Journal of Computer and Electronic Sciences

Journal of Computer and Electronic Sciences
Available online at jces.blue-ap.org
©2015 JCES Journal. Vol. 1(1), pp. 22-31, 28 February, 2015
E-ISSN: 2311-6730

# A Review and Provide a New Evaluation Method on the Hybrid Congestion Control for High-Speed Networks

**Mir Mahmoud Talebi[1*] and Shahram Jamali[1,2]**

1- Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran
2- Assistant Professor, Computer Engineering Department, University Mohaghegh Ardabili, Ardabil, Iran

**Corresponding Author:** Mir Mahmoud Talebi

*Received: 09 January, 2015* *Accepted: 28 January, 2015* *Published: 28 February, 2015*

**ABSTRACT**

In recent years, the majority of internet users use the Transmission Control Protocol (TCP) for fast transfer of big data over networks, and improvement of the TCP congestion control algorithm is an ideal solution to this challenge. In the last decade, many solutions; based on packet loss, based on RTT delay and the hybrid methods are presented, past research results show that the hybrid approaches have better efficiency of available bandwidth, more robustness, and better performance in Fairness, TCP-friendly. Several hybrid algorithms are presented for congestion control in high-speed networks. This paper emphasizes on the study of the behavior of these protocols for a range of parameters under heterogeneous networking environments, and also under various network conditions. We tried to analysis the strengths and weaknesses suggested hybrid protocol solutions for high-speed networks as a special study with uses a new evaluation index called Stability Index (SI).

***Keywords:*** *TCP, Hybrid congestion control, High-speed networks, Stability Index.*
©2014 JAAS Journal All rights reserved.

## INTRODUCTION

Congestion control is an important issue in a transport protocol for High-speed Networks (Stallings, 2004) that refer to the networks supporting with high data rates; from few Mbps to Gbps, like high speed Ethernet. In these networks to achieve a larger degree of performance from throughput, RTT fairness, TCP-friendly, utilization, etc. The additive increase and multiplicative decrease phase (AIMD) of the TCP protocol should be revised. Normal TCP, in AIMD phase, increases congestion window size by one packet per round trip time (RTT) and decreases it by half when a loss is detected. This approach leads to poor performance and decreased efficiency in high-speed networks, and effectively for full efficiency of the available bandwidth is inefficient, to overcome this problem, most proposed solutions, such as High-speed TCP (HSTCP, for short) (Floyd, 2003), Scalable TCP(Kelly, 2003), early version of TCP-Westwood (TCPW), BIC (Xu et al., 2004), CUBIC(Ha et al., 2008) alter the rate of increase and/or decrease based purely on loss information. Other groups, such as TCP-Vegas (Brakmo and Peterson, 1995) and FAST-TCP (Wei et al., 2006) adjust the window on the estimated queuing delay. In contrast them, several interesting proposals, such as TCP Africa(King et al., 2005), Compound TCP (CTCP) (Tan, Song, Zhang, and Sridharan, 2006), TCP-Illinois (Liu et al., 2008)and Hybrid Congestion Control TCP (HCC TCP) (Xu, , 2011) seek to combine both. This paper focuses on hybrid methods, and evaluates the performance of four such algorithms: TCP Africa, Compound TCP, TCP-Illinois and HCC TCP.Since the hybrid methods are made of two components; based on packet loss and delay, therefore, in section 2 introduces the overall

structure of the protocol described, in sections 3 and 4, mechanisms of loss-based component and delay-based component on each of the algorithms will be analyzed with new stability index.

***Overall Architecture***

In the hybrid methods, that uses synergistic benefits of both loss-based and delay-based approaches, many of the weaknesses of each of these methods are covered, however, they also have disadvantages. Mechanisms and how to use each of these algorithms from the packet loss data and delay information is different. In this section, the overall architecture of each of the algorithms is studied.

***Compound TCP (CTCP)***

Compound TCP (CTCP) algorithm is developed and presented by Microsoft on Windows Vista and Windows Server 2008.This algorithm domination property to set congestion window is designed for high-bandwidth networks, while not harm to fairness and uses the estimated queuing delay to prevent congestion.CTCP algorithm to determine the size of the sliding congestion window $W_s$ Uses the sum of the two congestion window: a regular AIMD window as loss-based congestion window $cwnd$, similar to TCP Reno acts and a delay-based window $dwnd$. If the delay time is small, as shown in Figure 1, CTCP increases delay-based congestion window rapidly to improve the utilization of the available bandwidth. Once delay time is long (queuing is experienced), the delay-based congestion window gradually decreases to compensate for the increase in the loss-based congestion window. The goal is to keep their sum approximately constant. CTCP is a parallel hybrid approach, which uses delay-based and loss-based components simultaneously to determine of the congestion window size and avoidance of network congestion.

