PAPER

An Efficient Backoff Algorithm Based on the Theory of Confidence Interval Estimation

Chunyang LEI[†a)], *Student Member*, Hongxia BIE[†], Gengfa FANG[††], Markus MUECK[†††], *and* Xuekun ZHANG[†], *Nonmembers*

SUMMARY Channel state estimation-based backoff algorithms for channel access are being widely studied to solve wireless channel accessing and sharing problem especially in super dense wireless networks. In such algorithms, the precision of the channel state estimation determines the performance. How to make the estimation accurate in an efficient way to meet the system requirements is essential in designing the new channel access algorithms. In this paper, we first study the distribution and properties of inaccurate estimations using a novel biased estimation analysis model. We then propose an efficient backoff algorithm based on the theory of confidence interval estimation (BA-CIE), in which the minimum sample size is deduced to improve the contention window tuning efficiency, while a fault-tolerance interval structure is applied to reduce the inaccurate estimations so as to improve the contention window tuning accuracy. Our simulation results show that the throughput of our proposed BA-CIE algorithm can achieve 99% the theoretical maximum throughput of IEEE 802.11 networks, thanks to the improved contention window tuning performance.

key words: *backoff algorithm, MAC protocol, wireless sensor network*

1. Introduction

Distributed wireless sensor networks (WSNs) have been widely used in surveillance systems to monitor and sense wide-range of environmental parameters. Due to the diverse applications they can be applied to, supporting the distributed wireless sensor networks becomes an essential part of 5G networks where Internet of thing (IoT) and vehicle to vehicle (V2V), machine to machine (M2M) networks are the enabling technologies. The future distributed wireless sensor networks may become extremely dense so that how to access and share the channel access among extremely large number of contending nodes becomes a new challenge.

In WSNs, contention based carrier sense multiple access with collision avoidance (CSMA/CA) protocol is applied to enable the channel sharing by multiple nodes in a distributed way. According to CSMA/CA, WSN nodes generate a random backoff time based on the predefined contention window before accessing the channel. The authors in [1], [2] found that a right contention window size that fits the channel contention state can efficiently solve the channel accessing problem. However, as the number of contending nodes becomes larger and more dynamic, precisely estimating channel contention state becomes very difficult. In this paper, we firstly focus on modeling and evaluating the channel contention, and then propose a new channel accessing algorithm which can tune the contention window size efficiently.

So far, the binary exponential backoff (BEB) based channel access algorithms [3] have been widely used in CSMA/CA based protocols, such as WiFi (IEEE 802.11b/g/n/ac), Zigbee (IEEE 802.15.4), etc. In these algorithms, the level of channel contention is estimated based on whether the previous packet has been transmitted successfully or not. Authors in [4] indicated that BEB based algorithms become unstable in highly dense and busy networks because of their over-estimation of the level of the channel contention. Thus, new backoff algorithms such as EIED [5] and MILD [6] are proposed in which contention window is tuned exponentially or linearly after each failure of channel access attempt. However, the estimation accuracy of channel contention in these algorithms has not been substantially improved.

Recently some backoff algorithms based on channel contention estimation have been proposed. On top of its own sending and receiving results, authors in [7]–[11] proposed ideas where nodes should monitor different kinds of channel state parameters to have a better understanding of the level of channel contention. For example, ABTMAC [7], EBA [8] and MLBA [9] algorithms estimate the channel contention level by computing the idle probability of the channel, while PCB [10] use the information on the number of interruptions during the backoff period, and ABE [11] is based on the number of consecutive idle slots of the channel. Based on the estimation results, PCB and ABE tune the contention window either exponentially or linearly, while MLBA applies a fast tuning scheme, and ABTMAC and EBA set the contention window by heuristic calculation. The above algorithms can make better decisions on channel sharing by sufficiently utilizing the precise channel state information. However, the above algorithms become very sensitive to the accuracy of the proposed channel contention estimation algorithms. In such algorithms, the estimation is done between two channel access attempts. The interval between two accessing attempts is a random variable, and the obtained sample size become very dynamic depending on the

Manuscript received December 18, 2015.
Manuscript revised March 22, 2016.
Manuscript publicized May 11, 2016.

[†] The authors are with School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China.

[††] The author is with School of Computing and Communications, University of Technology of Sydney, Australia.

[†††] The author is with Intel Mobile Communications, USA.

a) E-mail: leichunyang.2014@gmail.com

DOI: 10.1587/transcom.2015EBP3530

interval. Small number of samples can decrease the accuracy of the channel estimation, while high number of samples increases the complexity of channel estimation. As a result, how to make the channel estimation accurate and efficient becomes a critical issue for estimation based channel access algorithms.

In this paper, we solve the above problem by proposing a new backoff algorithm based on the theory of confidence interval estimation (BA-CIE). We analyze the characteristics of inaccurate estimations of wireless channel with different contention levels and analytically derive the distribution of the biased estimations. According to the theory of confidence interval estimation, we introduce the parameter of fault tolerant interval which presents the biased estimations while deducing the minimum sample size. By constructing the contention window tuning strategy, the negative impact from incorrect estimations can be effectively reduced. Mathematical analysis and simulation results show that higher tuning accuracy of our proposed algorithm is achieved by filtering large number of inaccurate estimations, while the tuning efficiency can be improved by adapting the contention window according to the predetermined number of samples.

