Adaptive Congestion Window Algorithm for the Internet of Things Enabled Networks

Ramadevi Chappala[1], Ch.Anuradha[2], Dr.P. Sri Ram Chandra Murthy[3]
Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India[1]
Assistant Professor, Department of CSE, V.R.K. Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India[2]
Assistant Professor, Department of CSE, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India[3]

***Abstract*—Heterogeneous constrained computing resources in the Internet of Things (IoT) are communicated, collected, and share information from the environment using sensors and other high-speed technologies which generate tremendous traffic and lead to congestion in the Internet of Things (IoT) networks. This paper proposes an Adaptive Congestion Window (ACW) algorithm for the Internet of Things. This algorithm is adapted to the traffic changes in the network. The main objective of this paper is to increase the packet delivery ratio and reduce delay while enhancing the throughput which can be attained by avoiding congestion. Therefore, in the proposed algorithm, the congestion window size is depending on the transmission rate of the source node, the available bandwidth of the path, and the receiving rate of the destination node. The congestion window size is altered when the link on the path needs to be shared/released with/by other paths of different transmission in the network. The proposed algorithm, ACW is simulated, evaluated in terms of packet delivery ratio, throughput, and delay. The performance of the proposed algorithm, ACW is compared with IoT Congestion Congrol Algorithm (IoT-CCA) and Improved Stream Control Transmission Protocol (IMP-SCTP) and proved to be better by 27.4%, 11.8%, and 33.7% than IoT-CCA and 44.1%, 22.6%, and 50% than IMP-SCTP concerning packet delivery ratio, throughput, and delay respectively. The variation in congestion window size with time is also projected.**

Keywords*—Congestion window; internet of things; packet delivery ratio; throughput

I. INTRODUCTION

Internet of Things (IoT) is emerging technologies where it connects and shares information globally from kitchen set to cars to industrial tools also target to interrelate and incorporate the physical world and information technology. IoT is the source for every industry to stand in the market and every industry realize that IoT is the key for development in the industries using various IoT technologies like Industrial Internet of Things (IIoT), Internet of Logistics Things (IoLT), Internet of Retail Things (IoRT), Internet of Workforce Management (IoWM) and Internet of Medical Things (IoMT). With these technologies, IoT has an extensive application including smart cities, smart homes, smart communities, utilities and appliances, intelligent transportation, industrial production, E-health, military and environmental monitoring. To acquire the information from various devices of the world, IoT uses capabilities of computing, communication, and perception by utilizing actuators and sensors. These sensors, actuators, or any physical device can be smart objects with the ability to sense, collect data from the environment, communicate and interact with these physical objects. These objects are smart because of their intelligent behaviour for their connection, communication using a wireless protocol. Congestion can be presented in both wired and wireless networks in an IoT environment. And the intelligent environment is created with the interconnection of IoT devices and these IoT devices generate tremendous traffic. Internet of Things (IoT) is attaining enormous study consideration because of the necessity of assimilation of various kinds of networks. Connecting more and more devices to provide various services which share information among them is the main objective of IoT. Presently, every single individual who is associated with the web is utilizing distinctive kinds of specialized gadgets. For two decades, IoT is increasing its popularity and huge work is being carried out by different analysts and business or investors. The objective of IoT is to construct our everyday social lives to be simple [1].

To control the congestion in both wired and wireless networks according to the transmission rate, Transmission Control Protocol (TCP) is the most reliable, connection-oriented transport layer protocol. Network bandwidth and delay modified according to network conditions. There are various application protocols like Advanced Message Queuing Protocol (AMQP) [2], Extensible Messaging Presence Protocol (XMPP) [3], XMPP Representaional State Transfer Hyper Text Transfer Protocol (XMPP RESTful HTTP) and Message Queuing Telemetry Transport (MQTT) [4] are the application protocols presented in Fig. 1, supported by TCP in IoT environment. Another application protocol running over the connectionless, unreliable transport protocol UDP is Constrained Application Protocol (CoAP) is used to provide communication among various gadgets in IoT networks. The open-source community-developed XMPP based on Extensible Markup Language (XML) for instant messaging in a real-time environment which supports the process to process communication among devices in IoT network. MQTT is another application protocol introduced by Arlen Nipper and Dr.Andy Stanford-Clark which is a lightweight protocol for machine-to-machine communication with different services like congestion-controlled, reliable, process2process, and connection-oriented facilities. Among heterogeneous devices in an IoT environment, AMQP provides data transfer services.

