ISSN 2278-3091
Volume 9, No.1.5, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse0391.52020.pdf
https://doi.org/10.30534/ijatcse/2020/0391.52020

Performance Analysis of TCP Variants in Named Data Networking

Asmaa EL-BAKKOUCHI[1], Mohammed EL GHAZI[2], Anas BOUAYAD[3], Mohammed FATTAH[4], Moulhime EL BEKKALI[5]
[1]Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, SMBA University Fez, Morocco, asmaa.elbakkouchi@usmba.ac.ma
[2]Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, SMBA University Fez, Morocco, mohammed.elghazi@usmba.ac.ma
[3]Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, SMBA University Fez, Morocco, anas.bouayad@usmba.ac.ma
[4]Moulay Ismail University Meknes, EST, Morocco, fattahm@gmail.com,
[5]Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, SMBA University Fez, Morocco, moulhime.elbekkali@usmba.ac.ma

ABSTRACT

In the traditional TCP network, the congestion window is reduced in the case of timeout or three duplicate ACKs. However, in Named Data Networking (NDN), it is done in the case of timeout, marked packets or NACKs etc. This makes direct application of TCP variants in NDN inefficient. The objective of this paper is to implement and simulate the TCP variants in NDN congestion control; we compare and analyze the performance of these variants in terms of delay measurement, packet loss rate and throughput.

Key words : NDN; Congestion Control; TCP Variants; CUBIC; Agile-SD; STCP; BIC-TCP; SACK-TCP

1. INTRODUCTION

The use of the internet has changed greatly due to the evolution of technologies and applications and has increased more than before, especially today with the continued spread of COVID-19 where people are working and studying at a distance using platforms, and others are watching videos and gaming online which increases the number of devices connected to the internet. According to [1], by 2020, 50 billion things will be connected to the internet. This increase in connected devices and platforms will increase internet traffic. As a result, the network can become overloaded and the congestion can occur.

This increase in the number of internet users poses challenges to the current internet architecture, which has prompted researchers to propose new architectures for the future internet. These architectures include Named Data Networking (NDN) [2] that has received the attention of many researchers as one of the most popular architectures for the future internet. To communicate, it uses content name instead of IP addresses. This name has a hierarchical structure formed by a sequence of variable-length components separated by the symbol "/" [3].

In NDN, the data is retrieved using two types of packets; an interest packet and a data packet [2] as shown in Figure 1. A consumer asks for the content by sending a packet of interest, and then any node who has the appropriate data may send it using a data packet. This data packet is routed in the inverse path of the corresponding packet of interest, and the intermediate routers can use their cache to store this data to satisfy future request for the same data. NDN node has three structures of data: CS, PIT and FIB as illustrated in Figure 2.

«Content Store (CS) functions as a cache of content [4]. » The data packet received by CS is stored temporarily in the cache and used when a request is made for the same data.

«Pending Interest Table (PIT) contains the interest packets that is transmitted upstream but is not yet satisfied [4], a list of incoming interfaces from where interest for this name was obtained and a list of outgoing interfaces to which the interest was sent.

«Forwarding Information Base: is a base containing name prefixes to identify where the content producers are located and a list of interfaces to determine which interface is needed to transmit the interest packet [5]. »

The new features of NDN architecture such as receiver-driven, one-interest-one-data, and multipath cause new challenges particularly when we apply TCP congestion control algorithms directly in NDN congestion control.

In NDN, congestion may occur in the downlink because the data packet is much larger than the interest packet. This happens if routers receive a large amount of data which cannot transmit it on time, which cause several problems such as a significant delay; packet loss and buffer overflow. Therefore, the researchers proposed to regulate the sending rate of interest packets to control the return rate of data packets [6].