The rest of this paper is organized as follows. We analyze the characteristics of inaccurate estimations in Sect. 2 and propose BA-CIE algorithm in Sect. 3. Simulation results are presented in Sect. 4 followed by the conclusions in Sect. 5.

2. Biased Channel Contention Estimation Analysis Model

2.1 Channel Throughput Optimization

Channel contention estimation is a key component in the design of estimation-based backoff algorithms since the channel contention level is very dynamic depending on the channel state and lots of other factors in the network. In this work we deal with the case where the wireless channel is shared by N nodes according to 802.11 DCF channel access method together with RTS/CTS scheme. As the data traffic pattens at application layer may be quite different across the nodes, we assume there are n out of the N nodes have data to transmit and thus try to access the wireless channel. We define these n nodes as contending nodes. The relationship between N and n can be fully derived given a data traffic model [12], [13].

We define τ as the probability that a contending node attempts to transmit in a given slot. As the work in [14] indicated, τ is only determined by its current contention window size (cw). Thus, we present τ as a function of cw:

$$\tau(cw) = \frac{2}{cw + 1} \tag{1}$$

The shared channel can be represented as a discrete time stochastic process consisting of three states: idle state, successful packet transmission state including processes of RTS/CTS/DATA/ACK and collision state which is indicated by a failed RTS. We denote T_I, T_S, T_C as the average durations of the three states accordingly and P_I, P_S, P_C as the probabilities of a node staying in the above three states accordingly.

Since the idle state appears only when no data arrives at any node, P_I can be expressed as a function of the data arrival rate τ and the number of contending nodes n, as shown in (2):

$$P_I(n, cw) = [1 - \tau(cw)]^n \tag{2}$$

The successful packet transmission in a given slot occurs when there is only one node transmitting. For the situation with n nodes contending the channel, P_S can be expressed as a function of n and cw as (3):

$$P_S(n, cw) = n\tau(cw)[1 - \tau(cw)]^{n-1} \tag{3}$$

The probability of the collision can be calculated based on P_I and P_S as follows:

$$P_C(n, cw) = 1 - P_S(n, cw) - P_I(n, cw) \tag{4}$$

From (1), (2), (3), (4), the system throughput can be expressed as (5). In this expression, the average transmission duration T_S, and the average collision duration T_C can be calculated at the Physical layer and MAC layer, and the slot duration T_I normally is defined by the standards such as the IEEE 802.11x [15], [16]. L_{data} is the average data frame size which can be easily estimated based on the previous frames transmitted.

$$TH(n, cw) = \frac{P_S(n, cw) \cdot L_{data}}{P_S(n, cw) \cdot T_S + P_C(n, cw) \cdot T_C + P_I(n, cw) \cdot T_I} \tag{5}$$

The relationship between cw and n for the optimal system throughput can be derived by making the first order derivative of (5) 0. We denote P_I^{opt} as the probability of an idle state when the channel throughput is maximized. Work in [17] proved that P_I^{opt} approximately equals to a constant.

Thus, the channel contention process can be represented by the probability of idle state. In order to achieve high channel utilization with different contention levels, we can just keep on estimating the probability of idle state of the channel, and try to make it reach P_I^{opt} as much as possible by correctly tuning the contention window size.

2.2 Biased Estimation Analysis

Although high channel utilization can be achieved by simply estimating the probability of idle state of the channel, inaccurate estimations are inevitable because the channel contention level changes very fast and there is a limited number of samples in practice. Next, we will focus on analyzing the channel estimation process considering the possible errors.

We denote $P_I(n, cw)$ as the right probability of idle state of a channel and $\hat{P}_I$ as the estimated one. In practice, $\hat{P}_I$ can be calculated by monitoring the channel state of the previous m slots. We use $X_k, k \in [1, m]$ to denote the channel state in the k^{th} slot. $X_k = 1$ if the channel is in idle state during the slot, otherwise $X_k = 0$. $\hat{P}_I$ can be expressed as bellow:

$$\hat{P}_I = \frac{1}{m}\sum_{k=1}^{m} X_k \tag{6}$$

Let's define $P(X_k)$ as the probability of X_k, and we can conclude that the distributions of $P(X_k), k \in [1, m]$ are independent $0-1$ of the successful probability of $P_I(n, cw)$. According to the properties of $0-1$ distribution, we calculate the expectation of X_k in (7) and the variance of X_k in (8) as follows:

$$\mu_I = P_I(n, cw) \tag{7}$$

$$\sigma_I^2 = P_I(n, cw) \cdot [1 - P_I(n, cw)] \tag{8}$$

According to the law of large numbers, $\hat{P}_I$ converges according to (9).

$$\lim_{m\to\infty} \hat{P}_I = \mu_I \tag{9}$$

The Central Limit Theorem (CLT) depicts the relationship between the sample size m and the distributional form of $\hat{P}_I$ around the deterministic number μ_I during this convergence. According to CLT, the distribution of $\hat{P}_I$ approaches the normal distribution with the mean of μ_I and the variance of σ^2/m while m increases.

