Abstract

Today, data centers are everywhere and constitute a significant portion of the network. In general, data centers concentrate large numbers of servers together and have massive networking requirements. The vast majority of this network traffic runs over TCP. In contrast to conventional networks, data center networks have a high delay-bandwidth product. However, for networks with a high delay-bandwidth product, TCP has some significant drawbacks. In the past few years, Several TCP updates have been developed to improve TCP in data center networks. In this survey paper, we will discuss the problems with TCP in data center networks, such as TCP Incast, Timeout issues, Queue accumulation in switches, and delays in data transfer. We will then analyze three existing data center TCP protocols, aka DCTCP, D2TCP, and ICTCP, with their benefits and drawbacks in relation to data center networks.

Keywords

TCP, Data center networks, data center network, DCTCP, D^2TCP, and A^2DTCP

I. INTRODUCTION

1.1 Data Centre Networks

For IT businesses, data centers are increasingly crucial computing platforms for real-time responses while performing various tasks, including system configuration and query services. Data center networks differ from typical IP networks in a number of ways, including their regular architecture, freedom to choose the endpoints of traffic flows, many-to-one communication patterns with high capacity, and low round trip times (RTT). Today data centers are ubiquitous and store enormous amounts of data and accommodate hundreds of thousands of users. These Data centers host multiple online services over the data, like querying the data in parallel, storing distributed files, and distributed execution engines. Infrastructure services like MapReduce, Hadoop file system, and OpenMP, are also hosted in data centers. Today's data centers use commodity switches for horizontally scaling the interconnected network and to create storage that is connected via high-speed links to the internet and the world.

The architecture of a modern-day data center network is seen in Figure 1. This three-tier architecture is made up of three layers of switches. The first layer of switches is edge switches that connect to end hosts to send/receive data to/from clients and servers. The second layer of switches is the core switches at the root. Finally, the third layer of switches is aggregation switches in the center. The key benefit of this design is that it is easy to scale and fault tolerant. The edge switches often offer connections to thousands of servers that are housed in racks and connected by 1 Gbps lines. For reliability and fault tolerance, these switches are further linked to a number of aggregate switches via 10 Gbps connections. The core switches are the ones that secure connections in the data center and transmit the required data from the servers to the clients. Core switches are directly connected to aggregator switches.

