

Simple Model Analysis and Performance Tuning of Hybrid TCP Congestion Control

Jiro Katto, Kazumine Ogura, Yuki Akae, Tomoki Fujikawa, Kazumi Kaneko and Zhou Su

Department of Computer Science, Waseda University.
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan.

E-Mail: {katto, ogura, akae, tomoki, kaneko}@katto.comm.waseda.ac.jp, zhou@asagi.waseda.jp

Abstract—This paper presents simple analytical models of hybrid TCP congestion controls, which switch loss-based mode and delay-based mode adaptively, and tries their performance tuning. We firstly present ideal behavior models of three kinds of TCP congestion controls (loss-based, delay-based and hybrid). We then give abstracted models of the actual hybrid TCPs and consider their performance tuning. Finally, experiments validate analytical expectations and effectiveness of the hybrid TCP.

I. INTRODUCTION

It is well known that TCP-Reno [1] and its siblings [2][3] are widely deployed in the current Internet. However, it has been revealed that performance of TCP-Reno families deteriorates in high-speed networks (and in wireless networks). This is primary due to its congestion control, whose congestion window (cwnd) is increased by one Maximum Segment Size (MSS) every RTT and halved upon packet losses.

To overcome this problem, a number of TCP variants have been proposed. They can be classified into three categories; loss-based (window-based), delay-based (rate-based) and their hybrid (mixed loss-delay-based). Loss-based control modifies the AIMD (Additive Increase Multiplicative Decrease) mechanism of TCP-Reno to quickly increase and slowly decrease the cwnd. Examples are High-speed TCP (HSTCP) [4], Scalable TCP (STCP) [5], early versions of TCP-Westwood (TCPW) [6,7,8], BIC [9], CUBIC [10] and H-TCP [11]. Delay-based control exploits RTT to predict network congestion before packet losses happen and shows excellent performance in throughput efficiency. Examples are TCP-Vegas [12] and FAST-TCP [13]. However, it is well known that their performances are degraded when competing with loss-based flows.

Hybrid TCP congestion control which combines both modes had appeared recently. Examples are Compound TCP (CTCP) [14], TCP-Adaptive Reno (ARENO) [15], TCP-Illinois [16], YeAH-TCP [17] and our TCP-Fusion [18]. They adaptively switch two modes according to congestion level measurement estimated from RTT. Hybrid TCP has two advantages against conventional methods; high throughput efficiency and inter-protocol fairness (friendliness). Throughput efficiency is provided thanks to its delay-based property especially when there is unused link capacity. Complete friendliness to TCP-Reno is provided thanks to its

loss-mode. Hybrid TCP can eliminate drawbacks of conventional methods as long as its mode switching is carried out in a deliberate manner.

On the other hand, performance modeling of conventional TCPs has been executed by many papers. The most famous one is for modeling TCP-Reno's congestion avoidance [19,20] which derives TCP throughput as a function of packet loss rate and RTT. This work is promptly extended to cover slow start phase [21], to derive TCP friendly rate control [22,23] and to define response functions as common performance measures of TCP variants [4,5,9,10,14]. Modeling of delay-based controls and TCPW are also conducted in [24,25,26,27].

However, within the authors' knowledge, performance analysis of the hybrid TCP has not yet been studied sufficiently. [14] derives a response function of CTCP, but its scope is limited to a narrow case of hybrid TCP usage. [28] provides useful results about CTCP and TCP-Illinois but no analytical evaluations are shown. Since there had been no analytical framework on hybrid TCP performance, we had to repeat simulations and implementations quite many times.

This paper therefore proposes simple analytical models of hybrid TCP congestion controls and tries their performance tuning. We firstly present ideal behavior models of loss-based, delay-based and hybrid TCPs. We then provide abstracted models of existing hybrid TCPs and consider their performance tuning. Experimental results validate our expectations and prove effectiveness of the hybrid TCPs. This paper is organized as follows. Section II presents analytical models of TCP variants. Section III gives various experimental results. Section IV concludes this paper.

II. SIMPLE ANALYTICAL MODELS OF HYBRID TCP CONGESTION CONTROLS

A. Ideal Models

We define ideal models of congestion avoidance behaviors of three congestion controls; loss-based, delay-based and hybrid. For loss-based control, we assume an AIMD control which is equivalent to TCP-Reno, in which cwnd is increased by one per RTT round and is halved upon packet losses. For delay-based control, we assume a model which tries to fill a pipe of the bottleneck link without causing buffering delay. In this case, when there is no residual capacity in a pipe, delay-based control has no chance to put packets into the pipe.