We verified the distribution process of $\hat{P}_I$ with 8−nodes contending to access the channel where the optimal contention window size based on [17] is assigned to each node. The distributions of $\hat{P}_I$ with different sample size $m = 20$, $m = 50$ and $m = 100$ are shown in Fig. 1. From the figure, we can see that the distribution of $\hat{P}_I$ gradually approaches normal distribution when the sample size increases. At the same time, a larger sample size has a smaller variance of normal distribution which can make the estimation results closer to the optimal value P_I^{opt}.

Figure 1 also shows the variance of the throughput as a function of $P_I(n, cw)$ according to (5). We can see that the throughput remains relatively constant even when $P_I(n, cw)$ deviates from P_I^{opt} slightly. The throughput decreases dramatically when $P_I(n, cw)$ is below a certain value. If we denote $TL(p)$ as the throughput difference for $P_I(n, cw) = p$ and $P_I(n, cw) = P_I^{opt}$, we can see that $TL(p)$ increases monotonically for p over $[0, P_I^{opt}]$ and decreases monotonically over $[P_I^{opt}, 1]$.

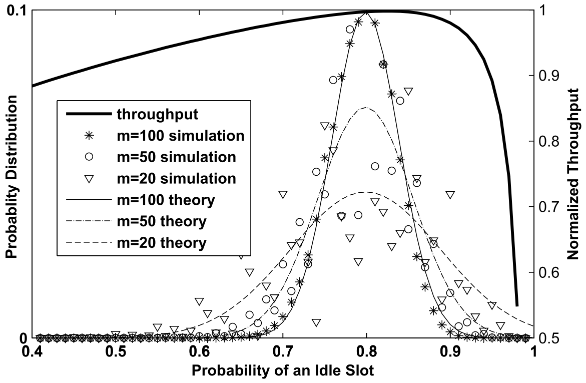

Fig. 1 Left coordinate: the distribution of $\hat{P}_I$ when $m = 20, 50, 100$ in 8-nodes contention channel. Right coordinate: the throughput performance of channels with different $P_I(n, cw)$.

3. Backoff Algorithm Based on Theory of Confidence Interval Estimation

3.1 Fault-Tolerant Interval

Based on the analysis above, we can conclude that although precise probability of idle state can hardly be estimated based on the limited number of samples, we still can make sure the biased estimations are within a certain range. For the central point of P_I^{opt}, we can define an interval with radius R. We call this interval as fault-tolerant interval in this paper. We make the interval contain the biased estimations as much as possible by applying the theory of confidence interval estimation.

According to central limit theorem, biased estimation is a normal distribution with expectation of P_I^{opt} and variance of $P_I^{opt} \cdot (1 - P_I^{opt})/m$. Based on the properties of normal distribution, the distribution of $\hat{P}_I$ can be transformed into the standard form as shown in (10):

$$U = \frac{\hat{P}_I - P_I^{opt}}{\sqrt{P_I^{opt} \cdot (1 - P_I^{opt})/m}} \sim N(0, 1) \tag{10}$$

According to the theory of confidence interval estimation, we can express the confidence interval of U with confidence level of $1 - \alpha$ as below:

$$P\{-u \le U \le u\} = 1 - \alpha \tag{11}$$

As u is a cumulative normal distribution, we have:

$$\Phi(u) = P(U \le u) = 1 - \frac{\alpha}{2} \tag{12}$$

$$u = \Phi^{-1}(\Phi(u)) = \Phi^{-1}(1 - \frac{\alpha}{2}) \tag{13}$$

By substituting (10) to (11), we can have

$$P\{-u \le \frac{\hat{P}_I - P_I^{opt}}{\sqrt{P_I^{opt} \cdot (1 - P_I^{opt})/m}} \le u\} = 1 - \alpha \tag{14}$$

and we can have

$$R = u \cdot \sqrt{P_I^{opt} \cdot (1 - P_I^{opt})/m} \tag{15}$$

we can see that when $P_I(n, cw) = P_I^{opt}$, $(1 - \alpha) \cdot 100\%$ sampled $\hat{P}_I$ will be within the range of $[P_I^{opt} - R, P_I^{opt} + R]$. In other words, if $1 - \alpha$ is large enough, almost all biased estimations will be in the range of $[P_I^{opt} - R, P_I^{opt} + R]$. Obviously, the length of fault-tolerant interval R is negative depending on the sample size m for certain confidence levels.

