

Congestion controlling schemes for high-speed data networks: A survey

Prashant Kharat* and Muralidhar Kulkarni

Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surthakal, India

E-mails: prashant.kharat@walchandsangli.ac.in, mkulkarni@nitk.ac.in

Abstract. The data networks are basically designed with the aim of maximum throughput and fair resource allocation by managing available resources. A transport layer plays an important role in throughput and fairness with the help of congestion control algorithms (variants). This survey targets mainly congestion issues in high-speed data networks to improve efficiency at connection or flow level. Transmission Control Protocol (TCP) is a dominating transport layer protocol in the existing network because of its reliable service and deployment in most of the routers. A cause of congestion may be different in wired and wireless network and needs to be handled separately. Packet delay, packet loss and time out (RTO) are not caused by congestion in case of wireless network. This has been taken into account in our consideration. To overcome the dominance of TCP, Google proposed UDP based solution to handle congestion control and reliable service with minimum latency and control overhead. In the literature several methods are proposed to classify transport layer Protocols. In this survey congestion control proposals are classified based on situation handled by the algorithm such as pure congestion, link loss, packet reordering, path optimization etc. and at the end congestion control at flow level has been addressed.

Keywords: Congestion control, congestion collapse, data networks, flow level congestion control, high-speed networks, transport layer

1. Introduction

The growth of data networks depends on Quality of Service (QoS) provided to the network users. The queuing dynamics and congestion in the data networks are playing a very important role in QoS. When nodes in the network carrying more data than link can handle QoS deteriorates due to congestion in network and packets are dropping. A network congestion generally results in delay and packet drop which block new connections. As offered load increases in the presence of congestion in network, results into a decrease in network throughput. When the amount of traffic approaches towards the maximum carrying capacity of the network that it can handle then the throughput starts to decrease. If this continues further results in network throughput collapse very rapidly and nothing is delivered. This state of transition referred or called as congestion collapse, shown in Fig. 1.

In recent days almost all computing, communication and control carried out through TCP/IP services. Many of us access some form of TCP/IP networks several times a day, either from fixed locations or while moving and either using wired or wireless medium. Today's network services ranging from traditional information gathering to critical business transactions. As an example in online business even if Page Load Time (PLT) is more users are irritated and this results into profit loss. To maximize the benefits of online services, it is absolutely important to enhance the performance of TCP/IP network.

The emergence of high-speed network is uncovered by deployed popular TCP variants like Reno, NewReno, SACK etc. Network speed normally depends on increase in congestion window size with respect to Round Trip Time (RTT). Also with care of number of data packets network can accept without get exhausted or likely to be called as congested. The growing demand for wireless network technologies highlighted the need of transport layer

*Corresponding author. E-mail: prashant.kharat@walchandsangli.ac.in.

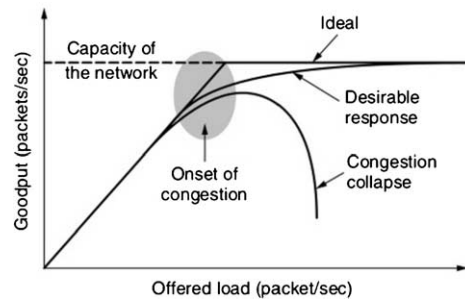


Fig. 1. Congestion collapse event [61].

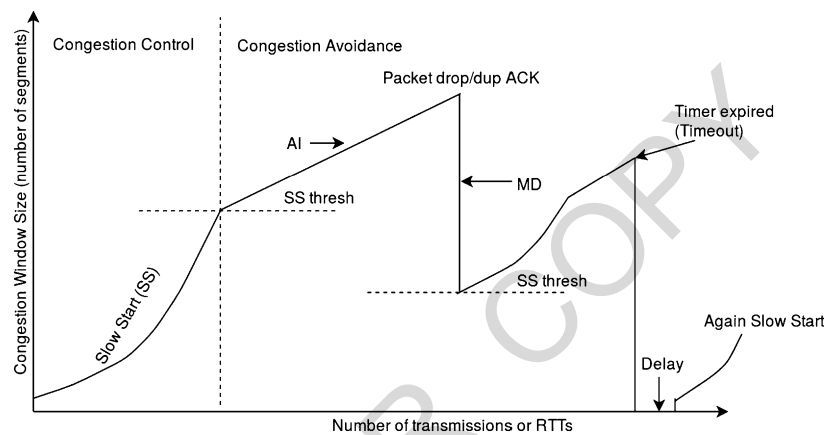


Fig. 2. Congestion control algorithm: AIMD.

protocol modification. As most of the TCP variants are designed and developed for wired networks. These variants blindly decide that congestion is the only cause of packet loss. If packet loss is due to non-congestion related issues like short term radio frequency interference or bit errors or random losses. This means there is no congestion as buffer overflow is not a cause of packet loss. So to recover from this situation there is a demand to develop loss recovery mechanism rather than simply halving congestion window size.

To resolve congestion collapse problem, lots of proposals were submitted by different researchers throughout the world. Most of them suggested similar kind of solutions like network aware rate control with, receiver driven flow control. This is the reason why congestion window concept related to network comes into existence. A congestion window size is the estimation of number of the data packets network can accept without any congestion. To calculate bound on the number of data packets present in the network for a particular situation e.g. receiver window (flow control limit), is less than congestion window (congestion control limit), is bound for outstanding packets in the network. This is the real definition for TCP rate bound, but in most of the proposals, the only congestion window is considered as a rate controlling parameter. Normally it is assumed that data processing capacity of the end nodes are much higher than the data transfer rate of the network.

The number of bytes transmitted by sender depends on congestion and receiver window size. A minimum of these two windows decides the number of bytes sent by the sender. When the connection gets established between the sender and receiver, sender initializes congestion window. The size of congestion window equal to the maximum segment in use on that connection (maximum one segment). As shown in Fig. 2, on arrival of each acknowledgment for burst (on arrival of every acknowledgment congestion window incremented by one) double the congestion window size. The congestion window size keeps growing exponentially until either a timeout occurs or it reaches up to the receiver's window size. This behavior of window size called Slow Start. The threshold is the

third parameter used to control window size up to certain level (initially it was 64KB) to avoid packet dropping. In case a packet is not received within RTO, that is time-out occurs then the threshold is set to half of previous and the congestion window is reset to one maximum segment size.

To improve up on performance in terms of utilization of resources and network congestion, congestion control algorithms are developed. But the effectiveness of these algorithms is based on effective handling of congestion, resource utilization, fairness (everyone gets a fair share of resource used) and efficiency (network resources are used well) of congestion control algorithm. To achieve above said objectives, Chiu and Jain [10] formulated fairness measure (F) shown in equation-1 called Jain's fairness index. This single dimensionless fraction is the network resource consumption function by each user sharing the same path.

$$F(f_1, f_2, \dots, f_n) = \frac{(\sum_{i=1}^n f_i)^2}{n \cdot \sum_{i=1}^n f_i^2} \quad (1)$$

where,

n = number of flows sharing the resource and f_i = resource allocation for i th flow.

Jain's fairness index is a predominant fairness measure for TCP flows. This fairness index ranges from $1/n$ to 1. The value of this index tends to 1 only if each flow has an equal share, and tends to $1/n$ if a single flow acquires all network resources.

A congestion window size is one of the important parameter and decide time required for sender to transfer available data to receiver. The Equation (2) is used to calculate time (T) in second, which is required for sender to send data.

$$T = RTT \cdot \log_2 \left[\frac{N}{cwnd_{initial}} \right] \quad (2)$$

where,

RTT = round trip time in second, N = number of segments required to transfer data, $cwnd_{initial}$ = initial congestion window size in segments and $1segment = 1520 - 40(TCP\ header\ size) = 1460B$.

Phase plot in Fig. 3 shows fairness and efficiency of the network. In this each axis represents a particular user or sender allocation. In this case, there are two users and their allocations are represented by x_1 and x_2 . If the capacity of the network is C , then an optimal operating line is at $x_1 + x_2 = C$.

In general, Additive Increase Multiplicative Decrease (AIMD) converges to fairness and efficiency. Suppose if at any point 1, TCP window size of both senders Additively Increase (AI) their sending rates, additive increase results moving line parallel to x_1 and x_2 towards efficiency line since both senders increase their rates by same amount. Additive increase continued until network becomes overloaded that is touching to efficiency line. At this point of time, both senders decrease their sending rate by a Multiplicative Decrease (MD) fashion with a specific factor, in this case, half of previous. If the line is drawn parallel to x_1 and x_2 then new operating point 2 comes, again senders increase their sending rates in additive increase fashion and once again hits efficiency line towards optimal point. In this way, sender's rate converges towards optimal operating point [55]. This means, if additive increase applied that increases efficiency where as when multiplicative decrease applied, improves fairness.

