Adaptive Time Window Improving Convergence of Practical Delay-based Congestion Control

Shotaro Ishikura\(^{1a}\), and Miki Yamamoto\(^{1b}\)

\(^{1}\) Graduate School of Science and Engineering, Kansai University
3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680, Japan

\(a\) k101355@kansai-u.ac.jp
\(b\) yama-m@kansai-u.ac.jp

Abstract: Copa\(^{[1]}\) is a practical delay-based end-to-end congestion control. It can attain a unique Nash equilibrium with other flows with achieving high throughput and low queueing delay. In this paper, we propose adaptive time window\(^{[2]}\) which can resolve Copa’s technical problem of initial phase overshoot of the congestion window size caused by slightly long time period for measuring the smallest roundtrip time. Our evaluation results show that our proposed adaptive window improves convergence of congestion window in the initial phase\(^{[2]}\) and also shows good performance even in the situation that network jitter fluctuates measured RTT.

Keywords: Congestion Control, Copa, Delay-Based, Convergence

Classification: Internet

References

[1] V. Arun, and H. Balakrishnan, “Copa: Practical Delay-Based Congestion Control for the Internet,” Proc. 15\(^{th}\) Conf. USENIX Symposium on Networked Systems Design and Implementation, WA, USA, pp. 329–342, April 2018.

[2] S. Ishikura, and M. Yamamoto, “Convergence Improvement for Practical Delay-based Congestion Control,” Proc. 11\(^{th}\) Conf. IEICE International Conference on Emerging Technologies for Communications, Tokyo, Japan, December 2020.

[3] M. Dong, Q. Li, D. Zarchy, P. Fodfrey, and M. Schapira, “PCC: Re-architecting Congestion Control for Consistent High Performance,” Proc. 12\(^{th}\) Conf. USENIX Symposium on Networked Systems Design and Implementation, CA, USA, pp. 395–408, May 2015.

[4] K. Weinstein, and H. Balakrishnan, “TCP ex Machina: Computer-Generated Congestion Control,” Proc. Conf. ACM SIGCOMM, Hong Kong, China, pp. 123–134, August 2013.

[5] F. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” the Operational Research Society, vol. 49, no. 3, pp. 237–252, May 1998.

[6] NS3, “network simulator 3,” https://www.nsnam.org/, accessed Jan. 27, 2021.
1 Introduction
Congestion control is still a hot topic in the network research area. Its goal is to achieve high throughput and low queueing delay in many variations of network scenarios. Machine learning approach\cite{3}\cite{4} is expected to be a promising approach because it can adjust its rate to various situations. Their weakness, however, is its complexity which is far beyond human understandings.

To achieve the goal with the algorithmic approach, Copa\cite{1} has been proposed. Copa is a delay-based end-to-end congestion control which uses AIAD window-update rule to adjust a congestion window size (cwnd) to a target window size (twnd). Here, twnd is \( \frac{RTT_{\text{min}}}{\delta \times d_q} \), where \( RTT_{\text{min}} \) is the measured minimum roundtrip times (RTTs) over long time and \( d_q \) is the (measured) queueing delay calculated by the difference between the smallest RTT over a time window, \( RTT_{\text{standing}} \), and \( RTT_{\text{min}} \). The term of \( \frac{1}{\delta \times d_q} \) means a steady-state throughput which maximizes the Utility function \( U = \log \lambda - \delta \log d_q \)\cite{5}, where \( \delta \) is weight for delay in Utility function. This steady-state throughput is proved to make the system to be at a unique Nash equilibrium (see \cite{1} in detail). The queueing delay used for calculating twnd is expected to be almost the same for all flows sharing the bottleneck link, which means Copa can achieve fairness among Copa flows.

However, its window control mechanism has a delayed factor, which causes overshoot of cwnd. Copa uses constant time window for estimating \( RTT_{\text{standing}} \), the tentative smallest RTT, though the amount of change in cwnd has dynamically changed. Overshoot of congestion window size leads slightly large queue length and also long period of zero queue length, which causes respectively long RTT and low throughput. In this paper, we propose adaptive time window\cite{2}, which adaptively changes the time window according to the elapsed time in exponential increase phase.

Our evaluation results for simple network setting, i.e. the case without network jitter, our proposed adaptive time window improves convergence performance of Copa\cite{2}. In Copa, the reason for estimating the smallest RTT in the time window is robustness for ACK compression and network jitter. So, it is very important for our proposed adaptive time window to check the robustness for the network jitter, and in this paper, we additionally valuate adaptive time window for the case with network jitter. Our evaluation results for the network jitter case also show that our proposed adaptive time window improves convergence performance, which verifies robustness for the network jitter.

2 Adaptive Time Window

2.1 Copa Algorithm
Copa has only two main actions, cwnd control with AIAD window-update rule and parameter setting according to the current network status. When
a sender receives an ack, it compares cwnd and twnd. When cwnd is larger than twnd, a sender decreases its cwnd, and otherwise increases it. The amount of cwnd change is $v \times \frac{\delta \times cwnd}{\delta < 0}$.

