UNIVERSITY OF WATERLOO
Faculty of Mathematics

Comparing the TCP CUBIC Algorithm to the Performance-Oriented Congestion Control Algorithm to Determine which Algorithm is the most Effective in Addressing Network Congestion

Huawei
Computer Networks and Protocols Department
Markham, Ontario, Canada

Prepared by
Siddhant Sharma
3B Computer Science
ID: 20773593
December 21st, 2021

Memorandum

To: Geng Li, Manager of Computer Networks and Protocols Department in Huawei

From: Siddhant Sharma

Date: December 21st, 2021

Re: Comparing the TCP CUBIC Algorithm to the Performance-Oriented Congestion Control Algorithm to Determine which Algorithm is the most Effective in Addressing Network Congestion

---

Hello Geng. As we agreed to in an earlier discussion, attached is the report on "Comparing the TCP CUBIC Algorithm to the Performance-Oriented Congestion Control algorithm to Determine which Algorithm is the most Effective in Addressing Network Congestion" which I have to create for my 3B work term report, and for the Computer Networks and Protocols Department. This work term report is the final and the fourth of four work term reports that the Co-operative Education Program at the University of Waterloo requires that I complete as part of my BMath Co-op degree requirements, has not received any academic credit.

As an Assistant Researcher Intern for the Computer Networks and Protocols Department my job was to research and present various congestion control algorithms and develop a testing application to run these algorithms. In my role, I have researched various congestion control algorithms including TCP CUBIC and the Performance-Oriented Congestion Control algorithm when building presentations and developing the testing application. The goal of this report is to explain the TCP CUBIC algorithm and the Performance-Oriented Congestion Control algorithm, and then compare these two algorithms to see which is more effective in handling network congestion.

The Faculty of Mathematics requests that you evaluate this report for technical analysis, command of topic and technical content. Once your assessment is complete, the report and your evaluation of the report will be grouped and then submitted to the Math Undergrad Office, in the University of Waterloo, for evaluation on campus by qualified work report markers. The combined marks from all evaluations will ultimately determine whether the report will receive academic credit and whether it will be considered for an award.

Thank you for your assistance in the preparation and completion of this report!

Siddhant Sharma

Table of Contents

Executive Summary

The purpose of this report is to determine whether TCP CUBIC or the Performance-Oriented Congestion Control algorithm is the more effective algorithm to use to address network congestion in large networks**.** Key points from the analysis include the point that the Performance-Oriented Congestion Control algorithm is considered by researchers to be a machine learning algorithm (Wenting et al., 2021). Another point from the analysis is that machine learning algorithms require a relatively large amount of computational and memory resources, thus the Performance-Oriented Congestion Control algorithm may require large amount of computational and memory resources to run (Zhang, 2018). Another key point from the analysis is that in a specific network environment in the global commercial internet the Performance-Oriented Congestion Control algorithm had a larger throughput compared to TCP CUBIC (Dong et al., 2014). However, real world networks can change over time and are complex and therefore the Performance-Oriented Congestion Control algorithm may not always have a higher throughput than TCP CUBIC in every network environment (Dong et al., 2014). The important point from the conclusion is that performance of a congestion control algorithm depends on the network environment and goals of the application. If memory and computational resources are scarce then TCP CUBIC should perhaps be used over the Performance-Oriented Congestion Control algorithm, due to the resource intensiveness of machine learning algorithms. (Zhang, 2018). If resources are sufficient to run Performance-Oriented Congestion Control algorithm and throughput is the most prioritized metric, then depending on the network environment either TCP CUBIC or Performance-Oriented Congestion Control algorithm may have a larger throughput and further testing should be done in order to select the algorithm.

1.0 Introduction

Applications such as Facebook Messenger, Google Hangouts and Discord have millions of users and these applications, and many others rely on networks to allow for the sending and receiving of data packets (Cressler, 2021). However, during the process of transmitting and receiving these data packets events can occur which sharply increase the amount of data packets being sent which in turn can cause a loss speed in the network but also cause network congestion (Amsterdam, 2021). Network congestion is when the amount of data packets being sent from a sending node goes past the sending node's limit which in turn causes data packets to be lost or the transmission of data packets to be slowed down considerably which therefore causes the service quality for a network to lower as well (Amsterdam, 2021). If network congestion is not handled, then it can escalate to the point that the service quality of a network is decreased enough that communication in the network is prevented, this is called congestive collapse (Amsterdam, 2021). To prevent congestive collapse, congestion control algorithms are used in order to adjust and modify the send rate of data packets in order to prevent network congestion from getting severe enough that it causes congestive collapse (Amsterdam, 2021). The actual amount of data being sent and received by nodes in a network for a given period time is called throughput (Petryschuk, 2021).

