GMC- 158

Evaluating TCP Protocol Performance in Cloud Environments

Abstract

This research investigates the performance of four different TCP algorithms—BBR, Reno, Vegas, and Cubic in a high-latency and congested condition within a cloud-based environment using EC2 instances and Mininet for network simulation. The study aims to evaluate the throughput and congestion window (cwnd) behavior of each algorithm under various network conditions to identify their strengths and weaknesses. By analyzing the performance metrics across different TCP algorithms, we provide insights into their suitability for cloud infrastructure, contributing to optimized network protocol choices for cloud-based applications and services. The results offer valuable guidance for enhancing network performance in dynamic cloud environments.

Introduction

TCP (Transmission Control Protocol) is a widely used and reliable transport layer protocol that ensures the accurate and ordered delivery of data across networks. With its ability to provide congestion control, flow control, and error recovery, TCP ensures efficient data transmission even in fluctuating network conditions.

The four TCP algorithms examined in this study—BBR, Reno, Vegas, and Cubic—each employ distinct mechanisms for congestion control and throughput management.

TCP Reno, one of the most widely used, relies on slow start, congestion avoidance, and fast recovery, but struggles in high bandwidth-delay product networks.

TCP Vegas, by contrast, aims to avoid congestion by estimating the network's available bandwidth, offering more proactive congestion control.

TCP Cubic, designed for high-speed and long-distance networks, adjusts the congestion window in a cubic fashion, providing better scalability than Reno.

TCP BBR, the newest among them, is designed to maximize throughput by modeling network bandwidth and round-trip time, leading to more efficient data transfer in environments with variable latency. Each algorithm has its advantages depending on the network environment, making them suitable for different use cases in cloud and data center applications.

Research Question(s)

•How do different TCP variants (Reno, Vegas, Cubic, and BBR) perform in terms of throughput under high-latency and congested conditions?

•How do different TCP variants (Reno, Vegas, Cubic, and BBR) affect the congestion window (cwnd) under high-latency and congested conditions?

Materials and Methods

To evaluate the performance of various TCP variants, this study use simulation-based methods in a cloud environment. The primary tool used was **Mininet**, an open-source discrete-event network simulator that allows controlled network modeling. For the cloud environment, an EC2 Ubuntu instance was used, running different TCP implementations for consistent comparison under controlled conditions.

Network Configurations: The experiments were conducted under a range of network conditions, including:

- High-latency environments (200ms delay, simulating long-distance communication)
- Congested networks (10Mbps bandwidth, representing typical urban internet usage)
- Wireless networks with fluctuating signal quality

Performance Metrics:

- **Throughput (Mbps)**: Total data transferred over a given time frame
- **Congestion Window (cwnd)**: Providing insights into how each TCP variant responds to changing network conditions and adjusts its transmission rate accordingly.