Wireless sensor networks (WSN) have been presumed a critical part of communication because of their extraordinary highlights (e.g., versatility and simplicity of association) that make them a significant transporter of information across networks [5-7]. The unique fundamental driver of the lessening in the lifetime of nodes in WSN and reduction in node's energy is due to lack of congestion control [8, 9]. This lessening prompts numerous different issues, for example, delay, loss of packets, and transmission capacity deprivation [10]. Different applications like query-focused, uninterrupted sensing, hybrid applications, event-based, etc are influenced by congestion control [11].

The rest of the paper is structured as follows: the outline of the related work is presented in Section 2. The proposed work is presented in Section 3. Results are discussed in Section 4 and finally, Section 5 concludes the paper.

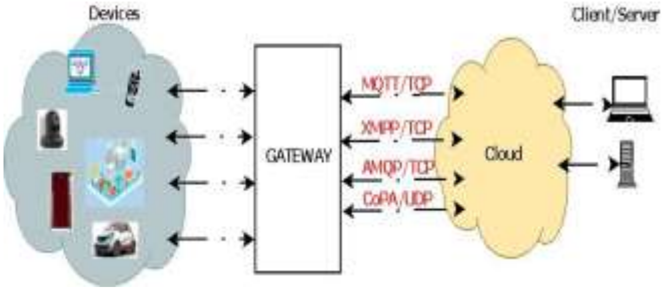

Fig. 1. IoT Application Protocol.

II. Related Work

With the introduction of mobile networks and smartphone technology, services like Internet of Vehicles, Internet of Things, Device to Device communication, and mobile networks undergoes many modifications and also most of the real-time implementations request for maximum throughput and least end-to-end delay which leads to congestion in the network. Wired backbone connections are used to integrate the mobile networks, with these changes the network and transport layers are affected a lot. To control the congestion, TCP accomplished various window-based flow control techniques uses CWND (congestion window) and SSTHRESH (slow-start threshold) which are two state variables. These two variables are used to control the rate of transmission in the network. The objective of CWND is to allow a sender to send data not more than the maximum capacity of the network in any condition and it automatically adjusts to the present network status. Threshold value is provided by slow start threshold variable to control congestion and these two variables are modified by TCP variants. Due to packet loss, the TCP sender identifies that congestion takes place in the network either by duplicate acknowledgment or timeout mechanism. CWND and SSTRHRESHOLD modified by the sender to TCP protocol which plays a key role in IoT development. Different congestion control mechanisms are supported by TCP variants in the IoT environment. In delay networks usage of bandwidth corresponding to the received acknowledgment is a great challenge for researchers. Therefore, in the proposed algorithm, the congestion window size is depending on the transmission rate of the source node, the available bandwidth of the path, and the receiving rate of the destination node. The congestion window size is altered when the link on the path needs to be shared/released with/by other paths of different transmission in the network.

As IoT applications increased day by day and the gadgets which are connected and communicated rose the huge amount of traffic in IoT networks. IoT consists of WSN along with the software as a separate layer that is installed among the computational devices over the cloud. Zigbee is the most familiar WSN protocol based on which IoT is implemented. This paper gives a review of different congestion control procedures utilized at the transport layer. Accessible congestion control procedures, their pros and cons, and prevailing issues with TCP in IoT are also incorporated in [12].

A decision tree (DT) is an AI model that makes more acceptable congestion control in 5G IoT networks. To decide an ideal parametric setting in a 5G network this framework was performed on a training dataset. To improve the behavior of the congestion control method, a dataset was employed to construct the AI model. A decision tree can be used with various capabilities, especially in estimation and grouping. To understand the estimation procedure by any client the DT method will give results [13].

An AI model was used to enhance congestion control, and the methodology delivered an optimistic outcome for the practical and uncertain evaluations of route protocols [14, 15]. Sangeetha et al. [16] suggested a decrease in energy and information misfortune due to congestion over the network. Essentially, the sensor node topology is adjusted intermittently at periodic time interim and node level to upgrade the power utilization of sensor nodes, the intervention, and give an "energy-efficient congestion aware routing procedure for WSNs—specifically, survivable path routing (SPR)".

Singh et al. [17] introduced a new congestion control method towards WSNs but the traditional procedures have more power utilization with more intricacy also got the ideal rate by retransmission with congestion control utilizing the basic Poisson procedure. The routing procedure to choose the ideal path projected by Shelke et al. [18], because of opportunistic hypothesis and by coordinating reasonable rest planning components to diminish congestion in the organization, builds singular node life, the whole organization lifetime, and diminishes division in the organization. Godoy et al. [19] researched and examined the conditions that lead to congestion of the correspondence channel dependent on node setup boundaries: transmission periods, the rate at which packets are generated, and transmitter yield power level.