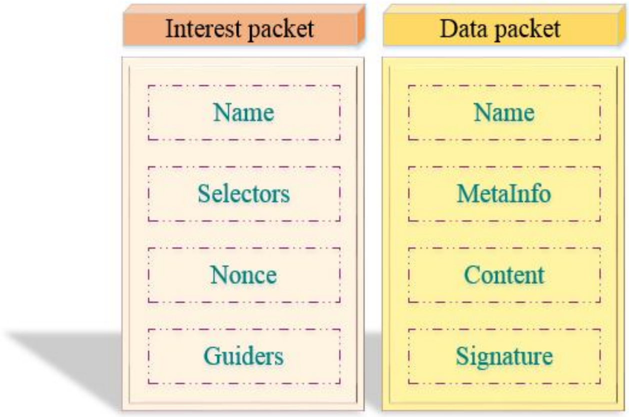

Figure 1: Packets in NDN architecture

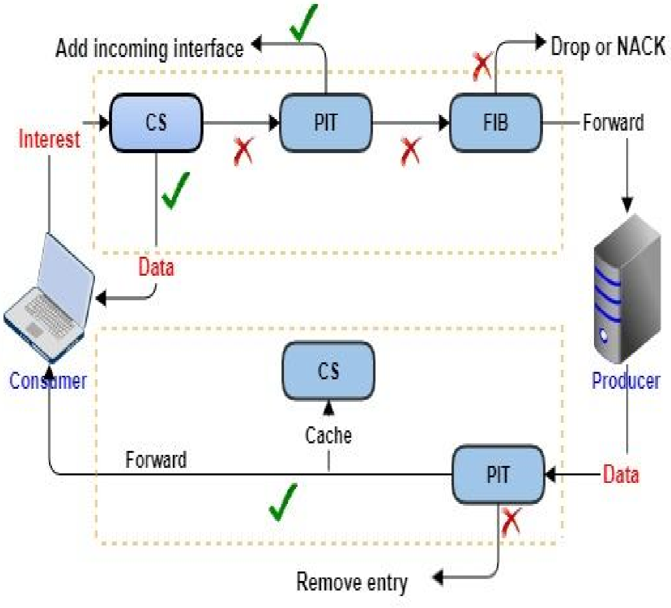

Figure 2: Forwarding process at NDN node

This paper analyzes and compares the packet loss, delay measurement and throughput of TCP variants in NDN congestion control. The remainder of this paper is organized as follows: Section II explains the congestion control in Named Data Networks. Section III gives an overview of TCP Variants. The performance evaluation of TCP variants is discussed in Section IV and conclusion in the last Section.

2. CONGEDTION IN NAMED DATA NETWORKS

Congestion refers to the situation where a node or link transports an excessive quantity of data affecting network performance such as packet loss [7].

Congestion can also be created in these situations [7]:

- Network overload.
- When routers are slow to execute a task.
- The packet rate exceeds the capacity of the outgoing link.
- When the outgoing capacity of the router is less than the sum of the inputs.

To avoid congestion, we adjust the window size of the interest packets at the consumer node to avoid buffer overflow in both consumer and routers nodes.

To do this, NDN uses the same way as that used in TCP. It uses the congestion window “cwnd” that reacts depending on the network status. In the case of network congested, two algorithms will be used, which are slow start and congestion avoidance [8] to manage the congestion window size. Similarly, NDN's new characteristics can be used to avoid network congestion, like caching, which can respond by a desired data if it content store contains it. Thus, the use of multipath can avoid the congestion path by divert data from another path. In NDN, the authors focuses on the way to take advantage of these new functionalities to control congestion [9].

TCP variants cannot be applied directly in the NDN architecture for the following reasons:

- In the current architecture, the mechanisms to control congestion are based on losses and delays at the sender node. However, in the future architecture, authors control congestion at both consumers and routers nodes.
- To communicate, TCP uses a single path between two hosts and uses RTT (Round Trop Time) as a network congestion indicator which is impossible in the future internet architecture NDN because it uses several paths with different RTTs.
- The current TCP Internet architecture is based on self-synchronization, which may pose a fairness issue between popular and unpopular content in the future NDN Internet architecture that uses caching in intermediate routers.
- The use of TCP mechanisms based on RTO (Retransmission Timeout) cannot be reliable in NDN, because the data in the latter can come from multiple sources [10], [11] and by multipath.

For these reasons, authors proposed to take advantage of TCP variants to increase/decrease the congestion window of NDN congestion control mechanisms while respecting the characteristics of this new internet architecture.

3. OVERVIEW OF SOME SELECTED TCP VARIANTS

In recent years, several variants of TCP have been proposed to control congestion in TCP/IP networks. These variants aim to control the size of the congestion window to increase network throughput and reduce the risk of becoming congested. In this paper, we will present five of them to evaluate their performance in NDN congestion control.

