Virtual Channel Management for Densely Deployed IEEE 802.15.4 LR-WPANs*

Tae Hyun Kim[1], Jae Yeol Ha[2], Sunghyun Choi[3], and Wook Hyun Kwon[4]
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea
Emails: [1]thkim@mwnl.snu.ac.kr, [3]schoi@snu.ac.kr, and {[2]hjy8099, [4]whkwon}@cisl.snu.ac.kr

Abstract

The number of channels specified for IEEE 802.15.4 Low-Rate Wireless Personal Area Networks (LR-WPANs) is too few to operate many applications of WPANs in the same area. To overcome this limit, we introduce Virtual Channel, *a novel concept to increase the number of available channels when various WPAN applications coexist. A virtual channel is basically created via superframe scheduling within the inactive periods in a logical channel preoccupied by other WPANs. To maximize the coexistence capability of WPANs using virtual channels, we propose (1) Least Collision superframe scheduler (LC-scheduler), (2) less complex heuristics, and (3) Virtual Channel Selector (VCS) to efficiently manage multiple available logical channels. In addition, a simple but practical synchronization method is developed to compensate different time drifts among coexisting WPANs. The simulation results demonstrate that a remarkable improvement on the coexistence capability of the 802.15.4 can be achieved through the proposed schemes.*

1 Introduction

IEEE 802.15.4 for Low-Rate Wireless Personal Area Networks (LR-WPANs) [1] has been standardized for low-rate, low-cost, low-power, and short-range wireless networking. We expect that this standard, which defines both Physical (PHY) and Medium Access Control (MAC) layers, would be widely adopted for various applications including industrial automation, home control, cable replacement, and wireless sensor networks. One can easily imagine that many applications of this standard, as enumerated in Table 1, would operate simultaneously in a given area so that the corresponding WPANs coexist in the same area.

However, the available logical channels are very limited.[1] That is, the current IEEE 802.15.4 specifies 27 logical channels across 868MHz, 915MHz, and 2.4GHz frequency bands. Especially, in the case of 2.4GHz ISM band, other wireless technologies, such as IEEE 802.11 Wireless Local Area Networks (WLAN) and IEEE 802.15.1 Bluetooth share the same frequency band. As discussed in [1, 2], considering the coexistence between IEEE 802.15.4 and IEEE 802.11, only four available channels (out of 16) remain for the 802.15.4 networks. Despite this coexistence problem, the recent literature has been limited to the analysis of the 802.15.4's performance degradation resulted by other coexisting wireless technologies. In [3, 4], the authors analyze the packet error rate of the IEEE 802.15.4 devices, coexisting with an 802.11 WLAN. Based on the analysis, they conclude that a careful channel selection and assignment resolve the coexistence issue.

In fact, the adoption of a carrier sense medium access with collision avoidance (CSMA-CA) algorithm in IEEE 802.15.4 can ameliorate this coexistence problem. Thanks to the inherent nature of CSMA-CA, coexisting WPANs are able to operate simultaneously even if devices from those WPANs in the same transmission range attempt to access the medium. However, as the number of such devices increases, the networks would suffer from severe performance degradation. Therefore, the separation among coexisting WPANs either temporally or spectrally is definitely required to guarantee a reliable operation of the networks.

In this paper, to enhance such a limited channel availability, we introduce a novel concept, called *Virtual Channel*, into IEEE 802.15.4 to increase available channels. Basically, a virtual channel is a newly-created conceptual channel via superframe scheduling within the inactive periods in a logical channel preoccupied by other WPANs, which parallels the adoption of "temporal agility" for IEEE 802.15.4 networks. To maximize the coexistence capability of WPANs, we develop *Least Collision superframe scheduler (LC-scheduler)* and two heuristic algorithms, *Simpli-*

*This research was in part supported by University ITRC. Tae Hyun Kim and Sunghyun Choi are with the School of Electrical Engineering and INMC, Seoul National University, Seoul 151-744, Korea.

[1] In the 802.15.4 PHY specification, the term 'logical channels' represents disjoint physical frequency channels. An 802.15.4 WPAN operates in a logical channel.