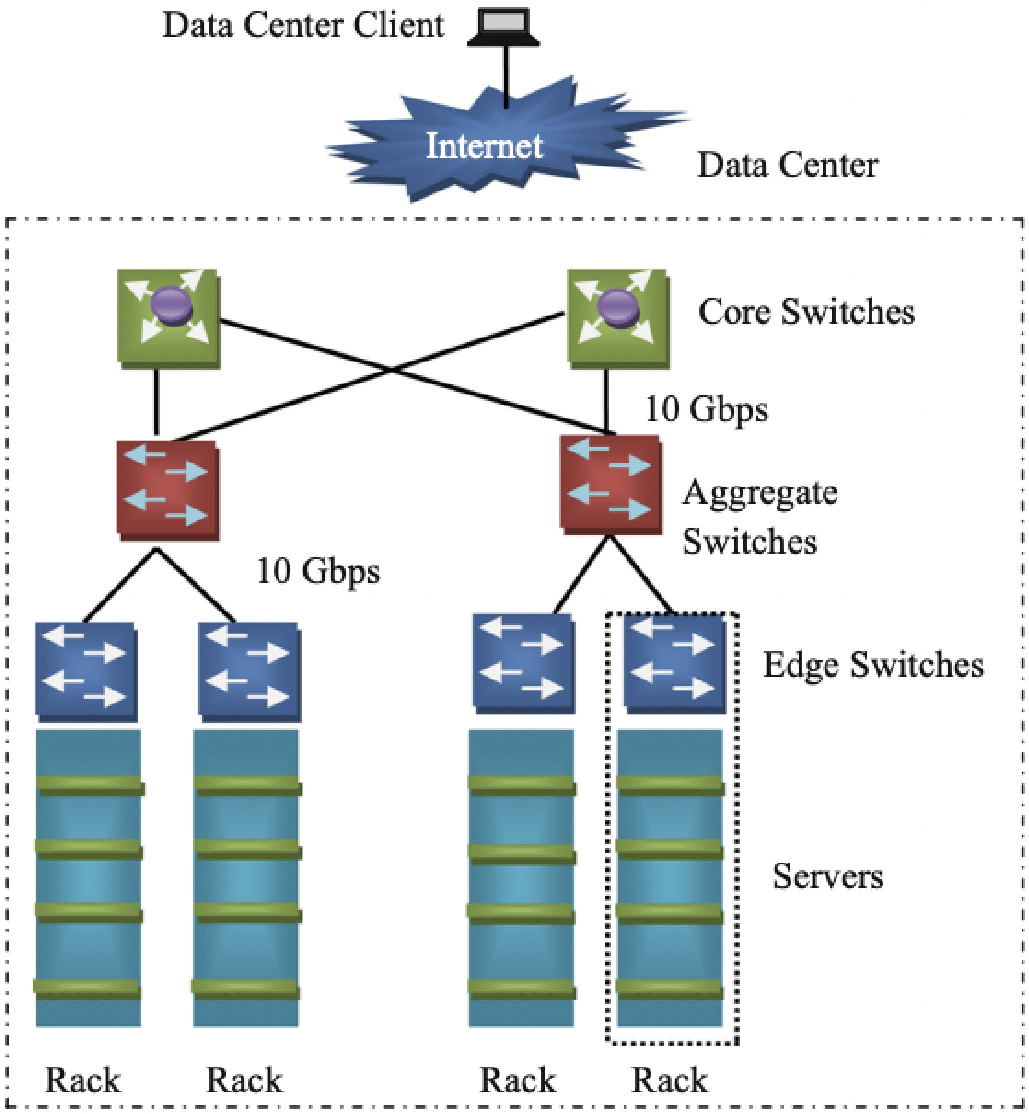

Fig 1 Architecture for current Data center networks

1.2 Motivation

Data centers today face challenges related to productivity, reliability, and adaptability due to a significant increase in their computing power, storage capacity, and the number of interconnected servers. Since all applications run on these data center networks are distributed in nature, the communication network significantly impacts the performance of data center networks. Data center networks see a lot of intermittent and bursty traffic. So the traffic is very unpredictable in terms of required throughput and latency. Datacenter applications typically result in small control flows and large data flows, both of which need high throughput and minimal latency. Transmission Control Protocol, aka TCP, however, doesn't perform well for networks with large delay-bandwidth products. Therefore transport protocol (TCP) in these data centers requires various desired traits, such as quick convergence and infrequent packet losses, in order to function well given the unique characteristics of data center networks. However, these characteristics of data center networks are not met by ordinary TCP. This is due to the fact that TCP encounters certain significant problems when it is employed in data center networks, including TCP Incast, TCP Outcast, latency, and packet reordering. For data center networks, a number of solutions have lately been put up to address these problems.

Transmission Control Protocol (TCP) is a tried-and-true transport protocol for the Internet, but data center network issues, including bandwidth constraints and a shortage of buffer space in switches, have prompted some to suggest methods that improve TCP execution for these data centers. The performance of TCP suffers when the delay-bandwidth product rises. One packet per RTT is the maximum amount of spare bandwidth that TCP can gain under its additive increase mechanism. Since a single flow across very high-bandwidth networks may use thousands of packets in the bandwidth-delay product, TCP may squander thousands of RTTs, ramping up to full utilization after a burst of congestion. Furthermore, the transfer latency of short flows is not made better by the increase in connection capacity (the majority of the flows in the Internet). Even when there is adequate bandwidth, short TCP flows cannot acquire it quicker than a "slow start" and will waste important RTTs ramping up.

The worldwide data center industry has grown fast in recent years. Different internet services made possible by data centers have encroached on every aspect of daily life. Data center network operators like Google, Facebook, Bing, etc continue looking for ways to speed up response times with the objective of maintaining a good customer experience. In general, a client request in a data center may cause a number of streams to be instantaneously formed at the back end of some data center networks. These back-end flows are allocated communication SLAs (Service Level Agreements) ranging from 10 milliseconds to 100 milliseconds to improve user experience. If a few flows in this communication process miss their deadlines, an aggregator (a server that monitors reactions on the flows), would not acknowledge the information they have transmitted. Missing an aggregator's inactivity restriction is seen as an SLA violation, in addition to causing a negative customer experience. Data center operators must update the deadline-uncertain service execution by enhancing the existing system protocols, such as TCP, in order to adopt the exceptional communication environment.
The modern-day architecture of data center networks is shown in Figure 1.