3.1 SACK-TCP

SACK [12] a Selective Acknowledgment, is an extension of Reno TCP [13] that was proposed to solve the problems such as the detection of more than one dropped packet and the retransmission of multiple packets lost per RTT encountered by Reno and New Reno [14]. In SACK, every ACK contains a section indicating the sequence number of packets that have been acknowledged. In addition, SACK estimates the number of unacknowledged packets using the pipeline parameter. SACK requires ACKs to be selective rather than cumulative. If no such segments are received, it transmits a new packet to the network. On top of that, SACK allows several lost segments to be sent in only one RTT.

3.2 Scalable TCP

STCP [15] the Scalable TCP is an algorithm that was introduced by Kelly in 2003 to address the problems of bandwidth growth in high-speed networks. STCP uses the following algorithm: if the congestion window is less than 32 packets, STCP functions like a traditional TCP. Else, STCP applies multiplicative growth and multiplicative reduction to adjust the congestion window size.

STCP algorithm is based on two parameters α, β where (($0 < \alpha < 1$) and ($0 < \beta < 1$). In the case of a congested network, STCP decreases its congestion window by the parameter β. Otherwise; STCP increases its congestion window by the setting α as follows:

$$\begin{array}{ll} cwnd = cwnd + \alpha & \alpha = 0.01 \qquad (2) \\ cwnd = cwnd - (\beta * cwnd) & \beta = 0.125 \end{array}$$

3.3 BIC-TCP

BIC-TCP (*Binary Increase* Congestion control) [16] is an algorithm that was proposed in 2004 to address the problem of RTT unfairness. It is based on three parameters to divide the congestion window: minwin to define the minimum congestion window, maxwin to define the maximum congestion window and tw to define the target window that takes the median value of maxwin and minwin as parameter. BIC works like a traditional TCP protocol if the congestion window is less than 31 packets. Otherwise, if congestion occurs, the value of the maximum congestion window will be updated by the current value of the congestion window else the value of the minimum congestion window will be updated by the current value of the congestion window.

3.4 CUBIC-TCP

CUBIC-TCP [17] was introduced for the first time in 2008 as an improved version of BIC-TCP. CUBIS is based on the use of a cubic function to increase the window size as follows:

$$W_{cubic} = C\,(t-K)^3 + W_{max} \qquad (1)$$

where, C is the scaling factor, t is the elapsed time from the last window reduction, W_{max} is the window size just before the last window reduction, and $K = (W_{max}\beta/C)^{1/3}$, where β is a constant multiplicative decrease factor applied for the window reduction at the time of loss event.

3.5 Agile-SD

Agile-SD [18] is a new algorithm that has been proposed for high-speed and short-distance networks which uses the agility factor to control congestion. Agile-SD starts with a slow start phase to ensure exponential growth. If congestion is detected, it reduces its packet-sending rate by the parameter β_1 (by default equal to 0.90) and then moves to the congestion avoidance phase, which characterized by an increase of α to have a convex curve. If congestion is detected in this phase, it reduces the packet-sending rate by the parameter β_2 (by default equal to 0.95). If the congestion window is approaching to the bandwidth limit, Agile-SD responds to this state by starting a linear increase until another packet drop occurs. Finally, if a timeout occurs, Agile-SD initializes its congestion window (by default, it starts with two packets).

4. PERFORMANCE EVALUATION OF TCP VARIANTS

In order to verify the effectiveness of TCP variants in controlling NDN congestion, the ndnSIM simulator (NS-3 based simulator) [19] was used to simulate experiments [20]. These variants have been evaluated in terms of packet loss rate, delay measurement and throughput.

- Delay Measurement: The time between sending an interest packet and receiving its data packet.
- Packet Loss Rate: Refers to the number of dropped packets per second. It is the difference between the number of packets sent by a node and the number of packets received by the same node over a given time. In NDN networks, packet loss include interest packets and data packets.

- Throughput: Refers to the number of data successfully transmitted from the source to the destination [21]. It varies depending on the amount of transmitted packets and the amount of packets dropped by the network and it is measured in bits per second.

4.1 Simulation scenario

The first topology consists of two consumers, a router and a content producer as shown in Figure 3. The bandwidth of the link between the consumer 1 and the router was set to 30Mbps with a 10ms delay. The bandwidth of the link between the consumer 2 and the router was set to 60Mbps with a 10ms delay. The bandwidth of the link between the consumer 3 and the router was set to 90Mbps with a 10ms delay and that of the link between the router and the producer was set to 20Mbps with a 30ms delay as shown in Table 1. In this topology, both consumers request the same content and start at beginning.