Table 1. IEEE 802.15.4 Target Applications [5]

Category	Application	Delay	BO
Vital Monitoring	Heart-rate monitor	1-5 sec	6-8
	Body heat monitor	1 min	12
	Personal equipment control	50 ms	2
Consumer Electronics	Remote controls	100 ms	3
	PC-peripherals	50 ms	2
	Control of blinds/shades/ rollers/windows	1 sec	6
	Dimmer/switches	200 ms	4
Automatic Meter Reader	Electricity/gas/water AMR	No bound	14
Alarm/Security System	Smoke detector	1 sec	6
	Burglary and social alarms	1 sec	6
	Access control	1 sec	6
	Water leakage alarms	1-5 sec	6-8
Environmental Monitoring	Temperature/carbondioxide/ humidity/vibration, HVAC	1-5 min	12-14
Industrial Automation	Facility control	100 ms	3
	Monitoring critical equipment	1 sec	6

fied LC-scheduler (SimLC-scheduler) and *NEarest Vacancy Search (NEVS)* under the assumption that only one logical channel is given to the scheduler.

There have been remarkable research efforts for different scheduling algorithms in both wireline and wireless networks so far. For example, in the field of IEEE 802.11 WLAN research, scheduling algorithms have been developed for the purpose of quality of service (QoS) provisioning and energy saving [10, 11, 12]. Nevertheless, the considered superframe scheduling problem is fundamentally different from the existing scheduling problems due to such distinct constraints as (1) binary exponential difference between beacon intervals with different beacon order values and (2) the lack of an interactive exchange of the required information caused by its inter-network scheduling characteristics.

Shifted to the case that multiple logical channels are available, e.g., 4 logical channels as discussed above, the problem to select one of those logical channels should be also considered. By using the investigated characteristics of the applications in Table 1, we design *Virtual Channel Selector (VCS)*, which offers more efficient virtual channel management. Lastly, we consider a time drift problem and provide *Inter-WPAN Synchronization (IWS)* since all our proposals are based on the perfect time synchronization among coexisting WPANs in the same logical channel. All these proposals are validated with simulations from which the results evince that our proposed schemes remarkably improve the coexistence capability of IEEE 802.15.4 while, at the same time, sustaining the backward compatibility.

The rest of this paper is organized as follows: Section 2 presents the system model considered in this paper. In Section 3, we present the proposed superframe schedulers, and then, by utilizing multiple available logical channels, VCS is developed in Section 4. In Section 5, the time synchronization problem is discussed, and IWS is proposed. The proposed algorithms are comparatively evaluated via computer simulations in Section 6. Then, we finally conclude the paper in Section 7.

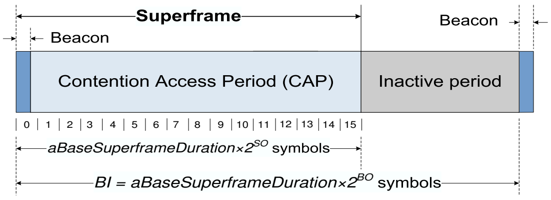
Superframe
Beacon
Beacon
Contention Access Period (CAP)
Inactive period
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15
aBaseSuperframeDuration×2^SO symbols
BI = aBaseSuperframeDuration×2^BO symbols

Figure 1. Superframe structure

2 System Model

IEEE 802.15.4 has two different operating modes with respect to the usage of periodic beacons, i.e., nonbeacon-enabled and beacon-enabled modes, respectively. The nonbeacon-enabled mode entails much more energy consumption than the other due to the absence of periodic sleep coordination. Meanwhile, beacon-enabled networks are coordinated by periodic beacons by which the time for sleeping is informed. We, basically, assume that most WPAN applications operate in beacon-enabled mode to conserve limited battery power. However, WPAN operating in nonbeacon-enabled mode can also adopt our scheme as a beacon-enabled WPAN with the duty cycle of 100%.

