International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 4, August 2022, pp. 4449~4456
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i4.pp4449-4456

Priority based flow control protocol for internet of things built on light fidelity

Belathuru Ramanna Vatsala, Vidyaraj Chitradurga
Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, India

Article Info

Article history:

Received Jun 10, 2021
Revised Dec 16, 2021
Accepted Jan 11, 2022

Keywords:

Congestion window
Flow control
Internet of things
LiFi
Transmission control protocol

ABSTRACT

Excessive usage of internet by most of the applications that use internet of things (IoT) has resulted in need for high bandwidth network. Light fidelity (LiFi) is one such network having bandwidth in terms of GigaHertz, but LiFi has a limited propagation range hence it can be deployed only in the local area. When IoT nodes are connected using LiFi network in the local area they start pushing large data to the cloud there by arising need for flow control. Some of the IoT applications such as patient monitoring systems and nuclear systems, generate critical data. The protocol for flow control in this case should be based on priority of data since critical data with high priority have to be transmitted first. We develop a flow control protocol named priority based flow control protocol (PFCP) by providing priority to flows that carry critical data especially in IoT system that use LiFi network. We evaluate performance of different transmission control protocol (TCP) variants and modify TCP variant that yields maximum goodput according to the priority based protocol developed and demonstrate that flows that carry critical data are given priority compared to non-prioritized flows.

Corresponding Author:

Belathuru Ramanna Vatsala
Department of Computer Science and Engineering, The National Institute of Engineering
Manandavadi Road, Mysore, Karnataka, India
Email: vatsalabr@nie.ac.in

1. INTRODUCTION

Internet of things (IoT) applications such as smart hospitals, smart security systems, home automation system, smart education systems, generate large amount of data which need to be transmitted as quickly as possible since they have minimum storage capacity [1]. As a consequence, there is a need for wireless network with large bandwidth. Visible light communication (VLC) is a potential solution for this with bandwidth in terms of GigaHertz. Light fidelity (LiFi) is wireless technology with VLC as physical medium. The propagation range of LiFi is around 10 meters and also light cannot penetrate through walls hence it is suitable for indoor environments [2]. IoT applications which can be deployed indoor such as hospital wards, home security systems, and nuclear plants can avail IoT setup upon VLC physical medium. Moreover, VLC is secure since the data is not breached and it is also safe to human eyes.

Since the IoT nodes built on LiFi use high bandwidth only in local area, need for flow control arises when these data are to be transmitted to outside LiFi network which may have comparatively low bandwidth thus, requiring suitable flow control protocol. The flow control protocol in this case should be light weight to accommodate processing constraint of IoT nodes. Context aware applications such as IoT based health monitoring systems [3], nuclear systems and surveillance systems generate data which can be critical at times, these data are to be given priority during data transmission. Our research work is focused on developing flow control protocol based on priority named priority based flow control protocol (PFCP) by modifying the transmission control protocol (TCP) protocol such that prioritized flows avail large window size during congestion compared to other flows. The major contribution of our work is to provide good quality service in terms of goodput, packet delivery ratio to prioritized flows when congestion is anticipated in IoT environment built on high bandwidth LiFi network

