Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

STCP: A Novel Approach for Congestion Control in IoT Environment

Ajay Kumar Gupta

Ph.D. Candidate, SCA, IFTM University, Moradabad, UP, India. E-mail: ajayguptagorakhpur@gmail.com

Devendra Singh

Associate Professor, Ph.D., SCA, IFTM University, Moradabad, UP, India E-mail: dev625g@gmail.com

Karan Singh*

*Corresponding author, Assistant Professor, Ph.D., SCSS, Jawaharlal Nehru University, New Delhi, India. E-mail: karancs12@gmail.com

Lal Pratap Verma

Associate Professor, CSE Deptt, Ph.D., Moradabad Institute of Technology, Moradabad, UP, India. E-mail: er.lpverma1986@gmail.com

Abstract

The main idea of IoT is to connect several objects to each other through Internet. In the field of Computer Network the main problem identified by researchers is network congestion. Now a day's network congestion is increasing very rapidly because IoT connect a huge number of devices to internet. A transport layer protocol TCP (Transmission Control Protocol) is accountable for network congestion control. The behavior of TCP is not stable as it takes long time to fill the available capacity of the network. It also continuously keeps assessing the capacity of data transmission through increasing the limits.TCP drops its data transmission rate aggressively when packets are dropped, which significantly reduces the throughput. This paper suggests a new approach, stable transmission control protocol for IoT applications. The experimental results show that stable transmission control protocol achieves better performance in terms of goodput.

Keywords: Internet of Things; Protocol; Internet; Computer Network.

DOI: https://10.22059/jitm.2022.85648 Manuscript Type: Research Paper University of Tehran, Faculty of Management

Received January 10, 2021 Accepted March 25, 2021

Introduction

As IoT provides a platform were network integration of different types is possible. So a new field is emerging for researchers.IoT provides a platform were various devices share their information and data among each other. During the advancement of the technology people uses a large number of different devices to connect with the internet for communication. Transmission control protocol is the most reliable protocols that are extensively used for transmission of data over the Internet. Transmission control protocol (TCP) Postel J(1981) JacobsonV(1988) Allman M et al.(1999) Floyd S & Henderson T(1999) Floyd S et al.(2004) provides congestion control technique to adjust the sending rate, in line with available bandwidth. IoT provides different application protocol for data transmission among various devices. To support the data transmission in IoT several TCP flavours like XMPP Saint-Andre P(2011), MQTT MQTT Project(2015) and RESTful HTTP are used.IoT connects different type of devices to Internet. As a result, congestion of the network increases exponentially. Therefore, IoT needs a congestion control algorithm to handle the large variety of devices that work in high or low bandwidth and long delay or short delay network.TCP Cubic Has et al.(2008) and Compound TCP Tan K et al.(2008) are the most deployed variants of TCP over the Internet. All the Android and Linux operating systems use TCP Cubic as the transport layer protocol, whereas Microsoft Windows 7 and above version uses the Compound TCP. The Compound TCP utilize the technique of delay based congestion control to adjust the *cwnd* (congestion window) while TCP Cubic utilize a cubic function to estimate the *cwnd* value. TCP Cubic is more aggressive than Compound TCP.

This paper suggests a novel approach STCP for congestion control in IoT environment. Stable TCP have a new window initialization method. A new congestion control method is also proposed in this paper with the help of this new window initialization method. New congestion control approach identifies the congestion before it happens, which minimize the loss of packet and retransmission.

The rest paper is arranged as follows: Section 2 depicts a concise review of various type of algorithms used in congestion control. Section 3 introduces a new *cwnd* (congestion window) initialization and congestion control algorithm. Section 4 represents the suggested approach with performance evaluation and Section 5 defines conclusion of the paper.