In transport layer till today, TCP contributed a lot for networking industry by delivering remarkable results, also it has some issues to deal with. Since about 30% to 35% of all Internet traffic route through Google server, Google has detail analysis of TCP performance in recent networks. In addition to this Google's Chrome browser most popular browser in the web world with approximate 40% of market share. Based on above experience in 2013, Hamilton *et al.* [28] from Google started implementation new protocol based on UDP named as Quick UDP Internet Connections (QUIC). Protocol stack is shown in Fig. 4, shows the deployment option of QUIC. QUIC is a simple protocol suitable for supports application protocols for booting which is on top of UDP. As per the

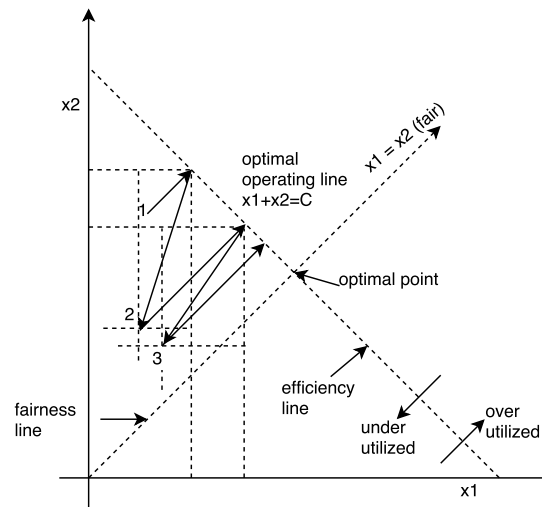


Fig. 3. Phase plot for fairness and efficiency.

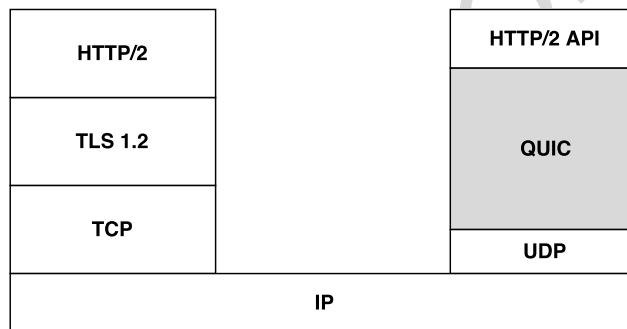


Fig. 4. QUIC placement in protocol stack.

discussion on replacing existing TCP transport layer protocol to get out of its complex structure by designing new. Google designed protocol based on UDP is really challenging and very complex task. As an example, UDP interactions with many layers of deployed middle boxes like Network Address Translation (NAT) traversal is a very complex task. Initially, developed TCP variants are designed to avoid overflow at input buffers at the receiver end. The mechanism developed is based on the receiver's window size. In this technique, a sender transmits a prepared data packet. This must not increase receiver's capacity (receiver window size) and in case receiver unable to process data as fast as sender sends the buffer overflow event occurs. And to avoid this receiver reports reducing sliding window size. That is the whole transmission will eventually synchronize with receiver's processing rate.

To avoid unnecessary packet drops precaution need to take by the end nodes. A Sally Floyd suggested solution by sending congestion notification to the sender [15]. When bulk of data is considered in that case data arriving time of last packet is only important and no need to take care of individual packets. For delay sensitive traffic such as mice traffic which exist for very little time in the network for example like button in Facebook or telnet traffic. If unnecessary packet drop or packet retransmission takes place then that result into very high throughput degradation for such low bandwidth network. So to avoid this Sally Floyd designed Explicit Congestion Notification (ECN) and this has been investigated by Kwon *et al.* [42] and further extended by Ramani and Karandikar for wireless and lossy links [3].

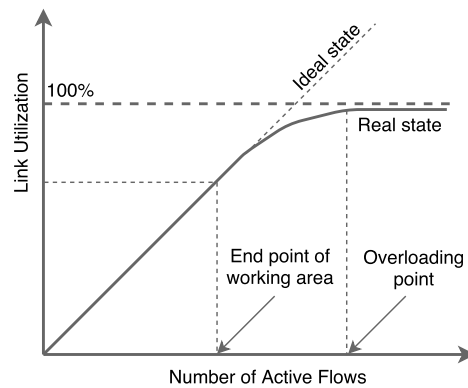


Fig. 5. A flow based traffic model for uncongested backbone links [70].

Mainly congestion control approaches are divided into two groups Reactive (Loss-based) and Proactive (Delay-based) to estimate congestion state in the network. But still after that due to popularity of TCP many more approaches comes into existence based on fine tune congestion state impairments. However, now days due to digitization more digital documents generated by numerous applications. This is an elastic traffic and network performance is depends on number of connections currently sharing links of the network, which varies based on begin and end of the flows. To analyze flow level performance of the network a flow based network state traffic model shown in Fig. 5 has been suggested by Sukhov *et al.* [70].

The flow level congestion control protocol standards sometimes remain unaware of network resources available, which creates unexpected effects on the Internet. The congestion control means intelligent network resource aware and makes effective use of resources available in packet-switched networks. Congestion control is one of the largely studied areas in the Internet research conducted over the last more than 25 years, and a number of proposals submitted and targeted to improve various aspects of the congestion-responsive data flows. Several surveys were prepared by researchers targeting specific requirement of interest like congestion control in ad hoc networks by Hanbali *et al.* [29], congestion control for mobile ad-hoc networks for single and multiple flow by Lochert *et al.* [49], and Ikeda *et al.* [32], congestion control for non-TCP protocols by Widmer *et al.* [77], congestion control for networks with high levels of packet reordering by Leung *et al.* [47], fairness issues in congestion control by Hasegawa and Murata [30], Host-to-host congestion control for TCP by Afanasyev *et al.* [2], congestion control for wireless networks by Balakrishnan *et al.* [3]. In this survey, we tried to collect, classify and analyze major congestion control schemes for wireless high-speed data networks. Table 1 gives summery information of standard TCP variants with their services and limitations.

1.1. Structure of the paper

Next part of the paper divided as, Section 2 is devoted to congestion controlling challenges that gives current scope of work in terms of challenges with their brief idea. In Section 3 congestion controlling strategies for random and link losses present due to a wireless medium are reviewed. In Section 4 congestion controlling algorithms which improve efficiency in high speed and long delay networks are reviewed. Section 5 deals with congestion control at flow level (see Fig. 6). Finally, paper end-up with conclusion and future research directions in the Section 6.

2. Congestion controlling challenges

In today's world, congestion control mechanisms are presented as an artificial feedback system and deployed in all routers. The Internet features are continuously growing in size, diversity, and applicability. These features

Table 1
Features and limitations of standard TCP variants

Variant (Year)	Features	Solution provided for	Limitations
TCP-Tahoe [34] (1988)	Reactive, slow start, congestion avoidance and fast retransmit.	Congestion control, fast network resource discovery and long-term efficiency.	Already obsolete, straining the network with high-amplitude periodic phases.
TCP-Reno [35] (1990)	Reactive, fast Recovery	Congestion control with slow start.	Severe performance degradation in the presence of consecutive packet losses, random losses and packet reordering, unfair for different RTT flows, low efficiency for high-speed/long-delay networks.
New Reno [18,19] (1999)	Reactive, fast recovery, offering resistance to multiple losses.	When multiple packet losses occurs as part of a single congestion event.	Prolonged recovery, waste resources if actual losses deviates from assumed pattern.
TCP-SACK [53] (1996)	Additional information in feedback messages, successfully delivered data packets report.	Limited information available in cumulative ACK.	TCP specification restricts the length of option field to 40 bytes, SACK is not a universal solution to the multiple loss.
Eifel [50,51] (2000)	Differentiation between transmitted and retransmitted data packets.	To alleviate negative effects of packet reordering in TCP throughput, distinguish reordering and real loss.	Prolonged recovery, waste resources.
DSACK [20] (2000)	Reporting duplicate segments, receiver side modification.	Reordering or replicate data packet.	Receiver can send wrong information, either by intention or unintentionally.
TCP-RR [72] (2004)	Reactive, fast recovery, exponentially backs off the sending rate after detecting the first packet loss within a window	Packet recovery in single congestion event, link utilization,	Fails to detect and recover in multiple loss, waste of resources if actual loss deviate from assumed.

