Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SURVEY OF CONGESTION DETECTION AND AVOIDANCE TECHNIQUES FOR TCP CONGESTION CONTROL

[1]LEE CHIN KHO, [2]XAVIER DÉFAGO, [3]YASUO TAN, [4]YUTO LIM
School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa-ken, 923-1292, JAPAN
E-mail: [1]s1120203@jaist.ac.jp, [2]defago@jaist.ac.jp, [3]ytan@jaist.ac.jp, [4]ylim@jaist.ac.jp

ABSTRACT

With the emergence of novel network applications, the non-stop growing traffic is starting to experience unexpected scenarios of network congestion. This paper comprehensively reviews congestion control techniques of the mostly used TCP variants. Those variants have been developed over a period of three decades. New classifications are made for those variants based on device entity, characteristic, and association. The survey also describes the parameters that are used in each variant and how the parameters are used to update of TCP congestion window size. Furthermore, congestion detection and avoidance techniques that are implemented in those variants are also investigated and classified. Among the contribution of this survey, the majority of variants studied here were found to be sender-centric.

Keywords: *TCP, Congestion Control, Congestion Detection, Congestion Avoidance, Device Entity*

1. INTRODUCTION

Internet [1] appears to be a great success for information sharing among people, at the same time Internet also faces numerous unsolved problems. Network congestion [2] is one of the unending problems that is depending on insufficient capacity of the underlying sub-network for the demanded amount of data transfer which is rapidly increasing. This growth of demand eventually might go beyond the Internet's capacity and Service Providers' ability to efficiently deliver the huge Internet traffic. As a result, Internet might tend to face tremendous and unpredictable network congestion. Network congestion might be avoided if technologies can keep increasing networking bandwidths, fast enough and never stop increasing. Still imperfect routing sometimes fails to balance the traffic. This causes congestion in the over utilized portions of the network.

When incoming traffic demand to some sub-network (node or link) exceeds the capacity of that sub-network (buffers and output capacity), congestion starts from that point and spreads along its links. Data crossing that sub-network would suffers from prolonged delays (buffer waiting) eventually leading to timeouts (loss rate).Internet has controlled network congestion for almost three decades to some degree. Still, Internet users sometimes experience poor performance due to congestion. Therefore, researchers are still trying to find better alternative solutions to eradicate congestion completely without sacrificing utilization.

Transmission control protocol (TCP) the core protocol of Internet provides a reliable delivery of a data stream in between two communicating devices. To achieve a degree of reliable data transmission under the congested network, different congestion control variants were designed. The first congestion control design (variant) was introduced by Van Jacobson in 1988 using end-to-end congestion control mechanisms. In his design, there are four congestion control stages: slow start, congestion avoidance, fast retransmit and fast recovery. Together with the techniques handling these four stages, they form the basic TCP flow and congestion control. Later, other TCP variants were designed to increase network throughput not just for congestion but also for other network conditions. These TCP variants include features like early congestion detection, available bandwidth detection, and loss type estimation. With these features, a sender can detect congestion state, buffer utilization, loss events and route condition changes to ensure network resources are fully utilized.

Recent research includes TCP/Network Coding [3], history-based TCP throughput prediction [4], and neural networks model in TCP throughput estimation [5]. Such research investigates communication networks, as fifth generation mobile networks, machine-to-machine (M2M) networks, and Internet of Things (IoT) networks, network coding, prediction, machine learning, neural networks, optimization theory and so on.

The vast development of TCP variants lead to many surveys that have been conducted by researchers. Those surveys can be generally classified into three different perspectives; performance in general, performance in certain network environments, and with specific parameters.

The first perspective, TCP general performance compare between different TCP variants. Westwood, NewReno, and Vegas were surveyed by L. A. Grieco et al. [6]. Similarly, H. Jamal et al. [7] compared a standard TCP Reno to various variants and also classified the variants into loss-based, delay-based, and mixed loss-delay based.

The second perspective, many surveys studied how different TCP variants perform in different network environments. A. A. Hanbali et al. [8] have discussed wireless issues and the major factors involved in TCP congestion control over mobile ad hoc networks (MANET).Similarly, H. Balakrishnan et al. [9] compared protocol categories of end-to-end, link-layer, and split-connection of TCP congestion control over wireless networks. K. S. Reddy et al. [10] delved into high-speed network, considering TCP variants such as BIC, CUBIC, FAST TCP, HSTCP, Layered TCP, STCP, and XCP.

The last perspective, A. Afanasyev et al. [1] collected and described a comprehensive set of TCP variants and mechanisms that optimize various parameters in different network environments. The survey also explained each TCP variant, its strengths and weaknesses.

Unlike previous surveys, this survey compares existing TCP variants with respect to the controlling device entity, characteristic and association. This can give a quick view of the types of TCP variants and their relationship. Association of the variants is important here because it allows the researches to understand the evolution and the importance of congestion control in current and future networks. In addition, this survey discusses TCP congestion detection in terms of the evolutionary chain and compares the variants according to the device of detection such as duplicated acknowledgement (ACK) and round trip time (RTT). Packet loss for duplicated ACK, and delay for RTT are used as indicator to detect the congestion in the network. Last, we consider TCP congestion avoidance (CA) techniques in terms of dependency on congestion window size (*cwnd*). This allows us to know how the TCP variants work in controlling the transmission rate to avoid congestion.