Background

Numerous studies have been targeted to enhance TCP performance for different environment. The original TCP Postel J(1981) have a serious problem of congestion collapse because of over and under estimation of the retransmission timer, have no packet loss detection mechanism. To enhance the TCP performance, Jacobson suggested a new policy in Congestion control i.e. TCP Tahoe Jacobson V(1988). It has a new RTO (Retransmission Timeout), a Slow Start, Fast Retransmit and Congestion Avoidance algorithm. When packet loss is identified TCP-Tahoe

minimizes the congestion window to one which degrades the throughput significantly. The above issue is noticed by Allman et al.(1999) and amend the congestion avoidance and initial slow start method by initializing the minor and major congestion event. In Floyd S & Henderson T(1999), authors identified the issue related of TCP Reno and suggested its new version called as TCP New Reno.

Internet congestion is directly proportional to the number of appliances connected to internet. Number of devices connected to internet is increasing tremendously, as a result congestion is also increasing. So TCP requires changes which is suitable for IoT environment and begun connection with good capacity, calibrate the rate of transmission when congestion increases and fix the rate of transmission when devices are fixed. According to different congestion control policy, all the protocol which is used for congestion control at transport layer are categorise in the following two groups:

- **Loss based algorithm for congestion avoidance:** This type of algorithm identify the congestion as a result of packet loss and use same methods for congestion avoidance on the basis of packet loss. JacobsonV(1988), initially proposed a loss based algorithm for TCP and further in Floyd et al.(2004) Fast Retransmit and Fast Recovery technique is provided as TCP Reno.

- **Delay based algorithm for congestion avoidance:** This type of algorithm identify the congestion as a result of delay in network. Congestion is major with the amount of delay, if delay is high congestion is high and if delay is low congestion is low. Many factors are use to identify the congestion with the help of delay are Round Trip Time, one way delay, Queuing delay etc.

Proposed work

We propose a novel congestion control approach which includes initial congestion window initialization, congestion control, and new fast retransmission method.

The TCP keeps assessing the network's capacity of forwarding traffic to the destination. This assessment starts with the initialization of the congestion window (*cwnd).* All TCP flavors initialize the congestion window(*cwnd)* with a fixed value, therefore, it takes relatively long time to reach up to the available bandwidth of the path. There is no proper method available to initialize congestion window(*cwnd)* according to available bandwidth except a fixed value Floyd S et al.(2007) Chuj et al.(2013) DukkipatiNet al.(2010) Sallantin R et al.(2013) Wang G et al.(2014). Here we propose a method for congestion window*(cwnd)* initialization based on the available bandwidth of the current path.

TCP congestion means a drop of packets or absence of getting acknowledgments from the receiver. Thus, we propose a proactive approach to detect congestion rather that just waiting for it to happen. We have based our approach on ECD (early congestion detection) factor which finds the limit up to which *cwnd* can grow. We ensure that the size of the *cwnd* is always within

its set limit, as a byproduct of this approach, the packet drop is avoided.

We compute ECD factor based on available bandwidth, round trip time and queue size of the most congested link on the path.

- The round trip time is the amount of time taken by a packet to make its journey from source to destination and back.
- The bandwidth is calculated using RTT and packet size.
- The queue size of the most congested link is estimated with the help of RTT, RTT_{min} and cwnd.

First of all, we estimate the RTT and RTT_{max}. TCP source calculates RTT, as shown in (1), for each ACK received from TCP sink.

$$RTT = ACK_{\text{Receive\_Time}} - Packet_{Send\_Time} \quad (1)$$

Where RTT is round trip time, $ACK_{Receive\_Time}$ is the time at which acknowledgement receives at TCP source and $Packet_{Send\_Time}$ is the time at which packet is sent from TCP source. RTT_{max} shows, maximum round trip time, that is observed continuously when TCP source received ACK packet. RTT is maximum, when packet drops due to congestion.

Now, on the basis of each received ACK packet, TCP source estimates the maximum capacity of bottleneck link and bandwidth. For the estimation of the bottleneck capacity of the path, we use Packet-pair capacity estimation approach. DovrolisC et al.(2004). In this approach, TCP source estimates the time gap between two packets ACK and the size of the packet. After getting these values, it calculates the capacity for at least 10 samples and then takes the average of it by using (2).

$$Capacity = \frac{Packet_{Size}}{ACK_{i+1} - ACK_i} \quad (2)$$

Where Capacity is the bottleneck maximum capacity, $Packet_{size}$ is the size of the packet sent by the TCP source and ACK_i is the i^{th}ACK time at TCP source.

