This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

Dynamic Channel Assignment for Wireless Sensor Networks: A Regret Matching Based Approach

Jiming Chen, *Senior Member, IEEE,* Qing Yu, Bo Chai, Youxian Sun
Yanfei Fan, Sherman Shen, *Fellow, IEEE*

Abstract—Multiple channels in Wireless Sensor Networks (WSNs) are often exploited to support parallel transmission and to reduce interference. However, the extra overhead posed by the multi-channel usage coordination dramatically challenges the energy-constrained WSNs. In this paper, we propose a Regret Matching based Channel Assignment algorithm (RMCA) to address this challenge, in which each sensor node updates its choice of channels according to the historical record of these channels' performance to reduce interference. The advantage of RMCA is that it is highly distributed and requires very limited information exchange among sensor nodes. It is proved that RMCA converges almost surely to the set of correlated equilibrium. Moreover, RMCA can adapt the channel assignment among sensor nodes to the time-variant flows and network topology. Simulations show that RMCA achieves better network performance in terms of both delivery ratio and packet latency than CONTROL [1], MMSN [2] and randomized CSMA. In addition, real hardware experiments are conducted to demonstrate that RMCA is easy to be implemented and performs better.

Index Terms—Channel assignment, regret matching, correlated equilibrium, wireless sensor network.

✦

1 INTRODUCTION

In general, many applications of Wireless Sensor Networks (WSNs) such as environment monitoring, medical care, target tracking, etc. may coexist in the same geographical region, as a result, the high sensor node density may exceedingly exacerbate the communication interference among sensor nodes. Single channel MAC protocols can not handle this surging interference efficiently. Moreover, current sensor nodes, which are usually equipped with one simple half-duplex transceiver, are able to operate on multiple channels. IEEE 802.11 standard for wireless communication provides multiple channels availability. By exploiting multi channel assignment, the sensor network can benefit better performance [1]. Hence, it is attractive to exploit multiple channels in WSNs to support parallel transmission and reduce interference in the highly dense sensor networks. Recently, there have been a considerable number of studies on multi-channel usage in wireless networks [4], [5], [6], [7], [8]. However, most of the existing works make some strong assumptions that the radio transceivers either use the frequency hopping spread spectrum wireless cards or can operate on multiple channels simultaneously. Unfortunately, such assumptions do not hold in WSNs, because current available sensor node has only one simple half-duplex radio transceiver. In addition, the extra overhead due to dynamic channel negotiations poses significant challenges to WSNs with constrained energy and limited bandwidth.

Recently, several multi-channel protocols have been proposed specially for WSNs and they can be divided into two categories. The first category is to assign channels in a static way based on the static network topology assumption [2], [9], [10], [11], [12], [13], [14]. These protocols cause very limited communication overhead. However, since they do not track the instantaneous transmission flows when assigning channels, they may make the links involved in the transmission flows bandwidth-tight but that not involved in the transmission flows bandwidth-excess. Moreover, both the network topology and the transmission flows are time-variant in practice. Thus, static channel assignment is not an efficient way to handle interference. The second category is to dynamically assign channels to links according to the instantaneous transmission flows [1], [15], [16], [17], [18], [19], [20], [21].

The MAC protocol for WSNs in [1] is designed and implemented on sensor motes with no specific assumptions on the application. The paper [15] focuses on how to incorporate both the advantages of multiple channels and TDMA into the MAC design with low overhead. The study [16] proposes an energy efficient multichannel MAC protocol, Y-MAC, for WSN to achieve both high performance and energy efficiency under diverse traffic conditions. A FDMA channel assignment in a non-cooperative wireless network is studied in [17]. The authors in [18] present an adaptive dynamic channel allocation protocol (ADCA) in wireless mesh network, which

---

- *Part of this paper was presented in IEEE ICC 2010, Cape Town, South Africa [3].*
- *Jiming Chen, Qing Yu, Bo Chai and Youxian Sun are with the State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China. (email: jmchen@ieee.org; chaibozju@gmail.com; {qyu, yxsun}@iipc.zju.edu.cn).*
- *Yanfei Fan and X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. (email: {yfan,xshen}@bbcr.uwaterloo.ca).*

contains both static and dynamic interfaces. The study [19] proposes a channel assignment scheme for cognitive radio networks (CRNs) that balances rate maximization and network connectivity. They focus on CRNs in which each node is equipped with multiple radios. [20] presents a comprehensive survey on spectrum assignment in spectrum assignment in cognitive radio networks. The study [21] proposes a dynamic spectrum assignment algorithm to maximize the number of secondary users that are satisfied in terms of throughput in a centralized CRN. Though these protocols can reduce interference to some degree, they all have to frequently exchange information globally or in a large neighborhood to perform channel usage negotiations and coordinations. Therefore, they cause considerable communication overhead to WSNs. Hence, an efficient channel assignment method for WSNs should be highly distributed with very limited information exchange.

Game theory has been adopted to construct distributed algorithms in WSNs, such as coverage [22], routing [23], sensor activation [24], querying [25], etc. In this paper, we model the channel assignment problem in WSNs as a game, and make each player perform a Modified Regret Matching procedure (MRM) [26] according to its own history information. In MRM, each player is highly autonomous, i.e., each player is only required to provide its own environmental information and with a low-level awareness of other players. Hence, MRM meets the requirements of channel assignment for WSNs: low overhead and decentralization.

In this channel assignment game, all sensor nodes are modeled as players, and the available non-overlapping channels which the sensor nodes use to receive packets are the actions of sensor nodes. The utility of each sensor node is made up of its valid receiving ratio (i.e., the ratio of the valid packets it has received to all the packets it has sensed) and the average packet transfer delay of all the valid packets it has received. Therefore, the utility of each sensor node can be completely measured by itself without exchanging information with other sensor nodes. To the best of our knowledge, this paper is the first attempt to use MRM to deal with the channel assignment problem in WSNs. Our main contribution is two-fold. First, we propose a highly distributed Regret Matching based Channel Assignment algorithm (RMCA). RMCA converges almost surely to the set of correlated equilibrium, in which the action of each sensor node is an optimal response to its environment and to the actions of other sensor nodes so that the whole network achieves a reasonable suboptimal network performance. RMCA also adapts the channel assignment dynamically to the time-variant transmission flows in the network to reduce interference efficiently. Simulation results of both the fixed flows and time-variant flows scenarios show that RMCA achieves better network performance than CONTROL [1], MMSN [2] and randomized CSMA. Second, we implement the proposed RMCA in real testbed and evaluate its performance. The experiment results demonstrate that: (1) RMCA is very convenient to be implemented in real hardware system; (2) RMCA is able to make the sensor nodes in the same collision domain use different channels; and (3) RMCA achieves better network performance in terms of both delivery ratio and packet latency than MMSN and randomized CSMA.

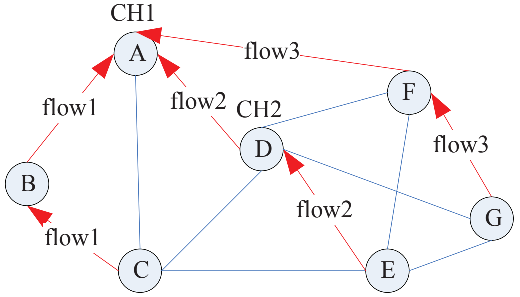


Fig. 1. An example of sensor network composed of seven sensor nodes with three transmission flows

The remainder of paper is organized as follows: Section 2 presents system model and describes the channel assignment problem. Section 3 proposes the regret matching based channel assignment algorithm for the channel assignment problem. The performance of the proposed algorithm is evaluated by extensive simulations and experiments in section 4. We conclude this paper in section 5.