In our previous work [11], we classified nearly thirty existing TCP variants based on the criteria of the controlling device entity (sender, receiver, both), the variants characteristics (complexity degree, congestion problem, and features), and the network environment (wired, wireless, high-speed/long-delay, and low priority of data transfer). In this survey, we extend our classification and associate TCP variants and also classify congestion detection and avoidance techniques based on their detection scope and *cwnd* respectively.

The objective of this survey is to review the major end-to-end TCP variants. TCP variants from 1988 to 2014 are collected, classified and analyzed. The variants considered in this research are the same set considered in all the previous surveys [1-10] as a representative set for all TCP variants.

The rest of this survey is organized as follows. Section 2 discusses comprehensively TCP variants by looking at three different classifications; device entity, characteristic, and association. Section 3 and 4 explain the congestion detection and avoidance techniques of TCP variants, respectively. At last, this survey is concluded in Section 5.

2. CONGESTION CONTROL OF TCP VARIANTS

In this section, TCP variants is discussed according to three different perspectives: *device entity*, *characteristic* and *association*.

2.1. Device Entity

TCP variants can be classified into four types according to the controlling device entity, i.e., *sender*, *receiver*, *sender-or-receiver*, and *sender-and- receiver*. The classification of the studied TCP variants among the four types, is shown in the top part of Table 1.

Sender category sometimes referred to as sender-centric protocol (SCP). In SCP, the sender performs essential tasks such as reliable data transfer; whereas the receiver only needs to transmit feedback packets in the form of acknowledgement to the sender [12]. The data transfer in between the sender and the receiver in SCP is also referred as data-acknowledgement message exchange. Upon receiving these feedback information, the sender tunes the Congestion

Window (*cwnd*) based on a window based mechanism to ensure the number of transmission bytes does not exceed network capacity.

The idea of having the congestion and flow controlled at the receiver side was introduced in 1997 [1],[12-15]. This way is also called receiver-centric control protocol (RCP). The RCP uses the same window based mechanism similar to SCP, but data-acknowledgement message exchange is no longer applicable. RCP uses request-data message exchanges for data transfer, in which a receiver sends an explicit request packet to the sender for requesting the data packets to be sent. This way, the sender only can transmit its data packets according to the transmission rate that is requested by the receiver. The receiver uses the incoming data packet as an acknowledgement to its previous request for data. According to [16], TCP performance can be significantly improved by using the RCP approach.

Unlike SCP and RCP, Y. Shu et al. [20] have proposed an alternative approach, called a mobile-host-centric transport protocol (MCP). In MCP, the device of control can function as either SCP or RCP at a specific period of time. If the mobile station is the sender, then the data-acknowledgement message exchange is adopted. When the mobile station is the receiver, the request-data message exchange is applied. In particular, when the mobile station is the receiver, a flag *m_flag* of a synchronization packet is set to one to notify other senders to use receiver-centric control. Otherwise, when the mobile station is the sender, it sets *m_flag* to be 0 and then sender-centric control is adopted.

Another type of controlling device entity is known as hybrid centric protocol (HCP), which was introduced by K. Shi et al. [25]. TCP mechanisms are coordinately controlled by both the sender and the receiver at the same time. For example, the receiver performs the function of flow control and participates in congestion control by computing *cwnd*. Then, the sender uses the receiver's information to adjust the size of *cwnd*. In HCP, TCP congestion control can reduce the waiting time of the sender to alleviate the impact of timeout problem of MCP [25].

2.2. Characteristic

To elaborate the characteristics of the TCP variants in this section, three subsections; feature, complexity degree, and network domain are used. The classification of TCP variants based on their characteristics is showed in Table 2.

2.2.1. Feature

Except for Tahoe, all of the TCP variants have four fundamental algorithms of TCP congestion control. Originally Tahoe had only 3 of those algorithms. V. Jacobson has refined Tahoe by adding the fast recovery algorithm and called the new TCP as Reno in 1990 [1]. New Reno is the enhancement of Reno. Reno unable to control network congestion efficiently because it focuses on detecting the packet loss of a single TCP transmission. It can't detect multiple packet loss. Variants which also cannot handle multiple packet loss includes Tahoe, Reno, and Vegas. This is due to the fact that those variants avoid congestion by using the feedback information (ACKs), which is referred to as primary feedback. To mitigate this problem, New Reno and its extension variants include a new feature, called multiple losses handling to detect loss of more than one packet at a time. Multiple losses handling uses partial feedback information or extended feedback information. The estimation/prediction feature is added to some TCP variants to predict congestion status. The estimation parameters include bandwidth, transmission rate, and queue delay. Table 2 shows in the second part, that most of the TCP variants studied include both features; multiple losses handling and estimation/prediction.

2.2.2. Complexity degree

Complexity degree defines in this paper as a difficulty in terms of implementation complexity of the TCP variant. High complexity referred to TCP variants using; the core four algorithms plus both the estimation and multiple loss detection features. While medium complexity lack the estimator feature. The low complexity is restricted to variants with only the core algorithms. As we can observe from Table 2, more than half of the TCP variants studied have a high complexity degree.