High-speed TCP is proposed by Floyd in [20] after identifying the effectiveness issue in a high-speed network. This procedure utilizes α to avoid congestion and β as a reduction factor for the duration of trivial loss discovery. The drawbacks of this procedure are dealt with in Scalable TCP which is presented in [21]. The "Multiplicative increase and Multiplicative Decrease" is used in Scalable TCP. The inter-fairness issue is the drawback eliminated in HTCP [22]. A lapse period is introduced before the most recent congestion occurrence in HTCP. BIC-TCP is proposed in [23] which is improved in TCP-CUBIC [24]. For estimating the size of the congestion window and RTT fairness, TCP-CUBIC uses the CUBIC function which is not depending on RTT. It makes use of packet loss as a congestion indicator but does not fully utilize the resources as it is not difficult than H-TCP and HS-TCP and it is used in the Linux kernel. TCP-CUBIC is enhanced in CUBIC-FIT [25].

A novel congestion control strategy is presented in [26] to adjust the transmission rate rapidly at whatever point the accessible transfer speed besides various delay. The suggested approach keeps up a consistent situation to decrease packet loss along with maximizing throughput. And also present versatile procedures to keep up reasonableness with broadly installed TCP Cubic.

A congestion control procedure called TCP Vegas based on delay is developed in [27]. It changes the cwnd as per the distinction between the estimated and the real rate. This procedure increases packet delivery ratio, yet it experiences low data transmission usage in fast networks. This genuine unfairness issue of TCP Vegas is perceived and suggested a new TCP variation known as Vegas$^+$ in [28]. TCP Westwood [29] approximates the accessible transmission capacity dependent on the rate at which acknowledgments are received.

Cheng Ding et al. [30] have suggested a mechanism to allocate cluster-head nodes uniformly, node clustering approach utilizes nodes with maximum traffic and more residual energy depending on energy consumption optimization and energy balancing. Depending on load balance the authors utilize a data forwarding approach to choose suitable routes for various services for delay and service priority specifications to differentiate various services in the network environment.

Al-Janabi et al. [31] have proposed a systematic algorithm depending on load adjustment for IoT-based SDN known as clustering algorithm makes use of storage units and data canters situated on the cloud as cloud resources to evaluate load-balanced PSO clustering algorithm. To build a clustering table, the PSO clustering algorithm utilizes transmission cost, load balancing, and other energy components in the SDN controller where the cluster table utilizes cluster members and cluster heads information of cluster.

Hussien Saleh Altwassi et al. [32] have instigated a well-structured load balancing protocol to improve the life span of the network and to estimate the transmission quality with the metrics power consumption and packet delivery ratio to minimize the congestion in RPL (IPV6 Routing Protocol for Low Power Lossy networks) networks.

Arfath Azeez et al. [33] provide various services to access the server's information and publish/subscribe mechanism to clients which are connected to servers using MQTT cloud-based protocol. The suggested mechanisms incorporate the Application Delivery Controller (ADC) which upgrades and controls the way how the client communicates to the data center to cache/read information or server for refining among the gadgets and the data center. And also takes responsibility to change the route for the entire data to different datacenters when it falls to handle the request or crashes as well as transfer data to the data center having less or nothing when load rises in the network.

Santiago et al. [34] have proposed an energy balancing algorithm that diminishes the energy consumption by selecting the better parent node with the event rate. From the results, it is proved to be that the life span of the parent node and the network is raised and also minimizes the energy consumption.

The main objective of this paper is to increase the packet delivery ratio and reduce delay while enhancing the throughput which can be attained by avoiding congestion. Therefore, in the proposed algorithm, the congestion window size is depending on the transmission rate of the source node, the available bandwidth of the path, and the receiving rate of the destination node. The congestion window size is altered when the link on the path needs to be shared/released with/by other paths of different transmission in the network.

III. Adaptive Congestion Window Algorithm for Internet of Things

The devices in IoT are heterogeneous and also keep increasing and hence more and more communication will be carried out at a time. The IoT devices continuously sense and transmit information. The IoT devices might communicate among themselves or with the cloud or with any other network like hospitals, etc.

Let us consider the devices in the IoT network as nodes and the communication among the nodes/devices will be carried out using wireless links and are represented with the dashed line. The sample scenario is shown in Fig. 2. The communication among the nodes is shown using black color dashed line. The communication between the devices and the network1/network2 is shown using a pink color dash and a single dotted line. The communication with the cloud is shown using a brown color dash and two dotted lines.