Some of the IoT protocols such as lightweight RESTful applications [4], and constrained application protocol (CoAP) [5], are deployed using user datagram protocol (UDP) since it is free from timeouts and retransmissions. However, middle boxes like firewall do not allow UDP flows. This resulted in usage of TCP in IoT applications such as CoAP [6], advanced message queuing protocol (AMQP) [7], message queue telemetry transport (MQTT) [8], and hypertext transfer protocol version 2 (HTTP/2) [9]. We use TCP in our work since it is a prominent protocol with reliability, in-order packet delivery and flow control also, it ensures secure and reliable transmission. Moreover, for applications such as critical health care and military Surveillance systems UDP is not suitable as it lacks quality of service (QoS) requirement like guaranteed delivery. Hence TCP can be a promising transport layer protocol for IoT with guaranteed delivery and flow control. Our work is carried out in three stages: i) First, we evaluate performance of TCP variants namely, NewReno [10], Vegas [11], Veno [12], Bic [13], Scalable [14], Hybla [15], HighSpeed TCP (HTCP) [16], HighSpeed [17], Illinois [18], Yeah [19], Westwood [20] and Westwood plus [21] in single flow IoT environment built on LiFi and find out the variant that yields maximum goodput; ii) Second, we demonstrate that the variant (yielding high goodput) does not guarantee any priority to flows in multi flow environment and some flows randomly monopolies; and iii) Third, we modify this TCP variant according to the priority-based protocol developed and demonstrate that flows that carry critical data are given priority compared to non-prioritized flows when congestion is anticipated. This paper is organized into 6 sections with focus on: related work in section 2, proposed PFCP algorithms in section 3, simulation setup for PFCP in section 4, results and discussions in section 5 and conclusion in section 6

2. RELATED WORK

Several flow control and congestion control algorithms are proposed for IoT paradigm. A mechanism for congestion control in IoT called packet discarding based on node clustering (PDNC) is suggested in [22], where congestion is controlled by discarding selected packets. The packets to discard are based on data size larger than 60 bytes or with time to live field 0. This method ensures lower packet loss percentage and lower end to end delay in comparison with the original method.

A network system is proposed in [23], where the authors concentrate on information centric networking based architecture instead of existing transmission control protocol/internet protocol (TCP/IP) based architecture. Here a dynamic congestion control mechanism is used to transmit the data packet. A data packet is delivered with appropriate time delay according to priority level, life time and content popularity. Caching partition is made based on content popularity. Congestion rate is reduced through diminished data traffic and high performance is achieved with low packet drop rate, high cache hit rate and throughput.

TCP congestion control for IoT over heterogeneous network model called Siam is designed in [24] using Markov decision process. Here congestion window is updated by calculating probability of packet loss and packet delay. Congestion window size is reduced during CA_Loss state where packets sent are lost due to congestion and CA_Recovery state where duplicate sequence numbers are received. The results of simulation show maximum window size, in-flight bytes and transmitted bytes for Siam compared to TCP variants veno, scalable, yeah, illinois and westwood plus.

An adaptive congestion control for IoT is proposed in [25] by improving random early detection (RED) algorithm based on Bell type fuzzy membership function. This algorithm automatically adjusts the probability of packet loss according to the network environment. The packets with lower curve probability are discarded, which can reduce congestion and improve utilization of link compared to RED which discards packets with high linear probability. The proposed algorithm assures high throughput, low packet loss and delay compared to S-RED, F-RED, BLUE and RED algorithms.

A policy for congestion control based on IoT is introduced in [26] where the transmission rate is adjusted according to the requirement of IoT device by combining both reactive and proactive congestion control policies. Here a new congestion window initialization mechanism is proposed based on changes in available bandwidth and delay. Initial window size is made as BW/α where BW is available bandwidth and α is sensitivity factor instead of 1 maximum segment size (MSS). The proposed method also includes adaptation of TCP to utilize available bandwidth according to the application demand and also maintains steady state by incrementing congestion window by a factor x, where x is calculated based on the product of current round-trip time (RTT) and slow start threshold (ssthresh). The proposed method yields better results in terms of throughput and inter-protocol fairness.

A congestion avoidance routing protocol based on priority is proposed in [27] for IoT based wireless body area networks. Here data traffic is classified into two types normal and emergency traffic, emergency packets are labeled with priority and are scheduled for transmission by using shortest hop count route. The proposed protocol assures that emergency packets are routed with minimum delay and higher throughput.

A link level flow control protocol based on priority is specified in IEEE standard 802.1Qbb [28], it is applicable for data center bridging network. When receiver buffer overflows, it sends pause frame to sender nodes. The pause frame contains pausing time interval. PFCP allows different pausing time based on class of service, which is derived based on individual priority of flows.