This article examines the new transport protocols that have been suggested as a means of reducing TCP problems in data center networks. This article has three main goals: first, it discusses the problems with TCP in data center networks; second, it introduces various transport layer solutions; and third, it compares and discusses the difficulties with existing solutions that have been suggested to enhance TCP performance in data center networks. The rest of the survey is divided into the following sections. We provide a thorough explanation of the TCP problems in data center networks in Section 2. As a means of addressing TCP's problems, several TCP variations have been proposed for data center networks. In Section 3, we describe these variants and compare them to other transport protocols using certain key performance indicators. In Sect. 4, we discuss the shortcomings of the research discussed in the survey, and possible future scope. We finally conclude the paper in section 5.

2 TCP RELATED ISSUES IN DATA CENTER NETWORK

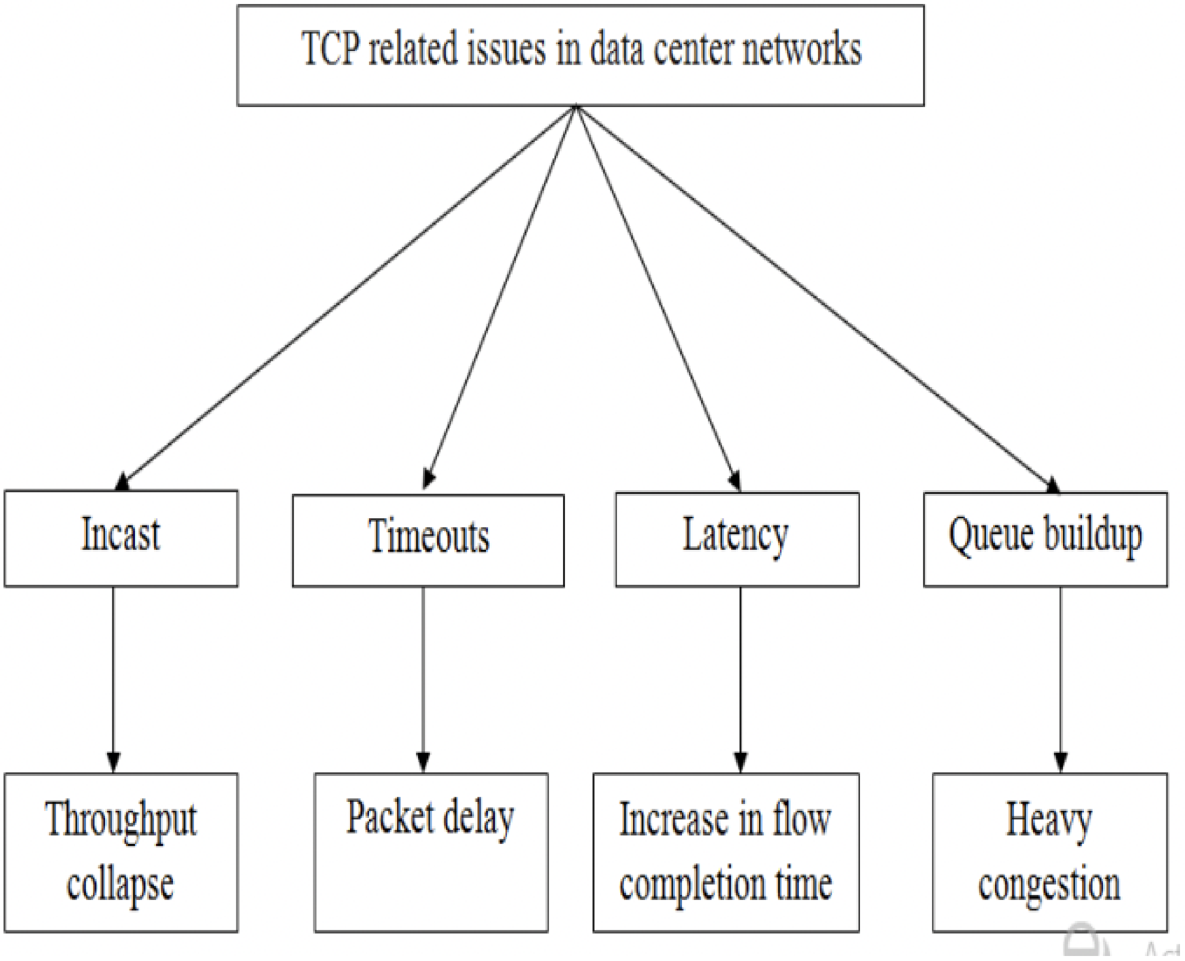

Fig 2 TCP Related issues in Data Center Networks

Due to its distinct characteristics when compared to other IP networks, TCP's performance is not acceptable in data center networks. In this section, we explore the substantial problems that TCP encounters when used in data center networks, as depicted in **Figure 2**.