In the second topology, the number of consumers and producers was increased to 6 consumers and 6 producers with a bottleneck link between these nodes as shown in Figure 4. The bandwidth for each consumer-router link was set to 100Mbps with different delays varying between 1ms, 10ms, 15ms, 20ms, 25ms and 30ms. For the router1-router2 link, the bandwidth was set to 5Mbps with a delay of 15ms. For the router 2 and the producers 1, 3 and 5 the bandwidth was fixed at 20Mbps with delays of 10ms, 5ms and 1ms respectively and for the router 2 and the producers 2, 4 and 6 the bandwidth was fixed at 10Mbps with delays of 10ms, 5ms and 1ms respectively as shown in Table 2. In this topology, the consumers request different data.

The TCP variants were implemented into ConsumerPcon [22], which is a hybrid congestion control mechanism. Simulations were conducted using ndnSIM, which is an NS-3 based simulator and the time of this simulation was set to 30 seconds.

Table 1: Parameters of simulation topology 1

Parameters	**Values**
Simulation Time	30s
Delay (Consumer 1/2/3 - Router)	10ms
Delay (Router – Producer)	30ms
Bandwidth (Consumer1 – Router)	30Mbps
Bandwidth (Consumer2 – Router)	60Mbps
Bandwidth (Consumer3 – Router)	90Mbps
Bandwidth (Router – Producer)	20Mbps

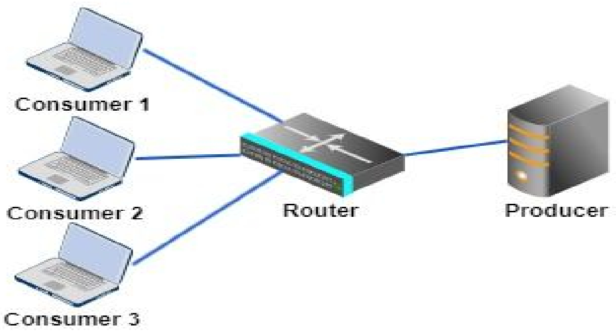

Figure 3: Simulation Topology 1

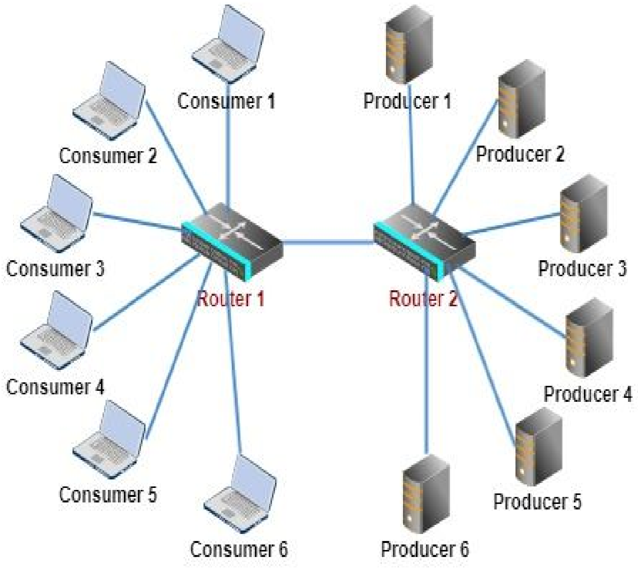

Figure 4: Simulation Topology 2

Table 2: Parameters of simulation topology 2

Parameters	**Values**
Simulation Time	30s
Delay (Consumer 1 - Router1) Delay (Router 2 – Producer 1 and 2)	10ms
Delay (Router 1 – Router 2) Delay (Consumer 3 – Router 1)	15ms
Delay (Consumer 1 – Router 1/ Router 2 – Producer 5 and 6)	1ms
Delay (Router 2 – Producer 3 and 4)	5ms
Delay (Consumer 4 – Router 1)	20ms
Delay (Consumer 5 – Router 1)	25ms
Delay (Consumer 6 – Router 1)	30ms
Bandwidth (Consumers – Router 1)	100Mbps
Bandwidth (Router 1 – Router 2)	50Mbps
Bandwidth (Router 2– Producer 1, 3 and 5)	20Mbps
Bandwidth (Router 2– Producer 2, 4 and 6)	10Mbps

4.1 Simulation results

The throughput, the packet loss and the delay measurement of CUBIC, STCP, BIC, Agile-SD and SACK were measured in two different scenarios as explained in the preceding paragraph.

