Abstract—Traditional Physical Carrier Sensing (PCS), which aims at eliminating hidden terminals in wireless networks, causes too many exposed terminals and deteriorates the throughput per user seriously. Some existing work has proven that an aggressive PCS can improve the throughput by balancing the tradeoff between hidden terminals and exposed terminals. However, little work has been conducted to optimize the PCS according to the network conditions. In this paper, we develop an analytical model to study the behaviors of a user in the aggressive PCS scenario, with which the optimal PCS threshold can be derived. Then, to avoid the complicated computation, we present a heuristic algorithm, in which the parameters required for PCS tuning are estimated by the carrier sensing mechanism employed in IEEE 802.11. Simulation results show that the proposed heuristic algorithm can make the PCS threshold approach to the theoretical optimal one. Moreover, the proposed heuristic algorithm can obtain significant throughput gain compared to the traditional PCS threshold setting solutions.

I. INTRODUCTION

In IEEE 802.11 [1] based wireless networks, increasing the throughput per user is of great challenge due to the restriction of Carrier Sensing Multiple Access (CSMA) mechanism. With CSMA, a user performs Physical Carrier Sensing (PCS) before data transmission to sample the energy in the channel. The transmission will proceed only if the sampled energy is below a threshold known as the PCS threshold. In the CSMA based wireless networks, the throughput per user can be approximately given by $(1 - P_L) \times C/n_c$, where $C$, $P_L$, and $n_c$ refer to the channel capacity, frame loss rate, and the number of users in the PCS range, respectively. Herein, $C$ is constrained by the employed physical technologies and the channel conditions. $P_L$ mainly depends on the number of hidden terminals which is relevant to the PCS threshold, and $n_c$ is determined by the PCS threshold and user density. From [3], $P_L$ generally increases with the PCS threshold, while $n_c$ decreases with the PCS threshold.

Traditional PCS threshold tuning algorithms [4][5] aim at eliminating the hidden terminals, i.e., $P_L \approx 0$. Under this design principle, optimizing the throughput per user is equivalent to minimizing $n_c$. One possible solution is to make the PCS range exactly cover the interfering range. However, it is too conservative to achieve high throughput. From experimental and theoretical points of view, the authors of [2] and [3] respectively demonstrated that an aggressive PCS threshold, which allows the existence of hidden terminals, enhances the throughput per user significantly. With the aggressive PCS threshold, both $P_L$ and $n_c$ vary with the PCS threshold, and it is very difficult to compute the optimal PCS threshold.

There are two major challenges for optimizing the PCS threshold which have not been solved yet: one is to construct a closed-form expression to describe the relationship between the throughput per user and the PCS threshold; the other is to adapt the PCS threshold to the varying network conditions. Given the user density, $n_c$ can be described by a closed-form expression with respect to the PCS threshold. Then, the essential part left is to construct the relationship between $P_L$ and the PCS threshold, which has been demonstrated very difficult in [3]. On the other hand, in practical networks, the optimal PCS threshold in fact dynamically changes with the varying network conditions (such as the user density and the transmit probability of each user), which are tough to be estimated on-line.

In this paper we develop an analytical model to investigate the relationship between the throughput per user and the PCS threshold. Then, we derive a balance equation, whose root is the optimal PCS threshold. Through some reasonable assumptions, all parameters of the balance equation are expressed with the network conditions those can be sensed by CSMA mechanism. Then, a heuristic algorithm is proposed to dynamically tune the PCS threshold to approach to the optimal value. One important feature of the heuristic algorithm is that it avoids the complicated computation to the optimal PCS threshold, and thus it can improve the network performance on-line. Extensive simulation results show that the proposed heuristic algorithm can make the PCS threshold approach to the optimal one.

The rest of the paper is organized as follows. In Section II, we develop an analytical model to investigate the throughput per user. With the obtained theoretical results, in Section III we propose a heuristic algorithm. In Section IV, extensive
simulation results are given to evaluate the proposed algorithm. Related work on our topic is summarized in Section V. Finally, the conclusions are drawn in Section VI.

II. THEORETICAL ANALYSIS

In this paper, we concentrate on a wireless network working in ad hoc mode. In physical layer, an interference-limited pathloss fading model is considered, and in MAC layer a simple $p$-persistent protocol is employed to approach to the effect of IEEE 802.11 MAC protocol for convenience of analysis. In [10], the authors illuminated that the $p$-persistent closely approximates the standard protocol with the same average backoff window size. Therefore, it is reasonable to infer the behaviors of the standard protocol from the analytical results based on the $p$-persistent protocol.

