



# Wireless Communication on Scalable Channel Allocation and Access Scheduling for Wireless Internet-of-Things

Nalinakshi M<sup>1</sup>, Dr B Narsimha<sup>2</sup>

<sup>1</sup>Assistant Professor, CSE, Holy Mary Institute of Technology and Science, Hyderabad, India

<sup>2</sup>Professor, CSE, Holy Mary Institute of Technology and Science, Hyderabad, India

**Abstract:** Wireless communication channels are a scarce resource shared among multiple users in either scheduled or randomized fashions. We challenge a few design aspects of the widely used IEEE 802.11 MAC in wireless sensor networks (WSNs), such as the use of RTS, CTS, and ACK handshaking and the binary exponential backoff mechanisms, and argue that these key mechanisms incur high channel overhead and cannot effectively eliminate hidden terminal problems in multihop scenarios. Instead, we propose a set of efficient grid-based channel allocation and access scheduling algorithms using Latin squares, called as GAALS, for scalable WSNs with single-radio multi-channel communication capabilities. Using nodal location information and forming grids over the WSN deployment area, GAALS maps Latin squares to the grids, and dynamically assigns multiple channels to the WSN grids for channel access scheduling purposes. The fairness and scalability of GAALS are analyzed and evaluated in multiframe multihop WSNs with multi-channel capabilities. The results show that GAALS achieves much better performance than other multichannel protocols.

**key words:** Internet-of-things, wireless sensor networks, channel resource allocation, latin squares.

## I. INTRODUCTION

The Internet-of-Things (IoT) [1] is an emerging paradigm where smart objects outfitted with sensors, actuators and radio frequency identification (RFID) interact with each other and cooperate with their neighbors for sensory data collection and control message delivery. In it, wireless sensor networks (WSNs) [2], [3] are becoming a critical part of the information infrastructure in industrial control, environmental monitoring, smart cities and military operations. Beside critical research problems such as energy consumption and network capacity planning, medium access control (MAC) stands out as the pivotal factor in the performance of large scale WSNs. Protocols based on IEEE 802.11/802.15.4 standards for high/low-rate transmissions have been widely accepted in both end-users and the research community, due to simple channel access control functions, low unit cost and flexible reconfigurations. In this paper, we mainly focus on the analysis of IEEE 802.11 for high-rate transmission in WSNs. Manuscript received January 31, 2013; revised May 3, 2013; accepted May 26, 2013. Date of publication June 12, 2013; date of current version August 21, 2013. This work was supported in part by the Raytheon Company under Grant RC-42621, the UCI Graduate Dean's Dissertation Fellowship, and the Distinguished Young Scholar Program of Beijing Institute of Technology. The associate editor coordinating the review of this paper and approving it for publication was Prof. Rose Qingyang Hu. (Corresponding authors: D. Wu and C. H. Liu.) D. Wu and L. Bao are with the Department of Computer Science, University of California, Irvine, CA 92617 USA (e-mail: dwu3@ics.uci.edu; lbao@ics.uci.edu). C. H. Liu is with the Beijing Institute of Technology, Beijing, 100081 China (e-mail: chliu@ieee.org). Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>. Digital Object Identifier 10.1109/JSEN.2013.2268159 The MAC protocol in IEEE 802.11, namely DCF (distributed coordination function), is achieved by the CSMA/CA mechanisms on listening the medium, leveraging a random exponential backoff algorithm over a shared wireless channel. However with a single channel in mission critical applications, WSNs based on IEEE 802.11 protocol suite are unable to deliver reliable and timely communication with high data rate due to limited bandwidth, increasing channel overheads and severe interference caused by hidden terminals [4]. On the other hand, current WSN hardware, such as Micaz and Telos that use the CC2420 radio, already provide multiple frequencies. So it is imperative to design multi-channel based protocols in WSNs to improve network throughput and provide reliable and timely communication services. In this paper, we focus on the channel resource sharing under two research thrusts in WSNs with single-radio multichannel (SRMC) communication capability: (a) the channel resource allocation problem, which asks for dynamic and adaptive conflict-free channel assignments to individual sensors or clusters in the macro-time scale; (b) the multiple channel access control problem, which asks for collision-free packet transmissions in the micro-time scale. Such channel resource coordination functions are especially challenging due to inherent distributed nature of large scale WSNs. The channel allocation and