2.2.3. Network domain

In this section, TCP variants are distributed into four overlapping network domains, namely *wired network*, *wireless network*, *high-speed/long delay network*, and *low priority of data transfer network* [1]. TCP is originally designed for wired network whereby the network congestion only occurs due to the packet loss. Therefore, TCP for wired network cannot react adequately to the packet loss of the wireless network. This is because the packet loss of the wireless network is mainly due to the lossy nature of radio links. Several solutions have been proposed to resolve this problem, such as the network status is verified by the explicit congestion notification from the

Table 1: The Characteristics Of TCP Variants And Distribution Among The Different Types Of Device Entity

Characteristics	Tahoe	Reno	Vegas	SACK	TD-FR	NewReno	DSACK	Westwood	Veno	TCPW CRB	Nice	LP	TCP-Real [15]	DOOR	TCPW BR	STCP	HS-TCP	FAST TCP	TFRC	BIC	Hybla	Vegas A	Casablance [19]	New Vegas	Africa	CTCP	MCP	Illinois	Fusion	CUBIC	Libra [21]	TCP-FIT [22]	CTCP [23]	Fwestwood [24]
Centric Control Protocol																																		
SCP	▨	▨	▨	▨		▨		▨	▨	▨	▨	▨			▨	▨	▨	▨		▨	▨	▨	▨	▨	▨	▨		▨	▨	▨	▨	▨		▨
RCP					▨		▨						▨																				▨	
MCP																											▨							
HCP														▨					▨															
Feature																																		
Slow start	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Congestion avoidance	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Fast retransmit	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
Fast recovery		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
ACK Feedback	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Multiple loss handling				▨	▨	▨	▨	▨	▨	▨		▨		▨	▨	▨	▨			▨	▨		▨		▨	▨		▨	▨	▨	▨			
Estimation technique			▨					▨	▨	▨	▨		▨		▨	▨		▨	▨			▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨	▨
Complexity Degree																																		
Low	▨	▨																															▨	▨
Medium			▨	▨	▨	▨	▨				▨	▨	▨	▨		▨		▨	▨	▨	▨	▨		▨			▨			▨		▨		
High								▨	▨	▨					▨		▨						▨		▨	▨		▨	▨		▨			
Network Domain																																		
Wired	▨	▨	▨	▨		▨			▨													▨												▨
Wireless					▨		▨	▨		▨			▨	▨	▨				▨				▨				▨					▨		
High-speed/ long delay [1]																▨	▨	▨		▨	▨			▨	▨	▨		▨	▨	▨	▨	▨	▨	
Low priority data transfer [1]											▨	▨																						

▨ denotes a selected column None mentioned references are all [1]

congested intermediate wireless terminal. Other solution uses intermediate wireless terminal to separate the cause of packet loss whether wireless or wired environments.

In addition, standard TCP variants may not be very efficient in high-speed/long-delay networks. In this kind of networks, the data packets are transmitted but not yet received at the receiving side due to the long distance end-to-end transmission. Those data packets shouldn't be treated as a real packet loss. This problem is also referred to as a bandwidth-delay product (BDP).

A low priority data transfer network is the network which consists of high and low priorities data flows. Both TCP-Nice and TCP-LP are designed to provide a guarantee of transmission rate for the low priority data flows even though in the presence of the high priority data flows.

In Table 1, the TCP variants with high complexity degree are often belonging to the network domain of wireless networks and high-speed with long-delay networks.

2.3. Association

The association of TCP variants is depicted in Fig. 1. Tahoe was the first introduced TCP variant with congestion control. However, the core three algorithms of Tahoe reduce *cwnd* to 1 when packet loss is detected which can lead to significant throughput degradation. Fast recovery (FR) algorithm was introduced in Reno to halve *cwnd* and hold the new value until no duplicated acknowledgements are received within a specific period of time. Reno evolved into five different TCP variants that are specifically targeted for wired and wireless environments. The first two, NewReno and SACK solved multiple losses in the wired environment by introducing the partial ACK and selective ACK respectively. The third, TCP-Real implement contention detection and congestion avoidance in wireless environment. The four, Vegas uses the queue length measurement to determine the buffer utilization to determine the congestion status and reacts proactively. Vegas quantify congestion status before an actual congestion using estimation, then uses packet delay to update *cwnd*. The last one, TCP- FIT performs gracefully in both wireless and high speed/long delay network by parallel TCP technique.

A few TCP variants are extended from Vegas. For example, New Vegas included rapid window convergence algorithm to reduce the convergence time and increase the network utilization of the high-speed/long-delay environment. FAST TCP is a scalable TCP variant of Vegas, but it defines a periodic fixed rate *cwnd* and a delay-based congestion estimation. In order to solve the problem of improper decrement of flow rate to nearly zero under certain condition in Vegas, Vegas A proposed an additional adaptive buffer mechanism. The threshold coefficient in Vegas A is adaptively tuned according to the actual

