Wireless Media Access Depending on Packet Size Distribution over Error-prone Channels

Farah Tanjeem, Md. Yusuf Sarwar Uddin and A. K. M. Ashikur Rahman
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
Email: farahtanjeem@yahoo.com, yusufsarwar@cse.buet.ac.bd, ashikur@cse.buet.ac.bd

***Abstract*—Ad Hoc Network is a decentralized type of network where wireless devices are allowed to discover each other and communicate in peer to peer fashion without involving central access points. In most ad hoc networks, nodes compete for access to shared wireless medium, often resulting in collision (interference). IEEE 802.11, a well-known standard, uses medium access control (MAC) protocol to support delivery of radio data packets for both ad hoc networks and infrastructure based network. But designing a Medium Access Control(MAC) protocol for ad hoc wireless networks is challenging, particularly when the protocol needs to achieve optimal performance both in terms of throughput and efficiency to deliver a packet. Error-prone channel has a significant impact on unsuccessful transmission probability which is often ignored by previous researches. Standard DCF (Distributed Coordination Function) operation of IEEE 802.11 enacted by binary exponential back-off (BEB) algorithm cannot differentiate collision from corruption and therefore sets forth a (time) separation between multiple nodes accessing the channel by (appropriately) adjusting contention window (CW) upon a failure. This leads to increased delay in error-prone network when nodes are not contending at all. Since packet corruption depends on bit error rate (BER) and length of packets, packet size can have significant impact on the throughput in error-prone environment. In this paper, we analyze effect of packet size in determining optimal CW to improve throughput and efficiency for error-prone networks. We propose a dynamic learning based scheme to adaptively select CW sub-range instead of whole selection range for different packet distribution. To validate our scheme extensive simulations have been done and simulation results show significant improvement in E2E delay performance.**

I. Introduction

In this paper we propose a contention window adjustment technique based on IEEE 802.11's DCF (Distributed Coordination Function) for error-prone wireless channels. IEEE 802.11 is one of the most popular standards for wireless media access that uses carrier sense multiple access with collision avoidance (CSMA/CA) with binary exponential backoff (BEB). Before starting a transmission, stations listen to channels to see whether there is any active transmission going on. If so, they retreat. When the station finds an idle channel, it attempts to transmit the packets. If the transmission results in a collision (that means someone else was also transmitting colluding all bits), the station initiates a recovery mechanism by entering a wait period determined by the BEB algorithm. Under BEB, the station waits a *random* number of time slots before it can make its next transmission attempt. This random number is sampled uniformly from a range, called the contention window (CW), the size of which doubles with every transmit attempt that was deferred, until a maximum size is reached. That means, after k consecutive collisions, the station picks a random number between 0 to $2^k - 1$ (where 2^k is the current size of the contention window) and remains idle for that number of slots, and attempt transmitting packets when the time elapsed.

978-1-4799-8126-7/15/$31.00

Generally the DCF operation cannot differentiate collision from corruption, which assumes packet losses are due to collision and therefore takes only collision resolution mechanism. This technique of randomization and scaling the contention window size for collision avoidance is used to reduce collisions by preventing multiple nodes from sending packets in the channel at the same time. For ideal channel selection of optimal CW impacts more because choosing a larger one leads to large backoff interval and larger overhead before each packet transmission. But a smaller CW will cause more collisions.

In a noisy environment, this measure indeed hurts by delaying unnecessarily some nodes when those are actually not contending at all. So in a non-ideal channel for non-contending nodes impact of optimal CW is less. But at the same time selection of larger CW induces longer delays. Since packet corruption depends on bit error rate (BER and length of packets, packet size can have significant impact on the performance in error-prone environment as considered by authors in paper ([1], [2]). In ideal channel larger packets are preferred to reduce protocol overhead and RTS/CTS handshaking scheme used based on packet length. But for error-prone channel, larger packets have more probability to be in error than the smaller ones. It is shown in [2] that under an error-prone channel environment, optimal packet size can have more significant impact on energy efficiency than optimal contention window, and combining both optimal contention window and optimal packet size can achieve the maximum performance.

We conjecture that the distribution of packet sizes has significant impact on channel performance and can lead to important clue for adjusting contention window that can result in improved performance. We analyze effect of two parameters, namely packet size and contention window on error-prone channel and design a dynamic CW setting scheme based on packet size distribution. To this end, we devise a method of adaptively tuning CW with the help of packet size distribution of the network under different error models and error rates.

II. Related Work

MAC Protocols can be modified based on backoff mechanism which in turn uses contention window to modify the behavior of the protocol. Many potential research activities focus of modification of CW using different approach. There are some approaches like MILD (Multiplicative Increase Linear Decrease) [5], DIDD (Double Increase double decrease) [6], which are mainly based on IEEE 802.11 DCF. But most of them are designed for single hop network and very few of them are tested in ad hoc network. Some works [4], [13], [10], [12], [11] modify CW considering history of packet loss. Based on channel state vector, these protocols select a CW sub-range instead of just doubling the upper bound of CW. Extra memory space and additional computations required to be done for keeping and maintaining channel state vector. There are some local information based CW modification scheme [7], [8], [9] where some computations are done at the end of every interval to choose which CW to use for next interval. These algorithms mainly focus on fairness issue and gives better performance for unloaded networks.