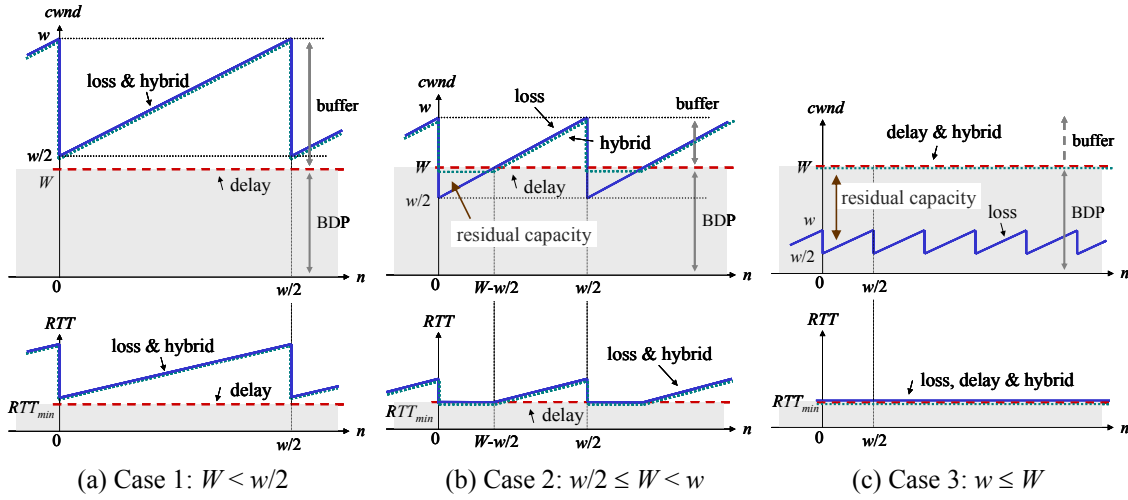


Figure 2: Congestion window and RTT behaviors of the single flow model.

Table 1: Expected number of transmitted packets and elapsed time during congestion avoidance round of the single flow model.

| | | Case 1: $W < w/2$ | Case 2: $w/2 \leq W < w$ | Case 3: $w \leq W$ |
|--------|---------------------|--|---|---------------------------------|
| Loss | transmitted packets | $\frac{3}{8}w^2$ | $\frac{3}{8}w^2$ | $\frac{3}{8}w^2$ |
| | elapsed time | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min}$ |
| Delay | transmitted packets | $\frac{1}{2}w \cdot W$ | $\frac{1}{2}w \cdot W$ | $\frac{1}{2}w \cdot W$ |
| | elapsed time | $\frac{1}{2}w \cdot RTT_{\min}$ | $\frac{1}{2}w \cdot RTT_{\min}$ | $\frac{1}{2}w \cdot RTT_{\min}$ |
| Hybrid | transmitted packets | $\frac{3}{8}w^2$ | $\frac{1}{2}w \cdot W + \frac{1}{2}(w-W)^2$ | $\frac{1}{2}w \cdot W$ |
| | elapsed time | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min}$ |

For hybrid control, we assume that it adaptively switches loss-mode and delay-mode according to network condition. When RTT stays at its minimum value, hybrid control works in delay-mode and fills a pipe. When RTT increase is observed due to packet buffering, hybrid control works in loss-mode until next packet losses happen.

Let p and w denote packet loss rate and cwnd when packet losses occur. In case of TCP-Reno, it is well known that their relationship is given by

$$p = \frac{8}{3w^2} \quad (1)$$

[21,22]. This equation is derived by observing periodic behavior of congestion avoidance phase of TCP-Reno. It may be applied to random loss cases due to channel errors and random drops by a RED router [21]. Let W represent cwnd which just fills a bottleneck link, that is equivalent to BDP (Bandwidth-Delay Product) of the pipe. We then represent long-term congestion avoidance behaviors by using two parameters w and W .

1) Single Flow Model

We consider a single flow model in which there are no competing flows. Fig.1 shows its connection topology. Then, congestion control behaviors can be classified into three cases as below.

- Case 1: Router buffer is bigger than BDP and packet losses are mainly caused by buffer overflow ($W < w/2$). In

this case, hybrid control always behaves as loss-based control.

