Study and Analysis of Different TCP Variants

Amol P. Pande
Datta Meghe College of Engineering
University of Mumbai
Mumbai, India
amolpande69@gmail.com

Dr. S. R. Devane
Datta Meghe College of Engineering
University of Mumbai
Mumbai, India
srdevane@yahoo.com

***Abstract*—TCP is a Transport Layer Protocol, part of the TCP/IP suite to maintain the flow, congestion, and performance of currently increasing internet traffic. Over this period of time, many TCP variants have been developed by researchers to cater the need of changing network media and dynamic topology for improving overall end to end QoS performance parameters such as throughput, jitter, delay, and packet loss for both wired and wireless networks. In this review paper, some variants are described in categories, such as Loss-Based, Delay-Based and Hybrid using the algorithms and network media. It has been observed that these variants are not developed taking in to consideration of the end applications which has a particular need and expectation from the TCP and hence need to be analyzed for same. This paper highlights on conventional and recent TCP variants and its comparative study which could be useful to many researchers to reduce the lacuna of different variants.**

Keywords—Transport Layer Protocol, TCP variants, Data Traffic, Congestion Control Mechanism.

I. INTRODUCTION

Transmission Control Protocol (TCP), developed by the Bob Kahn and Vint Cerf in 1974 to give reliable connection oriented services to the application with flow and congestion control for the wired network. Due to increase on the internet based advanced applications like E-Commerce, video conferencing, online games, streaming media and today's need of Artificial Intelligence (AI) based application requires well support by the underlying TCP with varying media i.e wifi, WiMAX, GSM CDMA etc.

The TCP is originally designed for wired network and remains the same with its variants that cope up with the enhancement of the communication network. This is the issue with the rapid developing wireless network. The media of wireless network has different characteristics. The packet transmitted through a wireless link may be lost due to wireless error like high bit error rate, mobility, frequent route failure etc. The TCP host may interpret the packet lost due to network congestion, and then take the action of reducing the size of the transmission window or waiting for the long idle time for retransmitting the lost packet. The result is degraded end-to-end performance. The reasons related to packet losses vary from network to network. Network congestion is one of the major problems because of the insufficient capacity of the underlying sub-network with increasing demand of data transfer. For wired or wireless communications, congestion prevention is the major challenge for TCP protocols. The continuous enhancements in communication network media and applications further lead to severe challenges to design efficient TCP variants. The TCP variants differs from each other as they are specially designed by considering the particular network conditions and scenarios but not considering the requirement of the application. For traditional TCP, it is difficult to understand the reason for packet loss. Therefore to cope up with this traditional TCP, various wired and wireless TCP variants are discovered. These TCP variants are categorized based on their features, network environment, congestion detection (loss or/and delay), congestion avoidance method and association between them.

In this paper, detailed comparative analysis on various TCP variants and their study based on different parameters is presented. Section II gives an overview of TCP functionality whereas section III describes Review of Literature. Section IV focuses on classification and working of various TCP variants. Section V describes a comparative analysis of TCP variants and section VI concludes the study.

II. OVERVIEW OF TCP

The Transmission Control Protocol (TCP) is widely used delivery data packet and data transfer mechanism. The TCP mechanism is specifically designed to provide a reliable transfer mechanism.

A. TCP and its Operations

TCP works with its four looped components that are Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery. Slow start balances the speed of a network connection. Slow start slowly increases the amount of data transmitted till it finds the network's maximum carrying capacity. A congestion avoidance phase includes a slow start phase and TCP inherits the congestion window value (cwnd) from the slow start phase. The Fast retransmit in TCP reduces the time and a sender waits before retransmitting a lost segment and allows it to quickly recover lost data packets. Fast recovery congestion control algorithm makes quicker lost data packet recovery.

III. REVIEW OF LITERATURE

Surveys carried out by the researchers bring vast development of TCP variants. These surveys are conducted considering various parameters like network domain, algorithms used, congestion detection etc. The first classification can be seen by the performance of the variant and categorized as a loss based, delay based or hybrid based. Several papers give the comparative analysis of the TCP variants based on their performance. L. A. Grieco et al. [1], H. Jamal et al. [2] compared a standard TCP Reno to various variants and also classified the variants into loss-based, delay-based, and mixed loss-delay based. B. Sikdar, et. Al [3] focuses on new algorithms for Tahoe, Reno, and SACK that improved TCP in the existence of multiple-losses from the same window. The study by W. Feng et. Al [4] shows that even though Reno is popularly used variant, Vegas is superior to TCP Reno. Secondly, the performance of TCP variants also varies with the network environment. So variants are also categories as wired and wireless. G. Morabitoet. al. [5], A. A. Hanbali et al. [6] have discussed on MANET focusing on congestion control and other wireless

978-1-5386-5257-2/18/$31.00 ©2018 IEEE

issues of ad hoc network. Similarly, H. Balakrishnan et al. [7] also compared protocol categories of end-to-end, link-layer, and split connection of TCP congestion control over wireless networks. K. S. Reddy et al. [8] studied and compared specifically high-speed network TCP variants such as BIC, CUBIC, FAST TCP, HSTCP, Layered TCP, and STCP. Siddharth Trivedi et. al. [9] described how TCP variants like Hybla have been proposed for optimized channel throughput in heterogeneous networks that incorporate satellite links. Some TCP variants are developed for a specific feature which is also compared and analyzed in various papers that shows their performance efficiency. Jiyan Sun et. al. [10] described TCP-FNC (TCP with fast network coding), that is designed with network coding to reduce the decoding delay. It describes that TCP-FNC can have shorter decoding delay than TCP/NC and VON with no swap in its good put performance. Jun ZHANGet. al. (2017) [11] propose a novel on TCP algorithm named TCP-ACC to handle challenges that integrates firstly real-time reorder metric for calculating the probabilities of unnecessary Fast Retransmit (FRetran) and Timeouts (TO) secondly, an improved RTT estimation algorithm giving more importance to packets that are sent (as opposed to received) more recently, and thirdly improved congestion control mechanism based on packet loss and reorder rate measurements.