The parameter $v$ is velocity parameter and is used to quickly bring cwnd closer to twnd when the difference between cwnd and twnd is large. After change direction (increase or decrease) of cwnd remains during 3 RTT, $v$ is doubled once in each RTT. And when change direction is changed, i.e. increase to decrease or vice versa, $v$ is reset to 1. This exponential increase of velocity parameter allows Copa increases its cwnd exponentially as in slow start phase of the conventional TCP.

In steady state, cwnd fluctuates near the bandwidth delay product (BDP) and achieves near zero queue length at the bottleneck link. This feature enables high throughput and low queueing delay. And also a new entry flow can measure the $RTT_{min}$ of the topology correctly.

### 2.2 Technical Issue of Copa

In Copa, $RTT_{standing}$ which is the smallest RTT during the current time window, is used for estimation of queueing delay instead of the current RTT. The time window is static length irrespective of the amount of the change in cwnd. When the change of cwnd is large, the network status might change a lot, i.e. the bottleneck queue length fluctuates so much. In the window increase phase, the current measured RTT is to be used for estimation because estimated queue length from the smallest RTT in the time window is too small. This is due to the control delay induced by the time window.

In exponential increase phase of cwnd, cwnd might overshoot from ideal value due to control delay by the time window of $\frac{rtt}{2}$, where $rtt$ is standard smoothed RTT samples. This control delay in exponential increase phase will bring extra growth of cwnd. This extra increase of cwnd causes next overshoot in exponential decrease phase (Copa exponentially decrease its cwnd with doubling velocity parameter $v$). This overshoot cycle will appear iteratively, and causes temporal steep increase of queueing delay, which should be generally prevented in congestion control.

### 2.3 Adaptive Time Window

In this paper, we propose adaptive time window which changes the time window according to the elapsed time in the exponential increase phase. Rather long interval for estimating $RTT_{standing}$ in Copa is for securing robustness in the face of ACK compression and network jitter. Therefore, in adaptive time window, we gradually reduce time window only in exponential increase phase. And time window is reset to the default value immediately after exponential phase terminates. With regulating time window only in exponential increase phase, quick response is adequately invoked only when necessary.

More specifically, when measurement interval of $RTT_{standing}$ is represented by $\frac{rtt}{2} \times r$ (in conventional Copa, $r$ is fixed and $r = 1$), parameter $r$ is decreased with increase of RTT. In our evaluation in the next section, we would like to evaluate our proposed approach for the following three cases,
Linear decrease \((1 - \frac{x}{10})\), Concave decrease \((\sqrt{\frac{-x}{10}} + 1)\) and Convex decrease \((-\sqrt{\frac{x}{10}} + 1)\), where \(x \times RTT_{\text{min}}\) is defined as elapsed time of exponential increase phase (Fig. 1).

**Fig. 1.** measurement interval parameter \(r\)

### 3 Performance Evaluation

#### 3.1 Basic Performance

In our evaluation, we would like to evaluate our proposed adaptive time window in exponential phase. We use ns3[6] for our evaluation. We use a simple tandem topology and just one flow starts its transmission, i.e. this flow starts its exponential increase phase just after its transmission. Bandwidth of the link between routers is 40Mbps and others are 100Mbps, respectively (Fig. 2(a)).

Figures 2(b) and 2(c) show the transition of cwnd and queue length of the router, respectively. All of our proposals have shorter convergence time than the conventional Copa. Among our 3 cases, ”Convex” shows the fastest convergence because it can estimate more accurate queue length with steep reduction of \(r\), i.e. steep reduction of control delay. Only from the viewpoint of convergence, Convex shows the best performance. One of the reason why Copa uses time window for estimation of the current smallest RTT is robustness in the face of network jitter. So, we should carefully evaluate adaptive time window in the case with network jitter.

**Fig. 2.** cwnd and queue length characteristics (w/o network jitter)
3.2 Robustness for network jitter

In this subsection, we evaluate adaptive time window in more realistic case, i.e. with network jitter. To simulate network jitter on ns3, we fluctuate RTT (propagation delay) between 28.5ms and 31.5ms according to truncated normal distribution with average 30ms and distribution 0.3.

![Figure 3. cwnd and queue length characteristics (with network jitter)](image)

Figure 3(a) and 3(b) shows cwnd and queue length characteristics for the case with network jitter, respectively. Conventional Copa shows fluctuation of cwnd in the initial phase (0.5-1.0sec) as also shown in the basic performance. And it has fluctuation of cwnd even after its initial phase. Copa estimates queueing delay as $RTT_{\text{standing}} - RTT_{\text{min}}$. Network jitter might cause increase of estimated queueing delay, which causes overestimation of queueing delay and finally large fluctuations of cwnd. Our proposed adaptive window improves the initial phase of fluctuation and also fluctuation of cwnd after the initial phase. “Convex”, especially, shows good convergence performance even for the case with network jitter, which means our proposed adaptive time window has robustness in the face of network jitter.

4 Conclusions

In this paper, we propose a new improvement of control delay for the promising new congestion control, Copa. Our evaluation results show that proposed adaptive time window with “Convex” parameter shows significantly better convergence than the conventional Copa and adaptive time window with other settings. And we also evaluate adaptive time window in the more realistic situation with network jitter. Our evaluation results show adaptive time window with “Convex” setting has also robustness for the network jitter.