- Case 2: Router buffer is smaller than BDP or packet losses happen during packet buffering due to random drops or random errors ($w/2 \leq W < w$). In this case, hybrid control behaves as delay-based control when cwnd is smaller than BDP (i.e. residual capacity exists) and behaves as loss-based control when cwnd is larger than BDP.
- Case 3: Router buffer stores no packets because packet losses happen severely due to random errors ($w \leq W$). In this case, because there is residual capacity, hybrid control always behaves as delay-based control.

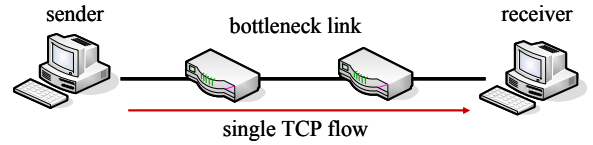


Figure 1: Connection topology of a single flow model.

Fig.2 demonstrates congestion window and RTT behaviors of each case. Horizontal axis denotes RTT rounds and vertical axes denote cwnd and RTT, respectively. When packets are buffered, RTT increases linearly in loss-mode. When no packets are buffered, RTT remains at RTT_{\min} in delay-mode. By taking integrals from $n=0$ to $w/2$, we can calculate the number of transmitted packets and elapsed time of single congestion avoidance round as shown in Table.1, where PS is a

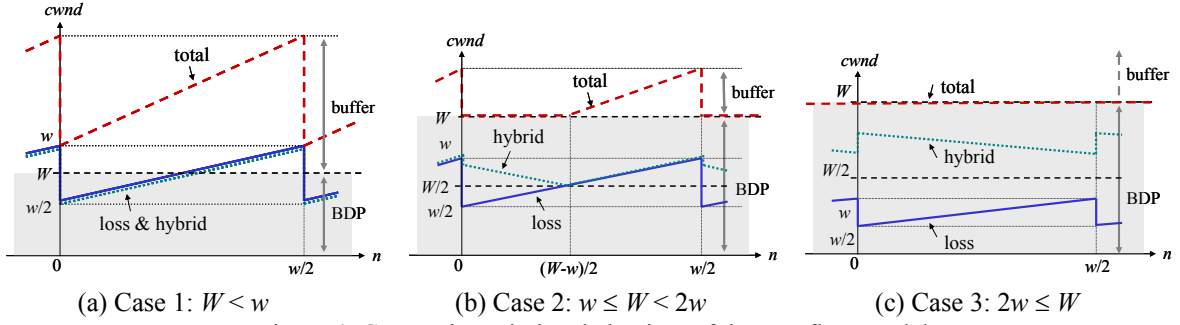


Figure 4: Congestion window behaviors of the two flow model.

Table 2: Expected number of transmitted packets and elapsed time during congestion avoidance round of the two flow model.

| | | Case 1: $W < w$ | Case 2: $w \leq W < 2w$ | Case 3: $2w \leq W$ |
|--------|-----------------------|--|--|---|
| Loss | transmitted packets | $\frac{3}{8}w^2$ | $\frac{3}{8}w^2$ | $\frac{3}{8}w^2$ |
| Hybrid | transmitted packets | $\frac{3}{8}w^2$ | $\frac{3}{8}w^2 + \frac{1}{4}(W-w)^2$ | $\frac{1}{2}w \cdot W - \frac{3}{8}w^2$ |
| | elapsed time (common) | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}w(3w-2W) \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}(2w-W)^2 \cdot \frac{PS}{B}$ | $\frac{1}{2}w \cdot RTT_{\min}$ |

packet size and B is bit rate of the bottleneck. We can incorporate timeout penalty of loss-based control given by

$$t_{RTO,loss} = T_0 \cdot (1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6)(1-p) \quad (2)$$

where T_0 is TCP retransmit timeout value [20] and correct this penalty for delay-based control (and for hybrid control) by

$$t_{RTO,delay} = \frac{K_{delay}}{K_{loss}} \cdot t_{RTO,loss} \quad (3)$$

where K_{loss} and K_{delay} are the expected number of transmitted packets of each control. Finally, we can calculate throughput estimates of each control by

$$\frac{\text{transmitted packets}}{\text{elapsed time} + \text{timeout penalty}} \quad (4)$$

Note that estimation of loss-based control is completely equal to that of [20] by considering elapsed time as average RTT.