2 SYSTEM MODEL AND PROBLEM DESCRIPTION

2.1 System Model

Consider a sensor network with multiple nodes. Each sensor node is equipped with one simple half duplex transceiver, and is able to operate on multiple channels. Moreover, each node can only choose one channel to deliver packet at each stage.

We consider that the assigned channels are all non-overlapping and do not interfere with each other, e.g., the channels which are at least two channels away from each other in the IEEE 802.15.4 compatible CC2420 chip [27] are such non-overlapping channels [10]. In addition, we select one channel from the available channels as a control channel to broadcast the channel assignment information. We assign the channels to sensor nodes in a receiver-centric way. Each sensor node selects a channel to receive packets, and broadcasts this information to its neighbors through the control channel. The neighbors with packets to delivery to the node should use this channel to send the packets. As shown in Fig. 1, sensor node A uses $CH1$ to receive, and thus sensor nodes B, D and F will use $CH1$ to transmit their packets to A. Furthermore, sensor nodes, which are in the same collision domain and use the same channel for transmission, perform CSMA/CA to contend for the medium access.

2.2 Problem Description

The interference suffered by the sender is quite related to the number of the receiver's neighbors which use the same channel to send packets as the sender. For example, in Fig. 1, if C, G and F also use $CH2$ to send packets, then the transmission from E to D would be interfered with by the transmissions of C, G and F. Therefore, E may perform more retries and even suffer collision. Moreover, in the receiver-centric channel

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

assignment way, the channel the sender uses to send is determined by the receiver of the transmission, and the receiver-sender relationship is determined by the flows in the network and the network topology. Therefore, the interference suffered by the network is quite related to the flows and network topology. Since the flows and network topology are usually time-variant in practice, the channel assignment should track the instantaneous flows and the time-variant network topology to reduce interference. Therefore, coordinations among sensor nodes are required. However, these coordinations usually result in large overhead to the network and challenge the energy-constrained WSNs. Such impact of the coordination packets has been studied in [2] by simulations. Its result reveals that the network throughput degrades significantly with the data packet length decreasing. Its also points out that the coordination packets brought by channel usage negotiation play a remarkable role in harming the network performance in WSNs, where the length of data packet is usually quite short and comparable to that of coordination packet. Therefore, reducing coordination packets is a critical goal when designing dynamic channel assignment algorithm.

In this paper, instead of explicitly coordinating, each sensor node only relies on a history of its observations to predict the environment variation and the actions of other sensor nodes, and then selects a channel to respond to this prediction. These observations are only a noisy aggregate indicator of the environment and the actions of other sensor nodes, rather than explicit observations of other sensor nodes. Thus, very limited information is exchanged, and energy consumption is potentially reduced. Furthermore, all the sensor nodes provide an immediate response to the variation of the flows and network topology, and the response can be improved over time.

3 REGRET MATCHING BASED CHANNEL ASSIGNMENT

3.1 Metrics to measure radio interference

We conduct a group of experiments to study the interference suffered by one sensor node. Three main metrics—Packet Delivery Ratio (PDR), Valid Receiving Ratio (VRR) and Average packet Transfer Delay (ATD)—are considered to evaluate the degree of interference. For a sensor node receiving packets, some of them are sent to it and called valid packets while others are not sent to it but overheard by it. In this case, VRR is defined as the ratio of the valid packets the sensor node has received to all the packets heard by it. ATD is defined as average packet transfer delay of all the valid packets.

The experiments are conducted with Imote2 nodes (we will specifically describe this type of sensor node in subsection 4.2). Twelve nodes are placed on a 4-by-3 grid with each edge equaling to 25cm. In such a deployment, each node can hear from the other 11 nodes. Sensor node A is sending packets to sensor node B incessantly, while the other sensor nodes use the same channel to send packets incessantly, which interfere with the communication between A and B. We call these sensor nodes interference nodes. All sensor nodes perform CSMA/CA to contend for medium access. We vary the number of interfering sensor nodes from 0 to 10. In each round of experiment, A tries to delivery 5000 packets with packet length of 50 bytes to B. The PDR, ATD and VRR of the link A toward B are shown in Fig. 2. It can seen that when the number of interfering nodes increases from 0 to 10, PDR decreases from 99% to 42%, VRR decreases from 100% to 11%, and ATD increases from 5ms to 20ms. These observations can be explained as follows. First, VRR reflects the number of sensor nodes contending for the access to the same channel that sensor node B uses in its collision domain, i.e., more interfering sensor nodes lead to a smaller VRR. Second, ATD reflects the time that a packet toward sensor node B should try for a successful receipt, i.e., more trying time results in a longer delay and thus indicates a larger interference [28].

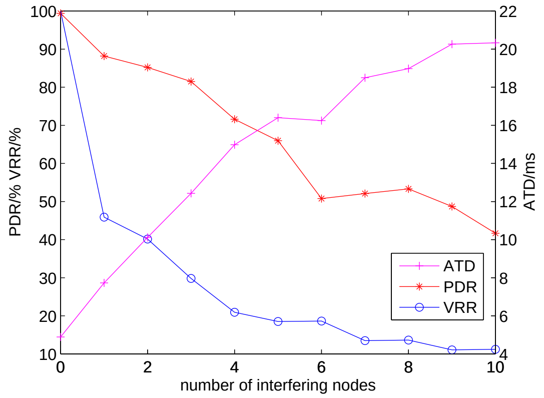


Fig. 2. PDR, ATD and VRR vs number of interfering nodes

In addition, when the interference increases, i.e., the number of interfering nodes increases, PDR and VRR decrease accordingly while ATD increases. Thus, PDR, VRR and ATD can represent the interference suffered by the sensor nodes in a great degree. In addition, PDR needs to gather information from both source and sink at the same time, while both VRR and ATD can be calculated independently without explicit requirement of the information of other sensor nodes. VRR also reflects the network performance in term of delivery ratio, and ATD reflects the network performance in term of packet latency as well. In order to simplify the data collection in real application as well as to properly measure the interference, we choose VRR and ATD as the metrics to characterize the interference.

3.2 Channel Assignment Game

To decentralize the coordinations of channel assignment, we model the channel assignment problem in WSNs as a game. In this game, all sensor nodes in the network are players and denoted by a set $P = \{1, 2, ..., n\}$, where n is the number of sensor nodes in the network. All the available non-overlapping channels make up the action space of each senor node, and are denoted by a set $C = \{1, 2, ..., c\}$ where c is the number of the available non-overlapping channels. The game is designed to be a repeated game. At each stage, sensor node $i \in P$ selects a channel $s_i \in C$ to improve its own benefit, and the channels selected by all the sensor nodes $s = (s_1, s_2, ..., s_n)$ form an instantaneous channel assignment.

The benefit of each sensor node is characterized by its utility function. To characterize the interference suffered by the sensor nodes, we construct the utility function of each sensor node $i \in P$ at stage k as follows:

$$u_i^k(s_i^k, s_{-i}^k) = \digamma \times r_i(k) - d_i(k), \tag{1}$$

where $r_i(k)$ represents the VRR of sensor node i at stage k, $d_i(k)$ represents its ATD at stage k, $\digamma$ is a weighted factor to balance the effect of the two parts, and s_i^k and s_{-i}^k represent the channel selected by sensor node i and the channels selected by all the other sensor nodes except sensor node i, respectively.