- Throughput

The figure 5 presents the comparison between the TCP variants in the first scenario while the figure 6 presents the comparison between them in the second scenario. Table 3 presents the average throughput of the five TCP variants.

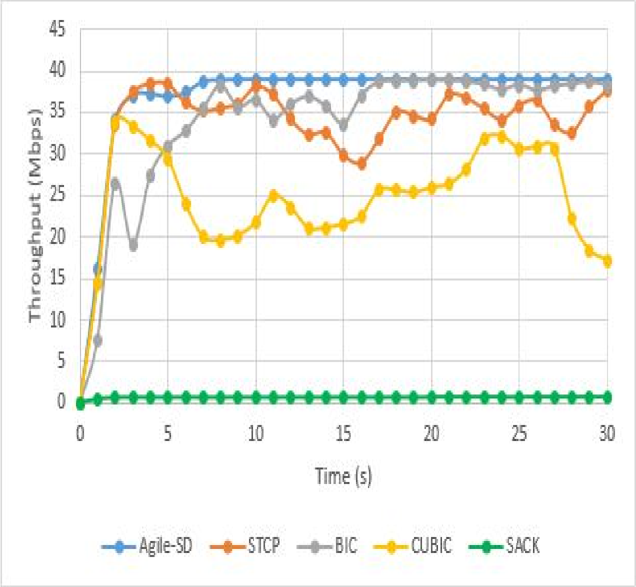

Figure 5: Throughput comparison of the examined TCP Variants in scenario 1

Table 3: Average throughput of the variants in NDN congestion control

Variants	CUBIC	STCP	BIC	Agile-SD	SACK
Scenario 1	28,95	35,85	34,74	37,33	**0,7**
Scenario 2	105,24	110,58	101,82	110,9	6,94

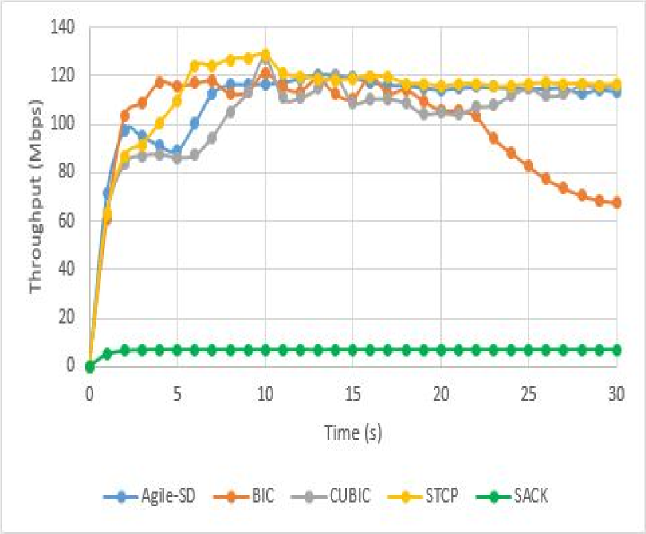

Figure 6: Throughput comparison of the examined TCP Variants in scenario 2

The result of the comparison shows that Agile-SD outperformed other TCP variants in terms of throughput in both scenarios. This is explained by the fact that Agile-SD is based on an agility factor mechanism that shows faster cwnd (congestion window) growth, which helps it to outperform the other TCP variants. On the other hand, it appears that the SACK has a very low throughput, that is explained by the slowly growth of cwnd (1/cwnd) resulting in less use of available bandwidth because it is designed for low speed network. For the others variants, STCP, CUBIC and BIC, we can see that their throughput is close with STCP slightly outperforms them in terms of throughput.

- Packet Loss Rate

The figure 7 presents the number of packets loss for the five variants of TCP in the first scenario while the figure 8 presents the number of packets loss for the five variants of TCP in the second scenario and the average packet loss rate was presented in Table 4.

