Delay Based Congestion Control for Cross-Datacenter Networks

Yantao Geng[13], Han Zhang [13], Xingang Shi[13], Jilong Wang[13], Xia Yin[23], Dongbiao He[4], and Yahui Li[13]

[1]INSC, Tsinghua University [2]DCST, Tsinghua University [3]Zhongguancun Laboratory [4]Sangfor Technologies Inc

2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS) | 979-8-3503-9973-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/IWQOS57198.2023.10188700

***Abstract*—Numerous distributed applications are deployed in the cross-datacenter networks (Cross-DC) where geographically distributed data centers (DC) are connected by wide area network (WAN). These online applications will generate both intra-datacenter and inter-datacenter traffic, each with distinct requirements and characteristics. We find that existing combined congestion control schemes ignore the interaction of the two types of traffic and the hybrid congestion control schemes fail to accurately estimate cross data center network congestion extent. In this paper, we propose IDCC a delay based congestion control scheme that uses delay to handle congestion inside the DC and in the WAN, respectively. We respectively utilize In-band network telemetry (INT) and round trip time (RTT) to measure the queuing delay inside DC and in the WAN and guarantee the stability of the algorithm by Proportional Integral Derivative (PID). Simultaneously, we demonstrate the empirical results of optimizing flow completion time (FCT) of intra-DC short flow by weight function in cross-DC. We implemented IDCC in simulation platform ns-3 and have performed extensive large scale simulation evaluations. Results show that IDCC decreases the FCT of intra-DC traffic by 3.6× to 12× and improves the throughput of inter-DC traffic by 9% to 16% compared to Gemini, Annulus.**

***Index Terms*—Data Center Networks, Wide-Area Network, Congestion Control**

I. Introduction

Many cloud services (e.g., web search, social networks, recommendation systems) have already been deployed to serve clients in geographically dispersed data centers around the world. The cross-DC network consists of intra-DC and inter-DC networks, with an approximate 6:1 intra-DC to inter-DC traffic ratio [1], [2].

The intra-DC and inter-DC traffic now share the networks of data centers, and their network requirements are constantly varied. The inter-DC traffic is immune to delay and requires *high bandwidth.* In comparison, most flows in data centers are short and they need *low latency* [3], [4]. The inter-DC traffic and the intra-DC traffic could impact each other, which can degrade application performance. Up still now, two categories of methods have already been raised to make up for deficiencies:

Type 1: combined congestion control. This type of methods employ a data center congestion control method (e.g., DCTCP [3]) and a WAN congestion control method (e.g., BBR [5], Cubic [6]) to optimize intra-DC and inter-DC traffic, respectively. They are simple, but they disregard *traffic interaction*, which may cause applications to behave poor. In reality, WAN congestion control methods are always aggressive and have long round trip time (RTT), which implies that the inter-DC traffic tends to produce deep queues or even buffer overflows within data center switches. Therefore, intra-DC traffic may experience significant latency or packet loss, resulting in poor performance of applications within the data center.

Type 2: hybrid congestion control. This type of approaches (e.g., Gemini [2], Annulus [1], GTCP [7]) use a hybrid signal to estimate intra-DC and inter-DC network condition and implement congestion control based on this information. Gemini [2], for instance, utilizes ECN signals to estimate intra-DC congestion and delays to estimate inter-DC congestion. Although hybrid congestion control schemes can mitigate performance degradation caused by traffic interaction, it is challenging to precisely balance the intra-DC and inter-DC congestion signals. Hence, either intra-DC or inter-DC applications may encounter defects in performance.

In this paper, we propose IDCC a delay based congestion control mechanism. IDCC derives queue depth from INT to adjust the behavior of inter-DC traffic to achieve lower latency of intra-DC traffic. Besides we use RTT signals to guarantee the bandwidth utilization of the WAN link, where INT may not be available. To preserve the stability and convergence speed of the method, we use the deviation and variation rate of the queue length and RTT as inputs to the two PID controllers to adapt the amount of variation in the congestion window, respectively. To optimize the performance of intra-DC short flows, we design a weight function based on sent bytes to give more resources to short flows when adjusting the congestion window so as to reduce the completion time of short flows. We have implemented IDCC on the simulation platform ns-3 [8] and performed large scale simulation evaluations with realistic workload. Results show that IDCC can significantly decrease the FCT of short flows in intra-DC by 3.6 to 12 times and improves the average throughput of inter-DC by 9% to 16% compared to state-of-the-art approach Gemini and Annulus.