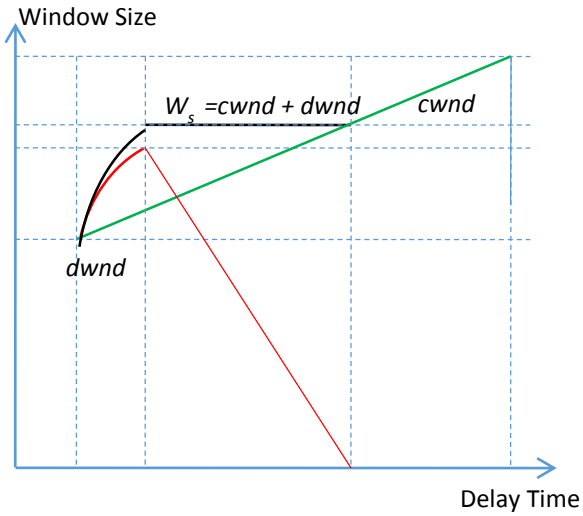



Figure 1. CTCP window evolution.

$$win = \min\ (cwnd + dwnd, awnd)$$

Where:

$$dwnd(t+1) = \begin{cases} dwnd(t) + (\alpha.win(t)^k - 1)\ if\ diff < \gamma\ link: free \\ (dwnd(t) - \varepsilon.diff)\ if\ diff \geq \gamma\ link: congestion \\ \left(win(t).(1-\beta) - \frac{cwnd}{2}\right)\ if\ loss\ is\ detected \end{cases}$$

$$diff = (expected - actual).base_RTT$$

$$expected = \frac{win}{base_RTT} \qquad actual = \frac{win}{RTT}$$

In normal case:

$$cwnd = cwnd + \frac{1}{(cwnd + dwnd)} \; for \; each \; ACK$$

In congestion avoidance case:

$$win(t+1) = win(t) + \alpha . win(t)^k \; for \; each \; ACK$$

In packet loss case:

$$win(t+1) = win(t).(1-\beta)$$

***TCP-Illinois***

TCP-Illinois algorithm, actually is a flexible version of standard TCP, which uses the queuing delay information to increase the window size *W* by *α/W* for each acknowledgment as a delay-based component, and uses the packet loss information to decrease the window size *W* by *β. W* for each loss event as a loss-based component, that the α and β values are updated once per RTT. Unlike the standard TCP, as shown in Figure 2, α and β are not fixed but depends on the functions of average queuing delay. TCP-Illinois is a switching approach, which, given the state of congestion, to one of the delay-based or loss-based component switches.

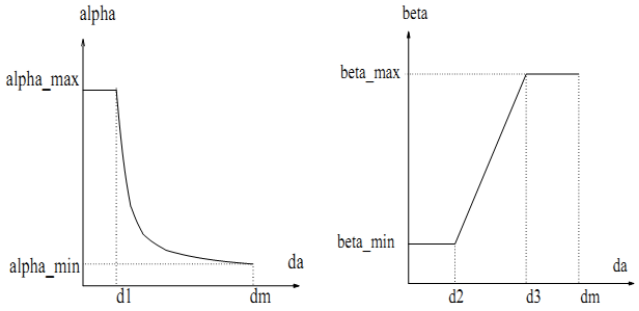



Figure 2. α and β curves Vs da.

When receiving an ACK:

$$W = W + \frac{\alpha}{W}$$

$$\alpha = f_1(d_a) = \begin{cases} \alpha_{max} d_\alpha \le d_1 \\ \dfrac{k_1}{k_2 + d_a} \; otherwise \end{cases}$$

When detect a loss:

$$W = (1-\beta).W$$

$$\beta = f_2(d_a) = \begin{cases} \beta_{min} d_a \le d_2 \\ k_3 + k_4 d_a d_2 < d_a < d_3 \\ \beta_{max} \; otherwise \end{cases}$$

When a timeout has happened:

1) $SS_{threshold} = \frac{W}{2}, \alpha = 1, \beta = \frac{1}{2}$

2)

3) Enters slow start phase, and α and β values are unchangeduntil one RTT after the slow start phase.