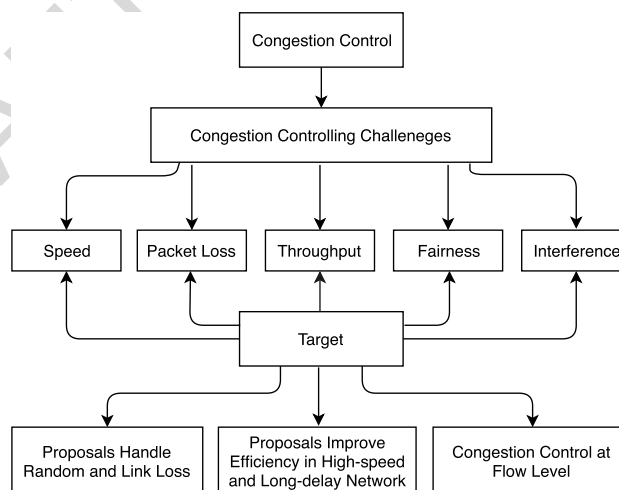


Fig. 6. Paper organization.

are playing very important role to integrate other networks like trading, banking etc. With a firm understanding of how this fundamental resource is controlled becomes even more decisive. Following are the key challenges in front of researchers working for congestion detection and controlling schemes [31,58].

2.1. High speed

Within past few years, the growth rate of Web technology and Performance Optimization industry indicates growing importance and demand for speed of operation. Results show that speed is not a psychological need in accelerating and connecting world. In online business, in which within one click items are get available to our door step. But the performance of this business decided by how fast we compare with a competitor.

- More the speed leads to faster sites and better user satisfaction.
- More the satisfaction better will be the user retention.
- More the retention lead to higher conversions.

Future demands more and more speed and to satisfy or fulfill demand we need to understand many factors and analysis of fundamental limitations. Latency and Bandwidth are the critical components plays very important role to analyze performance of the network in terms of speed.

2.2. High packet delivery ratio

This is the ratio of number of correctly received packets at the receiver and the total number of packets transmitted by the sender. Major sources of packet loss are (i) buffer overflow at the intermediate routers, and (ii) packet corruption caused by transmission errors. A high packet loss rate can severely degrade the performance of data and multimedia applications.

2.3. Effective throughput

This is defined by the number of application bytes transferred in seconds. For large file transfers, the effective throughput of the application is a key performance measure. An inefficient transport layer algorithm or implementation can significantly reduce the effective throughput even if the underlying network provides a very high-speed communication channel.

2.4. Less throughput variation

Throughput variation is a metric to measure the variability in the received bandwidth over a given time scale. In general, the larger the time scale, the lower the throughput variability. For a given context, it is important to define a time scale over which throughput variability should be measured.

2.5. Fairness

Fairness becomes important when two or more applications compete for resources in a congested router. Fairness can be defined over the long term or short term. Long term fairness refers to the fair allocation of resources in the long run. Short-term fairness is defined in much smaller time scales. A given network algorithm may allocate bandwidth fairly in the long run yet exhibit unfairness in the short-term.

2.6. Short-term radio frequency interference

Day by day demand for a wireless network is growing and surely it will grow further. So as our traditional protocols are basically designed and developed to serve wired network demands, needs modifications as per the today's demand for wireless networking. If we think of transport layer design in wired network, congestion is the primary cause of packet loss. So that traditional TCP is unable to react to packet loss which is not related to congestion. For example, if packet loss is due to interference caused by short term radio frequencies in which there is no router overflow event but still TCP reaction is to reduce transmission rate by reducing congestion window size. Instead of that reducing congestion recovery from loss is the solution to this problem and continue transmission with same rate this results into unnecessary throughput reduction.

3. Proposals handle random and link loss

The growing demand for mobile communication and wireless network technologies such as Cellular Communication or Wireless Local Area Network (WLAN) shown in Fig. 7, highlighted the need of transport layer protocol modification. As we know through different surveys most of the TCP variants are designed and developed for wired networks. These variants blindly decide that congestion is the only cause of packet loss [1]. If packet loss is due to non-congestion related issues like short term radio frequency interference or bit errors or random losses this means there is no congestion as buffer overflow is not a cause of packet loss. So to recover from this situation we need to develop loss recovery mechanism rather than simply halving congestion window size. Figure 7 shows typical hybrid network with lossy links in which link losses are mainly due to interference and random bit errors. Table 2 shows summary of proposals which contribute towards random and link loss.

First time in 1999, congestion control policies for wireless multimedia CDMA network is applied by Lu *et al.* [48] in which they have applied burst level congestion control is applied. In this they have studied reverse link and issues related to multimedia traffic in wireless network. Balakrishnan *et al.* [3] compared different mechanisms

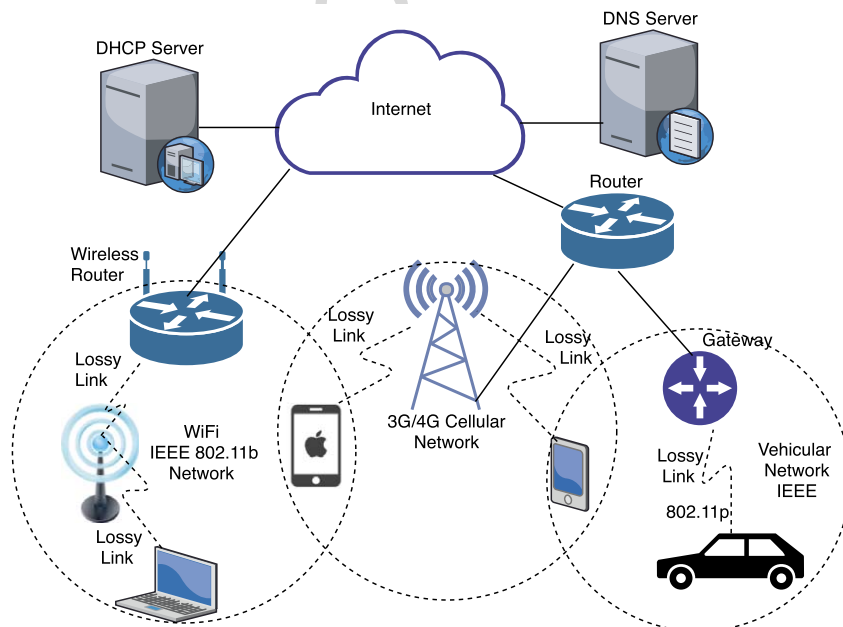


Fig. 7. Typical network scenario with lossy links [65].

Table 2
Proposals handling random and link losses

Variant (Year)	Features	Solution provided for	Limitations
TCP Westwood (2001)	Available bandwidth estimation (ACK granularity), faster recovery.	Random errors due to wireless interference.	Unfair if network has limited buffering capability, Bandwidth estimate gives inaccurate results.
TCP Westwood+ (2004)	Estimate of available bandwidth (RTT granularity)	Random errors due to wireless interference	Unfair if network has limited buffering capability, bandwidth estimate gives inaccurate results.
TCPW CRB (2002)	Available bandwidth estimate (combination of ACK and long-term granularity), identifying predominant cause of packet loss.	Random errors due to wireless interference	Unfair if network has limited buffering capability, Only simulation results are available no real life experimentation
TCPW ABSE (2002)	Available bandwidth estimate (continuously varied sampling interval), varied exponential smoothing coefficient	Random errors due to wireless interference	Unfair if network has limited buffering capability, only simulation results are available no real life experimentation
TCPW BR (2003)	Loss type estimation technique (queuing delay estimation threshold, rate gap threshold), retransmission of all outstanding data packets, limiting retransmission timer back-off	Random errors due to wireless interference	Only simulation results are available no real life experimentation
TCPW BBE (2003)	Effective bottleneck buffer capacity estimation, adaptive reduction coefficient, congestion window boosting	Unfair if network has limited buffering capability, hybrid estimation technique	Only simulation results are available no real life experimentation
TCP Jersey (2004)	Available bandwidth estimation and congestion warning approach	To improve performance in heterogeneous network	Only simulation results are available no real life experimentation

for improving TCP performance over wireless links and tested in LAN and WAN environments using throughput and goodput as the metrics. It has been shown that in traditional wired networks congestion is the primary cause of packet loss. Networks with wireless and other lossy links also suffer from significant losses due to bit errors and handoffs. The selective acknowledgments and explicit loss notification techniques are quite effective and show significant performance improvement in lossy link especially when losses occur in bursts. Ratnam and Matta [64] in their Wireless-TCP (WTCP) handle loss due to wireless link and mobility by using local retransmission strategy. Samaraweera [66] in his Non-congestion Packet Loss Detection (NCPLD) algorithm differentiates loss due to congestion from loss due to link noise. If there is packet loss due to any reason recovery is the reaction for that may be a retransmission of the packet is one of the solutions. But as soon as packet drop takes place reduction in the rate of transmission is also the reaction assuming that there is congestion. To identify loss is due to non-congestion type that is may be due to link noise. A RTT base calculations are used to identify network utilization and if it shows underutilized fast retransmission takes place assuming that loss is due to non-congestion. As we come to know that loss is due to non-congestion it continues transmission with the same rate that is it avoid window reduction.

