Speed Networks", *IEEE/ACM Transactions on Networking,* pp.: 1109-1122,Oct.2007. www.ieeexplore.ieee.org/iel5/90/4346537/04346548.pdf

[33] Sally Floyd, Vern Paxson, "Difficulties in simulating the Internet", *IEEE/ACM Transactions on Networking* 9 (4), 2001.

[34] E. Altman, K. Avrachenkov, C. Barakat, A.A. Kherani, B.J. Prabhu "Analysis of MIMD congestion control algorithm for High Speed Networks" Computer Networks: *The International Journal of Computer and Telecommunications Networking* Volume 48 , Issue 6, pp.: 972 – 989, 2021.

[35] For XCP: http://www.isi.edu/nsnam/ns/doc/node238.html

[38] E. Altman, K. Avrachenkov, C. Barakat, A.A. Kherani, B.J. Prabhu "Analysis of MIMD Congestion Control Algorithm for High Speed Networks", *Elsevier, Computer Networks Volume 48, Issue 6, pp.: 972-989, 2005*. linkinghub.elsevier.com/retrieve/pii/S1389128605000022

[39] K. Satyanarayan Reddy, C. Lokanatha Reddy "A Survey on Congestion Control Mechanisms in High Speed Networks" has been published in the *International Journal of Computer Science and Network Security (IJCSNS) Vol*. 8 No. 1, pp. 187 – 195, January 2008. http://paper.ijcsns.org/07_book/200801/20080126.pdf

[40] K. Satyanarayan Reddy, C. Lokanatha Reddy "An Efficient Explicit Congestion Reduction in High Traffic High Speed Networks through Automated Rate Controlling" abstract appeared in the *International Conference ICSTAORIT – 2006 – XXVI ISPS CONFERENCE Page no. 12*5 with Paper Id : IT-45 held at Tirupati, India during the period 7th January 2007 to 9th January 2007.

[41] Cheng Jin David X. Wei Steven H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance" IEEE Infocom 2004. http://www.ieee-infocom.org/2019/Papers/52_2.PDF

[42] Adrien Bullot, R. Les Cottrell and Richard Hughes-Jones, "Evaluation of Advanced TCP Stacks on Fast Long- Distance Production Networks" *Journal of Grid Computing, Volume 1, Number 4, pp.* 345-359, December 2003. http://www.springerlink.com/content/v7463765348771g6/

[43] Saverio Mascolo, Francesco Vacirca "Issues in Performance Evaluation of New TCP Stacks in High Speed Networks" *white paper on Bandwidth Issues, znet, 2005*. http://www.zdnet.com.au/whitepaper/0,2000063328,224372 28p-16001408q,00.htm

[44] D. M. Lopez-Pacheco, C. Pham, "Robust Transport Protocol for Dynamic High-Speed Networks: *enhancing the XCP approach" ICON, 2022.* http://web.univ-pau.fr/~cpham/Paper/icon05.pdf

[45] R. Les Cottrell, Saad Ansari, Parakram Khandpur, Ruchi Gupta, Richard Hughes-Jones, Michael Chen, Larry McIntosh, Frank Leers, "Characterization and Evaluation of TCP and UDP-based Transport on Real Networks", Presented at the *Third International Workshop on Protocols for Fast Long-Network.*

Proceedings Papers:

[10]. T. Kelly. "Scalable TCP: Improving performance in highspeed wide area networks". *In Proceedings of PFLDnet, Geneva, Switzerland, Feb. 2003.*

[13] I. Rhee and L. Xu. "CUBIC: A new TCP-friendly highspeed TCP variant". *In Proceedings of PFLDnet, Lyon, France, Feb. 2005.*

[18] L. Xu, K. Harfoush, and I. Rhee. "Binary increase congestion control for fast, long distance networks". *In Proceedings of IEEE INFOCOM, Hong Kong*, Mar. 2004

[24] R. El Khoury and E. Altman "Analysis of Scalable TCP" In *Proceedings of the International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks* (HET-NETS), July 2004. http://citeseer.ist.psu.edu/719259.html

[28] E. Altman, K. E. Avrachenkov, B. J. Prabhu "Fairness in MIMD Congestion Control Algorithms", *In Proceedings of the IEEE INFOCOM, 2005*. http://www.inria.fr/rrrt/rr5312. html

