Dynamic Optimization of IEEE 802.11 DCF based on Active Stations and Collision Probability

Nithya Balasubramanian, Shreenivas Bharadwaj Venkataramanan and Aravind Aatham
Department of Computer Science and Engineering
National Institute of Technology,
Tiruchirapalli-620015, Tamil Nadu, India
{bnithyanitt, vshreenivasbharadwaj, aravind.cps}@gmail.com

ABSTRACT
In IEEE 802.11 wireless networks, the back-off algorithm has an exponential Contention Window (CW) increment and resetting CW strategy. Under heavy network load conditions the traditional Binary Exponential Back-off (BEB) algorithm has very low throughput and high collision probability. In this paper, an algorithm called Dynamically Optimized Contention Window (DOCW) is introduced to estimate the optimal Contention Window CW_{opt} according to the number of active competing stations and the current collision probability. The proposed algorithm dynamically adapts CW with respect to the current network states. Simulation of the proposed DOCW algorithm has been conducted to measure various parameters like throughput, collision probability, control overhead and packet loss. The simulation results for varying number of station reveal that the proposed algorithm outperforms the existing Novel Contention Window Back-off (NCWB) and the traditional BEB in randomly formed topologies.

KEYWORDS
DCF, Contention Window, Collision Probability, Wireless Networks, Heuristic, Regression.

1 INTRODUCTION
IEEE 802.11 wireless networks utilize the Distributed Coordination Function (DCF) to capture the shared medium with minimum of collisions. DCF follows the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol with Binary Exponential Back-off algorithm (BEB). In this mechanism, when a station is ready to transmit a packet, it senses the channel for an idle period equal to one Distributed Inter Frame Space (DIFS) [1]. If the channel remains unoccupied, the station transmits packet, otherwise, the transmission is paused until the current transmitting event is terminated. The stations continue to sense the channel until the medium is idle for DIFS interval, then a random back-off interval is generated. Random back-off intervals are discrete and hence slotted that is the packets are transmitted only at the beginning of each time slot. The random back-off value is chosen from the range $[0, CW_{current} − 1]$ with all values having equal probability of selection.

Whenever there is a collision between two or more stations as they transmit their packets, CW size is doubled. The back-off counter is suspended when detecting busy medium, and resumed after sensing an idle medium for the duration of DIFS interval. The back-off counter starts decrementing and continues to decrease as long as the idle medium is sensed. When the back-off timer reaches zero, the station transmits a packet at the beginning of the next time slot.

On receiving the packet successfully, the receiver waits for duration of Short Inter Frame Space (SIFS) and transmits an Acknowledgment (ACK) packet. On the other hand, if the transmission is unsuccessful which is interpreted by a timeout event in transmitter, a retransmission of the packet is scheduled and the CW_{current} is doubled with each unsuccessful transmission until it reaches its maximum value, i.e., $CW_{max} = 2^m \times CW_{min}$, where $m$ indicates maximum number of retransmissions. After reaching the maximum limit, the pending packet may be dropped or handled appropriately based on different implementations.

The major dependence on BEB drastically limits the performance of IEEE 802.11 wireless networks [2]. Large available bandwidth is not efficiently utilized in a high traffic and a congested network which results in the extremely poor performance of BEB [3]. Several
backoff algorithms have been proposed to amend the efficiency of BEB such as the Exponential Increase Exponential Decrease (EIED)[4], Multiplicative Increase Linear Decrease (MILD)[5], Linear Increase Linear Decrease (LILD)[5], Smart Exponential Threshold Linear (SETL)[6], etc. which modified CW size either linearly or exponentially by various methods described later in Section 2. The problem with all these algorithms is that they cannot handle drastic variations of the network states and decrease the CW in a similar fashion on successful transmission. Apart from the heavy network load, collisions can also occur when two stations sharing the same channel have equal back-off counter value [7]. It clearly states the need for a back-off algorithm which is dynamically optimized based on the network state. Algorithms like NCWB which overcomes the above limitations have a drawback that it does not consider the factor of collision probability in computing CW [8].

In this paper, a new Contention Window back-off algorithm called Dynamically Optimized Contention Window Back-off (DOCW) is proposed. This algorithm utilizes two decision parameters such as the number of active contending stations [9][10] and the current collision probability derived from Bianchi’s Markov model. The current collision probability factor gives extra acute information about the traffic of the network [11], thus improving the efficiency and if not included in the calculation, contention window may frequently deviate by a significant amount from the ideal contention window. The collision probability factor provides an inconceivable boundary to the contention window keeping it from getting out of bounds. The collision probability can also enhance quality of service (QOS) considerably [12]. The CW opt is increased when number of active stations in the network increase and similarly reduce the CW opt when number of active competing stations decrease. The DOCW performs significantly better under heavy network load conditions.