Parsa and Garcia-Luna-Aceves proposed TCP Santa Cruz to improve TCP congestion control for mix networks maybe we can say heterogeneous network or wired cum wireless network. This protocol address problem created due to asymmetrical paths, out of order delivery, lossy links, less bandwidth and delay variation. Here only forward path delay is measured instead of round trip time. So that to avoid congestion specific threshold is assigned to a number of packets present in the bottleneck link. So here there is no need to count acknowledgments to set next limit for congestion window. The experimentation results through simulation show that TCP Santa Cruz achieves significantly higher throughput, smaller delays, and smaller delay variances than Reno and Vegas [57]. Ma *et al.* developed Fast-TCP for IP network with a wireless link. The basic idea of the Fast-TCP method is to delay the

ACKs being transferred from the TCP destination towards the TCP source. Fast-TCP can speed up TCP flow control time, reduce buffer oscillation, increase bandwidth utilization, increase throughput, and reduce packet losses in the IP networks [52].

Ramani and Karandikar suggested modified ECN mechanism for wireless and lossy links in this they developed sender aware mechanism which reports some of the packet losses are not because of congestion. This is done through explicit feedbacks from the network in the form of ECN (notification) about the link congestion status in the network [62]. Saverio *et al.* in TCP Westwood rely on end-to-end bandwidth estimation to differentiate the cause of packet loss (congestion or wireless channel effect) which is a major problem in TCP Reno. Unlike TCP Reno, Westwood using slow start threshold and congestion window parameters related to effective bandwidth at the time of congestion rather than three duplicate ACKs concept to control congestion window [67]. In Westwood random losses over radio link are differentiated from congestion and cause of random loss is radio interference and avoids unnecessary congestion window reduction. TCP Westwood is extremely effective in mixed wired and wireless networks where they claimed throughput improvements up to 550% by their observations [73,74].

Fu and Liew in TCP-Veno [23], monitors the network congestion level and uses that information to decide whether packet losses are likely to be due to congestion or random bit errors. TCP-Veno modifies multiplicative decrease to improve its applicability by using slow start threshold adjustment based on the level of congestion rather than fix factor. Veno trying to mention connections in the operating point region for longer or full time so that networks whole bandwidth gets utilized. One of the salient features of Veno is it's only sender side modification. Xu *et al.* developed TCP-Jersey [78] which is capable of distinguishing the packet loss due to wireless effects like radio interface, bit error, random errors from the congestion effect and decisions are made. TCP-Jersey divides into two parts Available Bandwidth Estimation (ABE) algorithm and routers with Congestion Warning (CW) mechanism. ABE is a sender side modification which contentiously monitors and estimates bandwidth available to the application. And based on ABE's estimation sender will adjust the rate of transmission. CW is a router configuration for a network through which routers generates and send an alert message to end nodes by marking all packets as soon as incipient (beginning to develop) congestion is detected. Based on packet marking sender comes to know whether packet loss is due to congestion or due to wireless link errors [38].

Biaz *et al.* [5] proposed de-randomization to distinguish congestion losses and random losses. As described in the introduction part most of the variants designed basically by assuming packet loss is due to congestion. To mitigate this misconception de-randomization is one of the solution in which bias is used to differentiate congestion and wireless link loss. Mainly this concept is used in heterogeneous wireless error prone links. Through simulation results, accurate boundaries are found out to differentiate congestion loss and wireless link loss. Due to this technique, they got 95% accuracy in congestion loss detection where as 75% accuracy in case of wireless link loss detection.

Lachlan *et al.* proposed Curtailing the Large TCP Advertised windows to Maximize Performance (CLAMP) protocol [44] to allocate resources fairly in Wireless Networks. On wireless link through interaction between the scheduling of MAC layer and flow control, they try to manage a share of each flow in such a way that each gets same bandwidth allocation. They have used the infrastructure based wireless network in which multiple users get serve by one common access point. Each user suffers from fading effect and scheduler allocates a channel to each user based on channel quality with consideration of fairness and latency. As per results, TCP NewReno compromises in fair channel allocation as far as MAC schedule is a concern. CLAMP is a receiver side modification which controls TCP sender rate by setting receiver window size (*rwnd*) of the receiver and accordingly, the scheduler allocates bandwidth to each user [13].

Rath *et al.*, proposed a cross layer based congestion control technique called Reno-2 for wireless networks. In this, both transport layer and physical layers are working hand in hand to control congestion. In Reno-2, as per channel conditions and interference level, physical layer controls the power of transmission signal and transport layer controls congestion by controlling flow [63]. Matlab simulation results show that the cross layer congestion control technique provides performance improvement in terms of throughput and congestion window variations. The data transmission using wireless network suffers from insufficient bandwidth, transmit latency and high interference. The standard TCP congestion control mechanisms are fail to identify available bandwidth or buffer size

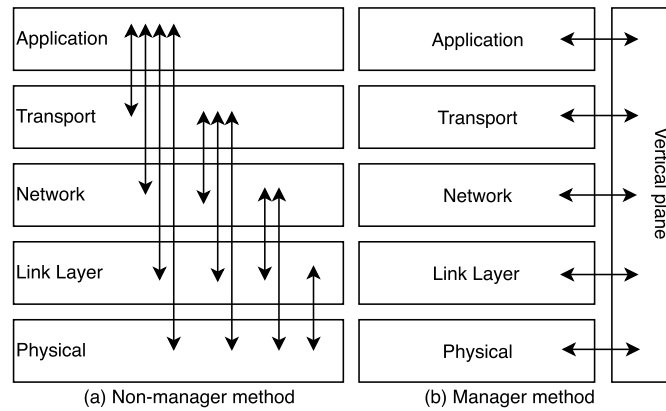


Fig. 8. Cross-layer-based approach [59].

and blindly halves congestion window after receiving three duplicate ACKs. Chang and Li proposed cross layer messages to identify bottleneck link status as a access or shared. The congestion window size has been adjusted as per the link location and packet loss reason is differentiated as network congestion or random loss. The cross-layer messages based on Adaptive Modulation and Coding are used to adjust congestion window size. In this the numerical results proved that the proposed approach significantly improve the Goodput with precise packet loss identification at various loss rates. As an example for wireless link with 4% packet loss rate the proposed approach increases Goodput up to 77.25%, compared with NewReno [9]. Figure 8 shows cross-layer design approaches (a) for small amount of information exchange direct communication between each other is possible, normally defined as a non-manager method, (b) in this indirect communication takes place between layers through vertical plane (shared information), normally defined as a manager method [22].

Berg *et al.* proposed a complete distributed algorithm, which creates prioritized TCP flows to allocate network resources (bandwidth) within flows. This enables autonomous conversions to loss of network resources due to cyber attack or failures by ensuring that users receive prioritized utility from available network resources. This approach is fully-distributed, self-adaptive to prioritization of mission-critical TCP with a weighted Nash Bargaining solution to distribute network bandwidth among the flows [71].

4. Proposals improve efficiency in high-speed and/or long-delay networks

To illustrate the problem of improving efficiency in high speed and long delay networks sometimes referred as a bandwidth delay product problem. To discover network resources minimum time required is $[D \times (cwnd/RTT)]$, where D is maximum data packet and $cwnd$ is congestion window size. In a network with Giga bit capacity, it would takes almost hours of time to deliver all packets that depend on the time required to discover network resources. In this section we have discussed algorithms for high-speed or long delay links in the network. This section simultaneously provide use of network resources, quick response to network changes and fair to other flows present in the network [6,43].

For speed, convergence time is a most important parameter; it is the speed (time) with which system reaches to its stable state (target load). However, because of binary nature of feedback used in the network to control congestion, the system generally does not converge to a single steady state. As shown in Fig. 9, time to reach target load value, determines responsiveness and size of oscillation, determine smoothness. Ideally, time and oscillations to be small [16,17]. Normal TCP variants are suitable to solve pure congestion problem but can't work for high-speed networks such as optical or satellite network in which reaction time is very less. That is normal variants are not efficient to handle congestion in high-speed network, so to improve efficiency, Floyd proposed High Speed (HS) TCP [16,17].

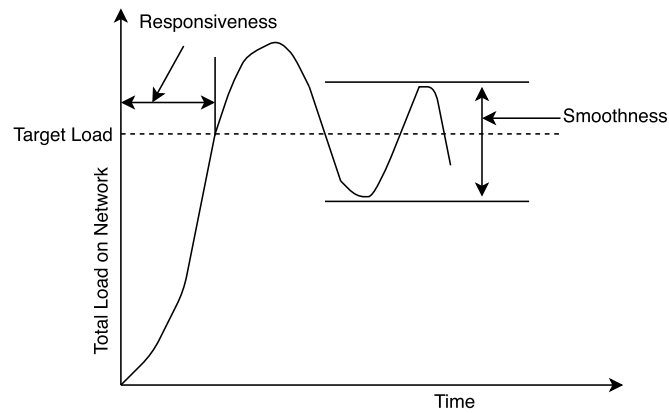


Fig. 9. Network convergence response [10].

