An Enhanced Cooperative MAC Protocol Based on Perceptron Training

Peijian Ju, Wei Song, and Dizhi Zhou
Faculty of Computer Science
University of New Brunswick, Fredericton, Canada
Email: {i1689, wsong, q5frc}@unb.ca

***Abstract*—Cooperation among wireless nodes at the medium access control (MAC) layer has attracted a lot of research attention in recent years. Most of existing cooperative MAC protocols focus on the scenarios with static helpers (relay nodes). However, when the helpers are moving around, the source node may choose a leaving helper with out-of-date information, which could cause performance deterioration. Hence, an optimal helper should not only support a high transmission rate but also have a low mobility. It can be a challenging problem to distinguish such an optimal helper when there are moving helpers of various mobility. In this paper, we extend the cooperative MAC protocol in [1] by means of perceptron training, referred to as *PTCoopMAC*. Making use of the handshaking messages in the original CoopMAC protocol, PTCoopMAC collects history data on the signal strength of overheard packets. Then, PTCoopMAC applies the perceptron training technique to obtain a weight vector to examine the stability of the helpers. Extending the CoopTable, PTCoopMAC selects the optimal helper depending on the achievable data rate as well as the prediction on whether a helper is reliable. The simulations results well demonstrate the throughput improvement of PTCoopMAC and its robustness to high mobility of helper nodes.**

***Index Terms*—Cooperative wireless communications, cooperative MAC protocols, throughput, mobility, perceptron training.**

I. Introduction and Related Work

Due to unique features such as path loss and fading, wireless links support a much less bandwidth than wired links. Although the multiple-input and multiple-output (MIMO) technology can exploit spatial diversity to improve wireless channel capacity, it is not feasible to integrate multiple antennas in palm-sized mobile terminals due to the constraints on size, weight and battery. As a result, cooperative communication techniques [2,3], which try to enable cooperation among mobile terminals to form virtual antennas, have received a significant amount of attention. A cooperating node can relay the received signal from the source in an analog or digital fashion, such as amplify-and-forward (AF) and decode-and-forward (DF) [4]–[6].

The medium access control (MAC) protocols can also take advantage of the benefit of cooperative diversity [7,8]. An indirect link via a relay can be used when the direct link between the source and destination is even worse. In IEEE 802.11 wireless local area networks (WLANs), the multi-rate capability can be further exploited [9] to enhance the aggregate performance of a multi-hop indirect link with a low-quality direct link. The cooperative MAC protocols need to address two fundamental questions [10,11]:

- ***When to cooperate?*** (Q1) The nature of this question is to find the conditions when cooperation can be enabled, or the regions where cooperation is beneficial.
- ***Whom to cooperate with?*** (Q2). This problem addresses two aspects: a) Who are the helpers (i.e., helper identification)? and b) Who is (are) the optimal helper(s).

TABLE I
Categories of cooperative MAC protocols.

Category	Q1 is answered by	Q2 is answered by
I	Source	Source
II	Helper(s)	Helper(s)
III	Source	Helper(s)
IV	Helper(s)	Source

According to the entities answering Q1 and Q2, Table I gives the four categories of cooperative MAC protocols:

- ***Category I.*** The source node answers both Q1 and Q2 such as in [1,9]. The source node maintains a table, known as *CoopTable*, by overhearing the transmission of other nodes. According to the entries in CoopTable, the source node decides whether a packet should be transmitted through the direct link or the indirect link, as well as which helper(s) will be chosen as the optimal one(s).
- ***Category II.*** The helpers answer both Q1 and Q2. The MAC protocol proposed in [12] is a typical example of Category II. Every node who overhears a packet transmission is a potential helper. Each potential helper estimates and decides whether to cooperate, that is, to answer Q1. If it is able to help, a handshaking signal is sent and indicates cooperation to perform. Afterwards, to answer Q2, a helper contention procedure is followed to select the optimal helper with the shortest response time.
- ***Category III and Category IV.*** In Category III, the source answers Q1, while the helpers answer Q2. In Category IV, the helpers answer Q2, while the source answers Q1.