There are some research works, which concentrate on performance and energy efficiency issues of wireless error prone environment. Works in paper [1], [2] developed an analytical model with Markov chain model to identify effects of packet size and CW in error prone channel. Some papers [3] provide saturation throughput analysis to show that channel errors have significant impact on system performance. Most of the papers limit their work on the throughput and delay under saturated traffic condition.

III. Proposed Media Access Protocol

In this section, we discuss our proposed contention window setting algorithm based on packet size distribution for wireless ad hoc error-prone networks.

A. *Basic Idea for Adjustment of CW Based on Packet Sizes*

To devise a modified MAC scheme for error-prone environment, we propose a dynamic contention window range control scheme. We make a selection of appropriate CW dependent on packet sizes.

As we know, delivery delay of packets in 802.11 DCF depends on two things: the size of the contention window and number of retransmissions. Retransmission depends on collision probability, which in turn depends on how many nodes are contending for the channel. It is also known that collision probability increases with increasing number of nodes. In our approach, we partition packets into a set of groups depending on their sizes and assigned them different sub-range from the entire contention window range, namely 0–CW.

To partition packets into groups based on their sizes, we introduce the notion of *labels* and *bins*. We divide packets flowing through the network into a set of bins based on their respective sizes each with their own start and end boundary. Packets having a certain size, i.e., higher than the low boundary of a certain bin but smaller than the high boundary of the same bin, belong to that particular bin. $label$ is used to classify packets based on which bin they reside. In our proposed scheme, we consider 4 bins for simplicity and we mark the boundary sizes by LOW, MEDIUM and HIGH respectively (from lower size packets to higher size packets).

Now, to create different contention domains for the packets belonging to different bins, we uniformly divide total CW range (0–CW) into several sub-ranges and each sub-range is assigned to each bins. That is, packets residing in a certain bin choose their next CW sampled from its respective CW sub-range. Since lower sized packets (packets from bin1) are usually control packets and should be tried immediately, they are assigned lower CW sub-range. Higher sized packets are assigned progressively higher CW sub-ranges. Every time when retransmission happens and CW gets doubled (range becomes twice as much as its earlier value), the respective sub-ranges for respective bins hold.

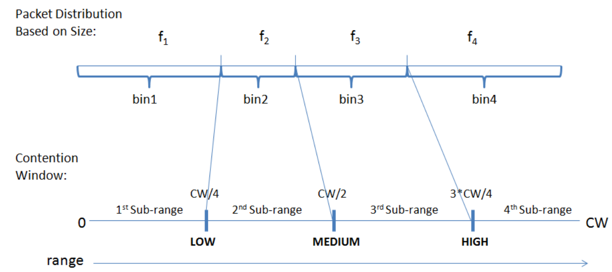


Fig. 1. Packet distribution in Bins for selection of CW sub-range

Now the question remains how these boundary sizes for binning and CW sub-ranges are determined. We used the following simple strategy:

Packets belonging to a certain bin take a CW subrange in proportion to the number of packets reside in that bin.

For example, if bin1 contains 30% packets generated in the network, it takes first 30% of the whole contention window, if bin2 contains 40% packets it takes the next 40% contention windows. Since total portions of packets across all bins would be 100%, CW sub-range would upto 1, thus taking up the whole range. Admittedly, packets in bin1 experiences less contention wait time than packets belonging in bin4, as they take sample from smaller contention window range. We cannot, however, put *all* packets into bin1 because in that bin1 packets would take entire CW range thus nullifying their very advantage of being in bin1. Now the question is what fraction of packets each bin should contain, or put it in a different way, how to determine the bin boundaries, so that the overall packet delivery improves, say packet delivery delay is reduced. In the following, we attempt to establish this based on some assumption for error-prone channels.

B. *Probabilistic Analysis of Binning*

We try to estimate expected delivery delay of packets as a function of fractions of packets that fall into different bins. Let f_1, f_2, f_3, and f_4 be the fractions of packets belonging to bin1, bin2, bin3 and bin4 respectively. While we analytically deduce the optimal values for the mentioned f's, we make the following assumptions:

- Collision probability remains constant across bins. That is, collision chances among packets remain unchanged as we do binning, and even in successive retransmission attempts.
- Probability of packet corruption due to bit error rate depends linearly on packet sizes.
- Packet of different sizes are equally likely.

- Packet delivery delay mainly depends of contention wait times and the number of retransmissions suffered by packets (due to either collision or corruption).