In TCP, slow start grows the *cwnd* exponentially and in congestion avoidance phase, it grows linearly. We proposed a new method to reduce the congestion using early congestion detection (*ECD)* factor and *Thresh*. In this method, we do not change the *cwnd* if *ECD>Thresh* and if *ECD<Thresh* then it increases *cwnd* with *ssthresh/cwnd*. Congestion control algorithm is shown below.

Algorithm 1. Congestion Control

Step 1 : For each received acknowledgement (ACK)

Step 2: Round Trip Time (RTT) = Current time – Original packet send time
Step 3 : Queue size = Current queue size of congested link
Step 4 : Bandwidth = Current available bandwidth
Step 5 : Capacity = Bottleneck link capacity
Step 6 : Early congestion detection (ECD) = (Capacity - Bandwidth) * Queue size *

Round Trip Time

Step 7 : Threshold = (Capacity – 0.5) * ($Queue_{max}$– 5) * Round Trip Time $_{max}$
Step 8 : If Early congestion detection Threshold
Step 9 : Then no change in congestion window
Step10: Else congestion window will increase by slow start threshold / current congestion window

Result and Discussion

NS-2(Network Simulator -2) is used to evaluate our performance and obtained result for proposed algorithm. We have taken dumbbell topology as our experimental setup as it is most recommended topology used for congestion control testing in IoT environment. Figure-1 shows the dumbbell topology of our approach for validation. This topology has two routers R1 and R2, n number of source nodes (S1, S2…..Sn) and n number of destination nodes (D1,D2….Dn). Different band widths are used for different links. According to requirement the value of source and destination nodes may varies in simulation. Various traffic generators are used in simulation topology like VBR, CBR and FTP. The queuing policies used at node are Drop-tail. The size of the queue is kept 50 packets for each link.

