TCP Incast Congestion Control in Data Center Networks: A Survey

Anonymous Author(s)

ABSTRACT

Transport Control Protocol (TCP) incast happens when multiple synchronized servers send data to the same receiver in parallel. TCP incast congestion has been a severe performance issue in high-bandwidth and low-latency networks. For example, it can introduce hundreds of milliseconds delay and up to 90% throughput degradation to the underlying networking system.

In this paper, we present a comprehensive literature review of TCP incast congestion control. We first study the basic concepts in TCP incast in the settings of data center networks, and then systematically characterize the old fashions and the state-of-the-art works. We characterize the existing techniques and summarize the general challenges in the scenario. We hope this research can shed some light on future TCP incast algorithm design in data center networks.

ACM Reference Format:
Anonymous Author(s). 2021. TCP Incast Congestion Control in Data Center Networks: A Survey. In *Large Scale Data Processing Systems.* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Data centers are crucial in today's Internet business like web searching and video streaming. Data center networks (DCNs) interconnect the data centers to better utilize the computational resources (*e.g.,* network, storage) to service the end-users. Large service providers lay particular emphasis on the data center network architecture design and optimizations to offer better user experiences with high speed and low latency. For example, Facebook develops a "5-post" data center network architecture that eases the resource management, network monitoring and measurements [4].

TCP incast congestion is a critical performance problem in data center networks by increasing response time for certain impacted network requests. In high-bandwidth low-latency data center networks with shallow buffers, the transport layer TCP is sensitive to package congestion collapses when multiple senders send data simultaneously (*i.e.,* many-to-one communications) [1]. Network package loss happens once the buffer is overflowed by the data sent from the massive senders. The network latency can thus dramatically increase to several hundred milliseconds. As a result, the performance of the underlying applications especially those involving barrier-synchronized communications (*e.g.,* MapReduce [3], Spark [5]) are severely degraded [2].

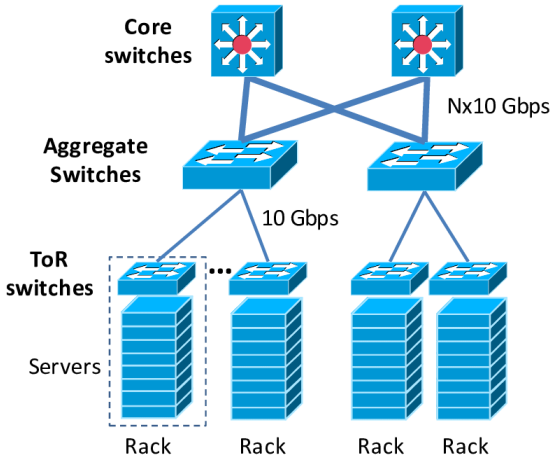

Figure 1: The typical architecture of data center networks.

Understanding prior solutions to tackle TCP incast congestion is important. However, to the best of our knowledge, there current exists little study about the problem. First, the performance impacts of the problem are understudied; second, the strengths and weaknesses of existing solutions have not been well characterized and summarized. To this end, we investigate how TCP incast congestion problem is solved or mitigated in the best practices. We aim to learn lessons o improve the state-of-the-art works and to shed some light on future research.

2 BACKGROUND

In this section, we present the basic background knowledge of TCP incast congestion in data center networks. Specifically, we first introduce the general archtecture of data center networks in §2.1. Next, we present the TCP congestion control mechanisms in §2.2

2.1 Data Center Networks

2.2 TCP Congestion Control

REFERENCES

[1] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center TCP (DCTCP). In *Proceedings of the ACM*

CSCI '5570, Spring 2021, CUHK
2021.

SIGCOMM. New Delhi, India.

[2] Wei Bai, Kai Chen, Haitao Wu, Wuwei Lan, and Yangming Zhao. 2014. PAC: Taming TCP incast congestion using proactive ACK control. In *IEEE 22nd International Conference on Network Protocols*.

[3] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M Hellerstein, Khaled Elmeleegy, and Russell Sears. 2010. MapReduce online. In *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. San Jose, CA.

[4] Nathan Farrington and Alexey Andreyev. 2013. Facebook's data center network architecture. In *2013 Optical Interconnects Conference*. Citeseer.

[5] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J Franklin, Scott Shenker, and Ion Stoica. 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. San Jose, CA.