Unsuccessful transmission occurs when packets collide with each other (simultaneous access of medium, usually arbitrated by contention wait times) or unsatisfactory channel conditions corrupt the packet at the receiver even if the packet contends successfully. Let P_c denote packet corruption probability and P_{col} denote packet collision probability. While collision probabilities P_{col}'s remain the same across four bins, corruption probabilities P_c's vary across bins as different bins contain different sized packets. So let P_{c_1}, P_{c_2}, P_{c_3} and P_{c_4} denote the corruption probability of packets that reside in bin1, bin2, bin3 and bin4 respectively.

Let s_{max} denote the size of the largest packet in the network. Since packets are assumed to be equally likely and bin1 contains f_1 fractions of packets, packets smaller than $f_1 \times s_{max}$ belongs to bin1. Similarly, packets in bin2 are smaller than $(f_1 + f_2) \times s_{max}$. Since as per assumption corruption probability linearly depends of packet lengths, we can write the following:

$$P_{c_1} \propto f_1 \times s_{max}$$

The above equation can be rephrased as the following for some suitable constant γ:

$P_{c_1} = f_1 \times \gamma$

For other corruption probabilities, we can write in a similar way:

$P_{c_2} = (f_1 + f_2) \times \gamma$

$P_{c_3} = (f_1 + f_2 + f_3) \times \gamma$

$P_{c_4} = \gamma$

Retransmission probabilities at bin1, bin2, bin3 and bin4 are denoted as P_{Ret_1}, P_{Ret_2}, P_{Ret_3} and P_{Ret_4} respectively. Since retransmission happens when collision or corruption occurs and these two events are independent, the retransmission probability per bin can be obtained as follows:

$P_{Ret_i} = 1 - (1 - P_{c_i})(1 - P_{col})$ where i = 1,2,..4.

We assume that retransmission probability remains the same in successive attempts (which in practice does not hold, because CW gets doubled in the next attempt that in fact is supposed to reduce contention chance. We ignore this detail). In that, the number of retransmissions per packet can be modeled as geometric random variable. Using $X \sim geom(p)$ gives $E[X] = \frac{1}{p}$, we get the expected number of retransmissions for bin1 packets as:

$E[Ret_1] = \frac{1}{1-P_{Ret_1}}$

Putting values of P_{Ret_1} and P_{c1} in the above equation we can get below mentioned equation,

$$\begin{aligned} E[Ret_1] &= \frac{1}{(1 - P_{c_1})(1 - P_{col_1})} \\ E[Ret_1] &= \frac{1}{(1 - (f_1 \times \gamma))(1 - P_{col_1})} \end{aligned}$$

Each time a transmission or retransmission happens, packet residing a certain bin chooses a sample CW (randomly pick one from its respective CW sub-range mapped to its bin) and waits that amount of time to access the channel. The amount of wait time thus depends on the (high) boundary values of CW sub-ranges. For example, wait times for bin1 packets are proportional to $f_1 \times CWMax$, for bin2 are $(f_1 + f_2) \times CWMax$, and so on.

Let delivery delay of bin1, bin2, bin3 and bin4 packets be D_1, D_2, D_3 and D_4. If we put values of P_c's, we get below mentioned equations for expected delay for packets in each bin. For bin1, we get:

$$E[D_1] = \frac{f_1 \times CWMax}{(1 - f_1 \times \gamma)(1 - P_{col_1})} \tag{1}$$

Replacing all constants with C, we get:

$$\begin{aligned} E[D_1] &= \frac{f_1}{(1 - (f_1 \times \gamma))} \times C && (2) \\ E[D_2] &= \frac{(f_1 + f_2)}{(1 - ((f_1 + f_2) \times \gamma))} \times C && (3) \\ E[D_3] &= \frac{(f_1 + f_2 + f_3)}{(1 - ((f_1 + f_2 + f_3) \times \gamma))} \times C && (4) \\ E[D_4] &= \frac{1}{(1 - \gamma)} \times C && (5) \end{aligned}$$

Hence, the overall expected delay can be denoted as $E[D]$, which is the summation of delays in each bin weighted by their respective fractions.

$$\begin{aligned} E[D] = \; & \frac{f_1 f_1 C}{(1 - (f_1 * \gamma))} + \frac{(f_1 + f_2) f_2 C}{(1 - ((f_1 + f_2)\gamma))} + \\ & \frac{(f_1 + f_2 + f_3) f_3 C}{(1 - ((f_1 + f_2 + f_3)\gamma))} + \frac{f_4 C}{(1 - \gamma)} \end{aligned} \tag{6}$$

We want to minimize $E[D]$ for certain values of f's. Using an optimization solver (www.ampl.com), we obtain the assignments, $0.250, 0.249, 0.250, 0.249$ respectively. For simplicity, we consider 25% packets in each bin. That means, each bin should contain equal 25% of packets. Therefore, CW range is also equally partitioned to 4 sub-ranges, each of size $CW/4$. This is shown in Figure 2. Although we assumed that packets are of uniform sizes, the same results follow for skewed distribution also (as we see in the experiment section).