2) Two Flow Model

We then consider a two flow model in which two TCP flows are competing (one is loss-based and the other is loss-based or hybrid). We omit delay-based because $\text{cwnd}_{\text{delay}} = \max(W - \text{cwnd}_{\text{loss}}, 0)$ in our model definition. Fig.3 shows connection topology of the two flow model. Similar to the previous subsection, congestion control behaviors are classified into three cases. Since there are two flows, w is halved when compared to the single flow case.

- Case 1: Router buffer is bigger than BDP ($W < w$). In this case, loss-based control and hybrid control behave completely the same.
- Case 2: Router buffer is smaller than BDP or packet losses happen during packet buffering due to random drops or random errors ($w \leq W < 2w$). In this case, hybrid control can gain residual capacity in delay-mode until cwnd reaches half of BDP ($W/2$). After that, it behaves as loss-based control until next packet losses happen.
- Case 3: Router buffer stores no packets because packet losses happen frequently due to random errors ($2w \leq W$).

In this case, hybrid control always behaves in delay-mode to fill a pipe.

Fig.4 shows congestion window behaviors of TCP flows and Table 2 summarizes the number of transmitted packets and elapsed time per single congestion avoidance round.

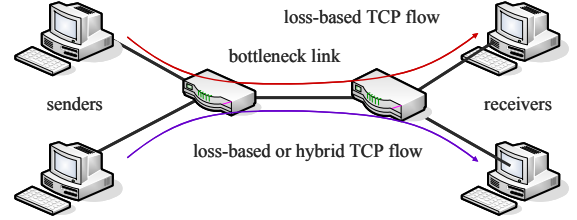


Figure 3: Connection topology of a two flow model.

B. Abstracted Models of Actual Hybrid TCPs

Next, we consider abstraction of the existing hybrid TCPs; CTCP, ARENO, YeAH-TCP and TCP-Fusion. For this purpose, we focus on their window increase/decrease mechanisms which can be abstracted as shown in Table 3. For TCP-Fusion, D_{\min} is TCP timer granularity to detect RTT change and N is the estimated number of competing flows. Window increase is carried out to fill a pipe in delay-mode and window decrease is carried out when packet losses happen. This window decrease/increase is executed simultaneously at $n=0$ and $w/2$ in Cases 2&3 of the ideal hybrid TCP model in Figs.2 and 4 though this is not possible for actual hybrid TCP.

Window increase rate determines how fast the hybrid TCP can refill a pipe after window decrease. Among four methods, TCP-Fusion, CTCP and ARENO show similar increase rates but YeAH-TCP does slower rate due to STCP. Due to this fact, it is expected that YeAH-TCP performance will degrade relatively when large residual capacity exists. Mathematically, TCP-Fusion is designed to refill a pipe in a single RTT round and moves to its steady state (keeping constant cwnd in the single flow case and smoothly decreasing cwnd to adapt to the competing TCP-Reno flow in the two flow case).

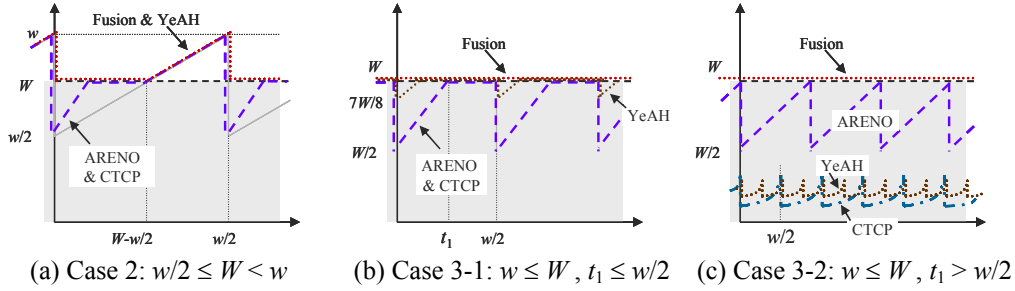


Figure 5: Abstracted models of congestion window behaviors of hybrid TCPs in the single flow case (no competing flow).