Where:

$$k_1 = \frac{(d_m - d_1)\alpha_{min}\alpha_{max}}{\alpha_{max} - \alpha_{min}} \qquad k_2 = \frac{(d_m - d_1)\alpha_{min}}{\alpha_{max} - \alpha_{min}} - d_1$$

$$k_3 = \frac{\beta_{min}d_3 - \beta_{max}d_2}{d_3 - d_2} \qquad k_4 = \frac{\beta_{max} - \beta_{min}}{d_3 - d_2}$$

***Hybrid Congestion Control TCP (HCC TCP)***

As shown in Figure 3, HCC algorithm is composed of three components: delay-based component, loss-based component and joint control. The delay-based estimation component uses the delay information as the primary congestion indicator and determines the congestion measure using queuing delay on the forward path to adjust the window size.The loss-based estimation component uses the packet loss as one bit information as the secondary congestion indicator and determines the congestion measure to adjust the window size. Joint control component of HCC algorithm, using delay and packet loss information for switches to one of the delay-based or loss-based estimation components.Similar to TCP-Illinois, HCC is also a switching approach, which given the state of congestion, to one of the delay-based or loss-based component switches.

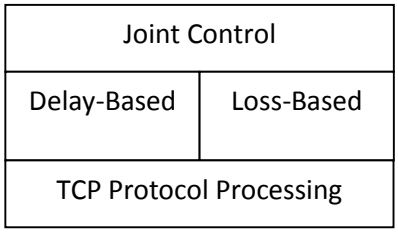



Figure 3. HCC TCP architecture.

***Performance evaluation***

In this section, we use NS-2 (The NS simulator - ns-2, 2011) to evaluate the three protocols of CTCP, TCP-Illinois and HCC by a new performance evaluation index called Stability Index (SI). As shown in Fig. 4 a network topology with bottleneck link shared between multiple source and destination is used. The routers are connected by a link with 200Mbps bandwidth, 98ms delay time and buffer size of 100 packets. Other links between source/destination and routers has been considered with 1Mbps bandwidth and the delay is equal 1ms.

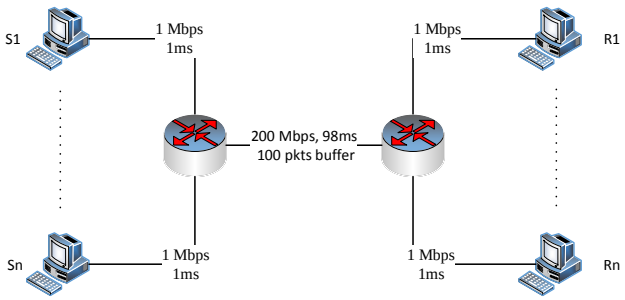



Figure 5. The dumbbell topology with multiple flows

***Stability Index (SI)***

One of the major challenges in network congestion control is stability of behavior proposed the protocol in different states of uses it, such as different variables of buffer size, packet loss, flow numbers and etc. In order to evaluate the stability of

congestion control approaches, we use the concept of variance To calculate the deviation of output results from index value (e.g. Available bandwidth, fairness index, efficiency index and etc.).

Let "I" be the ideal (expected) evaluation index, using the concept of variance and placement "I" instead of the mean we have

$$Var(x) = \frac{\sum_{i=1}^{n}(I - x)^2}{n} \quad \textbf{(1)}$$

Range of changes and maximum/minimum values will be as:

$$\mathrm{Var(x)} = \begin{cases} 0 \text{ where } \mathrm{I = x} \\ \dfrac{\sum_{\mathrm{i=1}}^{\mathrm{n}}(\mathrm{I})^2}{\mathrm{n}} \text{ where } \mathrm{x = 0} \end{cases} \quad \textbf{(2)}$$

Since variance can be from zero to a maximum value, in order to convert it to a standard and well-defined scope, we converts it into a range of 0 and 1.

$$\frac{\mathrm{Var(x)}}{\mathrm{E(x^2)}} = \frac{\frac{\sum_{\mathrm{i=1}}^{\mathrm{n}}(\mathrm{I - x})^2}{\mathrm{n}}}{\frac{\sum_{\mathrm{i=1}}^{\mathrm{n}}\mathrm{x}^2}{\mathrm{n}}} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{n}}(\mathrm{I - x})^2}{\sum_{\mathrm{i=1}}^{\mathrm{n}}\mathrm{x}^2} \quad \textbf{(3)}$$