Among all these categories, Category I is the most popular one. This is because that the idea of Category I is very similar

This research was supported by a research grant from Natural Sciences and Engineering Research Council (NSERC) of Canada.

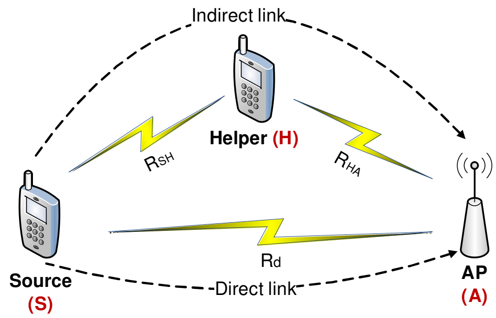


Fig. 1. The cooperation topology.

to 802.11 MAC. Hence, it is easier to revise 802.11 MAC to support cooperative links. Many studies on Category I protocols mainly focus on static scenarios. Although rDCF in [9] is compared with other protocols under mobile scenarios, it does not illustrate how the mobility of nodes affect the performance of rDCF. In [1], it is found that the throughput of CoopMAC degrades with a higher mobility level. However, how to address the node mobility remains unsolved.

In this paper, we propose an enhanced CoopMAC, referred to as *PTCoopMAC* to alleviate the negative impact of node mobility on throughput. In CoopMAC, the CoopTable maintained at the source is of crucial importance to make the cooperation decision. Since it is not feasible for the source to monitor every movement of a helper, the CoopTable entries may be out of date when the helpers are moving around instead of staying static. As a result, the source node is not timely aware that an optimal helper selected from CoopTable is leaving. In the proposed enhancement PTCoopMAC, a data mining technique known as *perceptron training* is employed to examine the history data of helpers and distinguish helpers of low mobility from highly mobile nodes. The CoopTable in PTCoopMAC is extended so that the selection of an optimal helper is dependent on not only current overheard packets but also history statistics of helper nodes. Simulation results show that PTCoopMAC with sufficient training data is more robust to helper mobility and achieves a much higher system throughput than CoopMAC, especially when helpers are moving frequently.

The remainder of this paper is organized as follows. In Section II, we introduce the cooperation topology and the CoopMAC protocol, and show the impact of helper mobility on system throughput. Then, an enhanced CoopMAC based on perceptron training is given in Section III. In Section IV, simulation results are presented and discussed, followed by conclusions and future work in Section V.

II. Impact Of Helper Mobility On CoopMAC

Consider a cooperation scenario in Fig. 1 for 802.11 WLAN. Suppose that multiple transmission rates are supported. Letting R_r denote the effective data rate over the indirect link, we have

$$R_r \triangleq \frac{1}{R_{SH}^{-1} + R_{HA}^{-1}} \tag{1}$$

where R_{SH} is the transmission rate between the source S and the helper H, and R_{HA} is the transmission rate between the helper H and the access point A. Given a fixed transmit power, the transmission rate decreases with a longer transmitter-receiver distance. Hence, it is reasonable to select the indirect link if the feasible transmission rate of the direct link (denoted by R_d) is less than the effective data rate of the indirect link, i.e., $R_d < R_r$. For example, when $R_{SH} = R_{HA} = 11$ Mbit/s and $R_d = 1$ Mbit/s, the indirect link is preferred since $R_r = (11^{-1} + 11^{-1})^{-1} = 5.5 > R_d = 1$.

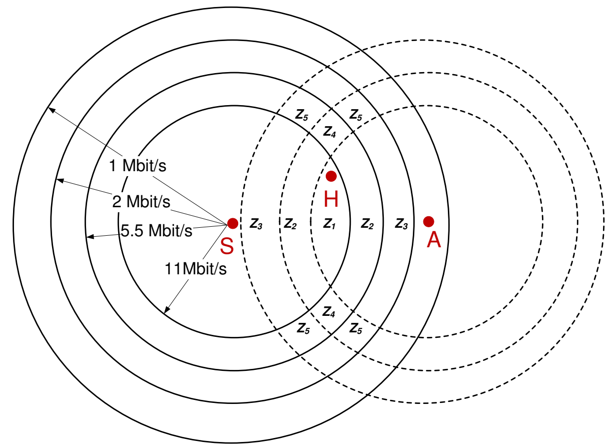


Fig. 2. The cooperation zones of S and A.