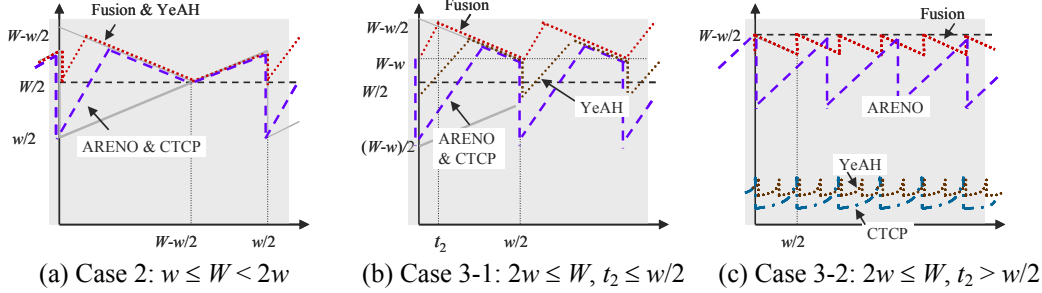


Figure 6: Abstracted models of congestion window behaviors of hybrid TCPs in the two flow case (competing with TCP-Reno).

Table 3: Window increase/decrease mechanisms of hybrid TCPs.

| | window increase | window decrease |
|----------------|-------------------------|----------------------------|
| CTCP[14] | $0.125 * cwnd^{0.75}$ | $1/2$ |
| ARENO[15] | $B/10\text{Mbps}$ | $1/2 \sim 1$ |
| YeAH-TCP[17] | STCP | $1/2, RTT_{\min}/RTT, 7/8$ |
| TCP-Fusion[18] | $B * D_{\min}/(N * PS)$ | RTT_{\min}/RTT |

Window decrease mechanisms can be roughly classified into two cases: halving cwnd similar to TCP-Reno or clearing a buffer similar to TCPW. CTCP is the former example and TCP-Fusion is the latter example. ARENO changes window decrease rate continuously from $1/2$ to 1 according to observed RTT. We approximate this behavior by 1 when $RTT = RTT_{\min}$ or $1/2$, otherwise. YeAH-TCP manages its decrease rate by $1 - \min(\max(1/8, 1 - RTT_{\min}/RTT), 1/2)$, which can be split into

$$\begin{cases} 1/2 & (RTT > 2 * RTT_{\min}) \\ RTT_{\min} / RTT & (2 * RTT_{\min} > RTT > 8/7 * RTT_{\min}) \\ 7/8 & (8/7 * RTT_{\min} > RTT) \end{cases} \quad (5)$$

Assignment of constants ($1/2$ and $7/8$) causes cwnd reduction even if RTT stays at its minimum. Therefore, it is expected that performances of CTCP and YeAH-TCP will degrade when residual capacity exists.

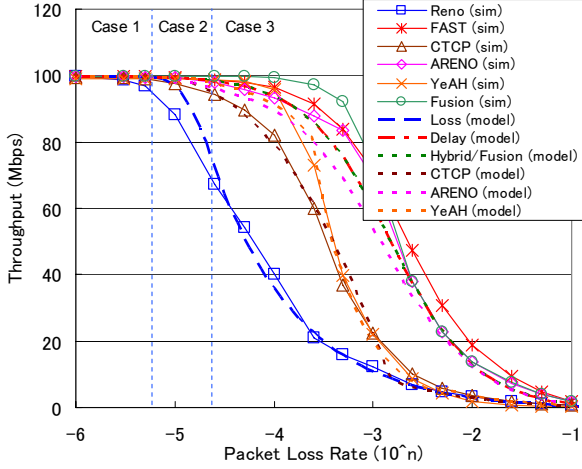
Figs.5 and 6 show congestion window behavior models of four hybrid TCPs for the single flow case (Fig.5) and for the two flow case (Fig.6). Case 1 is omitted because they are the same as those of Figs.2 and 4. Case 3 is separated into two cases according to whether packet losses happen before or after cwnd can refill a pipe, where t_1 and t_2 are the time when the total number of transmitted packets reach BDP. Case 3-2 is the case that CTCP and YeAH-TCP never reach BDP due to constant decrease rate which is smaller than one. We can calculate the number of transmitted packets and elapsed time of single congestion avoidance round of each hybrid TCP similar to Tables 1 and 2 though the results are omitted.