Table 3
Proposals improve efficiency in high-speed and/or long-delay network

Variant (Year)	Features	Solution provided for	Limitations
CP Santa Cruz (1999)	Path asymmetries, out-of-order packet delivery, and networks with lossy links, limited bandwidth and dynamic changes in delay.	High-speed, lossy network	Unfair for different RTT flows, low efficiency for high-speed/long-delay networks.
Fast-TCP (2000)	Flow control mechanism is located on the output of the access unit to the IP interface and controls ACK output rate according to congestion information from forward connection.	High-speed, lossy network	The flow cannot utilize the available network resources fairly.
HS-TCP (2003)	AIMD the congestion window size, limited slow start.	High-speed long-delay networks.	Fairness if flows have different RTTs.
STCP (2003)	MIMD scheme for congestion avoidance.	Data transfer effectiveness in high-speed/long-delay networks.	Due to MIMD policy flows are extremely unfair.

In this experimental approach mainly efficiency improvement in high BDP networks and fairness in the lossy environment are the set objectives. To achieve set objectives in HS-TCP standard NewReno protocol is modified by replacing increase factor α as a function of congestion window size (w), $\alpha(w)$ and decrease factor β by $\beta(w)$ in Congestion Avoidance (CA) phase and after loss detection respectively. When congestion window size is more that is when no loss HS-TCP grab network resources very fast or aggressively and vice-versa after loss detection. This opportunistic behavior improves an efficiency of high-speed/long-delay networks. But this aggressive and conservative behavior causes significant packet loss and occurs frequently. Table 3, shows summary of proposals which contribute towards High-speed and/or long-delay networks.

As many of the researchers claim that traditional AIMD congestion control approach is not suitable for high-speed and long-delay networks. Kelly says that, AIMD is not effective and proposed Scalable TCP (STCP) as an alternative to HS-TCP [40]. In AIMD complicated task is to calculate increase/decrease rate coefficients and in STCP this is simplified by introducing Multiplicative Increase Multiplicative Decrease (MIMD) approach. As increase/decrease approaches used in STCP are very aggressive it sharpens the characteristic graph. But because of this aggressiveness, it creates intra-fairness congestion and it leads further to the congestion collapse. Particularly for higher RTT values due to MIMD policy STCP flows are very unfair to both STCP (intra-fairness) and other TCP flows (inter-fairness) [60].

Table 3
(Continued)

Variant (Year)	Features	Solution provided for	Limitations
H-TCP (2004)	Congestion window increase steps as a function of time elapsed since the last packet loss detection, scaling increase step to a reference RTT, MD coefficient adaptation.	Fairness and effectiveness in high-speed/long-delay networks.	Prolonged recovery, waste resources.
TCP Hybla (2004)	Scaling the increase steps in slow start and congestion avoidance to the reference RTT, data packet pacing, initial slow-start threshold estimation.	Throughput with NewReno in long-delay networks, RTT fairness in heterogeneous networks.	Like Reno unable to work effectively in high-speed networks with relatively small delay.
BIC TCP (2004)	Binary congestion window search, limited slow start	RTT fairness of STCP and HS-TCP.	In certain environment have low RTT-fairness and inter-fairness values.
TCPW-A (2008)	Logarithmic congestion window increase.	Effectively utilize resources of high-speed or long-delay networks.	In some scenarios it loose its scalability like initial SS phase prematurely terminated will not able to rapidly discover new high-speed networks.
TCP Cubic (2008)	Window control based on cubic function of time elapsed from last congestion event.	Scales well in high-BDP networks and has good intra, inter and RTT fairness properties.	Network resource utilization is less than 100%, and more packet loss in the networks.
FAST TCP (2003)	Constant-rate congestion window equation-based update.	Idea of congestion control with queuing delay for high-speed long-delay networks.	Selection of alpha value is challenging task.
TCP Libra (2005)	Adaptation of the packet pairs to estimate the bottleneck link capacity, scale the congestion window increase step by the bottleneck link capacity and queuing delay.	To resolve scalability while preserving and improving the RTT-fairness.	Not always outperform other congestion control approach, worsen further because of estimation biases.
TCP NewVegas (2005)	Rapid window convergence, packet pacing, packet pairing.	Reduce convergence time and improve high-BDP link utilization, solution for generation of bursty traffic during initialization, bias estimation.	No scalable features are designed for congestion avoidance and fast recovery.
TCP AR (2005)	Congestion window increase steps as a function of the achievable rate and queuing delay estimates.	To improve TCP performance and preserve friendliness in high-speed networks.	When queuing delay is wrongly estimated to be close to the max, then losing ability to scale in high-BDP network.
TCP Fusion (2007)	Congestion window increase steps as a function of the achievable rate and queuing delay estimates.	Effectively utilize resources of high-speed or long-delay networks.	Manual fusion configuration is highly undesirable and usually impossible.
TCP Africa (2005)	Switching between fast (HS-TCP) and slow (NewReno) mode based network state estimation.	Effectively utilize resources of high-BDP networks.	Not been implemented in real network.
QUIC (2013)	UDP based, Multiplexed, secure equivalent to TLS and connection migration, reliability and congestion control.	High-RTT/low-bandwidth networks.	Poor performance in high-speed networks.

Leith and Shorten suggested different solution H-TCP to cope up with fairness (inter, intra and RTT) for different RTT value flows are competing and to improve the efficiency of TCP [45,46]. From previous studies, it's proved that flow having greater RTT value always loses to flows with shorter RTT value. In H-TCP, congestion window size increase function, α of CA phase is a non-decreasing function with respect to time elapsed, Δ since congestion event. Whereas H-TCP increase congestion window size by $\alpha(\Delta)$ for any reference RTT instead of per RTT to solve fairness problem.

Caini and Firrincieli, claim that standard TCP-NewReno gives lower throughput for long-delay networks [7]. Normally networks like satellite (long-delay) RTTs are differing largely causes terrible unfair resource distribution. To resolve this TCP-Hybla algorithm is proposed, which suggested modification in Slow-Start and Congestion Avoidance phases by making them semi-independent on RTT value. In TCP-Hybla scaling factor, ρ is calculated as, $\rho = RTT/RTT_{ref}$. That is higher RTT value higher scaling factor results into fast congestion window size, w increase with each received ACK. Increase factor in each phase is as follows. In Slow Start phase: $w = w + 2^\rho - 1$ and In Congestion Avoidance phase: $w = w + \rho^2/w$ In addition to this TCP-Hybla supports i) to calculate initial slow start threshold by estimating network capacity, which helps to improve convergence speed. ii) to smooth busy nature of transmission by setting the minimum delay between consecutive two packets using pacing.

In BIC (Binary Increase Congestion-control) TCP proposal, Xu *et al.* suggested a solution for RTT unfairness when the loss is detected simultaneously by two competing flows [79]. In such a situation in HS-TCP flow having RTT, x times smaller will get network share $x^{4.56}$ times more and in STCP for similar situation flow with larger RTT will get nothing. Basically, BIC is a modification in NewReno by adding Rapid Convergence phase. In rapid convergence phase, optimal congestion window size based on available network resources is discovered by binary search approach on detection of packet loss. There are two limits w_{min} and w_{max} are set to search congestion window size for successful packet delivery. On every achievement of w_{max} , w_{min} raises by setting to previous congestion window size. As soon as packet loss detects w_{max} set to current congestion window size and enters into the fast recovery phase of NewReno. Also, BIC using a different strategy for low loss network environment by reducing multiplicative decrease factor from 0.5 to 0.125 when congestion window size is more than 38 (value taken from HS-TCP, $w = 1.2/\sqrt{p}$, where, p is loss rate 10^{-3} pkt/s). To avoid uncontrolled growth of congestion window size as binary (logarithmic) search is used, precaution is taken by BIC. BIC uses limited slow start and also put a limit on rapid convergence when search range is too high (high BDP). That is again there are two threshold values to control growth factor by S_{max} to limit maximum congestion window size growth value and S_{min} to increase congestion window size minimum by this factor in rapid convergence stage. In BIC congestion window probing decreases as window size approaches to set target value.

Ha *et al.*, proposed an enhanced version of BIC as a CUBIC, in which they have introduced RTT independent congestion window growth function [27]. To achieve this CUBIC picked up H-TCP approach of congestion window size calculation as a cubic function of elapsed time Δ since last congestion event. $w = C(\Delta - (\beta \cdot w_{max}/C)^{1/3})^3 + w_{max}$ where, $C =$ is predefined constant This function preserves BIC properties RTT fairness, limited slow start and rapid convergence. As an additional precaution CUBIC have a mechanism to ensure performance must not be worse than standard Reno by checking simultaneously by calculating W_{reno} . With other experimental studies, it shows that performance and fairness property of CUBIC is very good. Also as this is available in Linux TCP suite (kernel version 2.6.16) currently this is most used congestion control algorithm.