access scheduling problems have been extensively studied in the literature. MC-LMAC proposed a multi-channel lightweight MAC protocol for WSNs. Time is slotted and each sensor is assigned the control over a time slot to transmit on a particular channel. MMSN assigned frequencies to the sensors of an 1-hop neighborhood, with sensors learning their neighbors' frequencies. A sensor needs to use the destination's frequency when transmitting, and its own frequency when receiving. MMAC [8] used the beacon messages to synchronize sensors over a common channel, and to negotiate the channel allocations within the Ad hoc Traffic Indication Message (ATIM) window. Although beneficial for channel throughput, switching among multiple channels may result in disconnections in wireless networks and then impact the network coverage and scalability [9]. A distributed clustering algorithm [10] was proposed to determine suitable cluster sizes depending on the hop distance to the data sink, while achieving approximate equalization of sensor lifetimes and reduced energy consumption levels. [11] grouped the sensors into clusters of unequal sizes and designed a greedy geographic and energy-aware routing protocol for inter-cluster communication. In this paper, we propose a set of algorithms, called "GAALS", Grid-based channel Allocation and Access scheduling using Latin Squares. It applies Latin squares [12] in 1530-437X © 2013 IEEE WU et al.: SCALABLE CHANNEL ALLOCATION AND ACCESS SCHEDULING FOR WIRELESS IoT 3597 two time scales — macro-time scale for channel assignment, and micro-time for channel access. GAALS provides fair, prompt and efficiency channel access under various network traffic scenarios due to the deterministic feature of Latin squares. It also saves energy by reducing communication overhead from RTS/CTS/ACK-free transmission. In addition, the grid structure in GAALS is capable of adaptively achieving network connectivity and coverage for scalable WSNs.

The rest of paper is organized as follows.

Section II introduces the Latin squares and network model.

Section III describes the channel access scheduling in micro-time scale.

Section IV presents the channel allocation in macro-time scale.

Section V addresses the grid-based coverage.

Section VI shows the evaluation results and

Section VII concludes the paper

## II. FUNDAMENTALS

A. Latin Squares Some useful knowledge of Latin squares in the development of our algorithm is introduced as follows.

Definition 1: A Latin square of order  $n$  is an  $n \times n$  square matrix that consists of  $n$  symbols  $\{1, 2, \dots, n\}$ , in which the symbols of each row and column are also distinct. For instance, the following matrices  $A$  and  $B$  are two Latin squares of order 4.  $A = \begin{bmatrix} 1234 & 2143 & 3412 & 4321 \\ 2143 & 3412 & 4321 & 1234 \\ 3412 & 4321 & 1234 & 2143 \\ 4321 & 1234 & 2143 & 3412 \end{bmatrix}$   $B = \begin{bmatrix} 4123 & 3214 & 1432 & 2341 \\ 3214 & 4123 & 2341 & 4123 \\ 1432 & 2341 & 4123 & 3214 \\ 2341 & 4123 & 3214 & 1432 \end{bmatrix}$ .

Definition 2: Two distinct  $n \times n$  Latin squares  $A = (a_{i,j})$  and  $B = (b_{i,j})$ , where  $a_{i,j}$  and  $b_{i,j} \in \{1, 2, \dots, n\}$ , are said to be mutually orthogonal Latin squares (MOLS) if the  $n^2$  ordered pairs  $(a_{i,j}, b_{i,j})$  are all different. Thus, the two  $4 \times 4$  Latin squares  $A$  and  $B$  shown above are mutually orthogonal. This concept can be generalized to entire families of Latin squares of the same order. A set of Latin squares of order  $n$  are a mutually orthogonal family of Latin squares if every pair in the set is orthogonal. Theorem 1: If  $n > 1$  and  $n = pk$ , where  $p$  is a prime number and  $k$  is a positive integer, then there is a complete MLOS family of order  $n$ . Proof: The proof can be found in [12].

Definition 3: Extending the Latin square concept, a  $k$ -dimensional Latin hypercube of order  $n$  is a  $k$ -dimensional array  $H_k = [h_{i_1, i_2, \dots, i_k}]$  in which each row is a permutation of symbols  $1, 2, \dots, n$ , i.e., Latin cube is a 3-dimensional Latin hypercube [13].

Although we also apply Latin hypercube later, our discussions are mostly based on Latin squares. There are many algorithms to generating Latin squares [12], [13]. The one that we adopt is based on the modulo multiplication group theory because of its adaptiveness and simplicity of computations. The Latin square generation algorithm is specified as follows: Suppose  $n + 1$  is a prime number, and vectors  $X = [x_i]$  and  $Y = [y_i]$  ( $i = 1, 2, \dots, n$ ) are two permutation arrays of symbol set  $\{1, 2, \dots, n\}$ , then a Latin square  $L_{n \times n} = [l_{i,j}]$  of order  $n$  can be generated by multiplying the transpose of vector  $X$ , denoted as  $X^T$ , and vector  $Y$  using the modulo multiplication operation:  $L_{n \times n} = X^T \cdot Y$ , where  $l_{i,j} = x_i \cdot y_j \pmod{(n + 1)}$ ,  $i, j \in \{1, 2, \dots, n\}$ .

(1) B. Network Model We consider a WSN where each sensor knows enough location information of itself and its neighbors [14]. In addition, we assume that each sensor is equipped with a single radio and an omni-directional antenna, capable of switching between multiple channels, and each radio only works in half-duplex mode, i.e. may only receive or transmit at a time, but not both. In WSNs, we model the effective ranges of radio communications in three categories, namely: the transmission range  $R_{tx}$ , the carrier sensing range  $R_{cs}$  and interference range  $R_i$  [15].  $R_{tx}$  is mainly determined by transmission power and radio propagation properties (i.e., attenuation), and represents the range within which a packet is successfully received if no interference from other radios.  $R_{cs}$  is the range that a receiver detects the carrier as busy if there is a transmission within  $R_{cs}$ .  $R_i$  is the range outside the carrier sensing range, and within which the radio signal appears as noise at the receivers.