Results

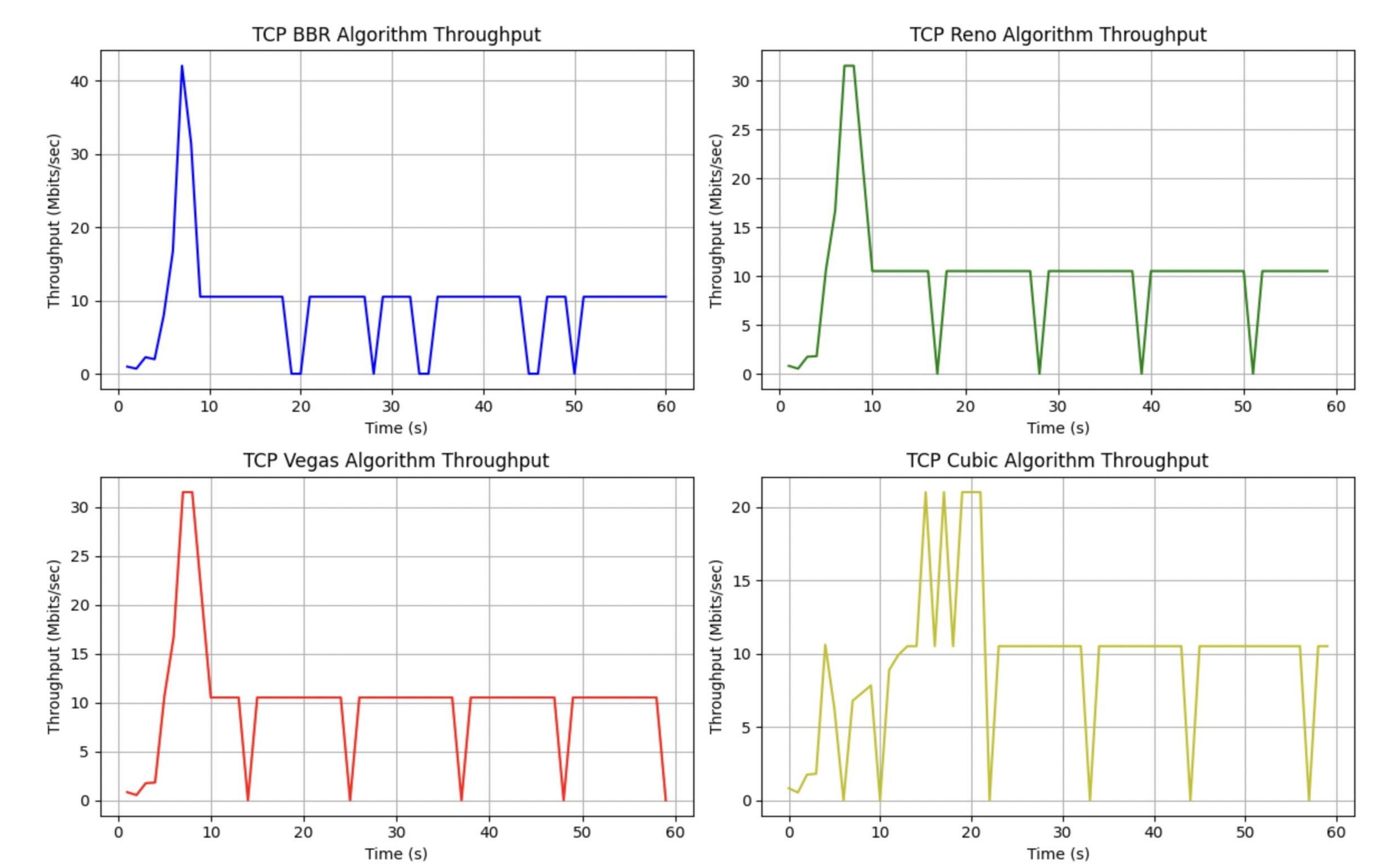
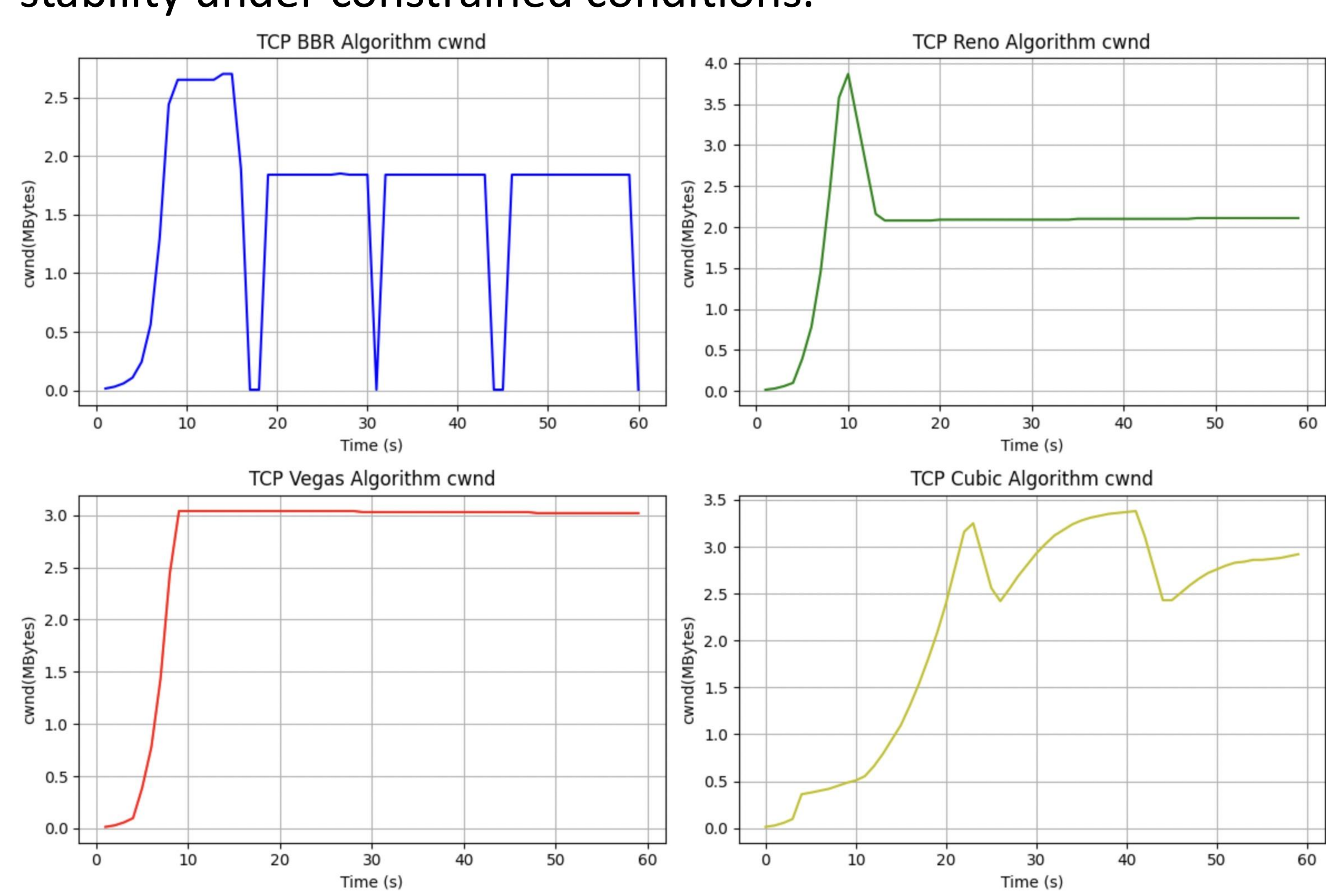