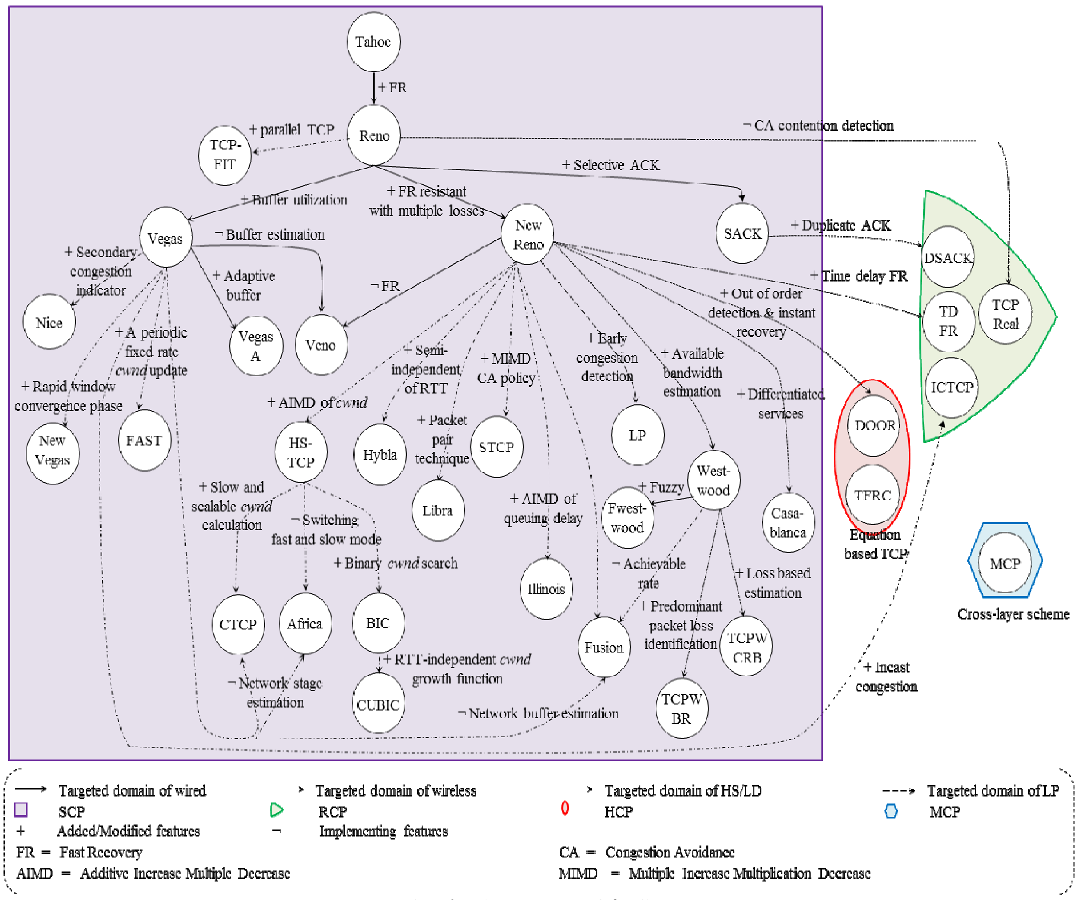
Tahoe
+ FR
Reno
TCP-FIT
+ parallel TCP
− CA contention detection
+ Selective ACK
+ Buffer utilization
+ FR resistant with multiple losses
Vegas
− Buffer estimation
New Reno
SACK
+ Duplicate ACK
DSACK
+ Secondary congestion indicator
+ Adaptive buffer
− FR
+ Time delay FR
TD FR
TCP Real
Nice
Vegas A
Veno
+ Out of order detection & instant recovery
+ A periodic fixed rate cwnd update
+ Semi-independent of RTT
+ MIMD CA policy
+ Early congestion detection
+ Available bandwidth estimation
ICTCP
+ Rapid window convergence phase
+ AIMD of cwnd
+ Differentiated services
DOOR
New Vegas
FAST
HS-TCP
Hybla
+ Packet pair technique
STCP
LP
West-wood
+ Slow and scalable cwnd calculation
− Switching fast and slow mode
Libra
+ AIMD of queuing delay
+ Fuzzy
Fwest-wood
Casa-blanca
TFRC
Equation based TCP
MCP
Cross-layer scheme
+ Binary cwnd search
− Achievable rate
+ Loss based estimation
Illinois
CTCP
Africa
BIC
+ Predominant packet loss identification
Fusion
TCPW CRB
+ Incast congestion
− Network stage estimation
+ RTT-independent cwnd growth function
CUBIC
− Network buffer estimation
TCPW BR
Targeted domain of wired
SCP
+ Added/Modified features
FR = Fast Recovery
AIMD = Additive Increase Multiple Decrease
Targeted domain of wireless
RCP
− Implementing features
Targeted domain of HS/LD
HCP
CA = Congestion Avoidance
MIMD = Multiple Increase Multiplication Decrease
Targeted domain of LP
MCP

Figure 1: The Association Of TCP Variants

transmission rate, and *cwnd* management algorithm is also adopted. Nice targets low priority data transfer domain considers all standard TCP variants. 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. ICTCP designed to handle the problem of incast congestion by adjust the TCP receiver window before packet loss happen.

One of the Reno family members, NewReno has been extended to a large number of new TCP variants for all kinds of network environments HS-TCP, Hybla, Libra, STCP, Illinois, Fusion, CTCP, Africa, BIC, and CUBIC are the enhanced NewReno versions for high-speed/long-delay. The enhanced features include AIMD of *cwnd* and queuing delay, semi-independent of RTT, packet pair technique, MIMD congestion avoidance policy, binary *cwnd* search, and so on.