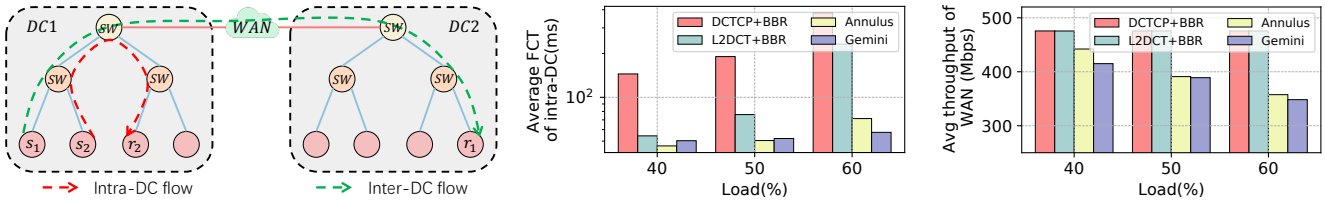
II. Motivation

The performance metric for data center applications(e.g., message queue) is to lower the average flow completion time. High bandwidth is required for applications(e.g., distributed

Corresponding authors: Han Zhang, zhhan@tsinghua.edu.cn

979-8-3503-9973-8/23/$31.00 © 2023 IEEE

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:42:24 UTC from IEEE Xplore. Restrictions apply.



(a) Cross-DC topology. (b) Average FCT of intra-DC flows. (c) Average throughput of inter-DC flows.

Fig. 1. Simulation results of the motivation section.

database) across data centers. All applications share the data center network links, therefore only considering for intra-DC or inter-DC flows might result in poor performance.

In ns-3 [8], we ran a simple experiment to demonstrate the drawbacks of combined congestion control methods and hybrid congestion control methods in meeting the needs of intra-DC and inter-DC applications. The topology is illustrated in Fig. 1(a), where two small data center sites connected by a WAN link with a capacity of 500 Mbps and a propagation latency of 5ms. Each link in data center has a capacity of 1Gbps and a propagation delay of 2us. S2 in DC1 runs web search traffic [3] to simulate the intra-DC applications and S1 runs a long flow to simulate the inter-DC applications. DCBR (DCTCP for intra-DC flows and BBR for inter-DC flows) and L2BR(L2DCT [9] for intra-DC flows and BBR for inter-DC flows) are selected as baselines for combined schemes. We take GEMINI [2] and Annulus [1] as the baselines for the hybrid schemes. Fig. 1 demonstrates the results, which reveals that despite the fact that both schemes are capable of optimizing application performance, they still have the following two **drawbacks**:

Firstly, the combined schemes could cause short intra-DC flows to experience large average FCT. In a modern data center network, more than 90% of flows are short (e.g., less than 100 KB) and switch buffers are always shallow.The combined schemes advocate optimizing flows independently, which may result in inter-DC flows occupying the switch buffer due to their delayed response to congestion. The deep switch buffers could also hamper the short flows. The results shown in Fig.1(b) also demonstrate this, as the average FCT of intra-DC short flows under DCBR and L2BR can be more than 3 × that of hybrid schemes.

Secondly, the hybrid schemes fail to accurately estimate network congestion extent. The hybrid schemes advocate using distinct intra-DC and inter-DC signals, then performing congestion control based on the hybrid congestion estimation signals. Although the hybrid schemes take network heterogeneity into account, they are difficult to precisely balance the two signals, limiting their practicality. Also, most schemes ignore the short flows in intra-DC. The results in Fig.1(c) are likewise consistent with this, as the hybrid methods exhibit low WAN bandwidth consumption due to a lack of accurate congestion balance.

Algorithm 1 Congestion control in sender.