The utility function quantifies the utility that each sensor received. Each sensor node aims to minimize ATD and maximize VRR. In general, the principle of setting parameter $\digamma$ can be described as follows: firstly, the parameter setting should be based on the application scenario. Secondly, if VRR should be considered in priority, we can choose a larger $\digamma$. Otherwise, we choose a smaller $\digamma$.

3.3 Regret Matching Based Channel Assignment Algorithm

In order to achieve a better tradeoff between energy consumption and network performance, we make each sensor node perform a Modified Regret Matching procedure (MRM) [26] to play the channel assignment game. The average regret of sensor node i from $x \in C$ to $y \in C$ at stage k is defined as follows:

$$R_i^k(x,y) = \left[\frac{1}{k} \sum_{t \le k : s_i^t = x} [u_i^t(y, s_{-i}^t) - u_i^t(s^t)] \right]^+, \tag{2}$$

where $[x]^+$ represents the larger one between x and 0, and $u_i^t(s^t) = u_i^t(x, s_{-i}^t)$. Equation (2) has a clear interpretation as a measure of the average "regret" of sensor node i at stage k for not having selected channel y every time that channel x was selected in the past. However, in order to save energy, MRM exchanges very limited information among sensor nodes. In this case, sensor node i knows neither its own explicit utility function u_i nor the actions of other sensor nodes s_{-i}^t. Thus, it can not directly compute $u_i^t(y, s_{-i}^t)$ in Equation (2). Instead, MRM provides a method to estimate the average regret as follows:

$$\hat{R}_i^k(x,y) = \left[\frac{\sum\limits_{t \le k : s_i^t = y} \frac{p_i^t(x)}{p_i^t(y)} u_i^t(s^t) - \sum\limits_{t \le k : s_i^t = x} u_i^t(s^t)}{k} \right]^+, \tag{3}$$

where $p_i^t(x)$ represents the play probability that sensor node i selects channel $x \in C$ at stage t, and $\hat{R}_i^k(x,y)$ represents the corresponding estimated average regret. Notice that $\sum\limits_{j=1}^{c} p_i^t(j) = 1$.

Based on its estimated average regret, each sensor node i performs MRM as follows. If it selects channel x at stage k, then its probability of switching to another channel y at stage $k+1$ is approximately proportional to the average regret from x to y; with the remaining probability, the same channel x is selected again. Thus, the channels with larger regrets at current stage will be selected with larger probabilities at next stage. Then, their regrets will be reduced and thus the average regret of any channel-pair for each sensor node will be minimized over time. When channel x is selected at stage k, the play probabilities of sensor node i at stage $k+1$ are assigned as follows: for every channel $y \neq x$ and $y \in C$,

$$p_i^{k+1}(y) = \left(1 - \frac{\delta}{k^\gamma}\right) \min \left\{ \frac{\hat{R}_i^k(x,y)}{\mu}, \frac{1}{c-1} \right\} + \frac{\delta}{k^\gamma c}, \tag{4}$$

where μ is a sufficiently large number to keep the sum of play probabilities not exceed one, the last term makes a tremble probability over channels and γ should be less than 0.25 [26]; And the play probability of channel x is

$$p_i^{k+1}(x) = 1 - \sum_{w \in C : w \neq x} p_i^{k+1}(w). \tag{5}$$

Since it may consume too much resource to compute the estimated average regret directly according to Equation (3), we propose a recursive approach to compute the estimated average regret. For each sensor node i, we first define a matrix M_i^k with the following entries:

$$M_i^k(x,y) = \sum_{t \le k : s_i^t = y} \frac{p_i^t(x)}{p_i^t(y)} u_i^t(s^k) \text{ for any } x, y \in C, \tag{6}$$

and then let the matrix evolve at stage k as follows:

$$M_i^k = M_i^{k-1} + \frac{u_i^k(s^k)}{p_i^k(s_i^k)} P_i^k \times e_{s_i^k}, \tag{7}$$

where $e_x = [0\ 0\ ...\ 1\ ...\ 0]$ with 1 in the x^{th} position, and $P_i^k = [p_i^k(1)\ p_i^k(2)\ ...\ p_i^k(c)]^T$. Thus, the estimated average regret can be computed as follows:

$$\hat{R}_i^k(x,y) = \left[\frac{M_i^k(x,y) - M_i^k(x,x)}{k} \right]^+. \tag{8}$$

Based on the recursive approach, we summarize the Regret Matching based Channel Assignment algorithm (RMCA) for each sensor node in Algorithm 1 given that each stage occupies a fixed period. The algorithm is designed to be distributed for each node to maximize the utility function (1). As stated in utility function (1), the algorithm will make the each node balance between VRR maximization and ATD minimization.

To describe the long-term characteristic of RMCA, we first give the following definitions [26]:

Empirical Distribution: The empirical distribution of play z^k is a distribution on the space of joint actions $\mathscr{C} = \underbrace{C \times C \times ... \times C}_{n}$ up to stage k, and calculated as follows:

$$z^k(s) = \frac{1}{k} |\{t \le k : s^t = s\}| \text{ for every } s \in \mathscr{C}, \tag{9}$$

where $|\cdot|$ represents the number of elements of a finite set.

Correlated Equilibrium: A probability distribution ψ on the space of joint actions $\mathscr{C}$ is a correlated equilibrium of the game, if, for every $i \in P$ and every $x, y \in C$, we have

$$\sum_{s \in \mathscr{C} : s_i = y} \psi(s)[u_i(x, s_{-i}) - u_i(s)] \le 0, \tag{10}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

Algorithm 1 RMCA: Regret Matching based Channel Assignment algorithm for any node i

```
select randomly a initial channel $s_i^1$ to receive;
set $p_i^1(x)$=1/$c$ for every $x \in C$;
set $M_i^0$=$O$;
for each stage $k$=1,2,... do
    $valid\_packet$=0, $sum\_delay$=0, $wrong\_packet$=0;
    while the end of stage k is not reached do
        use $s_i^k$ to receive packets;
        if a packet is sensed then
            if the packet is completely received and its destination is sensor node $i$ then
                $valid\_packet$++;
                $sum\_delay$+=$delay\_of\_the\_packet$;
            else
                $wrong\_packet$++;
            end if
        end if
    end while
    $r_i(k)$=$\frac{valid\_packet}{valid\_packet+wrong\_packet}$;
    $d_i(k)$=$\frac{sum\_delay}{valid\_packet}$;
    use Equation (1) to compute $u_i^k(s_i^k, s_{-i}^k)$;
    use Equation (7) to update $M_i^k$;
    use Equations (8), (4) and (5) to compute $P_i^{k+1}$;
    use play probabilities $P_i^{k+1}$ to select the channel used at next stage $s_i^{k+1} \in C$;
    if $s_i^{k+1}$ != $s_i^k$ then
        broadcast $s_i^{k+1}$ to the sender of sensor node $i$ via a common channel at the end of stage $k$;
    end if
end for
```

where $u_i(s)$ represents the utility of player i at one stage under joint actions s.

We demonstrate the long-term characteristic of RMCA in the following theorem.

***Theorem** 1:* If all sensor nodes in the network adopt RMCA to play the channel assignment game:

1) for every $i \in P$ and every $x, y \in C$, $R_i^k(x, y) \to 0$ almost surely as $k \to \infty$;

2) the empirical distributions of play z^k converges almost surely to the set of correlated equilibrium of the game as $k \to \infty$.

Proof:

The basic assumptions in the Theorem of [26] include: 1) It is a repeated game; 2) each player only knows set of available choices and its own actually realized payoffs; and 3) the modified regrets are bounded. In our work, it satisfies the following conditions.

