Survey on Congestion Control Frameworks in Data Center Networks

K. Arun Selvi[1] and Dr. K. Kumar[2]

[1] Department of Computer Science and Engineering, Government of Technology, Coimbatore. mail: aruncse24@gmail.com
[2]Department of Computer Science and Engineering, Government of Technology, Coimbatore. E-mail: kkumar@gct.ac.in

Corresponding author: K. ArunSelvi, aruncse24@gmail.com

Abstract– Cloud datacenters which implements scale out distributed storage and computing frameworks such as Hadoop, Map Reduce, HDFS, Cassandra, etc. commonly have many-to-one communication pattern called incast. This pattern relies heavily on TCP and begins when a singular parent server places a request to a cluster of nodes simultaneously, leading to microburst of many machines concurrently sending TCP data streams to one machine (many-to-one). The synchronicity of such incast traffic with other throughput-demanding elastic traffic flows in Data Center Networks (DCNs) causes severe performance degradation in applications such as web search, maps, social networks, data warehousing and analytics. It also brings poor Quality of Service (QoS) and enterprise revenue loss. Over the past few years, many promising proposals have been put to address this incast problem. This paper made an in-depth survey on these proposals and summarizes the principles, merits and applications of the main proposals for solving the TCP incast congestion problem.

Keywords–Data Center Networks, Receiver-oriented Congestion Control, Software Defined Networking, Incast, Congestion Control, TCP.

1 Introduction

1.1 Data Center Network

Data Center Network is a communication network which interconnects pool of resources such as computational, network and storage in a datacenter (Fig. 1). DCN runs tightly coupled computing tasks that must be responsive to users, e.g., thousands of backend computers (servers) may exchange information to process the user's request and all of the transfers and complete response to the user must be satisfied within 100 ms [1]. In order to meet these requirements, DCN offers high bandwidth (>>Gbps) and low latency

(<<ms)environment since even a small delay in transmission can cause degradation in applicationperformance.

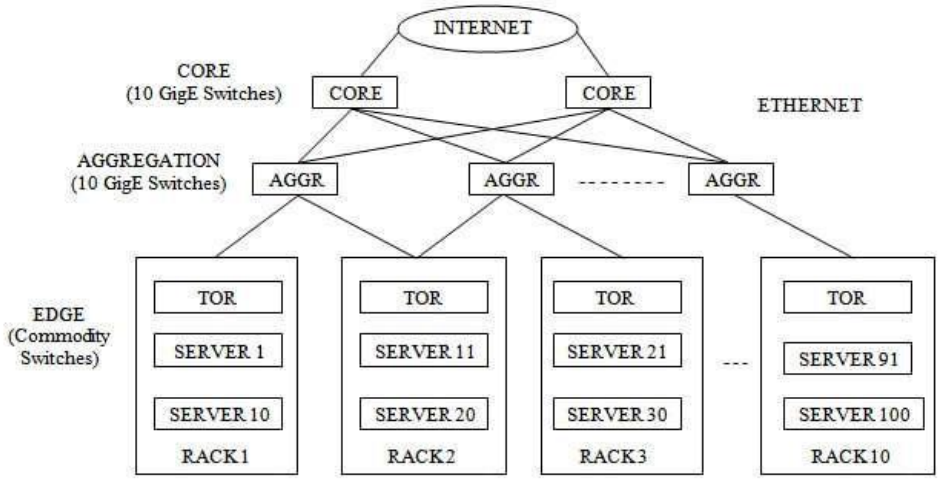

Figure 1Conventional Data Center NetworkArchitecture

The purpose of modern DCN is to accommodate multiple datacenter tenants with a variety of workloads. In such a network, servers are the components that provide users (and the programs working on their behalf) with requested services [2]. The simplest such networking services may be response to API function calls. Servers may also provide users/clients with applications, by way of web protocols, language platforms, or virtual machines that provide users with full desktops.