3.2 Backoff Algorithm Construction

As mentioned in the previous section, $\hat{P}_I$ can be estimated at the end of each m sample. The introduction of the fault-tolerant interval leads to three cases of $\hat{P}_I$ as below:

Case 1: $\hat{P}_I \in [P_I^{opt} - R, P_I^{opt} + R]$. When $\hat{P}_I$ deviates away from P_I^{opt} and still locates within the fault-tolerant interval, it is hard to tell if the deviation is caused by the wrong estimation or by the change of channel contention level. Thus in this case, our BA-CIE algorithm will simply ignore the estimation result if it is within the fault-tolerant interval.

Case 2: $\hat{P}_I \in [0, P_I^{opt} - R)$. As analyzed above, when $\hat{P}_I$ is smaller than $P_I^{opt} - R$, we believe that the channel contention has become more intense. As a result, the contention window needs to be increased to make P_I become closer to P_I^{opt}. In this case, we denote r_i as the new contention window size cw_{new} corresponding to the old contention window size cw_{old}. In our BA-CIE algorithm, r_i is calculated according to (16) below:

$$r_i = \frac{\log\left(P_I^{opt} - R\right)}{\log P_I^{opt}} \tag{16}$$

Case 3: $\hat{P}_I \in (P_I^{opt} + R, 1]$. In this case, $\hat{P}_I$ is larger than $P_I^{opt} + R$ which means the channel contention is less intense, so that the contention window should be decreased. We denote r_d as the ratio between the old contention window size cw_{old} and the new contention window size cw_{new} and is calculated based on (17) below:

$$r_d = \frac{\log P_I^{opt}}{\log\left(P_I^{opt} + R\right)} \tag{17}$$

There are four parameters, i.e., R, m, r_i and r_d, involved in our BA-CIE algorithm. R and m are correlated according to (15), while r_i and r_d can be derived from (16) and (17). As an example, we show some of the values and relations of the four parameters in Table 1.

We now present the detailed procedure of our BA-CIE algorithm in Alg. 1. In the BA-CIE algorithm, samples are acquired during backoff period. *BO_timer* is the value set for the backoff timer, while *BO_start* is the time when the backoff timer is triggered. *slot_cnt* is the total number of slots between two consecutive events that the backoff timer is triggered. *idle_slot_cnt* is the number of idle slots within the *slot_cnt* number of consecutive slots. *idle_slot_cnt*/*slot_cnt* is the probability of being in idle state for the channel in each slot above.

Table 1 Parameters configuration for BA-CIE ($\alpha = 0.99$).

R	r_i	r_d	m
0.0380	1.20	1.24	≥ 789
0.0915	1.50	1.80	≥ 136
0.1164	1.80	3.00	≥ 58
0.1723	2.00	4.98	≥ 39

In the BA-CIE algorithm, the above four parameters can be preset according to Table 1 as system parameters, and the other information for the backoff timer can be calculated by the Physical layer based on the previous communication results. The contention window tuning process will be updated according to the relationship between *idle_slot_cnt*/*slot_cnt* and the other two thresholds. Overall, our proposed backoff algorithm can be easily implemented in practice.

Algorithm 1 The procedure of BA-CIE

```
Require:
  Preset Parameters in Table 1
  $R, r_i, r_d, m$;
  Parameter Initial
  $BO\_timer, BO\_start$,
  $slot\_cnt = 0, idle\_slot\_cnt = 0$.
while monitored the backoff period do
  if the backoff timer is suspended or complete then
    $slot\_cnt+ = BO\_timer - BO\_start + 1$;
    $idle\_slot\_cnt+ = BO\_timer - BO\_start$;
    $BO\_start = BO\_timer$;
    if $slot\_cnt >= m$ then
      if $idle\_slot\_cnt/slot\_cnt < P_I^{opt} - R$ then
        $cw = cw * r_i$;
      end if
      if $idle\_slot\_cnt/slot\_cnt > P_I^{opt} + R$ then
        $cw = cw/r_d$;
      end if
      $slot\_cnt = 0, idle\_slot\_cnt = 0$;
    end if
  end if
end while
```

3.3 Backoff Algorithm Evaluation

Estimation results can be evaluated before they are applied by our BA-CIE algorithm. However, mismatches of contention window in the algorithm may not correctly represent the actual channel contention level. Next we will analyze the performance of our BA-CIE algorithm in details.

We denote the inverse function of (2) as:

$$N(p) = \frac{\log p}{\log(1 - cw) - \log(1 + cw)} \tag{18}$$

We define $n_{opt} = N(P_I^{opt})$ as the optimal number of contending nodes based on the current contention window size. By substituting (16) and (17) to (18), $n_U = N(P_I^{opt} - R) = N(P_I^{opt}) \cdot r_i$ and $n_L = N(P_I^{opt} + R) = N(P_I^{opt})/r_d$ respectively represent the number of contending nodes which will make the actual probability of idle state in channel deviates to the upper and lower boundaries of fault-tolerant interval. According to the conclusions in [8], [10], when the contention window size divided by the number of contending nodes in channel approximately reaches a constant, the probability of idle state in channel approaches P_I^{opt}. Here, we denote this certain constant as β. Therefore, we can easily calculate (19):

$$\frac{cw \cdot r_i}{n_U} = \frac{cw/r_d}{n_L} = \frac{cw}{n_{opt}} = \beta \tag{19}$$

Thus, we can conclude that in our BA-CIE algorithm, nodes ensure the changes of contending nodes by judging the location of $\hat{P}_I$, while if they obtain a $\hat{P}_I$ out of the fault-tolerant interval, they will immediately tune their contention window size to fit the new situation where $n = n_U$ or $n = n_L$, otherwise, the contention window is kept to fit the situation where $n = n_{opt}$.