A. Throughput per user

Let $S$ denote the average throughput per user, defined as the average payload transmitted by an arbitrary user in a time slot, i.e.,

$$ S = \frac{\mathbb{E}[\text{payload transmitted by the user in a time slot}]}{\mathbb{E}[\text{length of a time slot}]} . $$

(1)

To calculate $S$, let us analyze the behaviors of an arbitrary user in a time slot. Let $P_c$ denote the probability that the user transmits a data frame successfully in a time slot. Let $L$ denote the payload size of a data frame. Then, the average payload transmitted successfully by the considered user in a time slot is given by $P_cE[L]$, where $E[L]$ represents the expected value of $L$. The channel has three states: busy due to successful transmissions, busy due to collisions, and idle. Let $P_c$, and $P_i$ denote the probabilities that the channel stays in collision and idle, respectively. Then, we get the expression of $S$ as follows:

$$ S = \frac{P_c E[L]}{(1 - P_c - P_i)T_s + P_c T_c + P_i \sigma} , $$

(2)

where $T_s$ is the average time of a successful transmission, $T_c$ is the average time of a collision, and $\sigma$ is the time of an empty time slot. From [1], we have

$$ \begin{cases} T_s = H + E[L]/r + T_{ACK} + \text{SIFS + DIFS}, \\ T_c = H + E[L]/r + \text{DIFS}, \end{cases} $$

(3)

where $H$ is the transmission time of a frame header (consists of a physical header and a MAC header), $r$ is data rate, and $T_{ACK}$ is the transmitting time of an ACK frame. In addition, SIFS and DIFS are a short interframe space and a distributed interframe space, respectively.

To simplify the analysis, we consider a saturation scenario where each user always has a data frame available for transmission. We use $p$-persistent CSMA to approach to the behaviors of the CSMA/CA mechanism with exponential backoff employed in the IEEE 802.11 standard. Let $p$ denote the probability that a user transmits in a given time slot. Let $P_L$ denote the frame loss rate, which is defined as the ratio of colliding transmissions to total transmissions. Then, we obtain the expressions of the parameters in (2) as follows:

$$ \begin{cases} P_s = p \ (1 - P_L), \\ P_i = (1 - p)^{n_c}, \\ P_c = 1 - n_c p \ (1 - P_L) - (1 - p)^{n_c}, \end{cases} $$

(4)

where $n_c$ is the average number of users in the PCS range. For the considered networks, all users are deployed uniformly with density $\lambda$, we have $n_c = \lambda \pi R_c^2$. Substituting (4) into (2), we have

$$ S = \frac{p \ (1 - P_L) E[L]}{n_c p (1 - P_L) (T_s - T_c) + (1 - p)^{n_c} \sigma + [1 - (1 - p)^{n_c}] T_c} , $$

(5)

In practical networks, $T_s$ is in order of millisecond, and is much larger than $T_{ACK}$+SIFS (just several decades microseconds). Therefore, from (3) it is reasonable to use $T_s \approx T_c$ to simplify (2). Then, we have

$$ S = \frac{p \ E[L]}{1 - P_L \{T_s [1 - (1 - p)^{n_c}] + (1 - p)^{n_c} \sigma \}} , $$

(6)

Recalled that our target is to tune the PCS threshold to maximize the throughput per user. In (6), $p$ and $T_s$ are all independent of the PCS threshold, and thus maximizing the throughput per user equals to minimizing the denominator. It is well known that $n_c$ is in proportion to the area of the PCS range, and thus the optimization objective is converted to find a $n_c$ to minimize the denominator of (6). For the convenience of analysis, we define a function $f(n_c)$ to describe the denominator, i.e.,

$$ f(n_c) = \frac{T_s [1 - (1 - p)^{n_c}] + (1 - p)^{n_c} \sigma}{1 - P_L} , $$

(7)

From the optimization theory, the function reaches the extremum when $n_c$ approaches to the root of $f'(n_c) = 0$.