The emergence of cloud-based applications in systems such as iCloud, Dropbox, Facebook and Amazon EMR attracts an increasing number of users and extent their system positioning on DCN [3]. For example, Facebook announced that they had Hadoop clusters with 100 petabyte (PB) data across more than 50,000 servers [4]. Such cloud applications in DCN generate a larger number of traffic that ranges from barrier-synchronized, short-lived flows such as web searches (mice) to bandwidth-inclined, long-lived flows such as backups and virtual machine migration (elephants)[5]. Recent studies showed that DCNs are crowded with mice, in terms of the larger volume of such traffic goes to the elephants.

DCN uses Ethernet switches with small buffers for interconnecting the servers. In the presence of such small buffers, the ubiquitous many-to-one traffic pattern poses challenges for TCP. It occurs on the partition-aggregator architecture (Fig. 2) where many barrier-synchronized work nodes transmit data to the aggregator node. This can easily cause the TCP incast congestion problem in DCN. Inincast, data flows experiences packet loss and long

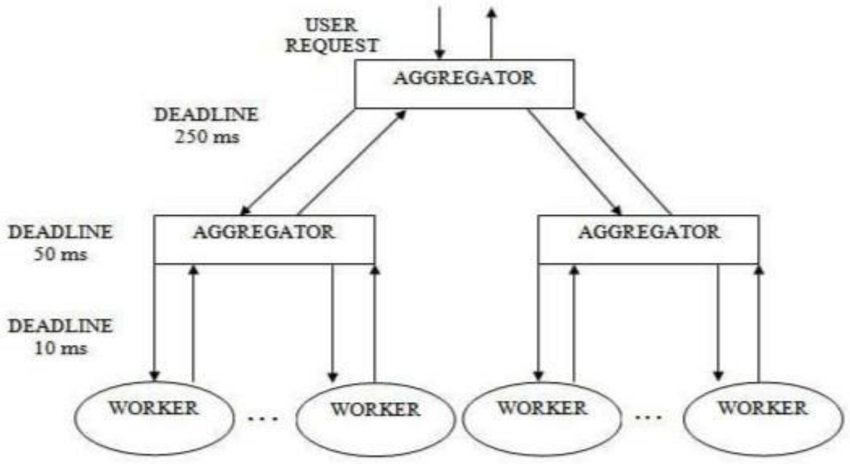

Figure 2 Partition-aggregate architecture

Flow Completion Times (FCTs) which brings users poor quality of service (QoS) and enterprise revenue loss [6,7].

In order to mitigate this problem, effectivecongestion control frameworks such as sender, receiver, switch- assisted and SDN- based have been proposed to enhance data flows in DCNs. This paper made a survey on these frameworks and provides application-adaptable proposalsfor TCP incast congestion problem. In this, section 2 analyzes different research literature works towards incast congestion. Section 3 deals with the survey on various protocols under receiver and SDN based approaches for solving many-to-one traffic pattern in DCNs. Section 4 tabulates the merits and demerits of various outcomes adopted .Section 5 deals with the area of applications and finally section 6 gives the overall interpretation and future aspects.

2. Methodologies for solving TCP Incast congestion problem

There are many promising proposals adopted by researchers to address the incast congestion problem. In general, these works are categorized into:

2.1 Sender-based: There is a mismatch observed between TCP timeout timers in the hosts and the actual round trip times (RTTs) sensed in DCNs [8]. By using high-resolution timers in the sender TCP stack, TCP timeout detection is enabled even in the microsecond particulate. DCTCP notifies congestion degree by ECN packets and suppress queue buildups of switches to moderate congestion [9]. RED-AQM parameters are strained to accomplish a small ECN-marking threshold for obtaining a small queue length. Both approaches achieve smaller delay for mice flows but require modifications of TCP sender and receiver algorithms as well as fine tuning of RED parameters at the switches.

2.2 Receiver-based: ICTCP utilizes throughput detection at receiver to handle incast traffic. The receive window of all low-RTT TCP connections are collectively adjusted to control throughput during congestion and the window adjustments happen proactively before packet losses occur [10]. The implementation and experiments in a real testbed demonstrate that ICTCP achieve almost zero timeout and high throughput for TCP incast. It is more effective when incast happens in the last hop of the switch at receiver side. But ICTCP does not address the impact of buffer buildups problem caused by the synchronicity of elephants in the same buffer as mice.