TABLE II
Indirect data rates within different cooperation zones.

Zone	Z_1	Z_2	Z_3	Z_4	Z_5
R_r (Mbit/s)	5.5	3.67	1.69	2.75	1.47

Assume that a transmission rate is achievable within a signal coverage of radius r_i. Then, several cooperation zones are formed as shown in Fig. 2 depending on the coverage regions of the transmitter (S) and the receiver (A) where different rates are supported. For example, if H is in Z_1 with its relative distance to S and A, a data rate of 11 Mbit/s is feasible for the source S and the helper H. According to (1), the effective data rate of the indirect link with a helper in Z_1 is 5.5 Mbit/s. For helpers in other cooperative zones, the effective rates of the indirect link can be found in Table II.

A. CoopMAC

Based on the above topology, CoopMAC [1] specifies the following procedure to accomplish a cooperation transmission:

- ***Overhearing and CoopTable Updating.*** When S is not using the channel, it keeps overhearing the transmission between H and A.
  - By overhearing RTS from H to A, R_{SH} is estimated.
  - By overhearing CTS from A to H, R_{HA} is estimated.
  - When S reserves the channel using RTS, the direct transmission rate (R_d) between S and A is obtained by extracting the information from the CTS of A.

  Then, R_{SA}, R_{SH}, and R_d are used to update the source's CoopTable as shown in Table III, where *Time* is when the

TABLE III
COOPTABLE STRUCTURE.

Helper Address	Time (s)	R_d (Mbit/s)	R_r (Mbit/s)	Failures
H_1 address	0.1	5	1.69	5
H_2 address	0.2	5	5.5	0
...	...	...	...	...

last packet is heard from a helper and *Failures* is the count of sequential transmission failures. Whenever the source has a packet to transmit, it checks whether there exists a better indirect link than the direct one. If $R_r > R_d$, cooperation is started with *Triangle Handshaking*. Otherwise, normal 802.11 transmission is used.

- ***Triangle Handshaking.*** As shown in Fig. 3, cooperative transmission through a indirect link is stared with a modified RTS to H and overheard by A. If H can help, it sends a Helper-ready-To-Send (HTS) to A. On receiving both RTS and HTS, A issues a Relay-Clear-To-Send ($RCTS$) to S and *Data Transmission* begins. Otherwise, a regular CTS is sent which indicates cooperation fails and normal 802.11 transmission is used instead.
- ***Data Transmission.*** On receiving $RCTS$, S sends its packet to H and H relays the packet to A to achieve a higher rate over the indirect link. On receiving the packet, an ACK is fed back directly to S.

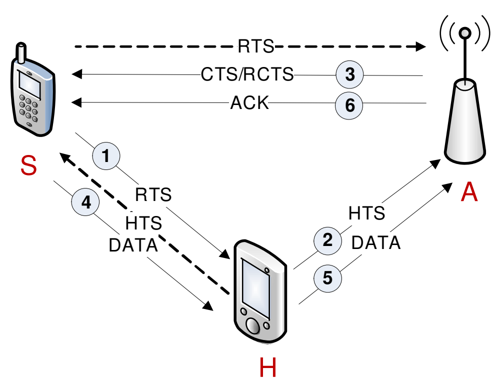


Fig. 3. Handshaking and data transmission procedures.