For beacon-enabled mode, the medium access strictly follows the superframe structure depicted in Fig. 1, and its parameters are controlled by a periodic beacon packet from the coordinator. A superframe is composed of a beacon and contention access period (CAP), which is followed by an inactive period. The length of the superframe is controlled by superframe order (SO) while beacon interval (BI) is determined by beacon order (BO). In this paper, we use the term ‘active period’ and ‘superframe’ interchangeably since we assume that all WPANs operate only in beacon-enabled mode. Note that periodic beacons are not transmitted based on CSMA-CA unlike other packets.

IEEE 802.15.4 supports *spectral agility* by dividing frequency band into multiple sub-bands, called, logical channels. Each WPAN can occupy one of these channels to operate without any interference from other WPANs. However, the occupied logical channel by one WPAN application does not tend to be fully utilized for the communications. For example, automatic meter readers might require very low duty cycle of 0.1% so that 99.9% of time resource is not utilized. *To utilize this under-utilized time resource, temporal agility can be introduced for the channel management of IEEE 802.15.4.*

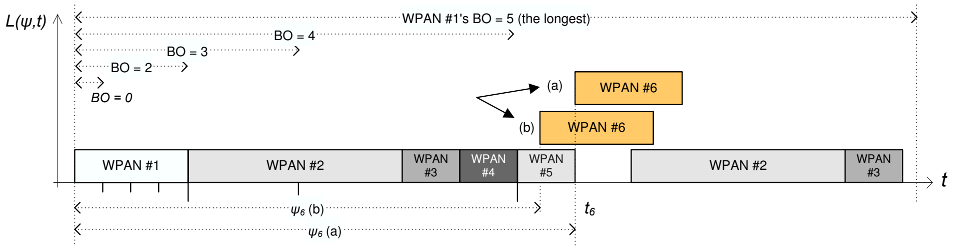

Figure 2. Example of virtual channel utilization; time offset (a) leads less beacon collisions than (b).

As a result, we argue that a new channel should be created by using the under-utilized time resource for other WPANs. This newly-created channel is referred to as a "virtual channel." Specifically, a virtual channel is created within the inactive periods of the channel preoccupied by other WPANs. Fig. 2 shows the logical channel containing five virtual channels, and it is about to create one more for an incoming WPAN, i.e., WPAN #6. In this figure, each block labeled by a WPAN represents the corresponding WPAN's superframe, and the horizontal line is a time axis representing one logical channel. Note that each block repeats over time with the period of "beacon interval."

The beauty of the proposed virtual channel is that a virtual channel not only increases the number of available channels significantly, but also is easily managed only by PAN coordinators in a completely distributed manner. What a PAN coordinator should perform for the virtual channel management is to select "a proper time offset" and "logical channel" for its beacon transmissions. Once the periodic beacon transmission starts, all child devices attempting to associate with the WPAN synchronizes with the superframes of the PAN coordinator.

All the proposals in this paper utilize two MAC-Layer Management Entity (MLME) primitives in the 802.15.4: MLME-START.request for starting the first beacon transmission at a specific time and MLME-SCAN.request to collect the information of the preoccupying WPANs in certain logical channels. In fact, IEEE 802.15.4 does not specify any time parameter to control the first beacon transmission time.[2] However, to manage the proposed virtual channels, the starting time of the first beacon must be under control. For this, we assume that adjusting the time to call MLME-START.request enables us to control the time for starting the first beacon transmission.

3 Superframe Scheduler

In this section, we analyze the "proper time offset" problem and propose *Least Collision superframe scheduler (LC-scheduler)*, *Simplified LC-scheduler (SimLC-scheduler)*, and *NEarest Vacancy Search (NEVS)*. Basically, these superframe schedulers assume that there exists only one available channel since the consideration of multiple available channels requires a different strategy as will be discussed in the next section.

3.1 Metric for Problem Formulation

Thanks to the CSMA-CA of IEEE 802.15.4, the overlap of the superframes gradually degrades the performance rather than shutting each coexisting WPAN's operation down. Once some overlap occurs, the consequent performance degradation is originated by two factors: (1) beacon collisions and (2) the increase of effective contending devices. In the case of beacon collisions, it cannot be avoided once the superframes start overlapping. These collisions have their roots in the fixed beacon transmission times of IEEE 802.15.4. Note that the beacons are transmitted periodically without using CSMA/CA. Therefore, an incoming WPAN, which cannot find enough time resource guaranteeing no overlap with existing others, shall experience intermittent beacon collisions.