The assumptions used in this report is that the network is a large network where enough data is being sent through this large network, that network congestion and congestive collapse are possible situations that can occur and must be avoided. The major points for the report are as follows, the first point is that TCP CUBIC uses the cubic function and congestion windows (Pandora FMS Team, 2021). The second point is that the Performance-Oriented Congestion

Control algorithm uses results from experiments to adjust the sending rate (Dong et al., 2014). The third point is that researchers consider the Performance-Oriented Congestion Control algorithm to be a machine learning algorithm (Wenting et al., 2021). The fourth point is that machine learning algorithms require relatively large amounts of memory and computational resources and therefore the Performance-Oriented Congestion Control algorithm may require relatively large amounts of memory and computational resources to run (Zhang, 2018). The fifth point is that determining which congestion control algorithm has better performance depends mainly on the network environment in which the congestion control algorithm is run and the goal of the application. If the goal of the application is to have the highest throughput and memory and computational resources are not an issue then an experiment has shown the Performance-Oriented Congestion Control algorithm had a larger throughput than TCP CUBIC, in a specific network environment in the global commercial internet (Dong et al., 2014). However, this does not mean the Performance-Orientated Congestion Control algorithm will always have a superior throughput rate in every network environment compared to TCP CUBIC since real world networks are complex and can change over time (Dong et al., 2014). If memory and computational resource usage is the main concern then given the relatively large amount of memory and computational resources machine learning algorithms require TCP CUBIC should perhaps be chosen over Performance-Oriented Congestion Control algorithm (Zhang, 2018). The situation being analyzed in this report is which congestion control algorithm, TCP CUBIC or Performance-Oriented Congestion Control algorithm, is the most effective at handling network congestion. The purpose of this report is to explain how TCP CUBIC works and how the Performance-Oriented Congestion Control algorithm works and then determine in what situations the Performance-Oriented Congestion Control algorithm is superior to TCP CUBIC

and in what situations it is not, when it comes addressing network congestion in a large network. The methodology of this report is to go over how TCP CUBIC and the Performance-Oriented Congestion Control algorithm work and then determine and compare the benefits and drawbacks of each algorithm in order to determine which algorithm is most effective in addressing network congestion in a large network.

2.0 Analysis

2.1 How Does the TCP CUBIC Congestion Control Algorithm Work?

The central idea behind TCP CUBIC is the use of congestion windows (Pandora FMS Team, 2021). The congestion window by TCP CUBIC determines how many data packets are being sent by the sender, and it is this congestion window that is modified when the TCP CUBIC algorithm determines that there is network congestion (Pandora FMS Team, 2021). Essentially the TCP CUBIC algorithm uses the cubic function, which is a function that grows exponentially fast until it nears a specified point where it slows growth and once past the specified point grows exponentially fast again, to determine how much the congestion window should be changed (Shanker, 2018). Specifically, the TCP CUBIC algorithm keeps note of the size of the previous congestion window and when creating a new congestion window, the algorithm also keeps track of if network congestion occurs during the adjustment of the size of the current congestion window (Pandora FMS Team, 2021). If there is no network congestion during the adjustment of the size of the current congestion window, then the algorithm grows the current congestion window exponentially fast until it nears the size of the previous congestion window, where in this case the growth slows, and then after passing the size of the previous congestion window the growth of the current congestion window grows exponentially fast again (Pandora FMS Team, 2021) However, if network congestion does occur during any portion of the growth of the current congestion window then the algorithm will set the current congestion window size as previous

congestion window size and make a new congestion window and set this as the current congestion window and repeat the process (Pandora FMS Team, 2021).

2.2 How Does the Performance-Oriented Congestion Control Algorithm Work?

The Performance-Orientated Congestion Control algorithm is a congestion control algorithm released in 2014 (Dong et al., 2014). The main idea behind the Performance-Orientated Congestion Control algorithm is to run small, miniature, and quick experiments while the network is transmitting and receiving data packets and based on the results of these live experiments the Performance-Orientated Congestion Control algorithm modifies the sending rate of the data packets (Dong et al., 2014). Specifically, the Performance-Orientated Congestion Control algorithm makes the sender modify the data packet send rate for a period of time after this, reports on the number of data packets lost and sent are recorded (Dong et al., 2014). The Performance-Orientated Congestion Control algorithm then changes the send rate and records the results/reports and repeats this process multiple times, during every iteration the algorithm keeps track of the tested rate and the results/reports and then after running the multiple experiments, the algorithm passes the rates and corresponding results to a function which assigns a score to each rate based on how effectively it sent data without causing network congestion, once this is done the algorithm selects the rate with the highest score and continues the whole process again (Dong et al., 2014). Based on the operations of this congestion control algorithm, researchers consider the Performance-Orientated Congestion Control algorithm to be a machine learning algorithm (Wenting et al., 2021).