IV. CLASSIFICATION OF TCP VARIANTS

On the basis of reviewed survey papers, the TCP variants are broadly classified as Loss based, Delay based and Hybrid as shown in TABLE I.

TABLE I. CLASSIFICATION OF TCP VARIANTS ON CONGESTION DETECTION METHOD

Category	TCP Variants
Loss Based & Loss + Estimation	TCP-Tahoe, TCP-Reno, TCP- New Reno, TCP-SACK, TCP-FACK, , BIC-TCP, TCP Cubic , TCP Hybla TCP, HS-TCP, H-TCP, S-TCP, LP, TCP Libra , TCP Westwood , TCPW CRB, TWPW BR, DOOR,DSACK, TD FR, TCP -FIT
Delay Based	TCP-Vegas , TCP Vegas-A, TCP New Vegas, FAST-TCP, Nice, TCP Real
Hybrid(Loss & Delay) Based & L+D+ Estimation	Compound TCP, , TCP Jersey , TCP Africa, TCP Veno , TCP Illinois', YeAH TCP, TCP Fusion, , TCP-FNC,TCP-ACC

A. *Loss Based :*

Loss based TCP variants uses packet loss as an indication of Congestion.

1. TCP-Tahoe: - TCP Tahoe [12] is the TCP variant developed in 1998 by Jacobson. It uses AIMD (Additive Increase Multiplicative Decrease) policy to adjust window size. The CWND is added by one for successful packet delivery and decrease window to half if data loss or delay occurred only when first NACK is received. In case of timeout event, it reduces congestion window to 1 MSS. It uses packet loss probability to adjust the congestion window size.
2. TCP-Reno: - TCP Reno [13] different from TCP Tahoe at congestion avoidance. When three duplicate ACKs are received, it will halve the congestion window, perform a fast retransmit, and enters fast recovery. If a timeout event occurs, Reno will enter slow-start, same as TCP Tahoe. TCP Reno is efficiently recover from a single packet loss, but the disadvantage is that it suffers from performance problems when multiple packets are dropped from a window of data.
3. TCP-New Reno: - TCP New Reno[14] tries to improve the TCP Reno's performance when a burst of packets are lost by modifying the fast recovery algorithm. In TCP New Reno, a new data ACK is not only to take TCP out of fast recovery to congestion avoidance, but it requires all the packets are acknowledged, which are outstanding at the start of the fast recovery period. TCP New Reno retransmit the packet by assuming that the packet that immediately follows the partial ACK received at fast recovery is lost.
4. TCP-SACK: - TCP-SACK [15] utilizes wired network domain and it allows receivers to stop repeated data packets which is received correctly. The TCP-SACK is defined in the RFC 2018. In this variant the acknowledgements specifies the number of SACK. SACK holds the starting and ending number range of packets which are received correctly.
5. TCP-FACK: - TCP-SACK [16] with Forward acknowledgement identified as TCP FACK and it also utilizes wired network domain. The TCP-FACK is little bit advanced as compare to the TCP-SACK. It calculates the better transmission of packets and utilizes HS/LD.
6. BIC-TCP: - TCP-BIC [17] is designed to achieve the RTT-fairness and utilizes HS/LD. It extends New Reno with its additional functionality. It calculates the maximum link capacity and maximizes the congestion window. The TCP-BIC reduces the calculation of big link capacity. This phase rapidly discovers, in a binary search manner, the optimal congestion window size by relying on detection of a packet loss as an indication of congestion window overshooting.
7. Cubic TCP: - Cubic [18] is an enhanced version of BIC. TCP-friendliness and RTT-fairness is improved and BIC window control is simplified. Cubic is a less aggressive as compare to other variants and utilizes HS/LD. The TCP-CUBIC is used in Linux operating systems. The window growth function of Cubic is governed by a cubic function in terms of the elapsed time since the last loss event and provides a good stability and scalability.
8. TCP Hybla: - TCP Hybla [19] uses the high latency from the satellite to remove penalization of TCP network and utilizes HS/LD. The purpose of the TCP Hybla is to get the same transmission rate of packets and comparatively fast reference TCP connection (e.g. wired) for long RTT connections (e.g., sat-com and wireless).
9. HS-TCP: - Basically High Speed TCP [20] describes the High Loss Rate and Low Loss Rate networks. When congestion window is less than threshold it behaves like a standard Reno. So its transmission rate is switched mode and thus has two rules to control transmission rate.
10. H-TCP: - TCP-Hamilton [21] utilizes HS/LD and improves the design RTT-fair by using RTT-scaling. The H-TCP increases the congestion window size base on time in between the successive congestion events and

the ratio of the minimum observed RTT to the maximum observed RTT.