Besides the extension of NewReno to high-speed/long-delay environments, NewReno also expands for wireless environment, for instance Westwood, Casablanca, DOOR, and TD-FR. In Westwood, a bandwidth estimation algorithm is added to speed up the fast recovery stage. This bandwidth estimation algorithm has an inherent concept of Vegas, whereby the RTT and the amount of transmitted data packets are used to calculate the data transfer rate. 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, three TCP variants, i.e., TCPW BR, TCPW CRB and Fwestwood are evolved from the Westwood. TCPW BR and TCPW CRB add the features of the predominant packet loss identification and the loss based estimation in the Westwood's congestion control algorithm, respectively. Fwestwood implements the fuzzy controller approach to enhance performance in wired network with high error rate.

On the other side of Westwood, Casablanca for example adds the feature of differentiated service (*Diffserv*) to enable a sender to identify accurately and react properly to deal with the packet loss due to the medium contention and the interference effect. Unlike Casablanca, DOOR includes the out of order detection and instant recovery and TD-FR uses time delayed fast recovery algorithm to improve the packet reordering problem.

A number of TCP variants is developed by merging the concepts and ideas from different TCP variants together. For instance, CTCP is constructed by implementing the network stage estimation of Vegas and adding slow and scalable *cwnd* calculation of HS-TCP. Meanwhile, Africa is built by implementing the network stage estimation of Vegas and the switching fast and slow mode of HS-TCP. Fusion is another integration of TCP variants from the Vegas, NewReno, and the Westwood. In the Fusion, the features of network buffer estimation from Vegas and achievable rate from Westwood are implemented. Fusion also maintains NewReno's congestion control mechanism.

Other TCP variants like DSACK, which is the extension of SACK for the wireless environment and RCP approach. DSACK requires the receiver to acknowledge each receipt of a duplicate packet to the sender in order to avoid the problem of misinterprets out of order delivery data packet problem. Two TCP variants do not come from the association root of the standard TCP variants, i.e., MCP and TFRC. MCP uses cross-layer scheme, whereas TFRC uses different way of congestion control mechanism by fixing the transmitting size of the data packets, but TFRC triggers data sending rate in terms of packet per second in response to network congestion status.

In overall, Table and Fig. 1 shows that nearly 79% of the 34 TCP variants are SCP, and only one TCP variant is MCP. Furthermore, about 41% of the 34 TCP variants are for high-speed/long-delay environments, about 32% is for wireless environments, and the rest of percentage is for wired and low priority data transfer environments. In the association of TCP variants, most of the TCP variants of high-speed/long-delay environment evolved from the TCP variants of the wired environments.

3. CONGESTION DETECTION OF TCP VARIANTS

The congestion detection plays an important role in TCP congestion control mechanism. In this section, the scope of the congestion detection of TCP variants can be classified into two fundamental categories, namely *loss based* (L) and *delay based* (D). Fig. 2 shows the evolutionary chain that exhibits the scope of the congestion detection in TCP variants. Loss based is

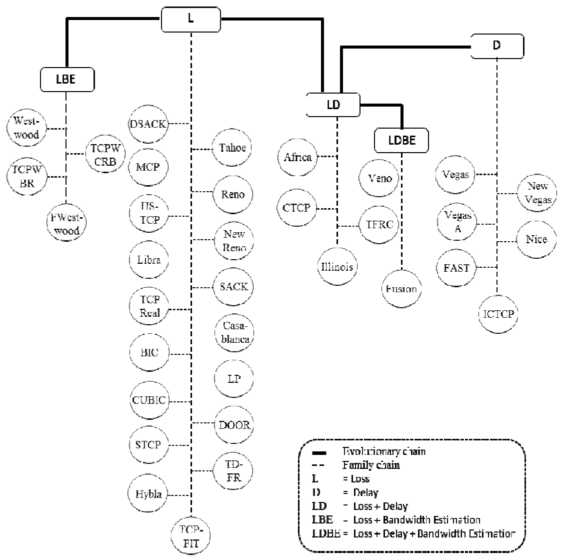

Figure 2: A Evolutionary Chain Of TCP Variants That Shows The Detection Scope Of The Congestion Control

the earliest reactive method used in the congestion control to detect the congestion network status. Loss based is originally built to amend the reliable end-to-end transmission at the transport protocol to deal with the congestion problem. In the normal operation of TCP protocol, the receiver transmits an ACK message to the corresponding sender upon receiving the data packet successfully. When the sender receives three duplicate ACK messages consecutively from the same receiver, this indicates that the transporting network is congested [1],[17].

The method used in the loss based tends to face the long delay of the sending ACK message. As a result, the loss based method cannot detect the congestion problem properly. To overcome this problem, the delay based that works as a proactive method is introduced by the Vegas. In the delay based, the network parameter, i.e., RTT that is used to measure the average end-to-end delay depicts the time required for sender to transport the data packets to reach the destination. RTT parameter can be also interpreted as a parameter to know the network congestion status. Through this RTT, *cwnd* is calculated as a function that is varying with the difference of RTT to reflect to the network congestion status.

Unlike RTT parameter, some TCP variants use an estimation technique to predict the bandwidth usage of the end-to-end transmission. This technique is called a bandwidth estimation technique, which is applied in conjunction with loss based estimation (LBE). The bandwidth can be estimated when the instantaneous bandwidth is calculated upon reception of the ACK message. In particular, the amount of acknowledged data by the ACK message is divided by the time elapsed since the last ACK message received. The TCP variants that fall under this detection scope are Westwood, the TCPW BR, and the TCPW CRB.