In this paper an efficient method to implement the algorithm and the environment of simulation is also presented. The simulation results for the traditional BEB, NCWB and the proposed algorithm DOCW are presented. The rest of the sections in the paper is structured as follows:

Section 2 discusses the related backoff algorithms with their CW adjusting strategies and highlights their limitations. In Section 3, the proposed algorithm is presented along with a fast efficient implementation for calculating CW opt. Section 4 presents the simulation results acquired from NS2 simulation and analyzes the performance of the proposed DOCW algorithm with other algorithms. Finally, the section 5 gives the concluding statements.

2 RELATED WORKS

2.1 EIED

This algorithm is based on exponential increment and decrements of CW.

\[
\begin{align*}
    cw_i &= \min(r_i, cw_{i-1}, cw_{\text{max}}), \quad \text{on collision} \\
    cw_i &= \max\left(\frac{cw_{i-1}}{r_D}, cw_{\text{min}}\right), \quad \text{on success}
\end{align*}
\]

where \(r_1\) and \(r_D\) are respectively the tweaking factors of the algorithm which can be optimized differently based on varying network conditions. The drawback of this algorithm is that it does not utilize the entire network bandwidth on high network loads. The CW becomes larger than desired and smaller than required, causing unnecessary delays.

2.2 MILD

Multiplicative increment and linear decrements of CW is proposed and which is given as follows:

\[
\begin{align*}
    cw_i &= \min(1.5, cw_{i-1}, cw_{\text{max}}), \quad \text{on collision} \\
    cw_i &= \max( cw_{i-1} - 1, cw_{\text{min}}), \quad \text{on success}
\end{align*}
\]

Though the algorithm slightly performs better than EIED, similar drawbacks such as the inefficient use of the total network bandwidth and usage of non-optimal CW is still prevalent.

2.3 LILD

This algorithm for the modification of CW is based on linear increment and linear decrement as shown below,

\[
\begin{align*}
    cw_i &= \min(cw_{i-1} + cw_{\text{min}}, cw_{\text{max}}), \quad \text{on collision} \\
    cw_i &= \max(cw_{i-1} - cw_{\text{max}}, cw_{\text{min}}), \quad \text{on success}
\end{align*}
\]

The linear increment and decrement does not accurately imitate the network load pattern and
thus cannot be used for an efficient utilization of the network bandwidth.

2.4 SETL

This algorithm is a novel combination of EIED and LILD. It defines a threshold value of CW threshold and when the CW current exceeds the threshold then it is tuned down in a clever manner. Though this algorithm is better than its predecessors it still does not adapt itself accordingly to the network state and thus can still be improved.

2.5 NCWB

This novel algorithm considers the number of competing stations to compute the CW as shown below,

\[
\begin{align*}
cw_{opt} &= \min \left( \frac{n \cdot cw_{min}}{2}, cw_{max} \right), \text{ on collision} \\
cw_{opt} &= \max \left( \frac{n \cdot cw_{min}}{2}, cw_{min} \right), \text{ on success}
\end{align*}
\]

where n is the number of active contending stations. The major drawback in this algorithm is that it does not consider collision probability in CW calculation.

2.6 LMILD

The algorithm Linear/Multiplicative Increase Linear Decrease provides a way to handle the excess traffic in the network. The factor mc, lc and ls provide a tweaking and optimizing opportunity. It can adapt the network to different conditions.

\[
\begin{align*}
cw &= \min \left( m_c \cdot cw, cw_{max} \right), \text{ on collision} \\
cw &= \min \left( cw + l_c, cw_{max} \right), \text{ overheard collision} \\
cw &= \min \left( cw - l_s, cw_{min} \right), \text{ on success}
\end{align*}
\]

The drawback is that though it provides an opportunity to adapt the algorithm it has to be done manually by tweaking the parameters. Manual optimization can be tedious and time consuming. It does not consider the collision probability in the CW computation thus losing vital information of the network.

2.7 MKBS

(m, k) back-off scheme uses the (m, k) form to determine the value of CW. VD^k is determined by the (m, k) form and the method defined as follows:

\[
\begin{align*}
cw_{new} &= 2^{V(D^k)} \cdot cw_{current}, \text{ no. of transmissions } \geq k \\
cw_{new} &= \alpha \cdot cw_{current}, \text{ no. of transmissions } < k
\end{align*}
\]

This algorithm does not adapt itself to the current network state and does not consider the influence of collision probability on the CW.