C. Modified MAC Protocol

Main steps of our proposed scheme are:

- Labeling packets into groups based on packet size distribution.
- Bins are determined based on percentile of packet size distribution, so that each bin contains 25% packets.
- Identifying boundary size of CW sub-range for each group.

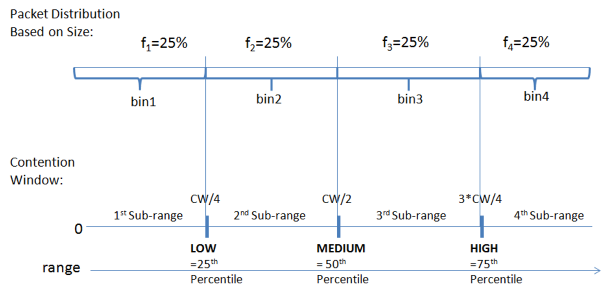


Fig. 2. Uniform Binning for selection of CW Sub-range

- CW adjustment for each packet belongs to a specific bin.

In order to incorporate the idea mentioned above, each node has been initially equipped with a window timer, which runs in background. To obtain current distribution of packet size, packets sensed by individual nodes have been considered here. After receiving a packet p with size s, node increments the frequency count for packet size s. Packet sizes and arrival frequency is required to adaptively set the packet labels based on network traffic. It also continues its learning process during the simulation and adjusts the labeling values dynamically from one interval to another. Based on data collected from learning phase, we have to adaptively set the values of packet $labels$ based on network traffic such that around 25% of packets fall into a specific bin for uniform binning. Total 4 bins have been considered in our proposed algorithm with 3 labels LOW, MEDIUM and HIGH. We have considered 25, 50 and 75 percentile values for labeling packets as LOW, MEDIUM and HIGH.

Let the total number of different packets be n, and X be a random variable denoting packet size. So to calculate f percentile, we need to find the packet size X where $f\%$ packets falls below or equal to X. At the end of every window timer, it wakes up and calculates f percentile for the next interval.

Using Cumulative distribution function (CDF) and linear interpolation, we calculate LOW, MEDIUM and HIGH label values. CDF, $P\{X \leq f\}$ measures the probability that a real valued random variable X with a given probability distribution will be found to have a value less than or equal to f. The following equations can be defined for calculating labels.

$$P\{X_1 \leq LOW\} = f_1 = 0.25$$

$$P\{LOW < X_2 \leq MEDIUM\} = f_2 = 0.25$$

$$P\{MEDIUM < X_3 \leq HIGH\} = f_3 = 0.25$$

$$P\{X_4 > HIGH\} = f_4 = 0.25$$

where f_1, f_2, f_3 and f_4 are the percentile values that fall into a specific bin.

Let us denote P_i be the probability that a packet size is less than or equal to s_i. Then mathematically,

$$P_i = \frac{(\sum_{j=1}^{i} fi)}{(\sum_{k=1}^{n} fk)}$$

Where, f_i is frequency count for packet size s_i

Here P_i is the cumulative distribution function (CDF). Using this CDF we have calculated values of LOW, MEDIUM and HIGH. Window Timer now calculates label values using linear interpolation. Calculation for LOW label is shown below:

$$LOW = \left\{\frac{(f_1 - P_{LG})*(S_{GL} - S_{LG})}{(P_{GL} - P_{LG})}\right\} + S_{LG}$$

Where,

P_{LG} =Greatest probability less than f_1

P_{GL} =Least probability greater than f_1

S_{LG} =Packet size at P_{LG}

S_{GL} =Packet size at P_{GL}

Based on packet labeling, we partition packets into 4 bins with almost $equal$ number of packets in each bin. LOW labeled packets pick from 0 to $\frac{CW}{4}$, MEDIUM labeled packets from $\frac{CW}{4}$ to $\frac{CW}{2}$, HIGH labeled packets from $\frac{CW}{2}$ to $\frac{3\times CW}{4}$ and packets greater than HIGH pick from $\frac{3\times CW}{4}$ to CW. After each failure, CW gets doubled and $CWMax$ remains same as per standard protocol. In our scheme, we have considered $CWMin$ as 127 for our initial learning phase.

IV. Experimental Results

We run simulation experiments using NS2. Network simulation scripts in NS2 are used to create the network scenarios and upon the completion of the simulation, trace files that capture events occurring in the network are produced. We use this captured information in performance study. We run our simulation in two different network topologies such as grid and random topology. In most cases, we consider a static network. For each environment, we run the simulation for 5-10 times with different seed values and we have taken the average of 10 results with the same setting.

We show the basic simulation parameters in Table I.