On the other hand, a *hybrid based* that is defined as the combination of both loss based and delay based (LD) is also introduced right after the LBE. Duplication ACK or RTT parameter cannot solely determine the network congestion status. Therefore, hybrid based was developed to overcome the weakness of either the loss based or the delay based. In hybrid based, TCP variants use the delay based to identify the network status of the connection. When the network is congestion is suspected, congestion is further confirmed by using the loss based method. Veno is a typical example of the hybrid based. However, TCP Fusion combines hybrid based and bandwidth (LDBE) estimation to determine the congestion status.

So far, the detection scope of the TCP variants is discussed. Selecting an appropriate parameter is an essential to justify the network congestion status. In this paper, the list of the parameters that used in the congestion detection of TCP variants is shown in Table 2.

The detection parameters that use in each congestion detection scopes are mainly selected based on several factors, which includes network structure, traffic pattern, transmission rate, network application, QoS requirements, and congestion probability [26]. In this paper, the congestion parameters are further classified into the device of detection, detection scope, and locality and globality. The device of detection describes a terminal that involves in detecting the congestion parameters. The terminal can be a sender, a receiver, or an intermediate. From the Table 2, most of the congestion parameters are performed at the sender side. Only three congestion parameters are detected by the receiver or the intermediate. We also can find out that most of the congestion parameters are used for the delay based method and estimation technique. As we can observe from Table 2, only the parameter of the internal queue length is local. Local means that a parameter is measured within a device and uses for the purpose of itself to justify the congestion status. This allows the faster response to the congested node, but it needs the per-flow state information at local node, which limited its scalability. Whereas global means that a parameter is measured from the sender, the receiver, and the intermediate. The global parameter that is shared among the network devices is used to identify the congestion status of a network. This will result in congestion control reacts slowly to the congestion. The congestion information in the intermediate nodes has to travel towards the receiver and then backward to the sender. Then, only the sender can adjust the transmission rate to release the congested networks.

In summary, several of detection parameters are used in congestion status determination, but this determinability might sometimes not accurate. This is because the congestion detection parameters such as duplication ACK, RTT, bandwidth and packet loss rate are mainly based on the feedback of acknowledgement packets. The acknowledgement packets can be lost or delayed due to non-congestion factors during the transmission. As a result, the congestion status might be misestimated and results in unnecessary congestion control handling or missing.

Table 2: A Comparison Of Parameters That Are Used In The Congestion Detection Of TCP Variants

Parameter	Device of Detection	Detection Scope	Location
Duplication ACK	Sender	L, LD, LBE, LDBE	global
Round trip time (RTT)	Sender	D, LD, LBE, LDBE	global
Retransmission timeout (RTO)	Sender	L, D, LD, LBE, LDBE	global
Bandwidth	Sender and intermediate	LBE, LDBE	global
Packet loss rate	Sender	L, D, LD, LBE, LDBE	global
Internal queue length	intermediate	L, D, LD, LBE, LDBE	Local
Inter packet arrival time	Receiver and intermediate	D, LD, LBE, LDBE	global
Retransmission time of packet	Sender	L, D, LD, LBE, LDBE	Global
Average delay	Receiver	D, LD, LDBE	Global
Jitter	Receiver	D, LD, LDBE	Global

4. CONGESTION AVOIDANCE OF TCP VARIANTS

In this section, CA algorithm of the TCP variants is discussed. In CA algorithm, the *cwnd* size is the main component parameter to be controlled and monitored. Generally, *cwnd* starts with exponentially increase in slow start stage. When the network congestion is detected, TCP congestion control will enter into the CA stage, in which the *cwnd* is reduced and slowly increased according to the additive or multiplicative way. Except for Tahoe, all the TCP variants in the CA stage immediately reduce *cwnd* to half then increase *cwnd* based on the its own policy either additive increase (AI) or multiplicative increase (MI). Tahoe directly set the *cwnd* size to zero when three duplicated ACK is received.

Table 3: The Dependency Of Previous Cwnd Size When The CA Algorithm Of TCP Variants Is Used

Dependency of Old *cwnd* Size	Function, f (*cwnd*)	
	w/ Condition	w/o Condition
Direct Dependent	Vegas, Nice, Vegas A, Veno, New Vegas, STCP, Africa, TCP-Real, FAST TCP, CTCP, Hybla, IIlinois, Libra, Fusion, Westwood (Mode 1), TCPW, BR (Mode 1), TCPW CRB (Mode 1), ICTCP	Tahoe, Reno, NewReno, SACK, LP, Casablanca, HS-TCP, TD-FR, DS ACK, DOOR, MCP, TCP-FIT
Indirect Dependent	Westwood (Mode 2), TCPW BR (Mode 2), TCPW CRB (Mode 2), FWestwood	None

The mathematical representation of AI and MI used in increment the *cwnd* size in CA algorithm is offered in this section. The main effect of the method uses in increment *cwnd* size will impact the network throughput, while avoiding the network returns back to congestion in the short period of time.