On the other hand, in a CSMA-CA network, the number of contending devices heavily affects the network performance such as throughput, delay, and so forth. Fortunately, even though the contending devices increase, it does not spoil all devices' channel accesses if the overlapped WPANs are synchronized with respect to the backoff period.[3] The level of performance degradation due to the increased effective contenders can be roughly measured using a new performance metric, defined based on our previous saturation throughput analysis of IEEE 802.15.4 [7].

To examine the level of additional competition caused by superframe overlaps, we define an instant collision probability p_c of a transmission by adopting the probability P_s of a successful transmission when a transmission procedure

[2] The emerging 802.15.4b [6], which is currently being standardized, would have such a function.

[3] A backoff period indicates the smallest time slot used for any channel access during CAP. Asynchronous backoff periods among WPANs can cause a severe malfunction of clear channel assessments if these WPANs are overlapped.

Table 2. Definition of Parameters

$u(t)$	a unit step function
$\delta(t)$	a delta function
N_j	the number of contending devices in j-th WPAN
ψ	a relative time offset
n_p	the number of WPANs in the logical channel including the incoming WPAN, also the index of incoming WPAN
D	a base superframe length [1] (=960 symbols)
BO_j	the beacon order of the j-th WPAN in the logical channel
BO_{max}	maximum beacon order among all WPANs'
SO_j	the superframe order of the j-th WPAN in the channel

starts, derived by τ in [7].

$$p_c(n) = 1 - P_s(n) = 1 - \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}, \qquad (1)$$

where n is the number of active contending devices including the device itself and τ is the conditional probability to perform the first clear channel assessment (CCA) when all other devices are in the backoff state.[4] Specifically, in Eq. (1), τ is a function of the MAC frame length as well as n, not a function of only n. However, we assume that the fixed-size frames are transmitted in every WPAN for simplicity.

Note that p_c is originally a steady-state probability, which cannot describe the instant collision level. Nevertheless, it can represent a relative level of contention by using the total number of contending devices within the overlaps. Accordingly, p_c is useful to investigate the superframe overlap effects, not to measure a specific absolute performance. That is, only a relative comparison of two p_c values is meaningful.

3.2 LC-scheduler

We here develop *Least Collision superframe scheduler (LC-scheduler)* to minimize both of (1) beacon collisions and (2) the increase of effective contending devices.

In order to identify how each WPAN occupies a channel, we define channel occupancy function $A_j(\psi, t)$ for WPAN j as follows:

$$A_j(\psi,t) = \sum_{m=-\infty}^{\infty} [u(t-\psi-m\cdot D\cdot 2^{BO_j}) - u(t-\psi-D\cdot(2^{SO_j}+m\cdot 2^{BO_j}))], \qquad (2)$$

where $0 \le \psi < D\cdot 2^{BO_j}$ and $j = 1, 2, \cdots, n_p$. The notations used in all equations are defined in Table 2. Two unit step functions shape a box representing a WPAN's superframe as shown in Fig. 2. Its height is one, and it occupies the channel periodically for the superframe duration of the WPAN. Its periodicity is represented by the infinite summation operator.

The time offset, ψ, indicates the relative time of the corresponding WPAN's superframe with respect to the beacon transmission time of the WPAN with the largest beacon order (BO) in the channel. For example, the incoming WPAN #6 in Fig. 2 is about to select ψ_6 (a) or (b). Since WPAN #1 has the largest BO, time offset ψ_6 is the interval from the beacon transmission time of WPAN #1 to the determined transmission time of the first beacon of WPAN #6, e.g., t_6. Therefore, the superframe scheduling itself is to choose a time offset ψ, which incurs the minimum performance loss. Additionally, due to the binary exponential increase of the beacon interval, as the beacon order increases by one, we do not need scan the whole time axis to find the proper time offset for the scheduling. The necessary scanning range for ψ is reduced to the beacon interval of the incoming WPAN[5].