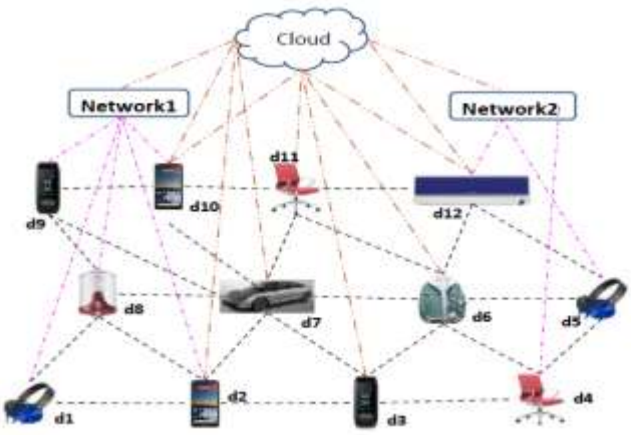

Fig. 2. Sample Scenario of IoT Network.

There might be ‘n’ communications taking place at the time ‘t’ which might be increasing or decreasing at the time ‘t+1’. Hence, the available bandwidth of a particular link will be changing from time to time. These changes in the available bandwidth of the links may lead to congestion in the network if the source node does not adapt its transmission rate / cwnd (congestion window) accordingly. Therefore, in this paper, the cwnd depends on the transmission rate with which the sender can transmit also the available bandwidth of the path and speed with which the receiver can receive the information.

Let us consider:

The transmission rate with which the sender/source node can transmit the information – R_{Tx}

The receiving rate with which the receiver/destination node can receive the information– R_{Rx}

The available bandwidth of the path = min (available bandwidth of the links in the corresponding path) –BW_{avl} -- eqation (1)

Then the congestion window, cwnd = min (R_{Tx}, R_{Rx}, BW_{avl}) -- equation (2)

The selection of the congestion window in this manner will reduce the number of packet drops as there would be a sufficient amount of bandwidth to transmit the information. The packet drops will not be only due to congestion, it might be because of any other network issue like node breakdown, link breakage, etc. This is handled by changing the path of the transmission. So, the cwnd size is modified only there is a change in the available bandwidth.

Initially, the congestion window size is determined using eq. 1 between a set of sender s1 and receiver r1. Later, during the transmission between s1 and r1, if there are any path changes or network changes, then there is a chance of change in the available bandwidth as the common link need to share its available bandwidth amid both the paths in which it is involved. In this regard, the source of the link is responsible to inform the source node of the path to change its cwnd size accordingly.

Hence, the congestion window size is recomputed. The available bandwidth might also change when any link of the selected path is being shared or stopped sharing by any other path of other transmissions as more number of transmissions among various nodes can be carried out synchronously. Periodically, the packet delivery ratio is computed. Whenever the packet delivery ratio is below the threshold value, the congestion window size is reduced. Algorithm 1 is executed when any node is ready for transmission. Algorithm 2 is executed when any node stops or completes its transmission of all the information.

Algorithm 1

```
Input:
Number of nodes – n
A transmission rate of nodes – $R_{Txi}$
Receiving rate of nodes – $R_{Rxi}$
Output:
cwnd size of the path, $p_j$
Begin
1. Determine the path, $p_i$ between the sender$T_{xi}$ and receiverRxi
2. For all paths $p_j$
   a. If any link of the path, $p_i$ is common to the path, $p_j$ then
      i. The available bandwidth is shared among paths, $p_{i,}$ and $p_j$
      ii. The source node of the common link informs about the change in available bandwidth to the source node of the path, $p_i$
      iii. The source node of the path, $p_i$ updates the cwnd size of the path, $p_i$
3. Determine the available bandwidth of all the links of the path determined in 1.a
4. Determine the available bandwidth, $BW_{avli}$ of the path, $p_i$ using the eq. (1)
5. Determine the cwnd size using eq. (2): cwnd = min($R_{Txi}$, $R_{Rxi}$, $BW_{avli}$)
6. Start the transmission
7. After every transmission
   a. If ($PDR_i$<$PDR_{th}$) then
      i. Reduce cwnd size by 10%
8. If any more packets to be transmitted than
   a. Goto 6.
End
```

Algorithm 2

```
Input:
$p_i$ – Transmission stops along this path
Begin
1. For all paths $p_j$
   a. If any link of the path, $p_i$ is common to a path, $p_j$ then
      i. The bandwidth of this common link is released by path, $p_i$
      ii. The source node of the common link informs about the change in available bandwidth to the source node of the path, $p_j$
      iii. The source node of the path, $p_j$ updates the cwnd size of path, $p_j$
End
```

Consider the network shown in Fig. 2.

Assume that,

Node d1 is transmitting the information to node d12 using the path, d1 → d8 → d7 → d6 →d12.