The network topology of a WSN is represented as an undirected graph  $G = \{V, E\}$ , in which  $V$  is the set of sensors, each assigned a unique ID number, and  $E \subseteq V \times V$  is the set of links between two sensors in  $V$  if they are within the transmission range  $R_{tx}$  of each other. The end-points of a link in  $E$  are called the one-hop neighbor of each other, and two sensors



without a link between them but sharing a common one-hop neighbor are called two-hop neighbors of each other. We assume the WSN can be time synchronized in this paper.

### III. GAALS: CHANNEL ACCESS SCHEDULING

We discuss the multiple channel access control problem in the micro-time scale, and present a collision-free packet transmission scheduling using mutually orthogonal Latin squares (MOLS) in this section.

A. BEB Based Channel Access Scheduling In relatively small wireless networks like WLANs, where nodes can carrier sense each other's transmission, CSMA works most efficiently. IEEE 802.11 DCF was based on CSMA/CA enhancements using RTS, CTS, ACK control messages, various inter-frame spacings and binary exponential backoff (BEB) mechanisms. The BEB mechanisms are applied in IEEE 802.11 DCF as follows. Whenever a node has packets to send, it carriersenses the channel to see if the channel is busy in case that any other nodes are already sending packets. If yes, the node keeps carrier-sensing the channel until the channel becomes idle, i.e., no carrier is detected. Then, instead of immediately starting transmission of its own packet, the node waits for a random amount of time, measured in 20 micro-second time slots. The number of time slots to wait for is calculated via the BEB algorithm, which selects a backoff interval  $T_{bo}$  randomly from a collision window (CW)  $[0, CW)$ , and CW is usually 3598 IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013 Fig. 1. Channel access backoff mechanism according to Latin Squares. set to  $CW_{min} = 32$  at the beginning. The node keeps carriersensing the channel while counting down the backoff interval  $T_{bo}$  in each time slot. If  $T_{bo}$  becomes 0 and the carrier is still idle, the node may start transmitting its own packet. In case collision happens, the BEB algorithm doubles the collision window CW, and repeats the backoff process. Although IEEE 802.11 DCF works in small networks, the BEB mechanism does not eliminate packet collisions because multiple transmitting nodes may choose the same backoff interval due to the randomness in the BEB algorithm.

B. Fair Access Scheduling Using MOLS In order to avoid choosing the same backoff intervals among multiple transmitters in distributed environments, we apply mutually orthogonal Latin squares (MOLSs) to calculate the backoff intervals  $T_{bo}$  for channel access in GAALS. The channel access scheduling is achieved by assigning specific sensors to transmit during specific slots, e.g., TDMA. Each TDMA frame consists of  $n$  subframes, each of which has  $n$  time slots. Each slot is  $s$  seconds long. Therefore, each frame consists of  $n^2$  slots. Suppose the size of each sensor packet is fixed (or upper bounded) at  $d$  bits, and each sensor has a link layer bandwidth of  $L$  bits per second, then  $s \geq \max(d/L, 1/T_{bo})$ . We have a family of  $m$  MOLSs. Each sensor  $i$  is assigned a particular symbol from a particular Latin square. We write this assignment function as  $LS(i)$ , with  $LS(i) : i \rightarrow n \times m$ . In our solution, the rows of a Latin square are assigned to different backoff intervals  $T_{bo}$ , and the columns to time slots. For instance, consider a small network with four sensors A, B, C and D in Fig. 1. We could assign sensor A to symbol 1, B to symbol 2, C to symbol 3 and D to symbol 4 in a common Latin square chosen from the MOLS family. The time is structured by periodic time frames of period  $T = 8$  milliseconds (ms), which are further divided into 4 time slots of period  $t_i = 2$  ms,  $i = 1, 2, 3, 4$ . In each time slot, the backoff intervals of each sensor for channel access (represented by the yellow hexagons) is equal to the value of the row index where its unique Latin square symbol is located. Therefore, the backoff intervals is 1 for sensor A transmits in the first time slot, 4 in the second time slot, 3 in the third time slot and 2 in the fourth time slot. All transmission schedules are repeated every  $n^2$  seconds. In addition, by Definition 2, our scheme also works if sensors are assigned symbols from different Latin squares in the MOLS family. It is especially useful in on-demand distributed environment when the number of active sensors is more than Latin square order  $n$ . Using MOLS for channel access scheduling, GAALS can be implemented by simply replacing the backoff computation module in IEEE 802.11 DCF. In practice, the duration of a time frame can be around tens of milliseconds, and a time slot can last a few milliseconds in Fig. 1. As we see, the channel backoff mechanism is synchronized based on the end of the previous busy period. In regular channel access, if the channel has been idle for a long time, the sensor should start accessing the channel by sending a busy signal burst first to synchronize the channel clock.