Jin *et al.* proposed TCP-FAST, which is similar to Vegas with some modifications in congestion window estimation. In FAST congestion window, w is updated with fixed-rate. $w = w_{current} \cdot (RTT_{min}/RTT_{current}) + \alpha$ Based on scalability and stability α is need to choose. There is tread-off if α is very large it can scale to any like the wrong estimation of queuing delay; value of BDP very fast but in this substantial convergence to w is a problem, nothing but stability problem, however, if α is small it stabilizes but scalability is a problem. To avoid parameter fluctuations in the FAST exponential smoothing is used to calculate congestion window size. Results with different experimentation show that FAST is not friendly with standard TCP variants like Reno, NewReno or SACK [36,37,75].

NewVegas is an extended version of Vegas suggested by Sing and Soh. NewVegas scheme defines a new phase called Rapid Window Convergence by retaining original benefits of Vegas and standard NewReno [69]. Particularly delay based congestion control approach is more beneficial for high-BDP networks due to sufficient time to

converge network for a different conditions. In Rapid Window Convergence phase slow-start state of the network is extended when buffering is more or it exceeds the certain threshold value. But as a precaution resource probing must be under control means it is exponential like Vegas but with reduced intensity. In this phase for every RTT, congestion window is increased by number of packets x as follows. $x = (w_r)^{(-2^{3+n})}$ where, w_r is congestion window value when window convergence moves from normal to rapid state. n number of times early congestion indicator triggers in Rapid Window Convergence phase. If n crosses certain value (suggested is >3) it terminates Rapid Window Convergence phase to normal Vegas Congestion Avoidance phase. In NewVegas if packet loss occurs it switches to fast recovery mode, where as when time out (RTO) occurs it will switch to Slow Start phase by resetting congestion window size. To avoid busty traffic at the initial phase of transmission because of rapid growth packet pacing technique like Hybla is used. In pacing technique, some minimum delay is set between consecutive packets. To improve efficiency NewVegas holds transmission till congestion window size increased to required congestion window size to transmit the available packet. So this resolves RTT estimation problem.

To improve TCP performance and preserve friendliness to standard TCP in high-speed networks, Shimonishi and Murase suggested a solution in terms of Adaptive Reno (AR) [68]. In this proposal combination of constant increase and scalable increase based on congestion state kind of technique is used. When the network is congestion free ($queuing\ delay = 0$) then scalable window update function is used which supports to a rapid growth of congestion window size. Whereas, when network experience congestion ($queuing\ delay = max$) then scalable function becoming zero. In this way, adaptation takes in terms of switching from normal growth to rapid growth of congestion window size based on congestion state. AR improves network utilization up to a certain extent with few of limitations it fails for high-BDP value.

A Modified Linear Quadratic Gauss (MLQG) [26] is proposed to control over long range dependence network and compared with standard LQG algorithm. In this congestion issue is solved with a stochastic optimal control problem. Kaneko *et al.* combine some good characteristics of Westwood, DUAL and Vegas called Fusion [39]. Normally in fusion technique, to achieve expected objectives properties of different variants were collected and used. In Fusion, algorithm author suggested three different levels (in terms of the threshold value) of queuing delay in second.

- If the current queuing delay is less than the threshold then congestion window size increased very fast per RTT with Westwood's scalable increase factor.
- If queuing delay is greater than three times of threshold value then congestion window decreased by a number of packets in the buffer at that time (similar to Vegas estimation).
- If queuing delay lies in between one to three times of threshold value congestion window size remain unchanged.

To take a precaution for optimum performance Fusion must be equal to standard Reno, if Fusion congestion window size (w_f) is less than Reno congestion window size (w_r), then w_f is reset to w_r . As well as congestion window reduction factor in fast recovery phase also modified to $\beta = \max(0.5, RTT_{min}/RTT)$. But in Fusion defining threshold is a big challenge and this makes Fusion more complex and undesirable. As our discussion is related to speed and latency issue, Radhakrishnan *et al.* [60] and Grigorik [25] addressed web traffic latency in their TCP Fast Open (TFO) proposal and book High-Performance Browser Networking respectively. While taking care of congestion control care must be taken for over all latency issues due to any reason. So to address this in web application latency and PLT are important factors that influence user satisfaction with a website. Even small improvements in latency lead to noticeable increases in site visits and user satisfaction and result in higher revenue. To achieve the 50% PLT improvement, SPDY (one of the application layer protocol) aimed to make more efficient use of the underlying TCP connection. SPDY has introduced a new binary framing layer to enable request and response multiplexing, prioritization, and header compression. Loading a web page often requires fetching hundreds of resources from dozens of different hosts. In turn, this might require the browser to establish dozens of new TCP connections, each of which will have to incur the overhead of the TCP handshake. Needless to say, this can be a significant source of web browsing latency, especially on slower mobile networks. TFO is a mechanism that aims to eliminate the latency penalty imposed on new TCP connections by allowing data transfer within the

SYN packet. However, it does have its own set of limitations: i) there are limits on the maximum size of the data payload within the SYN packet, ii) only certain types of HTTP requests can be sent, and iii) it works only for repeat connections due to a requirement for a cryptographic cookie [12,14].

In similar fashion based on a reduced handshake, Iyengar and Swett from Google suggested QUIC protocol in their Internet draft [8,33,54]. QUIC protocol is a new multiplexed and secure transport atop UDP. In both TFO and QUIC once connection gets establish between sender and receiver for next every transmission direct data transmission takes place within a time out. To improve bandwidth utilization and fast data delivery UDP protocol is the option, whereas for reliable service TCP through connection establishment services like handshake makes network slow and adds extra overhead. To overcome TCP disadvantage by preserving reliability property QUIC is a new multiplexed and secure transport layer protocol (there is difference of opinions about QUIC location in the protocol stack in some of the literature it appears as an application layer protocol) designed by Google group. QUIC is on top of UDP based protocol designed from the ground up and optimized for HTTP/2 semantics [11,24,28,76].

5. Flow level congestion control

A flow level congestion can be defined as when number of flows of mean size σ arrive at a link of capacity, C at mean rate λ and the link load $\rho > 1$, as the number of flows in progress increase then their throughput tends to zero since the arrival rate λ is greater than the average rate of flow completion. Figure 10 illustrate the behavior of flow level congestion control.

There is relatively limited literature related to flow level congestion control work is available. However, it's worth to do survey on flow level congestion control as this is an important aspect in network performance and QoS. Barkat *et al.* [4] designed traffic model for uncongested IP backbone links. In this they have modeled traffic at flow level by using Poisson shot-noise process. This model shows very good approximation with respect real backbone traffic. The applicability of this model is investigated with network dimensioning and provisioning, prediction of the total rate and generation of backbone traffic. Oueslati and Roberts [56] proposed Flow Aware Networking (FAN) in this admission control [41] and scheduling is applied on user defined flows. They claimed QoS and performance guarantee with cost effective solution as FAN. The FAN supports only for the cross-protect enabled routers. The cross-protect routers improve performance very slowly for network with flooded links.

A study of congestion at flow level using statical bandwidth sharing has been carried out by Fredj *et al.* [21]. In this packet level simulations are used to extract the properties of TCP, whereas theory of stochastic network has been proposed to trace the observations. They observed that if the sessions are appearing as per Poisson process then the throughput is insensitive to both the flow size distribution and the flow arrival process. In this author has demonstrated that fluid flow statistical bandwidth sharing models can accurately predict the results of

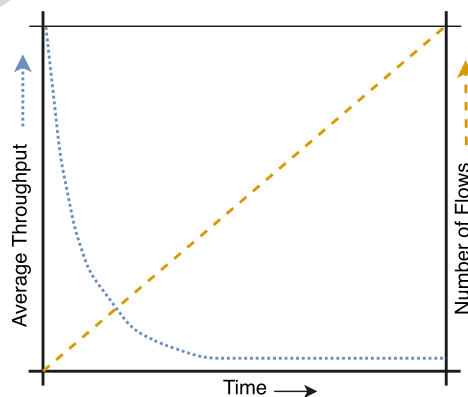


Fig. 10. Network throughput performance for shared bandwidth during transient overload in terms of number of flows.

ns packet-level simulations and also claimed that in fair network distribution of number flows in progress and the expected flow have very simple relation which is valid for wide range real traffic. Sukhov *et al.* [70] used Gaussian approximation to locate the working area of a link based on its utilization and vigilant operators indicating overload points. These indicators they have used to upgrade capacity to avoid congestion at flow level. To validate hypothesis they have prepared testbed using boarder gateway routers and tested for wide range of link utilization.