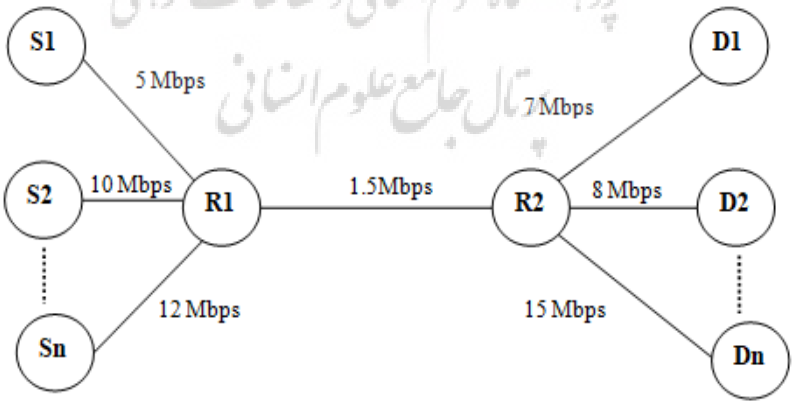

Figure 1. Simulation topology

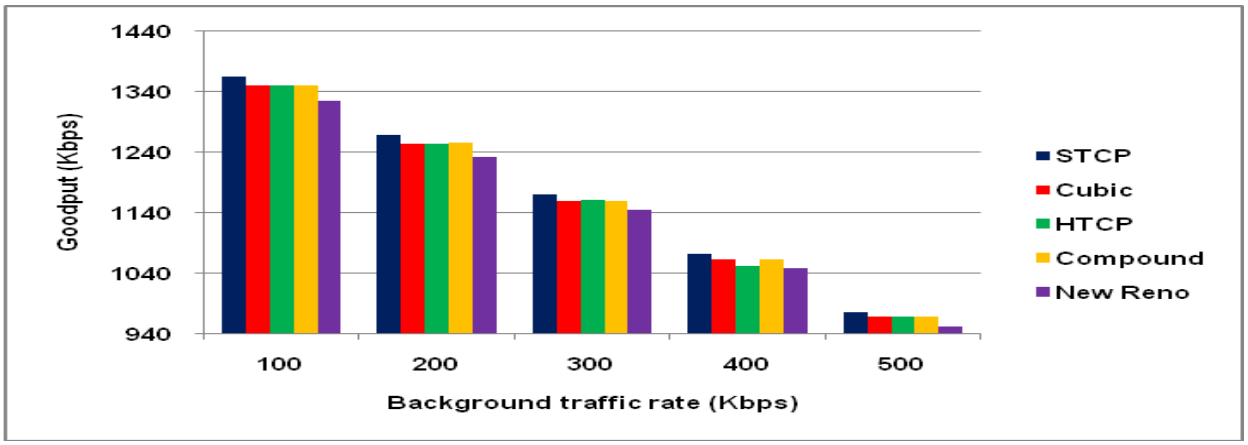

Figure 2. Background traffic Vs Goodput

IoT contains different devices which work in variable traffic environment. Therefore, we investigate the STCP performance in presence of different background traffic varies from 100Kbps to 500Kbps. During the simulation, propagation delay and simulation time is 120ms and 150seconds respectively. For comparative analysis, we use TCP Cubic, HTCP, Compound and New Reno. Fig. 2 shows the goodput of TCP variants when background traffic is variable. It shows that TCP New Reno has minimum utilization for each traffic condition, whereas Compound TCP shows better goodput than TCP Cubic, HTCP, and New Reno. However, STCP achieves better results for each traffic conditions as compared to all the TCP variants use for comparison. STCP gets better goodput due to its ECD congestion detection policy. Fig. 3 shows the comparison of average goodput of STCP with other variants of TCP. It also ensures that STCP achieves better goodput than TCP Cubic, Compound, HTCP and New Reno. The STCP improves goodput up to 2.6%, 0.94%, 1.03 and 1.21% as compared to New Reno, Compound TCP, TCP Cubic and HTCP respectively.

