Collision-free multichannel superframe scheduling for IEEE 802.15.4 cluster-tree networks

Xi Jin, Qiong Zhang and Peng Zeng*

Lab. of Networked Control Systems,
Shenyang Institute of Automation,
Chinese Academy of Sciences,
114 Nanta Street, Shenhe District, Shenyang,
Liaoning Province 110016, China
E-mail: jinxi@sia.cn
E-mail: zhangqiong@sia.cn
E-mail: zp@sia.cn
*Corresponding author

Fanxin Kong

School of Computer Science,
McGill University,
3480 University Street,
Montreal, Quebec, Canada H3A 0E9
E-mail: fanxin.kong@mail.mcgill.ca

Yang Xiao

Department of Computer Science,
The University of Alabama,
Tuscaloosa, AL 35487-0290, USA
E-mail: yangxiao@ieee.org

Abstract: The beacon-enabled mode is effective in improving real-time performance of wireless sensor networks (WSNs). Keeping cluster heads and members synchronised in cluster-tree networks becomes challenging in presence of beacon collisions. In this paper, we study a collision-free multi-channel superframe scheduling problem. We first formulate this problem in the satisfiability modulo theories (SMT) specification. It can be solved by an SMT solver but with limited scalability. Then, we present two more efficient approaches. Our results show that the proposed approaches can significantly improve the schedulability of superframes compared to the existing approach. Finally, we implement a real system based on a wireless network for industrial automation-process automation (WIA-PA) network to show the feasibility of our proposal.

Keywords: WSNs; wireless sensor networks; cluster-tree networks; beacon-enable mode; beacon collisions; superframe scheduling.

Reference to this paper should be made as follows: Jin, X., Zhang, Q., Zeng, P., Kong, F. and Xiao, Y. (2014) 'Collision-free multichannel superframe scheduling for IEEE 802.15.4 cluster-tree networks', *Int. J. Sensor Networks*, Vol. 15, No. 4, pp.246–258.

Biographical notes: Xi Jin is a Research Associate at Shenyang Institute of Automation, Chinese Academy of Sciences. Her research interests include communication scheduling, energy harvesting sensor networks, and real-time systems.

Qiong Zhang is a Research Associate at Shenyang Institute of Automation, Chinese Academy of Sciences. Her research interests include wireless communication and wireless sensor networks .

Peng Zeng is a Professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research has involved industrial communication and wireless sensor networks.

Fanxin Kong currently is working towards his PhD in the School of Computer Science at McGill University, Montreal, Canada. His research interests include smart grid, cyber-physical systems, and data centres.

Yang Xiao currently is a Full Professor of Department of Computer Science at The University of Alabama, USA. His research areas include communications/networks and security.

1 Introduction

The IEEE 802.15.4 protocol (IEEE-SA Standards Board, 2012), which focuses on low-complexity and low-power transmissions, is a main industry protocol for wireless sensor networks (WSNs). In WSNs for real-time applications with deadlines, the data wireless transmission must satisfy some real-time constraints: data transmissions must finish before a deadline; otherwise, disasters may happen. For example, in the forest fire detection, the forest fire must be detected and predicted promptly and accurately. If the detected data can not be transmitted in real-time, it will cause a great loss of forests and wild animals. The IEEE 802.15.4 protocol enables two operating modes: a non-beacon-enable mode and a beacon-enable mode, whose real-time performance differs a lot. In the non-beacon-enabled mode, sensor nodes use a classical and unslotted carrier sense multiple access with collision avoidance (CSMA/CA) protocol to access the ratio channel. This protocol is easy to implement, but the unpredictable transmission latency can not guarantee the real-time performance of data transmissions (Xiao et al., 2007, 2009). In the beacon-enabled mode, time is divided into periodic beacon intervals, and each beacon interval is further divided into an inactive section and an active section. The active section is called a *superframe*. *guaranteed time slots* (GTSs) existing in one superframe can be assigned to sensor nodes for collision-free data transmissions. Therefore, comparing with the non-beacon-enabled mode, the beacon-enabled mode can provide predictable latency making it attractive for real-time WSNs, and is also the operating mode focused in this paper.

Different topologies support different operating modes. The IEEE 802.15.4 protocol supports three types of topologies: peer-to-peer, star, and cluster-tree. Peer-to-peer networks only support the non-beacon-enabled mode. Star and cluster-tree networks support both operating modes, but the coverage of star networks is limited by the transmission range of the sensor node. Therefore, in this paper, we focus on the cluster-tree topology. In cluster-tree networks, each cluster is managed by one cluster head, which broadcasts periodic beacon frames to synchronise its cluster members and start its superframe. It will cause collisions if multiple heads send beacon frames simultaneously on the same channel; furthermore, cluster members can not synchronise with their heads and can not connect to the network. Therefore, the strategy of scheduling superframes must be designed to avoid beacon frame collisions.

The IEEE 802.15.4 protocol supports multiple ratio channels, which allow multiple beacon frames to be sent simultaneously on different ratio channels without colliding with each other. But the number of channels is limited. It is impossible to assign a channel to beacon frames. Therefore, even if multiple channels can eliminate some collisions, there are still collisions if superframes are scheduled arbitrarily. Existing works (Toscano and Bello, 2009, 2012) have addressed this problem. They used a multichannel-based approach to schedule multiple superframes in parallel. However, this approach only considered the coarse-grained parallelism between the clusters featuring the even (odd) tree depth, which causes that a part of the parallelism between non-communicating clusters is ignored. Therefore, we are based on three essential scheduling constraints and propose more flexible scheduling methods to improve the parallelism of superframe scheduling. Our contributions are listed as follows:

- We analyse the collision-free multichannel superframe scheduling problem and propose two necessary conditions for the schedulability of a superframe set.
- We formulate the collision-free multichannel superframe scheduling problem in the SMT specification. Then an SMT solver can be used to find a feasible solution.
- We also design two heuristic algorithms to address the problem of how to schedule multiple superframes without collisions, since the running time of the purely SMT-based approach may become unacceptable as the problem size grows.
- Evaluation results indicate that our algorithms effectively schedule superframes comparing to the state-of-the-art on the multichannel superframe scheduling.
- We show the feasibility of our proposal through a real system.

The rest of paper is organised as follows: Section 2 introduces the related works. Section 3 describes our problem model. Section 4 introduces SMT and formulates the problem as an SMT instance. Section 5 proposes two heuristic algorithms. Section 6 shows the evaluation results. Section 7 describes the implementation of a testbed and shows the feasibility of our proposal. Finally, Section 8 concludes this paper.

2 Related works

The related work is classified into the following two categories: single-channel superframe scheduling, in which superframes are scheduled on single ratio channel, and multichannel superframe scheduling, in which multiple channels are used to avoid collisions.

The single-channel superframe scheduling is a common problem in WSNs that have been addressed in many research works. Koubâa et al. (2007, 2008) presented a centralised beacon frame scheduling mechanism to schedule a superframe set with different durations and beacon intervals, based on the pinwheel scheduling algorithm. Hanzalek and Jurcik (2010) proposed a collision-free superframe schedule to meet end-to-end deadlines of data flows and minimise the energy consumption. Pan and Tseng (2008) proposed optimal beacon scheduling algorithms for special cases and heuristic scheduling algorithms for general cases to achieve quick convergecast in Zigbee beacon-enabled networks. Yeh et al. (2008) presented a low-delay, two-way (broadcast and convergecast) beacon scheduling algorithm for Zigbee/IEEE 802.15.4 standards. Koubâa et al. (2006a, 2006b) analysed the performance of the GTS allocation mechanism using network calculus formalism, and then improved bandwidth utilisation via an implicit GTS allocation mechanism. Some other works (Gao and He, 2008; Jeon et al., 2007; Neugebauer et al., 2005; Chen et al., 2010) addressed superframe duration adaptation methods to trade-off between power consumption and end-to-end delay, with a little consideration of the collision-free superframe scheduling. These previous works did not consider how to use multiple channels to avoid beacon frame collisions.