C. Analysis of MOLS Scheduling Theorem 2: Two sensors are conflict-free if they are assigned different symbols from the same Latin square. Proof: Two sensors can be in conflict only if they are in radio range and transmit during the same time slot. However, since both sensors determine their transmission backoff intervals based on the occurrence of their specific symbols from the same Latin square, then by Definition 1 their symbols and therefore their transmissions will be collision-free. Theorem 3: If two sensors are in radio range and assigned symbols from different orthogonal latin squares, there is at most one collision for them during the same time slot. Proof: This follows directly from Definition 2, since each ordered symbol pair  $(a, b)$  from two different latin squares, where  $a$  and  $b \in \{1, 2, \dots, n\}$ , can only occur once and only once. It implies that the two sensors will only have one chance for collision. Definition 4: Two sensors are conflicted if they are in radio range and are assigned the same symbol from the same Latin square. According to above theorems and definitions, for a specific sensor  $r$  surrounded by  $N$  other sensors in radio range, a neighboring sensor  $t$  can successfully transmit packets to it if sensor  $r$  and other  $N - 1$  neighbors do not transmit in the same slot by the same backoff as sensor  $t$ . Suppose  $X$  sensors are transmitting packets in the same slot as sensor  $t$ , then the probability  $P$  of  $i$  packets transmitted as sensor  $t$  is:  $P(X = i) = \binom{N-i}{i} \rho^i (1-\rho)^{N-i}$  (2) where  $\rho$  is the utilization factor,  $B$  is the number of backoff choices,  $n$  is the order of the chosen MOLS,  $\min(B, n)$  is the number of transmission slots in each subframe for sensors,  $i = 0, 1, \dots, N$ . For access scheduling using



MOLS, let  $Y$  out of these  $X$  active sensors decide their transmission schedules from the same Latin square as sensor  $t$ . We have:  $P(Y = j | X = i) = \frac{n - i - j}{S - n - i - j} \frac{S - 1 - i}{S - 1 - i}$ , (3) where  $S = n \times m$  is the total number of symbol patterns in the MOLS family,  $m$  is the size of the MOLS family,  $i = 0, 1, \dots, N$  and  $j = 0, 1, \dots, i$ . These  $Y$  sensors will not cause any collisions at sensor  $r$ , but the other  $X - Y$  neighbors may collide with the transmission from sensor  $t$  to sensor  $r$  by the same backoff. Hence, the WU et al.: SCALABLE CHANNEL ALLOCATION AND ACCESS SCHEDULING FOR WIRELESS IoT 3599 probability  $P_{tr}$  of a successful packet transmission from sensor  $t$  to sensor  $r$  is:  $P_{tr} = \frac{N - i}{N} \frac{i - j}{i} \frac{P(X = i, Y = j) \times P(\text{success} | X = i, Y = j)}{D} \frac{i - j}{N - i - j} \frac{S - n - i - j}{S - 1 - i} \frac{p_i}{(1 - p)^{N - i}} \times \min(B, n) \frac{i}{\min(B, n) - 1} \frac{i - j}{i}$  (4) Note that if a sensor is conflicted it still will be able to have collision free transmissions if none of the other sensors that it conflicts with have a packet to transmit at the same time. To avoid the situation in Definition 4 and maximize the conditions that make Theorem 1 true, we define a greedy assignment method. That is, each sensor periodically broadcasts its symbol and Latin square assignment. Sensors choose an unused symbol in the most used Latin square based on the broadcasts they receive from their neighbors during the learning period. A sensor declares itself disabled, if the broadcasts it receives show that any assignment would result in it being conflicted.