6. Conclusion

The data networks are scalable and used to accommodate more and more users in wide spread. These users are generating huge amount of data traffic and responsibility of network administrator is to manage this traffic. One of the important task among all responsibilities of network administrator is to manage data traffic without network congestion. Congestion controlling proposals discussed in this survey are limited to standard, lossy network and high-speed-long-delay data network. The congestion control at flow level is an another focused area of research which is helpful to improve overall network performance and QoS. There is no universal solution available to address all problems. Internet of Things is a technological development in which all devices/things will come into the network so surely there is a scope to handle further more congestion issues. In this survey, we have not addressed large (elephant), medium (cat) and small (mice) size live traffic handling situations. In above discussed proposals, all three are treated same but this is not the fact and need to handle separately otherwise Head of Line (HoL) blocking like situations will arise.

Fairness is most of the time decided with Jain's fairness index. Really, is this sufficient to check fair share of resources?, we need to check. Jain's fairness index calculation is very simple control system model of n flows sharing the same link and receiving the same feedback signal. It can well describe the static properties of competing flows. However, the characteristics of the new network architectures and environments where dynamic. The heterogeneity is more if new or different traffic characteristic cannot be well captured in all aspects by that model.

Still, a bufferbloat problem is unsolved in which processing delay is more than TCP connection time. In bufferbloat problem queuing delay is more than RTT in TCP, due to this TCP connection shows congestion in the network, so to avoid this need to use appropriate size RAM at a different level of networks. To solve this active and passive queue management techniques are suggested. Standard buffer size suggested is equal to BDP but if long-delay networks this is not the feasible solution. The cost of the buffer depends on buffer size more the size more need to pay. Industry is demanding solution to optimize the buffer size in such a way that they can pay less. Congestion control in Data Center Networks (DCN) is new opportunity to contribute, in this different congestion controlling approaches were need to consider. In DCN we can contribute to solve problems like TCP incast, queue buildup, buffer pressure, TCP outcast etc.

References

- [1] P.A.K. Acharya et al., Rate adaptation in congested wireless networks through real-time measurements, *IEEE Transactions on Mobile Computing* **9**(11) (2010), 1535–1550. doi:[10.1109/TMC.2010.108](https://doi.org/10.1109/TMC.2010.108).
- [2] A. Afanasyev et al., Host-to-host congestion control for TCP, *IEEE Communications Surveys & Tutorials* **12**(3) (2010), 304–342.
- [3] H. Balakrishnan et al., A comparison of mechanisms for improving TCP performance over wireless links, *IEEE/ACM Transactions on Networking* **5**(6) (1997), 756–769.
- [4] C. Barakat, P. Thiran, G. Iannaccone, C. Diot and P. Owezarski, A flow-based model for Internet backbone traffic, *IEEE Transactions on Signal Processing – Special Issue on Signal Processing in Networking* **51**(8) (2003), 2111–2124. doi:[10.1109/TSP.2003.814521](https://doi.org/10.1109/TSP.2003.814521).
- [5] S. Biaz and N.H. Vaidya, “De-randomizing” congestion losses to improve TCP performance over wired-wireless networks, *IEEE/ACM Transactions on Networking* **13**(3) (2005), 596–608. doi:[10.1109/TNET.2005.850205](https://doi.org/10.1109/TNET.2005.850205).
- [6] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, J. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz and M. Welzl, Reducing Internet latency: A survey of techniques and their merits, *IEEE Communications Surveys & Tutorials* **18**(3) (2016), 2149–2196. doi:[10.1109/COMST.2014.2375213](https://doi.org/10.1109/COMST.2014.2375213).
- [7] C. Caini and R. Firrincieli, TCP Hybla: A TCP enhancement for heterogeneous networks, *International Journal of Satellite Communications and Networking* **22**(5) (2004), 547–566.

- [8] G. Carlucci, L. De Cicco and S. Mascolo, HTTP over UDP: An experimental investigation of QUIC, in: *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Ser. SAC'15*, 2015, pp. 609–614. doi:10.1145/2695664.2695706.
- [9] B.-J. Chang and Y.-H. Li, Cross-layer-based adaptive TCP algorithm in 4G packet service LTE-advanced relaying communications, in: *Advances in Intelligent Systems and Applications – Volume 1: Proceedings of the International Computer Symposium ICS 2012*, Springer Science & Business Media, 2012.
- [10] D. Chiu and R. Jain, Analysis of the increase and decrease algorithms for congestion avoidance in computer networks, *Computer Networks and ISDN Systems* **17**(1) (1989), 1–14.
- [11] Chromium, QUIC, a multiplexed stream transport over UDP, [Online]. Available: <https://www.chromium.org/quic>.
- [12] L. De Cicco, G. Carlucci and S. Mascolo, Understanding the dynamic behavior of the Google congestion control for Rtcweb, in: *Packet Video Workshop (PV), IEEE 20th International*, 2013, pp. 1–8.
- [13] A. Eryilmaz and Srikant, Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control, *IEEE/ACM Transactions on Networking (TON)* **15**(6) (2007), 1333–1344. doi:10.1109/TNET.2007.897944.
- [14] T. Flach, N. Dukkipati, A. Terzis, B. Raghavan, N. Cardwell, Y. Cheng, A. Jain, S. Hao, E. Katz-Bassett and R. Govindan, Reducing web latency: The virtue of gentle aggression, *ACM SIGCOMM Computer Communication Review* **43**(4) (2013), 159–170.
- [15] S. Floyd, TCP and explicit congestion notification, *ACM SIGCOMM Computer Communication Review* **24**(5) (1994), 8–23.
- [16] S. Floyd, RFC3649 HighSpeed TCP for large congestion windows, RFC, 2003.
- [17] S. Floyd, High speed TCP and quick-start for fast long-distance high speed TCP and quick-start for fast long distance networks, Slides, TSVWG, IETF, 2003.
- [18] S. Floyd and T. Henderson, RFC 2582 the NewReno modification to TCP's fast recovery algorithm, RFC, 1999.
- [19] S. Floyd, T. Henderson and A. Gurtov, RFC3782 the NewReno modification to TCP's fast recovery algorithm, RFC, 2004.
- [20] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, RFC 2883 an extension to the selective acknowledgement (SACK), RFC, 2000.
- [21] B. Fred, T. Bonald, A. Proutiere, G. Regnie and J.W. Roberts, Statistical bandwidth sharing: A study of congestion at flow level, *ACM SIGCOMM Computer Communication Review* **31**(4) (2001), 111–122. doi:10.1145/964723.383068.
- [22] B. Fu, Y. Xiao, H. Deng and H. Zeng, A survey of cross-layer designs in wireless networks, *IEEE Communications Surveys & Tutorials* **16**(1) (2014), 110–126. doi:10.1109/SURV.2013.081313.00231.
- [23] C.P. Fu and S.C. Liew, TCP enhancement for transmission over wireless access networks, *IEEE Journal on Selected Areas in Communications* **21**(2) (2003), 216–228. doi:10.1109/JSAC.2002.807336.
- [24] A. Gizis, Taking Google's QUIC for a test drive, 2016. Retrieved: Feb. 2016. [Online]. Available: <http://www.connectify.me/taking-googlequic-for-a-test-drive/>.
- [25] I. Grigorik, *High Performance Browser Networking*, O'Reilly Media, 2013.
- [26] X. Guan, B. Yang, C. Long and Z. Liu, Congestion control based on optimal problem for long range dependence network, *Journal of High Speed Networks* **14**(3) (2005), 215–226.
- [27] S. Ha, I. Rhee and L. Xu, CUBIC: A new TCP-friendly high-speed TCP variant, *ACM SIGOPS Operating Systems Review* **42**(5) (2008), 64–74.
- [28] R. Hamilton, J. Iyengar, I. Swett and A. Wilk, QUIC: A UDP based secure and reliable transport for HTTP/2, Internet Draft (Informational), 2016.
- [29] A. Hanbali, E. Altman and P. Nain, A survey of TCP over ad hoc networks, *IEEE Communications Surveys & Tutorials* **7**(1–4) (2005), 22–36. doi:10.1109/COMST.2005.1610548.
- [30] G. Hasegawa and M. Murata, Survey on fairness issues in TCP congestion control mechanisms, *IEICE Transactions on Communications* **84**(6) (2001), 1461–1472.
- [31] M. Hassan and R. Jain, *High Performance TCP/IP Networking: Concepts, Issues and Solutions*, Prentice-Hall, 2003.
- [32] M. Ikeda, E. Kulla, M. Hiyama, L. Barolli and M. Takizawa, Investigation of TCP and UDP multiple-flow traffic in wireless mobile ad-hoc networks, *Journal of High Speed Networks* **19**(2) (2013), 129–145.
- [33] J. Iyengar and I. Swett, QUIC loss recovery and congestion control draft-tsvwg-quic-loss-recovery-01, IETF, draft-tsvwg-quic-protocol-01, 2015.
- [34] V. Jacobson, Congestion avoidance and control, *ACM SIGCOMM* **18**(4) (1988), 314–329.
- [35] V. Jacobson, Modified TCP congestion avoidance algorithm, email to the end2end list, April 1990.
- [36] C. Jin, D. Wei, S. Low, G. Buhrmaster, J. Bunn, D. Choe, R. Cottrel, J. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot and S. Singh, FAST TCP: From theory to experiments, *IEEE Network* **19**(1) (2005), 4–11.
- [37] C. Jin, D. Wei, S. Low, G. Buhrmaster, J. Bunn, D. Choe, R. Cottrel, J. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot and S. Singh, FAST TCP: From theory to experiments, December 2003.
- [38] S. Kai et al., Receiver-assisted congestion control to achieve high throughput in lossy wireless networks, *IEEE Transactions on Nuclear Science* **57**(2) (2010), 491–496. doi:10.1109/TNS.2009.2037622.
- [39] K. Kaneko, T. Fujikawa, Z. Su and J. Katto, TCP-Fusion: A hybrid congestion control algorithm for high-speed networks, in: *Proceeding PFLDnet*, Vol. 7, 2007, pp. 31–36.
- [40] Kelly, Scalable TCP: Improving performance in highspeed wide area networks, *Review* **33**(2) (2003), 83–91.
- [41] A. Kortebi, S. Queslati and J. Roberts, Cross-protect: Implicit service differentiation and admission control, in: *Proceedings of HPSR'04*, Phoenix, 2004.