Input: New Incoming ACK
Output: New Congestion Window $cwnd$
procedure PROCESS_ACK(ACK)
  /* Update RTT according to the timestamp. */
  rtt_base, rtt_min = update_RTT($ACK.timestamp$)
  /* Update the queue length according to INT. */
  $qLen_min$ = update_qLen($ACK.INT$)
  $u(t) = u(t-1) + K_{p1} \times (qLen_min - qLen_T) + K_{d1} \times (qLen_min - qLen_old)$
  $m(t) = m(t-1) + K_{p2} \times (rtt_min - rtt_T) + K_{d2} \times (rtt_min - rtt_old)$
  if $u(t) > m(t)$ **then**
    $cwnd = cwnd \times (1 - tanh(f(BS) \times u(t)))$
  else
    $cwnd = cwnd \times (1 - tanh(m(t)))$
  end if
end procedure

TABLE I
PARAMETERS AND VARIABLES USED IN ALGORITHM 1

Parameter	Description
K_{p1}, K_{d1}	Parameters of proportional integral control for intra-DC queue length
K_{p2}, K_{d2}	Parameters of proportional integral control for RTT
$qLen_T$	Threshold for intra-DC queue length
rtt_T	Threshold for RTT

Variable	Description
$cwnd$	Congestion window
rtt_base	Minimum RTT observed during a long time
rtt_min	Minimum RTT observed in current RTT
rtt_old	Minimum RTT observed in previous RTT
$qLen_min$	Minimum intra-DC queue length observed in current RTT
$qLen_old$	Minimum intra-DC queue length observed in last RTT
BS	Bytes sent by flow
$f(\cdot)$	Weight function to adjust the magnitude of the window change

III. DESIGN

To address the limitations of the aforementioned algorithms, we suggest a delay based congestion control scheme called IDCC. This algorithm is window-based and utilizes multiplicative increase and multiplicative decrease (MIMD) mechanism. IDCC uses the PID algorithm to control queuing depth in both data center (DC) and wide area network (WAN)

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:42:24 UTC from IEEE Xplore. Restrictions apply.

simultaneously. This is achieved through queue length of intra-DC switches and RTT. The INT-acquired queue length signal primarily addresses intra-DC congestion, meeting strict queuing requirements for shallow caches within the DC. On the other hand, RTT is responsible for bandwidth utilization in WAN since INT may not be supported there. It further optimizes the performance of short flows by adapting the magnitude of the window modification based on the sent bytes of the flow.The pseudo code of IDCC is shown in Algorithm 1. The parameters and variables are summarized in Table I.

Controlling the DC and WAN queues separately by proportional integral control. The sender dynamically adapts the size of the congestion window by proportional integration based on the queue length signal and the delay signal.The queue length in the DC carried by the INT and the WAN queuing delay estimated by the delay signal are essentially the identical signals, so they can be separately controlled using two conventional PI controllers based on AFD [10], PIE [11], and RoCC [12]. The final congestion window is the minimum value calculated by the two algorithms. We take the queue length within the DC carried by INT as an example to illustrate the specific design of the PI algorithm here.

The controller continuously adjusts the congestion window to match the queue length within the DC to the target $qLen_T$. We calculate the control factor $u(t)$ based on two quantities obtained from the queue size: (i) the deviation of the queue size from $\Delta qLen = (qLen_min - qLen_T)$, which represents the degree of congestion, and (ii) the direction of queue change $\Delta \dot{qLen} = (qLen_min - qLen_old)$, which indicates the trend of congestion. The parameters K_{p1} and K_{d1} determine the weights of the two components. To calculate the control factor $u(t)$ used to adjust the sending window, the controller applies a correction based on proportional and differential terms. Equation 1 represents the overall control function. The proportional term guarantees that the PID-based congestion control mechanism converges to a fixed point, while the derivative term is used to achieve fast convergence.

$$u(t) = u(t-1) + K_{p1} \times \Delta qLen + K_{d1} \times \Delta \dot{qLen} \quad (1)$$

Due to the different thresholds and orders of magnitude difference in queue lengths within RTT and DC, two PID algorithms with distinct parameters are required to calculate the corresponding control factors separately. The larger the control factor, the more severe the congestion and the faster the growth rate of congestion, so the maximum value of both needs to be chosen to decelerate the window. The final window adjustment is illustrated in Equation 2.

$$cwnd \ = \ cwnd \times (1 - tanh\left(\max\left(u(t), m(t)\right)\right)) \quad (2)$$

Optimizing short flow performance in combination with sent bytes. Equation 2 divides the bandwidth equally among all flows which ignores the disparate demand for short foreground and long background flows. To optimize the completion time of short intra-DC flows, we use the sent bytes of the flow to adjust the magnitude of the window change. Specifically, when $u(t) > m(t)$ which means the most serious congestion occurs within the DC, we design a function $f(BS)$ that gradually increases with the number of sent bytes. This function is multiplied by the control factor $u(t)$. In case of congestion, shorter intra-DC flows with smaller $f(BS)$ incur smaller fallback penalties and can also be more aggressive when grabbing bandwidth.