Table 4: Average packets loss rate of the variants in NDN congestion control

Variants	CUBIC	STCP	BIC	Agile-SD	SACK
Scenario 1	0,8	2,35	1,064	19,48	0
Scenario 2	1,7	11,2	173,96	49,46	0

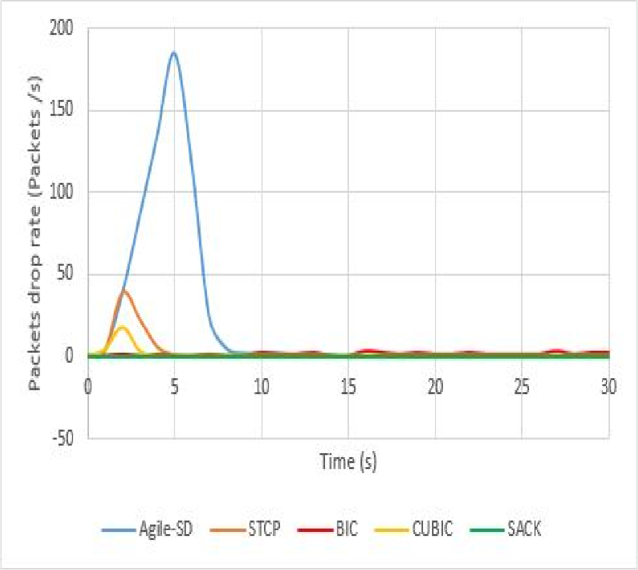

Figure 7: Packets loss rate of scenario 1

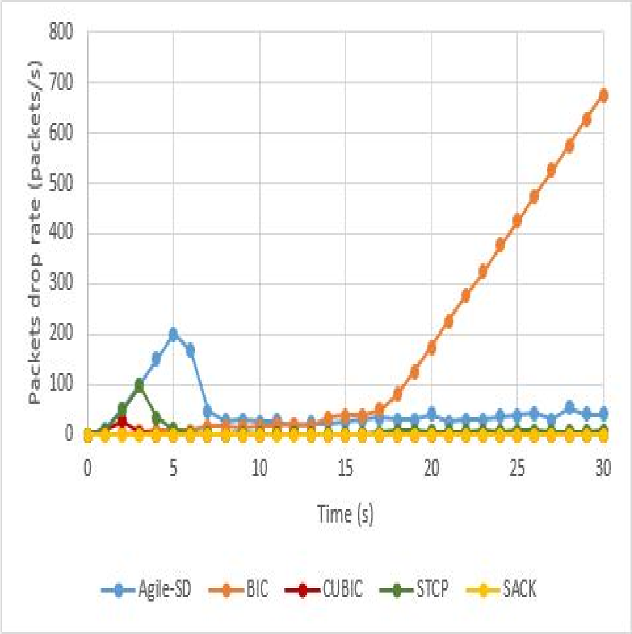

Figure 8: Packets loss rate of scenario 2

In these figures, CUBIC and SACK are the best two variants in terms of packet loss. SACK has a very low throughput, which explain the zero dropped. This is because SACK increases its congestion window very slowly by (1/cwnd). CUBIC is based on the cubic function, whose growth window only depends on the real time between two successive congestion events, which allows it to carefully increase its congestion window and consequently reduce the number of lost packets. This explains the lowest loss in the two scenarios.

Agile-SD has the highest packet loss rate because it has the highest throughput. The mechanism of the agility factor used in this variant is characterized by a faster growth of the cwnd (congestion window) which increases congestion and consequently increases the packet loss rate. This explains the higher rate of packet loss in both scenarios. For BIC and STCP, the packet loss rate can be negligible when looking at the number of packets exchanged.

The exponential increase between the second 3 and 8 of the other variants is explained by the fact that these variants, at the end of the slow-start phase, quickly increase their window of congestion to fully use the available bandwidth before moving to the congestion avoidance phase which causes problems such as packet loss. This problem is known as the burst loss.

- Delay measurement

The figure 9 presents delay measurement of the five variants of TCP in the first scenario while the figure 10 presents the delay measurement of the five variants of TCP in the second scenario and the average delay measurement was presented in Table V.