We discuss the performance of the proposed algorithm by considering the next three cases:

Case 1: $n \in (n_L, n_{opt}) \bigcup (n_{opt}, n_U)$. In this case, no matter if the contention window is increased, decreased or just kept the same, the probability of idle state of the channel will never reach P_I^{opt}. As a result, the throughput will decrease for sure. According to the definition of $TL(p)$ in Sect. 2.2, we can easily get the upper bound of such throughput decreasing in (20):

$$TL_U = \max\{TL(P_I^{opt} + R), TL(P_I^{opt} - R)\} \tag{20}$$

Case 2: $n = n_L$ or $n = n_U$. According to the distribution of biased estimations, such changes can be detected with the probability of 50% from a single $\hat{P}_I$. The maximal throughput decreasing from the mismatches can be calculated by (20). When the the changes are successfully detected, the throughput will reach its optimal.

Case 3: $n \in (-\infty, n_L) \bigcup n \in (n_U, +\infty)$. Because of the change of n, the throughput will decrease evidently. However, the probability of detecting the changes exponentially grows. Especially, when $n \geq N(P_I^{opt} - 2R)$ (or $n \leq N(P_I^{opt} + 2R)$) the changes can be 100% detected. Tuning the contention window after each m samples, the throughput performance can be quickly back to the optimal.

Based on the analysis above, we can see that the throughput of BA-CIE algorithm is very close to the optimal one by adopting a pair of well maintained m and R.

4. Simulation Results and Discussions

In this section, we use the OMNET++ simulator [18] to study the performance of the proposed backoff algorithm. The basic PHY layer and MAC layer are based on standard IEEE 802.11 protocol. The related parameters used in our simulations are shown in Table 2. The realization of our proposed BA-CIE algorithm has extremely low complexity, so that it can be easily implemented on top of the standard IEEE 802.11 module without any change to the protocol or the hardware. The channel state statistics in the BA-CIE algorithm can be directly obtained from the carrier sense module which is used to accomplish the backoff counting in standard 802.11 module. Besides, several statistical parameters are calculated to implement all the functions of the BA-CIE algorithm. Next, we evaluate the performance of the BA-CIE algorithm from two aspects: the accuracy and the efficiency of contention window tuning.

Table 2 PHY AND MAC layer parameters in simulation.

Parameters	Value	Parameters	Value
Bit Rate	11 Mbps	Payload Length	1 KB
Slot Time	20 μ sec	MAC Header	224 bit
SIFS	10 μ sec	RTS	160 bit
DIFS	50 μ sec	CTS	112 bit
PHY Header	192 μ sec	ACK	112 bit

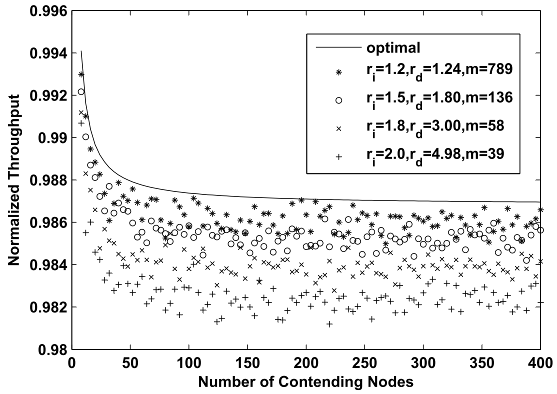

Fig. 2 Throughput performance of BA-CIE for different number of contending nodes varying from 4 to 400 with different parameters listed in Table 1.

4.1 Tuning Accuracy of BA-CIE

Firstly, we study the tuning accuracy by fixing the number of contending nodes in channel during the whole simulation. In this case, the tuning accuracy can be represented by the system throughput. If the system throughput is very close to the theoretically optimal one, we can claim that we achieved a high tuning accuracy. Figure 2 shows the simulation results of system throughput of the BA-CIE algorithm with different settings listed in Table 1. In our simulation, the number of contending nodes varies from 5 to 400. All simulation results are sampled during the time interval [20s,320s] when the network is stable. In the figure, the curve of optimal throughput is based on the theoretically optimal values calculated using (5). From the figure, we can see that our proposed BA-CIE algorithm can achieve more stable and higher system throughput not only in settings with low contention level but also in settings with extremely high contention level. The system throughput oscillates around a certain value and this is reasonable because when $P_I(n, cw)$ deviates slightly from the P_I^{opt}, contention window tuning process is not triggered. When the deviation reaches a pre-defined threshold, the contention window size is updated according to the proposed algorithm. The system throughput of the case with $r_i = 1.2, r_d = 1.24$ is very close to the optimal value on average. On the other hand, if we apply larger r_i and r_d, the contention window tuning becomes slightly more rough. Simulation results show that even in the case with $r_i = 2.0, r_d = 4.98$, the BA-CIE algorithm can

Table 3 Contention window tuning efficiency of BA-CIE with different parameters.