2.3 Switch-assisted: AQM schemes regulate TCP sending rate with minor modifications to DropTail AQM.Receiver window queue (RWNDQ) maintains queue occupancy within a predestined target level to achieve greater efficiency. Good fairness is observed in both long and short lived flows. IQM monitors TCP connection and tear down events at the switch to predict possible congestion events in the next few RTTs [11]. It reduces mice latency to maintain higher throughput for elephants. If congestion is forthcoming, the receiver window of ACKs are reset to 1MSS to slow down elephants making room for awaited incast traffic. All schemes are shown to curb timeouts and achieve higher throughput for elephants but require switch software modification.

2.4 SDN-based:SDN enables controller to find a longlived flow to slow down transmission rate by modifying the TCP receive window of ACK packet after OpenFlow-switch send a congestion message to controller [12].The modification is based on the openflow rules setup at the switches. Hence we can accurately reduce the rate of long-lived flow to ensure the performance of other flows. The experiments conducted in an emulation environment (Mininet) shows almost zero packet loss for TCP incast with greater effect on throughput of the elephants. Theproposed mechanisms are achieved without modifying the TCP sender and receiver algorithms and in-place commodity networking hardware switches.

2.5 Conclusion:Based on the inferences from the above methodologies, receiver-oriented and SDN based approaches can handle possible contention on the buffer space before the surge of incast traffic. In addition, both frameworks can achieve all improvements without any major changes in TCP algorithms and the networking hardware. Both approaches can obtain more accurate congestion information and helps in the reduction of completion time of mice flows that are known to constitute the majority of flows in data centers. Hence this paper made a survey on these congestion control frameworks to respond to congestion. The datacenter networks in this paper are regarding wired networks instead of wireless networks [13].

3. Literature Survey

The majority of congestion control transport protocols spring up based on TCP. The state-of-the-art TCP makes effective use of bandwidth by adjusting window sizes according to the Additive-Increase and Multiplicative-Decrease (AIMD) approach. The following presents typical TCP-based protocols based on the above approaches for congestion control in DCN to avoid incast problem in brief.

3.1 Congestion Control protocols for DCN

Mohammad Alizadehet al., proposed “DCTCP: Efficient Packet Transport for the Commoditized Data Center” [9]. In that, DCTCP uses Explicit Congestion Notification (ECN) and a simple multi-bit feedback mechanism at the host. In the datacenter, operating with commodity, shallow buffered switches, DCTCP provides the same or better throughput than TCP, while using 90% less buffer space. Further, a 10X increase in foreground traffic does not cause any timeouts, henceincast problems are largely eliminated. The main goal of DCTCP is to achieve high burst tolerance, low latency, high throughput and operating with very small queue occupancies, without loss of throughput.

M.Alizadehet al., proposed “pFabric: Minimal Near-Optimal Data center Transport” [14]. In that, pFabric produces optimal FCTs even at the 99th percentile for short flows, while still minimizing average FCT for long flows. It delivers this attainment with a key concept: datacenter transport should duple flow scheduling from rate control. pFabric’s realization in large scale testbed experiments shows that it has the ability to readily schedule flows based on their priorities which helps to achieve near-optimal performance for traffic scenarios with no bound as well as scenarios where there is a mix of bound and non-bound traffic.

Radhika Mittal et al., proposed “TIMELY: RTT-based Congestion Control for the Data center” [15]. TIMELY is implemented in host software operating over NICs with OS-bypass capabilities. It modifies transmission rates based on RTT measurements to keep packet delay low while having high bandwidth delivery. It takes recognition of modern NIC support for timestamps and fast ACK turnaround for performing congestion control based on definite RTT measurements. TIMELY can encounter and respond to tens of microseconds of queuing to deliver low packet delay and high throughput, even in the presence of uncommon RTT signals and NIC offload.