Firstly, we can easily see that the utility function of each player i at any stage k, i.e., $u_i^k(s)$, is only related to the channels chosen by all the players. This is because the interference suffered by one player is completely determined by how many players in its neighborhood are using the same channel that it uses. Also, the action space of each player is C, which is obviously a finite set. Accordingly, we have $u_i : C \to R$, and thus the setting of our game is in accordance with the model considered in [26].

Secondly, according to Equation (1), we know

$$|u_i| = |\digamma \times r_i - d_i| < \digamma + D_{max}, \tag{11}$$

where D_{max} is the largest packet delay in the network. Thus, the utility function of any player is bounded. For the MRM, we can find a constant $\mu > 2(\digamma + D_{max})(c-1)$.

Finally, according to RMCA, we have: (1) $\gamma < 0.25$; and (2) in the initial stage, the play probabilities of all channels of any player are the same, i.e., $1/c > \delta/c$.

Therefore, all the essential assumptions of Theorem 2 in [26] are satisfied in our game with RMCA, and thus its conclusion also holds in our game, i.e., the empirical distributions of play z_k converge almost surely as $k \to \infty$ to the set of correlated equilibrium of the game, and the regrets converge almost surely to zeros as $k \to \infty$. □

Theorem 1 reveals that when the channel assignment becomes stable, all sensor nodes do not regret dynamically selecting channels according to RMCA, which means that each sensor node optimally responds to the environment and to the actions of other sensor nodes. Each sensor node achieves a reasonable utility (i.e., the largest average valid receiving ratio, the shortest average packet transfer delay, and the least average interference) against the past variants of transmission flows, network topology and channels selected by other sensor nodes. As a result, these local optimal performances of all the sensor nodes lead to a considerable suboptimal network performance. Moreover, since the play probabilities assignment is based on a history of past experience, which is a kind of feedback, Theorem 1 also implies that the channel assignment by RMCA can adapt itself to the variation of the flows and network topology, and improve over time.

Notice that the sensor node only receives the packet from the sender in each stage. Extra packets for supporting the channel assignment algorithm are not needed. Thus, compared with other dynamic channel algorithms, the communication overhead can be greatly reduced. The communication complexity of the proposed algorithm is $O(n)$. In addition, as shown in the algorithm, each sensor node has to calculate the number of valid packets and the sum delay. Suppose there are m packets sent to node i at stage k. After obtaining the utility function at stage k, each node updates the matrix M_i^k and calculates the probability. Therefore, the computation complexity is $O(m + C * C)$, which guarantees that the proposed algorithm achieves low computation complexity.

In the game theoretical framework, the problem is formulated as a multi-agent multi-objective problem. Pareto optimality, is a state of allocation of resources in which it is impossible to make any one player better off without making at least one individual worse off.

***Pareto-optimal**:* A point, $s^* \in S$, is Pareto-optimal iff there does not exist another point, $s \in S$, such that $U(s^*) \leq U(s)$, and $u_i(s^*) < u_i(s)$ for at least one function [29][30]. Here, $U(s) = \{u_1(s), u_2(s), ..., u_n(s)\}$.

***Theorem** 2:* The final result of RMCA is Pareto-optimal.

Proof: Denote the final result of RMCA as $s^{k*} = (s_1^{k*}, s_2^{k*}, ...s_n^{k*})$. Suppose the final result of the repeated game is not Pareto-optimal. That is to say, there exists one final condition, i.e., $s^{k'} = (s_1^{k'}, s_2^{k'}, ...s_n^{k'})$ that satisfies the following condition,

$$u_i^{k*}(s_i^{k*}, s_{-i}^{k*}) \leq u_i^{k'}(s_i^{k'}, s_{-i}^{k'}), \ \forall i \in P \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

Since the utility function (1) is decreased with the number of interference nodes. Thus the conclusion can be obtained if and only if the interference nodes decreases, the utility function will increase. Moreover, it is rational that all channels will be selected at final condition since more fully channel utilization will result in the interference mitigation. Denote $N^* = (n_1^*, n_2^*, ...n_c^*)$ as the channel selection number set in which n_j^* means the number of nodes selecting channel j at the final result of RMCA. This condition $\sum_{i=1}^{c} n_i^* = n$ should be satisfied since all nodes are involving in the channel assignment. Similarly, $N' = (n'_1, n'_2, ...n'_c)$ denotes the channel selection number set in which n'_j means the number of nodes selecting channel j at the certain condition.

Thus, we have $n_j^* \geq n'_j, \forall j \in C$. Finally, we get $\sum_{j=1}^{c} n'_j < \sum_{j=1}^{c} n_j^* = n$. It is in conflict with the reality that all nodes are involving in the channel assignment. Therefore, there does not exist a final condition that all the utility function of the certain condition is larger or equal to that of RMCA. $\square$

4 PERFORMANCE EVALUATION

4.1 Evaluation by Simulation

We evaluate the network performance of RMCA by simulations in this section. We use OMNET++ to compare RMCA with the dynamic assignment algorithm proposed by ref. [1] (we call it "CONTROL"), MMSN and randomized CSMA multiple channel assignment algorithm by ref. [31] (we call it "randomized CSMA") in terms of packet delivery ratio and latency. The main idea of "CONTROL" is that the nodes exchange state information about messages received and degrees of estimated communication success probability, which can be regarded as feedback from the perspective of control [1]. The main idea of "randomized CSMA" is that each node randomly selects a channel to perform CSMA at each stage. We implement the even-selection of MMSN, which can be used when non-overlapping channels are not sufficient and leads to less interference [2]. In all simulations, the time parameters of sensor nodes are accordance with CC2430 [32], 50 sensor nodes are deployed randomly in a 100m×100m field, and the communication radius of each sensor node is 40m. We set the number of available non-overlapping channels to 3 (i.e., channel #1, #2, and # 3) and the stage length to 6s, and let F=0.01, μ=0.011, δ=0.01 and γ=0.24 in RMCA. For CONTROL, we set its parameters in Table 1 according to the requirements in [1]. The network routing is organized as a tree and the network topology is fixed in all simulations. Each flow is from a leaf to the root and its data rate is 56 Kbyte/min. All the sensor nodes randomly select a channel to receive packets at the initial stage of RMCA.

TABLE 1
The parameter settings of CONTROL

	settings for each channel		
parameters	channel #1	channel #2	channel #3
α_{ref}^{up}	0.8	0.4	0
α_{ref}^{down}	0.85	0.9	1
K_r^{up}	0.0375	0.075	0
$\hat{K}_r^{up}$	0.075	0.15	0
K_r^{down}	0.2	0.3	0
$\hat{K}_r^{down}$	0.3	0.4	0

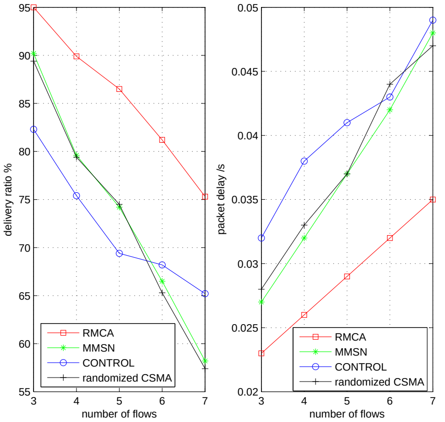


Fig. 3. Compare RMCA with CONTROL, MMSN and randomized CSMA in terms of packet delivery ratio and latency for static flows