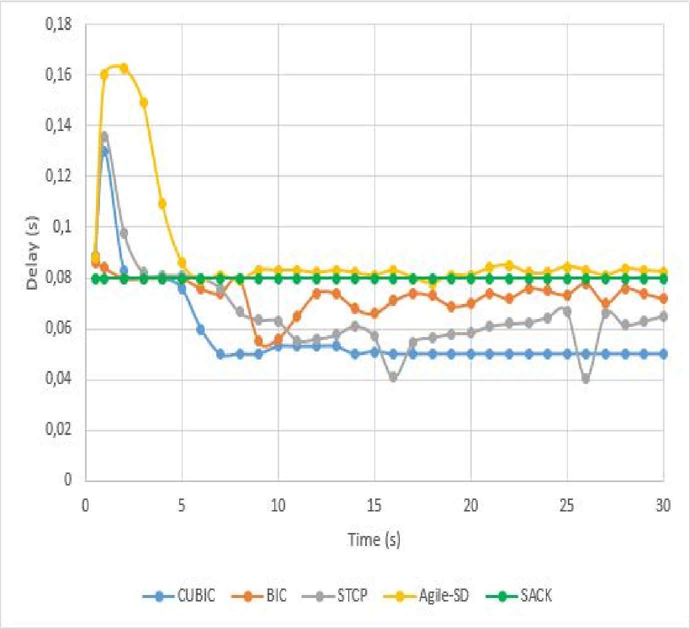

Figure 9: Delay measurement analysis for scenario 1

Table 5: Average delay measurement of the variants in NDN congestion control

Variants	CUBIC	STCP	BIC	Agile-SD	SACK
Scenario 1	0,058	0,067	0,073	0,09	0,08
Scenario 2	0,054	0,065	0,076	0,074	0,053

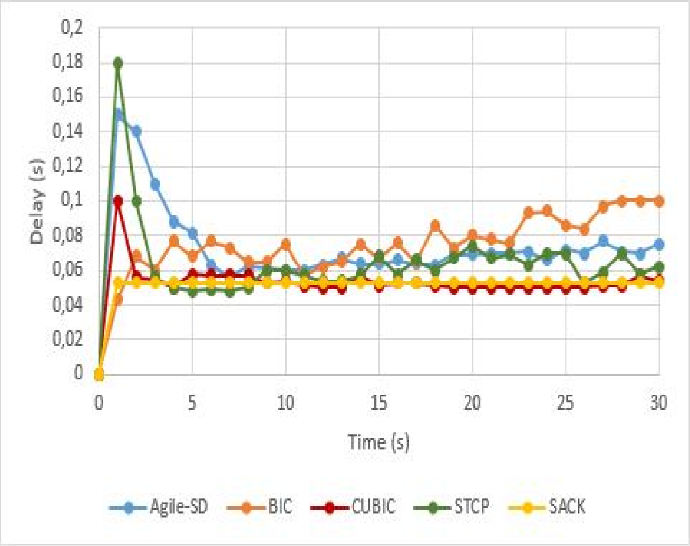

Figure 10: Delay measurement analysis for scenario 2

In these figures, CUBIC has the minimum average delay with 0,056s and Agile-SD had the maximum average delay with 0,082s. The behavior of throughput and packet loss rate affect the delay measurement of these variants. The increasing rate of packets loss increases the delay measurement of packets.

Finally, all of these variants behave like TCP standard when the window size is less than the threshold; they use the same algorithm of slow-star phase. The difference resides in the algorithm used in the congestion avoidance phase, which affect the performance metrics of these variants, such as throughput, packet loss, delay measurement and it can affect fairness too.

The analyses show that there is a significant need to modify the TCP protocol to adapt it to NDN congestion control, especially with the exponential growth in the number of users. The results indicate that CUBIC may hold promise for avoiding packet drop and increasing the use of throughput to control congestion in NDN.

5. CONCLUSION

In Named Data Networking (NDN), the TCP variants cannot be applied directly to control congestion because they cannot give effective results due to NDN's new features such as multipath and caching. In this paper, we proposed to implement five variants of TCP (CUBIC, BIC, Agile-SD, STCP and SACK) in a hybrid congestion control mechanism PCON using ndnSIM and evaluated them in terms of delay, packet loss rate and throughput. The simulation results show that CUBIC is more efficiency than the others to control congestion in NDN networks.