3. PROPOSED PFCP ALGORITHMS

PFCP algorithms are developed at source, destination and router level, source level algorithm is shown in Figure 1. The applications such as health monitoring system, temperature tracking in nuclear systems that are running at the source IoT device calculate the priority of the flow based on criticality of data and furnishes this information to underlying transport protocol. Options field used for priority in the header is marked as ‘prio’ if it is of high priority based on information received from application layer. Destination level algorithm is shown in Figure 2. The receiver copies the priority information of the received segment into its corresponding acknowledgment packet (ACK) segment. The PFCP is designed in such a way that there is minimum computation at source and destination as they are constrained with respect to memory and processing capabilities.

Router level algorithm is shown in Figure 3. The router checks the queue occupancy and if current queue size is below the threshold Th, then enters flow control phase otherwise it will be in normal phase, during flow control phase ACK segments marked with ’prio’ are not disturbed but segments not marked with ‘prio’ are to undergo window tailoring. In this phase receiver window size of segments not marked with ‘prio’ is tailored proportionate to available bandwidth as mentioned in (1):

$$BA = BW - U$$

$$Wrn = BA/BW * Wr$$

$$Wr = Maximum\ (Wrn, 1\ MSS) \tag{1}$$

where BA is available bandwidth, BW is Bandwidth of the link, U is used bandwidth, Wr is receiver window size, and Wrn is new receiver window calculated. The set of equations make sure that Wr is at least made equal to 1 MSS. The reduction in the receiver window size of non-prioritized segments alleviates congestion and more bandwidth is given to packets with high priority there by achieving flow control based on priority. The flow diagram of the protocol is depicted in Figure 4.

```
Algorithm 1: Priority based flow control Algorithm at the source
Input: incoming packet output: prioritized packets are marked

1. Begin
2. If Priority for the packet is set by application
3. Then Seg.Mark <- prio
4. End If
5. End
```

Figure 1. PFCP algorithm at source node

```
Algorithm 2: Priority based flow control Algorithm at the Receiver
Input: incoming packet
output: Ack packets of prioritized packets are marked

1. Begin
2. If Seg.Mark =Prio
3. Then
4. ACK Seg.Mark=Prio
5. End If
6. End
```

Figure 2. PFCP algorithm at destination node

```
Algorithm 3: Priority based flow control Algorithm at the router
Input: incoming packet
output: receiver window updating during congestion

1. Begin
2. If Queue_Size > Th
3. Then Return
4. End If
5. SEG <- Queue.get_first
6. If SEG_TYPE != ACK
7. Then Return
8. End If
9. If SEG_MARK = Prio
10. Then Return
11. End If
12. BA <- BW-U
13. Wrn <- BA/BW*Wr
14. If Wrn< 1 MSS
15. Then Wrn <- 1 MSS
16. End If
17. Wr <- Wrn
18. End
```