The transmission rate with which the Node d1 can transmit the information – 80Mbps

The rate at which the node d12 can receive the information – 65Mbps

The available bandwidth of this path –72Mbps

Hence, the cwnd size is set to 65 Mbps as min(80,65,72) = 65Mbps.

At this instant, assume that node d2 communicates with d9 using the path, d2 → d7 → d9. As there are no common links, there will no change in the cwnd size of node d1.

After some time, assume that node d4 starts its communication with d9 using the path, d4 → d6 → d7 → d9.

It can be observed that the link d6 → d7 is common in both the communications between d1 – d12 andd4 – d9. Also, the link d7 → d9 is common in both the communications between d2 – d9 and d4 – d9. When node d4 starts its communication, the bandwidth of the links d6 → d7 and d7 → d9 need to be shared among both the paths. Therefore, d6 informs d1 about the change in the available bandwidth of the link, d6 → d7, and d7 informs d2 about the change in the available bandwidth of the link, d7 → d9. Hence, the cwnd size of source nodes d1, d2, and d4 is determined accordingly. Later, when the node d2 completes its communication, the cwnd size of d4 needs to be updated but d1 needs not to be updated.

IV. Results and Discussion

The proposed algorithm, the Adaptive Congestion Window (ACW) algorithm is simulated using NS-2. The simulation is executed for 150s. The topology used for the simulation purpose includes 50 nodes and cloud environments. All the nodes in the network are capable of transmitting, receiving, and forwarding the packets. The bandwidth and propagation delay of links between nodes is different. The traffic type used for simulation purposes includes both CBR and VBR. Both types of traffic are included as the information might be in the form of text, images, audio, or video. The parameters used to evaluate the performance of the proposed ACW algorithm are throughput, packet delivery ratio, and delay for time and congestion window size. The proposed algorithm, ACW proved to be performing better when compared with IMP-SCTP [10] and IoT-CCA[23] in terms of throughput, packet delivery ratio, and delay.

The performance of the proposed algorithm, ACW in terms of packet delivery ratio in comparison with IMP-SCTP and IoT-CCA is shown in Fig. 3 and proved to be performing better. The results are shown at varying times. It can be observed that the packet delivery ratio decreases as time increases. The packet delivery ratio is better when compared to IMP-SCTP and IoT-CCA as the congestion window size is based on the transmission rate of the sender, the available bandwidth of the path, receiving rate of the destination node. As the congestion window size is less than or equal to the amount of bandwidth available, there are fewer chances of packets being dropped. There are some packets dropped because of unavoidable or unexpected issues in the network like link breakage or node breakdown. The performance of ACW is 27.4% better than IoT-CCA and 44.1% better than IMP-SCTP in the case of packet delivery ratio.

The performance of the proposed algorithm, ACW is compared with IMP-SCTP and IoT-CCA in terms of throughput with varying time is shown in Fig. 4. It can be observed that ACW performs better than IMP-SCTP and IoT-CCA. The throughput decreases as time increases. The throughput is dependent on the packet delivery ratio. As there is an improvement in the packet delivery ratio, throughput also enhances. The performance of ACW is 11.8% better than IoT-CCA and 22.6% better than IMP-SCTP in the case of throughput.

The efficiency of the proposed algorithm, ACW is compared with IMP-SCTP and IoT-CCA in terms of delay with varying time is shown in Fig. 5. It can be observed that ACW performs better than IMP-SCTP and IoT-CCA. The delay increases as time increases. The delay occurs due to the minimum propagation delay at nodes in the path. When the number of nodes decreases, then the delay can be further reduced. As the minimum available bandwidth is considered to be the available bandwidth of the path which is one of the parameters to determine the congestion window size, mostly the packets need not be buffered at intermediate nodes. This leads to the reduction of delay in ACW when compared to IMP-SCTP and IoT-CCA. The enhancement of performance of ACW in terms of delay in comparison with IoT-CCA is 33.7% and IMP-SCTP is 50%.

