

Research Article

Wireless Networking Performance in IoT Using Adaptive Contention Window

R. M. Bhavadharini,^{1,2,3} S. Karthik,^{1,2,3} N. Karthikeyan,^{1,2,3} and Anand Paul ^{1,2,3}

¹Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, India

²SNS College of Engineering, Coimbatore, India

³The School of Computer Science and Engineering, Kyungpook National University, Republic of Korea

Correspondence should be addressed to Anand Paul; paul.editor@gmail.com

Received 18 November 2017; Revised 22 March 2018; Accepted 8 April 2018; Published 3 July 2018

Academic Editor: Jaime Lloret

Copyright © 2018 R. M. Bhavadharini et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things (IoT) network contains heterogeneous resource-constrained computing devices which has its unique reputation in IoT environments. In spite of its distinctiveness, the network performance deteriorates by the distributed contention of the nodes for the shared wireless medium in IoT. In IoT network, the Medium Access Control (MAC) layer contention impacts the level of congestion at the transport layer. Further, the increasing node contention at the MAC layer increases link layer frame drops resulting in timeouts at the transport layer segments and the performance of TCP degrades. In addition to that, the expiration of maximum retransmission attempts and the high contentions drive the MAC retransmissions and the associated overheads to reduce the link level throughput and the packet delivery ratio. In order to deal with aforementioned problems, the Adaptive Contention Window (ACW) is proposed, which aims to reduce the MAC overhead and retransmissions by determining active queue size at the contending nodes and the energy level of the nodes to improve TCP performance. Further, the MAC contention window is adjusted according to the node's active queue size and the residual energy and TCP congestion window is dynamically adjusted based on the MAC contention window. Hence, by adjusting the MAC Adaptive Contention Window, the proposed model effectively distributes the access to medium and assures improved network throughput. Finally, the simulation study implemented through ns-2 is compared with an existing methodology such as Cross-Layer Congestion Control and dynamic window adaptation (CC-BADWA); the proposed model enhances the network throughput with the minimal collisions.

1. Introduction

In the IoT realm, there are still many challenges open in innovative research as well as technical solutions to interoperability, integration, and interconnection of heterogeneous IoT systems, at a different level. Homogeneous cross platform domain in various IoT applications hinders the developer with various challenges including the cost, scalability, and reusability of IoT solutions. Figure 1 scheme contributes to the wireless technology which in turn supports the interoperability of IoT systems.

A smart object can be a sensor, a thing, or any physical device that has the capability to sense the environment, to collect the data from it, and to interact and communicate with any other physical device. These devices are called smart since they show intelligent behavior in connection and

interaction with other devices via some wireless protocol and operate interactively and autonomously over the Internet [1–4]. Examples of these devices include, but are not limited to, smart watches, building automation sensors, medical and healthcare devices, smartphones, smart TVs, vehicles, and security system (Figure 2). These devices need to transmit the data by capturing the medium in a fair manner. The contention window in the MAC layer provides backoff time when the medium is busy. The backoff time is set according to node's queue size and the energy level to reduce MAC layer retransmissions. This current research approach in Adaptive Backoff Window for MAC model can contribute to the overall systems.

The distributed access to the shared wireless medium is controlled by Medium Access Control (MAC) protocol. Ad hoc based networks deploy IEEE 802.11 standard MAC

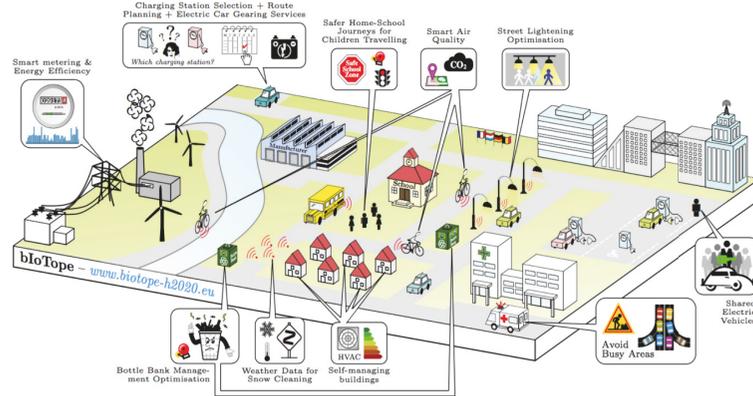


FIGURE 1: Pan European Interoperable IoT System.