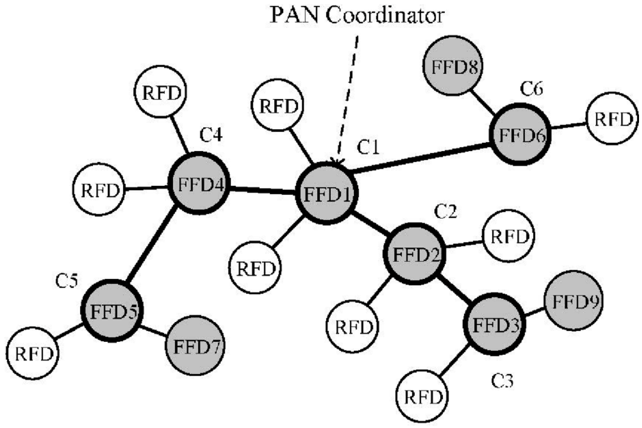
There has been some works on multichannel superframe scheduling. Franchino and Buttazzo (2012) proposed a beacon-enabled medium access control (MAC) protocol, which addressed two issues: real-time communication and energy saving. But the MAC protocol is not to optimise the multichannel assignment. Kim et al. (2014) designed a beacon-based WSNs with multiple channels, but the system did not support GTSs. Toscano and Bello (2009, 2012) proposed a multichannel superframe scheduling (MSS) algorithm and analysed the performance of the algorithm. Different from the MSS algorithm, our approaches study the more fine-grained parallelism as shown in Section 1.

3 Network model, problem statement and problem analysis

3.1 Network model

Figure 1 shows the topology of a typical cluster-tree network. In an IEEE 802.15.4 network, there are two device types: *full-function devices* (FFD) and *reduced-function devices* (RFD). FFDs can act as cluster heads, but RFDs cannot. As shown in Figure 1, devices FFD1 – FFD6 are cluster heads, and other devices are cluster members. If the network topology is fixed and known in advance, an offline scheduling algorithm is performed and the scheduling information is pre-defined in cluster heads; otherwise, a lightweight scheduling algorithm is implemented in the personal area network (PAN) coordinator. In second case, before cluster heads broadcast their first beacon frames, they send a command with characteristics of their superframes to the PAN coordinator via the tree network. The PAN coordinator runs the scheduling algorithm and sends the result to cluster heads. Then cluster heads periodically send beacon frames to synchronise their members. Note that a FFD is the member of its parent cluster and the head of its cluster, such as FFD2 is the member of Cluster 1 and the head of Cluster 2.

Figure 1 The cluster-tree network



A cluster-tree network is characterised by $N =< C, L, E >$:

- We consider a cluster-tree network of n clusters $C = \{c_1, \ldots, c_n\}$. Each cluster contains one cluster head and multiple cluster members. Cluster members only communicate with their cluster head.
- Matrix $L : C \times C$ is the set of links. If c_i and c_j can communicate reliably with each other, the element l_{ij} in L is equal to 1; otherwise, $l_{ij} = 0$.
- Matrix $E : C \times C$ is the set of collisions. If c_i and c_j can collide with each other, the element e_{ij} in E is equal to 1; otherwise $e_{ij} = 0$. When $e_{ij} = 0$, the cluster c_i and c_j can be scheduled simultaneously on the same channel, which is called the spatial channel reuse. Additionally, the transmitting range of a sensor node is smaller than its interfering range.

In the beacon-enabled mode, the beacon frame divides time into periodic *Beacon Interval*s (BI), as shown in Figure 2. Each BI contains two portions: an active portion, also called *superframe duration* (SD); and an inactive portion, where sensor nodes will enter the sleep mode to save energy. The duration of BI and SD is determined by two parameters the *beacon order* (BO) and the *superframe order* (SO), respectively, as follows:

$$BI = aBaseSuperframeDuration \times 2^{BO}$$
$$SD = aBaseSuperframeDuration \times 2^{SO}$$

where $aBaseSuperframeDuration$ is a constant defined in the IEEE 802.15.4 standard, and $0 \leq SO \leq BO \leq 14$. The *duty cycle* (DC) of clusters is defined as

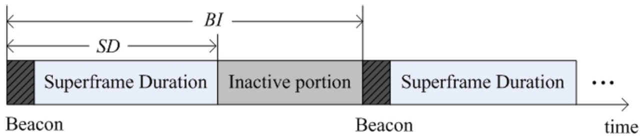
$$DC = \frac{SD}{BI} = 2^{SO-BO}.$$

The superframe of the cluster c_i is characterised by $\delta_i =< BO_i, SO_i >$ and Δ is used to represented the set of superframes.

3.2 Problem definition

Each cluster generates an independent superframe. In order to avoid colliding with other clusters' superframes, the cluster head should schedule its superframe during other clusters' inactive portions. The 15.4b Task Group has proposed an approach that assigns a time offset for each cluster to stagger their active portions. But they do not detail how to select the time offset. We will address the problem. o_i denotes the time offset of the cluster c_i, which is equal to the time between the time zero and the beginning of c_i's first superframe. In the IEEE 802.15.4 standard, the time offset was defined as "the time between the parent and the child superframes", which is different from ours. But it can be easily achieved according to our defined offset o_i.

Figure 2 Superframe structure (see online version for colours)



There are 16 non-overlapping channels in an IEEE 802.15.4 network. But not all of these channels can be accessed always, since they may suffer from persistent external interference. We use $M(M \leq 16)$ to denote the number of available channels. If two superframes are allocated to different channels, there are no collisions between them. Therefore, scheduling superframes over multiple channels is another perspective to avoid superframe collisions. We use r_i to denote the channel allocated to the cluster c_i.

We only consider that the superframes are scheduled within a hyper-period $H = LCM(2^{BO_1}, 2^{BO_2}, \ldots, 2^{BO_n}) = \max_{i\in[1,n]} (2^{BO_i})$, since after that all the superframe schedulings are cyclically repeated.

To sum up, the problem that we address in this paper can be stated as follows. *Given the cluster-tree network N, the superframe set Δ and the number of available channels M, our objective is to find each cluster's offset o_i and allocated channel r_i such that all of superframes can be scheduled without collisions under the following constraints:*

- *If $l_{ij} = 1$, the superframes δ_i and δ_j cannot be scheduled at the same time.* As shown in Figure 1, we assume that the cluster head $FFD3$ collects data from its cluster members and send them to $FFD2$. If the superframe δ_2 and δ_3 are scheduled at the same time, the cluster head $FFD3$ receives and sends data simultaneously. But a sensor node cannot both receive and send at the same time.
- *If $l_{ij} = 0$ and $e_{ij} = 1$, the superframes δ_i and δ_j can be scheduled simultaneously on the different channels, or at the different timeslice.* Since multiple channels can isolate interferences.
- *If $e_{ij} = 0$, the schedules of the superframes δ_i and δ_j are completely independent.* It is because the spatial channel reuse technique.

3.3 Problem analysis

The three constraints make superframes difficult to schedule without collisions. First, the intricate relationships (l_{ij} and e_{ij}) between clusters restrict the assignments of timeslices and channels. Second, one superframe may present different features to different superframes due to the spatial reuse technique. Therefore, it is hard to design an algorithm to find the optimal schedule for every superframe set. However, based on our problem model, we identify some conditions that refuse some unschedulable superframe sets before the algorithm running. In other words, we can use these conditions to quickly decide whether the superframe scheduling may have a feasible solution.

According to Constraint (1), if two clusters communicate with each other, their superframes can not overlap in the time dimension. Therefore, we can derive Theorems 1 and 2.

Theorem 1: *If the superframe set Δ is schedulable, there must be*

$$\forall i, j \in [1, n] \text{ and } l_{ij} = 1 : DC_i + DC_j \leq 1$$

Proof: We assume by contradiction that there are two communicating clusters c_a and c_b, and the sum of their duty cycles exceeds one. In the hyper-period H, there are $\frac{H}{BI_a}$ instances of the superframe δ_a and $\frac{H}{BI_b}$ instances of the superframe δ_b. To schedule all of these instances, $\frac{H}{BI_a} \times SD_a + \frac{H}{BI_b} \times SD_b$ timeslices are needed. This formula can be rewritten as $H \times \left(\frac{SD_a}{BI_a} + \frac{SD_b}{BI_b}\right) = H \times (DC_a + DC_b)$. It is greater than H if our assumptions hold. Thus they can not be scheduled at the different timeslices in the hyper-period H. Therefore, the above assumptions do not hold. □

Theorem 2: *If the superframe set Δ is schedulable, there must be*

$$\forall i, j \in [1, n] \text{ and } l_{ij} = 1 : SD_i + SD_j \leq \min\{BI_i, BI_j\}$$

Proof: To avoid the collision between the two superframes δ_i and δ_j, the superframe δ_i (δ_j) is scheduled during the inactive portion of the cluster c_j (c_i). So we can get that $SD_i \leq BI_j - SD_j$ and $SD_j \leq BI_i - SD_i$. Then $SD_i + SD_j \leq BI_j$ and $SD_i + SD_j \leq BI_i$. Therefore, $SD_i + SD_j \leq \min\{BI_i, BI_j\}$. □

Corollary 1: *If a superframe set satisfies Theorem 2, then it must satisfy Theorem 1.*

Proof: For $l_{ij} = 1$, we set that $BI_i \leq BI_j$. If the superframe δ_i and δ_j satisfy Theorem 2, then $\frac{SD_i}{BI_i} + \frac{SD_j}{BI_i} \leq 1$. The sum of duty cycles of δ_i and δ_j is $DC_i + DC_j = \frac{SD_i}{BI_i} + \frac{SD_j}{BI_j}$, which is less than or equal to $\frac{SD_i}{BI_i} + \frac{SD_j}{BI_i}$. Then we can get that $DC_i + DC_j \leq \frac{SD_i}{BI_i} + \frac{SD_j}{BI_i} \leq 1$. So the superframes δ_i and δ_j satisfy Theorem 1. □

Theorems 1 and 2 are necessary conditions on the schedulability of a superframe set. Furthermore, Theorem 2 is the sufficient condition for Theorem 1. Therefore, when we determine the unschedulability of a superframe set, we only consider Theorem 2. In other words, if a superframe set does not satisfy Theorem 2, it can not be scheduled without collisions; otherwise, we may find a feasible solution for it.