3.2 Congestion Control Protocols Based On Receiver Approach

Haitao Wu proposed et al.,"ICTCP: Incast Congestion Control for TCP in Data Center Networks"[10]. ICTCP is a receiver window based congestion control algorithm. It uses the concept of window adjustment to control congestion i.e., all low-RTT TCP connections are jointly fine-tuned to control throughput during incast connection. The adjustment is based on the ratio of difference of obtained and expected per connection throughputs over conventional ones, as well as the last-hop available bandwidth to the receiver. It improves the performance of TCP for incast in datacenter networks. ICTCP is implemented on a real testbed with 47 servers together with a 48-port Ethernet Gigabit switch. It is effective in avoiding congestion by achieving almost zero timeout for TCP incast, and it provides high performance and fairness among challenging flows.

V. Tsaoussidiset al., proposed ‘‘TCP-Real: receiver-oriented congestion control"[16]. TCP-Real controls congestion as standard TCP, in addition to that it also uses "wave" communication pattern which allow for two enhancing mechanisms: (i) congestion avoidance, which lowers unnecessary transmission gaps that reduces the performance of time-constrained applications, and (ii) advanced error detection and classification, which designates recovery tactics responsive to the nature of the errors. Monitoring the level of contention permits the receiver to rule on the cause of packet drops. It was used to demonstrate the potential of this mechanism in multiplexed wired/wireless channels and with delay-tolerant and intolerant applications.

Wei Baiet al., proposed "PAC: Taming TCP Incast Congestion using Proactive ACK Control"[17]. Proactive ACK Control is an effective design to control TCP incast congestion at the receiver. It treats ACK not only as the acknowledgement of received packets but also as provoke for new packets. Considering the characteristics of datacenter network, this novel ACK control passes ACKs in a way that the ACK-provoked in-flight data can entirely utilize the bottleneck link without making incast collapse even when the senders are above thousands. PAC is the first protocol to prove that the receiver is capable of avoiding incast traffic.

Lei Xuet al., proposed"Enhancing TCP Incast Congestion Control Over Large-scale Data center Networks" [18]. For enhancing TCP incast congestion control, Receiver-oriented Datacenter TCP (RDTCP) is proposed. It is motivated by oscillatory queue size when handling heavy incast traffic and substantial potential of receiver in congestion control. It adopts both open- and closed-loop congestion controls. RDTCP is implemented at TCP receiver side. Its receive window size rwndwill be eventually sent off to sender through the advertisement window in TCP header. And the RDTCP sender adopts only rwndas its unique congestion window to send data. Evaluation results proved that RDTCP has the minimum 99 percentage FCTs.

Lei Xuet al., proposed “Throughput Optimization of TCP incast congestion control in large-scale data center networks”[19].This paper designs a novel transport protocol in congestion control for heavy incast to satisfy requirements from large-scale datacenter networks. RCC leverages both an open-loop congestion control, i.e., centralized scheduler for suppressing the burstiness of incast, and a closed-loop congestion control, i.e., ECN, at receiver to achieve normal congestion control in data centers. It is implemented in ns3 and the evaluation results indicate that RCC has an average decreases of 47.5% in the mean queue size and 51.2% in the 99 percentage latency in the heavy incast over TCP.

3.3 Congestion Control protocols based on SDN Approach

Yifei Lu et al.,proposed“SDN-based TCP Congestion Control in Data Center Networks”[12]. SDN-based TCP (SDTCP) is a congestion control mechanism at network side. It enables controller to select elephants (longlived flow) to reduce sending rate by adjusting the TCP receive window of ACK packet after OpenFlow-switch triggered a congestion message to controller. SDTCP is designed to reduce throughput of background flows to guarantee bursty flows which are usually more important. It deals with TCP incast problem excellently as it guarantees the throughput of burst flows and long-lived flows at the same time with no packet loss.

Ahmed M. Abdelmoniemet al., proposed “Reconciling Mice and Elephants in Data Center Networks” [20].Reconciling of mice and elephants mechanism is a classic congestion control in flow-aware networks such as ATM-ABR, where the switch sets a field in packet headers to enforce the sending rate at the source. This approach overwrites Rwndand the scaling option value n to indicate the bottleneck fair share of bandwidth available for a given flow on a given path between the source and destination. As the ACK from a receiver traverses the switches in the reverse path towards the sender, each switch examines the packet and modifies the Rwndvalue taking into account the window scaling value if necessary.