TABLE I. Simulation Settings

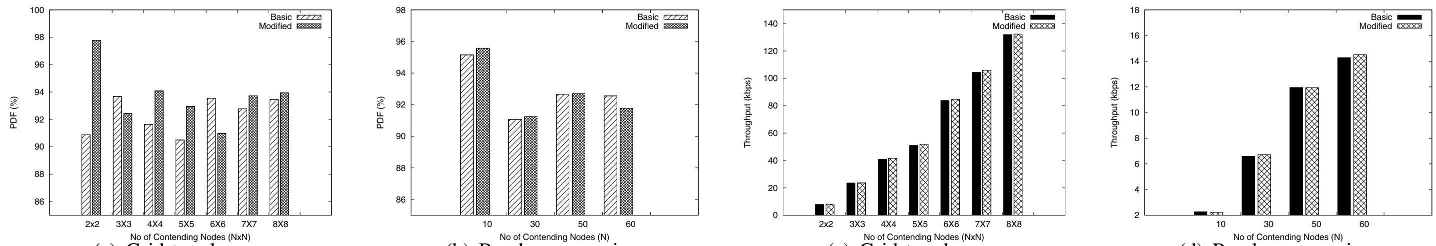
Parameter	Value	Parameter	Value
Topology Setting		General Parameters	
Simulation scenario	Random, Grid	Simulation time	$300s$
Grid Size (N * N)	$2-8$	Window timer	$30s$
No of Nodes(N)	10 60	CWMin, CMMax	$127, 1023$
Max(X),Max(Y)	$500, 500$	RTSThershold	3000
Packet Labeling		Error Model	2 State Markov
Low	0 to< 25	Error rate	$(0, 0.9)$
Medium	25% to < 50	TX range	$250m$
High	$50\% to < 75\%$	CS range	$550m$
Remaining	$Above > 75\%$	Traffic	CBR (Rate 0.1,2)

A. Impact of Error Prone channel on IEEE 802.11 MAC

We compare the *Average E2E delay*, *Packet Delivery Fraction* and *Throughput* parameters evaluated at error models with different error rates for original IEEE 802.11 MAC to find the effect of error prone channel and after that compare the result with modified one.

Average E2E Delay: End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination.Average E2E delay is the summation of the delay of all packets divided by the number of generated packets.

Packet Delivery Fraction(PDF): The delivery ratio is the number of received packets divided to the number of generated packets.



(a) Grid topology (b) Random scenario (c) Grid topology (d) Random scenario

Fig. 3. Packet delivery ratio (PDF) and throughput variation for grid and random scenarios.

Throughput: Throughput can be defined as time average of the number of bits that can be transmitted by each node of the network to its destination.

We use varied error rates like 0.25, 0.5, 0.75, 0.9 and 1.0 in bad state of our simulation to observe the impact of error prone channel and error rates on wireless media. In most of the cases higher error rate causes more degradation in average E2E delay performance as depicted in Figure 4(a) and Figure 4(b) for random and grid scenario. Higher error causes more retransmissions, which will in turn causes more delay. We can also observe from figures that, E2E delay variation is high for higher error rates like 0.75,1.0.

From the above simulations, we can find that error prone channel has a significant impact in performance metrics of wireless ad hoc network. Impact of error prone channel is more on average E2E delay performance variation rather than PDF and throughput as depicted from Figure 4(c) and Figure 4(d).

B. Performance Results of the Proposed Scheme

We have done extensive simulations to observe the performance of modified MAC protocol in comparison with original IEEE 802.11 MAC protocol. In our simulations we vary number of nodes, CBR rate, network topology to observe the behavior of some performance metrics like *E2E delay*, *Throughput* and *Packet Delivery Fraction(PDF)*.

We have done our simulations considering uniform packet distribution for random packet generation. To add another variant, beta packet size distribution has also been used in some experiments. We generate packets with sizes ranged from a Minimum and Maximum value using Random variable class available in OTcl.

From Figure 5(a) and Figure 5(b), we observe that in almost all cases average E2E delay decreases in modified scheme in comparison with original one for a range of contending nodes in random and grid topology with different network loads and upto 47% improvement observed in average E2E delay. PDF remains almost same as the original one, which is depicted in Figure 3(a) and Figure 3(b) for both grid and random scenario. We are getting almost 0-4% variation in PDF value and almost same throughput between modified and original one as depicted in Figure 3(c) and Figure 3(d). Simulation has been done for both grid and random scenario with varied network load.

To see the effect of skewed packet generation in our scheme, we use Beta(2, 4) distribution to observe the impact of packet distribution scheme in our modified algorithm. We can observe experimental results from Figure 5(c) for beta packet distribution, where minimum and maximum packet size ranging from 128-1024 Bytes. Previously, we have observed E2E delay variation with uniform packet distribution. From both results, we can deduce that our modified algorithm performs better for both type of packet distribution.