To evaluate the ability of RMCA to handle interference, we perform the first group of simulations, in which the flows in one simulation are fixed throughout the simulation. The flows are randomly generated at the beginning of the simulation. We vary the number of flows from 3 to 7, and show the network performance of RMCA, CONTROL, MMSN and randomized CSMA in Fig. 3. Each point is the average result of 50 independent simulations. We have the following observations. Firstly, RMCA outperforms CONTROL, MMSN and randomized CSMA in terms of both delivery ratio and packet latency. The average delivery ratio of RMCA is larger than the ones of CONTROL, MMSN and randomized CSMA. We have the similar observation for the packet latency. This is because, instead of statically assigning channels, RMCA can adapt the channel assignment to the transmission flows and achieve no regret. Secondly, the advantage of RMCA over MMSN and randomized CSMA becomes more remarkable with the increase of the number of flows. When the number of flows increases, the average differences in delivery ratio between RMCA and MMSN or randomized CSMA become larger, and the average differences in packet latency get smaller. This is because that the increase of flows exacerbates the interference and provides more chance for RMCA to adapt the channel assignment among sensor nodes to achieve better network performance. Notice that the dynamic channel assignment CONTROL seems to achieve the worse performance than the static MMSN, because MMSN always involves multiple channels regardless of the network load while CONTROL has little chance to hop from the initial channel when network load is light. Actually, we can find that the performance of CONTROL gradually surpasses MMSN when the number of flows increases.

To evaluate the ability of RMCA to track the variations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

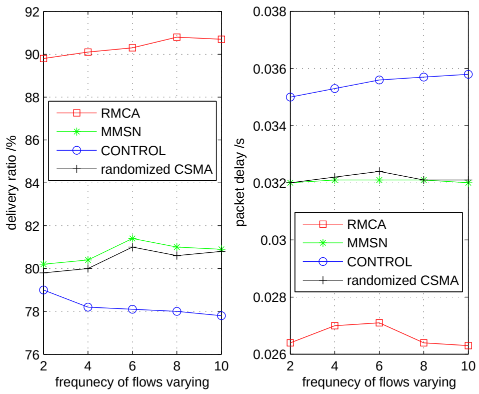


Fig. 4. Compare RMCA with CONTROL, MMSN and randomized CSMA in terms of packet delivery ratio and latency for time-variant flows

in environment, we conduct the second group of simulations, in which the flows in one simulation are time-variant. We set the number of flows to 4, and vary the frequency of flows from 2 times per simulation to 10 times per simulation. Each time the flows are regenerated randomly. The network performance of RMCA, CONTROL, MMSN and randomized CSMA is shown in Fig. 4. Each point is the average result of 50 independent simulations. We observe the following. Firstly, RMCA outperforms CONTROL, MMSN and randomized CSMA in terms of both delivery ratio and packet latency. Secondly, the performance of RMCA keeps almost the same with a slight improvement while that of CONTROL decreases, when the frequency of flows varying increases. Since both MMSN and randomized CSMA are static channel assignment methods, their network performance are not quite related to the frequency of flows varying. On the contrary, the performance of dynamic channel assignment method such as CONTROL usually degrades with the frequency increasing. Thus, the second observation implies that RMCA can track the variation of flows in some degree.

To illustrate the adaption in RMCA, we snapshot an adjusting process in Fig. 5 when the flows variation takes place at stage 3000, and compare RMCA with MMSN and randomized CSMA. The flow variation indicates that both the source node and sink node of the flows change. In this simulation, all the sensor nodes select the same channel to receive packets at the initial stage of RMCA. The delivery ratios of randomized CSMA, MMSN and randomized CSMA are decreasing after stage 3000.

Both MMSN and randomized CSMA/CA drop directly to the next stable delivery ratios, but RMCA first drops and then climbs up to the next stable delivery ratio. This is because that RMCA immediately learns the inefficiency of current channel assignment from the average regret and adapts the channel assignment to the flows variation. The observation of packet latency is similar to the delivery ratio. The large spikes of RMCA in Fig. 5 result from the tremble over channels in the adjusting process.

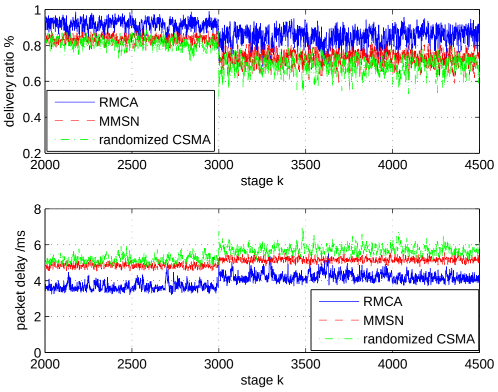


Fig. 5. An adjusting process of RMCA

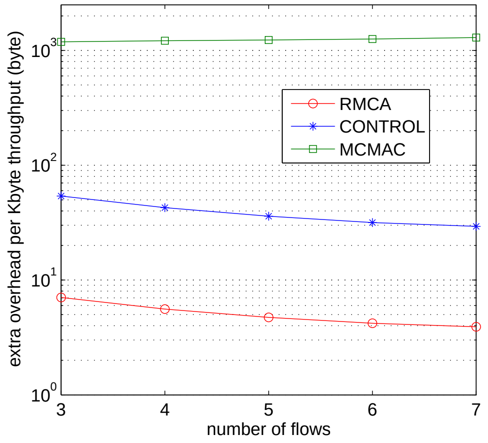


Fig. 6. Comparison in terms of extra overhead caused by channel usage negotiation

To evaluate the performance of RMCA in reducing extra overhead caused by channel usage negotiation, we conduct the third group of simulations to compare RMCA with two typical dynamic channel assignment algorithms—MCMAC [33] and CONTROL. In this group of simulations, the periods of RMCA and CONTROL are the same (i.e., the stage length 6s) since they both reconsider the channel assignment periodically. We depict the results in Fig.6, where each point is the average result of 50 independent simulations. As shown in Fig. 6, per Kbyte throughput, RMCA causes the least extra overhead about several bytes, CONTROL causes an extra overhead about tens of bytes, and MCMAC results in the most one about thousands of bytes. In addition, Fig. 6 shows that with the network load (number of flows) increasing, the extra overhead per Kbyte throughput of RMCA and CONTROL decreases while that of MCMAC increases slightly. These observations are consistance with the designs of these algorithms: RMCA exchanges coordination packets only when the channel used by sensor node changes, CONTROL exchanges coordination packets periodically, and MCMAC exchanges coordination packets for each transmission. Due to the reduction of extra overhead, both of the communication complexity and computation complexity can be reduced, which helps to save energy for WSNs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2307868, IEEE Transactions on Parallel and Distributed Systems

4.2 Evaluation by Test-bed Experiment

We further conduct a series of test-bed experiments to evaluate the performance of RMCA comparing with MMSN and randomized CSMA. In these experiments, we use Imote2 [34] sensor node, which contains the Marvell PXA271 Xscale processor at 13-416 MHz, including a Marvell wireless MMX DSP coprocessor. It is equipped with the IEEE802.15.4 radio compatible transceiver (CC2420) which supports 250kb/s data rate with 16 channels in the 2.4 GHz band. We use USB client with on-board mini-B Connector and Host Adapters to connect the Imote2 with computer. GCC in Linux is used to realize channel assignment algorithms—RMCA, MMSN and randomized CSMA.