11. S-TCP: - The Scalable-TCP [22] is designed to improve in deployment and uses identical traditional TCP variants stacks. The Scalable-TCP (STCP) are High Speed TCP originally designed for the High Speed backbone links that uses multiple clients. Instead of core AIMD concept it introduces a multiplicative increase multiplicative decrease (MIMD) theory.

12. LP: - TCP-LP [23] is TCP Low Priority variant. It also aims low priority data transfer for background applications. It is designed to achieve the goal of utilize network information to calculate the "fair share". TCP-LP has the key mechanism for utilization of unidirectional packet delays.

13. TCP Libra: - Marfia et al. [24] proposed Libra as another variant of congestion control to resolve the scalability issues in standard TCP, while preserving and improving the RTT- fairness properties. Libra's design is based on New Reno and modifies the Congestion Avoidance congestion window increase steps to follow a specially designed function of both the RTT and the bottleneck link capacity.

14. TCP Westwood: - The TCP Westwood [25] proposes an end-to-end bandwidth estimation algorithm totally based on TCP Reno. It produces slow begin and congestion avoidance stages like TCP Reno, however instead of halving the congestion window size as in TCP Reno when congestion happens, TCP Westwood adaptively estimates the exits bandwidth and sets the congestion window size and slow start threshold for that reason to enhance the link usage.

15. TCPW CRB: - Wang et al. [26] acknowledged the critical vulnerability of Westwood: under certain network conditions the bandwidth estimation (BE) technique gives highly inaccurate results. As a solution to this problem they proposed TCPW CRB (Westwood with Combined Rate and Bandwidth estimation), which refines the estimation algorithm by complementing it with a conservative long-term bandwidth calculation ("rate estimation" RE) technique. It is similar to one from the Westwood+ proposal, but the sampling period is some predefined constant T , instead of a measured RTT.

16. TCPW BR: - TCPW BR [27] Westwood with Bulk Repeat algorithm is also based on Westwood, but additionally integrates a special loss-type detection mechanism. Upon each loss detection, if it is estimated to be non-congestion related, BR applies very aggressive (highly optimistic) recovery policies instead of the original ones. The queuing delay estimation threshold and rate gap threshold algorithms are the two loss detection algorithms.

17. TCP DOOR: - TCP DOOR [28] improves TCP performance and avoid unnecessary congestion control, by detecting and responding to out-of-order (OOO) packet delivery events. when a packet sent earlier arrives later than a subsequent packet then OOO occurs. Because of route changes occur in ad hoc networks, OOO may happen multiple times in one TCP session. OOO detection is carried out at both ends: the receiver detects the OOO data packets and the sender detects the OOO ACK packets.

18. DSACK: - DSACK [29] is Duplicate Selective Acknowledgements that specifies complements the standard and provides a backward-compatible way to report such duplicates. The Requirement of DSACK is that the receiver to report each receipt of a duplicate packet to the sender. However, there are two possibilities of duplication, which should be treated in slightly different ways. First, the duplicated data can be some part of the acknowledged continuous data stream. Second, it can be a part of some isolated block. DSACK provides a way to report packet duplication without breaking the SACK standard. Similar to SACK, the DSACK specification does not specify any particular actions for the sender.

19. TD FR: - TD FR [30] Paxson [33] proposed a simple way to eliminate the penalties of reordering through TD-FR, time delayed Fast Recovery. If a receiver does not respond immediately to out-of-order data packets with duplicate ACKs, but postpones the action, a majority of the reordering events will be hidden from the sender. However, the advantage of this solution is, at the same time, a disadvantage as well. The artificial delay, aimed at preventing overreaction, adds to the time required to detect actual losses. The "fast" loss detection mechanism becomes slower than a conventional loss detection based on RTO If the delay grows big.

20. TCP-FIT: - TCP-FIT [31] is a new congestion control algorithm, which could perform gracefully in both wireless and high BDP networks. The algorithm TCP-FIT is motivated by parallel TCP, but with the important distinction that is for each TCP session, only one TCP connection with one congestion window is established, and that no changes to other layers of the end-to-end system need to be made.

B. Delay Based:

Delay based TCP variants considers packet delay rather than packet loss as congestion in the network.

21. TCP Vegas: - TCP-Vegas [32] is designed to avoid the congestion from the TCP Networks which is affected on the packet loss and packet delay. The TCP-VEGAS uses RTT (Round-Trip Time) such as TCP-Reno and TCP-New Reno. With the proactive approach, it aims to estimate buffer size at the bottleneck of the router. Vegas tries to quantify an absolute number of packets queued at the bottleneck router as a function of the expected and actual transmission rate.

22. TCP Vegas-A: - Srithith et al. [33] have presented the Vegas A (Vegas with Adaptation) algorithm, which extends the original Vegas congestion control with an adaptable mechanism. The threshold coefficients α and β from the Vegas algorithm are adjusted depending on the steady state dynamics of the actual transmission rate. Vegas A adds additional conditions to the congestion window management algorithm. An increase is allowed in three cases: (1) if the estimate shows no congestion and a lower threshold has a minimal value ; (2) if the actual rate has increased and the estimate is showing no congestion (3) the actual rate has decreased while the flow is in a steady state.