Figure 3. PFCP algorithm at router

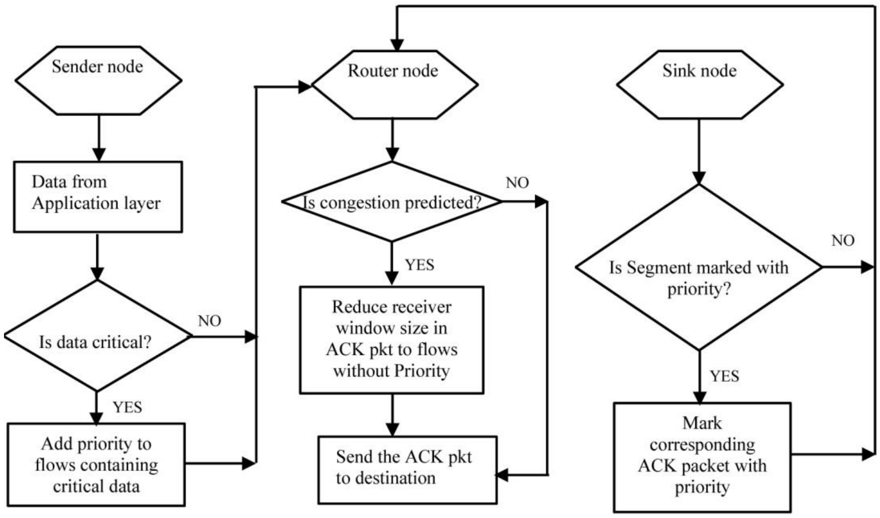

Figure 4. PFCP flow diagram

4. RESEARCH METHOD: SIMULATION SETUP FORPFCP

IoT system is built over LiFi containing VLC links in NS3 by using IoT protocol stack and VLC module [29]. It consists of ten source nodes (S1 to S10), ten sink nodes (R1 to R10) connected via two gateway nodes (G1 and G2) in dumbbell shape topology as shown in Figure 5. Network layer protocol for IoT namely 6LoWPAN [30], is installed on all nodes with IPV6 addressing scheme. The source nodes are connected to gateways through VLC channels. Transmitter and Receiver devices are attached to each VLC channel. The two gateway nodes are connected using point to point wired Link. The Sink nodes are also connected to gateway through point-to-point wired links. The network parameter values used for simulation are as shown in Table 1. BulkSend applications namely flow1to flow10 are run between source nodes S1 to S10 and receiver nodes R1 to R10 respectively. The channels are set with error model such that receiver error value (Rerror) is 0.0 in first case and 0.1 in second case.

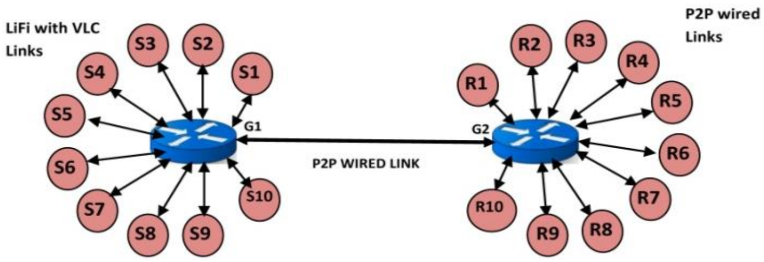

Figure 5. Simulation topology

Table 1. Network parameters

Parameters	Value
Bandwidth and channel delay between source and gateway	10 Mbps, 0.01 ms
Bandwidth and channel delay between gateways	2 Mbps, 45 ms
Bandwidth and channel delay between sink and gateway	2 Mbps, 45 ms
Receiver error value (Rerror)	0.0 and 0.1
Packet Size	1024 Bytes
Socket Buffer size	4 MB

5. RESULTS AND DISCUSSION

The simulation is run in three stages: stage 1, stage 2 and stage3. In stage 1, only flow 1 is run between source node S1 and sink node R1 (single flow), the simulation is run for 100 seconds in both error free and error prone cases (Rerror 0.0 and Rerror 0.1) by considering the TCP variants viz NewReno, Vegas, Veno, Bic, Scalable, Hybla, HTCP, HighSpeed, Illinois, Yeah, Westwood and Westwood plus. Figure 6 shows the plot of goodput values for TCP-variants considered in the research work with respect to flow 1. It is evident from the resulting graph that TCP-HighSpeed exhibits highest goodput of 364.63 Kbps in first case and fairly well i.e., 33.19 Kbps in second case since it is basically designed to increase throughput in high bandwidth network like VLC. Hence we use TCP high speed in the next stages of simulation.