In Eq. (3), since the resulting function shows a deviation from the ideal state, so if the protocol is not ideal, its value will be zero and at worst, will be an amount equal to 1. But given that, in evaluating the performance of the protocols followed are positive indicators, so instead of Eq. (3), we use Eq. (4).

$$\boldsymbol{SI} = 1 - \frac{\sum_{i=1}^{n}(I - x)^2}{\sum_{i=1}^{n} x^2} \quad \textbf{(4)}$$

***Stability of throughput versus buffer size***

One of the most important issues in evaluating the performance of the TCP congestion control protocols, is the behavior it versus of the buffer size changes.In this section of the simulation, Stability Index for different buffer sizes were calculated for each of the protocols, and results shown in Tables 1,2 and 3 for each protocol of CTCP, TCP-Illinois and HCC TCP, respectively.

Table 1. The Stability Index (SI) of throughput versus buffer size for CTCP

| Buffer Size (pkts) | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 50 | 57 | 20449 | 3249 |
| 100 | 86 | 12996 | 7396 |
| 200 | 114 | 7396 | 12996 |
| 500 | 183 | 289 | 33489 |
| 1000 | 189 | 121 | 35721 |
| 1500 | 191 | 81 | 36481 |
| 2000 | 197 | 9 | 38809 |
| 2500 | 196 | 16 | 38416 |
| 3000 | 194 | 36 | 37636 |
| 3500 | 194 | 36 | 37636 |
| 4000 | 193 | 49 | 37249 |
| 4500 | 194 | 36 | 37636 |
| Sum | | 41514 | 356714 |
| Average | | 3459.5 | 29726.17 |
| Stability Index (SI) | | | 0.88 |

Table 2. The Stability Index (SI) of throughput versus buffer size for TCP-Illinois

| Buffer Size (pkts) | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 50 | 60 | 19600 | 3600 |

| | | | |
|---|---|---|---|
| 100 | 117 | 6889 | 13689 |
| 200 | 125 | 5625 | 15625 |
| 500 | 155 | 2025 | 24025 |
| 1000 | 180 | 400 | 32400 |
| 1500 | 185 | 225 | 34225 |
| 2000 | 195 | 25 | 38025 |
| 2500 | 195 | 25 | 38025 |
| 3000 | 195 | 25 | 38025 |
| 3500 | 195 | 25 | 38025 |
| 4000 | 195 | 25 | 38025 |
| 4500 | 195 | 25 | 38025 |
| Sum | | 34914 | 351714 |
| Average | | 2909.5 | 29309.5 |
| Stability Index (SI) | | | 0.9 |

Table 3. The Stability Index (SI) of throughput versus buffer size for HCC TCP

| Buffer Size (pkts) | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 50 | 100 | 10000 | 10000 |
| 100 | 180 | 400 | 32400 |
| 200 | 183 | 289 | 33489 |
| 500 | 193 | 49 | 37249 |
| 1000 | 195 | 25 | 38025 |
| 1500 | 195 | 25 | 38025 |
| 2000 | 195 | 25 | 38025 |
| 2500 | 195 | 25 | 38025 |
| 3000 | 195 | 25 | 38025 |
| 3500 | 195 | 25 | 38025 |
| 4000 | 195 | 25 | 38025 |
| 4500 | 195 | 25 | 38025 |
| Sum | | 10938 | 417338 |
| Average | | 911.5 | 34778.17 |
| Stability Index (SI) | | | 0.97 |

***Stability of throughput versus packet loss rate***

Since in the hybrid protocols, both the delay and packet loss parameters used for congestion control, How to deal with the detection of packet loss is very important, especially the recovery phase and achieve maximum throughput in high-speed networks. In Tables 4, 5 and 6, the value calculated for the stability of three protocols CTCP, TCP-Illinois and HCC TCP is given.