2.1. TCP Incast

Due to the data center's unique characteristics compared to other IP networks, TCP's execution is unsuitable in a data center. One of the major execution problems in data center networks is TCP Incast. When a large number of servers send data simultaneously to a specific recipient with a high data transfer capacity and a short rounding time, it results in a terrible throughput collapse. It has been described as TCP's uncompromising behavior, which causes the connection limit in many-to-one communication schemes to be underutilized.

TCP Incast was initially discovered in the Big Data-driven personal protective equipment stockpiling framework. In this many-to-one correspondence design, the customer sends boundary-synchronized information demands to various servers via a switch in order to improve performance, and constantly achieve good quality In other words, the client waits to request data again until the majority of the senders have satisfied the client's initial request. Every server keeps a square of permitted information known as a server request unit (SRU). The TCP Incast throughput collapse problem occurs when all servers take roughly the same amount of time to provide the required data to the client. This causes buffers at the bottleneck connection to overflow, which causes significant packet losses and timeouts. Refer to **Figure 3** to see incast issue of TCP in data center networks.

The preconditions for TCP Incast are outlined in, where they are given in the following order:

- Tiny switch buffer networks with high bandwidth and low latency.
- Clients that simultaneously send barrier-synchronized requests.
- Servers that respond to requests with a small quantity of data.

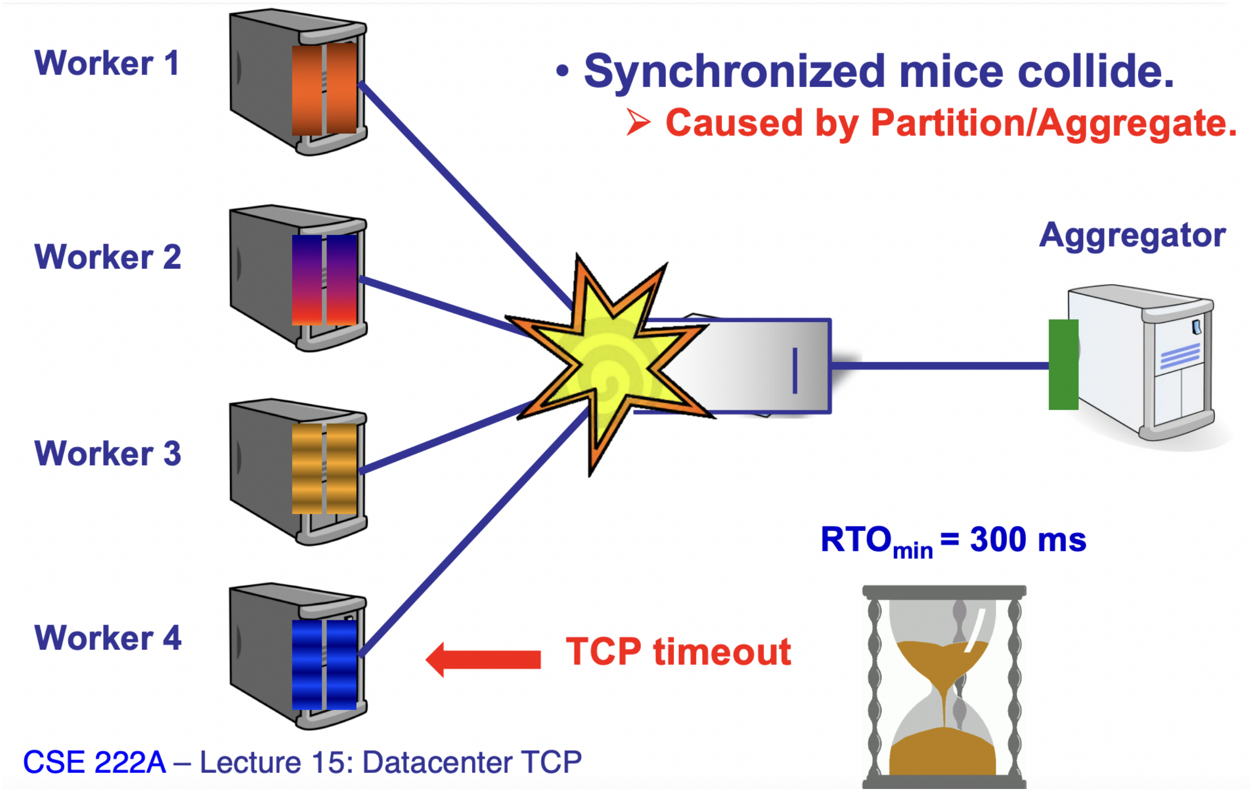
Worker 1
Worker 2
Worker 3
Worker 4
• Synchronized mice collide.
➢ Caused by Partition/Aggregate.
Aggregator
RTO_{min} = 300 ms
TCP timeout
CSE 222A – Lecture 15: Datacenter TCP