Ahmed M. Abdelmoniemet al., proposed “Incast-Aware Switch-Assisted TCP Congestion Control for Data Centers”[11].Switch-assisted TCP congestion control (ICTCP) is proposed to address the congestion problems of incast traffic and its interaction with other elastic traffic in DCNs. ICTCP makes use of Incast Queue Management (IQM) algorithm, an event-drivenmechanism which extends the simple drop-tail queue managementwith two major event handlers: packet arrivals and incast detection timer expiry to trigger window updates. ICTCP supports short-lived incastflows, that are known to constitute the majority of flows in data centers. It has achieved small flow completion times for incast traffic without impairing the throughput of elephant flows.

Simon Jouetet al., proposed "OTCP: SDN-Managed Congestion Control for Data Center Networks" [21]. Omniscient TCP (OTCP), a SDN approach can compute environment-specific congestion control parameters based on centrally available network properties.With these parameters, it calculates suitable TCP retransmission timers, RTO_{min} and RTO_{init}, as well as the initial and maximum congestion window size that matches the route Bandwidth Delay Product (BDP) between end-hosts. Under TCP incast, OTCP significantly outperforms TCP for short-lived, soft-real-time flows, with a 12× improvement in Flow Completion Time at the mean and 31× at the 95th percentile.

Ahmed M. Abdelmonieumet al., proposed "SICC:SDN-based Incast Congestion Control for Data centers"[22].SDN-based Incast Congestion Control (SICC) can handle possible contention on the buffer space before the surge of incast traffic. SDN framework provides useful statistics on the ongoing number of flows and the queue occupancy for each switch port. Hence, the SDN controller upon forecasting possible incast event, it can send a special message to the sending endhosts. In turn, end-hosts start rewriting the receiver window Rwndin the incoming ACKs to 1 MSS. Simulations and testbed experiments shows that it improves the flow completion times for incast traffic without impairing the throughput of elephant flows.

4. Benefits and Limitations of existing proposals

Table 1Benefits and Limitations of existing proposals

S.No.	Author	Title	Benefits	Limitations
1	M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye and P. Patel.	DCTCP: efficient packet transport for the commoditized data center	It alleviates queue build-up, buffer pressure and incast.	It does not focus on window adjustments during congestion.
2	Mohammad Alizadeh, Shuang Yang, Milad Sharif andSachinKatti.	pFabric: Minimal Near-Optimal Data center Transport	Shows how large buffersor complex rate control are largely unnecessary in data centers.	pFabric is not integrated withlatency-sensitive applications.

3	R. Mittal, V.T. Lam, N. Dukkipati, E. Blem, H. Wassel and M. Ghobadi.	TIMELY:RTT-based congestion control for the data center	Can detect and respond to tens of microseconds of queuing to deliver low packet latency and high goodput.	It does not focus on how effective RTTs continue to be for congestion control.
4	H. Wu, Z. Feng, C. Guo and Y.Zhang	ICTCP: incast congestion control for TCP in data center networks	Avoid congestion by achieving almost zero time-out for TCP incast.	Extension of ICTCP is an issue when sender and receiver are not in the same switch.
5	V.Tsaoussidis and C.Zhang	TCP-Real: receiver-oriented congestion control.	Wave pattern call for window adjustments prior to congestion.	Fail to achieve objectives: friendliness and efficiency.
6	Sam Liang and David Cheriton	TCP-RTM: Using TCP for Real Time Multimedia Applications	Ensures real-time applications be responsive to network congestion.	TCP-RTM is not incorporated in the standard implementation of TCP.
7	W. Bai, K. Chen, H. Wu, W.Lan and Y. Zhao	PAC: taming TCP incast congestion using proactive ack control.	PAC does not introduce spurious timeout and retransmission even when the measured 99 percentage RTT is only 3.6ms.	PAC does not touch TCP-related protocols.
8	Lei Xu, KeXu, Yong Jiang , FengyuanRen and Haiyang Wang	Enhancing TCP Incast Congestion Control Over Large-scale Data center Networks	In evaluations, RDTCP outperforms other protocols in terms of queue buildup, goodput, FCT, etc.	Less effective only if incast happens at sender side.
9	M. Ghobadi, S.H. Yeganeh, Y. Ganjali	Rethinking end-to-end congestion control in Software-Defined Networks	Adaption results in up to 59% improvement in FCTs while keeping the network stable.	Cannot support programmable and adaptive TCP congestion control.
10	Lei Xu, KeXu, Yong Jiang, FengyanRen, Haiyang Wang	Throughput Optimization of TCP incast congestion control in large-scale data center networks	Solves heavy incast in DCN that are expanding continuously.	Challenges are more in extending RCC from ns3 to real world.
11	Yifei Lu and Shuhong Zhu	SDN-based TCP Congestion Control in Data Center Networks	Using global perspective, rate of long-lived flow is decelerated accurately to ensure the performance of other flows.	Controller does not have priority to punish the flows.