4 SMT based collision-free multichannel superframe scheduling approach

This section describes the formulation of the collision-free multichannel superframe scheduling problem in the SMT specification. First, we will review SMT. Then the used constants and variables are introduced, and the constraints that apply on the variables are proposed.

4.1 SMT introduction

Satisfiability (SAT) problem (Cook, 1971) is to determine if there exists a solution that satisfies a given boolean formula. While some problems cannot be described by the boolean formula, e.g., the arithmetic problem containing the operator $+$. To address these problems, satisfiability modulo theories (SMT) (Barrett et al., 2009) are proposed to solve logical formulas that are interpreted modulo a background theory (such as the theory of real numbers).

The most famous algorithm for solving instances of SAT is DPLL (Davis and Putnam, 1960; Davis et al., 1962), which is based on the systematic search. Its extension DPLL(T) (Ganzinger et al., 2004) is widely used to determine the solutions of SMT problems. Based on DPLL(T), many SMT solvers are proposed, such as Z3 (Moura and Bjørner, 2008), Yices (Yeh et al., 2008), CVC (Barrett and Tinelli, 2007) and so on. In recent researches (e.g., Huang et al. (2012) andRai et al. (2012)), the solutions solved by SMT solvers have been as an excellent standard to evaluate the effectiveness of other algorithms. Therefore, in this paper we first use the SMT solver to solve the collision-free multi-channel superframe scheduling problem.

4.2 Constants and variables

The list of the constants is as follows:

- *The hyper-period*: $\boldsymbol{H}$, which is described in Section 3.2.
- *The number of superframe instances of the cluster c_i in a hyper-period*: $\boldsymbol{p}_i = {}^{H}\!/_{2^{BO_i}}$.
- The link set $\boldsymbol{L}$, the collision set $\boldsymbol{E}$ and the superframe set Δ, which are defined in Section 3.1.

For each superframe δ_i, as previously stated, two variables are used to denote its offset and channel:

- $\boldsymbol{o}_i \in \left[0, 2^{BO_i} - 2^{SO_i}\right]$ is the time offset of the superframe δ_i. The end of the superframe δ_i must be not greater than the end of its BI. Therefore, the upper bound of o_i is $(2^{BO_i} - 2^{SO_i})$.
- $\boldsymbol{r}_i \in [0, M)$ is the r_ith channel, which is allocated to δ_i.

Additionally, the binary variable $\boldsymbol{s}_{it} \in \{0, 1\}$ is 1 iff at the timeslice t, the superframe δ_i is being scheduled and 0 otherwise.

4.3 Constraints

Timeslice constraint: The offset o_i determines which timeslice s_{it} is used by the superframe δ_i. We can see that the jth ($0 \leq j < p_i$) instance of the superframe δ_i is scheduled between the timeslice $\eta^1_{ij} = (o_i + j \times 2^{BO_i})$ and the timeslice $\eta^2_{ij} = (o_i + j \times 2^{BO_i} + 2^{SO_i} - 1)$. Hence, the timeslice constraint is:

$$\forall i \in [1, n], \forall t \in [0, H) : \bigvee_{j \in [0, p_i)} ((\eta^1_{ij} \leq t) \wedge (t \leq \eta^2_{ij})) = 1 \rightarrow s_{it} = 1$$

Collision-free constraints: In Section 3.2, we have listed three constraints for the collision-free scheduling. The corresponding SMT formulations of Constraints (1) and (2) are as follows. There is no Constraint (3), because the superframes δ_i and δ_j can not collide with each other if $e_{ij} = 0$.

- *Constraint (1)*:

$$\forall i, j \in [1, n] : l_{ij} = 1 \rightarrow \bigvee_{t \in [0, H)} (s_{it} \wedge s_{jt}) = 0$$

- *Constraint (2)*:

$$\forall i, j \in [1, n] : (e_{ij} = 1) \wedge (l_{ij} = 0) \rightarrow \left(\bigvee_{t \in [0, H)} (s_{it} \wedge s_{jt}) \right) \wedge (r_i = r_j) = 0.$$

According to the above formulation, the SMT solver can solve this problem.

5 Efficient collision-free multichannel superframe scheduling algorithms

Since SMT solvers are based on the systematic search, they suffer from the scalability issue as the problem size increases. To address this problem, we propose two heuristic algorithms to improve the scalability.

5.1 Modified simulated annealing algorithm

Simulated annealling is a generic metaheuristic for finding the global optimisation solution in a large search space. It has the ability to avoid trapping in a local optimal solution by accepting the worse solution with a probability. The collision-free multichannel superframe scheduling is a search problem in the 2D solution space $< time, channel >$. Therefore, we propose a Modification of the basic Simulated Annealing (called MSA) to find the available scheduling in this space. Each superframe's offset o_i and allocated channel r_i are encoded as a variable $f_i (0 \leq f_i < 1)$, which are as follows:

$$r_i = \left\lfloor \frac{\lfloor M \times w_i \times f_i \rfloor}{w_i} \right\rfloor$$

$$o_i = \lfloor M \times w_i \times f_i \rfloor - r_i \times w_i$$

where $w_i = 2^{BO_i} - 2^{SO_i}$. Therefore, we can use the state $F = \{f_1, ..., f_n\}$ to represent the scheduling of all superframes. Then the superframe collision is mapped to the state's energy. For the superframe δ_i, its energy is defined as:

$$\varepsilon_i = \sum_{j=1}^{n} (\alpha_{ij} + \beta_{ij}) \quad (1)$$

where α represents the collision between two communicating clusters and β is the collision between two non-communicating clusters:

$$\alpha_{ij} = \begin{cases} \sum\limits_{t=1}^{H} (s_{it} \times s_{jt}) & \text{if } (l_{ij} = 1) \\ 0 & \text{others} \end{cases}$$

$$\beta_{ij} = \begin{cases} \sum\limits_{t=1}^{H} (s_{it} \times s_{jt}) & \text{if } (r_i = r_j) \text{ and } (l_{ij} = 0) \text{ and } (e_{ij} = 1) \\ 0 & \text{others} \end{cases}$$

The definition of s_{it} is the same as that in Section 4.2, which is formulated as

$$s_{it} = \begin{cases} 1 & \text{if } \exists j \in [0, p_i) \rightarrow (\eta_{ij}^1 \leq t \leq \eta_{ij}^2) \\ 0 & \text{others} \end{cases}. \quad (2)$$

The energy of the state F (denoted by $Energy(F)$) is equal to $\sum_{i=1}^{n} \varepsilon_i$. If $Energy(F) = 0$, there are no collisions, and the offsets $\{o_1, ..., o_n\}$ and allocated channels $\{r_1, ..., r_n\}$ corresponding to the current state F are the feasible solution.

Our proposed MSA is shown in Algorithm 1. $T_{\max}$ and $T_{\min}$ denotes the maximum and minimum temperature, respectively. In the first two lines, we randomly choose the initial state $F0$ and calculate its energy $Energy(F0)$. If the initial $F0$ is an available solution, the algorithm returns it (lines 3–4). Then in the basic iteration (lines 6–14), the neighbouring state $F1$ is generated by the function $Neighbour()$ (shown in Algorithm 2) according to the current state $F0$, and lines 12–13 decide between accepting the new state $F1$ or staying in the state $F0$. The iteration is stopped until the temperature tem reaches $T_{\min}$ or Energy(F) reaches zero (lines 6 and 9–10).

During the iteration of Algorithm 3, our objective is to decrease the energy. Therefore, for each generated neighbouring state, its energy should be smaller than that of the current state. The function Neighbot() randomly changes the state f_j with the maximum energy (lines 3 and 6) and tries to achieve a new state with the smaller energy. Lines 4–5 probabilistically choose others as the changed state in order to avoid the solution trapping in the local optimisation, where the threshold THR is defined by designers.