C. Variations of the Proposed Protocol

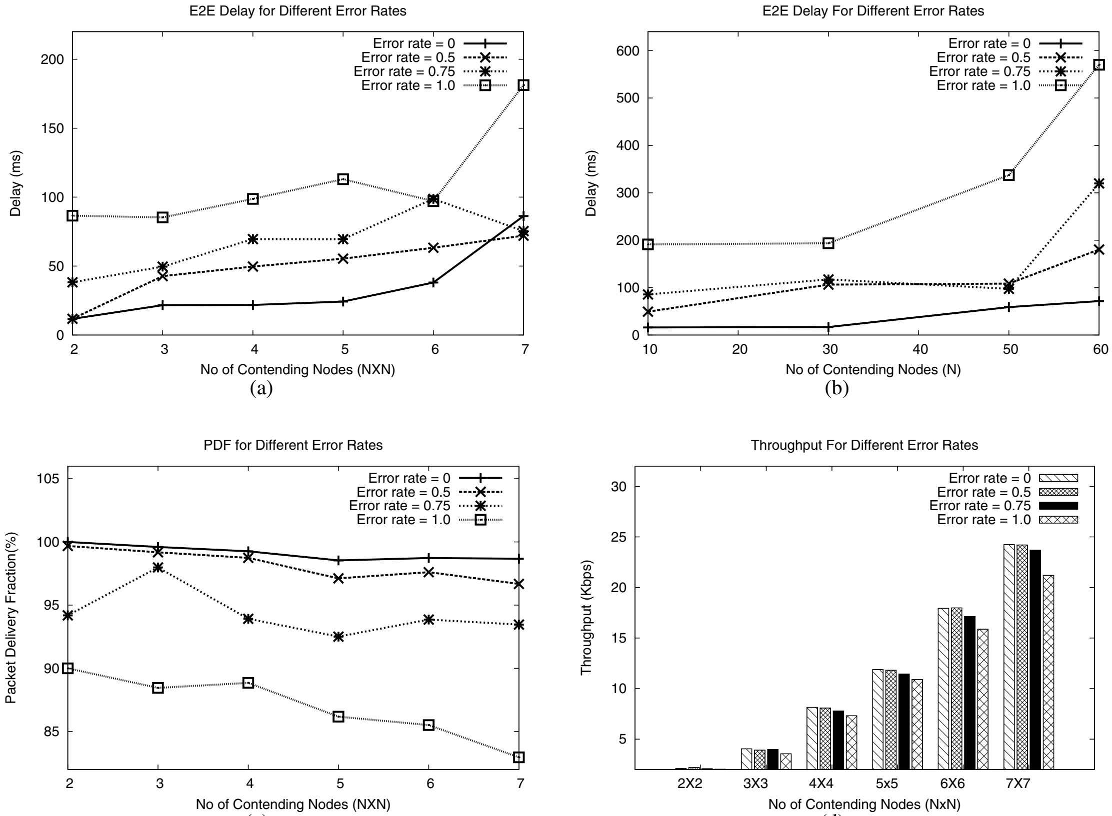
And at last, we make some alternative approaches to our modified scheme using 6X6 grid topology in a highly loaded network to observe the performance impact on average E2E delay. From previous simulations, we can find that impact of error-prone channel found mostly in average E2E delay parameter. So we simulate further variants of modified scheme only with E2E delay parameter.

For this variant, we assign packet labels from random selection over 4 classes/bins instead of packet size based selection. After that, we set CW as per our proposed scheme. Figure 6(a) depicts the scenario and result shows that random class assignment scheme performs worse than our proposed packet distribution.

In another experiment, we assign arbitrary bin boundary instead of equal 25%. Figure 6(b) depicts the scenario of varying distribution of packet sizes (here 10%, 30%, 20%, 40% considered) in each bin and from the result we can find that our uniform binning method performs better. We also try reversing the CW sub-range assignment, that is, assigning smaller CW sub-range to larger packets and so on. The results are shown in Figure 6(c). Our modified scheme performs betters than this variant.

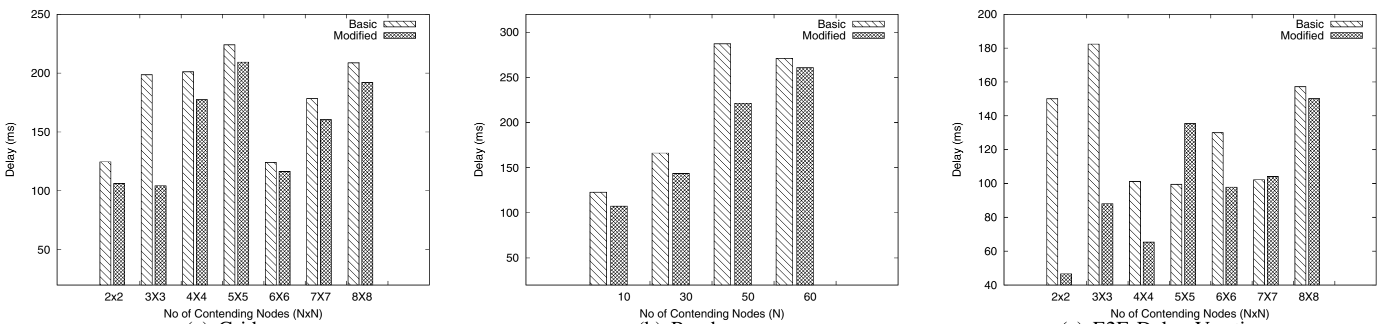
In our modified algorithm, we use a simple strategy: *Packets belonging to a certain bin take a CW sub-range in proportion to the number of packets reside in that bin*. Using this concept, we partition packets into 4/8 bins based on packet labeling and assign a separate contention window sub-range for packets belonging to a certain bin. To follow this, first 25% packets belong to bin1, next 25% to bin2 and so on for 4 bins and first 12.5% packets belong to bin1, next 12.5% to bin2 and so on for 8 bins. In this variant, we choose different number of bins with non-equal number of packets in each bin associated with non-equal sub-range. The impact of this choice on average E2E delay has been depicted in Figure.6(d) in comparison with choice of 3, 5 and 6 bins. And our proposed one with 4/8 bins outperform other ones.