III. PERFORMANCE EVALUATIONS AND DISCUSSIONS

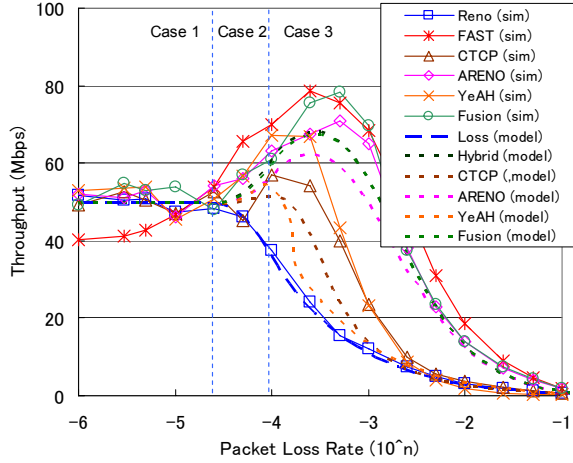
A. Throughput Efficiency and Friendliness

Fig.7 shows analytical throughputs of three ideal models (loss, delay and hybrid) and four hybrid models (CTCP, ARENO, YeAH-TCP and TCP-Fusion) when packet loss rates (p) are changed. We also add ns-2 [29] simulation results of TCP-Reno (SACK), FAST-TCP and four hybrids. Experiment parameters of Figs.1 and 3 are as follows: access link is 1Gbps, bottleneck link is 100Mbps, end-to-end RTT_{\min} is 40ms, and router buffer size is equal to BDP (333 packets). For analysis, TCP retransmit timeout T_0 is $2 * RTT_{\min}$. Two vertical dashed lines denote borders between Cases 1 and 2 and Cases 2 and 3, respectively. For the single flow case of Fig.7(a), analytical throughput of TCP-Fusion is equal to that of the ideal hybrid. Analytical throughputs of the ideal delay and hybrid models are almost similar though their RTTs of Case 1 are different. For the two flow case of Fig.7(b), analytical throughput of the ideal delay case is not shown.

From these figures, we can recognize that our models can sufficiently approximate simulation results of the single flow case regardless of its simplicity. In the two flow case, approximation is not sufficient for CTCP and YeAH-TCP, but overall tendency can be captured enough. Among TCP variants, TCP-Fusion and FAST always compete for the best performance. They are aligned with the ideal behavior models of delay and hybrid, respectively. A noticeable advantage of TCP-Fusion is observed in Case 1 of Fig.7(b); where it can keep inter-protocol fairness (friendliness) to TCP-Reno, but FAST suffers from unfairness due to its delay-based property. A criticism might happen against the importance of Case 3 region, that such a region of high packet loss rates is never used. However, Case 3 region shifts to left (toward low packet loss rate) as bandwidth becomes larger (above 10Gbps).



(a) Single flow case without competing flow.



(b) Two flow case competing with TCP-Reno flow.

Figure 7: Estimated throughputs and ns-2 simulation results of TCP variants.

B. Performance Tuning

As shown in Fig.7, CTCP and YeAH-TCP relatively degrade in Case 3 region (though better than TCP-Reno). The reasons are described in the previous section; due to slow window increase of YeAH-TCP and constant (non-adaptive) window decrease of CTCP. Therefore, to validate our analytical assumptions and to promote hybrid TCP potentials, we carried out both the analytical estimations and ns-2 simulations in which their window increase/decrease parameters are modified. In case of YeAH-TCP, we changed STCP's window increase parameter from 1.01 (original) to 1.1. In case of CTCP, we changed a window decrease parameter from 1/2 (original) to 7/8. Such parameter changes are expected to improve their throughput performances (i.e. residual capacity usage) especially in Case 3 region.

Although we omitted details of the evaluation results, parameter tuning above indeed brings performance improvement to CTCP and YeAH-TCP, and they can approach to the ideal hybrid performance in Fig.7. However, our

analytical model cannot reflect influence of too aggressive packet insertion because it truncate surplus packets after it reaches BDP. Therefore, we also evaluate router queue behaviors and impact on competing flows by simulations. As expected, too aggressive parameter assignment like 1.1 for STCP expels competing flows and loses inter-protocol friendliness.

An example result is shown in Fig.8, in which impact of long-lived TCP variants onto short-lived flows [30] are evaluated. In this experiment, a long-lived flow is one of TCP variants as listed in horizontal axis with their parameter setting. It shares 100Mbps bottleneck link and competes with 100 short-lived flows which appear randomly. The numbers in Fig.8 give total throughputs of short-lived flows. This result show that aggressive parameter assignment expels short-lived flows though it contributes to its own throughput increase. If we assume that TCP-Reno is the best in fairness sense, TCP-Fusion, YeAH(1.03) and CTCP(7/8) provide good balance between throughput and inter-protocol fairness.