In Algorithm 2, the complexity of lines 1, 2 and 3 is $O(nH)$, $O(n^2H)$ and $O(n)$, respectively. Therefore, the complexities of $Neighbour()$ and $Energy()$ are $O(n^2H)$. In Algorithm 1, the number of iterations of *while* loop in line 6 is $O(T_{\max} - T_{\min})$. Therefore, the complexity of Algorithm 1 is $O(n^2HT_{\max})$, where $H \leq 2^{14}$.

Algorithm 1 Modified simulated annealing algorithm

Require: N, Δ, M, T_{max}, T_{min};
Ensure: F;
1: randomly choose the initial set $F0$;
2: $e0 = Energy(F0)$;
3: **if** $e0 == 0$ **then**
4:   **return** $F0$;
5: $tem = T_{max}$;
6: **while** $tem > T_{min}$ **do**
7:   $F1 \leftarrow Neighbor(F0)$;
8:   $e1 = Energy(F1)$;
9:   **if** $e1 == 0$ **then**
10:     **return** $F1$;
11:   $dE = e1 - e0$;
12:   **if** $(dE < 0) \,||\, (exp(-dE/tem) > random())$ **then**
13:     $F0 \leftarrow F1$;
14:   $tem - -$;
15: **return** FAIL;

Algorithm 2 $Neighbor()$ in Algorithm 1

Require: the current state F, THR, N, Δ;
Ensure: a new state F;
1: calculate the 2D array s according to the state F (Formula (2));
2: calculate the energy ε_i for each superframe δ_i according to the array s (Formula (1));
3: $j = \arg(\max\limits_{i \in [1,n]} \{\varepsilon_i\})$;
4: **if** $random() < THR$ **then**
5:   j = a random value between the range of 1 to n;
6: $f_j = random()$;
7: **return** F;

5.2 *Heuristic collision-free multichannel superframe scheduling algorithm*

In this subsection, we propose a collision free multichannel superframe scheduling algorithm (called CFSS) to solve the problem stated in Section 3. The algorithm CFSS is shown as Algorithm 3. First, we sort the superframe set Δ as the following rules (line 1):

- if it exists $BO_i > BO_j$, then put δ_j before δ_i
- if $BO_i = BO_j$ and $SO_i < SO_j$, then also put δ_j before δ_i
- if BO and SO are both equal between δ_i and δ_j, then sort them in the breath-first order of the tree-network.

In our algorithm CFSS, the definition of array s_{it} is the same as before, and is initialed with 0 (line 2). In the iteration of lines 3–24, we assign the time offset o_i and the channel r_i to each superframe δ_i. For the superframe δ_i, we traverse all of its available time offsets $\left[0, 2^{BO_i} - 2^{SO_i}\right]$ (line 4). If the offset o_i can be assigned as the current offset t (line 6) and there is no collisions between the superframe δ_i and the previous assigned superframes (lines 9–14), then t is the offset o_i. The set Γ includes the available channels in the current timeslice. If the superframe δ_i collides with the superframe

δ_j on the channel r_j, the channel r_j is removed from Γ (lines 11–12). During the traversal of lines 4–20, if the set Γ has no elements (lines 13–14), there is no available channel to be assigned to the superframe δ_i. In this case, the superframe set Δ can not be scheduled (line 24). Otherwise, the superframe δ_i can be scheduled without collisions. For choosing channel (lines 16–19), we first choose the even channel, if there are no even channels in the available channel set Γ, the first odd channel is chose. Using this choosing method, the cross-channel interference between adjacent channels can be avoided (Bello and Toscano, 2009; Toscano and Bello, 2012). When the offset and the channel of the superframe δ_i are assigned, its timeslice array s_{it} is updated (lines 21–22) to assist the following superframes in calculations of collisions α and β.

The number of iterations of *for* loop in lines 3, 4 and 7 is $O(n)$, $O(H)$ and $O(n)$, respectively. The complexities of calculations of collisions α_{ij} and β_{ij} are both $O(H)$. Therefore, the time complexity of Algorithm 3 is $O(n^2H^2)$, where $H \leq 2^{14}$.

Algorithm 3 CFSS

Require: N, M, Δ;
Ensure: $\{r_i, ..., r_n\}$ and $\{o_1, ..., o_n\}$;
1: sort the superframe set Δ according to the sorting rules, where δ_1 has the smallest BO;
2: initial each s_{it} with 0;
3: **for** each $i \in [1, n]$ **do**
4:   **for** each $t \in \left[0, 2^{BO_i} - 2^{SO_i}\right]$ **do**
5:     $\Gamma \leftarrow$ all of available channels;
6:     $o_i = t$;
7:     **for** each $j \in [1, i)$ **do**
8:       $r_i = r_j$;
9:       **if** $\alpha_{ij} \neq 0$ **then**
10:         break;
11:       **if** $\beta_{ij} \neq 0$ **then**
12:         $\Gamma \leftarrow \Gamma - \{r_j\}$;
13:         **if** $\Gamma == \Phi$ **then**
14:           break;
15:     **if** $j == i$ and $\Gamma \neq \Phi$ **then**
16:       **if** Γ has even channels **then**
17:         $r_i =$ choose the first even channel in Γ;
18:       **else**
19:         $r_i =$ choose the first odd channel in Γ;
20:       break;
21:   **if** $t \leq (2^{BO_i} - 2^{SO_i})$ **then**
22:     for each $t \in [0, H)$, update s_{it} according to Formula (2);
23:   **else**
24:     **return** FAIL;
25: **return** $\{r_i, ..., r_n\}$ and $\{o_1, ..., o_n\}$;

If all the clusters in a superframe set have the same BI and SD, the superframe set is described as *homogeneous*. In this case, Toscano and Bello analysed the schedulability space of the classical multichannel superframe scheduling MSS (Toscano and Bello, 2009, 2012). We prove that, for homogeneous superframe sets, the schedulability space of our algorithm CFSS is the same as that of MSS.

Theorem 3: *If a homogeneous superframe set is able to be scheduled by the algorithm MSS, our algorithm CFSS can also schedule it.*

Proof: We set that the superframe set Δ can be scheduled by the algorithm MSS. For our algorithm CFSS, if all clusters in the superframe set Δ have the same BIs and SDs, our sorting rules (1) and (2) are useless. Then the clusters are sorted in the breadth-first order and the PAN coordinator with the even depth zero is the first scheduled cluster, whose timeslice interval is from 0 to $SD - 1$. After that the clusters that have depth 1 are scheduled. Since these clusters communicate with the PAN coordinator, their timeslice interval is from SD to $2 \times SD - 1$. The clusters with depth 2 communicate with the clusters with depth 1, and therefore, they are assigned for the timeslice interval 0 to $SD - 1$. This process is repeated until all of clusters are scheduled without collisions. We can see that the clusters with the even depth are assigned for the timeslice interval 0 to $SD - 1$, and those with the odd depth are assigned for the timeslice interval SD to $2 \times SD - 1$. These are the same as the two timeslice intervals $TS1$ and $TS2$ of the algorithm MSS. Furthermore, in a timeslice interval the channel assignment (lines 16–19) of our algorithm CFSS is the same as that of the algorithm MSS. The two algorithms have the same timeslice intervals and the channel assignment, and they all support the spatial channel reuse technique. Therefore, the result of our algorithm CFSS is the same as that of the algorithm MSS. □

6 Evaluation

In this section, we compare our approaches (SMT-based algorithm (abbreviated as SMT), MSA and CFSS) against the algorithm MSS (Toscano and Bello, 2009, 2012) with spatial reuse, which is the state of the art on multichannel superframe scheduling. The SMT formulation is implemented in C API of Z3 (Moura and Bjørner, 2008), which is the winner of the SMT solver competition 2011 and outperforms the winner of the last SMT solver competition (Satisfiability Modulo Theories Competition, http://smtcomp.sourceforge.net/2012/). Furthermore, the two heuristic algorithms are implemented in C++. These programs run on a Windows machine with 3.4GHz CPU and 4GB memory.

We use *schedulable ratio* and *running time* as the performance metrics in the following comparisons. *Schedulable ratio* is defined as the percentage of superframe sets for which an algorithm is able to find a feasible solution. Furthermore, *the running time* is the total time required to find a feasible schedule for a superframe set within the hyper-period.