#### IV. GAALS: SCALABLE CHANNEL ALLOCATION

In this section, we address the efficient clustering and channel resource allocation in WSNs, which ask for dynamic and adaptive conflict-free channel assignments to sensors or clusters in the macro-time scale. A. RTS/CTS/ACK Based Data Transmission IEEE 802.11 has implemented several collision avoidance mechanisms, such as different inter-frame spacing (IFS) to prioritize control and data packets, NAV (network allocation vector) to reserve the channel, and RTS/CTS to avoid hidden terminal problems. However, due to the different impact ranges of transmission, carrier sensing and interference, RTS/CTS and ACK mechanisms occasionally fail to avoid hidden terminal problems in multi-hop WSNs [15]. In addition, the CTS packets might not be correctly received all the time in the real world due to either the CTS packets are unable to be decoded at remote hidden terminals or the CTS packets are collided with other packets at the hidden terminals [16]. Furthermore, the energy consumption in WSNs using CSMA/CS includes four main causes: collision, overhearing, idle hearing and overhead. Especially, MAC protocol overhead results in energy waste when transmitting and receiving control packets. The RTS and CTS control packets used in some protocols do not carry any useful data to applications, but their transmission consumes energy. For example, the exchange of RTS/CTS induces high overheads in the range of 40% to 75% of the channel capacity, because data frames are typically very small in WSNs [17]. Therefore, we can not rely on RTS/CTS and ACK mechanisms in GAALS for collision avoidance. Instead, a grid-based Fig. 2. Network grid formation. spatial division multiple access scheme has to be applied in WSNs for channel allocation using Latin squares, and saving energy using RTS/CTS/ACK-free transmission. In addition, CSMA scheme has to be limited within individual grids. B. Network Grid Formation Similar to the channel allocation scheme in cellular networks, we allocate channel resources in a grid formation. The left-hand side of Fig. 2 shows the 7-cell clustering scheme, which is applied in channel reuse by the cells with the same number assignments. Similarly, we organize WSNs by the grid and the grid cluster. For simplicity, we choose a square grid formation scheme with 9-grid clusters as shown on the right-hand side of Fig. 2. Given multiple channels in a multi-hop wireless network, we will apply the same channel assignment results to all the 9-grid clusters. We assume the sensors have the same transmission range  $R_t \times$ . The carrier sensing range  $R_{cs}$  is an adjustable parameter in the physical layer modules, and is usually set to 2 to 2.78 times  $R_t \times$  [15]. Therefore, the two-hop distance in WSNs is a good approximation of the carrier sensing range, and sensor activation scheduling usually requires all neighbors of a sensor within two hops be silent when the sensor transmits. Similar to cellular technology, each grid edge is set to be twice the transmission range so that the distance between two nearby grids with the same channel allocation is about three times the side length. Such a separation of the same channel allocation guarantees interference freedom for concurrent transmissions over the same channel. C. Channel Allocation In WSNs, especially those based on IEEE 802.11 standards, we have limited number of channels for conflict-free assignment to the wireless grids. These limited amount of channels have to be time-shared among all the grids of the network. There are two methods to assign channels, namely the one based on Latin square and the other based on Latin cube. Fig. 3(a) shows a  $10 \times 10$  Latin square for the 9-grid clustering case, where the Latin square is generated by order  $n = 10$  in Eq. (1). Suppose there are 5 available channels in this example. In order to assign these channels into each 9-grid cluster, we map the grids to the rows and the time slots to the columns in the Latin square. For each 9-grid cluster, channels 1, 2,  $\dots$ , 5 will be allocated to 5 grids in each cluster that have been assigned with Latin square symbols 1, 2,  $\dots$ , 5 in every time slot, correspondingly. For instance in Fig. 3(a), in time slot  $t_1$ , grids 3600 IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013 Fig. 3. (a) Latin square allocation, and (b) Latin cube allocation. Fig. 4. Dynamic Latin squares scheduling in different time frames.  $G_1, G_2, \dots$  and  $G_5$  are allocated with channel  $C_1, C_2, \dots, C_5$ , respectively. However, in some time slots of the  $10 \times 10$  Latin square, the required Latin square symbols may appear on the 10-th row, where  $G_{10}$  is not available because each cluster only have 9 grids. In this case, we swap the required Latin square symbols on the 10-th row with the Latin square symbol 10 located on other rows, so as to map the corresponding channels to the existing grids. For example, in time slot  $t_2$ , we swap Latin square symbol 1 on the 10-th row with Latin square symbol 10 on the 9-th row, so that grid  $G_9$  is allocated with channel 1 and all 5 channels are fully utilized. Different from the