B. Impact of Helper Mobility

To evaluate the impact of moving helpers on system performance, we conduct simulations for the following mobility scenario. Assume that a helper node stays in a cooperation zone with a probability P_s. That is, the helper moves away from the current zone with a probability $1 - P_s$ and enters another zone with an equal chance. Hence, the larger the parameter P_s, the lower the mobility level that a helper has. Fig. 4 shows the simulation results on system throughput when there is one static helper in zone Z_1 and 4 moving helpers that move around in all the five zones with the same probability $1 - P_s$. Apparently, the static helper is the optimal helper, since it provides the highest transmission rate over the indirect

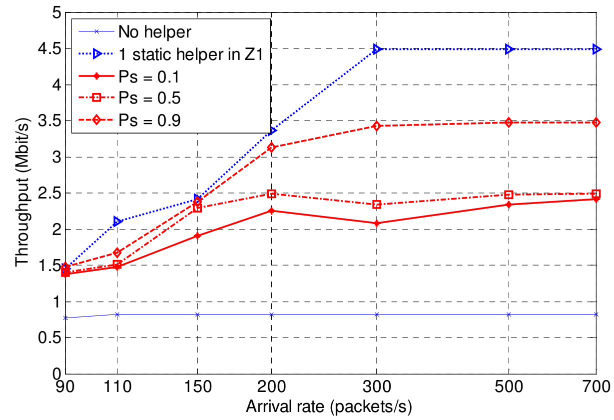


Fig. 4. Impact of moving helpers.

link. An intelligent cooperative MAC protocol should be able to distinguish the static helper from the other moving helpers and always choose this optimal helper. However, we find some interesting observations in Fig. 4 about CoopMAC:

- When there exist moving helpers and even these helpers have a very high probability of staying at a cooperation zone (e.g., $P_s = 0.9$), the system throughput degrades in comparison to the static scenario.
- The more frequently that helpers are moving (i.e., the lower the probability P_s), the more significant their impact on the throughput performance.
- The moving helpers act like noise and interfere with CoopMAC to make the right decision on relay selection.

These observations agree with the results in [1], where the mobility is evaluated in terms of pause time. The reason for the performance degradation with mobility is the out-of-date information in CoopTable. That is, when the source S chooses a helper to cooperate with, this helper may be no longer helpful because it has moved away from its original location. Intuitively, the more frequently the helpers are moving, the more cooperation failures S may experience and the worse the system performance is.

III. ENHANCED COOPMAC WITH PERCEPTRON TRAINING

To alleviate the negative impact of mobility, we enhance the CoopMAC protocol with perceptron training, which is referred to as *PTCoopMAC*. In particular, the CoopTable is extended with the statistics of each helper's history data obtained from overheard packets. By using a data mining technique known as perceptron training (PT), PTCoopMAC is able to distinguish and select a reliable helper of low mobility as the optimal one. In this section, the perceptron training technique is briefly introduced, followed by the details of PTCoopMAC.

A. Perceptron Training

Perceptron training (PT) is an artificial neural network to learn information from history data [13]. It is usually applied to make decisions when the data are continuous and linear separable. Considering the continuous data extracted from overheard packets and the implementation simplicity, we apply the PT technique to augment the CoopMAC protocol. Specifically, perceptron is defined as the following function

$$o(\vec{x}) \triangleq sgn(\vec{w} \cdot \vec{x}) \tag{2}$$

where $\vec{w} = [w_1, ..., w_i, ..., w_N]$ is a weight vector, $\vec{x} = [x_1, ..., x_i, ...x_N]^T$ (T for transpose) is a real-valued input vector of training data, and the function $sgn(\cdot)$ is given by

$$sgn(y) = \begin{cases} 1, & \text{if } y > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

The main task of perceptron training is to obtain a good weight vector $\vec{w}$ by processing the history data. For each input vector $\vec{x}$, a training output $o(\vec{x})$ can be obtained from (2), while a target output $t(\vec{x})$ is also known. Then, a gradient descent stochastic approximation algorithm [13] can be applied to determine the weight vector $\vec{w}$.