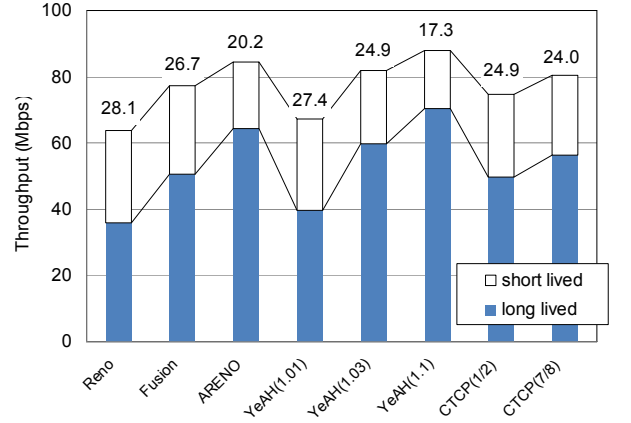


Figure 8: Impact of long-lived TCP variants onto short-lived TCP-Reno flows.

C. Response Function

Finally, we provide a response function of hybrid TCP, which is defined as the averaged number of transmitted packets per RTT as a function of packet loss rate [4]. Fig.2 directly contributes to this definition. By translating w into p by Eq.(1), the response function of hybrid TCP is given by

$$R_{\text{hybrid}}(p) = \begin{cases} \sqrt{3/2p} & (\text{Case 1}) \\ \sqrt{8/3p} + \sqrt{3p/8} \cdot W^2 - W & (\text{Case 2}) \\ W & (\text{Case 3}) \end{cases} \quad (6)$$

Three cases corresponds to those of Fig.2. Case 1 is TCP-Reno. Case 2 is originally $(w-W)^2/w+W$ which is hybrid mode. Case 3 is delay-mode. Since the value of Case 3 is unrealistic in lossy condition, we replace it by Eq.(4). Then, Fig.9 presents response functions of TCP-Reno, HS-TCP, STCP and hybrid TCP, where we assume RTT=100ms similar to previous work.

For hybrid TCP, bandwidth parameter W is newly introduced. This is to represent a transition point from Case 1 (loss-mode) to Case 2 (hybrid mode), which scales according

to bandwidth. We plotted two example curves of 10Gbps and 10Mbps. Since lower packet loss rate is required to always fill up BDP (Case 1), a point at which the hybrid curve hits TCP-Reno's curve moves left as bandwidth increases. On the other hand, at the right side of the figure (high packet loss rate), the hybrid curve keeps larger packets than HSTCP and STCP as bandwidth increases. This means that hybrid TCP becomes more efficient for high BDP network by exploiting its delay-mode.

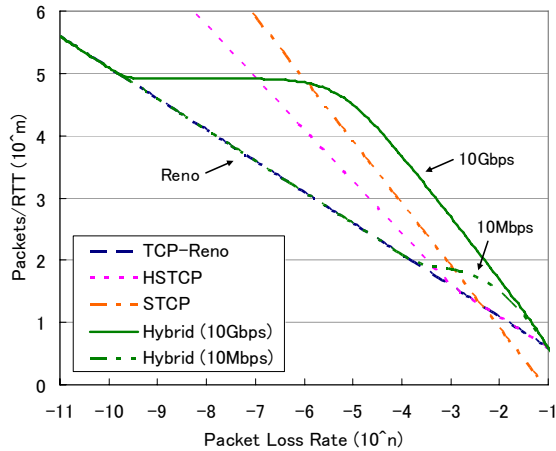


Figure 9: Comparison of response functions.

IV. CONCLUSIONS

This paper presented simple but useful models and experimental results to promote the hybrid TCP. They quantitatively proved its effectiveness and provided its performance tuning strategies. They can be also utilized for various performance estimation and improvement of the hybrid TCP, some of which are omitted in this paper (e.g. buffering effect onto delay-mode). However, a problem remains that approximation capability of the proposed models is not sufficient especially in Case 3 region and they have to be refined in future. We had published a part of MATLAB code, ns-2 simulation code and Linux implementation code of TCP-Fusion at [31].

REFERENCES

- [1] M.Allman, V.Paxson and W.R.Stevens: "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," IETF RFC 2581, April.1999.
- [2] C.H.Janey: "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," ACM SIGCOMM 1996, August 1996.
- [3] M.Mathis et al.: "TCP Selective Acknowledgment Options", IETF RFC 2018, Oct.1996.
- [4] S.Floyd: "Highspeed TCP for Large Congestion Window", IETF RFC3649, Dec.2003.
- [5] T.Kelly: "Scalable TCP: Improving Performance in High-speed Wide Area Networks." PFLDnet 2003, Feb.2003.
- [6] C.Casetti, M.Gerla, S.Mascolo, M.Y.Sanadidi, and R.Wang: "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", ACM Mobicom 2001, Jul.2001.