Fig 3 TCP Incast problem in Data Center Networks

2.2. TCP Timeouts

The root cause of the incast problem in data center networks is TCP timeouts. Rapid information transfers from several servers basically fill switch buffers which results in a flood of packet failures. This multiple packet failure causes TCP timeouts. In networks with a round trip length of 10 or 100 microseconds, these timeouts cause delays that can reduce performance by 90%. Additionally, the continual timeouts might impair the way that data center applications are run. The main causes of timeouts in the data center are the loss of packets during retransmission and the loss of packets from the beginning and the end of data blocks. TCP timeouts' causes are depicted in Figure 3. As seen in Figure 4, the loss of packets from the tail of data blocks (LTTO), from the head of data blocks (LHTO), and the loss of retransmitted packets (LRTO) are the major causes of timeouts in data center networks.

Lack of adequate duplicate acknowledgments results in LTTO timeouts. A timeout caused by packet losses from the tail of a data block is illustrated in Figure 4a.

As illustrated in Fig. 4b, if all of the packets from a data block are dropped, the receiver is unable to provide any acknowledgments for the data that was delivered. LHTO timeouts result from this. As a result, after a prolonged idle period following the end of the retransmission timer, the TCP sender might identify a packet loss.

Timeouts resulting from the loss of retransmitted packets, also known as LRTO, are another significant form of timeout. When a packet is lost, the TCP sender promptly retransmits it using the TCP loss detection method. The sender must wait until the retransmission timeout expires if the retransmitted packet is lost once again, as seen in Fig. 4c, since there aren't enough duplicate acknowledgments.

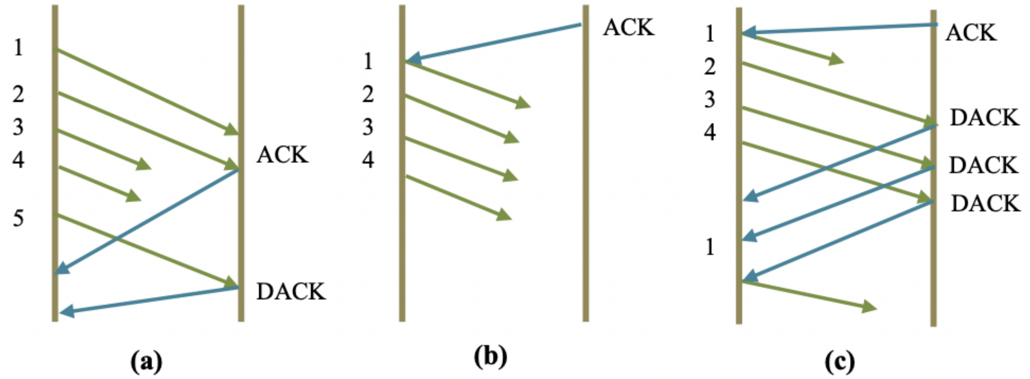

Fig 4 Different forms of TCP timeout:
(a) LTTO, (b) LHTO, (c) LRTO

2.3. Latency

In a data center network, latency is another problem. The major cause of latency in the data center network is the lengthy queuing time in switches that characterizes data center TCP traffic. Data Center Networks support both short-lived and long-lived TCP flows, with a typical range of 2 KB to 100 MB. Short-lived flows, in this case, are latency conscious, however, long-lived flows are not, which might transfer a lot of traffic and requires the growth of the bottleneck queue in order to prevent packet loss. Finally, when big and small flows separate a bottleneck queue, the large flows' queue-building causes the short flows' practice latency to increase. This leads to a significant number of packet losses and frequent retransmissions. Additionally, the majority of traffic in data center networks is bursty, which causes TCP flows to lose short-lived packets regularly. Moreover, the bulk of traffic in data center networks is bursty; as a result, packets of transient TCP flows are commonly lost.