$\{r_i, r_d\}$	α	m	$TL(Kbps)$	$Time(s)$
{1.20,1.24}	**0.99**	**789**	**9.063**	**8.91**
	0.70	130	12.490	1.63
	0.50	54	15.896	0.77
{1.50,1.80}	**0.99**	**136**	**12.593**	**0.77**
	0.90	57	16.350	0.36
	0.70	23	21.720	0.2
{1.80,3.00}	0.9999	132	16.400	0.52
	0.99	**58**	**19.398**	**0.27**
	0.9	24	28.422	0.14
{2.00,4.98}	0.9999	89	26.425	0.41
	0.99	**38**	**24.810**	**0.16**
	0.9	16	37.585	0.08

effectively achieve over 99% of the optimal throughput.

4.2 Tuning Efficiency of BA-CIE

From Table 1 we can see that smaller r_i and r_d can achieve higher throughput while requiring larger sample size to keep high confidence level. However when channel contention level changes, a lot of time will be spent on sampling during which the wrong contention window size is applied which will result in the decreasing of the system throughput. We can just study the contention window tuning efficiency in the scenarios where the number of contending nodes changes.

We start the simulation with 4 contending nodes, and increase the number of nodes to 100 after 20 seconds. We calculate the system throughput of the BA-CIE algorithm with different system parameters every 0.01 s until the throughput becomes close to the optimal value. Table 3 shows the contention window tuning efficiency of the BA-CIE algorithm with different system parameters where TL represents the maximum throughput gap away from the theoretical optimal values, and the parameter $Time$ is the total tuning time spent. In the table, we bold the the cases with $\alpha = 0.99$ which have been studied in Fig. 2. From these results, we can see that $\{r_i, r_d\} = \{1.20, 1.24\}$ with $\alpha = 0.99$ can be hardly applied in dynamic networks because of the long tuning time of $8.51s$. Instead, $\{r_i, r_d\} = \{2.00, 4.98\}$ with $\alpha = 0.99$ can achieve the most efficient tuning efficiency with less than 0.2s because of its larger tuning amplitude and smaller sample size, while its system throughput does not perform as good as others. In conclusion, either smaller $\{r_i, r_d\}$ or larger α should be applied in static networks, otherwise either larger $\{r_i, r_d\}$ or smaller α can be applied for dynamic situations where channel contention level changes frequently.

4.3 Performance Comparison

In this subsection, we compare the performance of our proposed BA-CIE algorithm with that of other classic backoff algorithms including the BEB algorithm in IEEE 802.11 DCF [15], [16], and EIED [19], as well as the well-known feedback-based Idle Sense algorithm [7].

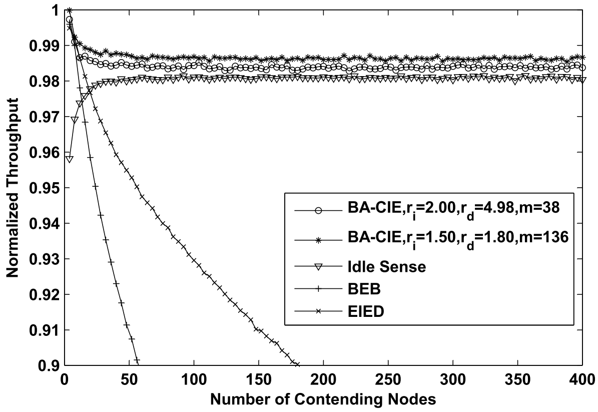

Fig. 3 Throughput performance of BA-CIE, BEB, EIED, and Idle Sense and for different number of contending nodes varying from 4 to 400.

In the simulation, two cases with parameters of $r_i = 1.50, r_d = 1.82, m = 136$ and $r_i = 2.0, r_d = 4.98, m = 38$ for our proposed BA-CIE algorithm are conducted. They represent the typical system throughput and contention window tuning efficiency that the BA-CIE algorithm can achieve. We run the situations with a different number of contending nodes ranging from 4 to 400 nodes, and focus on the performance of throughput in both dynamic and static settings.

Figure 3 shows the system throughput of our proposed BA-CIE algorithm in comparison with these of other algorithms. From the figure, we can see that the throughput of BEB and EIED decrease rapidly while the number of contending nodes increases because of the rough contention window tuning scheme applied. Idle Sense algorithm can achieve high and stable throughput performance in high contention settings, but its throughput decreases when channel contention level is low. This is because low channel contention leads to shorter backoff time during which fewer samples about the channel contention level can be obtained. Channel state information based on insufficient samples decreases the system throughput dramatically. In our BA-CIE algorithm, the insufficient samples problem is well solved by changing the sample size to better suite the situation. At the same time, the contention level prediction in high congestion situations is also improved by choosing better confidence intervals. Because of the above new features, our proposed algorithm provides much more stable and higher throughput than those of the other algorithms in different network settings.