23. TCP New Vegas: - TCP-New Vegas [34]. To reduce the convergence time and improve to some degree the high-BDP link utilization, the proposed New Vegas algorithm defines a new phase called Rapid Window Convergence. The key idea of this phase is not to immediately terminate the Slow Start phase when the estimate of network buffering exceeds the threshold , but to continue the opportunistic exponential-like resource probing with reduced intensity.

24. FAST-TCP: - FAST-TCP [35] is used to avoid the congestion in TCP network. The FAST-TCP is mostly used in the long-distance and high latency links. It is designed in Net Lab, California Institute of technology but it is commercial. The FAST-TCP supports existing algorithms too. It defines a periodic congestion window update based on the internal delay-based estimate of the network state.

25. TCP Nice:- TCP Nice [36] is a system intended to provide the abstraction of free networks bandwidth for background data packets. Its primary goals are to eliminate interference of regular demand traffic and reap a significant fraction of the spare network bandwidth available.

26. TCP Real:- TCP-Real [37] is receiver oriented approach to congestion control and demonstrated by an experimental protocol. Due to receiver oriented nature and wave communication pattern allow amending mechanisms like congestion avoidance which reduces unnecessary transmission gaps that hurt the performance of time constrained applications and advanced error detection and classification.

C. Hybrid (Loss-Delay Based):

Based In hybrid type of TCP variant both packet loss and delay of packet is considered as a congestion in network.

27. TCP Compound: -The design of Compound TCP [38] is to fulfill the efficiency requirement and TCP friendliness requirement concurrently. The important concept is that if link is under-applied, the high-speed protocol should be competitive and expands the sending price fast. The factor that is integrated in Compound TCP is astandard TCP congestion avoidance algorithm.

28. TCP-Jersey: - TCP-Jersey [39] is other scheme that adapts the transferring price proactively consistent with the community situation. The bandwidth estimation algorithms and the congestion warning (CW) router configuration are the two factors included in TCP Jersey. The routers alert end stations via marking all packets whether there is a signal of incipient congestion by using congestion warning which is a configuration of network routers.

29. TCP Africa: The TCP-Africa [40] variant uses an aggressive and scalable window increase rule to allow quick utilization of available bandwidth. But it uses packet round trip time measurements technique to predict eminent congestion events.

30. TCP Veno: TCP Veno [41] is congestion control module to improve the TCP performance over wireless networks. TCP Veno is intelligent enough to differentiate them. The connection can stay for longer in the operating region by avoiding base RTT to change from time to time.

31. TCP Illinois: TCP Illinois [42] This sender-side-modified protocol was introduced in 2008 Liu et al. at UIUC. Standard TCP's AIMD algorithm is modified in this protocol. Delay and loss are used as congestion signals to decrement or increment cwnd. Its performance is high than the standard TCP and network bandwidth is shared fairly. If congestion is not detected, TCP-illinois appraises its cwnd on each ACK arrival in one RTT by (α/cwnd). Otherwise, cwnd is decreased by (β cwnd).

32. YeAH TCP : YeAH TCP [43] is the abbreviation of "Yet Another High-speed" TCP. It was proposed in 2007 by Baiocchi et al. YeAH is like CTCP and TCP Africa. In this protocol, network delay is predicted by RTT estimation and loss detection. STCP and NewReno are combined in YeAH. In every RTT, congestion window is increased by one and if loss is detected, congestion window is decreased by half when 3 duplicated ACKs are received. YeAH have two thresholds (α, φ) whose values are predefined. If (Q/RTTmin< φ) and (Δ<α), YeAH behaves like CTCP by switching to fast mode. On other hand, YeAH is switched to New Reno slow mode. Intra-fairness, inter-fairness, RTT-fairness and efficiency of YeAH is higher than other protocols in higher BDP-networks. It also has disadvantage of RTT transmission due to Vegas algorithm.

33. TCP Fusion: TCP Fusion [44] is congestion TCP Fusion was proposed in 2007 by Kaneko et al. It has combined DUALs queuing delay, Westwood's achievable rate and network buffering estimation using Vegas algorithm. In this protocol, fusion can be switched into three modes. This switching depends on queuing delay's threshold value. Fast mode is applied when predefined threshold value is greater than queuing delay and congestion window value is increased by Westwood's estimation fraction. Congestion window is decreased by no. of packets, if threshold value is three times less than current queuing delay. Congestion window is neither increased nor decreased if the value of queuing delay is between 1 and 3 times of the threshold. Its fairness and bandwidth utilization is greater as compared to other protocols. It has some disadvantages like threshold value is calculated manually.

34. TCP-FNC: - The TCP-FNC [45] is designed to reduce decoding time delay in the TCP with network coding. The TCP-FNC continues the framework of TCP/NC and combines two schemes to obtain the demand of online network coding.