2.4. Queue Buildup

Small, medium, and large volumes of traffic coexist in a data center network due to the diverse nature of cloud services. The network becomes very congested, and the bottleneck buffer overflows due to the persistent and self-centered character of the huge traffic. As a result, when heavy and small traffic uses the same route, the presence of the large traffic significantly degrades the performance of the Small traffic.

Due to the presence of large traffic flows, the following types of small traffic flows are worsened:

- Large traffic takes up the majority of the buffer, dropping packets from small traffic with a high likelihood. Similar problems occur when using TCP Incast since timeouts, and frequent packet losses significantly degrade the performance of small traffic flows.

- No packets from the small traffic are lost, but the backlog behind the packets from heavy traffic causes longer queuing delays. Queue build-up is the name given to this issue. The only solution to solve the issue of queue build-up in Data Center Network switches is to reduce queue occupancy.

The majority of the TCP alternatives now in use adopt the next strategy for congestion control, which fails to reduce queue occupancy. To decrease queue occupancy and solve the issue of queue build-up, a pragmatic strategy is necessary. **Figure 5** shows the **Queue Buildup** problem of TCP in data center networks.

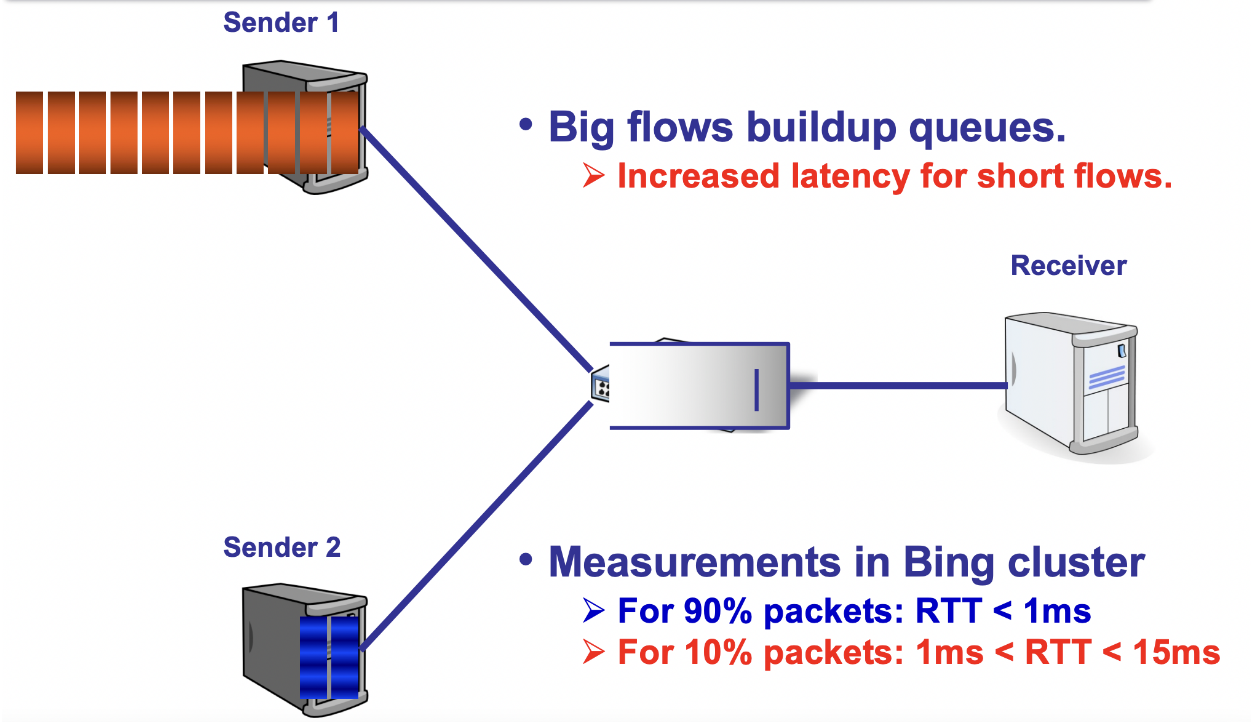
Sender 1
• Big flows buildup queues.
➢ Increased latency for short flows.
Receiver
Sender 2
• Measurements in Bing cluster
➢ For 90% packets: RTT < 1ms
➢ For 10% packets: 1ms < RTT < 15ms