B. *PTCoopMAC Design*

To deal with system performance deterioration caused by moving helpers, PTCoopMAC employs handshaking messages of CoopMAC to capture history data of other nodes, so that perceptron training can be performed. Each sample of the history data has the following format

$$\vec{s} \triangleq [\sigma, \varepsilon, G] \tag{4}$$

where σ and ε are the variance and mean of the signal strength of overheard packets, respectively, and G is the cooperation result. If an RCTS is received, which indicates a successful cooperation, G takes the value 1. Otherwise, G is set to -1 if a regular CTS message is received. The first and the second elements in (4) comprise the input vector in (2), that is

$$\vec{x} \triangleq [\sigma, \varepsilon]^T. \tag{5}$$

For each RTS message, a training sample as (4) is stored in an extended CoopTable of the source node, which is given in Table IV. When sufficient history data are collected, the perceptron training algorithm is applied to update the weight vector $\vec{w}$. The optimal helper can then be selected according to the algorithm given in Table V.

The extended helper selection algorithm works as follows:

- Whenever the source S has a packet to transmit, it first checks the CoopTable. If the CoopTable is empty (i.e., no helper satisfies $R_r > R_d$), the regular 802.11 MAC will be applied. Otherwise, for each helper in the CoopTable, S selects the fields of the signal strength variance and mean (i.e., σ and ε) to form an input vector $\vec{x}$ as in (5).
- Each helper is evaluated by using the perceptron function defined in (2) with the weight vector $\vec{w}$ obtained above. If the output of the perceptron function satisfies $o(\vec{x}) = 1$, S adds the helper to the potential helper set $\mathbb{H}$. Otherwise, S just ignores this helper.
- After iterating all available helpers, S selects among the potential helper set $\mathbb{H}$ the one with the highest achievable data rate R_r. If multiple helpers provide the highest data rate, S should choose the one with the latest update time.

TABLE V
PERCEPTRON HELPER SELECTION ALGORITHM.

Initialize the source S with the weight vector $\vec{w}$
if a packet is ready to transmit **then**
  Potential helper set $\mathbb{H} \leftarrow \varnothing$
  Optimal helper $\hat{H} \leftarrow \varnothing$
  if CoopTable is empty **then**
    Transmit using 802.11 MAC
  else
    for each helper H_i recorded in CoopTable **do**
      $\vec{x} \leftarrow \{\sigma, \varepsilon\}$
      if $o(\vec{x}) == 1$ **then**
        $\mathbb{H} \leftarrow \mathbb{H} \cup H_i$
      end if
    end for
    $\hat{H} \leftarrow$ helper(s) in $\mathbb{H}$ with highest R_r
    if $size(\hat{H}) > 1$ **then**
      $\hat{H} \leftarrow$ helper in $\hat{H}$ with latest update time
    end if
  end if
  Return $\hat{H}$
end if

IV. SIMULATION RESULTS

Following the topology in Fig. 2, we conduct simulations to evaluate the performance of PTCoopMAC under different scenarios. The main simulation parameters are selected according to IEEE 802.11b. In this section, the system throughput of PTCoopMAC is compared to that of 802.11 MAC and original CoopMAC with different packet arrival rates and mobility levels for the helpers. We also show the impact of training data amount on the performance of PTCoopMAC.

TABLE VI
MAIN SIMULATION PARAMETERS.

Parameters	Values
PHY header	128 bits
MAC header	272 bits
RTS, HTS	160 bits + PHY header
CTS, RCTS, ACK	112 bits + PHY header
Packet payload	8184 bits
Path loss exponent	3
Propagation delay	1 μs
Radius of rate 1 Mbit/s	48.2 meters
Radius of rate 2 Mbit/s	67.1 meters
Radius of rate 5.5 Mbit/s	74.7 meters
Radius of rate 11 Mbit/s	100 meters

A. *Throughput Improvement with Different Arrival Rates*

Consider 8 helpers available to relay packets for the source S to the destination access point (AP). There are 4 helpers moving within all the five zones, 3 helpers moving in Z_1, Z_2,

TABLE IV
EXTENDED COOPTABLE STRUCTURE.