From the analysis of [2][3], in the network allowing the existence of hidden terminals, the frame loss rate is dominated by the hidden terminals. In this way, we ignore the frame loss due to simultaneous transmissions in the interfering range. From the analysis in our previous work [14], the frame loss rate is determined by the average number of users in the PCS range and that in the hidden region, and the average number of users in the hidden region depends on the PCS radius. Thus, for the convenience of analysis, we employ a function $P_L = g(n_c)$ to describe the relationship between $P_L$ and $n_c$. Substituting $P_L = g(n_c)$ into $f(n_c)$ and making differentiation with respect to $n_c$, we have

$$ f'(n_c) = \frac{(\sigma - T_s) (1 - p)^{n_c} \ln (1 - p)}{1 - g(n_c)} \ + \ \frac{g'(n_c) \{T_s [1 - (1 - p)^{n_c}] + (1 - p)^{n_c} \sigma \}}{(1 - g(n_c))^2} \ (8) $$

$3$There are two reasons that cause a transmission fail: collisions due to the simultaneous transmissions in the interfering range, and collisions due to the concurrent transmissions in the hidden region. The probability of failure due to simultaneous transmissions is given by $1 - (1 - p)^{n_c - 1}$, and thus we have $0 < P_s < n_c p (1 - p)^{n_c - 1} < 1$. $4$The detailed derivation can be found in [14].
The optimal PCS threshold can be computed by \( f'(n_c) = 0 \). However, the theoretical results cannot be applied into the practical networks directly due to two reasons: first, there is no centralized point to collect network status information; second, due to the randomness of the CSMA backoff mechanism, it is very difficult to precisely estimate the network conditions on-line (such as the user density and the transmit probability \( p \)), which are needed to calculate the optimal PCS threshold. In this paper, a heuristic algorithm is developed, in which we employ limited information obtained by means of the CSMA mechanism to estimate the network conditions and to adapt the PCS threshold to the network condition.

III. HEURISTIC ADAPTIVE CARRIER SENSING

By the CSMA mechanism, each user can sense all transmissions occur in its PCS range. In this way, without the help of centralized point each user can maintain the following statistical information on-line:

- \( T_{\text{success}} \), the time that the channel is occupied by the successful transmissions of the user itself.
- \( T_{\text{capture}} \), the time that the channel is captured due to the transmissions of users within the reception range, which consists of the successful transmissions and the collided transmissions.
- \( T_{\text{busy}} \), the time that the channel is busy due to the transmissions of the users within the PCS range, which includes the transmissions in the reception range.
- \( T_{\text{idle}} \), the time that the channel is idle due to no transmissions in the carrier sensing range.
- \( n_r \), the number of users in the reception range.

Recall expression (8), we employ \(-p\) to approach to the value of \( \ln (1-p) \), and then \( f'(n_c) = 0 \) is simplified to

\[
\frac{1 - g(n_c)}{T_s [1 - (1-p)^{n_c}]} + \frac{(1-p)^{n_e}}{\sigma} + g'(n_c) = 0. \tag{9}
\]

From (6), the first item of (9) can be rewritten as

\[
\frac{1 - g(n_c)}{T_s [1 - (1-p)^{n_c}]} = \frac{(1-p)^{n_e}}{\sigma} = \frac{(T_s - \sigma)}{T_s}. \tag{10}
\]

In addition, from (4) and (6), \( (1-p)^{n_e} \) can be estimated by

\[
(1-p)^{n_e} = \frac{T_{\text{idle}}}{T_{\text{busy}}} + \frac{T_{\text{busy}}}{T_{\text{idle}}}. \tag{11}
\]

Then, substituting (10) and (11) into (9), the balance equation is given by

\[
\frac{(T_s - \sigma)}{\sigma T_{\text{busy}} + T_{\text{idle}}} + g'(n_c) = 0. \tag{12}
\]

As the analysis above, \( g(n_c) \) is the ratio of the number of collided packets and the total number of transmitted packets. In the reception range, the average number of data frames transmitted in a given time interval is in proportion to the sensed \( T_{\text{capture}} \), and the average number of successful transmissions of a single user is in proportion to \( T_{\text{success}} \). The number of users in the reception range is known as \( n_r \) in advance. In this way, we can estimate \( g(n_c) \) and \( n_c \) by

\[
g(n_c) = \frac{T_{\text{capture}} - n_r T_{\text{success}}}{T_{\text{capture}}}, \quad n_c = n_r \frac{T_{\text{busy}}}{T_{\text{capture}}}. \tag{13}
\]