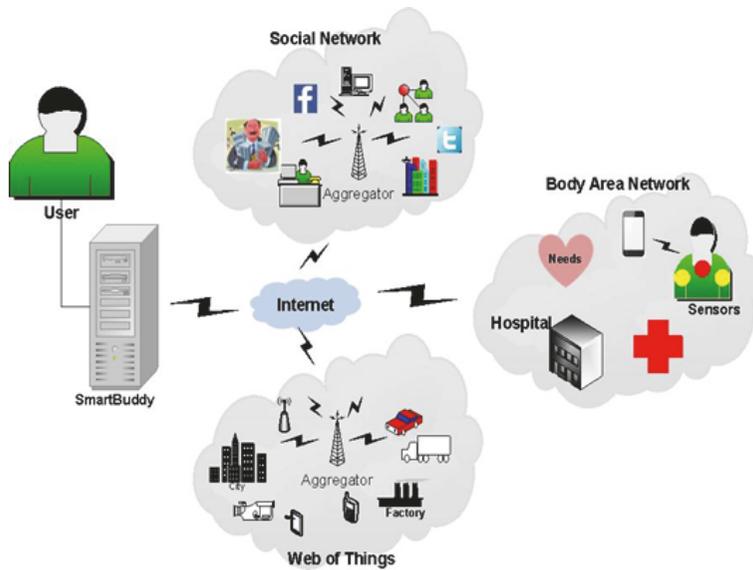


FIGURE 2: An overview of designed system.

protocol for ascertaining single hop reliability. IEEE 802.11 MAC uses two different access mechanisms. Distributed Coordinated Function (DCF) allows asynchronous data transfer from the nodes. No controlling infrastructure is used in DCF communication. Point Coordinated Function (PCF), on the other hand, has a central coordinator to control access to the shared medium.

The basic access mechanism used in DCF is Carrier Sense Multiple Access Collision Avoidance (CSMA/CA). In CSMA/CA channel state is sensed by the node before data transmission. The transmission is allowed or deferred based on either the channel is free or idle. Nodes in the sensing range of each other end up with collided packets when they sense the medium as idle and transmit at the same time. CSMA/CA uses a positive acknowledgment from the receiver to guarantee the reception of the transmitted frame. DCF features an exchange of control packets Ready to Send/Clear to Send (RTS/CTS) between the sender and the receiver to alleviate the hidden node or hidden terminal problem [5].

To transmit a frame, the sender transmits RTS signal to the receiver if the medium is sensed idle after DIFS interval time and random backoff time (BT). BT value is decremented after every slot if the medium is found to be idle; otherwise, BT is frozen until next DIFS (DCF Interframe Spacing) period of channel's idle time. The random value of backoff time reduces the chances of collisions among the frames. The receiver responds with CTS signal after SIFS (Short Interframe Spacing) period when the medium is free. After receiving CTS from the receiver, the sender sends DATA frame after SIFS and ACK are received from the receiver after SIFS. This sequence of message exchange is termed as four-way handshaking and successful data transmission includes DIFS+RTS+CTS+DATA+ACK+3SIFS time period.

The following sequence depicts successful frame transmission of IEEE 802.11 RTS/CTS access method as depicted in Figure 3:

t_0-t_1 : STA1 senses the medium as idle for DIFS time

t_1-t_2 : STA1 enters the contention window

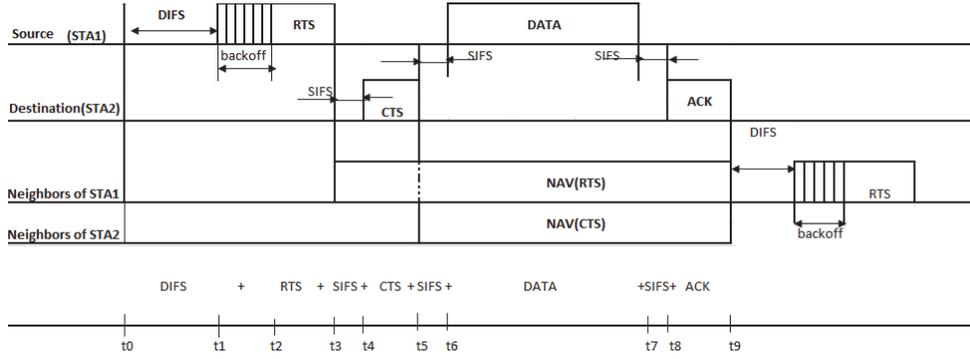


FIGURE 3: IEEE 802.11 DCF using RTS / CTS.

- t2-t3: STA1 broadcast RTS
- t3-t4: SIFS for transmitting CTS
- t4-t5: CTS from STA2
- t5-t6: SIFS before transmitting DATA
- t6-t7: DATA transmitted from STA1 to STA2
- t7-t8: SIFS before ACK
- t8-t9: ACK sent from STA2 to STA1

If the RTS sender does not receive CTS for transmitted RTS or ACK for DATA frame, then the retransmission is triggered until the maximum retransmission limit (7 for control frames and 4 for data frames) after which the frame (either control frame or data frame) is dropped. This scenario is presumed as packet drop due to link failure at the MAC level and considered as the route failure at the network layer.