Table 4. The Stability Index (SI) of throughput versus packet loss rate for CTCP

| Packet Loss Rate | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 0.000001 | 190 | 100 | 36100 |
| 0.000002 | 186 | 196 | 34596 |
| 0.000003 | 181 | 361 | 32761 |
| 0.000004 | 176 | 576 | 30976 |
| 0.000005 | 171 | 841 | 29241 |
| 0.000006 | 166 | 1156 | 27556 |
| 0.000007 | 161 | 1521 | 25921 |
| 0.000008 | 156 | 1936 | 24336 |
| 0.000009 | 151 | 2401 | 22801 |
| 0.00001 | 146 | 2916 | 21316 |
| Sum | | 12004 | 285604 |
| Average | | 1200.4 | 28560.4 |
| Stability Index (SI) | | | 0.96 |

Table 5. The Stability Index (SI) of throughput versus packet loss rate for TCP-Illinois

| Packet Loss Rate | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 0.000001 | 173 | 729 | 29929 |
| 0.000002 | 145 | 3025 | 21025 |
| 0.000003 | 116 | 7056 | 13456 |
| 0.000004 | 110 | 8100 | 12100 |
| 0.000005 | 105 | 9025 | 11025 |
| 0.000006 | 90 | 12100 | 8100 |
| 0.000007 | 85 | 13225 | 7225 |
| 0.000008 | 78 | 14884 | 6084 |

| | | | |
|---|---|---|---|
| 0.000009 | 75 | 15625 | 5625 |
| 0.00001 | 65 | 18225 | 4225 |
| Sum | | 101994 | 118794 |
| Average | | 10199.4 | 11879.4 |
| Stability Index (SI) | | | 0.14 |

Table 6. The Stability Index (SI) of throughput versus packet loss rate for HCC. TCP

| Packet Loss Rate | x: Throughput (Mbps) | (BW-x)^2 | x^2 |
|---|---|---|---|
| 0.000001 | 198 | 4 | 39204 |
| 0.000002 | 194 | 36 | 37636 |
| 0.000003 | 190 | 100 | 36100 |
| 0.000004 | 188 | 144 | 35344 |
| 0.000005 | 183 | 289 | 33489 |
| 0.000006 | 180 | 400 | 32400 |
| 0.000007 | 175 | 625 | 30625 |
| 0.000008 | 173 | 729 | 29929 |
| 0.000009 | 167 | 1089 | 27889 |
| 0.00001 | 160 | 1600 | 25600 |
| Sum | | 5016 | 328216 |
| Average | | 501.6 | 32821.6 |
| Stability Index (SI) | | | 0.98 |

***Stability of efficiency versus flow numbers***

Scalability is one of the main challenge in the design of congestion control protocols, it should not be a significant decrease in throughput when flow numbers are increased, in other words, despite the large number of flows, have a good efficiency and high utilization. Stability of efficiency versus flow numbers is another index that we considered, and to calculate it, let optimum efficiency is equal to 1, thus in Eq. (4) Instead of "I" put 1.

Table 7. The Stability Index (SI) of efficiency versus flow numbers for CTCP

| Flow Numbers | x: Efficiency (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 0.97 | 0.0009 | 0.9409 |
| 2 | 0.96 | 0.0016 | 0.9216 |
| 4 | 0.8 | 0.04 | 0.64 |
| 6 | 0.74 | 0.0676 | 0.5476 |
| 8 | 0.73 | 0.0729 | 0.5329 |
| 10 | 0.69 | 0.0961 | 0.4761 |
| 12 | 0.67 | 0.1089 | 0.4489 |
| 14 | 0.68 | 0.1024 | 0.4624 |
| 16 | 0.69 | 0.0961 | 0.4761 |
| 18 | 0.7 | 0.09 | 0.49 |
| 20 | 0.71 | 0.0841 | 0.5041 |
| 22 | 0.72 | 0.0784 | 0.5184 |
| Sum | | 0.839 | 6.959 |
| Average | | 0.07 | 0.58 |
| Stability Index (SI) | | | 0.88 |

Table 8. The Stability Index (SI) of efficiency versus flow numbers for TCP-Illinois

| Flow Numbers | x: Efficiency (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 0.8 | 0.04 | 0.64 |
| 2 | 0.85 | 0.0225 | 0.7225 |
| 4 | 0.94 | 0.0036 | 0.8836 |
| 6 | 0.95 | 0.0025 | 0.9025 |
| 8 | 0.89 | 0.0121 | 0.7921 |
| 10 | 0.87 | 0.0169 | 0.7569 |
| 12 | 0.57 | 0.1849 | 0.3249 |
| 14 | 0.55 | 0.2025 | 0.3025 |
| 16 | 0.53 | 0.2209 | 0.2809 |
| 18 | 0.52 | 0.2304 | 0.2704 |
| 20 | 0.51 | 0.2401 | 0.2601 |

| | | | |
|---|---|---|---|
| 22 | 0.5 | 0.25 | 0.25 |
| Sum | | 1.4264 | 6.3864 |
| Average | | 0.12 | 0.53 |
| Stability Index (SI) | | | 0.77 |