To track the changes in network status on-line, our algorithm updates the estimates of the network status at intervals. In the simulation, the interval is set as 100 ms. During the \( i \)th interval, the values of \( \{T_{\text{success}}, T_{\text{capture}}, T_{\text{busy}}, T_{\text{idle}}\} \) are estimated by statistical measurement and denoted by \( \{T_{\text{success}}^{(i)}, T_{\text{capture}}^{(i)}, T_{\text{busy}}^{(i)}, T_{\text{idle}}^{(i)}\} \). Then, \( \{T_{\text{success}}, T_{\text{capture}}, T_{\text{busy}}, T_{\text{idle}}\} \) are updated after the \( i \)th interval as follows

\[
\begin{align*}
T_{\text{success}} &= \gamma T_{\text{success}}^{(i)} + (1 - \gamma) T_{\text{success}}^{(i)} \\
T_{\text{capture}} &= \gamma T_{\text{capture}}^{(i)} + (1 - \gamma) T_{\text{capture}}^{(i)} \\
T_{\text{busy}} &= \gamma T_{\text{busy}}^{(i)} + (1 - \gamma) T_{\text{busy}}^{(i)} \\
T_{\text{idle}} &= \gamma T_{\text{idle}}^{(i)} + (1 - \gamma) T_{\text{idle}}^{(i)}
\end{align*} \tag{14}
\]

where \( \gamma \) is a smooth factor, which is widely adopted in network protocols to obtain reliable estimates. Extensive simulations show that \( \gamma = 0.9 \) is a good compromise between accuracy and promptness, and hereby we adopt \( \gamma = 0.9 \) as the default value in the heuristic algorithm. After obtaining \( \{T_{\text{success}}, T_{\text{capture}}, T_{\text{busy}}, T_{\text{idle}}\} \), the frame loss rate \( g^{(i)}(n_c) \) and \( n_c^{(i)} \) can be updated by (13). Then, the PCS threshold can be dynamically tuned according to the value of (12). The pseudo code of the proposed algorithm is shown below. Herein, \( \eta \) is a constant used as learning-rate coefficient, which determines the convergence speed and the stability of the algorithm.

**Heuristic Algorithm**

1. Initialize \( \eta \), \( n_r \)
2. Do
3. Obtain the statistics of \( \{T_{\text{success}}^{(i)}, T_{\text{capture}}^{(i)}, T_{\text{busy}}^{(i)}, T_{\text{idle}}^{(i)}\} \)
4. Update \( \{T_{\text{success}}, T_{\text{capture}}, T_{\text{busy}}, T_{\text{idle}}\} \) by (14)
5. Estimate \( g^{(i)}(n_c) \) by (13)
6. Estimate \( n_c^{(i)} \)
7. Calculate \( g'(n_c) = \frac{g^{(i)}(n_c) - g^{(i-1)}(n_c)}{n_c^{(i)} - n_c^{(i-1)}} \)
8. Update the carrier sensing threshold by
9. \( P_{\text{pcs}} = P_{\text{pcs}} + \eta \left[ \frac{T_{\text{success}} - n_r T_{\text{success}}}{T_{\text{success}}^{(i)} + T_{\text{success}}^{(i-1)}} \right] \)
10. End do

IV. SIMULATION RESULTS

In this section, we take IEEE 802.11a protocol for example to evaluate the performance of the proposed heuristic algorithm with NS-2 [16] simulator. The characteristics of the physical layer and MAC layer used in the simulations can be found in IEEE 802.11a protocol. The SINR required for each data rate is the same to that shown in [4]. The reception radius is normalized as \( R_r = 1 \) unit length, and then the user density is referred to the number of users in a unit area. For each simulation scenario, the simulation time is 300 seconds, and the results are obtained via averaging values from 10 different runs with different seeds.
A. Convergence of Heuristic Algorithm

It is well known that the convergence speed depends on the learning-rate coefficient $\eta$. In this way, we implement a simulation to evaluate the convergence speed for various values of $\eta$. The results are plotted in Fig.1. From the figure, it is observed that the convergence speed increases with the learning-rate coefficient $\eta$: for $\eta = 1$, the user costs 2.6 s to reach the stable state; for $\eta = 4$, the user costs 0.7 s to reach the stable state. At the same time, we find that the oscillation extent increases with $\eta$ as well, and moreover the average throughput per user degrades with $\eta$ in stable state. This is mainly because that large learning-rate coefficient results in large oscillation around the optimal PCS threshold, which degrades the probability of attaining the optimum. From the consideration of the joint effect of convergence speed and oscillation extent, we employ $\eta = 2$ as the default value in our simulations.