While $A_j(\psi, t)$ describes a periodic superframe of one WPAN in a certain logical channel, all the previously occupying WPANs and the incoming WPAN are simultaneously portrayed by aggregate channel occupancy function $L(\psi, t)$ given as:

$$L(\psi,t) = \sum_{j=1}^{n_p-1} A_j(\psi_j, t) + A_{n_p}(\psi, t), \qquad (3)$$

where $0 \le \psi < D\cdot 2^{BO_{n_p}}$. The summation term in Eq. (3) represents the WPANs residing in the channel already. Adding the incoming WPAN's individual channel occupancy function to it, we can obtain the aggregate channel occupancy function as depicted in Fig. 2. Technically, $L(\psi, t)$ represents the number of coexisting superframes at a specific time t when ψ is given.

With this function, we find a set of time offset ψ's, which incur the least possible beacon collisions. The main idea how to find such a set is depicted in Fig. 2 (a) and (b). There pre-exist 5 WPANs whose BO and superframe order (SO) pairs are (5,2), (4,3), (4,1), (5,1) and (5,1), respectively. An incoming WPAN with (5,2) is looking for the least overlapping location. Although both of the locations (a) and (b) incur the same amount of overlapping time period, (b) has a higher beacon collision probability than (a) since the beacons are transmitted at the beginning of superframes. Analytically, the set of ψ's resulting in superframes stuck to each other, e.g., Fig. 2 (a), ensures the least beacon collisions. To define the set Ψ, comprising ψ's which result in the least possible beacon collisions, we consider the channel occupancy $Q(t)$ of all existing WPANs, given by

$$Q(t) = L(\psi, t) - A_{n_p}(\psi, t).$$

[4] Under the 802.15.4 CSMA-CA, the CCA is performed twice before a transmission attempt.

[5] The least common multiple of beacon intervals is always the longest beacon interval among them.

Since ψ should indicate the time offsets corresponding to the steep increase, or $\psi + D \cdot 2^{SO_{n_p}}$ should indicate the steep decrease of $Q(t)$, we differentiate $Q(t)$ in order to find the discontinuous points of $Q(t)$.

$$Q'(t) = \sum_{i \in I_p} \alpha_i \cdot \delta(t - t_i) + \sum_{i \in I_n} \beta_i \cdot \delta(t - t_i),$$

where $I_p = \big\{ i | t_i$ is the time of positive infinity occurrence and $\lim_{d \to 0} Q'(t_i - d) = 0 \big\}$ and I_n for negative infinity, respectively. α_i and β_i are the corresponding coefficients of $\delta(t - t_i)$, respectively. By the limit condition in I_p and I_n definitions, unreasonable candidates, which would potentially cause more beacon collisions, are excluded. Using above I_p and I_n, we can obtain Ψ as follows:

$$\begin{aligned} \Psi_n &= \Big\{ t_i \; rmod \; 2^{BO_{np}} | i \in I_n \Big\}, \\ \Psi_p &= \Big\{ (t_i - D \cdot 2^{SO_{n_p}}) \; rmod \; 2^{BO_{np}} | i \in I_p \Big\}, \\ \Psi &= \Psi_n \cup \Psi_p \setminus \big\{ \psi_j | j = 1, 2, \cdots, n_p - 1 \big\}, \end{aligned} \tag{4}$$

where $rmod$ represents the modulo operator for real number. By using this $rmod$, we can limit our interests only to the beacon interval of the incoming WPAN. The set subtraction means that newly selected time offset shall not be the same as the existing ones; the same time offset tends to cause the periodic beacon collisions with very high probability.

During the above operations, many candidates resulting in no beacon collisions are also excluded. Suppose that, for instance, the length of the inactive interim between WPAN #5 and #2 is as twice long as incoming WPAN #6's superframe length. Even except for two ψ's rendering the incoming superframe stuck to the anterior WPAN #5's superframe (ψ_a) or the posterior WPAN #2's (ψ_p), many time offsets between ψ_a and ψ_p can guarantee no collisions of beacons. However, the best ψ among those in terms of the coexistence of all WPANs can only be known when the whole BOs and SOs of future incoming WPANs are known, and it is practically impossible. Virtual Channel Selector (VCS) in the subsequent section considers this. For superframe schedulers collaborating with VCS, ψ's to make incoming WPAN's superframe stuck to the previously residing superframes, such as ψ_a and ψ_p, can maximize the efficiency of VCS.