Table 3 classifies the TCP variants based on the effect of their previous *cwnd* size when CA algorithm is used. Dependency in Table 3 refers to the function of *cwnd* that depends on the old *cwnd* or not in order to decide the new *cwnd* size in the CA algorithm. The condition refers to the function of *cwnd* being bounded by any constraints or not. From the Table 3, we can observe that most of the functions of *cwnd* depend on the old *cwnd*. For example, Tahoe starts to update the new *cwnd* size without condition based on $cwnd_{new} = cwnd_{old} + 1/cwnd_{old}$ when the packet loss is detected. But, Vegas will need to ensure the product of $\chi = cwnd_{old} \times RTT\text{-}RTT_{min}/RTT$ is fulfilled the condition in order to enter the CA stage. If χ is positive, the cwnd is decrease by one-eighth; if χ is negative or zero, $cwnd_{new} = cwnd_{old} + 1$. Meanwhile, only Westwood, TCPW BR, and TCPW CRB are using other parameters to determine the new *cwnd* with condition in their CA algorithms.

Table 4 shows the methods of AI, MI, and equation-based of the function, *f*(*cwnd*). Some TCP variants consist of two modes in their CA algorithms, e.g., Westwood, TCPW BR, TCPW CRB, TCP-Real, Africa, CTCP, and Fwestwood. These two modes are operating in CA algorithms depend 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 is totally depending on the old *cwnd* size. And β in the additive increase is depending on the zero, constant, scaling, quotient, and other. If β is zero, the new *cwnd* is equal to the old *cwnd*. If β is constant, the new *cwnd* can be a function that varies with a fixed value, such as Libra, Vegas, Vegas A, New Vegas, and Nice. If β is scaling, e.g., $\gamma \times cwnd_{old}$, the new *cwnd* can be a function that varies with a factor. For instance, γ = 0.125 in STCP and γ = 0.01 in Africa (Mode 1). If the β is equal to a quotient, e.g., $\omega/cwnd_{old}$, the new *cwnd* size can be a function that varies with ω, in which can be 1. If β is otherwise, the new *cwnd* size can be a function that varies with an estimation value. On the other hand, α in the multiplicative increase can be divided into two types, i.e., bandwidth estimation (BE) and rate estimation (RE). But, these two types of multiplicative increase share the common β of RTT_{min}. An alternative method that is found in Table 5 is an equation-based method. In this equation-based method, the new *cwnd* is determined by a new form equation, which depends on the maximum or minimum limit of *cwnd* size and the computed end-to-end throughput.

In summary, the function, *f*(*cwnd*) of CA algorithms are mainly depends on network structure, traffic pattern, type of application, and QoS requirement. In other words, the multiplicative increase policy of new *cwnd* can recover the network throughput faster than additive increase policy. However, the multiplicative increase policy easily causes the network to be congested again.

Table 4: The Methods Of Additive Increase, Multiplicative Increase, And Equation-Based Of The Function, F(Cwnd)

Method	*f*(*cwnd*)		TCPVariant
	α	*β*	
Additive Increase(AI) $cwnd_{new}= \alpha+\beta$	*cwnd*	Zero	Westwood(Mode 1), TCPW BR(Mode 1), TCPW CRB (Mode 1), TCP-Real*(Mode1), FWeastwood* (Mode 1)
	cwnd	Constant	Vegas, Vegas A, New Vegas, Nice, FAST TCP, Fusion(Mode 2), ICTCP *, TCP-FIT (Mode 1)
	cwnd	Scaling	STCP,Africa,(Mode 2), CTCP(MODE 1), TCP-FIT(Mode 2)
	cwnd	Quotient	Tahoe, Reno, NewReno, SACK, HS-TCP, TD-FR*, DSACK*, Veno, DOOR***, Hybla, Libra, Africa(Mode 1), TCP-Real*(Mode 2), CTCP(Mode 2), Fusion(Mode 3), MCP**
	cwnd	Other	IIlinois
Multiplicative Increase(MI) $cwnd_{new}=\alpha\times\beta$	BE	RTT_{min}	Westwood(Mode 1), TCPW BR(Mode 2), Fwestwood(Mode 2),
	BE	RTT_{min}	TCPW CRB (Mode 2)
Equation-based	*cwnd*		BIC,CUBIC, TRFC***

w/o footnote mark = SCP, *=RCP, **=MCP, ***=HCP

5. 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. In other words, congestion occurs when the available bandwidth is not enough for the requested amount of data. This survey extensively reviewed TCP congestion control variants with their congestion detection and avoidance mechanisms. This survey shed the light on the future possible inadequacy of the aforementioned TCP variants to handle the communication demand of the ever growing traffic.

We have summarized the association of TCP variants shows that most of the TCP variants are biased toward the SCP method. It is the type to which the first deployed variant belongs. Later it was thoroughly extended and modified more than the other variant. This can mainly be attributed to the fact that it existed longer than the other variants. Future TCP variants might better focus on extending the latest combined SCP and RCP methods, this can facilitate fine-tuning according to the different network environments. The parameters that are used in the congestion detection stage are limited in a number and some of them are highly dependent on one or more of the others. The network congestion detection accuracy achieved using those parameters have only been slightly improved during the last few decades. The majority of the well-known TCP variants only modify the pre-existing equations mostly with different coefficients.

In summary, this survey has reviewed the classifications and association of TCP variants in depth. This survey also further investigated congestion detection and avoidance techniques. The survey can provide valuable hints for network research, to know how the congestion detection and avoidance techniques are associated.

Previous surveys were too deeply detailed that easily loses the main idea. Other surveys focus only on one particular network condition or topology.