The WSNs used in our experiments are generated randomly. There are n clusters placed in a square area. The PAN coordinator is placed at the centre, and other $n - 1$ clusters are placed randomly in the playground area A. Then each cluster selects the nearest cluster, which must be in its transmitting range and have been connected to the PAN coordinator, as its father node. If some clusters do not find their father nodes, their locations are generated randomly again. Repeat this process until all clusters are connected to the PAN coordinator. According to the suggestion in Toscano and Bello (2012), the number of clusters n, the playground area A and the network density D should satisfy

$$\frac{n}{D \times A} = \frac{2\pi}{d^2\sqrt{27}} \tag{3}$$

where the transmitting range d is set as 40m. Furthermore, the channel reuse distance is equal to $2\sqrt{3}d$. In addition, BI and SD are distributed randomly in the range that we specify. To show their impacts on schedulable ratio, we specify two ranges for BO and SO : (1) a BO is allowed to be smaller than other superframes' SOs, this case is denoted as OV, in which $BO \in [1, 6]$ and $SO \in [0, 2]$; (2) a BO must be larger than all of SOs, this case is denoted as NOV, in which $BO \in [3, 6]$ and $SO \in [0, 2]$. The suffixes -OV and -NOV point out which range is used. When a network and its superframe set are generated, we check whether they satisfy the necessary condition (Theorem 2). If not, they can not be scheduled without collisions and will be abandoned.

In the following experiments, we compare the four algorithms under varying number of channels M, the network density D and the number of clusters n, respectively, and there are 200 superframe sets solved at each measurement point.

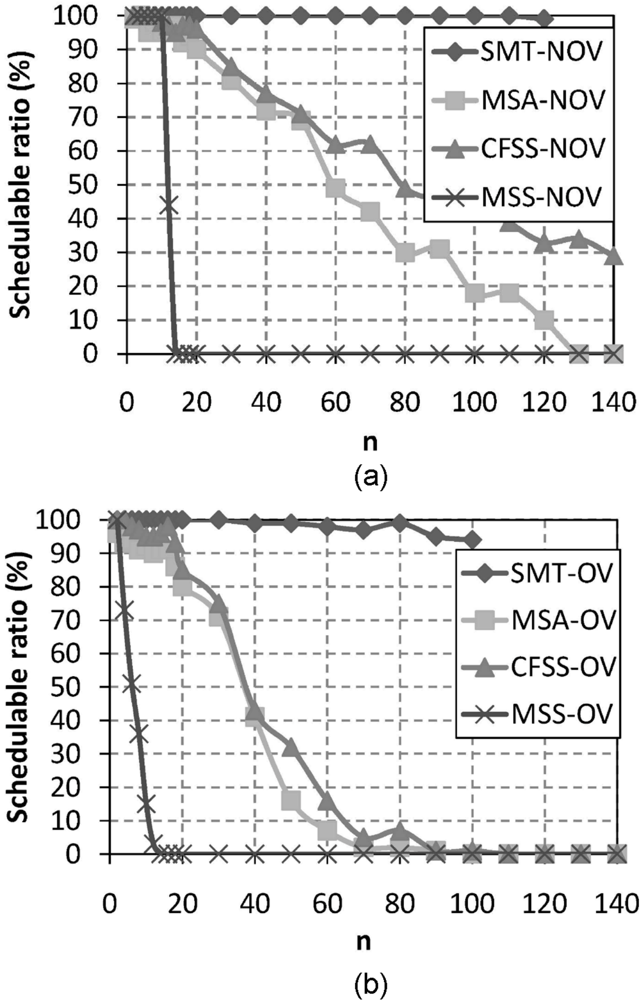
6.1 Comparison under varying number of clusters n

We configure the number of clusters to be from 2 to 140 with OV and NOV. Other parameters are set as follows. The network density is 1. Furthermore, in order to make the superframe set solved by the Z3 SMT solver, we set the number of channels M as 6. The comparison of schedulable ratios is shown in Figure 3(a) (-NOV) and Figure 3(b) (-OV).

Figure 3(a) shows that the schedulable ratio decreases as n increases. When $n > 120$, there are no the solutions of SMT, since it can not find a feasible solution in an acceptable time. The solved superframe set must satisfy Theorem 2, and the schedulable ratio of the SMT formulation is almost 1. These indicate that the necessary condition in Theorem 2 is highly effective. The schedulable ratios of our algorithms MSA and CFSS are greater than that of MSS. Comparing MSA and CFSS, as the number of clusters increases, CFSS is superior to MSA. The reason is that MSA is a searching method, while CFSS is to construct a solution based on a heuristic method. The large solution space with more clusters makes the searching method MSA difficult to find a feasible solution.

Figure 3(b) shows the results of four algorithms with -OV. Comparing with Figure 3(a), all of schedulable ratios become small. It is because that the superframe with small BO divides the schedulable period into multiple small periods, which reduces the probability of scheduling superframes with a large SO. For the algorithm MSS, if the maximum SD in a timeslice interval is greater than or equal to the minimum BI in another interval, the superframe set can not be scheduled, which is proven by Toscano and Bello (2009). Therefore, the lower bound of the range of BO has a great impact on MSS, and the schedulable ratio of MSS-OV is the lowest.

Figure 3 Schedulable ratio under varying number of clusters and $M = 6$, $D = 1$: (a) setting the range of BO and SO as NOV and (b) setting the range of BO and SO as OV



The schedulable ratio of the algorithm MSS with -NOV is quite different from that presented in Toscano and Bello (2012). It is because that in these figures the number of channels is set as 6, instead of 16 (Figure 4 shows the comparison with $M = 16$). However, when the number of channels is 6 and $n = 12$, MSS should have scheduled all of superframe sets, but in Figure 3(a), its schedulable ratio is 44%. The first reason is that without loss of generality, the generating method of our networks is random. While that of Toscano and Bello (2012) is based on a fixed probability distribution. Therefore, in our networks, the distribution of nodes is asymmetrical. Once the number of clusters with the odd (or even) depth is greater than 6, MSS with 6 channels can not schedule them. In addition, recall that the channel reuse distance is $2\sqrt{3}d$ and n, A and D satisfy formula (3). We can get that when $D = 1$, in each channel non-reuse area $\pi(\sqrt{3}d)^2$, there are 11.4 clusters on average (i.e., $n = 11.4$). Therefore, the random generating method probably makes more than 12 clusters in a channel non-reuse area. Once this happens, MSS can not schedule it. The two reasons also result in that the schedulable ratio of MSS is zero, when $M = 6$ and $n \geq 14$.

Figure 4 shows the comparison of schedulable ratios of MSA, CFSS and MSS with $M = 16$. There are no SMT, since its running time is unacceptable when $M = 16$. From the figure we can see that the schedulable ratio of our algorithm CFSS is almost unaffected by the different ranges of BO and SO. While Simulated Annealing is an intelligent random search method and suitable for a large search space. Therefore, when $n \leq 50$, the result of MSA-NOV is less than that of MSS-NOV. As the search space enlarges, comparing with MSS-NOV, MSA-NOV is more effective. The schedulable ratio of MSS-NOV is also different from those in experiments of Toscano and Bello (2012), and the schedulable ratio of MSS-OV is only 16% even though the number of clusters is 10. These are also because of the two reasons that we have mentioned in the previous paragraph.

Figure 4 Schedulable ratio with $M = 16$ and $D = 1$: (a) setting BO and SO as NOV and (b) Setting BO and SO as OV