Helper Address	Time (s)	R_d (Mbit/s)	R_r (Mbit/s)	Failures	Variance of Signal Strength	Mean of Signal Strength
H_1 address	0.1	5	1.69	5	1.3029	-0.3727
H_2 address	0.2	5	5.5	0	1.1254	-0.1144
...	...	...	...	...	...	...

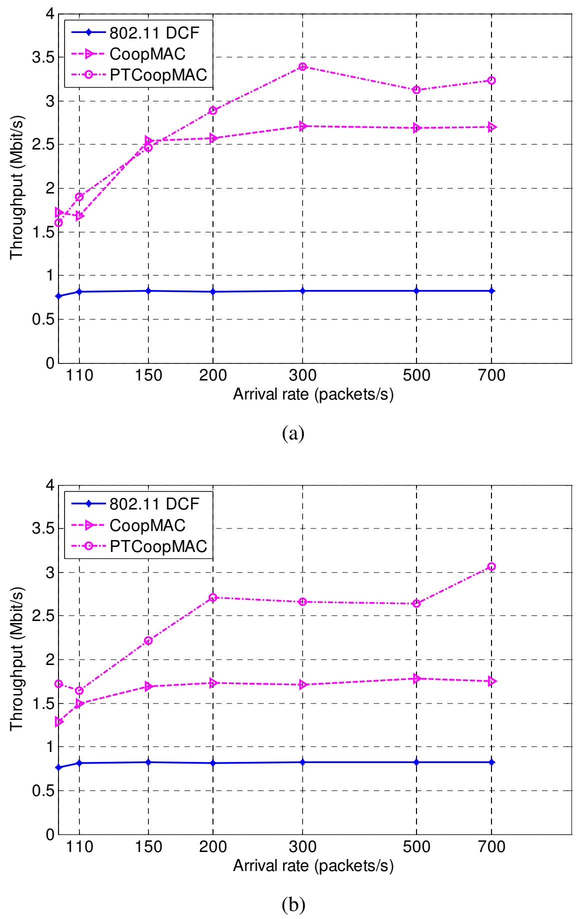


Fig. 5. Throughput with different arrival rates. (a) $P_s = 0.9$. (b) $P_s = 0.1$.

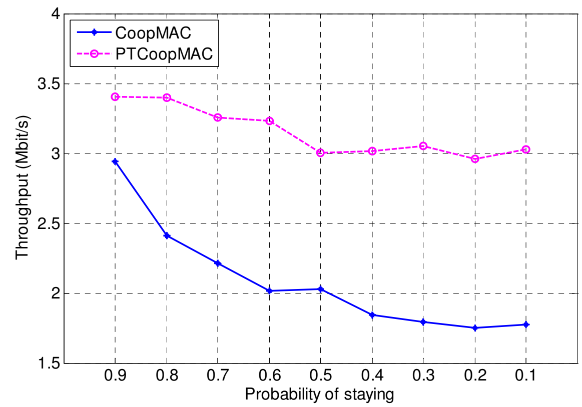


Fig. 6. Throughput with different values of P_s.