12	Ahmed M. Abdelmoniem and BrahimBensaou	Reconciling Mice and Elephants in Data Center Networks	Persistent queue size is maintained to absorb incast burst.	RWNDQ is not having true deployment potential in real world datacenter networks.
13	Ahmed M. Abdelmoniem and BrahimBensaou	Incast-Aware Switch-Assisted TCP Congestion Control for Data Centers	Achieves small FCTs for incast traffic without impairing the throughput of elephants.	IQM is not having true deployment potential in real world datacenter networks.
14	MasoumehGholamiand BehzadAkbari	Congestion Control using OpenFlow in Software Defined Data Center Networks	Throughput enhancement and average packet delay reduction.	Suffers overhead problem due to single controller.
15	Simon Jouet, Colin Perkinsand DimitriosPezaros	OTCP: SDN-Managed Congestion Control for Data Center Networks	Improves performance of soft real-time, partition-aggregate applications.	End-to-end latency is not much reduced.
16	Ahmed M. Abdelmonieum, BrahimBensaou, Amuda James Abu	SICC:SDN-based Incast Congestion Control for Data Centers	No modification in TCP algorithms and the networking hardware.	Needs testing in an operational environment with realistic workloads and scaling.

5. Applications

Generally, the TCP incast problem could appear in clustersrunning applications such as search and batch computing jobslike MapReduce that follow the partition-aggregate processingmodel. In search applications, a server may query a largenumber of other servers the results to which may be returnedto that server at the same time creating sudden increase inincoming traffic. In a MapReduce job, a Reduce instance maydownload outputs of many Map instances for reduction whichcan cause a similar situation. Both scenarios follow the many-to-one communication pattern.When these applications are subjected to receiver and SDN based congestion control frameworks, the incast congestion problem could be mitigated and the throughput of corresponding data flows can be improved.

6. Conclusion and Future Work:

The trend of large-scale data center networks is indomitable now-a-days. Cloud based applications are attracting large number of users and scaling their system deployments on data center networks. The ubiquitous many-one-traffic pattern makeTCP face challenges from incast congestion problem. This paper made a survey on the promising proposals for avoiding this problem and provides Receiver and SDN based congestion control frameworks as the significant productive approach for TCP incast congestion control mechanism. The two frameworks considered have exact realization of real-world TCP. They suits higher-degree many-to-one communication patterns and has effectively suppressed congestion. The contribution is done without modifying the networking hardware to enable quick and true deployment potential in critical DCNs.

Future work involves practical aspects of RCC and SDN in at-scale simulations (ns3) in network scenarios with diverse and heavy workloads. Also work will be extended to provide transport performance evaluations regarding flow priority, dynamic traffic, multiple bottleneck topology as well as convergence and coexistence with TCP.

7. References

[1] Al-Fares, M.Loukissas, A. and Vahdat, A.(2008). A scalable, commodity data center to network architecture. *ACM SIGCOMM Conference on Data 3 Communication*, Seattle, WA, 63-74.

[2] Guo, C.Wu, H.Tan, K. Shi, L. Zhang, Y. and Lu, S.DCell, (2008). A scalable and fault tolerant network structure for data centers.*ACM SIGCOMM Computer Communication Review*, 75-86.