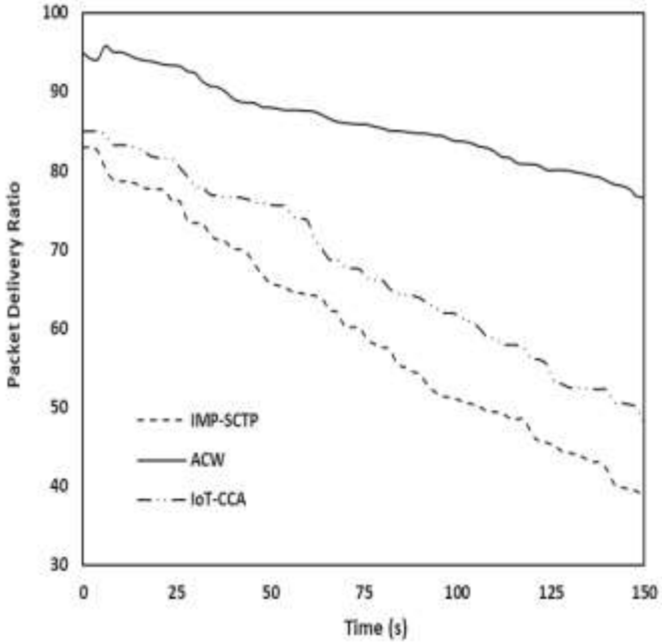

Fig. 3. Packet Delivery Ratio vs Time.

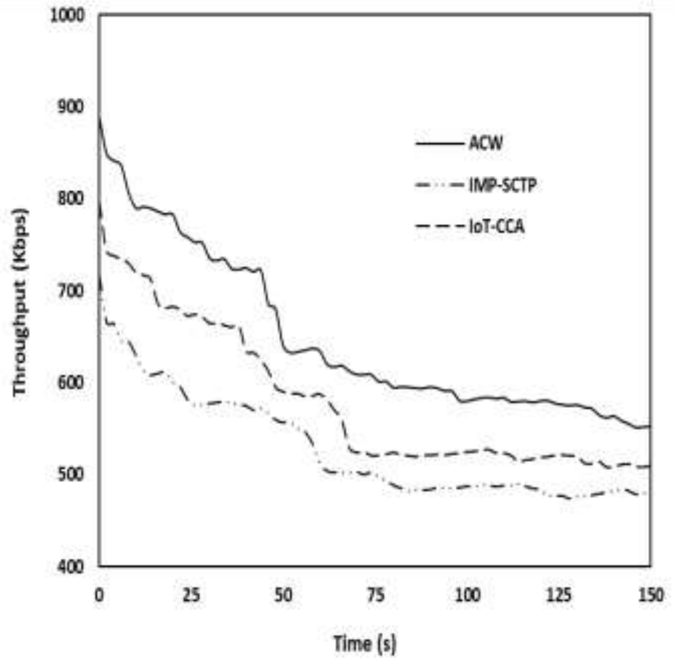

Fig. 4. Throughput vs Time.

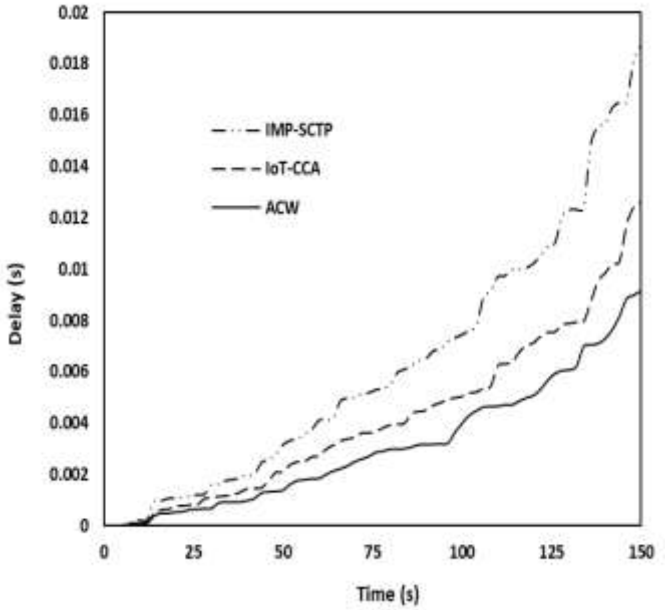

Fig. 5. Delay vs Time.

Fig. 6 shows how congestion window size varies with time for the proposed algorithm, ACW. This graph is plotted for a particular pair of source and destination nodes. The size of the congestion window is either increased or decreased depending on the links in the path and also the links being shared with other paths. The congestion window size is decreased if one of the available links in the path is needed to be shared with another path using which another transmission is about to start. The size of the congestion window is raised if any of the links of the path being shared are released by another path.