As a result, TCP runs over IEEE 802.11 suffer a timeout because of the MAC contention losses and are wrongly perceived as congestion. Thus, congestion control measures are invoked resulting in retransmissions at the transport layer and degraded network throughput. Therefore, the contention in MANET impacts the TCP performance clearly. MAC layer uses Binary Exponential Backoff algorithm [BEB] that selects backoff counter value as $random [0, CW]$ where CW is the current contention window. $CW \in (CW_{min}, CW_{max})$ and the initial value of CW is set to CW_{min} and doubled after every unsuccessful attempt until reaching a value of CW_{max} [6, 7]. BEB prevents simultaneous access to the wireless medium by randomizing the access times through random backoff procedure. The values of CW_{min} for 802.11a and 802.11b are 16 and 32, respectively. The value of CW_{max} is 1024 in both cases [8, 9].

BEB works as follows:

$$CW = \begin{cases} CW_{min} & \text{Initial value} \\ CW_{min} & \text{After successful frame transmission or max retry limit} \\ \text{Min}(2 * CW_{old}, CW_{max}) & \text{After unsuccessful frame transmission} \end{cases} \quad (1)$$

$$BT = random[0, CW] * Channel_slot_time.$$

$Channel_slot_time$ is dependent on the physical layer characteristic of the node.

In BEB, the nodes with successful transmissions get the minimum backoff time, leaving them to access the medium with fewer contention slots. Other nodes deferring access to the shared medium and their increased contention window suffer more packet drops after maximum retry limit (short retry count for the control signals and long retry count for data signals). Retransmissions initiated for the dropped frames at the MAC level and unsuccessful retransmissions after retry limit result in a timeout at the TCP sender. As a result, TCP sender calls for false congestion control measure which worsens the condition. Added to this, MAC layer drops are also misinterpreted as a lost link and route rediscovery process is initiated which increases overhead in

the network. Therefore, it is clear that CW value has an impact on congestion at the transport layer. Thus, if proper value setting of a contention window (CW) at the MAC layer is not suitably considered, the resulting network performance is seriously degraded.

The proposed MAC model ACW analyzes the current state of the network as well as the node to predict CW and backoff window values. Rather than fully depending on transport layer factors to decide congestion, ACW brings in cross-layer parameters for considering collision lead congestion. ACW sets the CW based on the node's queue size and the node's residual energy level. The MAC parameter CW is then used to adjust the transport layer congestion window $CWND$. Proper setting of CW reduces MAC collisions and retransmissions. Consequently, the packet delivery ratio increases and TCP throughput also gets improved.

2. Related Work

To improve the network performance, various research studies have been carried out on IEEE 802.11 for determining the proper value for CW. To resolve the problem of BEB, the effort taken in [10] described a contention window adjustment scheme called MILD (Multiplicative Increase and Linear Decrease) in which the CW is multiplied by 1.5 on collision and decreased by 1 on successful transmission. It includes a CW copy mechanism to overhear the copy of CW of successful transmission. It prevents quick escalation to minimum CW after successful transmission, thus improving the performance. Authors of [11] contributed Exponential Increase and Exponential Decrease (EIED) that improved the performance when more competing nodes exist to access medium in a wireless region. EIED doubled the contention window after the collision and halved after a successful transmission. An extension of MILD proposed in [12] is Linear/Multiplicative Increase and Linear Decrease (LMILD). The sender increases CW multiplicatively in account of a collision. Any node overhearing collision sets CW linearly and successful nodes decrease CW linearly.

In [13], the authors predicted the amount of outstanding data in the network with the help of contention delay experienced by the packets and the throughput achieved by the connections. Along the traveling path, every packet includes contention delay per hop and the receiving end estimates the average contention delay per hop. The variation in the successive RTT found by the receiver is then advertised to the sender to adjust the sending rate. They studied TCP intraflow instability in detail and the proposed method dynamically adjusts the TCP transmission rate to avoid excessive build-up packets in the queue. The research in [14] discusses the cross-layer contention control method to alleviate CWND overshooting problem. It measured the contention RTT through forwarding data frame and the backward ACK at the medium access layer and considered the variance in contention RTT per hop (VCRH) as the contention status.

Upon reception of new ACK, if VCRH exceeds the threshold value, congestion window is reduced by 1 MSS (Maximum Segment Size). If RTO expires, CWND is set to $2 * MSS$. Along with other parameters, VCRH and timeout are quantified to congestion window adaptation.

Fu et al. [15] authored two mechanisms to improve the TCP to react to link overload: link RED (LRED) and adaptive pacing. Similar to random early detection for wired network, LRED drops frames based on a number of average MAC retransmission count. With increasing retry attempt, the packet drop probability increases because this situation indicates congestion. The mechanism adaptive pacing adds additional sender's MAC backoff time in addition to the current deferral time.