REFERENCES

1. Cisco, “**Cisco visualization: the internet of things**,” *Cisco*, p. 6, 2014.
2. L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, “**Named data networking (ndn) project**,” *Relat´orio T´ecnico NDN 0001, Xerox Palo Alto Research Center-PARC*, 2010.
3. M. Amadeo, A. Molinaro, C. Campolo, M. Sifalakis, and C. Tschudin, “**Transport layer design for named data wireless networking**,” in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 464–469.
4. A. El-bakkouchi, A. Bouayad, and M. ELBekkali, “**A hop-by-hop Congestion Control Mechanisms in NDN Networks – A Survey**,” *2019 7th Mediterr. Congr. Telecommun*, pp. 1–4, 2019.
5. A. El-Bakkouchi, M. El Ghazi, A. Bouayad, M. Fattah, M. El Bekkali, and S. Mazer, “**Packet Loss and Delay Measurement Analysis of TCP Variants in NDN Congestion Control**,” *2020 1st Int. Conf. Innov. Res. Appl. Sci. Eng. Technol. IRASET 2020*, pp. 2–6, 2020.
6. Y. Liu, X. Piao, C. Hou, and K. Lei, “**A CUBIC-Based explicit congestion control mechanism in named data networking**,” *Proc. - 2016 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2016*, pp. 360–363, 2017.
7. P. Chaudhary and S. Kumar, “**A Review of Comparative Analysis of TCP Variants for Congestion Control in Network,**” *Int. J. Comput. Appl.*, vol. 160, no. 8, pp. 28–34, 2017. https://doi.org/10.5120/ijca2017913087
8. M. J. K. Jacobson, Van, “**Congestion avoidance and control**,” *Symp. Proc. Commun. Archit. Protoc. SIGCOMM 1988*, no. 1, pp. 314–329, 1988.
9. Shuo Yang, “Applying TCP Congestion Control Schemes on NDN and their Implications for NDN Congestion Control Shuo Yang,” pp. 1–10.
10. L. Saino, C. Cocora, and G. Pavlou, “**CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking**.” in Proc. of IEEE ICC’13, 2013.
11. G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, **Multipath Congestion Control in Content-Centric Networks**, in Proc. of IEEE INFOCOM 2013 Workshop on Emerging Design Choices in Name-Oriented Networking, 2013.
12. A. R. M. Mathis, J. Mahdavi, S. Floyd, “**TCP Selective Acknowledgment Options**,” no. May, pp. 1–12, 1996.
13. Fall, K. and S. Floyd, "**Simulation-based Comparisons of Tahoe, Reno and SACK TCP**", Computer Communication Review, July 1996. URL "ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z".
14. S. Floyd, T. Henderson, and A. Gurtov, “**The newreno modification to tcps fast recovery algorithm**,” tech. rep., RFC 2582, April, 1999.
15. T. Kelly, "**Scalable TCP: Improving performance in highspeed wide area networks**," Computer Communication Review, vol. 33, pp. 83-91, 2003
16. XU L, Harfoush K, Rhee 1, "**Binary Increase Congestion Control for Fast Long-Distance Networks**," In proceeding of INFOCOM, 2004.
17. S. Ha, 1. Rhee and L. Xu, "**CUBIC: a new TCP-friendly high-speed TCP variant**," SIGOPS Oper. Syst. Rev, vol. 42, pp. 64-74, 2008.
18. M. A. Alrshah, M. Othman, B. Ali, and Z. M. Hanapi, “**Agile-SD: A Linux-based TCP congestion control algorithm for supporting high-speed and short-distance networks**,” *J. Netw. Comput. Appl.*, vol. 55, no. June, pp. 181–190, 2015.
19. A. Afanasyev, I. Moiseenko, and L. Zhang, “**ndnSIM: NDN simulator for NS-3**,” Tech. Rep. NDN-0005, pp. 1–7, 2012.
20. A. Es-Saqy, M. Abata, M. Fattah, S. Mazer, M. Mehdi, and M. El Bekkali, “**Study and design of a MMIC voltage controlled oscillator for 5G MM-wave band applications**,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2124–2129, 2020.

21. S. J. Shobana, M. V. Prakash, M. Sivaram, and V. Porkodi, “**FCCP - NS: A fair congestion control protocol with N-sinks in wireless sensor networks**,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1, pp. 43–51, 2019.
22. K. Schneider, C. Yi, B. Zhang, and L. Zhang, “**A Practical Congestion Control Scheme for Named Data Networking**,” in Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking - ACM-ICN ’16, 2016, pp. 21–30. https://doi.org/10.1145/2984356.2984369