2.8 Priority based

The priority based method [13] seeks to solve the congestion problem by assigning priority weights to each node based on the traffic pattern.

\[
cw_i = \begin{cases} 
\frac{cw_{max}}{3} - 1, & \text{if priority } = 1 \\
\frac{2 \cdot cw_{max}}{3} - 1, & \text{if priority } = 2 \\
\frac{2 \cdot cw_{max}}{3} - 1, & \text{if priority } = 3 
\end{cases}
\]

Priority based method is a manual approach to tackle the congestion problem and can become very tedious. It does not consider the current traffic pattern and collision probability in the computation of CW.

All the above algorithms suffer from the disadvantage that they either modify CW linearly or exponentially or cannot handle drastic variations of the network states. They decrease CW in a similar fashion on successful transmission and also not considering collision probability factor in the CW. Apart from that, in heavy network load, collisions can also occur when two stations sharing the same channel may have equal back-off counter. The collision probability keeps the CW in check by providing a numerical restraint preventing it from going out of bounds. It also supplies the algorithm with vital information about traffic of the network improving efficiency of the algorithm and stabilizing it by not allowing it to fluctuate much.

The objective and motivation of the proposed DOCW algorithm is to handle all these drawbacks, include collision probability factor and the network state in CW computation and come up with a better algorithm.
3 PROPOSED DOCW ALGORITHM

The proposed DOCW algorithm makes use of the number of active stations computed from the Markov chain model, the current collision probability. It updates CW size appropriately thus resulting in an algorithm which efficiently adapts itself accordingly to the current network state and load.

![Figure 1. State Transitions for the Proposed Algorithm](image)

The number of active stations and the current collision probability is calculated in the state module in the lower part of the diagram and both the parameters (pc, n) are sent to all other modules for further computations. The collision state is invoked whenever there is a collision and success state is invoked on a success. There is also an initial state with CW value of 32 from which the success and collision states can be reached depending on whether success or failure. There are self-loops in collision and success states indicating the possibility of multiple successes or collisions. Similarly the success and collision states can directly interchange between themselves depending on whether success or failure. The collision and success modules are defined later in the paper.

3.1 Markov Chain model

The number of active stations and collision probability are computed from the model proposed by Bianchi [14]. Saturation condition is assumed in this model that is all stations always have ready packets for transmission. Each packet waits for a random back-off time prior to its transmission. For a given station, let c(t) be the stochastic process constituting it’s back-off time counter. Note that t is discrete and slotted. The back-off time counter of each station decrements at the beginning of each slot time which is unrelated to real time. The back-off timer decrements are halted when detecting busy medium. The time interval between beginnings of two consecutive slots may be longer than the slot time of constant size T, as it may also include a packet transmission. Value of the back-off counter c(t) of each station depends on its transmission history and thereby violating the Markov property.

![Figure 2. Markov Chain model](image)

We define \( cw = CW_{\text{min}} \) and m be the maximum back-off stage, such that \( CW_{\text{max}} = 2^m \cdot cw \), and let \( cw_i = 2^i \cdot cw \), where \( i \in (0, m) \). Let \( CW(t) \) be the stochastic process which is used to represent the back-off stage \( 0 \rightarrow m \) of the station at time t. The key approximation based on the Markov chain model is that, at each transmission attempt, all packets collide with constant and independent probability \( pc \) irrespective of the number of retransmissions incurred. The Markov chain process is represented by \( \{c(t), CW(t)\} \) as shown in Figure 2 [14]. In this Markov chain model, the non-null one-step transition probabilities are:
The society of digital information and wireless communications, 2017 ISSN: 2225-658X (online); ISSN 2412-6551 (print)

International Journal of Digital Information and Wireless Communications (IJDIWC) 7(2): 66-74

The first equation in (1) points that, at the beginning of each slot time, the back-off counter is decreased. The second equation in (1) gives the inference that a new packet following a successful transmission starts with back-off stage 0, and thus the back-off is uniformly picked from the range (0, cw0 − 1). Remaining two equations in (1) explain the system after an unsuccessful transmission.