Now, in order to address the issue of the increased contention due to overlapping WPANs, we define the channel occupancy function $W(\psi, t)$ by reflecting the increase of the contending neighbors within the overlapped periods.

$$W(\psi, t) = A_{n_p}(\psi, t) \sum_{j=1}^{n_p} N_j \cdot A_j(\psi_j, t), \tag{5}$$

where $0 \leq \psi < D \cdot 2^{BO_{n_p}}$. The summation in Eq. (5) is to consider the existing WPANs in the logical channel and the incoming WPAN with a given ψ. By multiplying $A_{n_p}(\psi, t)$, we can eliminate the superframe portions, which do not overlap with the incoming WPAN's superframes since they are out of incoming WPAN's concern. With this function, the superframe scheduling is finally formulated as a minimization problem seeking for the time offsets, incurring the least collision among the packets from different WPANs.

$$\begin{aligned} \tilde{\Psi} = \Big\{ \psi | \arg\min_{\psi} H(\psi) = \frac{1}{T} \int_T [p_c(W(\psi, t))] dt, \\ \text{s.t. } \psi \in \Psi, \; T = D \cdot 2^{BO_{max}} \Big\}, \end{aligned} \tag{6}$$

where $p_c(\cdot)$ is given by Eq. (1). Note that this minimization problem involves only a single unknown variable, ψ. Every other parameter can be acquired from the PAN descriptors reported by the scanning procedure.

Even after the above minimization, we have a possibility that $\tilde{\Psi}$ still has more than one element. The choice from the multiple candidates should be made while taking the next incoming WPAN into account, yet the information cannot be known. In LC-scheduler, the earliest time offset is chosen for the delay reduction of the newly-starting WPAN.

In fact, time t in the searching space does not need to be continuous because both superframe duration and beacon interval are multiples of $aBaseSuperframeDuration$ defined in the standard. It is obvious that the resolution of t, more precise than $aBaseSuperframeDuration$, cannot provide any ψ by which less collision is ensured. Thus, we adopt the discrete time unit equal to $aBaseSuperframeDuration$, and this reduces the searching space of Eq. (6) while guaranteeing no miss of a better ψ selection.

3.3 Heuristic Algorithms

For the practical usage, we simplify LC-scheduler and propose *Simplified LC-scheduler (SimLC-scheduler)* by assuming (1) the fixed number of devices in every WPAN and (2) a limited length, represented as BO_{limit}, for the consideration of the channel status in $W(\psi, t)$. For the fixed device number, since the practical algorithm cannot know the actual number of activated devices in each WPAN, instead of using N_j, this heuristic algorithm simply assumes that N_j for every j is equal to a certain fixed value, N_{fix}.

Using BO_{limit} further simplifies the algorithm. In solving the minimization problem in Eq. (6), we have so large search space that lots of memories are required to build the channel status arrays such as $W(\psi, t)$. For example, when the BO of a WPAN in a logical channel is 14, 2^{14} bytes are required and the memory needs to be searched for every $\psi \in \Psi$ to execute the minimization. Limiting BO_{max} by

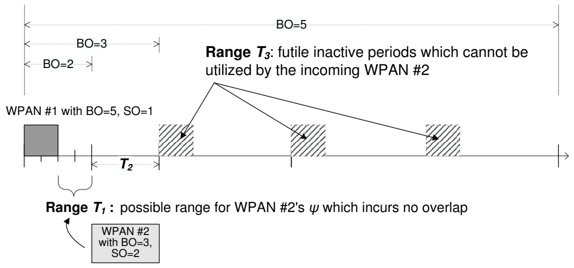
BO=5
BO=3
BO=2
Range T_3: futile inactive periods which cannot be utilized by the incoming WPAN #2
WPAN #1 with BO=5, SO=1
T_2
Range T_1 : possible range for WPAN #2's ψ which incurs no overlap
WPAN #2 with BO=3, SO=2

Figure 3. Futile inactive periods

BO_{limit} enables not only a faster execution of the algorithm, but also less memory requirement. In many cases, this limitation works generously because the periodicity of superframes degrades the probability to find a proper time offset at the tail of the searching interval if an offset could not be found during the head of the interval. Note that even when BO_{limit} is set to 13, the searching space is reduced to a half of the worst case, i.e., when one of the residing WPANs' BOs is 14.