channel access scheduling in micro-time scale using MOLS in Section III, the macro-time solution of scheduling Latin squares for channel allocation in each grid cluster is based on Eq. (1), which regenerates Latin squares in different time frames by creating different vectors for modulo multiplication operation. As shown in Fig. 4 for a 4-grids case, 4 different Latin squares are generated for channel allocation in 4 different time frames by the method introduced in Eq. (1). Lemma 1: The modulo multiplication operation can generate square matrix to distribute distinct and fair channel assignments over the grids. Proof: In order to prove channel assignments are distinct, we just have to show that the symbols in each column of the square matrix  $L_{n \times n}$  generated in Eq. (1) are distinct. The distinctness of the symbols in each row follows the same proof. That is, we need to show that the following two elements are distinct,  $li_{1,j} = li_{2,j}$ ,  $\forall i_1, i_2, j \in \{1, 2, \dots, n\}$ , and  $i_1 \neq i_2$ . These two elements are actually calculated as follows,  $li_{1,j} = xi_1 \cdot y_j \bmod (n+1)$ ,  $li_{2,j} = xi_2 \cdot y_j \bmod (n+1)$ . Because  $n+1$  is a prime number and  $|xi_1 - xi_2| < n+1$ ,  $y_j < n+1$ , therefore  $li_{1,j} - li_{2,j} = (xi_1 - xi_2) \cdot y_j \bmod (n+1) = 0$ , That is,  $li_{1,j} = li_{2,j}$ . Meanwhile, each permutation of vector X or Y in Eq. (1) creates a different Latin square. Therefore, by simply permuting vector X or Y or both, we can generate a sequence of different Latin squares of order n for fair channel assignments. Fig. 3(b) shows another way of allocating the 5 channels to 5 grids of each 9-grid cluster using Latin cubes. The Latin cube used in Fig. 3(b) is generated by order  $n = 10$  too. The grids, channels and time slots are mapped to the columns, rows and layers of the Latin cube. In each time slot, specific channel is allocated to individual grid if the grid has a symbol 10 on the corresponding rows. For example, in time slot  $t_1$ , grid G1 is allocated with channel C3, and grid G2 is allocated with channel C2. Similar to Latin square based channel allocation, the symbol 10 may appear on the last column, which maps to none of the grids in our scenario. In this case, we shift the channel mapping by skipping the rows that give no channel allocations. For instance, because row 4 does not map symbol 10 to any of the existing grids, the channel mapping skips over row 4 and maps channel 4 to row 5 in Fig. 3(b). In other cases, the number of channels may be more than that of grids in a grid cluster. It can be resolved by taking a modulo operation on the number of channels by the number of grids, and the remainder number of channels still follows the same channel allocation algorithms as illustrated in Fig. 3(a) and (b). As for the other channels which are multiples of the number of grids, we evenly allocate them over all the grids. However, the utilization of multiple channels with each grid is a separate issue, not addressed in this paper. Once a grid is assigned a channel, the sensors inside the grid can communicate using the channel access protocol, specified in Section III. While sensors that are inside grids without any channel allocated have to remain silent during the corresponding time slots. D. Grid Shifting Note that time slot for channel allocation is different from the time slot for channel access. In order to explain the temporal relationship between the time units for channel access scheduling and channel allocation, we introduce the notation:  $t_{acc}$  and  $T_{acc}$  as the time slot and time frame durations for channel access schedules, respectively, and  $t_{all}$  and  $T_{all}$  for channel allocation schedules. In addition, denote the size of the Latin square for channel access scheduling as  $L_{acc}$ , and that for channel allocation as  $L_{all}$ . Then the relations between these components are:  $T_{acc} = t_{acc} \cdot L_{acc}$ ,  $t_{all} = 4 \cdot T_{acc}$ , (5) WU et al.: SCALABLE CHANNEL ALLOCATION AND ACCESS SCHEDULING FOR WIRELESS IoT 3601 Fig. 5. Four-step grid shifting for channel allocations. Fig. 6. Network connection activation in one of the grid shifting steps. and  $T_{all} = t_{all} \cdot L_{all} = 4 \cdot t_{acc} \cdot L_{acc} \cdot L_{all}$ . (6) The reason that each channel allocation time slot includes four access time frames  $T_{acc}$  in Eq. (5) is due to the grid shifting operation to maintain network connectivity. This is because when different channels are allocated to the grids in the virtual grid based sensor network, sensors that belong to different grids and channels cannot communicate with each other, thus potentially breaking the existing network connections. In order to guarantee the network connectivity, we have to apply a grid shifting algorithm so as to ensure that every pair of neighbor sensors have a chance to stay in the same channel to communicate. We schedule channel access periods in four steps, as shown in Fig. 5. In each step, the grid coordinate origin shifts in four directions consecutively, which are right, down, left and up directions, respectively. During each step, the grid only shifts half of the grid size. Because sensors may belong to different grids of a grid cluster in the four steps, every sensor needs to re-calculate its channel number in each step. In practice when we build a WSN using IEEE 802.11 devices, a reasonable configuration about the temporal relations between channel access and channel allocation periods is to set  $t_{acc} = 2$  ms,  $T_{acc} = 40$  ms,  $t_{all} = 160$  ms and  $T_{all} = 1.6$  s, when the Latin square size for channel access is  $L_{acc} = 10$ , and  $L_{all} = 10$ . That is, the longest interval for activating a grid is up to 1.6 seconds using our grid-based channel allocation scheme in the worse case scenario. Lemma 2: The four-step grid shifting algorithm guarantees that each and every network connection can be activated on the same channel. Proof: This lemma originated from a problem in multichannel access scheduling problem, where neighboring sensors may not be able to communicate with each other because of different channel allocations. We prove the lemma using a heuristic by throwing “sticks” into the grids, as shown in Fig. 6. The length of a “stick” is the transmission range of the sensors in WSNs. Because each grid in Fig. 6 represents an area in which all sensors have the same channel assignment, the fact that the “stick” crosses none of the grid boundaries proves that the end points of the “stick” can communicate over the same channel within one of the four steps in the grid shifting algorithm. In Section IV-B, we have designed the grid formation such that the edge length of each grid is twice the transmission range of sensors. In addition, the grid shifting algorithm has applied grid shifting steps by half of the grid size, which is exactly the transmission range of sensor. Therefore, we have these intermediate conclusions. • The grids are symmetric, which means that the analysis about any grid applies to all other grids. • The “stick”, representing the spatial one-hop neighbor relation, has only three ways of



placement in a grid: completed contained in a grid, crossing boundary of two adjacent grids, and crossing two boundaries of adjacent grids, as shown in Fig. 6. Figure 6 shows that all the three possible placements of “sticks” can be contained within a grid in one of the fourstep grid shifts. Otherwise, the length of stick would have to be longer than half the grid size. E. Channel Transmission Scheduling Because the channels are allocated and accessed according to grids in GAALS, the neighbors of a sensor may not always be on the same channel as the sensor. Therefore, the sensor needs to carefully schedule in which time slot(s) to transmit a packet, and how many times the packet has to be transmitted. If the sensor transmits a broadcast packet, it has to schedule the transmission of the broadcast packet for up to 4 times in order to reach each and every one-hop neighbor. If the sensor transmits a unicast packet, it has to schedule the packet transmission while the sensor and its one-hop destination are located within the same grid. • If the packet is a unicast packet, the packet only has to be sent once to the correct receiving one-hop neighbor in the corresponding time slot, when the two neighbors are in the same grid point. • Otherwise, the packet has to be duplicated in multiple time slots so that the neighbors would be in the corresponding time slot on the same channel for receiving the packet. One way to implement such multi-channel packet transmission scheduling is to maintain a separate packet buffer for each channel. Data packets are placed in a channel buffer on a sensor if the one-hop neighbor will join the the same channel as the sensor itself in the nearest future.