[36] D. M. Lopez-Pacheco and C. Pham "Robust Transport Protocol for Dynamic High-Speed Networks: enhancing the XCP approach" in *Proc. of IEEE MICCICON, 2005*.
[37] Yunhong Gu, Xinwei Hong, and Robert L. Grossman, "Experiences in Design and Implementation of a High Performance Transport Protocol ", *Supercomputing, Proceedings of the ACM/IEEE SC2004 Conference, pp. 22, 2004.*

Sunil Kumar Kushwaha is pursuing Ph.D. degree in Computer Science and Engineering from Medi-Caps University, Madhya Pradesh-Indore. He did his M.Tech in 2003 from Devi Ahilya Vishwavidyalaya-Indore. Mr.Sunil Kushwaha has worked on SWAM , Network Security and IT act for Govt. of India. He had also worked on Foreign Trade Policy , SEZ and various other projects for Govt. Of India. He had published various research papers in International Journals of reputed publications.
Presently he is serving as a Sr. Assistant Professor at Medi-Caps University, Indore. He is a reviewer of various international and National journals and Conferences. He had global certification of CISCO during 2009-2013 and presently as a CISCO certified Instructor.

Dr. Suresh Jain is a Professor of Computer Science & Dean Academics at Acropolis Institute of technology & Research to Indore. He has experience over 34 years in the field of academics, research & administration. He is an engineering graduate of MANIT Bhopal, alumina of SGSITS Indore and Ex-Professor IET-DAVV.
Dr. Jain has published more than 40 research papers in reputed journals and guided 12 PhD in the area of NLP, Machine learning, Fuzzy systems and Computer Networks

APPENDIX-A

S.No.	Algorithms	Approach	Behaviour	Best-Parameters	Calculative Parameters
1	SQRT [19]	Window-based	Sub-linear	packet drop minimum	a=l.O, p=0.67, k=0.5, 1=0.5
2	SIMD [20]	Window-based	Superlinear	low latency	p=0.0625, k=-0.5, l=l
3	CUBIC[21]	Window-based	Linear	low latency	Wcubic=C(t-K)3+Wmax, K= 3 Wmaxp / C
4	LDA+ [22]	Rate-based	Linear	packet drop minimum	Bandwidth=avearge packets/response time
5	TFRC [23]	Rate-based	Linear	packet drop minimum	X = s / (R * f(p)) for f(p) = sqrt(2*p/3) + (12*sqrt(3*p/8) * p * (1+32*]i'2)).
6	TFCBR [24]	Rate-based	Sublinear	tluoughput constant under all load	Sampling rate is 8 or less
7	Reno	Instanataneous queue	AIMD	Minimum paket loss	ACK:w+=l/w Congestion: w-=0.5w
8	TCPW	Instanataneous queue	MIMD	Bandwidth utilization	ACK:w+=llw Congestion: wi=RiRTT
9	HSTCP	Instanataneous queue	linear with exponential	Packet loss minimum	ACK: w+=a(w)/w Congestion: wb(w)w
10	Fast TCP	Window-based	proportional linear	Minimum Queuing delay	RTT: w+=wbase RTT/RTT+a
11	XCP	Window-based with average	linear	Fairness	ACK: cwnd=rnax(cwnd+ H_feedback, s)
12	BIC [1], [18]	Flow based	convergence increasingly	Responsiveness	(cwnd < wmax) // fast convergence wmax = cwnd * (2-P) / 2; else wmax = cwnd; cwnd = cwnd * (1-P);
13	CUBIC [13], [14]	Bursty traffic	slow convergence	Good stability	
14	FAST [5],[8],[9], [15], [41]	Window-based with fast forward	strange convergence	Good Throughput	CWND by MSS * MSS / CWND
15	HS TCP [16], [17]Jl9], [21], [27], [30],[31]	greater window size with good fairness	converge linearlily	Bandwidth utilization	average throughput of a connection= (.75*W*MSS)/RTT.
16	LTCP [2], [25], [27]	variable flow size	Promising convergence	Fairness Index	Not available
17	STCP [10], [34], [36], [37], [38]	low value of window size with low flow	slow convegence	Throughput with fuimess	cwnd = sstluesh; Timeout: /* Multiplicative decrease */
18	XCP [3], [24], [35]	high bandwidth, large window size based	fast responsiveness with linear drop	Throughput with fuimess	Not available