IV. Evaluation

We construct mixed traffic scenarios on simulation platform ns-3 [8] and perform large scale simulation to evaluate the performance of IDCC.

Network topology: We perform simulations on data centers with Fat-tree, VL2 [13] and spine-leaf [14] topologies, respectively. The topologies are shown in Fig 2. All servers to Tor switches have a link bandwidth of 10 Gbps. Core switches to aggregation switches and aggregation switches to Tor switches both have a link capacity of 40 Gbps. All links have a propagation delay of 1us, so the maximum base RTT within the DC is 12us. All switches have a per port buffer of 1MB. At the same time, to simulate inter-DC traffic, we set up each two identical DCNs connected via WAN link. We set the WAN link delay between the DCN 50ms, and the link capacity 500Mbps.

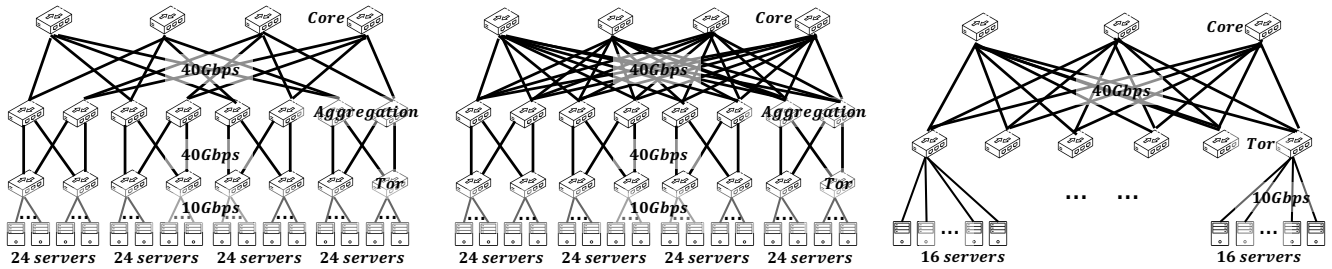
Workload: We generate the load based on the real traffic distribution in the web search workload [3] and the workload of the facebook hadoop cluster [4]. We generate intra-DC and inter-DC traffic in the ratio of 6:1.

Baseline and metrics: We compared IDCC with Annulus, Gemini, DCBR, and L2BR. In IDCC , the queuing threshold within the DC is $qLen_T = 50pkts$ and the delay threshold is $rttT_=rtt_base + 5ms$. The weight function $f(BS)$ decreases linearly with the sent bytes with the range of 0 to 2. Our evaluation metrics include the average FCT of intra-DC, the average rate of inter-DC traffic.

Large-scale Simulations: We perform simulations in three topologies and merge the ultimate results. Fig. 3 show the average FCT of the intra-DC flows under different schemes for web search workloads with loads ranging from 30% to 70%. we break down the results into short[0, 100KB], medium[100KB, 1MB], and long($>$ 1MB) flows.

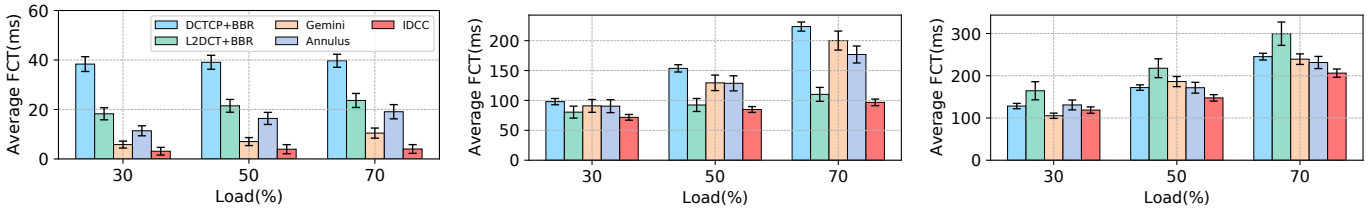
IDCC has the most significant improvement in FCT for short flows, with an FCT degradation of 12, 6, 1.9 and 3.6 $\times$ for web search workloads compared to DCBR, L2BR, Gemini and Annulus. This is because IDCC provides optimization for the performance of short flows. When congestion occurs within the DC, the long flows concede bandwidth to the short flows, thus speeding up the completion of the short flows. Simultaneously, IDCC can guarantee persistent shallow caching because of better convergence speed and stability, which contributes greatly to short flows. The effect of IDCC on the optimization of the FCT of the flows decreases as the size of the flow increases. Even so, the FCT can still be reduced by 10% to 26% for medium size flows compared to other schemes. We measured the average send rate per flow for inter-DC traffic under different workloads, and the results are shown in Fig. 4. There is a small improvement of 9% to