## V. GAALS: ADAPTIVE GRID-BASED COVERAGE

One of the effects of the greedy method introduced in Section III-C is that part of the sensing area may be left uncovered due to sensors declaring themselves disabled to avoid transmission collision. In addition, though the grid edge is set to be twice the transmission range in Section IV-B to reduce interference in two-hop distance, transmission collision still can happen in some grids, since the sensors in practical deployment are heterogeneous with various transmission range and carrier sensing range. We present here a way to estimate the expected number of conflicted sensors so that people can use it to adjust grid structure for adaptive coverage in WSNs. 3602 IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013 Assume that  $K \geq 2$  sensors are deployed over a  $D \times D$  grid, by a uniform random distribution. The average carrier sensing range  $R_{cs}$  is 1 times the average transmission range  $R_t$ . Let  $C(K)$  be the average number of neighbors each sensor has. An approximation of  $C(K)$  is  $C(K) = \pi R_{cs}^2 / D^2 (K - 1) = \pi R_t^2 / D^2 (K - 1)$ . (7) Suppose each sensor  $i$  is assigned a Latin square symbol given by  $LS(i)$ . Let  $n$  be the order of the Latin square. Note that this is the same as the number of symbols. Let  $M$  be the number of MOLS. Define a random variable  $X_{i,j}$  as  $X_{i,j} = 1$ , if  $LS(i) = LS(j)$  and  $i \neq j$ , 0, otherwise. (8) where the value of  $X_{i,j}$  is 1 if  $i$  and  $j$  have the same symbol assignment. Let  $n \times M$ , then for any arbitrary pair of sensors  $i, j, i \neq j$ , the probability  $P(X_{i,j} = 1) = 1/M$ . Since sensors can only be conflicted if they are in sensing range of each other, for each disk with radius  $R_{cs}$  within the grid, let  $E(S_k)$  be the expected number of conflicted sensors. Then  $E(S_k) = C(k) \sum_{i=2}^k \sum_{j=1}^{i-1} P(X_{i,j} = 1) = \pi R_t^2 / D^2 (K - 1) (K - 2)$ . (9) It provides a heuristic for determining the parameters  $K, R_t$  and  $D$ . A larger value of  $R_t$  results in a smaller number of conflicted sensors but a larger time delay for each frame. Further, if  $R_t$  is set too low, the network will be partitioned. If it is set too high, there will be many conflicted neighbors. By network scaling law results, we can determine reasonable  $R_t$ , and then apply the power control to adjust its value in WSNs. For instance, the work in [18] showed that with random placement of reliable nodes, an network will remain connected if  $R_t \sim \log K / D$ , where  $D$  is normalized to 1. Substituting this result into Eq. (9) yields:  $E(S_k) = 1 - 3 / (K - 2) \pi R_t^2 / D^2$ , (10) which can be used as a guideline to find the balance between performance, coverage, and network partitioning. Without loss of generality, assume that the sensing range is equal to the radio range:  $R_{cs} = R_t = R$ , we can express  $R$  as a function of the grid dimension  $D$  with Eq. (9). Fig. 7 provides an upper bound on the coverage area given  $R/D$  ratios and a fixed numbers of sensors. For sizes of  $D$  between 100 and 1000, in increments of 100, the figure shows that it is possible to cover all of the area with reasonable  $R/D$  ratios, because of the close correlation in the resulting coverage and collision percentages for fixed numbers of sensors, assignment types, and loads.

## VI. PERFORMANCE EVALUATION

To evaluate the effectiveness of GAALS mechanism in more practical scenarios, we implemented GAALS in NCTUns v5.0 and compared with other existing solutions in two simulation scenarios. The first was in a fully connected network scenarios with multiple flows, and the second was in a multi-hop multiflow networks with SRMC (single radio multiple channel) communication capabilities. We used AODV as the routing protocol to find the end-to-end route. The transmission range of each sensor was set to 50 m and the carrier sense range was 110 m by tuning the radio reception threshold values. The packet size of the transmission traffic was set to 1024 bytes, and the link bandwidth for each channel was 11 Mbps. To fully demonstrate the multi-channel communication capacities under different channel access scheduling protocols, we assumed that nine non-overlapping channels were available for SRMC operations. A sensor may be the source and destination for multiple flows. Each simulation was carried out for a duration of 30 seconds, and the delay, jitter and network throughput performance metrics were collected in each run. A. Fully Connected Network



In fully connected networks, we placed 10 sensors in a  $200\text{ m} \times 200\text{ m}$  area and created 3 flows between 3 random pairs of sensors. Then all sensors are within each other's transmission range and transmit data within a single hop. The Latin square order and symbol were assigned using MOLS by our greedy method. Such a scenario is rather to examine the performance of channel access scheduling protocols by comparing GAALS with IEEE 802.11 DCF protocol. Fig. 8 shows the average end-to-end traffic performance regarding the delay, delay jitter and overall throughput.