2.3 Comparing TCP CUBIC and the Performance-Oriented Congestion Control Algorithm.

In a specific network environment in the global commercial internet the Performance-Orientated Congestion Control algorithm was able to achieve a higher throughput than TCP CUBIC (Dong et al., 2014). It is important to note that real world networks are complex and can change over time (Dong et al., 2014). Therefore, if in one specific network environment the Performance-Orientated Congestion Control algorithm achieved a higher throughput than TCP CUBIC this does not mean the Performance-Orientated Congestion Control algorithm will always have a higher throughput than TCP CUBIC in every network environment (Dong et al., 2014).

The Performance-Orientated Congestion Control algorithm is an effective congestion control algorithm that specifically ensures that it does not assume that all packet loss is caused by network congestion (Dong et al., 2014). This assumption from the Performance-Orientated Congestion Control algorithm means the algorithm can make the optimal decisions when data packet loss occurs, and it is caused by network congestion or when data packet loss occurs, and it is not caused by network congestion (Schapira, 2018). TCP CUBIC on the other hand, runs under the assumption that all instances of packet loss, which is when data packets do not reach the receiver from the sender, means network congestion and reduces the congestion window size when faced with packet loss based on this assumption, however sometimes packet loss does not mean network congestion and the resulting reduction in the congestion window may lower performance significantly if the assumption is incorrect (Dong et al., 2014). Researchers do consider the Performance-Orientated Congestion Control algorithm to be a machine learning

algorithm (Wenting et al., 2021). Since machine learning algorithms require relatively large amount of memory and computational resources, the Performance-Orientated Congestion Control algorithm might be too resource intensive for certain applications to use if computational and memory resources for the application are scarce (Zhang, 2018).

3.0 Conclusions

Both the Performance-Orientated Congestion Control algorithm and TCP CUBIC have benefits and drawbacks and determining which congestion control algorithm performs better is largely dependent on the goals of the application and networks in which these algorithms are run. If memory resources and computational resources are not a concern, then the Performance-Orientated Congestion Control algorithm has been shown to have a higher throughput than TCP CUBIC in a specific network environment in the global commercial internet (Dong et al., 2014). However, this does not indicate that the Performance-Orientated Congestion Control algorithm will always have a higher throughput rate than TCP CUBIC since real world networks are complex and can change over time (Dong et al., 2014). Therefore, even if resources are enough to run either algorithm, testing both algorithms on the specific network environment should be done to determine which algorithm to select. The Performance-Orientated Congestion Control algorithm is a machine learning algorithm according to researchers (Wenting et al., 2021). However, machine learning algorithms tend to require relatively large amounts of memory and computational resources and therefore if conserving memory resources and computational resources is the main priority of the application TCP CUBIC should perhaps be used over the Performance-Orientated Congestion Control algorithm (Zhang, 2018).

References

Amsterdam, T. (2021, July 7). *Understanding Congestion Control*. Granulate. Retrieved November 27, 2021, from https://granulate.io/understanding-congestion-control/.

Cressler, C. (2021, July 12). *8 most popular instant messaging & chat protocols*. CometChat. Retrieved November 27, 2021, from https://www.cometchat.com/blog/popular-chat-and-instant-messaging-protocols.

Dong, M., Li, Q., Zarchy, D., & Godfrey, B. (2014, September). *PCC: Re-architecting Congestion Control for Consistent High Performance*. ResearchGate. Retrieved November 28, 2021, from https://www.researchgate.net/publication/266147390_PCC_Re-architecting_Congestion_Control_for_Consistent_High_Performance.

Pandora FMS Team. (2021, June 10). *TCP congestion control: Basic outline and TCP cubic algorithm*. Pandora FMS Monitoring. Retrieved November 27, 2021, from https://pandorafms.com/blog/tcp-congestion-control/.

Petryschuk, S. (2021, November 30). *Network Throughput vs Bandwidth and How to Measure It*. Auvik Networks Inc. Retrieved December 21, 2021, from https://www.auvik.com/franklyit/blog/network-throughput-vs-bandwidth/

Schapira , M. (2018, May 28). *It's time to replace TCP with online-learning congestion control*. APNIC. Retrieved November 28, 2021, from https://blog.apnic.net/2018/05/29/its-time-to-replace-tcp-with-online-learning-congestion-control/.

Shanker , S. (2018, August 1). *Congestion Control II: CUBIC*. Congestion control II: Cubic. Retrieved November 27, 2021, from https://squidarth.com/rc/programming/networking/2018/08/01/congestion-cubic.html.

Wenting, W., Gu, H., & Li, B. (2021, February). *Congestion Control: A Renaissance with Machine Learning*. Research Gate. Retrieved December 21, 2021, from https://www.researchgate.net/publication/349188825_Congestion_Control_A_Renaissance_with_Machine_Learning

Zhang, B. (2018, June 27). *A solution to the memory limit challenge in Big Data Machine Learning*. Medium. Retrieved December 21, 2021, from https://petuum.medium.com/a-solution-to-the-memory-limit-challenge-in-big-data-machine-learning-49783a72088b