Let,

\[ \lim_{t \to \infty} P(c(t) = i, CW(t) = k) = b_{i,k}, \]
\[ i \in (0, m), \ k \in (0, cw_{i-1}) \]

be the stationary function of chain.

\[ b_{i-1,0} \cdot p_c = b_{i,0} = p_c^i \cdot b_{0,0} \quad 0 < i < m \] (2)

\[ b_{m-1,0} \cdot p_c = (1 - p_c) \cdot b_{m,0} = \frac{p_c \cdot b_{0,0}}{1 - p_c} \] (3)

By chain regularities for k ∈ (1, cw1 − 1)

\[ b_{i,k} = \frac{cw_i - k}{cw_i} \cdot \left( \frac{1 - p_c}{p_c} \right) \sum_{j=0}^{m} b_{j,0} \]
\[ 0 < i < m \]
\[ b_{i,k} = \frac{cw_i - k}{cw_i} \cdot b_{i,0}, \quad i = 0 \] (4)

From equation (2), (3) and

\[ \sum_{n} b_{i,0} = b_{0,0} / (1 - p_c), \]

we can reduce (4) to,

\[ b_{i,k} = \frac{cw_i - k}{cw_i} \cdot b_{i,0}, \quad i \in (0, m), \quad k \in (0, cw_i - 1) \] (5)

Further by (2), (3) and (5) values b_{i,k} are in terms of b_{0,0} and p_c. After normalization b_{0,0} simplifies to,

\[ b_{0,0} = \frac{2.1 - 2p_c)}{(1 - p_c)} \] (6)

Now the transmission probability of a station can be computed as,

\[ p_t = \sum_{0}^{m} b_{i,0} = b_{0,0} / (1 - p_c) \]
\[ p_t = \frac{2.1 - 2p_c)}{(1 - p_c)} \] (7)

One should note that the value of collision probability of a transmitted packet is the same as, the probability that at least one of the n - 1 remaining stations transmit in a given time slot. Thus this gives,

\[ p_c = 1 - (1 - p_t)^{n-1} \] (8)

3.2 Estimation of Contention Window

The present number of active competing stations n is estimated from the above list of equations and from the Markov chain model [14],

\[ n = 1 + \frac{\log(1 - p_c)}{2(1 - p_c)} \] (9)

The proposed CW updates are as follows:

\[ CW_{opt} = \min \left( \frac{n \cdot CW_{min}}{2(1 - p_c)} \cdot CW_{max}, CW_{max} \right), \quad \text{on collision} \]
\[ CW_{opt} = \max \left( \frac{p_c \cdot n \cdot CW_{min}}{2}, CW_{min} \right), \quad \text{on success} \] (10)

The DOCW model is an improvement on the NCWB algorithm. An extra factor of 1 / (1 − pc) is introduced in collision update and the extra factor pc in success update. The collision probability factor gives a very important perspective view of the network traffic and helps in improving the efficiency of the model [15] and if it is not included, it may cause the CW to frequently deviate by a significant amount from the ideal value. The collision probability factor
provides a restraining effect on the CW and thus avoids the larger deviations of the CW from the ideal value. The collision probability pc is directly proportional to the optimal Contention Window; hence CW\textsubscript{opt} is directly proportional to pc during both collision and success. Intuitively one can say that on collision CW\textsubscript{opt} has to be increased and also at the same time should be directly proportional to pc so a factor 1 / (1 − pc) is added. Similarly on success CW\textsubscript{opt} has to be decreased and at the same time should be directly proportional to pc so a factor pc is added. Both these factors satisfy the incremental and decrement proportionality constraints of the CW.

\[
\begin{align*}
\text{cw}_{\text{opt}} & \propto pc, \quad pc \in (0,1), \quad \text{decremental} \\
\text{cw}_{\text{opt}} & \propto \frac{1}{1−pc}, \quad pc \in (0,1), \quad \text{incremental}
\end{align*}
\]

Collision probability being a crucial factor in determining the value of CW has been efficiently incorporated in the algorithm. Various simulations showed that this particular method gives the optimum results.

### 3.3 Implementation

For every contention, the collision probability is first numerically estimated using equations (7) and (8) by Newtonian regression and using this value of converged pc the number of active stations n is computed from (9). Newtonian regression is an efficient method by which the roots of an equation can be found by iterative convergence. Then, a station determines CW\textsubscript{opt} using the proposed algorithm in (10) based on whether the event is collision or success and then begins to perform the back-off procedure which is the same as DCF.

Note that only 5-6 iterations are required for the Newtonian regression to converge on equations (7) and (8) while finding pc because of the convex nature of the equations hence making the implementation very fast. In Newtonian regression, to determine the roots of f(x) = 0, f(x)' is required. Repeating the update x = x − f(x) / f(x)' till convergence, typically 5-6 iterations will yield the value of roots of the equation.