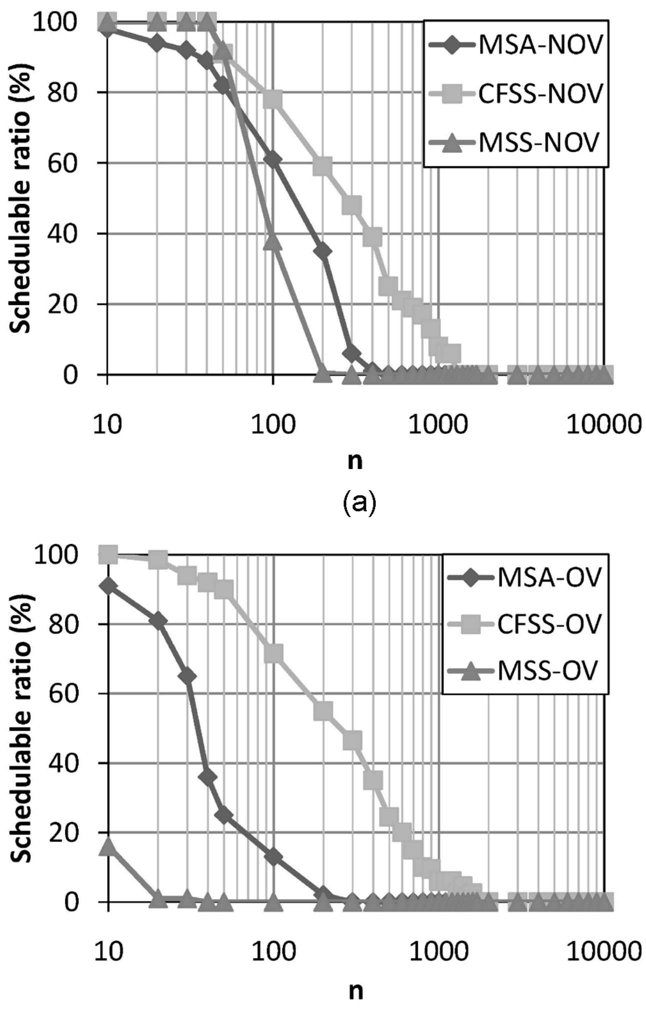
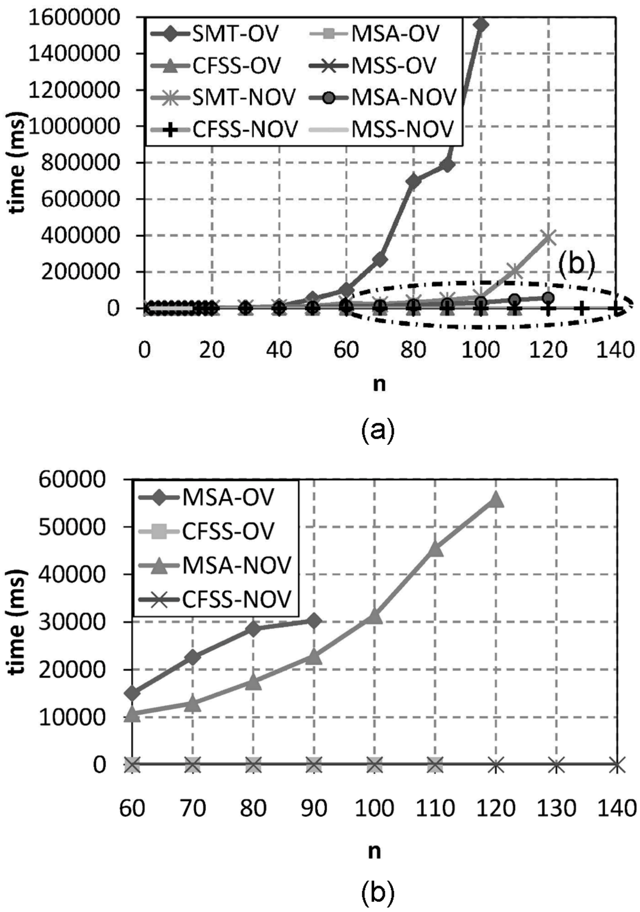


Figure 5 shows the running time for the experiments in Figure 3. Some points do not exist due to no corresponding feasible solution in Figure 3. We can see that

- SMT has the longest running time, since it is based on the systematic search
- CFSS has the lowest running time ($\leq$ 6 ms) and better scalability. Since it is to construct a solution and has less searching time
- the running time of an algorithm with NOV is less than that of the algorithm with OV, which is because that in NOV setting the long schedulable period has more scheduling flexibility and is easy to schedule.

Figure 5 Running time for the experiments in Figure 3: (a) all of results and (b) a part of (a)



6.2 Comparison under varying number of channels M

In this subsection, we compare the four algorithms when the number of channels ranges from 5 to 16. The network density and the number of clusters is set as 1 and 50, respectively. The result is shown in Figure 6. For our algorithms CFSS and MSA, when the number of channels is larger than a certain value, such as 7 for -NOV and 8 for -OV, their schedulable ratios do not increase as the number of channels increases. In this case, the parallelism provided by multiple channels can satisfy the scheduling requirement, and the schedulability is mainly limited by Constraint (1) in which the communicating clusters cannot be scheduled in parallel. It shows that our algorithms CFSS and MSA is effective even if some channels suffer from external interference.

From Figure 7, which shows the running time for the experiments in Figure 6, we can see that

- For SMT, there is a special number of channels to achieve the lowest running time, such as 6 for SMT-NOV and 8 for SMT-OV. If M is greater than the special number, the large solution space needs more time to be searched. While if M is less than the special number, there are less feasible solutions. Therefore, even though the solution space is small, more time will be spent to find a feasible solution.
- The running time of MSA decreases as the number of channels increases. This can be explained that the network with more channels has less collisions, i.e. it is easy to find a collision free solution if more channels can be used.]item The constructive algorithm CFSS still has the lowest running time ($\leq$ 3 ms) between all of algorithms.

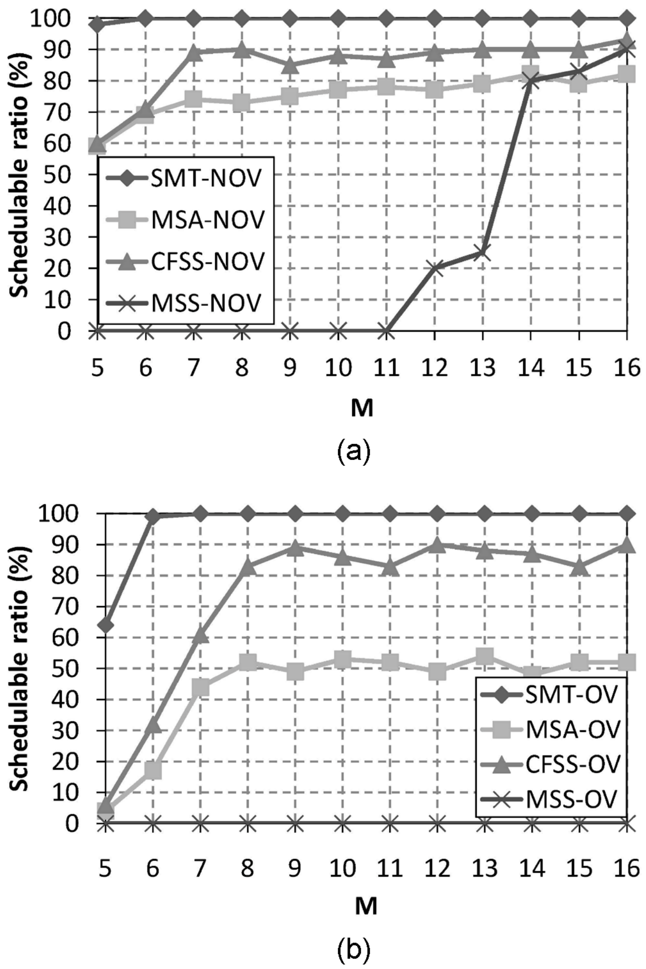


Figure 6 Schedulable ratio under varying number of channels and $N = 50$, $D = 1$: Setting the range of BO and SO as NOV and Setting the range of BO and SO as OV

6.3 *Comparison under varying network densities D*

Figure 8 shows the schedulable ratio of the four algorithms with various network densities. The side length of the playground is 100, 200 and 500 m, respectively. Except SMT, our algorithm CFSS is the most effective. The schedulable ratio decreases as A and D increase, which is mainly because that more superframes need to be scheduled. Figure 9 shows the running time of Figure 8(a) (other subfigures' running times are similar and omitted here). With the increase of the network density, the search space enlarges and superframe collisions increase. Therefore, MSA's running time rapidly increases. While there is no impact on CFSS, no matter the network density.