- [7] R.Wang, M.Valla, M.Y.Sanadidi, B.K.F.Ng and M.Gerla: "Efficiency/Friendliness Tradeoffs in TCP Westwood", Seventh IEEE Symposium on Computers and Communications, 2002. Jul.2002.
- [8] R.Wang, M.Valla, M.Y.Sanadidi and M.Gerla: "Adaptive Bandwidth Share Estimation in TCP Westwood", IEEE Globecom 2002, Nov.2002.
- [9] L.Xu, K.Harfoush and I.Rhee: "Binary Increase Congestion Control for Fast, Long Distance Networks", IEEE INFOCOM 2004, Mar.2004.
- [10] I.Rhee and L.Xu: "CUBIC: A New TCP-Friendly High-speed TCP Variant", PFLDnet 2005, Feb.2005.
- [11] R.N.Shorten and D.J.Leith: "H-TCP: TCP for High-Speed and Long-Distance Networks", PFLDnet 2004, Feb.2004.
- [12] L.S.Brakmo and L.L.Peterson: "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet," IEEE Journal on Selected Areas in Communications, Vol. 13, No.8, 1995.
- [13] C.Jin, D.X.Wei and S.H.Low: "FAST TCP: Motivation, Architecture, Algorithms, Performance", IEEE INFOCOM 2004, Mar.2004.
- [14] K.Tan, J.Song, Q.Zhang, and M.Sridharan: "Compound TCP: A Scalable and TCP-Friendly Congestion Control for High-speed Networks", PFLDnet 2006, Feb.2006.
- [15] H.Shimonishi, T.Hama and T.Murase: "TCP-Adaptive Reno for Improving Efficiency-Friendliness Tradeoffs of TCP Congestion Control Algorithm", PFLDnet 2006, Feb.2006.
- [16] S.Liu, T.Başar and R.Srikant: "TCP-Illinois: A Loss and Delay-Based Congestion Control Algorithm for High-Speed Networks", VALUETOOLS 2006, Oct.2006.
- [17] A.Baiocchi, A.P.Castellani and F.Vacirca: "YeAH-TCP: Yet Another Highspeed TCP", PFLDnet 2007, Feb.2007.
- [18] K.Kaneko, T.Fujikawa, S.Zhou and J.Katto: "TCP-Fusion: A Hybrid Congestion Control Algorithm for High-speed Networks", PFLDnet 2007, Feb.2007.
- [19] M.Mathis, J.Semke, J.Mahdavi, T.Ott.: "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, Vol.27, No.3, Jul.1997.
- [20] J.Padhye, V.Firoiu, D.Towsley and J.Kurose: "Modeling TCP Throughput: A Simple Model and its Empirical Validation", ACM SIGCOMM 1998, Oct.1998.
- [21] N.Cardwell, S.Savage and T.Anderson: "Modeling TCP Latency", IEEE INFOCOM 2000, Mar.2000.
- [22] S.Floyd, M.Handley, J.Padhye and J.Widmer: "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM 2000, Aug.2000.
- [23] M. Handley et al.: "TCP Friendly Rate Control (TFRC): Protocol Specification", IETF RFC 3448, Jan.2003.
- [24] C. Samios and M. K. Vernon: "Modeling the Throughput of TCP Vegas", ACM SIGMETRICS 2003, June 2003.
- [25] C.Jin, D.X.Wei and S.H.Low: "The Case for Delay-based Congestion Control", CCW 2003, Oct.2003.
- [26] L.A.Grieco and S.Mascolo: " TCP Westwood and Easy RED to Improve Fairness in High-Speed Networks", PfHSN 2002, Apr.2002.
- [27] E.Altman et al.: "Analysis of TCP Westwood+ in High Speed Networks", PFLDnet 2006, Feb.2006.
- [28] D.J.Leith et al.: "Experimental Evaluation of Delay/Loss-based TCP Congestion Control Algorithms", PFLDnet 2008, Mar.2008.
- [29] "ns-2 network simulator(ver.2)," <http://www.mash.cs.berkeley.edu/ns>.
- [30] L.Guo and I.Matta: "The War between Mice and Elephants," IEEE ICNP 2001, Nov.2001.
- [31] <http://www.katto.comm.waseda.ac.jp/TCP-Fusion/>.