35. TCP-ACC: - Theoretical analysis describes that the equilibrium throughput of TCP-ACC [46] are higher than the traditional TCP. It is maintaining the good fairness with regard to some other TCP algorithms in ideal network conditions"

TABLE II. COMPARISON OF TCP VARIANTS ON DIFFERENT PARAMETERS

TCP Variants & Year	**Base variant**	**Mod. @ [1]**	**CD [2]**	**N/w Env. [3]**	**Added/Changed Modes or Features**	**Congestion Avoidance Method *cwnd*$_{new}$**
TCP Tahoe 1988	RFC 793	Sender	Loss	Wired	Slow Start, Congestion Avoidance, Fast Retransmit	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP Reno 1990	Tahoe	Sender	Loss	Wired	Slow Start plus Congestion Avoidance plus Fast Retransmit and Fast Recovery	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP New Reno 1999	Reno	Sender	Loss	Wired	Fast Recovery resistant to multiple losses	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP SACK 1996	Reno	Sender& Receiver	Loss	Wired	Extended information in feedback messages	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP FACK 1996	Reno, SACK	Sender	Loss	Wired	SACK-based loss recovery algorithm	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP-Vegas 1995	Reno	Sender	Delay	Wired	Utilizes Bottleneck buffer as a primary feedback for the Congestion Avoidance and secondary for the Slow Start.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
TCP Vegas-A 2005	Vegas	Sender	Delay	wired	Adaptive bottleneck buffer state aware Congestion Avoidance	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
TCP Veno 2002	Vegas, New Reno	Sender	LDBE	Wired	Bottleneck buffer state estimation for Reno-type Congestion Avoidance and Fast Recovery increase/decrease coefficient adaptation.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
Nice 2002	Vegas	Sender	Delay	LPDT	Delay threshold is used as a secondary congestion indicator.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
LP 2002	New Reno	Sender	Loss	LPDT	Early congestion detection	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
HS-TCP 2003	New Reno	Sender	Loss	HS/LD	AIMD factors as functions of the congestion window size, Limited Slow-Start.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
STCP 2003	New Reno	Sender	Loss	HS/LD	MIMD CA policy, Multiplicative Increase Multiplicative Decrease congestion avoidance policy	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Scaling
H-TCP 2004	New Reno	Sender	Loss	HS/LD	CWND raise as a function of time elapsed since the last packet loss detection, multiplicative decrease coefficient adaptation, scaling increase step to a reference RTT.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP Hybla 2004	New Reno	Sender	Loss	HS/LD	Range the increase steps in Slow-Start and Congestion Avoidance to the reference RTT, data packet pacing, initial slow-start threshold estimation,	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
BIC TCP 2004	HS-CP	Sender	Loss	HS/LD	Binary congestion window search, Limited Slow-Start	Equation Based *cwnd*
TCP Cubic 2008	BIC	Sender	Loss	HS/LD	The congestion window control as a cubic function of time elapsed since a last congestion event	Equation Based *cwnd*
FAST TCP 2003	Vegas	Sender	Delay	HS/LD	Constant-rate congestion window equation-based update, Periodic fixed rate *cwnd* update	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
TCP Libra 2005	New Reno	Sender	Loss	HS/LD	Packet pair technique, scale the congestion window increase step by the bottleneck link capacity and queuing delay	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP New Vegas 2005	Vegas	Sender	Delay	HS/LD	Rapid window convergence, packet pacing, packet pairing Rapid	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
TCP Fusion 2007	Vegas, West Wood	Sender	LDBE	HS/LD	CWND increase steps as a function queuing delay estimates and of the achievable rate.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Constant
TCP Africa 2005	Vegas, HS-TCP	Sender	LD	HS/LD	Exchange between fast of HS-TCP and slow start of NewReno mode depending on the Vegas-type network state estimation	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
Compound TCP 2005	Vegas, HS-TCP	Sender	LD	HS/LD	Two components (slow and scalable) in the congestion window calculation, Slow and scalable *cwnd* calculation,	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Scaling
TCP Illinois 2006	New Reno	Sender	LD	HS/LD	AIMD of queuing delay, Additive increase steps and multiplicative decrease factors as functions of the Q-Delay	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Other
YeAH TCP 2007	STCP, Vegas	Sender	LD	HS/LD	Depending on Vegas and Dual type estimation fast (STCP) and slow (NewReno) mode can be switched.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient
TCP Westwood 2001	New Reno	Sender	LBE	Wireless	Estimate available bandwidth (ACK granularity), Faster Recovery	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Zero MI $cwnd_{new} = \alpha x \beta$ α= BE, β= RTTmin
TCPW CRB 2002	West Wood	Sender	LBE	Wireless	Available bandwidth estimate (combination of ACK and long-term granularity), predominant cause of packet loss identification.	MI $cwnd_{new} = \alpha x \beta$ α= BE, β= RTTmin
TCPW BR 2003	West Wood	Sender	LBE	wireless	Loss type estimation technique (queuing delay estimation threshold, rate gap threshold), retransmission of all outstanding data packets, limiting retransmission timer back off.	MI $cwnd_{new} = \alpha x \beta$ α= BE, β= RTTmin
DOOR 2002	New Reno	Sender & Receiver	Loss	Wireless	Out-of-order detection and feedback, disables congestion control temporarily. Recovery is instant.	AI $cwnd_{new} = \alpha+\beta$ $\alpha = cwnd_{old}$, β= Quotient