E2E delay performance varies based on selection of window timer interval. The length of this interval must not be too short to avoid possible oscillation and not be too long to



(a) (b) (c) (d)

Fig. 4. Variations of the Error Rate in IEEE 802.11 MAC (a) E2E Delay(Grid) (b) E2E Delay(Random), (c) PDF(Grid), (d) Throughput(Grid)



(a) Grid (b) Random (c) E2E Delay Varation

Fig. 5. (a, b) End to end packet delivery delay for grid and random topology, (c) E2E Delay Variation for packet size following Beta(2, 4) distribution

decrease the reactivity of the algorithm. From below mentioned Figure 7, we can find that different window timer value causes different E2E delay with all other settings remain the same. Here, WT = 30s gives optimum result for 300s simulation time for both beta and uniform packet distribution. We have to set Window timer with such a value that a considerable number of iterations can be done within simulation time.

V. Conclusion

In this paper, we focus on performance issues of wireless error-prone network. We consider packet error as an important source of performance variation along with collision in error-prone environment. Contrary to IEEE 802.11 BEB algorithm, our protocol dynamically selects contention window sub-range instead of whole selection range based on packet size distribution. We formulate a theoretical study of our approach and produce a formal verification of the underlying uniform binning concept in the architecture. Besides developing a modified protocol, we also simulate the protocol using NS2 simulator to make closer observations regarding various performance issues and analyze the simulation output against the original IEEE 802.11 MAC DCF protocol. The performance of the scheme in terms of average E2E delay, packet delivery fraction (PDF) and throughput have been analyzed and our scheme performs better in most of the cases. Evaluated results showed that our scheme achieves less retransmissions, which causes reduced average E2E delay. At the end of the simulation, we have introduced some variants of our proposed scheme. We further intend to experiment more variants of our proposed approach and tuning the used parameters for more better result.

References

[1] Yin, J., Wang, X., Agrawal, D.P., “Optimal Packet Size in Error-Prone Channel for IEEE 802.11 Distributed Coordination Function”, *Wireless*