Sarairoh et al. [16] employed an approach to improve IEEE 802.11 quality of service by fine-tuning the contention window dynamically. ToS (type of service) field of IP packet is used to classify high and low priority traffic. Collision Rate (CR) and Collision Rate Variation (CRV) are the parameters used for CW setting. High priority traffic is increased for smaller values of CR and CVR. In highly loaded network

traffic, CW of the high priority traffic gets increased to reduce the delay. Differentiated services are prompted to access the medium with less collision rate both at light load and highly loaded network conditions. Knag et al. [17] examined estimation based backoff algorithm that estimates the number of idle slots during the backoff period. When there is a dramatic change in a number of active contending nodes or amount of traffic, the contention window is appropriately chosen.

Al-Hubaishi, Mohammed, et al. [18] suggested Enhanced Binary Exponential Backoff algorithm (E-BEB) that sets CW based on backoff time threshold value. The backoff threshold value is adaptively adjusted based on CW. It increases CW upon unsuccessful transmission and decreases to zero upon successful transmission. Authors [19] took into account the network history to optimize CW size. They implemented a channel state vector to store the network levels and based on the network level, CW is set.

Qin Yu et al. [20] projected the average idle interval time of an active node to dynamically adjust its contention window. They suggested this fair index to sense the network condition and less sensitive to the frame length and number of competing nodes. In [21], the authors proposed a dynamic control backoff time algorithm to handle nonsaturated traffic load of the network. They described the backoff algorithm with CW-Threshold as a reference value. When current CW is less than or equal to CW_{Th} , the CW is incremented exponentially for unsuccessful transmission and linearly decremented by 1 for successful transmission. When the network load goes high, then CW is incremented exponentially and added with two for transmission failure and linearly decremented by two. Gaurav et al. [22] presented a cross-layer mechanism for TCP performance improvement. They focused on link level contention, route failure, and congestion estimation to infer TCP sender about the network congestion.

Bhalerao et al. [23] designed and implemented CoCoA (Constrained Application Protocol Congestion Control Advanced) 4-state strong, an adaptation of 4 state estimator for variable backoff to improve the throughput. The proposed work has clearly distinguished between wireless losses and congestion losses. Rahman et al. [24] proposed a MAC strategy to mitigate the congestion issue in IoT enabled Intravehicular networks.

The research studies discussed above reveal that the value set for the contention window at the MAC layer of IEEE 802.11 has a significant improvement in the performance of wireless local area networks. To resolve the fairness issue among the competing nodes, the CW value is set based on the network condition. To avoid collisions and the packet drop rate, the contending stations must have contention slots such that all stations have fair access right to the medium. When collisions are reduced, the amount of legitimate traffic increases leading to less misprediction of a link failure at the network layer and the timeout at the TCP sender.

The proposed work emphasizes adjusting contention window size adaptive to the prevalent condition of the network and the node itself; thus, excessive waiting backoff slots are reduced.

3. The Objective of the Proposed Work

The shared and distributed access to the medium in the wireless network reasons the MAC layer collisions when the network is highly loaded. The existing backoff algorithms reduce/increase the contention window using decrement/increment factor when the medium is sensed to be busy. The MAC layer collision resulting from unfair CW triggers retransmissions and amplifies packet drops at MAC layer. More packet drops at lower layer thus induce a timeout for TCP segment and TCP sender initiates the congestion control algorithm unnecessarily. Hence, it is obvious that proper setting of CW value impacts TCP's improved performance.

4. Proposed Work Description

The proposed ACW considers the queue occupancy level and the residual energy at the node to adjust the contention window. CW is set smaller when queue occupancy level is more and node's energy is less to allow the node to access the medium immediately to avoid packet drop at MAC level. Backoff time estimation uses link scheduling among the neighboring nodes. The average transmission rate of node's neighbor's plays vital role in determining backoff time for transmitting and receiving packets in the contention region.

5. Active Queue Parameter (AQP)

The link layer queue stores packets to be transmitted to the next immediate hop node. When the queue becomes full, the packets get dropped. The packets stored in the queue that is reaching its full occupancy level should be transmitted quickly to alleviate packet drop. To provide significance to the queue with more packets, the CW considers the available queue length at the MAC and network layer of the node. Let AQS be the available queue size i.e., remaining queue capacity at the node, let TQS be the total queue size, and let $AQP_{avg(n-1)}$ be the AQP till (n-1) transmissions and a_1, a_2 are the random values between [0, 1]; then

$$\text{Active Queue Param (AQP)} = \frac{AQS}{TQS} \quad (2)$$

The AQP uses exponential weighted moving average method to estimate the value.

$$AQP(n+1) = a_1 * \left(\frac{AQS}{TQS} \right) + a_2 * (AQP_{avg(n-1)}) \quad (3)$$