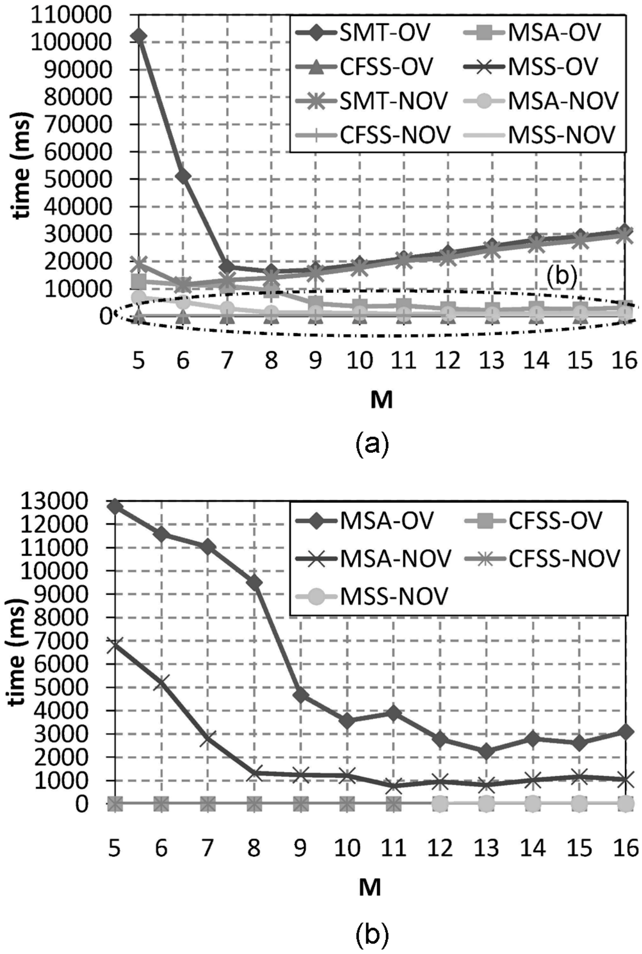


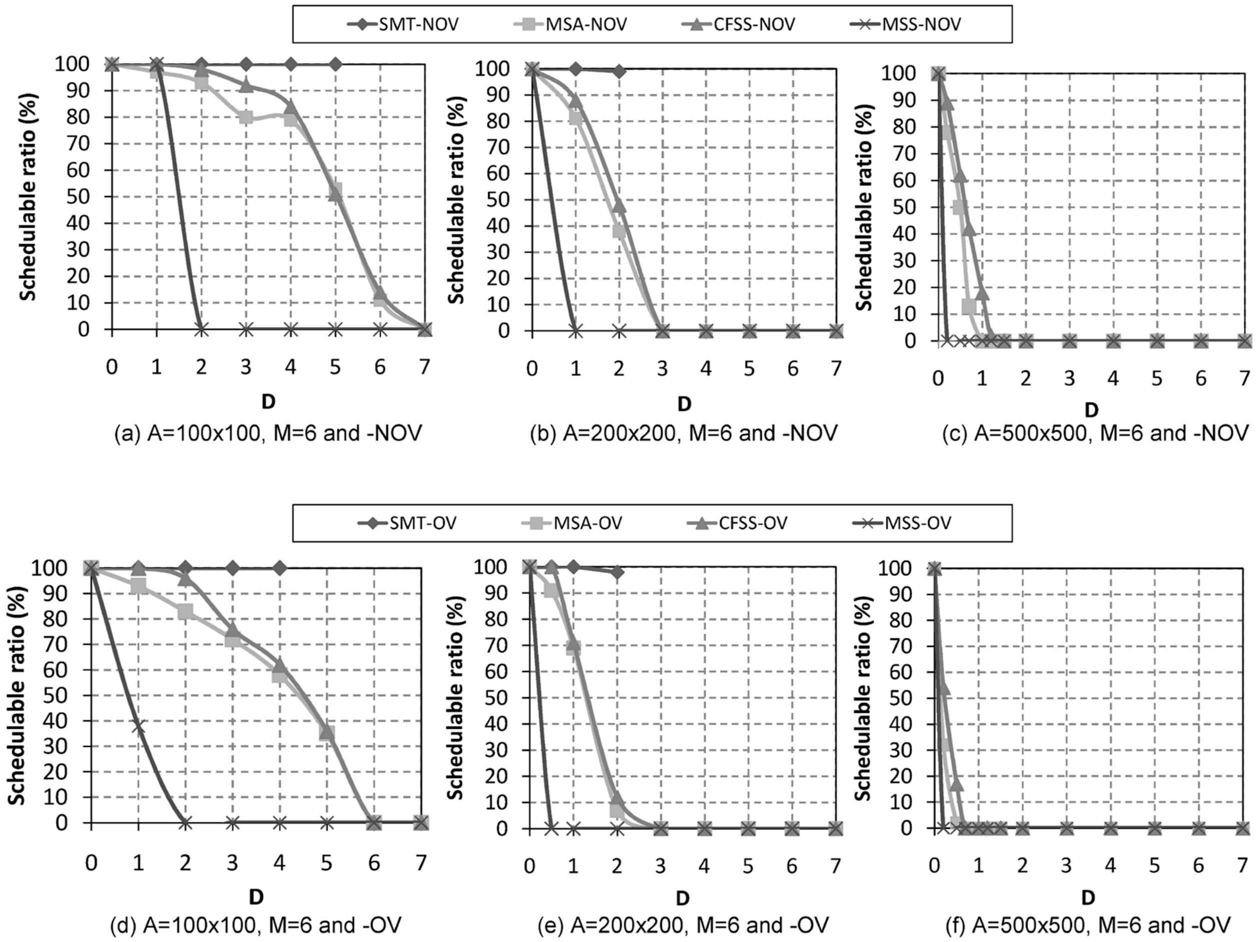
Figure 7 Running time for the experiments in Figure 6: (a) all of results and (b) a part of (a)

7 Real implementation

To show the feasibility of our proposal, we implement a testbed based on the wireless network for industrial automation-process automation standard (WIA-PA) (Liang et al., 2011; Wang et al., 2011). WIA-PA follows the ISO/OSI 7-layer model, and its physical layer and MAC layer are based on the IEEE 802.15.4 protocol.

In WIA-PA, five types of physical devices are defined. But we only use three of them: gateway device, routing device and field device. The gateway device (WIA gateway, http://www.wia.org.cn/en/03.asp?pd=cp&id=65&anclassid=14&nclassid=628), corresponding to the PAN coordinator, adopts a low power SoC AT91RM9200 and a CC2420 transceiver chip. It manages the

Figure 8 Schedulable ratio under varying network densities



WIA-PA network and connects the WIA-PA network with other plant networks. The routing device (WIA router, http://www.wia.org.cn/en/03.asp?pd=cp&id=64&anclassid=14&nclassid=628), which is implemented on a MSP430 and a CC2420, corresponds to other FFDs except the PAN coordinator. Besides a MSP430 and a CC2420, field device also equips a temperature and humidity sensor SHT15, and acts as a RFD. We modify the WIA-PA protocol stack according to our requirements. The algorithm CFSS is an effective and lightweight, which has been proven by the above evaluation results. So, as the online scheduling algorithm, CFSS is implemented in the PAN coordinator. Channel 23 is used to add new devices. Six schedulable channels are 15–20. Additionally, six routing devices are configured as sniffers to monitor beacon packets transmitted on the six channels. Then the sniffed packets with a timestamp are sent to a PC via a 8-port RS-232 PCI Express serial board.

Figure 10 shows our testbed. Ten clusters are deployed in a building. Their BOs are both equal to 6, and $\{SO_1, ...SO_{10}\}$ are set to $\{4, 3, 2, 3, 1, 2, 1, 2, 1, 1\}$. The network runs as follows:

- The gateway first comes to life, and then permits other devices to join the network via channel 23.
- The routing devices send their superframe configurations to the gateway via the tree network.
- According to the network topology and the received superframe configurations, the gateway runs our CFSS algorithm to find a feasible scheduling, and then sends the scheduling information to each routing device.
- The gateway and routing devices periodically schedule their superframes. Figure 11 shows the beacon packets which are sniffed when the network reaches a steady state. From the figure, we can see that beacon frames can be scheduled without collisions.

Figure 9 Running time for the experiments in Figure 8(a)

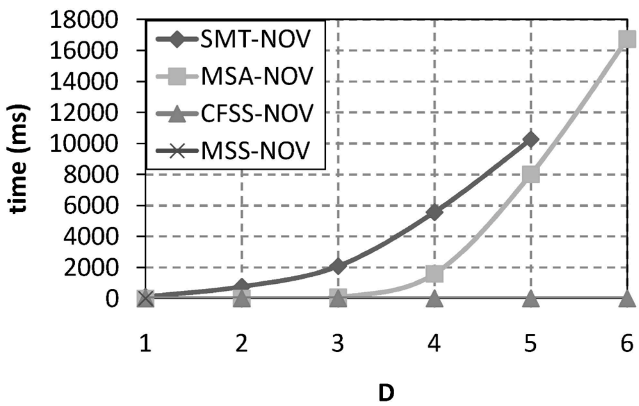


Figure 10 Our testbed

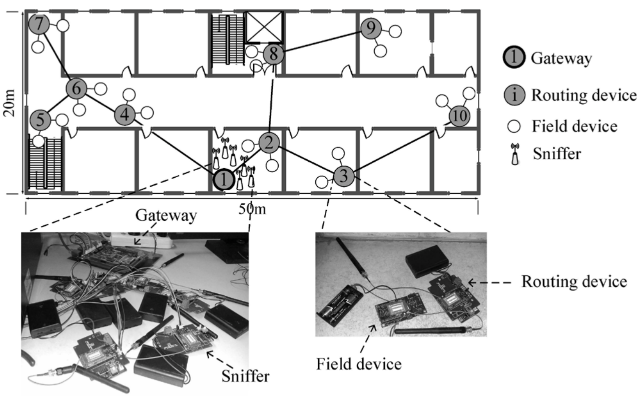
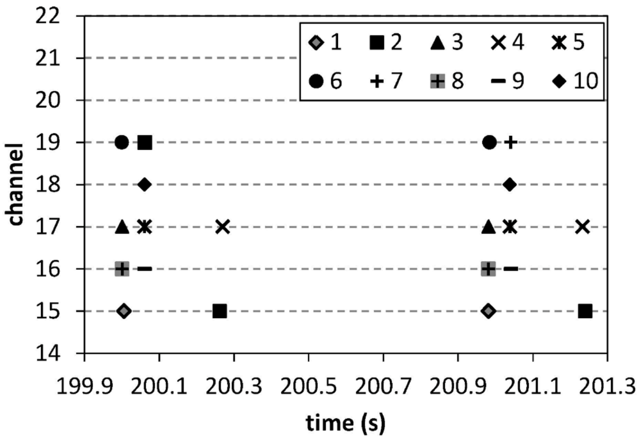


Figure 11 Beacon frames



8 Conclusion

In this paper, we studied the collision-free multichannel superframe scheduling algorithms for IEEE 802.15.4 cluster-tree networks. According to the features of superframe collisions, we gave three essential scheduling constraints, which ensured that superframes are scheduled without collisions, and analysed two necessary conditions on the schedulability of a superframe set. Then we formulated the superframe scheduling problem as an SMT instance. Since the scalability of an purely SMT-based approach is very limited, we proposed two heuristic algorithms to improve the scalability. Experimental results shown that our proposed methods outperform existing multichannel superframe scheduling algorithm. Finally, to show the feasibility of our proposal, we implemented a real system based on a WIA-PA network.