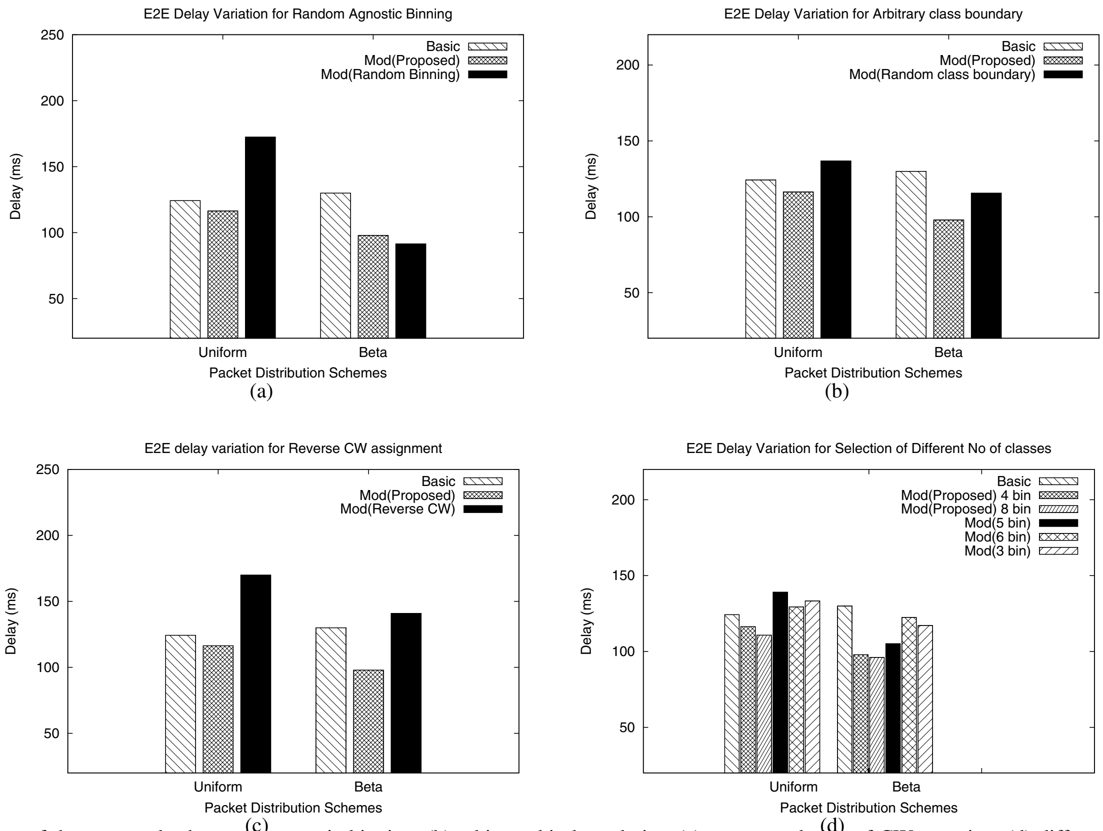


Fig. 6. Variations of the proposed scheme (a) agnostic binning, (b) arbitrary bin boundaries, (c) reverse ordering of CW mapping, (d) different number of bins

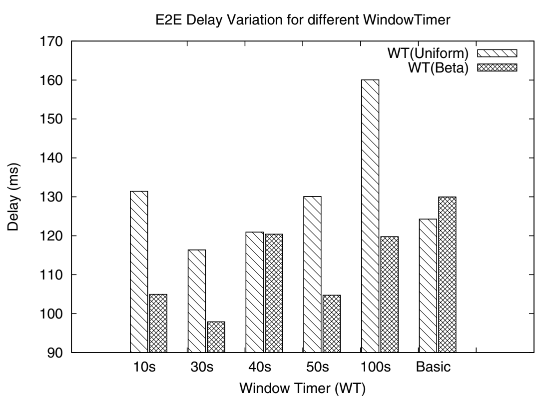


Fig. 7. E2E Delay Variation for different Window Timer

Communications and Networking Conference, WCNC, IEEE, Vol. 3, pp. 1654-1659, 2004.

[2] Wang, X., Yin, J., Agrawal, D.P., "Effects of Contention Window and Packet Size on the Energy Efficiency of Wireless Local Area Network", *Wireless Communications and Networking Conference, IEEE*, Vol. 1, pp. 94 - 99, 13-17, 2005.

[3] Ni, Q., Li, T., Turletti, T., Yang, X., "Saturation Throughput Analysis of Error-Prone 802.11 Wireless Networks", *Wireless Communications and Mobile Computing*, Volume. 5 (8), pp. 945 - 956, 2005.

[4] Balador, A., Movaghar, A., Jabbehdari, S., "History Based Contention Window Control in IEEE 802.11 MAC Protocol in Error Prone Channel", *Journal of Computer Science*, 6 (2), pp. 205-209, 2010.

[5] Bharghavan, V., Demers, A., Shenker, S., Zhang, L., "MACAW: a media access protocol for wireless LAN's. ", *ACM SIGCOMM Computer Communication Review*, London, United Kingdom, pp. 212–225, 1994.

[6] Chatzimisios, P., Boucouvalas, A.C., Vitsas, V., Vafiadis, A., Oikonomidis, A.,Huang, P., " A simple and effective backoff scheme for the IEEE 802.11 MAC protocol", *Cybernetics and Information Technologies*, Systems and Applications(CITSA), Orlando, Florida, USA, 2005.

[7] Fang, Z., Bensaou, B., Wang, Y., "Performance evaluation of a fair backoff algorithm for IEEE 802.11 DFWMAC", *ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Lausanne, Switzerland, pp. 48–57 (2002)

[8] Nandagopal, T., Kim, T., Gao, X., Bharghavan, V., "Achieving MAC Layer Fairness in Wireless Packet Networks", *ACM Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, United States, pp. 87–98 (2000)

[9] Luo, H., Lu, S., Bharghavan, V., "A new model for packet scheduling in multihop wireless networks", *ACM Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, United States, pp. 76–86,2000.

[10] Balador, A., Movaghar, A., Jabbehdari, S., Kanellopoulos, D., "Novel Contention Window Control Scheme for IEEE 802.11 WLANs", *IETE TECHNICAL REVIEW*, Vol. 29(3), pp. 202-212, 2012.

[11] Saraireh, M., AL-Saraireh, J. and Saraireh, S., "A Novel Adaptive Contention Window Scheme for IEEE 802.11 MAC Protocol", *Trends in Applied Sciences Research*, 9: 275-289, 2014.

[12] Ksentini, A., Nafaa, A., Gueroui,A., Naimi, A., "Deterministic Contention Window Algorithm for IEEE 802.11", *IEEE Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2712-2716, 2005.

[13] Cali, F., Conti, M., Grergori, E., "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit", *IEEE/ACM Transactions on Mobile Computing*, Vol. 1(1), 2002.