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:42:24 UTC from IEEE Xplore. Restrictions apply.



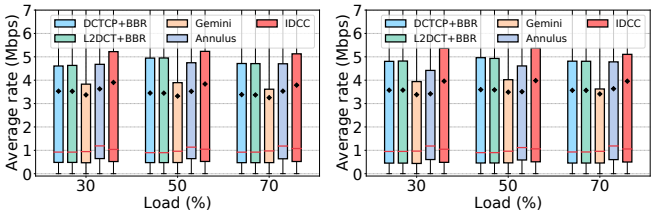
(a) Fat-tree topology (b) VL2 topology (c) Spine-leaf topology

Fig. 2. DC Topology



(a) Short(< 100KB) (b) Medium(100KB, 1MB) (c) Large(> 1MB)

Fig. 3. Average flow completion times of intra-DC for web search workload



(a) Web search (b) Facebook hadoop

Fig. 4. Average rate of inter-DC flows

16% in the average rate for IDCC compared to the congestion control algorithms Annulus, Gemini under facebook hadoop workload and 7% to 15% under web search workload.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their valuable comments. The research is supported by the Natural Science Foundation of China through projects 62002009.

REFERENCES

[1] A. Saeed, V. Gupta, P. Goyal, M. Sharif, R. Pan, M. Ammar, E. Zegura, K. Jang, M. Alizadeh, A. Kabbani *et al.*, "Annulus: A dual congestion control loop for datacenter and wan traffic aggregates," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020, pp. 735–749.

[2] G. Zeng, W. Bai, G. Chen, K. Chen, D. Han, Y. Zhu, and L. Cui, "Congestion control for cross-datacenter networks," *IEEE/ACM Transactions on Networking*, 2022.

[3] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *Proceedings of the ACM SIGCOMM 2010 Conference*, 2010, pp. 63–74.

[4] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 123–137.

[5] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time," *Queue*, vol. 14, no. 5, pp. 20–53, 2016.

[6] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *ACM SIGOPS operating systems review*, vol. 42, no. 5, pp. 64–74, 2008.

[7] S. Zou, J. Huang, J. Liu, T. Zhang, N. Jiang, and J. Wang, "GTCP: Hybrid Congestion Control for Cross-Datacenter Networks," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2021, pp. 932–942.

[8] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[9] A. Munir, I. A. Qazi, Z. A. Uzmi, A. Mushtaq, S. N. Ismail, M. S. Iqbal, and B. Khan, "Minimizing flow completion times in data centers," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2157–2165.

[10] R. Pan, L. Breslau, B. Prabhakar, and S. Shenker, "Approximate fairness through differential dropping," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 23–39, 2003.

[11] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "Pie: A lightweight control scheme to address the bufferbloat problem," in *2013 IEEE 14th international conference on high performance switching and routing (HPSR)*. IEEE, 2013, pp. 148–155.

[12] P. Taheri, D. Menikkumbura, E. Vanini, S. Fahmy, P. Eugster, and T. Edsall, "Rocc: robust congestion control for rdma," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 2020, pp. 17–30.

[13] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 51–62.

[14] M. Alizadeh and T. Edsall, "On the data path performance of leaf-spine datacenter fabrics," in *2013 IEEE 21st annual symposium on high-performance interconnects*. IEEE, 2013, pp. 71–74.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:42:24 UTC from IEEE Xplore. Restrictions apply.