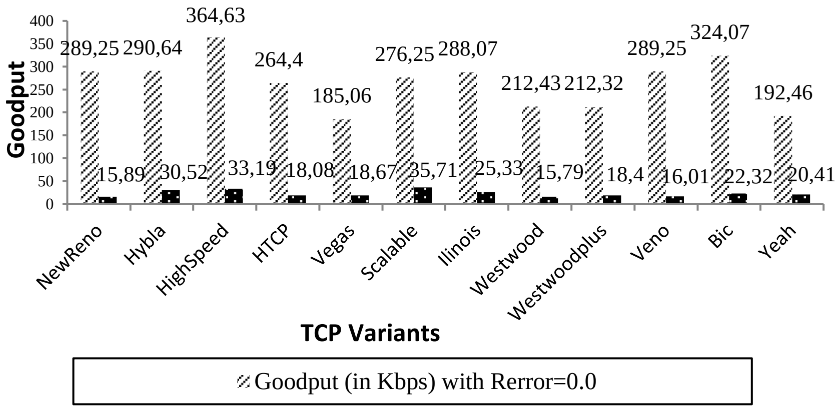

Figure 6. Goodput of TCP Variants in error free and error prone network

In the second stage flow1 to flow10 are run between source nodes S1 to S10 and receiver nodes R1 to R10 respectively (multi flow). The simulation is run for 100 seconds using TCP HighSpeed as the transport layer protocol. The goodput of each flow from flow1 to flow10 is plotted as shown in Figure 7 in both cases. It is noticed that flow4, flow10 and flow1 exhibit maximum goodput of 215.55 Kbps, 213.46 Kbps and 203.96 Kbps respectively in error free case and flow5, flow8 and flow3 exhibit maximum goodput of 18.62 Kbps, 15.89 Kbps and 14.28 Kbps respectively in error prone case. This random flow monopolization behavior of TCP makes it infeasible for the scenarios where we need to prioritize particular flows that are critical.

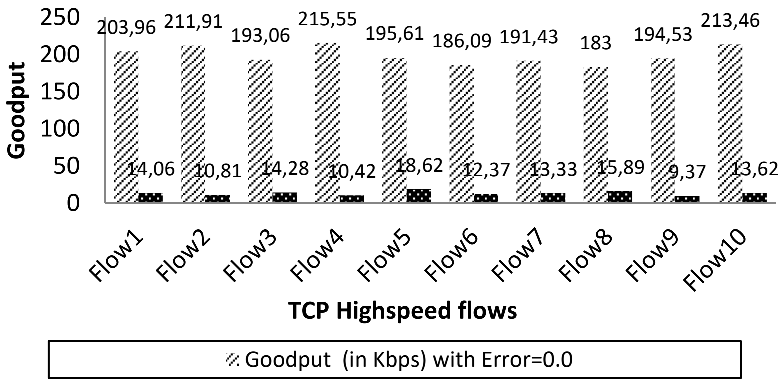

Figure 7. Goodput of flows with TCP Highspeed in error free and error prone network

The packet delivery ratio (PDR) of each flow is as depicted in Figure 8, It is apparent that flow 2 exhibits PDR of 99.02% which is slightly more than other flows in error free case, and PDR of flow1 is 83.57% in error prone case (highest among all flows). Thus, we observe maximum PDR for some random flows in TCP HighSpeed. In the third stage, PFCP is employed on all source, router and receiver nodes by modifying TCP HighSpeed. Flow4 and flow 9 are made prioritized flows out of the 10 flows. The simulation is run for 100 seconds. The goodput of each flow from flow1 to flow10 is plotted as shown in Figure 9 in both cases. It is apparent from the plot that flow 9 and flow 4 have highest goodput of 426.24 Kbps and 416.48 Kbps respectively in error free case and 52.4 Kbps and 51.96 Kbps in 10% error prone case compared to other flows. This is due to tailoring of receiver window for non-prioritized flows during congestion. Thus, flows that carry critical data assure highest data transmission compared to non-prioritized flows