DSACK 2000	SACK	Receiver	Loss	Wireless	Reports duplicate segments	AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Quotient
TD FR 1997	Reno	Receiver	Loss	Wireless	Time delayed fast recovery	AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Quotient
TCP Real	Reno	Receiver	Delay	Wireless	congestion avoidance reduces unnecessary transmission gaps affects time constraints applications and advanced error detection and classification gives recovery tactics improves performance of wireless link.	AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Zero and AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Quotient
TCP Jersey PPT	New Reno	Sender	LD	Wireless	Bandwidth estimation to ease the effect of wireless errors. Packet loss due to wireless error from congestion warning is provided.	AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Zero MI $cwnd_{new} = αxβ$ α= BE,β= RTTmin
TCP-FIT Paper	--	Sender	Loss	Wireless	AIMD mechanism is employed to adjust the congestion control window. It combines linear growth of the congestion control window with an exponential reduction when a packet loss event occurs.	AI $cwnd_{new} = α+β$ $α= cwnd_{old}$,β= Constant
TCP FNC 2015	Vegas	Sender & Receiver	LD	Wireless	Feedback based method, FCWL (Feedback based Coding Window Lock) is used to reduce the waiting time. Optimized decoding algorithm used EFU eliminated Gaussian Jordan to minimize the computational time.	Equation-based
TCP ACC 2017	Reno	Sender	LD	Wireless	The algorithm detects packet reordering level and packet losses by combining a packet reordering measurement and congestion control to avoid unnecessary slow down of the data transmission rate while preventing congestion to maintain good fairness.	Equation-based

[1] TCP specification modification, [2] Congestion Detection Scope, [3] Network Environment, * HS/LD- High Speed or Long Delay , = * LPDT- Low Priority Data Transfer * S- Sender side * R- Receiver side * SR – Sender and Receiver side , * LD- Loss and Delay * LBE- Loss Based Estimation * LDBE – Loss Delay Bandwidth Estimation

V. COMPARATIVE ANALYSIS

The TCP variants in this section are compared on their connection with each other and other parameters such as features, network environment, congestion detection methods etc.. Summary of TCP variants is showed in TABLE II. The first simplest TCP variant to control congestion was Tahoe. However, when packet loss is detected three algorithm of Tahoe reduces cwnd to 1 which causes notable throughput degradation. Reno has same congestion control algorithms as Tahoe with an addition of fast recovery. In fast recovery, ssthresh and new cwnd is set to half of the current cwnd instead of setting the cwnd to 1.

New Reno, SACK, Vegas, TCP-Real, and TCP-FIT are evolved from TCP-Reno which is specifically targeted for wired and wireless environments. The New Reno and SACK solved multiple losses problem by using partial ACK and selective ACK respectively in the wired environment. The Vegas basically calculates a base RTT and compares it with RTT of a packet with recently received ACK. If compared RTT is much smaller than base RTT it increases its sending window and if RTT is greater than the base RTT, it decreases its sending window. The TCP-Real implements contention detection and congestion avoidance in the wireless environment. The last one, TCP- FIT performs gracefully in both wireless and high speed/long delay network by parallel TCP technique.

A few TCP variants like New Vegas, FAST, Vegas-A, Nice are extended from Vegas and are sender centric, used in High speed/Long delay environment and its scope to detect congestion is delay based. For example, New Vegas increase network utilization of the high-speed/long-delay environment and reduce convergence time by including rapid window convergence algorithm. FAST TCP defines a periodic fixed rate cwnd and delay-based congestion estimation. Vegas, under certain condition inappropriately decrease flow rate to nearly zero, to overcome this problem Vegas-A proposed an additional adaptive buffer mechanism. In Vegas A, according to the actual transmission rate, the threshold coefficient is adaptively tuned according to the actual transmission rate and it also adopts cwnd management algorithm. Nice focuses on low priority data transfer domain. The proactive method allows the Nice to consume the network resource for the low priority data flows when the high priority data flows are not used.

New Reno, member of the Reno family has been further extended to a large number of new TCP variants that support all kinds of network environments. HS-TCP, Hybla, Libra, STCP, Illinois are the enhanced New Reno versions. While BIC is developed from HS-TCP which is further evolved into CUBIC. Other variants such as Fusion, CTCP and Africa are implemented by combining features of two or more TCP variants. Above mentioned variants use sender centric approach, works in High speed/Long delay environment. The scope of these variants to detect congestion is Loss or Loss-Delay based. The enhanced features included in above variants are AIMD of cwnd and queuing delay, semi-independent of RTT, packet pair technique, MIMD congestion avoidance policy, binary cwnd search, and so on.

Apart from an extension of New Reno to high-speed/long- delay environments, it also expands for wireless environment, for example, Westwood, DOOR, and TD-FR all are sender centric, except DOOR which uses both Sender & Receiver centric approach. In Westwood, in order to speed up fast recovery stage bandwidth estimation algorithm is added. This bandwidth estimation algorithm has an inherent concept of Vegas, where to calculate the data transfer rate, the RTT and the number of transmitted data packets is used. Parameters of cwnd and slow start threshold, sssthresh in Westwood are then set near to the estimated data transfer rate when the packet loss is detected. Furthermore, TCP variants, i.e., TCPW BR, TCPW CRB are evolved from the Westwood. In the Westwood's congestion control algorithm, features of the predominant packet loss identification and the loss based estimation are added in TCPW BR and TCPW CRB respectively. DOOR includes Out-of-order detection and feedback, temporary congestion control disabling and instant recovery and TD-FR uses time delayed fast recovery algorithm to improve the packet reordering problem. Combining the concepts and ideas from different TCP variants together a number of TCP variants are developed. For instance, Compound TCP is constructed by implementing the network stage estimation of Vegas and adding slow and scalable cwnd calculation of HS-TCP. Whereas, Africa is built by implementing the fast and slow mode switching of HS-TCP and the network stage estimation of Vegas. Fusion is another integration of TCP variants from the Vegas, New Reno, and the Westwood. In the Fusion, the features of network buffer estimation from Vegas and achievable rate from Westwood are executed. Fusion also maintains New Reno's congestion control mechanism.