For every transmission, AQP is estimated at the transmitting node. The estimated current buffer availability indicates whether queue reaches its full capacity soon or it has sufficient room for more packets. The value of AQP is reaching 1, meaning that the queue becomes full; the packets should be allowed to contend for the medium with a minimum value for CW. Otherwise, the packets will be dropped triggering retransmissions and timeouts.

$$\begin{aligned} & \text{If}(\text{energy} > \text{Energy}_{TH}) \\ & \{ \\ & \text{NEP} = w_1 * 1 + w_2 * \left(\frac{\text{Converted_energy_wrt_OE}}{OE} \right) \\ & \} \\ & \text{Else}(\text{energy} \leq \text{Min_Threshold}) \\ & \{ \\ & \text{NEP} = w_1 * \left(\frac{AE}{\text{Min_Threshold}} \right) \\ & \} \end{aligned}$$

Box 1

TABLE I: NEP values.

Available Energy (Joules)	NEP
95	0.98
50	0.8
15	0.51

6. Node Energy Parameter (NEP)

The energy level of the node is taken into account for adjusting the contention window of the node. When the node's energy level is getting reduced, the node is given priority to transmit the packet. A threshold energy value is used to estimate the NEP. When the node's energy level reduces beyond the threshold value, it is given high priority to access the medium for data transmission with CW set to minimum value. Let OE be the optimal energy, let IE be the initial energy at a node, and let AE be the available energy at a node; then NEP is estimated as in Box 1.

The *Min_Threshold* value considered for minimum contention window is 20 and the optimal energy (OE) considered for the optimal contention window is 80. NEP and *Converted_Energy_wrt_OE* are as follows:

$$\begin{aligned} & NEP \\ & = w_1 \left(\frac{\text{Current_energy_after_Th}}{\text{Min_Threshold}} \right) \end{aligned} \quad (4)$$

$$\begin{aligned} & + w_2 \left(\frac{\text{Converted_Energy_wrt_OE}}{OE} \right) \\ & \text{Converted_Energy_wrt_OE} \end{aligned}$$

$$= \left(\frac{CE}{IE} \right) * (IE - \text{Min_Threshold}) \quad (5)$$

The value obtained from $(\text{Current_energy_after_Th}/\text{Min_Threshold})$ defines the node's energy parameter when the energy level goes below the threshold energy value. The value obtained from $(\text{Converted_Energy_wrt_OE}/OE)$ defines the node's energy parameter when the energy level is optimal.

Table 1 tabulates the NEP values for different residual energy levels at a node. Table 1 clearly reveals that NEP value is low for low energy at a node and thus the node is allowed

to access the medium with minimum CW value if it has the data packets to transmit.

Let α_1, α_2 be the weight factors. Now, the contention window is dynamically adjusted as follows:

$$CW_{\text{param}} = \alpha_1 \text{NEF} + \alpha_2 \text{AQP} \quad (6)$$

Let CW_{max} be the maximum CW for the standard; then

$$CW = CW_{\text{max}} * CW_{\text{param}} \quad (7)$$

The nodes with minimum energy and maximum buffer occupancy are allowed to contend for the medium quickly with a minimum value for CW. The minimum CW should allow the nodes to transmit the packets immediately by capturing the channel.

7. Slot Time

Total time is split into small time frames during which the nodes can sense for channel state and transmit the data. The proposed method implements link scheduling to assign links to the active transmissions. The total slot duration is shared among the neighbors intact and the node is given by

$$\text{Slot time} = \frac{\text{slot duration}}{\text{active transmissions}} \quad (8)$$

Active transmission refers to node's current data transmissions. To identify the slot time for a neighbor node, Average Packet Transmission Rate (APTR) over last unit time and the Average Packet Transmission Rate over a certain time are considered

$$\begin{aligned} \text{APTR} = & \beta_1. \left(\frac{\text{No_of_Pack_transmitted}}{\text{unit_time}} \right) \\ & + \beta_2. \left(\frac{\text{No_of_Pack_transmitted}}{\text{Observation_time}} \right) \end{aligned} \quad (9)$$

Both $(\text{No_of_Pack_transmitted}/\text{unit_time})$ and $(\text{No_of_Pack_transmitted}/\text{Observation_time})$ denote rate of data transmission of the nodes. For the neighbors (n_1, n_2, \dots, n_i) of node node n , the APTR is calculated as APTR₁, APTR₂, APTR₃...APTR_i. Depending on the APTR value, the node's slot time is estimated. Let sum_APTR be the total of APTR of all active nodes, let Neighbor_share_n be the transmission rate of node_n with respect to all neighbors, and let Slottime_n be the slot time for node_n, to transmit the data; then backoff time for node_n is given by

$$\begin{aligned} \text{sum_APTR} &= \text{APTR}_1 + \text{APTR}_2 + \text{APTR}_3 + \dots \\ &+ \text{APTR}_n \\ \text{Neighbor_share}_n &= \frac{\text{APTR}_i}{\text{sum_APTR}} \quad (10) \\ \text{Slottime}_n &= \text{Neighbor_share}_n * \text{slot_duration} \\ \text{BT}_n &= \text{CW} * \text{slottime} \end{aligned}$$

BT_n denotes the time the node has to wait for capturing the medium for transmission. The nodes with more data transmission rate will get a fairer share of the wireless medium than nodes with less data transmission.