In these experiments, as shown in Fig. 7, twelve sensor nodes are deployed in a 4-by-3 grid with each edge equaling to 90 cm. In addition, there is a base sensor node with two functions: 1) initializing the transmission process; and 2) sending beacon packets to each of the operating 12 sensor nodes periodically to synchronize them. The base sensor node uses the highest power to send beacon packets, which ensures that other nodes can hear from it. There are 6 end-to-end flows, with one source and one sink node for each flow. Consequently, all the following results are the average performance of the 6 flows. For parameter control, the deployment and topology are statically throughout the whole experiment.

The parameters of RMCA are set as follows: F=10, μ=20, δ=0.6 and γ=0.24, and the unit of ATD is millisecond. For multi-channel configuration, we use 4 channels out of the 16 channels at 2.4 GHz band according to IEEE 802.15.4; three of them are available for data packet delivery, and the fourth one for coordination packets. The common control channel is 2415 MHz; three data channels are 2430 MHz (named as channel #1), 2450 MHz (channel #2), and 2470 MHz (channel #3). These four channels are not adjacent from each other, in order to avoid the interference among adjacent channels. The time is divided into stages and each stage is composed of two sub-stages: coordination packet time and data packet time. During coordination packet time, coordination packets about the channels exchange, while during data packet time, data packets exchange. The length of data packet time is chosen to be 5 seconds. And at the initial stage for RMCA, sensor nodes randomly select a channel to receive packets.

The first group of experiments is to check the validity of RMCA. During the data packet time, the sources of 6 flows send one packet with packet length of 50 bytes every 30 ms. After the data packet time, sensor nodes enter coordination packet time. They send their new channel choice information packet to each other and will be synchronized by the beacon packet from the base node. After synchronization, sensor nodes can send their packets almost at the same time, so that we can check how effective RMCA can help adjust the channels according to interference. It should be noticed that though the sensor nodes are of the same type, their oscillating frequency and inner timer are slightly distinctive, and that is why sensor nodes do not send packet at the same time exactly. Fig. 8 shows the change of two network performance metrics (i.e., packet latency and delivery ratio) when stage k increases. From stage 1 to stage 400, average packet latency varies from 4.96 ms to 4.92 ms, and packet delivery ratio increases from 94.2% to 95.6%, after a small drop to 94%. The results indicate that sensor nodes have better performance by the adjustment of RMCA.

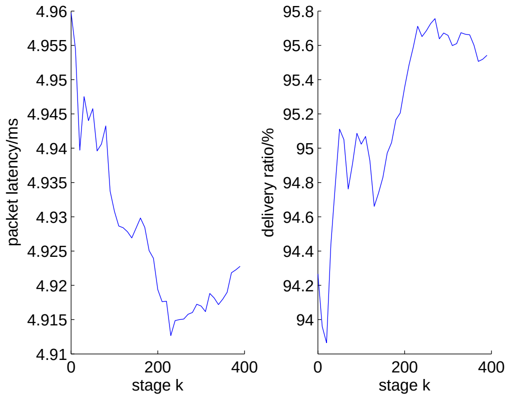


Fig. 8. Adjusting process of RMCA along with the increase of stage k

The first group of experiments can also be used to check the effectiveness of RMCA, i.e., how RMCA can adjust neighbor nodes using different channels. The topology in these experiments makes sensor nodes able to hear from part of the other nodes. Generally, one of the source nodes (node A) is sending packets, if another node (node B) can successfully receive more than 90% packets that A sends, then B is within A's transmission range. It turns out that most of sinks are within transmission range of most sources, however, there are exceptions. For deployment depicted in Fig. 7, node 12 is not within node 1's transmission range, node 2 is not within node 11's, and node 10 is not within node 3's and node 1's.

TABLE 2 provides how channels have been assigned in the experiment. For each flow, there is one main channel assigned, which takes more than 50% channel occupation in more than 400 stages. And the 3 main channels are equally occupied: channel #1 for flows 4 and 5, channel #2 for flows 1 and 6, and channel #3 for flows 2 and 3. The result just shows that RMCA can assign different channel equally. Moreover, according to RMCA, if two flows' sinks are within the transmission range of each other's source, they should try the best to avoid staying in the same channel, in order to decrease interference. For example, the sinks of flow 2 and flow 4 can hear from each other's source, and the results in TABLE 2 indicate that their channel assignment is quite complementary, which is exactly the desirable result to be achieved by using RMCA. While the sinks of flow 1 and flow 6 are not within transmission range, and they have both taken channel #2 as their main channel without bearing much interference. Therefore, the results in TABLE 2 also support the validity of RMCA in adjusting channel assignment to avoid interference.

The second group of experiments is designed to compare network performance of the three algorithms: RMCA, MMSN and randomized CSMA. In the experiments, deployment, topology and power of sensor nodes remain the same, as

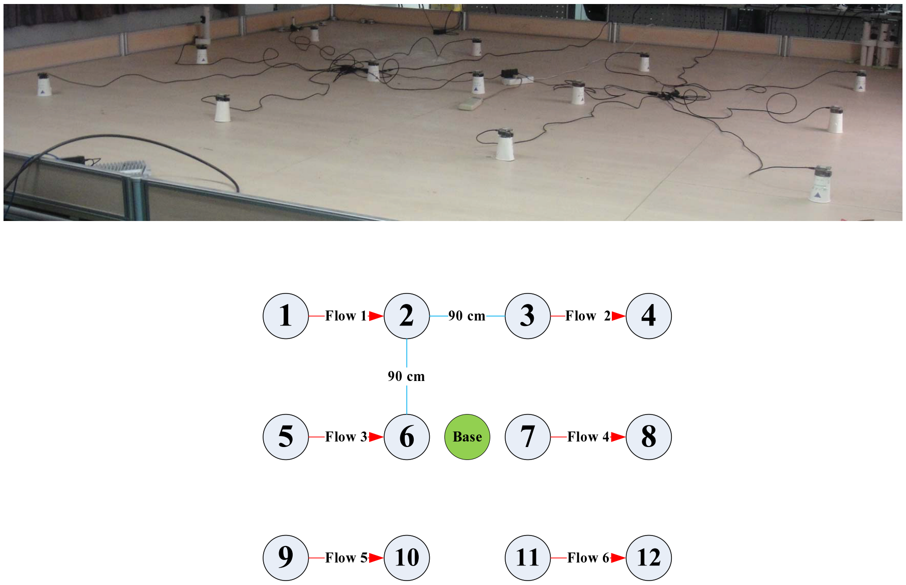


Fig. 7. Twelve Imote2 sensor nodes deployed on a 6m×6m board

TABLE 2
channel occupation percentage for different flows

	occupation percentage/%		
flow	channel #1	channel #2	channel #3
1	17.62	54.62	27.75
2	8.13	21.67	70.21
3	23.39	21.03	55.58
4	66.32	20.58	13.09
5	57.21	17.38	25.41
6	26.16	59.17	14.67

depicted in Fig. 7. For RMCA, the parameter settings are the same as described above. For MMSN, there are 3 channels used to transmit data packets and one common channel used to transmit coordination packets. These channels are exactly the same as RMCA channel settings. Randomized CSMA uses multiple channels to transmit packets for all the 12 sensor nodes and base node. For MMSN and randomized CSMA, the 12 sensor nodes also receive beacon packets from the base node periodically to synchronize themselves. In order to fairly compare with RMCA, MMSN and randomized CSMA also divide their time into stages which are composed of data packet time and coordination packet time. In data packet time, sources of flows are set to send packets of 50 bytes length, with time interval of 30ms, 50ms and 100ms, in order to compare network performance under different network loads. In the coordination packet time, twelve sensor nodes process data and receive beacon packets from the base node.