Since GAALS adopts the CSMA scheme as well as a deterministic backoff function based on Latin square assignments, and removes the control overhead on RTS/CTS and ACK frames, it provides about 25% higher throughput, 25% lower average delay and 25% lower delay jitters than IEEE 802.11 DCF under different network loads in a 10-sensor WSN. B. Multi-Hop Multi-Flow Network In order to evaluate the functionalities and performance of GAALS in multi-hop multi-channel WSNs, we ran the same simulation scenarios, which was to place  $10 \times 10 = 100$  sensors over an area of size  $1400\text{ m} \times 1400\text{ m}$ , and to use 9 channels and create 30 flows in the multi-hop wireless networks. After the delay, jitter and throughput metrics were collected, we compared the performance of GAALS to the corresponding performance of IEEE 802.11 DCF. WU et al.: SCALABLE CHANNEL ALLOCATION AND ACCESS SCHEDULING FOR WIRELESS IoT 3603 Fig. 8. Average end-to-end attributes in a 10-sensor scenario with 3 flows. Fig. 9. Performance comparisons in a 100-sensor WSN with 30 flows. Fig. 10. Performance comparisons in a 50-sensor WSN with 25 flows. Fig. 9 shows that, in comparison with the unstable performance of IEEE 802.11 DCF in the multi-hop multi-flow network, GAALS can provide stable results with lower delay and jitter, and higher network throughput under most network loads. The reason that GAALS is more stable under multi-hop transmission is that we divide the network into grid clusters with reasonable size to avoid collision caused by hidden terminals, and use grid shifting with Latin square based channel allocation to achieve network connectivity. Then GAALS can guarantee network coverage for stable multi-hop multi-flow transmissions in scalable WSNs. We also compared the performance of GAALS with that of MC-LMAC [6], MMSN [7] and MMAC [8] under the same settings. We referred to the layout of the University Center, a neighborhood center adjacent to the University of California, Irvine (UCI), as the simulation scenario of a multi-hop multi-flow WSN with SRMC communication for Latin square scheduling as shown in Fig. 10(a). There are over 50 various eateries, businesses and offices in the University Center. We placed 50 sensors with a distribution by the density of customers over a shopping area of size  $1500\text{ m} \times 1500\text{ m}$ , so that the WSN can be configured to monitor the University Center with a reasonable deployment. We randomly created 25 flows in the network. As shown in the Fig. 10(a), some popular places, i.e. University Office Tower, Edwards Theater, 24 Hour Fitness, were configured with more sensors, which matches the pattern of human's social behavior.

Fig. 10(b) and (c) show the throughput and average delay results, where all the multi-channel protocols can achieve better performance than single channel protocols. GAALS can achieve better performance than others. It adopts grid-based structure to allocate multi-channel and schedule transmission, 3604 IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013 therefore the multi-hop collision is reduced and network connectivity is guaranteed. Meanwhile, the RTS/CTS/ACKfree transmission in GAALS can reduce unnecessary control overheads and help to save the channel resource for efficient multi-hop communication. MC-LMAC, MMSN and MMAC are heuristic multi-channel scheduling protocols that assign channel and schedule channel access from local aspect without considering fully coordination of multi-hop neighbors. MMSN only has the knowledge of 1-hop neighbors and can not prevent the collisions when the sensors failed to sense the channel. In comparison with MMSN, time-slot based MCLMAC can achieve higher throughput. However, the schedule mechanism in MC-LMAC also causes higher delay. GAALS uses compact Latin square based scheduling, according to the size of clusters, to eliminate the schedule redundancy. The separate packet buffer for each channel also can reduce transmission failure. Therefore, GAALS can guarantee reliable delivery and presents better performance than other multichannel protocols, especially when the load rate is high.

## VII. CONCLUSION

We presented "GAALS", a novel grid-based MAC protocol using Latin squares for channel allocation and access scheduling in WSNs with multiple channel communication capabilities, as an enabling technology for IoT. In GAALS, the network is organized into grids and clusters according to location information, and guarantees the network connectivity using the grid shifting algorithm.

The channel access efficiency and fairness were achieved by using compact Latin square based channel access and channel allocation schedules. Especially, we challenged and revised a few design aspects of IEEE 802.11 DCF protocol, such as the RTS/CTS handshaking and NAV mechanisms to avoid the hidden terminal problem in multi-hop WSNs. In our simulation based studies, GAALS demonstrates strong advantages over other multi-channel protocols, in fully connected network and multi-hop multi-flow WSNs, therefore stands as a promising candidate in future multi-channel based WSN deployments.

**REFERENCE:**

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
- [3] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Li, and Y. Liu, "CitySee: Urban CO<sub>2</sub> monitoring with sensors," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1611–1619.
- [4] F. Wu, S. Zhong, and C. Qiao, "Globally optimal channel assignment for non-cooperative wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1543–1551.
- [5] (1995). *XBOW MicaZ Mote Specifications*, Crossbow, Graz, Austria [Online]. Available: <http://www.xbow.com>
- [6] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: A multi-channel MAC protocol for wireless sensor networks," *Ad Hoc Netw.*, vol. 9, no. 1, pp. 73–94, 2011.