8. TCP Congestion Window

To avoid excess data transmission rate into the network, the TCP CWND is adjusted to determine the number of packets that could be in the network. Instead of choosing a random value for MAC backoff window, the current network parameter based contention window setting reduces the collisions and thus the MAC retransmissions. The backoff time decides the contention severity level of the node and accesses the medium which in turn is used to regulate the TCP transmission rate into the network. Thus, TCP congestion window is set based on the cross-layer MAC parameter as

$$\text{CWND} = \text{CWND}_{\text{max}} * \text{CW}_{\text{param}} \quad (11)$$

9. Performance Parameters

9.1. Throughput. It is defined as the amount of data packets received by the receiver in a unit of time. Generally it is measured in bits per second.

$$\text{Throughput} = \frac{\sum \text{Number_of_Packets}_{\text{received}}}{\sum \text{Total_time}} \quad (12)$$

9.2. Control Overhead. Control overhead is defined as the total number of control packets generated to transmit the data packets.

9.3. Normalized Overhead. Normalized overhead is the ratio of the amount of control packet sent to the amount of data packets received.

$$\begin{aligned} \text{Normalized_Overhead} \\ = & \frac{\sum \text{Number_of_Control_Overhead}}{\sum \text{Number_of_received_Packets}} \end{aligned} \quad (13)$$

9.4. Average Energy Consumption. Average energy consumption is defined as the energy consumed by all the nodes in the network.

10. Results and Discussions

The proposed method ACW is an improvement of our previous work CC-BADWA. The results of ACW is compared with CC-BADWA and RACC (Receiver Assisted Congestion Control) congestion control methods. The results acquired show that ACW has bettered the results of other two existing methods. The analytical implementation was carried out using Network Simulator, 2. The proposed model has been analyzed in 1000 x 1000 topological area with 50 nodes moving with random waypoint model. The MAC standard IEEE 802.11 is used to access the medium.

Figure 4 illustrates that throughput of the proposed ACW outperforms other throughput models in the wireless environment. The proposed method evaluates CW over the network parameters and dynamically adjusts CWND not only based on segment loss but considering the contention

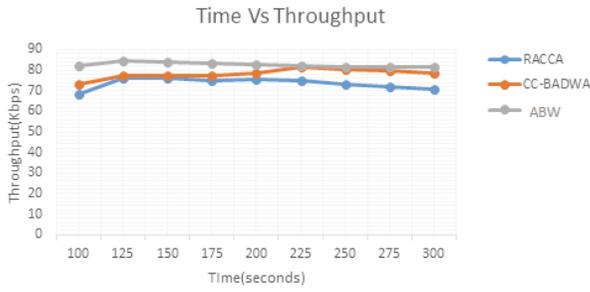


FIGURE 4: Comparison of throughput: RACC, CC-BADWA, and ACW.

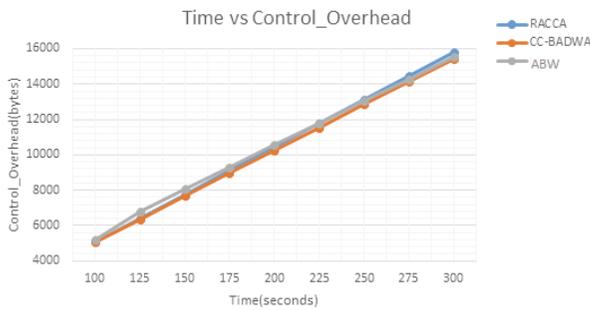


FIGURE 5: Comparison of control overhead: RACC, CC-BADWA, and ACW.

status in the network, thus gaining improvement in TCP throughput. The collision probability is reduced because of the dynamic setting of CW value and avoiding false trigger of TCP congestion algorithm resulting in less number of retransmissions, whereas in conventional TCP, the contention losses are treated as congestion when there is no congestion. ACW uses optimal contention window size based on the network conditions leading to a fair share of the channel access to all contending nodes. The value of CWND depends on the value of CW and hence the outstanding packets in the network are also controlled by the TCP sender. Eventually, the sending rate adaptively increases or decreases based on contention level in the network. So the PDR and throughput in ACW are improved.