Figure 5: Queue Buildup problem of TCP in data center networks

3. VARIOUS TRANSPORT PROTOCOLS FOR DATA CENTER NETWORK

PROTOCOL	Modification needed	TCP Incast	TCP Timeout	Solve Latency	TCP Queue Buildup	Deadline aware
DCTCP	Both at Sender and receiver end	YES	NO	NO	YES	NO
ICTCP	Only at Sender end	YES	NO	NO	YES	YES
D²TCP	Both at Sender and receiver end	YES	YES	YES	YES	YES

3.1. Transmission Control Protocol

Data centers are important for storing a vast number of data and databases used in many different company services and operations. Because the data center network had many single-link prototypes with a high data transfer rate, it set itself apart from other networks. data center network uses several tree architectures, reduces low barrier switches, low data transfer, and automated scaling. These data center networks are detrimental to standard TCP. As a result, there are now diplomatic solutions available to address the issue of regular TCP in Data Center Networks in order to o offer sensible fixes for the current transport layer protocol.

3.2. Data Center Transmission Control Protocol (DCTCP)

One specific TCP protocol for data center networks is called DCTCP. DCTCP was suggested for employing a special Explicit Congestion Notification (ECN) approach to produce multi-bit input to the end hosts inside the network. DCTCP evaluation at 1Gbps to 10Gbps speeds with shallow buffered artifact switches The DCTCP approach that TCP offered provides the same or greater performance than TCP while discriminating less buffer space. Additionally, DCTCP offers low latency and great burst tolerance for data communication between two points. Without affecting front traffic, DCTCP enables the apps to manage this background traffic.

There are two primary parts to the DCTCP algorithm:

- **Simple marking at the Switch:**

DCTCP uses a very simple and dynamic queue management approach. The incoming packet is tagged with a code point if the queue occupancy is over a defined point when it is received; otherwise, it is not corrected. This system makes sure that sources are promptly informed when the queue exceeds capacity. The majority of newly implemented switches may be converted to DCTCP using the RED labeling technique.

- **ECN at the receiver end:**

The sole difference between a DCTCP and TCP receiver is the delivery of information in the code points back to the sender. The receiver sets a series of ACK packets with the ECN-Echo flag. Otherwise, the sender has received the congestion notification (through the congestion window receiver). A DCTCP relays the precise sequence of designated packets from the receiver back to the sender.

The advantages of DCTCP are demonstrated below:

Queue Buildup:
As soon as the queue length on an interface surpasses a certain point, DCTCP senders begin to react. reducing the wait time for queues at busy switch ports. to make more buffer space accessible and lessen the effect of big flows on the time it takes for tiny flows to reach their ending point.

Buffer pressure:
DCTCP also overcomes the problem of buffer pressure because a clogged port's queue length does not increase significantly.

Incast:
When a large number of matching tiny flows hit the same queue, it is difficult to manage. One packet from each of the small flows can still overrun the buffer on a synchronized burst if the number of small flows is very high.

3.3. Dead-Line Aware Transmission Control Protocol (D^2TCP)

A kind of Transmission Control Protocol called Deadline-Aware TCP (D2TCP) manages bursts in a deadline-aware and instantaneous manner. D2TCP offers two guarantees:

1. D2TCP enables the use of a distributed and direct method for allocating bandwidth.
2. D2TCP uses the conventional congestion avoidance method in conjunction with the ECN response.

The conversion from deadlines to congestion window is performed using a gamma-correction function. Compared to DCTCP, performance evaluation of D2TCP results in fewer missed deadlines.

The following are D2TCP's features:

1. Compared to DCTCP, which needs fewer than 100 more lines of kernel code, D2TCP reduces missed deadlines by 20%.
2. achieves high transmission capacity as TCP for base streams without compromising the operation of Online Data Interchange (OLDI).
3. High bandwidth is achieved through background flow.
4. reduces the rate of missed deadlines by 75 and 50 percent, respectively, compared to DCTCP and deadline aware (D3).
5. When deadlines for D3 are missed, OLDIs give extra time for accurate calculation.