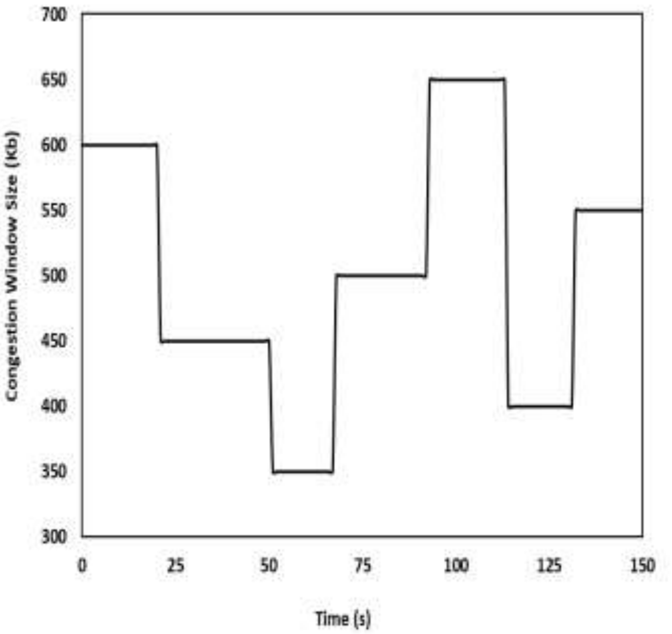

Fig. 6. Congestion Window Size vs Time.

V. Conclusion

For the Internet of Things (IoT), this paper proposes an adaptive congestion window algorithm. Congestion window size (cwnd) of the proposed algorithm, ACW is dependent on the transmission rate of the source node, the available bandwidth of the path, receiving rate of the destination node. The congestion window size of a particular path is increased or decreased with the release/sharing of the available bandwidth of one of the links in the path. The results of the proposed algorithm, ACW are simulated and evaluated in terms of packet delivery ratio, throughput, and delay. The variation of the congestion window size for time is also shown. The performance of the proposed algorithm is compared against IMP-SCTP and IoT-CCA and proved to be better. And in future, this algorithm can be tested against more parameters by including the priority of the nodes to prove that the future results would be better than the proposed work and enhance the performance of network utilization in IoT sensor networks.

Acknowledgment

The author is grateful to Acharaya Nagarjuna University for providing facilities to carry out research work. Also, I would like to acknowledge the Dean of Acharya Nagarjuna University for the comments provided while preparing this work

References

[1] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., & Doody, P. (2011). Internet of things strategic research roadmap. January, pp. 9-52.

[2] Saint-Andre,P.(2011)“Extensible messaging and presence protocol(XMPP): ore,” IETF RFC 6120, https://tools.ietf.org/html/rfc6120 Accessed 12-June-2019.[15]J.Postel,(1981). Transmission Control Protocol. https://tools.ietf.org/html/rfc793. Accessed 12-June-2019.

[3] Saint-Andre,P.(2011).Extensible Messaging and Presence Protocol (XMPP):Core.https://tools.ietf.org/html/rfc6120. Accessed 10 December 2018.

[4] MQTT Version3.1.1 (2015). http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.doc. Accessed 12- December - 2018.

[5] Najm, A.,Ismail, M. Rahem, T. Al Razak, A. (2014). Wireless implementation selection in the higher institution learning environment. J. Theor. Appl. Inf. Technol, 67(2),pp.477-484.

[6] Ismail, M.,Najm, I.A., and Balfaqih,M.(2017).Topology sense and graph-based TSG: efficient wireless ad hoc routing protocol for WANET. Telecommunication Systems, 65(4),pp.739-754.

[7] Aalsalem, M.Y.,Khan, W.Z.,Gharibi, W.,Khan, M.K., andArshad, Q. (2018). Wireless Sensor Networks in oil and gas industry: Recent advances taxonomy requirements and open challenges. Journal of Network and Computer Applications, 113,pp.87-97.

[8] Sunny, A., Panchal, S., Vidhani, N., Krishnasamy, S., Anand, S. V. R., Hegde, M., & Kumar, A. (2017). A generic controller for managing TCP transfers in IEEE 802.11 infrastructure WLANs. Journal of Network and Computer Applications, 93, pp.13-26.

[9] Flora, D.J., Kavitha,V. &Muthuselvi, M. (2011). A survey on congestion control techniques in wireless sensor networks. International Conference on Emerging Trends in Electrical and Computer Technology,1146-1149. IEEE.

[10] Karunakaran,S.,and Thangaraj,P.(2010).A cluster based congestion control protocol for mobile ad hoc networks. International Journal of Information Technology and Knowledge Management, 2(2), pp.471-474.

[11] Kafi, M. A., Djenouri, D., Ben-Othman, J., and Badache, N. (2014). Congestion control protocols in wireless sensor networks: A survey. IEEE communications surveys and tutorials, 16 (3), pp.1369-1390.

[12] Mishra, N., Verma, L. P., Srivastava, P. K., and Gupta, A. (2018). An analysis of IoT congestion control policies. Procedia computer science, 132,pp. 444-450.