Table 9. The Stability Index (SI) of efficiency versus flow numbers for HCC TCP

| Flow Numbers | x: Efficiency (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 0.8 | 0.04 | 0.64 |
| 2 | 0.81 | 0.0361 | 0.6561 |
| 4 | 0.81 | 0.0361 | 0.6561 |
| 6 | 0.7 | 0.09 | 0.49 |
| 8 | 0.78 | 0.0484 | 0.6084 |
| 10 | 0.77 | 0.0529 | 0.5929 |
| 12 | 0.72 | 0.0784 | 0.5184 |
| 14 | 0.7 | 0.09 | 0.49 |
| 16 | 0.71 | 0.0841 | 0.5041 |
| 18 | 0.73 | 0.0729 | 0.5329 |
| 20 | 0.75 | 0.0625 | 0.5625 |
| 22 | 0.77 | 0.0529 | 0.5929 |
| Sum | | 0.7443 | 6.8443 |
| Average | | 0.06 | 0.57 |
| Stability Index (SI) | | | 0.89 |

***Stability of fairness versus flow numbers***

Fairness is the latest index that stability it against the increase of flow numbers were investigated, similar "efficiency", variance of fairness index(FI) obtained from Eq. (5) (Jain, Chiu, and Hawe, 1984) to the optimal value "1" by Eq. (4) was calculated.

$$\boldsymbol{FI} = \frac{(\sum_{i=1}^{n} \bar{x}_i)^2}{n.\sum_{i=1}^{n} \bar{x}_i^2} \qquad \textbf{(5)}$$

Where $\bar{x}_i$ is the average throughput of the flow *i*.

Table 10. The Stability Index (SI) of fairness versus flow numbers for CTCP

| Flow Numbers | x: Fairness (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 2 | 0.99 | 0.0001 | 0.9801 |
| 4 | 0.97 | 0.0009 | 0.9409 |
| 6 | 0.98 | 0.0004 | 0.9604 |
| 8 | 0.99 | 0.0001 | 0.9801 |
| 10 | 0.97 | 0.0009 | 0.9409 |
| 12 | 0.83 | 0.0289 | 0.6889 |
| 14 | 0.9 | 0.01 | 0.81 |
| 16 | 0.95 | 0.0025 | 0.9025 |
| 18 | 0.8 | 0.04 | 0.64 |
| 20 | 0.7 | 0.09 | 0.49 |
| 22 | 0.6 | 0.16 | 0.36 |
| Sum | | 0.3338 | 9.6938 |
| Average | | 0.03 | 0.81 |
| Stability Index (SI) | | | 0.96 |

Table 11. The Stability Index (SI) of fairness versus flow numbers for TCP-Illinois

| Flow Numbers | x: Fairness (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 |
| 4 | 0.97 | 0.0009 | 0.9409 |
| 6 | 0.95 | 0.0025 | 0.9025 |
| 8 | 0.92 | 0.0064 | 0.8464 |
| 10 | 0.9 | 0.01 | 0.81 |
| 12 | 0.82 | 0.0324 | 0.6724 |
| 14 | 0.87 | 0.0169 | 0.7569 |
| 16 | 0.96 | 0.0016 | 0.9216 |

| | | | |
|---|---|---|---|
| 18 | 0.77 | 0.0529 | 0.5929 |
| 20 | 0.71 | 0.0841 | 0.5041 |
| 22 | 0.6 | 0.16 | 0.36 |
| Sum | | 0.3677 | 9.3077 |
| Average | | 0.03 | 0.78 |
| Stability Index (SI) | | | 0.96 |

Table 12. The Stability Index (SI) of fairness versus flow numbers for HCC TCP

| Flow Numbers | x: Fairness (0<x<1) | (1-x)^2 | x^2 |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 |
| 4 | 0.99 | 0.0001 | 0.9801 |
| 6 | 0.92 | 0.0064 | 0.8464 |
| 8 | 0.97 | 0.0009 | 0.9409 |
| 10 | 0.95 | 0.0025 | 0.9025 |
| 12 | 0.87 | 0.0169 | 0.7569 |
| 14 | 0.89 | 0.0121 | 0.7921 |
| 16 | 0.96 | 0.0016 | 0.9216 |
| 18 | 0.78 | 0.0484 | 0.6084 |
| 20 | 0.7 | 0.09 | 0.49 |
| 22 | 0.65 | 0.1225 | 0.4225 |
| Sum | | 0.3014 | 9.6614 |
| Average | | 0.03 | 0.81 |
| Stability Index (SI) | | | 0.96 |

***Benchmark of stability index***

According to the calculated values for the four different aspects of stability index (Fairness, Efficiency, throughput versus packet loss and throughput versus buffer size) for protocols, Fig. 5 as a benchmark for the three protocols CTCP, TCP-Illinois and HCC TCP provided.