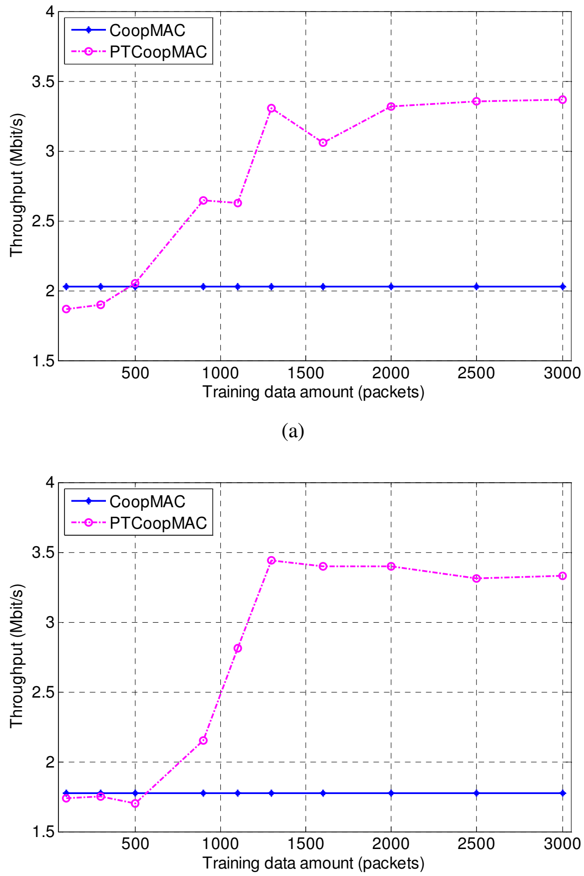
and Z_3, and one helper in Z_1. Fig. 5 shows the throughput achievable with PTCoopMAC, CoopMAC, and regular 802.11 MAC without helper relaying. According to the topology in Fig. 2, a transmission rate of 1 Mbit/s is feasible for direct transmission between S and A with 802.11 MAC. Excluding signaling overhead for RTS, CTS and ACK, the actual throughput is slightly below 1 Mbit/s. As seen in Fig. 5(a), CoopMAC and PTCoopMAC achieve a much higher throughput than 802.11 MAC without helpers. Given $P_s = 0.9$, when the packet arrival rate is below 200 packets/s, the throughput of PTCoopMAC is similar to that of CoopMAC. However, when the arrival rate is above 200 packets/s, PTCoopMAC consistently outperforms CoopMAC. The saturated throughput of CoopMAC is around 2.7 Mbit/s, while the throughput of PTCoopMAC is around 3.5 Mbit/s. As the arrival rate goes up, PTCoopMAC can make effective use of history data to select a stable optimal helper. Similar observation is found in Fig. 5(b) for $P_s = 0.1$. In this case, when the arrival rate exceeds 110 packets/s, PTCoopMAC starts to outperform CoopMAC.

B. Effectiveness to Address Helper Mobility

Comparing the saturated throughput of CoopMAC and PTCoopMAC in Fig. 5(a) and Fig. 5(b), we can see that the throughput degrades with a lower value of P_s. That is, when the helpers move around more frequently, the out-of-date problem of CoopTable becomes severer, which causes throughput deterioration. To further demonstrate the robustness of PTCoopMAC to helper mobility, Fig. 6 shows the performance of CoopMAC and PTCoopMAC with different P_s. Here, we consider 7 helpers moving within all zones from Z_1 to Z_5 and one helper within Z_1. The arrival rate is set to 300 packets/s, so that the saturated throughput can be achieved. As seen, the throughput of PTCoopMAC slowly drops from 3.4 Mbit/s to 3 Mbit/s when $P_s < 0.5$ and remains at around 3 Mbit/s when $P_s \geq 0.5$. In contrast, when the helpers have a higher mobility with a smaller P_s, the throughput of CoopMAC decreases fast from 3 Mbit/s to 1.8 Mbit/s. In other words, when P_s decreases from 0.9 to 0.1, the throughput of CoopMAC drops by 40%, while the throughput of PTCoopMAC only drops by 11.8%. Taking advantage of history data of helpers, PTCoopMAC can effectively address the mobility of helpers.

C. Impact of Training Data

To further investigate the impact of perceptron training on the performance of PTCoopMAC, we consider the simulation scenario in Section IV-B with different amounts of history data.



(b)

Fig. 7. Throughput with different training data amount. (a) $P_s = 0.5$. (b) $P_s = 0.1$.

Given $P_s = 0.5$, Fig. 7(a) shows the throughput performance of CoopMAC and PTCoopMAC. Since CoopMAC is irrelevant to training data, the throughput is constant at around 2.1 Mbit/s. As seen, when the training data amount goes up from 500 to 1300, the throughput of PTCoopMAC increases fast and becomes much higher than that of CoopMAC. This is because an accurate weight vector $\vec{w}$ can be obtained with sufficient training data, which increases the chance of locating an optimal helper. Finally, when the training data amount goes beyond 2000, the throughput of PTCoopMAC is stabilized at around 3.4 Mbit/s.