Figures 5 and 6 compare the control overhead and normalized overhead of the compared mechanism. The control overheads are almost the same in all three methods, whereas the normalized overhead is reduced in ACW because the rate of collisions is reduced and hence the packet drop is reduced. The backoff time is chosen depending on the neighbor node's transmission rate and the slot time. The node with more data will get longer slot time than the rest of the nodes. This adaptive backoff time reduces the random backoff time used in BEB and thus the number of RTS/CTS control messages is reduced. So the control packets exchanged for the retransmitted packets also reduced leading to less normalized overhead than the other two methods.

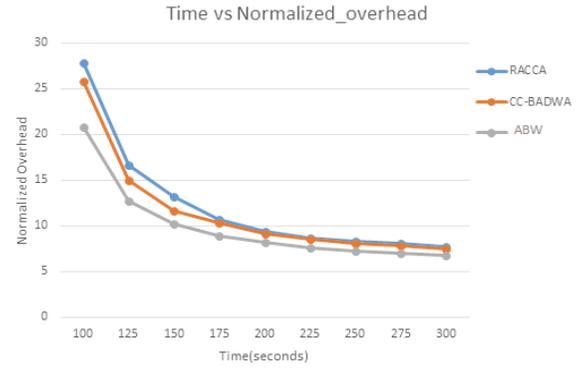


FIGURE 6: Comparison of normalized overhead: RACC, CC-BADWA, and ACW.

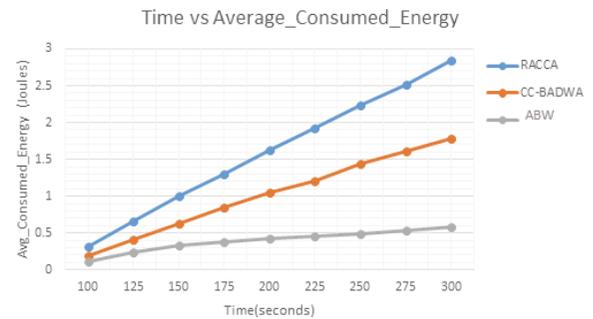


FIGURE 7: Comparison of Average_Consumed_Energy: RACC, CC-BADWA, and ACW.

Ultimately, the successful transmission rate increases reducing the retransmissions.

As shown in Figure 7, ACW reduces the average energy consumption since the dynamic setting of the backoff time is adaptive to current network condition. The Node Energy Parameter (NEP) estimated for contention window setting chooses the threshold value such that the node with low energy gets earlier access to the medium, thus avoiding packet losses and redundant transmissions. Also, the adaptive CW sets the value in line with network parameters not like BEB where CW is set to CWmin and $2 * CW$ after successful and unsuccessful transmissions, respectively. The conventional BEB causes starvation among the nodes that have lost the race and double CW every time. After the maximum retry count, the packet is dropped, timeouts at TCP trigger congestion control, and more energy consumption results. ACW's contention parameter impacts congestion window which interprets congestion accordingly.

11. Conclusion

The proposed Adaptive Backoff Window (ACW) for MAC model is to reduce MAC layer collisions by adaptively setting the value for a backoff window for every successful or unsuccessful MAC frame transmissions. The nodes willing to transmit the frame after waiting for a random backoff time

attempt to capture the medium to transmit the frames. When the medium is sensed busy, the random contention window size is chosen by the current network and node parameters. CW in ACW is set upon node's energy parameter, node's queue parameter, the active nodes in the region, and RP value. The dynamic value of backoff and CW decreases the MAC layer contentions and collisions which could otherwise have caused packet drops at the link layer and timeouts at the transport layer. Thus, the timeouts and unnecessary invocations of TCP congestion control algorithms are reduced. Thus, reductions in MAC retransmissions due to dynamic backoff improve the TCP performance. The extensive simulations carried out prove that ACW has reduced the packet drop ratio and end-to-end delay. The results clearly indicate that the number of successful transmissions is increased compared to the existing methods. Eventually, the MAC packet drops and timeouts are reduced. Since the adaptive CW indicates the present network condition, it is used to set the TCP CWND which decides the amount of data packets that could be in flight. Both CW and CWND decide the appropriate number of packets that can be in the network pipe so that the drop ratio and timeouts are reduced and TCP performance is improved in terms of network throughput.

Additional Points

Scope for Future Enhancement. The proposed work considers the MAC layer contention status to predict the data sending rate at the sender node. Future work will further proceed with the feedback information from the physical layer since the PHY layer characteristics can promptly reveal the wireless medium status to predict real congestion in the network.

Conflicts of Interest

The authors declare that the grant, scholarship, and/or funding mentioned in Acknowledgments does not lead to any conflicts of interest. Additionally, the authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (NRF-2017R1C1B5017464).