Unlike previous surveys, this survey focuses on the concepts of congestion control techniques more than the details. That helps getting the overall picture without being indulged into details that is not necessary in this state.

The association between the variants is also shown in this research as time relation of the evolution of variants. The bigger cluster of the variants shows clearly that NewReno (during the last 11 years) is still by far the variant of choice by most researchers and similarly network designers. Similarly Loss based detection is the choice of most TCP variants designers and standards organizations.

Once a clear idea of congestion techniques is reached according to this survey, through comparison with simulations or networking logs of real traffic network is due. Future variants are recommended to start with NewReno again as the base for modification with its loss based detection. Finding some of the good modifications in other variants and add them back, would come out with new more suitable variants.

REFRENCES:

[1] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, “Host-to-host congestion control for TCP,” IEEE Commun. Surveys and Tutorials, vol. 12, no. 3, pp. 304--342, May 2010.

[2] M. Allman, “TCP Congestion Control,” RFC5681, Sept. 2009.

[3] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, “Network coding meets TCP: theory and implementation,” Proc. IEEE, vol. 99, no. 3, pp. 490--512, Mar. 2011.

[4] S. Vazhkudai, J. Wu, and I. Foster, “Predicting the performance of wide area data transfer,” Proc. IEEE Int. Parallel and Distributed Processing Symp., Ft. Lauderale, FL, Apr. 2002. DOI:10.1109/IPDPS.2002.1015510.

[5] S. M. H. Shah, A. U. Rehman, A. N. Khan, and M. A. Shah, “TCP throughput estimation: A new neural networks model,” Proc. Int. Conf. on Emerging Tech., Islamabad, pp. 94--98, Nov. 2007.

[6] 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.

[7] 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.

[8] 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.

[9] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katsz, “ A comparison of mechanisms for improving TCP performance over wireless link,” IEEE/ACM Trans. Netw., vol 5, no. 6, pp. 756--769, Dec. 1997.

[10] 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.

[11] K. L. Chin, X. Defago, Y. Tan, and A. O. Lim, "A taxonomy of congestion control techniques for TCP in wired and wireless networks," Proc. IEEE Symp. on Wireless Tech. and Applications, Kuching, Malaysia, pp 22--25, Sept. 2013.

[12] H. Y. Hsieh, K. H. Kim, Y. Zhu, and R. Sivakumar, "A receiver centric transport protocol for mobile hosts with heterogeneous wireless interfaces," Proc. MobiCom 2003, New York, USA, pp. 1--15, Sept. 2003.

[13] R. Gupta, M. Chen, S. McCanne, and J. Walrand, "A receiver-driven transport protocol for the web," Telecommun. Syst., vol. 21, no. 2--4, pp. 213-230, Aug. 2002.

[14] P. Mehra, C. Vleeschouwer and A. Zakhor, "Receiver-driven bandwidth sharing for TCP and its application to video streaming," IEEE Trans. on Multimedia, vol. 7, no. 4, pp. 740--752, Aug. 2005.

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

[16] N. T. Spring, M. Chesire, M. Berryman, V. Sahasranaman, T. Anderson, and B. Bershad, "Receiver based management of low bandwidth access links," Proc. INFOCOM 2000, Tel-Aviv, Isael, vol. 1, pp. 245--254, Mar. 2000.

[17] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in 4.3 - Tahoe BSD TCP congestion control," ACM Comp. Commun. Rev., vol. 22, no. 2, pp. 9--16, Apr. 1992.

[18] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," Proc. INFOCOM 2004, Hong Kong, vol. 4, pp. 2514--2524, Mar. 2004.

[19] S. Biaz, and N. H. Vaidya, "Differentiated service: a new direction for distinguishing congestion losses from wireless losses," Tech. Rep., Auburn University, pp.1--15, Feb. 2003.

[20] Y. Shu, W. Ge, N. Jiang, Y. Kang, and J. Luo, "Mobile host centric transport protocol for EAST experiment," Proc. 15th IEEE-NPSS Real Time Conf., Betavia, IL, pp. 1--8, Apr. 2007.

[21] G. Marfia, C. Palazzi, G. Pau, M. Gerla, M. Sanadidi, and M. Roccetti, "TCP Libra: deriavation, analysis, and comparison with other RTT-fair TCPs," J. of Comp. Netw., vol. 54, no. 14, pp. 2327--2344, Oct. 2010

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

[23] H. Wu, Z. Feng, and Y. Zhang, "ICTCP: Incast congestion control for TCP in data-center networks," IEEE/ACM Trans. on Networking, vol. 21, no. 2, pp. 345--358, Apr. 2013.

[24] Z. T. Alisa, and S. R. Qasim, "A fuzzy based TCP congestion control for wired networks," Int. J. of Comp. Application, vol. 89, no. 4, pp. 36--42, Mar. 2014.

[25] K. Shi, Y. Shu, O. Yang, and J. Luo, "Receiver assistant congestion control in high speed and lossy networks," Proc. 16th IEEE-NPSS Real Time Conf., Beijing, China, pp. 91--95, May 2009.

[26] H. Sheikhi, M. Dashti, and M. Dehghan, "Congestion detection for video traffic in wireless sensor networks," Proc. Int. Conf. on Consumer Electronics, Commun., and Netw., Xian Ning, China, pp. 1127--1131, Apr. 2011