Figure 4 shows the contention window tuning efficiency of different algorithms as a function of the number of contending nodes according to the sequence of 4, 50, 4, 100, 4, 200, 4, 300, 4, 400, 4. In the simulation, each number of contending node runs for 5 seconds. From the figure, we can see that more time is spent on contention window tuning when the channel contention changes more dramatically. Our proposed BA-CIE algorithm achieves fastest tuning speed compared to other backoff algorithms. This is because the change of channel contention level can be

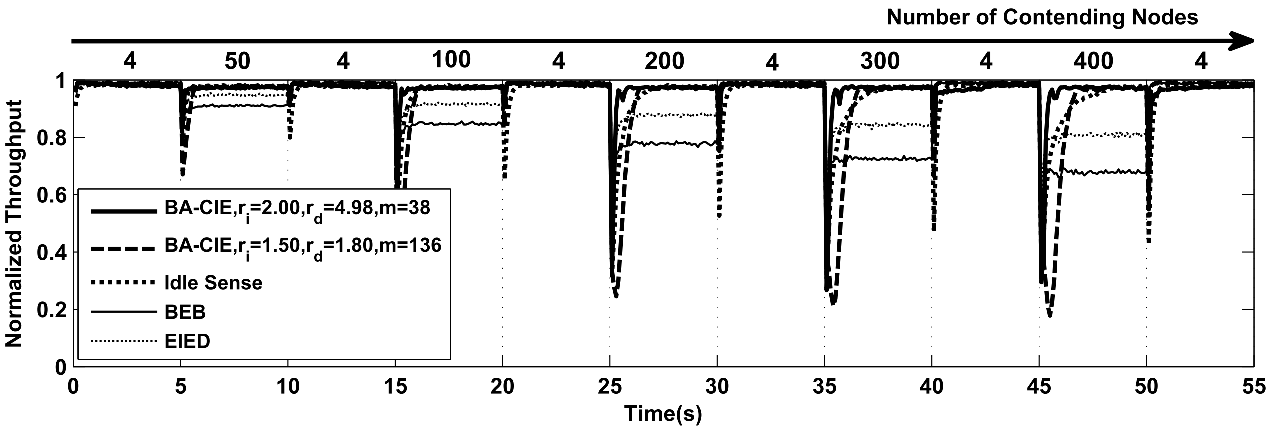

Fig. 4 Contention window tuning efficiency of BA-CIE, BEB, EIED, and idle sense for ever-changing networks.

quickly identified by the BA-CIE algorithm and the contention window can be tuned substantially towards its optimal value. For example, for the case with parameters of $r_i = 1.5, r_d = 1.82, m = 136$, our BA-CIE algorithm's total adaptation time is less than 0.5s even when the number of contention nodes suddenly changes from 4 to 400.

5. Conclusion

In the IEEE 802.11 based wireless networks, optimizing the contention window tuning process through estimating the channel contention level can improve the system throughput. However, it is very hard to precisely estimate the channel condition on-the-fly. As a result, biased channel contention level estimation methods are more practical. Previous work does not specifically address this problem introduced by the biased channel contention level estimation. In this paper, we analyzed the probability distribution of these results at different contention levels, and proposed an efficient backoff algorithm based on the theory of confidence interval to filter incorrect and useless estimations so as to improve the accuracy and efficiency of contention window tuning in dynamic settings. To evaluate the performance of the proposed algorithm, we performed comprehensive simulations to compare with BEB, EIED and Idle Sense backoff algorithms. Simulation results show that the system throughput of our proposed algorithm is much higher and is very close to the theoretical throughput limit of the IEEE 802.11 DCF access scheme with different network settings. Besides, by selecting appropriate system parameters, the contention window tuning efficiency of our proposed algorithm is also improved remarkably.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 41174158) and The Ministry of Land and Resources P.R.C Special Project in the Public Interest (No. 201311195-04).

References

[1] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," IEEE/ACM Trans. Netw., vol.8, no.6, pp.785–799, 2000.

[2] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism," IEEE J. Sel. Areas. Commun., vol.18, no.9, pp.1774–1786, 2000.

[3] H. Al-Ammal, L.A. Goldberg, and P. MacKenzie, "Binary exponential backoff is stable for high arrival rates," STACS 2000, Lecture Notes in Computer Science, vol.1770, pp.169–180, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[4] D. Xu, T. Sakurai, and H.L. Vu, "An analysis of different backoff functions for an IEEE 802.11 WLAN," 2008 IEEE 68th Vehicular Technology Conference, pp.1–5, 2008.

[5] D. Singh, B. Pandey, G.S. Tomar, and B.K. Sarkar, "Performance evaluation of backoff method — Effect of backoff factor on exponential backoff algorithm," 2013 5th International Conference on Computational Intelligence and Communication Networks, pp.82–86, 2013.