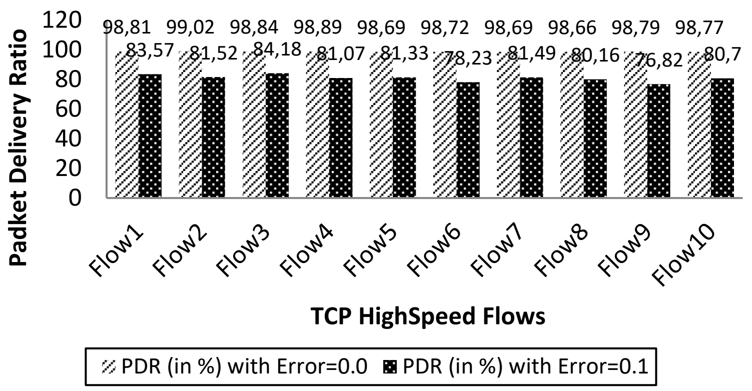

Figure 8. PDR of flows with TCP HighSpeed in error free and error prone network

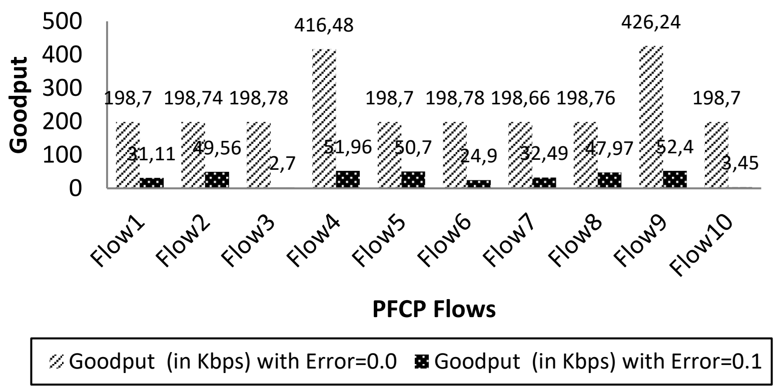

Figure 9. Goodput of flows with PFCP in error free and error prone network

The packet delivery ratio (PDR) of each flow employing PFCP is shown in Figure 10. Flow 4 and flow9 have PDR of 100% and 99.97% which is slightly more than that of other flows. Thus, PFCP also ensures maximum PDR for prioritized flows compared to non-prioritized flows. The results prove that in TCP HighSpeed flows monopolies randomly whereas prioritized flows dominate when PFCP is employed.