Acknowledgements

This work was partially supported by the National High Technology Research and Development Program of China (863 Program: No. 2011AA040103), the Important National Science and Technology Specific Project No. 2013ZX03005004 and the Natural Science Foundation of China No. 61100023.

References

Barrett, C. and Tinelli, C. (2007) 'Cvc3', *Computer Aided Verification*, Springer, pp.298–302.

Barrett, C., Sebastiani, R., Seshia, S. and Tinelli, C. (2009) 'Satisfiability modulo theories', *Handbook of Satisfiability*, Vol. 185, Chapter 26, pp.825–885.

Bello, L.L. and Toscano, E. (2009) 'Coexistence issues of multiple co-located IEEE 802.15.4/Zigbee networks running on adjacent radio channels in industrial environments', *IEEE Transactions on Industrial Informatics*, Vol. 5, No. 2, pp.157–167.

Chen, S., Almeida, L. and Wang, Z. (2010) 'A dynamic dual-rate beacon scheduling method of ZigBee/IEEE 802.15. 4 for target tracking', *The 6th International Conference on Mobile Ad-hoc and Sensor Networks*, Hangzhou, China, pp.103–109.

Cook, S. A. (1971) 'The complexity of theorem-proving procedures', *The Third Annual ACM Symposium on Theory of Computing*, Ohio, USA, pp.151–158.

Davis, M. and Putnam, H. (1960) 'A computing procedure for quantification theory', *Journal of the ACM*, Vol. 7, No. 3, pp.201–215.

Davis, M., Logemann, G. and Loveland, D. (1962) 'A machine program for theorem-proving', *Communications of the ACM*, Vol. 5, No. 7, pp.394–397.

Dutertre, B. and Moura, L.D. (2006) *System Description: Yices 1.0*, The 2nd SMT competition.

Franchino, G. and Buttazzo, G. (2012) 'WBuST: A real-time energy-aware MAC layer protocol for wireless embedded systems', *The 17th IEEE Conference on Emerging Technologies & Factory Automation*, Krakòw, Poland, pp.1–8.

Ganzinger, H., Hagen, G., Nieuwenhuis, R., Oliveras, A. and Tinelli, C. (2004) *DPLL(T): Fast Decision Procedures*, Computer Aided Verification, Springer, pp.175–188.

Gao, B. and He, C. (2008) 'An individual beacon order adaptation algorithm for IEEE 802.15.4 networks', *The 11th IEEE International Conference on Communication Systems*, Krakòw, Poland, pp.12–16.

Hanzalek, Z. and Jurcik, P. (2010) 'Energy efficient scheduling for cluster-tree Wireless Sensor Networks with time-bounded data lows: application to IEEE 802.15.4/ZigBee', *IEEE Transactions on Industrial Informatics*, Vol. 6, No. 3, pp.438–450.

Huang, J., Blech, J. O., Raabe, A., Buckl, C. and Knoll, A. (2012) 'Static scheduling of a time-triggered Network-on-Chip based on SMT solving', *The Conference on Design, Automation and Test in Europe*, Dresden, Germany, pp.509–514.

IEEE-SA Standards Board (2012) *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std 802.15.4e-2012.

Jeon, J., Lee, J.W., Ha, J.Y. and Kwon, W.H. (2007) 'DCA: duty-cycle adaptation algorithm for IEEE 802.15.4 beacon-enabled networks', *The 65th IEEE Conference on Vehicular Technology*, Dublin, Ireland, pp.110–113.

Kim, Y.D, Yang, Y.M., Kang, W.S. and Kim, D.K. (2014) 'On the design of beacon based wireless sensor network for agricultural emergency monitoring systems', *Computer Standards & Interfaces*, Vol. 36, No. 2, pp.288–299.

Koubâa, A., Alves, M. and Tovar, E. (2006a) 'i-GAME: an implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks', *The 18th Euromicro Conference on Real-Time Systems*, Dresden, Germany, pp.10–20.

Koubâa, A., Alves, M. and Tovar, E. (2006b) 'GTS allocation analysis in IEEE 802.15. 4 for real-time wireless sensor networks', *The 20th International Parallel and Distributed Processing Symposium*, Rhodes, Island, Greece, pp.76–83.

Koubâa, A., Cunha, A. and Alves, M. (2007) 'A time division beacon scheduling mechanism for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks', *The 19th Euromicro Conference on Real-Time Systems*, Pisa, Italy, pp.125–135.

Koubâa, A., Alves, M., Tovar, E. and Cunha, A. (2008) 'An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: theory and practice', *Real-Time Systems*, Vol. 39, No. 1-3, pp.169–204.

Liang, W., Zhang, X, Xiao, Y., Wang, F., Zeng, P. and Yu, H. (2011) 'Survey and experiments of WIA-PA specification of industrial wireless Network', *Wireless Communications and Mobile Computing*, Vol. 11, No. 8, pp.1197–1212.

Moura, L. D. and Bjørner, N. (2008) 'Z3: An efficient SMT solver', *The 14th International Conference Tools and Algorithms for the Construction and Analysis of Systems*, Budapest, Hungary, pp.337–340.

Neugebauer, M., Plonnigs, J. and Kabitzsch, K. (2005) 'A new beacon order adaptation algorithm for IEEE 802.15. 4 networks', *The 2nd European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, pp.302–311.

Pan, M.S. and Tseng, Y.C. (2008) 'Quick convergecast in Zigbee beacon-enabled tree-based wireless sensor networks', *Computer Communications*, Vol. 31, No. 5, pp.999–1011.

Rai, A., Bhagwan, R. and Guha, S. (2012) 'Generalized resource allocation for the cloud', *ACM Symposium on Cloud Computing*, New York, USE, pp.15–26.

Toscano, E. and Bello, L.L. (2009) 'A multichannel approach to avoid beacon collisions in IEEE 802.15.4 cluster-tree industrial networks', *The 14th IEEE Conference on Emerging Technologies & Factory Automation*, Mallorca, Spain, pp.1–9.

Toscano, E. and Bello, L.L. (2012) 'Multichannel superframe scheduling for IEEE 802.15.4 industrial wireless sensor networks', *IEEE Transactions on Industrial Informatics*, Vol. 8, No. 2, pp.337–350.

Wang, F., Zeng, P. and Yu, H. (2011) 'An energy-efficient time synchronization algorithm for wireless sensor networks', *Information and Control*, Vol. 40, No. 6, pp.753–759.

Xiao, Y., Li, H. and Choi, S. (2009) 'Two-level protection and guarantee for Multimedia Traffic in IEEE 802.11e Distributed WLANs, *ACM/Springer Wireless Networks*, Vol. 15, No. 2, pp. 141-161.

Xiao, Y., Li, H. and Li, B. (2007) 'Bandwidth sharing schemes for multimedia traffic in the IEEE 802.11e contention-based WLANs, *IEEE Transactions on Mobile Computing*, Vol. 6, No. 7, pp.815-831.

Yeh, L.W., Pan, M.S. and Tseng, Y.C. (2008) 'Two-way beacon scheduling in Zigbee tree-based wireless sensor networks', *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taichung, Taiwan, pp.130–137.

Websites

The Satisfiability Modulo Theories Competition, http://smtcomp.sourceforge.net/2012/

WIA gateway, http://www.wia.org.cn/en/03.asp?pd=cp &id=65& anclassid=14&nclassid=628

WIA router, http://www.wia.org.cn/en/03.asp?pd=cp&id=64&anclas sid=14& nclassid=628