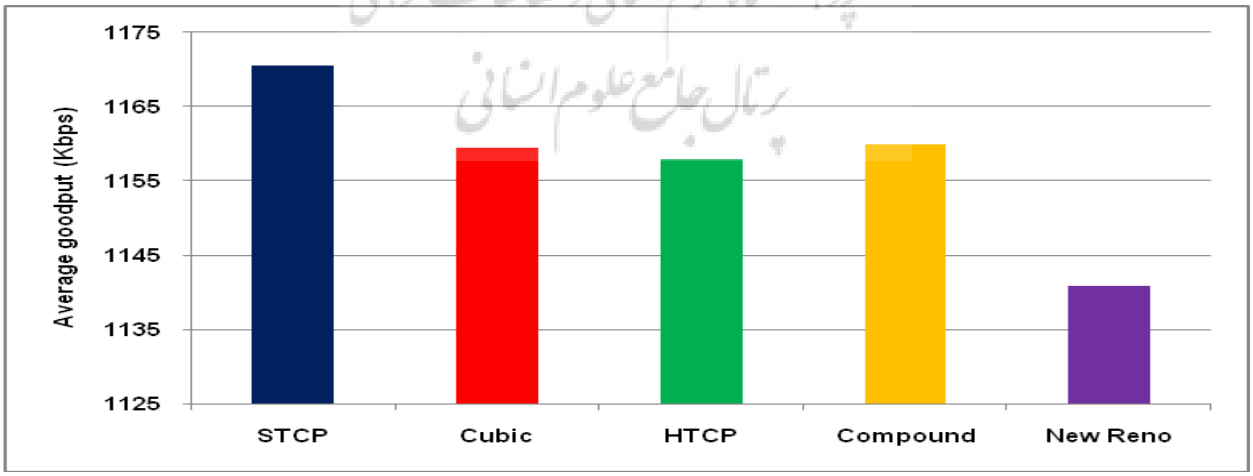

Figure 3. Average goodput of TCP variants

Conclusion

This paper presents STCP, a novel congestion control method for IoT applications. The STCP makes TCP adaptive to control congestion in the different network delay environment. The proposed congestion control method used ECD to adjust the transmission rate in proportion to different network conditions. STCP also has a new way ofcongestion window(*cwnd)* initialization to start TCP with minimal good capacity. Results of simulation show that the STCP achieves better results for goodput. Thus, the STCP meet the requirement of IoT application protocol MQTT, that works formonitoring of smart city, light weight appliances for smart home, healthcare providers and sensors communicating with servers through satellite link.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

References

Allman M,Paxson V, Stevens W (1999) TCP Congestion Control.IETF Internet Drfat.https://tools.ietf.org/html/rfc2581. Accessed 14 June 2019

Chu J, Dukkipati N, Cheng Y, Mathis M (2013) Increasing TCP's Initial Window.IETF Internet Drfat.https://tools.ietf.org/html/rfc6928.Accessed20 April 2019

DovrolisC, RamanathanP, MooreD (2004) Packet-Dispersion Techniques and a Capacity-Estimation Methodology.IEEE/ACM Transaction on Networking. 12:963-977.

DukkipatiN, ReficeT, ChengY (2010)An argument for increasing TCP’s initial congestion window.ACM SIGCOMM Computer Communication Revview. 40:27–33.

Floyd S, AllmanM, Jain A,Sarolahti P (2007) Quick-Start for TCP and IP.IETF Internet Drfat.https://tools.ietf.org/html/rfc4782. Accessed14 June 2019

Floyd S, Henderson T (1999) The NewReno modification to TCP’s fast recovery algorithm.IETF Internet Drfat. https://tools.ietf.org/html/rfc2582.Accessed 14 June 2019

Floyd S, HendersonT, Gurtov A (2004) The NewReno modification to TCP’s fast recovery algorithm.IETF Internet Drfat. http://tools.ietf.org/html/rfc3782. Accessed 14 June 2019.

HaS, Rhee I, Xu L (2008) CUBIC: a new TCP-friendly high-speed TCP variant.ACM SIGOPS Operating

Systems Review. 42:64–74.

JacobsonV (1988) Congestion Avoidance and Control.ACM SIGCOMM Computer Communication Review. 18:314–329.

Khan, Tayyab, Karan Singh, Mohamed Abdel-Basset, Hoang Viet Long, Satya P. Singh, and Manisha Manjul. (2019) "A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks." IEEE Access 7: 58221-58240.

MathisM,Mahdavi J (1996) Forward acknowledgement: refining TCP congestion control.ACM SIGCOMM Computer Communications Review. 26:281–291.

MQTT Project (2015) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.doc.

Postel J (1981) Transmission Control Protocol.IETF Internet Drfat.https://tools.ietf.org/html/rfc793. Accessed 14 June 2019

RESTful HTTP http://www.infoq.com/articles/designing-restful-http-apps-roth.

Saint-Andre P (2011) Extensible Messaging and Presence Protocol (XMPP): Core, https://tools.ietf.org/html/rfc6120.

Sallantin R, Baudoin C, ChaputE, ArnalF, Dubois E,Beylot A (2013) Initial Spreading: a Fast Start–Up TCP Mechanism.in IEEE 38th annual conference on Local Computer Network. 492 – 99.

Tan K, SridharanM, Bansal D, Thaler D (2008) Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks. IETF Internet Drfat.https://tools.ietf.org/html/draft-sridharan-tcpm-ctcp-02. Accessed 19 June 2019

Wang G, RenY, LiJ (2014) An effective approach to alleviating the challenges of transmission control protocol.IET Communication. 8:860-869.

Bibliographic information of this paper for citing:

Gupta, Ajay Kumar; Singh, Devendra; Singh, Karan & Verma, Lal Pratap (2022). STCP: A Novel Approach for Congestion Control in IoT Environment . *Journal of Information Technology Management*, *Special Issue*, 44-51.