[13] Najm, I. A.,Hamoud, A. K., Lloret, J., and Bosch, I. (2019). Machine learning prediction approach to enhance congestion control in 5G IoT environment. Electronics, 8(6),pp. 607-614.

[14] Mirza, M., Sommers, J., Barford, P., and Zhu, X. (2007). A machine learning approach to TCP throughput prediction. ACM SIGMETRICS Performance Evaluation Review, 35(1),pp. 97-108.

[15] Kong, Y., Zang, H., and Ma, X. (2018). Improving tcp congestion control with machine intelligence. In Proceedings of the 2018 Workshop on Network Meets AI & ML pp.60-66.

[16] Sangeetha, G., Vijayalakshmi, M., Ganapathy, S., and Kannan, A. (2018). A heuristic path search for congestion control in WSN. In Industry Interactive Innovations in Science, Engineering and Technology pp.485-495 Springer, Singapore.

[17] Singh, K., Singh, K., and Aziz, A.(2018). Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm. Computer Networks, 138,pp. 90-107.

[18] Shelke, M., Malhotra, A., and Mahalle, P. N. (2018). Congestion-aware opportunistic routing protocol in wireless sensor networks. In Smart computing and informatics , pp.63-72. Springer, Singapore.

[19] Godoy, P. D., Cayssials, R. L., and Garino, C. G. G. (2018). Communication channel occupation and congestion in wireless sensor networks. Computers & Electrical Engineering, 72, pp.846-858.

[20] Floyd,S. (2003). “High Speed TCP for large congestion windows”. https://tools.ietf.org/html/rfc3649.

[21] Kelly, T. (2003). Scalable TCP: Improving performance in high speed wide area networks. ACM SIGCOMM computer communication Review, 33(2),pp.83- 91.

[22] D.Leith,RShorten(2007).“H-TCP:TCP congestion control for high bandwidth-delay product paths”. https://tools.ietf.org/html/draft-leith-tcp-htcp-03.

[23] Xu, L., Harfoush, K., and Rhee, I. (2004). Binary increase congestion control (BIC) for fast long-distance networks. In IEEE Infocom Vol. 4, pp.2514-2524. IEEE.

[24] Ha, S., Rhee, I., and Xu, L. (2008). Cubic: a new TCP-friendly high-speed TCP variant. ACM SIGOPS operating systems review, 42(5), pp.64-74.

[25] Wang, J., Wen, J., Han, Y., Zhang, J., Li, C., and Xiong, Z. (2013). CUBIC-FIT: A high performance and tcp CUBIC friendly congestion control algorithm. IEEE Communications Letters, 17(8), pp.1664-1667.

[26] Verma, L. P., and Kumar, M. (2020). An IoT based Congestion Control Algorithm. Internet of Things, 9, 100157.

[27] Brakmo, L. S., and Peterson, L. L. (1995). TCP Vegas: End to end congestion avoidance on a global Internet. IEEE Journal on selected Areas in communications, 13(8),pp.1465-1480.

[28] Hasegawa, G., Kurata, K., and Murata, M. (2000). Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet. In Proceedings 2000 International Conference on Network Protocols , pp.177-186. IEEE.

[29] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. Y., and Wang, R. (2001). TCP westwood: Bandwidth estimation for enhanced transport over wireless links. In Proceedings of the 7th annual international conference on Mobile computing and networking ,pp.287-297.

[30] Ding, C., Xu, S., Chen, X., Zhou, G., Zheng, P., & Li, Y. (2019). A Delay and Load-balancing based Hierarchical Route Planning Method for Transmission Line IoT Sensing and Monitoring applications. In 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp.207-215. IEEE.

[31] Al-Janabi, T. A., and Al-Raweshidy, H. S. (2017). Optimised clustering algorithm-based centralised architecture for load balancing in IoT network. In 2017 International Symposium on Wireless Communication Systems, 269-274. IEEE.

[32] Altwassi, H. S., Pervez, Z., Dahal, K., and Ghaleb, B. (2018). The rpl load balancing in iot network with burst traffic scenarios. In 2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA) pp. 1-7. IEEE.

[33] Md.ArfathAzeez,D.T. Supreeth Rao and ChinmayKini (2019).Load Balancing and Crash Management in IoTEnvironment,International Research Journal of Engineering and Technology (IRJET), Vol-6, No-3, pp.46-50.

[34] Santiago, S., Kumar, A. D. V., and Arockiam, L. (2018). EALBA: energy aware load balancing algorithm for IoT networks. In Proceedings of the 2018 International Conference on Mechatronic Systems and Robots pp. 46-50.