Other TCP variants like DSACK, which is the extension of SACK for the wireless environment and which uses receiver centric approach and its scope to detect congestion delay based. In order to avoid the problem of misinterpret out of order delivery data packet, DSACK requires the receiver to acknowledge each receipt of a duplicate packet to the sender.

All these TCP variants consist of two modes in their Congestion avoidance algorithms and operate depending on the some constraints and conditions. The first component and second component of the function, f(cwnd) is defined as α and β to represent the increment policies of additive and multiplicative, respectively. α in the additive increase which totally depends on the old cwnd size. And β in the additive increase depends on whether it is zero, constant, scaling, quotient or other.

In this paper around 35 TCP variants are studied and compared which is mention in Table 2. From this table, it is observed that nearly 83% of TCP variants are modified at sender side and 17% are modified at receiver and both sides. Also we can observe that, around 43% of TCP variants are for high-speed/long-delay environments, about 29% is for wireless environments and the rest of percentage is for wired and low priority data transfer environments. Most of the TCP variants of high-speed or long-delay environment evolved from the TCP variants of the wired environments.

VI. CONCLUSION AND FUTURE WORK

The amount of data demanded for transfer might be more than the available bandwidth of a sub-network leading to congestion. Congestion is the major issues for network and to deal with this issue TCP variants are evolved. This survey broadly reviewed TCP congestion control variants with their congestion detection, avoidance mechanism and their interconnection. Congestion detection of TCP variants are categorized as delay based, loss based or hybrid. They are modified according to the controlling device entity, i.e., sender side , receiver side , sender-or-receiver side , and sender-and- receiver side. It has been observed that, most of the variants are designed at sender side and are also designed considering network domain i.e. wired, wireless or High Speed/Long Delay. It reveals that these variants are not developed taking into consideration of the end applications which has a particular need and expectation from the TCP, also these variants are developed for specific network environment.

This analysis can be useful for researchers to overcome the lacuna in case of inventing or enhancing a particular variant and also to design a single smart TCP that can handle the range of payloads based on different applications for various network environments.

REFERENCES

[1] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood, New Reno, and Vegas TCP congestion control," ACM SIGCOMM Comp. Commun. Review, vol. 34, no. 2, pp. 25--38, Apr. 2004.

[2] H. Jamal and K. Sultan, "Performance analysis of TCP congestion control algorithm," Int. J. of Comp. and Commun., vol. 2, no. 1, pp. 30--38, 2008.

[3] B. Sikdar ; S. Kalyanaraman ; K.S. Vastola, " Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK" Association for Computing Machinery, 2003.

[4] W. Feng ; S. Vanichpun, " Enabling compatibility between TCP Reno and TCP Vegas," Orlando, FL, USA, USA, 2003.

[5] G. Morabito ; R. Narcisi ; S. Palazzo ; A. Panto, "TCP-Peach and FACK/SACK options: putting the pieces together", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, 2002.

[6] A. A. Hanbali, E. Altman, and P. Nain, "A survey of TCP over Ad Hoc Netw.," IEEE Commun. Surveys and Tutorials, vol. 7, no. 3, pp. 22--36, June 2005.

[7] H. Balakrishnan, V. N. Padmanabhan, S.Seshan, and R. H. Katsz, " A comparison ofmechanisms for improving TCP performanceover wireless link," IEEE/ACM Trans. Netw.,vol 5, no. 6, pp. 756--769, Dec. 1997.

[8] K. S. Reddy and L. C. Reddy, "A survey on congestion control protocols for high speed network," Int. J. of Comp. Sci. and Netw. Security, vol. 8, no. 7, pp. 44--53, July 2008.

[9] Siddharth Trivedi ; Sanjay Jaiswal ; Rituraj Kumar ; Shrisha Rao, "Comparative performance evaluation of TCP Hybla and TCP Cubic for satellite communication under low error conditions", Internet Multimedia Services Architecture and Application(IMSAA), 2010 IEEE 4th International Conference.

[10] Jiyan Sun†, Yan Zhang†∗, Ding Tang†, Shuli Zhang†, Zhijun Zhao† and Song Ci, "TCP-FNC: A Novel TCP with Network Coding forWireless Networks', IEEE ICC, 2015.

[11] Jun ZHANG 1, JiangtaoWEN1, YuxingHAN2, "TCP-ACC: performance and analysis of an active congestion control algorithm for heterogeneous networks", Department of Computer Science and Technology, Tsinghua University, Beijing 2017.

[12] Jacobson, V. Congestion avoidance and control. In Proceedings of SIGCOMM '88 (Stanford, CA, Aug. 1988), ACM.

[13] O. Ait-Hellal, E.Altman "Analysis of TCP Reno and TCP Vegas".

[14] T. Henderson, et al. "Request for comments 6582 - the new reno modification to tcp's fast recovery algorithm". University of Oulu, April 2012. Available at: <http://tools.ietf.org/pdf/rfc6582.pdf>.

[15] M. Mathis, et al. "Request for comments 2883 - an extension to the selective acknowledgement (sack) option for tcp". Pittsburgh Supercomputing Center, July 2012. Available at: <http://tools.ietf.org/search/rfc2883>

[16] G. Morabito ; R. Narcisi ; S. Palazzo ; A. Panto, "TCP-Peach and FACK/SACK options: putting the pieces together", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, 2002.