For network performance evaluation, both average packet latency and average delivery ratio are the main metrics for comparison among RMCA, MMSN and randomized CSMA. The results depicted in Fig. 9 are the average of 6 flows after 400 stages. As illustrated in Fig. 9, with time interval increasing and the network load decreasing, delivery ratio increases for all algorithms, which makes senses theoretically. It may seem confusing that delivery ratio in Fig. 9 is much higher than that in Fig. 2 with similar network load, because experiment in Section 3.2 requires packets to be sent incessantly, while hereby packets are sent with an timer that can vary slightly for different sensor nodes. In addition, Fig. 9 shows that RMCA achieves higher delivery ratio than MMSN and randomized CSMA do, and their delivery ratio differences increase when the network load gets heavier.

The comparison result for average packet latency of the 6 flows is depicted in Fig .10. Firstly, Fig .10 shows that as the network load becomes lighter, i.e., the longer time interval to send packets, the packet latency gets shorter. Actually, packet latency varying range is bounded, because back off time after collision in such condition only occupies a small part of the total time taken to send a packet. In addition, Fig .10 shows that RMCA almost outperforms MMSN and randomized CSMA.

We evaluate network performance of RMCA comparing with two existing algorithms from different perspectives, by conducting two groups of experiments based on Imote2 sensor nodes. The first group shows network using RMCA has great ability to adjust channel assignment gradually and dynamically to reduce interference in neighborhoods. Such channel adjustment makes RMCA achieve better network performance such as higher delivery ratio and shorter packet latency, and also makes RMCA flexible enough to deal with network flow and topology variation. Correspondingly, MMSN cannot adapt to flow and topology changes. In the second group, RMCA outperforms the other two algorithms for packet delivery ratio and average packet latency, when time interval to send packets varies from 30ms to 100ms. The two groups of experiments altogether indicate that RMCA is an effective channel assignment algorithm, both for network performance and dynamic network adjustment.

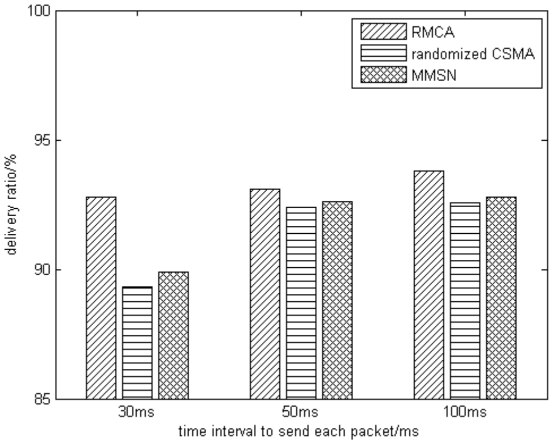


Fig. 9. Comparison of average packet delivery ratio for RMCA,MMSN and randomized CSMA

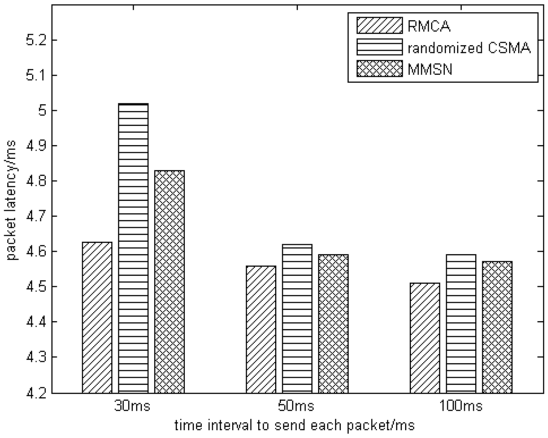


Fig. 10. Comparison of average packet latency for RMCA,MMSN and randomized CSMA

5 CONCLUSIONS

In this paper, we have studied the dynamic channel assignment in WSNs to exploit parallel transmission and reduce interference. Different from existing dynamic channel assignment protocols, we have considered the challenges posed by the multi-channel coordinations to the energy constraint of WSNs, and proposed a Regret Matching based Channel Assignment algorithm (RMCA). RMCA is highly distributed and exchanges very limited information for sensor nodes to dynamically select channels. It converges almost surely to the set of correlated equilibriums. The correlated equilibrium implies that all sensor nodes optimally respond to the environment and to the actions of other sensor nodes. The whole network can also achieve a reasonable suboptimal network performance. Moreover, RMCA can adapt the channel assignment among sensor nodes to the time-variant flows and network topology, and improve the network performance over time. Simulations of both fixed flows and time-variant flows scenarios, and test-bed experiments demonstrate that RMCA can achieve better network performance than CONTROL, MMSN and randomized CSMA in terms of delivery ratio and packet latency.

REFERENCES

[1] H.K. Le, D. Henriksson, and T.F. Abdelzaher. A practical multi-channel media access control protocol for wireless sensor networks. In *Proc. IPSN 2008*, pages 70–81, St. Louis, Missouri, USA, 2008.

[2] G. Zhou, C.D. Huang, T. Yan, T. He, J.A. Stankovic, and T.F. Adbelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In *Proc. IEEE INFOCOM 2006*, pages 1–13, Barcelona, Catalunya, Spain, 2006.

[3] Q. Yu, J. Chen, Y. Sun, Y. Fan, and X. Shen. Regret matching based channel assignment for wireless sensor networks. In *Proc. IEEE ICC 2010*, Cape Town, South Africa, 2010.

[4] Z. Tang and J.J. Garcia-Luna-Aceves. Hop reservation multiple access for multichannel packet radio networks. *Computer Communications*, 23(10):877–886, 2000.

[5] A. Raniwala and T. Chiueh. Architecture and algorithm for an ieee 802.11-based multi-channel wireless mesh network. In *Proc. IEEE INFOCOM 2005*, pages 2223– 2234, Miami, Florida, USA, 2005.

[6] J. So and N. Vaidya. Multi-channel mac for ad-hoc networks: Handling multi-channel hidden terminal using a single transceiver. In *Proc. ACM MOBIHOC 2004*, pages 222–233, Roppongi Hills, Tokyo, Japan, 2004.

[7] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar. Component based channel assignment in single radio, multi-channel ad hoc networks. In *Proc. ACM MOBICOM 2006*, pages 378–389, Los Angeles, CA, USA, 2006.

[8] Y. Bi, K.H. Liu, L.X. Cai, X. Shen, and H. Zhao. A multi-channel token ring protocol for QoS provisioning in inter-vehicle communications. *IEEE Transactions on Wireless Communications*, 8(11):5621–5631, 2009.

[9] X.D. Wang, X.R. Wang, X. Fu, G.L. Xing, and N. Jha. Flow-based real-time communication in multi-channel wireless sensor networks. In *Proc. EWSN 2009*, pages 33–52, Cork, Ireland, 2009.

[10] Y.F. Wu, J.A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Proc. IEEE INFOCOM 2008*, pages 1193–1201, Phoenix, AZ, USA, 2008.

[11] J. Chen, Q. Yu, P. Cheng, Y. Sun, Y. Fan, and X. Shen. Game theoretical approach for channel allocation in wireless sensor and actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2332–2344, October 2011.

[12] A. A. Bertossi, C. M. Pinotti, and R. B. Tan. Channel assignment with separation for interference avoidance in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(3):222–235, March 2003.

[13] Y. Wang, W. Wang, X. Li, and W. Song. Interference-aware joint routing and tdma link scheduling for static wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(12):1709–1726, December 2008.