References

- [1] A. Paul, A. Ahmad, M. M. Rathore, and S. Jabbar, "Smartbuddy: defining human behaviors using big data analytics in social internet of things," *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 68–74, 2016.
- [2] A. Paul, T. Victoire, and A. Jeyakumar, "Particle swarm approach for retiming in VLSI," in *Proceedings of the 46th International Midwest Symposium on Circuits and Systems*, pp. 27–30, Cairo, Egypt, 2003.
- [3] A. Paul, "Real-time power management for embedded M2M using intelligent learning methods," *ACM Transactions on Embedded Computing Systems*, pp. 1–22, 2014.
- [4] A. Paul, "Graph based M2M optimization in an IoT environment," in *Proceedings of the Research in Adaptive and Convergent Systems (RACS '13)*, 2013.
- [5] W. Liu, H. Jin, X. Wang, and M. Guizani, "A novel IEEE 802.11-based MAC protocol supporting cooperative communications," *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1480–1495, 2011.
- [6] K. Hong, S. K. Lee, K. Kim, and Y. H. Kim, "Channel condition based contention window adaptation in IEEE 802.11 WLANs," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 469–478, 2012.
- [7] M. Barry, A. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proceedings of the INFOCOM*, Anchorage, AK, USA, 2001.
- [8] Y.-H. Zhu, X.-Z. Tian, and J. Zheng, "Performance analysis of the binary exponential backoff algorithm for IEEE 802.11 based mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2011.
- [9] Q. Pang, S. C. Liew, J. Y. B. Lee, and V. C. M. Leung, "Performance evaluation of an adaptive backoff scheme for WLAN," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 867–879, 2004.
- [10] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a Media Access Protocol for Wireless LANs," in *Proceedings of the SIGCOMM'94*, 1994.
- [11] N. Song, B. Kwak, J. Song, and M. E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," in *Proceedings of the 57th IEEE Semiannual Spring VTC*, vol. 4, pp. 2775–2778, 2003.
- [12] A. Paul and S. Rho, "Probabilistic model for M2M in IoT networking and communication," *Telecommunication Systems*, vol. 62, no. 1, pp. 59–66, 2015.
- [13] E. Hamadani and V. Rakocevic, "TCP contention control: A cross-layer approach to improve TCP performance in multihop ad hoc networks," in *Proceedings of the 5th Int. Conf. Wired/Wireless Internet Commun.*, pp. 1–16, 2007.
- [14] X. M. Zhang, W. B. Zhu, N. N. Li, and D. K. Sung, "TCP congestion window adaptation through contention detection in Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4578–4588, 2010.
- [15] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP performance," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 209–221, 2005.
- [16] M. Saraireh, J. AL-Saraireh, and S. Saraireh, "A Novel Adaptive Contention Window Scheme for IEEE 802.11 MAC Protocol," *Trends in Applied Sciences Research*, vol. 9, no. 6, pp. 275–289, 2014.
- [17] S.-W. Kang, J.-R. Cha, and J.-H. Kim, "A novel estimation-based backoff algorithm in the IEEE 802.11 based wireless network," in *Proceedings of the Consumer Communications and Networking Conference (CCNC)*, 2010.
- [18] M. Al-Hubaishi, T. Alahdal, R. Alsaqour, A. Berqia, M. Abdelhaq, and O. Alsaqour, "Enhanced binary exponential backoff algorithm for fair channel access in the IEEE 802.11 medium access control protocol," *International Journal of Communication Systems*, vol. 27, no. 12, pp. 4166–4184, 2014.
- [19] A. Balador, A. Movaghar, S. Jabbehdari, and D. Kanellopoulos, "A novel contention window control scheme for IEEE 802.11 WLANs," *IETE Technical Review*, vol. 29, no. 3, pp. 202–212, 2012.

- [20] Q. Yu, Y. Zhuang, and L. Ma, "Dynamic contention window adjustment scheme for improving throughput and fairness in IEEE 802.11 wireless LANs," in *Proceedings of the Global Communications Conference (GLOBECOM)*, 2012.
- [21] H. Alkadeki, X. Wang, and M. Odetayo, "Improving Performance of IEEE 802.11 By a Dynamic Control Backoff Algorithm Under Unsaturated Traffic Loads," *International Journal of Wireless & Mobile Networks (IJWMN)*, vol. 7, no. 6, 2015.
- [22] G. Bhatia and V. Kumar, "CTCP: A Cross-Layer Information Based TCP for MANET," *International Journal of Ad Hoc, Sensor & Ubiquitous Computing*, vol. 5, no. 1, 2014.
- [23] R. Bhalerao, S. S. Subramanian, and J. Pasquale, "An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol," in *Proceedings of the Consumer Communications & Networking Conference (CCNC), 13th IEEE Annual. IEEE*, 2016.
- [24] M. A. Rahman, M. N. Kabir, S. Azad, and J. Ali, "On mitigating hop-to-hop congestion problem in IoT enabled Intra-Vehicular communication," in *Proceedings of the Software Engineering and Computer Systems (ICSECS), 4th International Conference on. IEEE*, 2015.