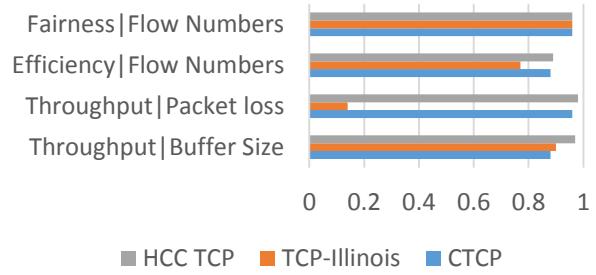



Figure 5. Benchmark of Stability Index (SI)

As shown in Figure 5, the stability of fairness versus the flow numbers, the three protocols are in good performance.For the efficiency evaluation, can be seen that the TCP-Illinois weaker than the other two protocols, although the performance all of three protocols are not ideal.TCP-Illinois are extremely sensitive to packet loss and has a poor performance, this implies a fundamental flaw in the loss-based component of TCP-Illinois hybrid algorithm.Can be seen that the HCC TCP protocol has good performance, and sensitivity of HCC TCP versus the buffer size is less than the other two protocols.As a whole as shown in Figure 5, the HCC TCP at stability index is efficient.

## CONCLUSION

In this paper, we have presented a new evaluation method to create a benchmark list for TCP congestion control algorithms, since the stability of protocols in achieve full throughput, utilization, fairness and etc. is an important issue, thus we have defined a new function called Stability Index (SI) for evaluate it that use concept of variance for calculate the deviation of evaluate index from the optimum value.

## REFERENCES

Brakmo L and Peterson L. 1995. TCP Vegas: end to end congestion avoidance on a global internet. *I EEE Journal on Selected Areas in Communications, 13 (8)*, 1465– 1480.

Cui, T and Andrew L. 2013. *FAST TCP simulator module for ns-2. Version 1.1*. Retrieved January 14, 2015, from CUBINlab: http://www.cubinlab.ee.mu.oz.au/ns2fasttcp.

Floyd S. 2003. High-speed TCP for large congestion windows. *RFC editor*.

Ha S, Rhee I and Xu L. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *Operating Systems Review (ACM), 42*, 64–74.

Jain R, Chiu D and Hawe W. 1984. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. DEC Research Report TR-301.

Kelly T. 2003. Scalable TCP: improving performance in high speed wide area networks. *Computer Communication Review, 33(2)*, 83–91.

King R, Baraniuk R and Riedi R. 2005. TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. 3*, pp. 1838-1848. IEEE. doi:10.1109/INFCOM.2005.1498463

Liu S, Basar T and Srikant R. 2008. TCP -Illinois: a loss- and delay-based congestion control algorithm f or high-speed networks. *Performance Evaluation, 65(6–7)*, 417-440.

Stallings W. 2004. High-Speed Networks and Internets Performance and Quality of Service. In *High-Speed Networks and Internets Performance and Quality of Service* (Vol. 2). Pearson Education.

Tan K, Song J, Zhang Q and Sridharan M. 2006. A compound TCP approach for high-speed and long distance networks. *Proceeding of IEEE INFOCOM*.

*The NS simulator - ns-2*. 2011. Retrieved January 14, 2015, from USC/ISI: http://www.isi.edu/nsnam/ns/

Wei D, Jin C, Low S and Hegde S. 2006. FAST TCP. Motivation, architecture.

Xu L, Harfoush K and Rhee I. 2004. Binary increase congestion control (BIC) for fast long-distance networks. *Proceedings of IEEE INFOCOM*.

Xu W, Zhou Z, Pham D, Ji C, Yang M and Liu Q. 2011. Hybrid congestion control for high-speed networks. *Journal of Network and Computer Applications, 34*, 1416-1428.