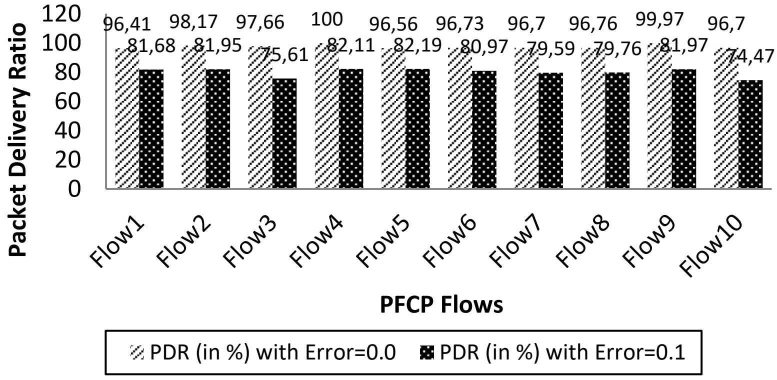

Figure 10. PDR of flows with PFCP in error free and error prone network

6. CONCLUSION AND FUTURE ENHANCEMENTS

Modification to TCP protocol called PFCP is suggested to cater to the need of priority for flows that carry critical data in IoT environment built on LiFi. The protocol developed is aimed at reducing the receiver window size of non-prioritized flows without affecting that of prioritized flows to enforce flow control during congestion. At first, TCP variants viz NewReno, Vegas, Veno, Bic, Scalable, Hybla, HTCP, HighSpeed, Illinois, Yeah, Westwood and Westwood plus were evaluated with reference to the performance parameter goodput in both error free and error prone cases in IoT with VLC medium for a single flow scenario. It was found that in error free case, TCP HighSpeed achieves highest goodput of 364.63 Kbps and also performs well in error prone case with goodput of 33.19 Kbps compared to other flows. Later, in multi flow environment TCP HighSpeed was used as a transport layer protocol to evaluate goodput and PDR of each flow and it was noticed that some flows monopolize the network, even though they are not associated with any priority. Finally, PFCP has been employed by modifying TCP High speed. This approach assures that flows with priority have increased goodput and slightly increased PDR compared to non-prioritized flows. The protocol developed can be implemented in real time for IoT applications that require priority during transmission. The protocol can be augmented with security since IoT devices are prone to attacks and also the memory and processing constraints of IoT systems must be considered during this enhancement

REFERENCES

[1] A. Ghasempour, "Internet of things in smart grid: architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, pp. 22–33, Mar. 2019, doi: 10.3390/inventions4010022.

[2] H. Haas *et al.*, "Introduction to indoor networking concepts and challenges in LiFi," *Journal of Optical Communications and Networking*, vol. 12, no. 2, p. A190, Feb. 2020, doi: 10.1364/JOCN.12.00A190.

[3] A. O. Bang and U. P. Rao, "Context-aware computing for IoT: history, applications and research challenges," Springer Singapore, 2021, pp. 719–726.

[4] L. Richardson and S. Ruby, *RESTful web services*. O'Reilly, 2008.

[5] IETF, "rfc7252 - IETF Tools," *datatracker.ietf.org*. https://tools.ietf.org/html/rfc7252 (accessed Feb. 02, 2020).

[6] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor, "CoAP (constrained application protocol) over TCP, TLS, and WebSockets," *IETF*. https://tools.ietf.org/id/draft-ietf-core-coap-tcp-tls-11.html (accessed Feb. 03, 2020).

[7] ISO, "ISO/IEC 19464:2014(en), Information technology," *iso.org*. https://www.iso.org/obp/ui/#iso:std:iso-iec:19464:ed-1:v1:en (accessed Feb. 03, 2020).

[8] ISO, "ISO/IEC 20922:2016(en), Information technology," *iso.org*. https://www.iso.org/obp/ui/#iso:std:iso-iec:20922:ed-1:v1:en (accessed Feb. 03, 2020).

[9] and M. T. M. Belshe, R. Peon, "Hypertext transfer protocol version 2 (HTTP/2)," May 2015. doi: 10.17487/RFC7540.

[10] S. Floyd, T. Henderson, and A. Gurtov, "Rfc3782: The NewReno modification to TCP's fast recovery algorithm," *IETF*, 2004. https://datatracker.ietf.org/doc/rfc3782/ (accessed Feb. 03, 2020).

[11] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995, doi: 10.1109/49.464716.

[12] Cheng Peng Fu and S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216–228, Feb. 2003, doi: 10.1109/JSAC.2002.807336.

[13] Lisong Xu, K. Harfoush, and Injong Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *IEEE INFOCOM 2004*, 2004, vol. 4, pp. 2514–2524, doi: 10.1109/INFCOM.2004.1354672.

[14] T. Kelly, "Scalable TCP," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83–91, Apr. 2003, doi: 10.1145/956981.956989.

[15] C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, pp. 547–566, Sep. 2004, doi: 10.1002/sat.799.

[16] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," *Procceedings of PFLDnet*, 2004.

[17] S. Floyd, "HighSpeed TCP for large congestion windows," *RFC Editor, RFC3649*. Dec. 2003, doi: 10.17487/rfc3649.