[17] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast, long distance networks," in Proc. IEEE INFOCOM, vol. 4, March 2004, pp. 2514–2524.

[18] I. Rhee; L. Xu. "Cubic: A new tcp-friendly high-speed tcp variant". 2008.

[19] Siddharth Trivedi ; Sanjay Jaiswal ; Rituraj Kumar ; Shrisha Rao, "Comparative performance evaluation of TCP Hybla and TCP Cubic for satellite communication under low error conditions", Internet Multimedia Services Architecture and Application(IMSAA), 2010 IEEE 4th International Conference.

[20] S. Floyd, "RFC3649—HighSpeed TCP for large congestion windows,"RFC, 2003.

[21] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long distance networks," in Proceedings of PFLDnet, 2004.

[22] T. Kelly, "Scalable TCP: improving performance in high speed wide area networks," Computer Communications Review, vol. 32, no. 2, April 2003.

[23] A. Kuzmanovic and E. Knightly, "TCP-LP: low-priority service via end-point congestion control," IEEE/ACM Trans. Netw., vol. 14, no. 4, pp. 739–752, 2006

[24] G. Marfia, C. Palazzi, G. Pau, M. Gerla, M. Sanadidi, and M. Roccetti, "TCP Libra: Exploring RTT-Fairness for TCP," UCLA Computer Science Department, Tech. Rep. UCLA-CSD TR-050037, 2005.

[25] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," in Proc. ACM MOBICOM, 2001, pp. 287–297.

[26] R. Wang, M. Valla, M. Sanadidi, B. Ng, and M. Gerla, "Efficiency/ friendliness tradeoffs in TCP Westwood," Proc. Seventh International Symposium on Computers and Communications, pp. 304–311, 2002.

[27] G. Yang, R. Wang, M. Sanadidi, and M. Gerla, "TCPW with bulk repeat in next generation wireless networks," IEEE International Conference on Communications 2003, vol. 1, pp. 674–678, May 2003.

[28] F. Wang and Y. Zhang, "Improving TCP performance over mobile adhoc networks with out-of-order detection and response," in Proceedings of the 3rd ACM international symposium on mobile ad hoc networking & computing, New York, NY, 2002, pp. 217–225.

[29] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: a reordering robust TCP with DSACK," International Computer Science Institute, Tech. Rep. TR-02-006, July 2002.

[30] I. Rhee and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64–74, July 2008.

[31] J. Wang, J. Wen, J. Zhang, and Y. Han, "TCPFIT: an improved TCP congestion control algorithm and its performance," Proc. INFOCOM 2011, Shanghai, China, pp. 2894--2902, Apr. 2011.

[32] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," IEEE J. Sel. Areas Commun., vol. 13, no. 8, pp. 1465–1480, October 1995.

[33] K. Srijith, L. Jacob, and A. Ananda, "TCP Vegas-A: Improving the performance of TCP Vegas," Computer Communications, vol. 28, no. 4, pp. 429–440, 2005.

[34] J. Sing and B. Soh, "TCP New Vegas: Improving the Performance of TCP Vegas Over High Latency Links," in Proc. 4th IEEE International Symposium on Network Computing and Applications (IEEE NCA05), 2005, pp. 73–80.

[35] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," IEEE/ACM Trans. Netw., vol. 14, no. 6, pp. 1246–1259, 2006.

[36] A. Venkataramani, R. Kokku, and M. Dahlin, "TCP Nice: A Mechanism for Background Transfers," Operating Systems Review, vol. 36, pp. 329–344, 2002.

[37] V. Tsaoussidis, and C. Zhang, "TCP-Real:Receiver-oriented congestion control," J. of Comp. Netw., vol. 40, no. 4, pp.477--497, Nov.2002.

[38] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," July 2005.

[39] Kai Xu ; Ye Tian ; N. Ansari, "TCP-Jersey for wireless IP communications" IEEE Journal on Selected Areas in Communications (Volume: 22, Issue: 4, May 2004).

[40] R. King, R. Baraniuk, and R. Riedi, "TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP," in Proc. IEEE INFOCOM, vol. 3, March 2005, pp. 1838–1848.

[41] C. P. Fu and S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," IEEE J. Sel. Areas Commun., vol. 21, no. 2, February 2003.

[42] S. Liu, T. Basar, and R. Srikant, "TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks," in Proc. First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), 2006.

[43] A. Baiocchi, A. P. Castellani, and F. Vacirca, "YeAH-TCP: yet another highspeed TCP," in Proc. PFLDnet, ISI, Marina Del Rey (Los Angeles), California, February 2007.

[44] K. Kaneko, T. Fujikawa, Z. Su, and J. Katto, "TCP-Fusion: a hybrid congestion control algorithm for high-speed networks," in Proc. PFLDnet, ISI, Marina Del Rey (Los Angeles), California, February 2007.

[45] Jiyan Sun†, Yan Zhang†∗, Ding Tang†, Shuli Zhang†, Zhijun Zhao† and Song Ci, "TCP-FNC: A Novel TCP with Network Coding for Wireless Networks', IEEE ICC, 2015.

[46] Jun Zhang 1, JiangtaoWEN1, YuxingHAN2, "TCP-ACC: performance and analysis of an active congestion control algorithm for heterogeneous networks", Department of Computer Science and Technology, Tsinghua University, Beijing 2017.