[14] W. Wang, X. Li, and W. Song. Interference-aware joint routing and tdma link scheduling for static wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(12):1709–1726, December 2008.

[15] J.B. Zhang, G. Zhou, C.D. Huang, S.H. Son, and J.A. Stankovic. Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks. In *Proc. IEEE ICC 2007*, pages 3554–3561, Glasgow, Scotland, 2007.

[16] Y. Kim, H. Shin, and H. Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *Proc. IPSN 2008*, pages 53–63, St. Louis, Missouri, USA, 2008.

[17] F. Wu, S. Zhong, and C. Qiao. Strong-incentive, high-throughput channel assignment for noncooperative wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(12):1808–1821, December 2010.

[18] Y. Ding, K. Pongaliur, and L. Xiao. Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, 12(2):206–218, February 2013.

[19] R. E. Irwin, A. B. MacKenzie, and L. A. DaSilva. Resource-minimized channel assignment for multi-transceiver cognitive radio networks. *IEEE Journal of Selected Areas in Communications*, 31(3):442–450, March 2013.

[20] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris. Spectrum assignment in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 15(3):1108–1135, Third Quarter 2013.

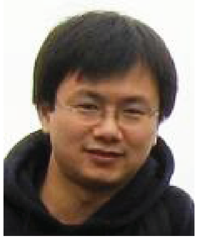
[21] D. Gzpek, B. Eraslan, and F. Alagz. Throughput satisfaction-based scheduling for cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 61(9):4079–4094, November 2012.
[22] X. Ai, V. Srinivasan, and C.K. Tham. Optimality and complexity of pure nash equilibria in the coverage game. *IEEE Journal on Selected Areas in Communications*, 26(7):1170–1182, September 2008.
[23] R. Kannan and S.S. Iyengar. Game-theoretic models for reliable path-length and energy-constrained routing with data aggregation in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1141–1150, August 2004.
[24] V. Krishnamurthy, M. Maskery, and G. Yin. Decentralized adaptive filtering algorithms for sensor activation in an unattended ground sensor network. *IEEE Transactions on Signal Processing*, 56(12):6086–6101, December 2008.
[25] M. Demirbas, X. Lu, and P. Singla. An in-network querying framework for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(8):1202–1215, August 2009.
[26] S. Hart and A. Mas-Colell. A reinforcement procedure leading to correlated equilibrium. *Economics essays: a Festschrift for Werner Hildenbrand*, pages 181–200, 2001.
[27] TI. *CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, http://focus.ti.com/lit/ds/swrs041b/swrs041b.pdf edition.
[28] X. Ling, J.W. Mark, X. Shen, and Y. Cheng. A generic framework for modeling mac protocols in wireless broadband access networks. *IEEE Network*, 21(5):28–33, 2007.
[29] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
[30] J. Jia and Q. Zhang. A non-cooperative power control game for secondary spectrum sharing. In *Proc. IEEE ICC*, pages 5933–5938, 2007.
[31] M. Doddavenkatapp, M. C. Chan, and B. Leong. Improving link quality by exploiting channel diversity in wireless sensor networks. In *Proc. IEEE Real-Time Systems Symposium (RTSS)*, pages 159–169, 2011.
[32] TI. *A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee(TM)*, http://focus.ti.com/lit/ds/symlink/cc2430.pdf edition.
[33] X. Chen, P. Han, Q.S. He, S.L. Tu, and Z.L. Chen. A multi-channel mac protocol for wireless sensor networks. In *Proc. IEEE CIT 2006*, 2006.
[34] Crossbow. *Imote2, high-performance wireless sensor network node*, http://www.xbow.com/products/produc_pdf_files/wireless_pdf/imote2_da-tasheet.pdf edition.

Jiming Chen (M'08-SM'11) received B.Sc degree and Ph.D degree both in Control Science and Engineering from Zhejiang University in 2000 and 2005, respectively. He was a visiting researcher at INRIA in 2006, National University of Singapore in 2007, and University of Waterloo from 2008 to 2010. Currently, he is a full professor with Department of control science and engineering, and the coordinator of group of Networked Sensing and Control in the State Key laboratory of Industrial Control Technology at Zhejiang University, China. His research interests are estimation and control over sensor network, sensor and actuator network, coverage and optimization in sensor network. He currently serves associate editors for several international Journals including IEEE Transactions on Industrial Electronics. He is a guest editor of IEEE Transactions on Automatic Control, Computer Communication (Elsevier), Wireless Communication and Mobile Computer (Wiley) and Journal of Network and Computer Applications (Elsevier). He also serves as a Co-chair for Ad hoc and Sensor Network Symposium, IEEE Globecom 2011, general symposia Co-Chair of ACM IWCMC 2009 and ACM IWCMC 2010, WiCON 2010 MAC track Co-Chair, IEEE MASS 2011 Publicity Co-Chair, IEEE DCOSS 2011 Publicity Co-Chair, IEEE ICDCS 2012 Publicity Co-Chair and TPC member for IEEE ICDCS 2010, IEEE MASS 2010, IEEE SECON 2011, IEEE INFOCOM 2011, IEEE INFOCOM 2012, IEEE ICDCS 2012 etc.

Qing Yu received the B.Sc degree and the M.Sc degree both in control science and engineering from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively. He was a member of the Group of Networked Sensing and Control (IIPC-nesC) in the State Key Laboratory of Industrial Control Technology, Zhejiang University. His research interests include game theoretic approaches, time synchronization and medium access control in wireless sensor networks.

Bo Chai received the B.Sc degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2010. He is a member of the Group of Networked Sensing and Control (IIPC-nesC) in the State Key Laboratory of Industrial Control Technology, Zhejiang University. His research interests include game theoretic approaches, smart grid and cognitive radio.

Youxian Sun received the Diploma from the Department of Chemical Engineering, Zhejiang University, China, in 1964. He joined the Department of Chemical Engineering, Zhejiang University, in 1964. From 1984 to 1987, he was an Alexander Von Humboldt Research Fellow and Visiting Associate Professor at University of Stuttgart, Germany. He has been a full professor at Zhejiang University since 1988. In 1995, he was elevated to an Academician of the Chinese Academy of Engineering. His current research interests include modeling, control, and optimization of complex systems, and robust control design and its application. He is author/co-author of 450 journal and conference papers. He is currently the director of the Institute of Industrial Process Control and the National Engineering Research Center of Industrial Automation, Zhejiang University. He is President of the Chinese Association of Automation, also has served as Vice-Chairman of IFAC Pulp and Paper Committee and Vice-President of China Instrument and Control Society.



Yanfei Fan received the B.Sc. degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2002, the M.Sc. degree in computer science from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2010. His current research interests include wireless network security and network coding.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the BSc (1982) degree from Dalian Maritime University (China) and the MSc (1987) and PhD (1990) degrees from Rutgers University, New Jersey, all in electrical engineering. He is a professor and University research chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. He served as the Technical Program Committee chair for IEEE VTC10, the Symposia Chair for IEEE ICC10, the tutorial chair for IEEE ICC08, the Technical Program Committee chair for IEEE Globecom07, the General CoChair for Chinacom07 and QShine06, the founding chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also served as a founding area editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, editor-in-chief for Peer-to-Peer Networking and Application, associate editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Computer Networks; and ACM/Wireless Networks, etc., and the guest editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered professional engineer of Ontario, Canada, an IEEE fellow, an Engineering Institute of Canada Fellow, and a Distinguished lecturer of IEEE Communications Society.