3.4 Adaptive Acceleration Data Center Transmission Control Protocol (A^2DTCP)

When a considerable amount of link bandwidth is available, A2DTCP slowly increases the flow transmitting rate in order to prevent congestion, leaving plenty of bandwidth resources unutilized. Long flow completion times and underutilized bandwidth resources are the results. The total outcome is a long flow completion time and a high ratio of deadlines missed. Another was that MI mechanisms might acquire significantly more bandwidth than AI in the same amount of time. Its disadvantage is also clear, though. When there is already significant congestion, MI may introduce extra packets into the network and hence exacerbate the existing congestion. As a result, a new window-rising method should capitalize on their positive traits and minimize their negative ones. More specifically, the new technique boosts the CW assertively in cases of light congestion to utilize the available bandwidth as rapidly as feasible. In order to prevent congestion, the window rising should be slowed down when the available bandwidth is insufficient.

These criteria should be met by this approach, according to the study mentioned above:

- Allow as many flows to complete their tasks as is practical.
- Reduce the time it takes for deadline-sensitive flows to complete,
- In the congestion avoidance phase, swiftly collect the available bandwidth.
- Be receptive to the historical TCP protocol and friendly to the hardware of the current switches.

Amiable performance in non-rush traffic makes sure that as compared to other protocols, the A2DTCP's acceleration of important flows does not significantly delay non-deadline flows. A2DTCP does this by carefully balancing the level of network congestion and flow urgency. Particularly, even with a high degree of urgency, flows with no hard deadline will still be able to obtain the necessary bandwidth so long as there is sufficient bandwidth. In a data-intensive setting with a network architecture, mix deadline and non-deadline traffic to demonstrate this characteristic. The performance of A2DTCP in a multi-hop network is assessed here using two common network topologies seen in data centers. Due to its fabric switch architecture, the topology of the OLDI atmosphere is also multi-hop; however, the flow might only be inefficient at the Top of the Rack (ToR) switches.

IV. Shortcomings of current work/Future Scope.

Three new TCP protocols discussed in the paper have revolutionized the way traffic is sent across data center networks today, and have opened doors to a lot more research. But nevertheless, these protocols have their own limitations, and there is always room for improvement. For instance, DCTCP fails to perform for DCN with more than 35 nodes. It fails if all at least 35 nodes send TCP Incast flow to the aggregator. In the case of D^{2}TCP, scalability is the major concern as it uses the same worker nodes in parallel. It also struggles to beat the TCP Outcast problem. More research is required to update this protocol to overcome the TCP Outcast problem. A^{2}DTCP design cannot detect end-to-end congestion. For this reason, it fails to avoid the difficulty of deploying ECN.

In fact, there exist a lot more TCP protocols out there besides the three discussed that are suitable for data center networks, each with its own set of pros and cons. Recently, a lot of research has been focussing on using designs that do not use TCP altogether. For example, in a paper published by a professor at Stanford University, the Homa system was put forward, which suggests not using TCP at all in DCN. Also, in new HTTP/3, uses UDP in the transport layer, and achieves order and congestion avoidance at the application layer.

V. CONCLUSION

Data centers have evolved into a necessary economic infrastructure for storing a lot of data and hosting a variety of cloud services. TCP is an older technology that offers reliable and scheduled bidirectional transmission of a stream of bytes between two applications running on the same or other machines. This is how data center flows traditionally function. To achieve performance efficiency, the majority of these applications employ a many-to-one communication structure. The primary issues with TCP Incast, Timeout, and latency are that it cannot provide high throughput and is set up in data center networks. TCP must thus be redesigned in order to manage the traffic in data center networks. In order to lessen the TCP issues in the data center, this article will give a thorough assessment of recently suggested transport protocols.

REFERENCE

Zhenqian Feng., Chuanxiong Guo., Haitao Wu., and Yongguang Zhang. ICTCP: Incast Congestion Control for TCP in DataCenter Networks. In Proceedings of the IEEE/ACM Transaction on Networking, 2013.

D., Maltz. Data., A. Greenberg., and M. Alizadeh. Data Center TCP (DCTCP). In Proceedings of the ACM Special Interest Group on Data Communication (SIG-COMM), 2010

WangJ., WenJ.,LiC., HanY.Dc-vegas., and XiongZ. A delay-based TCP congestion control algorithm for data center applications. In Proceedings of the Journal of Network Computer Applications, 2015.