[3] Dropbox clears 1 billion file uploads per day. Received from http://cnet.co/YYwgB3

[4] How big is facebooks data? 2.5 billion pieces of content and 500+ terabytes ingested every day. Received from http://goo.gl/n8xhq

[5] Kandula, S. Sengupta, S.Greenberg, A.Patel, P. and Chaiken, R.(2009). The nature of data center traffic.*Proceedings of IMC*.

[6] Zhang, J.Ren F. and Lin, C.(2011). Modeling and understanding tcpincast in data center networks. *Proceedings of IEEE INFOCOM*, 1377–1385.

[7] Dukkipati, N.and McKeown N. (2006). Why flow completion time is the right metric for congestion control. *ACM SIGCOMM Comput*,59–62.

[8] Vasudevan, V. Phanishayee, A. Shah, H. Krevat, E. Andersen, D.G. Ganger, G.R. Gibson, G.A. and Mueller, B. (2009). Safeand effective fine-grained TCP retransmissions for data centercommunication *ACM SIGCOMM CCR*, vol. 39.

[9] Alizadeh, M. Greenberg, A. Maltz, D.A. Padhye, J. Patel, P.Prabhakar, B.Sengupta, S. and Sridharan, M. (2010). Data center TCP (DCTCP). *ACM SIGCOMM CCR*, vol. 40,63–74.

[10] Wu, H. Feng, Z.Guo, C. and Zhang, Y. (2013). ICTCP: Incast congestion control for TCP in data-center networks. *IEEE/ACMTransactions on Networking*, vol. 21, 345–358.

[11] Abdelmoniem, A.M. and Bensaou, B.(2015). Incast-Aware Switch-Assisted TCP congestion control for data centers. *IEEEGlobal Communications Conference (GlobeCom)*.

[12] Lu, Y. and Zhu, S. (2015). SDN-based TCP Congestion Control in Data Center Networks. *Proceedings of IEEE IPCCC*.

[13] Cui, Y. Wang, H. Cheng, X.Chen, B. (2011). Wireless data center networking, *IEEE Wireless Commun.18*, 46–53.

[14] Alizadeh, M. Yang, S. Sharif, M. Kattin, S. McKeown, N. Prabhakar, B. Shenker, S. (2013).Pfabric: minimal near-optimal data center transport, *Proceedings of ACM SIGCOMM* 43, 435–446.

[15] Mittal, R.Lam, V.T. Dukkipati, N.Blem, E.Wassel, H.Ghobadi, M. Vahdat, A. Wang, Y. Wetherall, D. and Zats,D. (2015). Timely: rtt-based congestion control for the data center, *Proceedings of ACM SIGCOMM*, 537–550.

[16] Tsaoussidis, V. Zhang, C. (2002).Tcp-real: receiver-oriented congestion control. *Computer Networks* 40, 477–497.

[17] Bai, W.Chen, K.Wu, H.Lan, W.Zhao, Y.(2014). Pac: taming tcpincast congestion using proactive ack control. *Proceedings of the 2014 IEEE 22nd International Conference on Network Protocols*, 385–396.

[18] Xu, L.Xu, K. Jiang, Y.Ren, F. Wang, H.(2015). Enhancing tcpincast congestion control over large-scale data center networks. *Proceedings of IEEE IWQoS*, 225–230.

[19] Xu, L. Xu, K. Jiang, Y. Ren, F. Wang, H. (2017). Throughput optimization of TCP incast congestion control in large-scale data center networks. *Computer Networks*124,46-60.

[20] Abdelmoniem, A.M. and Bensaou, B.(2015). Reconciling mice and elephants in data center networks. *In IEEE 4th International Conference on Cloud Networking CloudNet*.

[21] Jouet, S. Perkins, C. and Pezaros, D.(2016). Otcp: Sdn-managed congestion control for data center networks. *In NOMS IEEE/IFIP Network Operations and Management Symposium*, 171–179.

[22] Abdelmoniem, A.M. Bensaou, B. and Abu, A.J. (2017). SICC:SDN-based incast congestion control for data centers. *IEEE ICC SAC Symposium Cloud Communications and Networking Track*.