Fig. 7(b) for $P_s = 0.1$ illustrates a similar trend as Fig. 7(a). A difference is that the throughput of PTCoopMAC becomes stabilized when the training data amount reaches 1300 rather than 2000 as in Fig. 7(a). When P_s goes lower, the patterns hidden in the history data of higher-mobility helpers become more distinguishable and more linear separable from those of optimal helpers. According to the properties of perceptron training, the more linear separable the data are, the faster the weight vector $\vec{w}$ converges. As a result, an accurate estimate of the weight vector $\vec{w}$ can be obtained with less training data with a smaller value of P_s. Fig. 7 clearly demonstrates the effectiveness of PTCoopMAC especially when the helpers exhibit a high mobility.

V. Conclusions and Future Work

Cooperation at the MAC layer provides a good approach to make use of the multi-rate capability of 802.11 WLANs. As a popular cooperative MAC protocol, CoopMAC can improve the throughput performance by involving relaying helpers. Nonetheless, CoopMAC suffers from the throughput degradation with helper mobility. Especially when the helpers exhibit different mobility patterns, CoopMAC cannot identify the optimal helper with a lower mobility and higher achievable data rate. To address various mobility behaviors of helpers, we extend the CoopMAC protocol with perceptron training, referred to as *PTCoopMAC*. By collecting history data on the signal strength of overheard packets, PTCoopMAC can obtain a weight vector to examine the stability of the helpers. Extending the CoopTable, PTCoopMAC selects the optimal helper depending on the achievable data rate as well as the prediction on whether a helper is reliable. Simulation results demonstrate the significant throughput improvement and high tolerance to helper mobility. In the future, we would use more practical mobility and channel models to evaluate the performance of PTCoopMAC in various networking scenarios. We are also interested in considering advanced data mining techniques to provide more powerful enhancement to cooperation performance.

References

[1] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, “CoopMAC: A cooperative MAC for wireless LANs,” *IEEE J. Select. Areas Commun.*, vol. 25, no. 2, pp. 340–354, 2007.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity–Part I: System description,” *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, 2003.

[3] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity–Part II: Implementation aspects and performance analysis,” *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, 2003.

[4] A. Nosratinia, T. E. Hunter, and A. Hedayat, “Cooperative communication in wireless networks,” *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 74–80, 2004.

[5] A. S. Ibrahim, A. K. Sadek, W. Su, and K. J. R. Liu, “Cooperative communications with relay-selection: When to cooperate and whom to cooperate with?” *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2814–2827, 2008.

[6] W. Song, P. Ju, and D. Zhou, “Performance of cooperative relaying with adaptive modulation and selection combining,” in *Proc. International Conference on Computing, Networking and Communications (ICNC’13)*, 2013.

[7] M. S. Gokturk, O. Ercetin, and O. Gurbuz, “Throughput analysis of ALOHA with cooperative diversity,” *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 468–470, 2008.

[8] W. Song and W. Zhuang, “Performance analysis and enhancement of cooperative retransmission strategy for delay-sensitive real-time services,” in *Proc. IEEE GLOBECOM*, 2009.

[9] H. Zhu and G. Cao, “rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 9, pp. 1201–1214, 2006.

[10] H. Shan, W. Zhuang, and Z. Wang, “Distributed cooperative MAC for multihop wireless networks,” *IEEE Wireless Commun. Mag.*, vol. 47, no. 2, pp. 126–133, 2009.

[11] W. Zhuang and M. Ismail, “Cooperation in wireless communication networks,” *IEEE Wireless Communications*, vol. 19, no. 2, pp. 10–20, 2012.

[12] H. Shan, H. Cheng, and W. Zhuang, “Cross-layer cooperative MAC protocol in distributed wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2603–2615, 2011.

[13] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.