Fig1:throughput of 4 TCP algorithms in 60secs

Throughput:

Reno, and Vegas exhibit similar patterns once congestion occurs, in contrast, Cubic employs a more aggressive approach to congestion control using a cubic function for growth, allowing it to push the network harder. As a result, Cubic may initially achieve higher throughput, but like the other algorithms, it eventually stabilizes at a lower throughput of 10.5 Mbps as they adjust to the available bandwidth. This behavior highlights the trade-offs between aggressive congestion control and maintaining network stability under constrained conditions.



Congestion window analysis:

Reno and Vegas stabilize after reaching certain cwnd values, with Reno reducing its window upon congestion and Vegas being more conservative to avoid it.

BBR fluctuates regularly due to its bandwidth estimation approach, dynamically adjusting to optimize throughput.

Cubic shows irregular fluctuations due to its aggressive, cubic-based growth mechanism, which rapidly adjusts the cwnd in response to network conditions, often pushing the network harder than the others.

Conclusions

In conclusion, this research highlights the distinct congestion control mechanisms of different TCP algorithms and their varying performance under high-latency and congested network conditions. While Reno, Vegas, and BBR stabilize at lower throughputs, Cubic exhibits more aggressive and irregular fluctuations, demonstrating its unique approach to congestion control.

Acknowledgments

Advisor: Professor Ahyoung Lee

This research is for CS6027 (Computer Network)

Contact Information

Email: nming@students.kennesaw.edu

Linkedin: linkedin.com/in/nongming362

References

[1] Mo, Jeonghoon, et al. "Analysis and comparison of TCP Reno and Vegas." IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320). Vol. 3. IEEE, 1999.

[2] Balakrishnan, Hari, et al. "A comparison of mechanisms for improving TCP performance over wireless links." IEEE/ACM transactions on networking 5.6 (1997): 756-769.

[3] Grieco, Luigi A., and Saverio Mascolo. "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control." ACM SIGCOMM Computer Communication Review 34.2 (2004): 25-38.

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Author(s): Nong Ming

Advisor(s): Professor Ahyoung Lee