- [42] M. Kwon and S. Fahmy, On TCP reaction to explicit congestion notification, *Journal of High Speed Networks* **13**(2) (2004), 123–138.
- [43] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide and F. Jahanian, Internet inter-domain traffic, *ACM SIGCOMM Computer Communication Review* **40**(4) (2010), 75–86.
- [44] L.H.A. Lachlan, S.V. Hanly and R. Mukhtar, Active queue management for fair resource allocation in wireless networks, *IEEE Transactions on Mobile Computing* **7**(2) (2008), 231–246. doi:10.1109/TMC.2007.70724.
- [45] D. Leith, H-TCP: TCP congestion control for high bandwidth delay product, IETF Internet Draft, 2008.
- [46] D. Leith and R. Shorten, H-TCP: TCP for high-speed and long-distance networks, in: *Proceedings of PFLDnet*, 2004.
- [47] K.C. Leung, V.O. Li and D. Yang, An overview of packet reordering in Transmission Control Protocol (TCP): Problems, solutions, and challenges, *IEEE Transactions on Parallel and Distributed Systems* **18**(4) (2007).
- [48] T.K. Liu and J.A. Silvester, Congestion control policies for wireless multimedia CDMA networks, *Journal of High Speed Networks* **8**(2) (1999), 135–147.
- [49] C. Lochert, B. Scheuermann and M. Mauve, A survey on congestion control for mobile ad hoc networks, *Wiley Wireless Communications and Mobile Computing* **7**(5) (2007), 655–676. doi:10.1002/wcm.524.
- [50] R. Ludwig and A. Gurtov, RFC 4015 the Eifel response algorithm for TCP, RFC, 2005.
- [51] R. Ludwig and R.H. Katz, The Eifel algorithm: Making TCP robust against spurious retransmissions, *SIGCOMM Computer Communication Review* **30**(1) (2000), 30–36. doi:10.1145/505688.505692.
- [52] J. Ma, J. Ruutu and J. Wu, An enhanced TCP mechanism-fast-TCP in IP networks with wireless links, *Wireless Networks* **6**(5) (2000), 375–379.
- [53] M. Mathis, J. Mahdavi, S. Floyd and A. Romanov, RFC2018 TCP selective acknowledgment options, RFC, 1996.
- [54] P. Megyesi, Z. Kramer and S. Molnar, How quick is QUIC?, in: *IEEE International Conference on Communications (ICC)*, 2016.
- [55] S. Molnar, B. Sonkoly and T.A. Trinh, A comprehensive TCP fairness analysis in high speed networks, *Computer Communications* **32**(13) (2009), 1460–1484.
- [56] S. Oueslati and J. Roberts, A new direction for quality of service: Flow aware networking, *Next Generation Internet Networks* (2005), 226–232.
- [57] C. Parsa and J.J. Garcia-Luna-Aceves, Improving TCP congestion control over internets with heterogeneous transmission media, in: *IEEE Proceedings, 7th International Conference on Network Protocols (ICNP'99)*, 1999, pp. 213–221.
- [58] L.L. Peterson and B.S. Davie, *Computer Networks: A Systems Approach*, Elsevier, 2007.
- [59] Q.V. Pham and W.J. Hwang, Network utility maximization-based congestion control over wireless networks: A survey and potential directives, *IEEE Communications Surveys & Tutorials* **19**(2) (2017), 1173–1200. doi:10.1109/COMST.2016.2619485.
- [60] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain and B. Raghavan, TCP Fast Open, in: *Proceeding of the 7th Conference on Emerging Networking Experiments and Technologies*, ACM, 2011, p. 21.
- [61] M.A. Raja, The network layer, <https://www.slideshare.net/madilraja/the-network-layer-38487922>.
- [62] R. Ramani and A. Karandikar, Explicit congestion notification (ECN) in TCP over wireless network, in: *IEEE International Conference on Personal Wireless Communications*, 2000, pp. 495–499.
- [63] H.K. Rath A. Sahoo and A. Karandikar, A cross layer congestion control algorithm in wireless networks for TCP Reno-2, in: *Proceedings NCC*, 2006.
- [64] K. Ratnam and I. Matta, WTCP: An efficient mechanism for improving TCP performance over wireless links, in: *3rd IEEE Symposium on Computers and Communications (ISCC'98) Proceedings*, 1998, pp. 74–78.
- [65] L. Reiner et al., Optimizing the end-to-end performance of reliable flows over wireless links, *Wireless Networks* **8**(2/3) (2002), 289–299.
- [66] N.K.G. Samaraweera, Non-congestion packet loss detection for TCP error recovery using wireless links, *IEEE Proceedings – Communications* **146**(4) (1999), 222–230. doi:10.1049/ip-com:19990597.
- [67] M. Saverio et al., TCP Westwood: Bandwidth estimation for enhanced transport over wireless links, in: *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ACM, 2001.
- [68] H. Shimonishi and T. Murase, Improving efficiency-friendliness tradeoffs of TCP congestion control algorithm, in: *Global Telecommunications Conference, GLOBECOM'05*, Vol. 1, 2005, p. 5.
- [69] J. Sing and B. Soh, TCP New Vegas: Improving the performance of TCP Vegas over high latency links, in: *Fourth IEEE International Symposium on Network Computing and Applications*, 2005, pp. 73–82. doi:10.1109/NCA.2005.52.
- [70] A.M. Sukhov, D.I. Sidelnikov, A.P. Platonov, M.V. Strizhov and A.A. Galtsev, Active flows in diagnostic of troubleshooting on backbone links, *Journal of High Speed Networks* **18**(1) (2011), 69–81.
- [71] E. van den Berg, I. Sebuktekin, M. Tauil, A. Ghetie and S. Alexander, Self adaptive robust resource allocation for prioritized TCP flows in wireless networks, in: *IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, 2015, pp. 108–113.
- [72] H. Wang and K.G. Shin, Robust TCP congestion recovery, *Journal of High Speed Networks* **13**(2) (2004), 103–121.
- [73] R. Wang, M. Valla, M. Sanadidi, B. Ng and M. Gerla, Efficiency/friendliness tradeoffs in TCP Westwood, in: *Proceeding 7th International Symposium on Computers and Communications*, 2002, pp. 304–311.
- [74] R. Wang, M. Valla, M. Sanadidi, B. Ng and M. Gerla, Adaptive bandwidth share estimation in TCP Westwood, in: *Proceeding IEEE GLOBECOM*, Vol. 3, 2002, pp. 2604–2608.

- [75] D.X. Wei, C. Jin, S.H. Low and S. Hegde, FAST TCP: Motivation, architecture, algorithms, performance, *IEEE/ACM Transactions on Networking* **14**(6) (2006), 1246–1259. doi:[10.1109/TNET.2006.886335](https://doi.org/10.1109/TNET.2006.886335).
- [76] S. Wei and V. Swaminathan, Low latency live video streaming over HTTP 2.0, in: *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop, Ser. NOSSDAV'14*, ACM, New York, NY, USA, 2014, pp. 37:37–37:42.
- [77] J. Widmer, R. Denda and M. Mauve, A survey on TCP-friendly congestion control, *IEEE Network* **15**(3) (2001), 28–37.
- [78] K. Xu, Y. Tian and N. Ansari, TCP-Jersey for wireless IP communications, *IEEE Journal on Selected Areas in Communications* **22**(4) (2004), 747–756. doi:[10.1109/JSAC.2004.825989](https://doi.org/10.1109/JSAC.2004.825989).
- [79] L. Xu, K. Harfoush and I. Rhee, Binary increase congestion control (BIC) for fast long-distance networks, in: *INFOCOM 2004. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 4, 2004.

AUTHOR COPY