[18] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks," *ACM International Conference Proceeding Series*, 2006, doi: 10.1145/1190095.1190166.

[19] A. Baiocchi, A. P. Castellani, and F. Vacirca, "YeAH-TCP: yet another highspeed TCP," *Proc. PFLDnet*, 2007.

[20] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood," in *Proceedings of the 7th annual international conference on Mobile computing and networking - MobiCom '01*, 2001, pp. 287–297, doi: 10.1145/381677.381704.

[21] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 25–38, Apr. 2004, doi: 10.1145/997150.997155.

[22] N. H. B. Halim, N. B. Yaakob, and A. B. A. M. Isa, "Congestion control mechanism for internet-of-things (IOT) paradigm," in *2016 3rd International Conference on Electronic Design (ICED)*, Aug. 2016, pp. 337–341, doi: 10.1109/ICED.2016.7804663.

[23] R. Sukjaimuk, Q. N. Nguyen, and T. Sato, "Dynamic congestion control in information-centric networking utilizing sensors for the IoT," in *2018 IEEE Region Ten Symposium (Tensymp)*, Jul. 2018, pp. 63–68, doi: 10.1109/TENCONSpring.2018.8691983.

[24] T. Toprasert and W. Lilakiataskun, "TCP congestion control with MDP algorithm for IoT over heterogeneous network," in *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, Sep. 2017, pp. 1–5, doi: 10.1109/ISCIT.2017.8261189.

[25] L. Chen and J. Cao, "Adaptive congestion control of internet of things based on improved RED algorithm," in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 295–298, doi: 10.1109/CAC.2018.8623124.

[26] L. P. Verma and M. Kumar, "An IoT based congestion control algorithm," *Internet of Things*, vol. 9, Mar. 2020, doi: 10.1016/j.iot.2019.100157.

[27] K. M. Awan *et al.*, "A priority-based congestion-avoidance routing protocol using IoT-based heterogeneous medical sensors for energy efficiency in healthcare wireless body area networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, Jun. 2019, doi: 10.1177/1550147719853980.

[28] IEEE Standard, "802.1Qbb-2011 - IEEE Standard for Local and metropolitan area networks--media access control (MAC) bridges

and virtual bridged local area networks--Amendment 17: Priority-based flow control," *802.1Q-2005), IEEE Std 802.1Q-2011 (Revision of IEEE Std*. 2011, doi: 10.1109/IEEESTD.2011.6032693.

[29] A. Aldalbahi, M. Rahaim, A. Khreishah, M. Ayyash, and T. D. C. Little, "Visible light communication module: an open source extension to the ns3 network simulator with real system validation," *IEEE Access*, vol. 5, pp. 22144–22158, 2017, doi: 10.1109/ACCESS.2017.2759779.

[30] J. Hu, D. Culler, and S. Chakrabart, "6lowpan: incorporating IEEE 802.15. 4 into the IP architecture," *IPSO Alliance.WhitePaper*. pp. 1–17, 2009.

BIOGRAPHIES OF AUTHORS

Belathuru Ramanna Vatsala Assistant Professor, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India. Teaching and research interest areas are internet of things, computer architecture, compiler design, microprocessors, embedded systems and computer networks. Presented papers in Conferences and published papers in International Journals. Awarded with Elite Certificate with gold medal printed on it for scoring 90% in NPTEL online Course on "Introduction to Internet of Things". She can be contacted at email: vatsalabr@nie.ac.in.

Vidyaraj Chitradurga Professor, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India. Research and Teaching Interest areas are Quantum Computing, Software Engineering Computer Networks, Wireless networks, Storage Area Networks and Internet of Things. Presented papers in various Conferences and published papers in reputed International journals. She is life member of Indian society for Technical Education, Fellow member of Institute of Engineers. Received best paper awards for research papers. She has guided and is guiding research scholars. She has given several technical talks in various institutions. She can be contacted at email: vidyarajc@nie.ac.in.