[6] S. Suming, Y. Zhi, Y. Min, and P. Yali, "Logarithm increase-linear decrease backoff algorithm based on self-adaptive optimal contention window," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), pp.V10-657–V10-661, 2010.

[7] A. Jamali, S.M.S. Hemami, M. Berenjkoub, and H. Saidi, "An adaptive MAC protocol for wireless LANs," J. Commun. Netw., vol.16, no.3, pp.311–321, 2014.

[8] S.-W. Kang, J.-R. Cha, and J.-H. Kim, "A novel estimation-based backoff algorithm in the IEEE 802.11 based wireless network," 2010 7th IEEE Consumer Communications and Networking Conference, pp.1–5, 2010.

[9] C. Lei, H. Bie, G. Fang, and X. Zhang, "An adaptive channel access method for dynamic super dense wireless sensor networks," Sensors, vol.15, no.12, pp.30221–30239, 2015.

[10] H.-M. Liang, S. Zeadally, N.K. Chilamkurti, and C.-K. Shieh, "A novel pause count backoff algorithm for channel access in IEEE 802.11 based wireless LANs," International Symposium on Computer Science and its Applications, pp.163–168, 2008.

[11] N. Abdeddaim, F. Theoleyre, M. Heusse, and A. Duda, "Adaptive IEEE 802.15.4 MAC for throughput and energy optimization," 2013 IEEE International Conference on Distributed Computing in Sensor Systems, pp.223–230, 2013.

[12] K. Fukuda, "Towards modeling of traffic demand of node in large scale network," 2008 IEEE International Conference on Communi-

cations, pp.214–218, 2008.

[13] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic Gaussian processes for data fusion in large scale terrain modeling," 2010 IEEE International Conference on Robotics and Automation (ICRA), pp.3452–3459, 2010.

[14] F. Alizadeh-Shabdiz and S. Subramaniam, "Analytical models for single-hop and multi-hop ad hoc networks," Mobile Netw. Appl., vol.11, no.1, pp.75–90, 2006.

[15] LAN/MAN Standards Committee of the IEEE Computer Society, IEEE Standard for Local and Metropolitan Area Networks Part 11b: Higher Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std 802.11b-1999, IEEE: New York, NY, USA, 1999.

[16] LAN/MAN Standards Committee of the IEEE Computer Society, IEEE Standard for Local and Metropolitan Area Networks Part 11n: Enhancement for higher throughput, IEEE Std 802.11n-2009, IEEE: New York, NY, USA, 2009.

[17] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," SIGCOMM Comput. Commun. Rev., vol.35, no.4, pp.121–132, 2005.

[18] OMNeT++, "OMNeT++ discrete event simulator." https://omnetpp.org/

[19] N.-O. Song, B.-J. Kwak, J. Song, and L.E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," The 57th IEEE Semiannual Vehicular Technology Conference, 2003, VTC 2003-Spring, pp.2775–2778, 2003.

Chunyang Lei received his B.S. degree in Information Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2010. He has been taking successive postgraduate and doctoral programs of study for doctoral degree since September 2010. He is currently pursuing the Ph.D degree in Signal and Information processing from Beijing University of Posts and Telecommunications, Beijing, China. His research interests include wireless access technologies and wireless sensor networks.

Hongxia Bie received the Ph.D degree from JiLin University, China in 2000. She is currently a professor in the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. Her main research interests include multimedia information processing and wireless data transmission.

Gengfa Fang received his Master in Telecommunications from Zhejiang University and Ph.D in Wireless Communications from the Institute of Computing Technology, Chinese Academy of Sciences in 2002 and 2007 accordingly. He is now a Senior Lecturer at the Department of Engineering, Macquarie University. He has published over 70 papers and 8 patents on embedded wireless network systems, MAC protocols, cross-layer design, wireless resource management for 5G networks. His research has been supported by CSIRO, NICTA, NXP, Zarlink and Intel.

Markus Mueck is with oversees Intel's technology development, standardization and partnerships in the field of spectrum sharing. In this capacity, he has contributed to standardization and regulatory efforts on various topics including spectrum sharing within numerous industry standards bodies, including ETSI, 3GPP, IEEE, the Wireless Innovation Forum and CEPT. Dr. Mueck is an adjunct professor of engineering at Macquarie University, Sydney, Australia, he acts as ETSI Board Member supported by INTEL and as general Chairman of ETSI RRS Technical Body (Software Radio and Cognitive Radio Standardization). He has earned engineering degrees from the University of Stuttgart, Germany and the Ecole Nationale Supérieure des Télécommunications (ENST) in Paris, France, as well as a doctorate degree of ENST in Communications.

Xuekun Zhang received his B.S. degree in Electrical & Information Engineering from Anhui University of Technology in 2009 and his M.S. degree in signal and information processing from Chengdu University of Technology in 2012. He is currently pursuing the Ph.D degree from Digital Media Lab. of Multimedia Technology Center in Beijing University of Posts and Telecommunications, Beijing, China. His research interests include OFDM transmission, MIMO, Synchronization and Channel estimation.