From equation (7),

\[
g(pc) = pt = \frac{2.(1−2.pc)}{(1−2.pc).(cw+1) + pc.cw(1−(2.pc)^n)}
\]

By rearranging equation (8),

\[
pt + (1−pc)^{\frac{1}{(n−1)}} − 1 = 0, \quad pt = g(pc)
\]

By solving the above equation (12), the value of pc can be computed. f(pc) is defined from (12) as,

\[
f(pc) = pt + (1−pc)^{\frac{1}{(n−1)}} − 1, \quad pt = g(pc)
\]

Objective is to find a value of pc such that f(pc) = 0. This can be accomplished by Newtonian regression as specified below,

\[
f(pc)' = \frac{d(pt)}{d(pc)} = \frac{(1−pc)^{\frac{2−n}{n−1}}}{n−1}, \quad pt = g(pc)
\]

pc is initialized to a random value to begin the regression process.

\[
\text{pc} = pc − f(pc) / f(pc)'
\]

Equation (14) is repeated till convergence. The resulting value of pc is used to compute n in (9) and further used to determine the optimized Contention Window using the proposed algorithm (10).

### 4 SIMULATIONS AND PERFORMANCE ANALYSIS

This section analyses the performance of the proposed DOCW, NCWB and BEB algorithm in terms of throughput, collision probability, control overhead and packet loss.

#### 4.1 Simulation Setup

Simulation is done based on a network scenario with n saturated stations. In saturated network condition, a station is always ready to transmit data packets. The simulation model is built using Network Simulator 2 (NS2). The simulation was carried out in a dynamic fashion and no manual
intervention was required. Table 1 shows the values of parameters utilized in the simulation of proposed DOCW, NCWB and the traditional BEB algorithms.

Table 1. Simulation parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload size (bits)</td>
<td>8184</td>
</tr>
<tr>
<td>MAC header size (bits)</td>
<td>272</td>
</tr>
<tr>
<td>ACK size (bits)</td>
<td>128</td>
</tr>
<tr>
<td>Data rate (bits / s)</td>
<td>112</td>
</tr>
<tr>
<td>Propagation delay (s)</td>
<td>1000000</td>
</tr>
<tr>
<td>Slot time length (s)</td>
<td>0.000001</td>
</tr>
<tr>
<td>SIFS length (s)</td>
<td>0.000050</td>
</tr>
<tr>
<td>DIFS length (s)</td>
<td>0.000128</td>
</tr>
<tr>
<td>Ack timeout length (s)</td>
<td>0.000300</td>
</tr>
</tbody>
</table>

4.2 Throughput analysis

The consistent throughput performance DOCW > NCWB > BEB is observed. As the network’s dynamic state is considered by including number of active stations there is a drastic improvement in throughput over the traditional BEB as shown in Figure 3. The proposed DOCW algorithm also performs significantly better than NCWB as collision probability (pc) is effectively utilized. The drawback of not including the collision probability factor (pc) in NCWB is clearly visible.

4.3 Collision probability analysis

The collision probability follows the trend DOCW < NCWB < BEB. The dependence of the algorithm on network’s dynamic state and the number of active stations (n) shows a drastic improvement in collision probability over the traditional BEB as shown in Figure 4. The Proposed algorithm also performs slightly better than NCWB as collision probability (pc) is efficiently used. The DOCW and NCWB start approximately from the same point but again as number of contender stations (n) increase DOCW clearly has a better and lower collision probability (pc) than the NCWB.

4.4 Control Overhead analysis

As shown in Figure 5 the graph for control overhead follows DOCW < NCWB < BEB. The control overhead packets RTS/CTS even though they are small if there are lots of collisions and packet losses the overhead will be considerably high. The proposed algorithm clearly has lower control overhead than NCWB and BEB. This is again due to the dynamic nature and effective use of collision probability factor. Both NCWB and DOCW start with little difference but as the number of contending stations (n) increases that is the traffic increases.
NCWB losses are higher than that of DOCW considerably. Introducing the collision probability factor (pc) has stabilized the DOCW and causes little deviation from the ideal value resulting in a higher performance than the NCWB.

4.5 Packet Loss analysis

Packet losses are also found to follow a different pattern. Both NCWB and the proposed algorithm perform better than BEB. But as network load and the number of active stations (n) increases the proposed algorithm clearly overtakes the NCWB and outperforms it. DOCW < NCWB < BEB can be approximated and inferred from Figure 6.

6 CONCLUSION

This paper proposed a Dynamically Optimized Contention Window (DOCW) algorithm for the IEEE 802.11 wireless network. The optimal Contention Window CW_{opt} is estimated according to active competing stations and current collision probability in order to achieve higher throughput by minimizing collision probability. The advantage of introducing collision probability is that the algorithm adapts Contention Window more closely to the network traffic and load pattern. The simulation graphs prove our hypothesis and outperform other algorithms. Future work can be to introduce load traffic pattern learning and faster convergence algorithms.

REFERENCES