B. Performance Comparison

For different data rates, the optimal PCS threshold varies. In this subsection, we first compare the average throughput per user obtained with the heuristic algorithm over different data rates. As shown in Fig.2, we get a surprising result that large data rate does not always lead to large throughput. The performance relates to the channel fading factor closely. From the figure, 18 Mbps can bring maximum throughput per user, and the gap compared to that with larger data rate (36 Mbps and 54 Mbps) degrades with the channel fading factor. It is well known that the SINR required for decoding increases with the data rate, which implies the interfering radius increases with the data rate as well. Therefore, for the same $P_L$, the PCS range increases with the data rate, which results in larger $n_c$, i.e., a large data rate does not always mean a large throughput. In addition, the larger the channel fading factor is, the smaller the increasing speed of interfering radius is. This explains why the throughput per user with large data increases faster than that with small data rate.

The performance comparison among the heuristic algorithm and those based on the optimal PCS threshold setting and the traditional PCS setting is plotted in Fig.3. It is observed that the proposed heuristic algorithm outperforms the traditional PCS threshold setting and is capable of approaching to the maximum throughput obtained with the theoretical optimal PCS threshold. In particular, when the data rate is 54 Mbps and user density is larger than 1.0, the proposed heuristic algorithm can obtain up to 200% throughput gain compared to the traditional PCS threshold setting.
In the end, we explore the impact of the traffic load on the performance gain. Herein, for the fixed user density, the variation of traffic load corresponds to the changes of transmit probability. The larger the transmit probability is, the larger the traffic load is. As shown in Fig.4, for different traffic load, the proposed heuristic algorithm can always approach to the results obtained from the optimal PCS threshold setting and outperforms the traditional PCS setting without hidden terminals.

V. RELATED WORKS

In the past decade, much attention has been attracted to improve the efficiency of the MAC protocol. For IEEE 802.11 based wireless networks, early works concentrate on optimizing the parameters of MAC protocol, such as transmit power [7][8], RTS/CTS handshaking [9], frame size, contention windows [10], and data rate [11], to improve the throughput. However, with the increasingly development of network scale, PCS, which determines the spatial reuse efficiency directly, has been an important research topic.

Existing works mainly concentrate on conservative PCS, which is designed to eliminate the hidden terminal problem completely, i.e., the PCS range covers the whole interfering range. With the conservative PCS threshold the frame loss rate is very low due to no hidden terminals, and then the throughput per user is mainly dependent of the spatial reuse efficiency. To maximize the spatial reuse efficiency, [5] proposed to tune the PCS threshold to make the PCS range exactly cover the interfering range. In addition, [4] and [13] discussed the effect of MAC header multi-rate multi-hop on conservative PCS tuning. However, so far less attention has been paid on aggressive PCS.

Among the few, in [3], by means of a Markov model and extensive simulations the authors investigated the effect of an aggressive PCS threshold on the frame loss rate and the aggregate one-hop throughput. However, the relationship between the PCS threshold and the throughput was just expressed with a set of equations but no closed-form solutions. The method for calculating the optimal PCS threshold was not addressed in [3], and no PCS on-line tuning algorithm was proposed, which we believe are the main contributions of this paper. In [2], we developed an experimental testbed based on Intel PRO/Wireless 2200 MiniPCI card to investigate the impact of aggressive PCS threshold on the throughput per user. Based on the testbed, we proposed a combination of Receiving-Sensitivity adaptation to reduce strong-last collision and a PCS adaptation to balance the hidden/exposed terminal problems. Due to no analysis on the optimization of the PCS threshold, the proposed method cannot set the PCS threshold to the optimum but just tune it to make the frame loss rate fall into a region given in advance.

VI. CONCLUSION

The PCS threshold is the key of the tradeoff between hidden terminal problem and exposed terminal problem, and thus optimizing the PCS threshold is an important and promising approach for enhancing the throughput per user. In this paper, we developed an analytical model to investigate the impact of the aggressive PCS threshold on the throughput per user. With a few reasonable simplification and a polynomial fit, we obtained a closed-form expression of the optimal PCS threshold. Moreover, based on the model, the access delay was studied as well. Then, according to the characteristics of practical networks, we proposed a heuristic algorithms, in which the parameters required to computing the optimal PCS threshold are estimated with the limited information obtained by the CSMA mechanism, to tune the PCS threshold dynamically to approach to the optimum. Extensive simulation results showed that with the proposed algorithm the throughput per user can approach to the optimum and is much larger than that with conservative PCS threshold.

REFERENCES