One more heuristic is *NEarest Vacancy Search (NEVS)*. The main idea of this heuristic is simple. After building $L(\psi, t)$, NEVS searches the closest vacancy, which corresponds to the inactive period, for the incoming WPAN's superframe. If there is not enough vacancy for its superframe, NEVS searches the longest available vacancy and determine ψ. In the worst case, no inactive period in the channel makes NEVS deny to create a virtual channel. This heuristic is intuitively simpler than SimLC-scheduler due to the absence of the procedure to compute $W(\psi, t)$. Additionally, BO_{limit} concept is also employed to decrease the vacancy searching space.

4 Virtual Channel Selector (VCS)

4.1 Principles for VCS

Using the previously proposed superframe scheduler, i.e., LC-scheduler and etc., the most efficient virtual channel for coexistence can be found in a given logical channel. As for multiple available logical channels, the logical channel offering the lowest $H(\psi)$ among all of them, used in Eq. (6), can be simply selected to create a virtual channel. However, this approach might lead to the loss of time resources depending on the potential incoming WPANs in the future. In this subsection, we provide considerable observations on this problem, and establish three principles to design a *Virtual Channel Selector (VCS)*.

Once potential incoming WPANs in the future are considered, the superiority among the available virtual channels becomes ambiguous. Specifically, this ambiguity is grounded on the unknown BOs of the potential incoming WPANs. The BO differences among the potential incoming WPANs and existing WPANs constrain the number of ψ candidates for the newly incoming WPAN as shown in Fig. 3. In this figure, WPAN #1 is already operating in one channel when WPAN #2 tries to create a virtual channel in that logical channel. Since WPAN #2 has much shorter beacon interval than that of WPAN #1, it is possible that ψ's can be only within the range marked as T_1; other ψ's definitely incur superframe overlap with the superframe of WPAN #1 since the incoming WPAN #2's superframe length is equal to T_2. As a result, futile inactive periods are induced as the range marked as T_3 in the figure. That kind of futile inactive periods, which cannot be used for other WPANs, would gradually increase, if WPANs with different BOs keep entering. Reversely, as an ideal case, if there are sufficiently many logical channels, the scheduling efficiency can be maximized by placing virtual channels of the same BO into the same logical channel, which incurs no futile periods. From this fact, we conclude that the more similar the BOs are in the same logical channel, the more WPANs can coexist smoothly.

Fortunately, the predictable incoming WPANs' BOs are quite limited. In Table 1, we can assess that the BOs of the predictable applications comprise approximately three groups: BOs equal to 2, 6 and 12. Exploiting this fact, the efficiency of the virtual channel management can be enhanced by grouping the WPANs of the similar BOs into the same logical channel.

Furthermore, we find that the adjustment of the original BOs and SOs are allowable under certain conditions. Fig. 4(a) and 4(b), from the ns-2 simulations, show the performance of IEEE 802.15.4 star-topology network with 10 child devices and a PAN coordinator under 50% duty cycle. As for packet delay, the traffic load corresponding to 50% of the saturation throughput is applied to the network. These figures indicate that the BO values from 5 to 14, which correspond to the SO values from 4 to 13, result in almost the same throughput and energy performance. Since IEEE 802.15.4 specifies that the packet transmission has to be deferred if the remaining CAP is not enough to finish it, a short superframe duration would cause the frequent deferment, and, as a result, the overhead for each transmission increases. This phenomenon is observed from the BO of 0 to 4. On the other hand, the delay performance is enhanced if the BO and SO values are reduced. Note that the increase of the BO and SO are not considered because we postulate that those orders explicitly represent the requirement of the operating application, especially, the acceptable packet delay.

Considering all the above factors, the following principles should be adopted to the design of VCS.

1. *Each available channel should be identified as a logical channel for certain BOs corresponding to a certain*

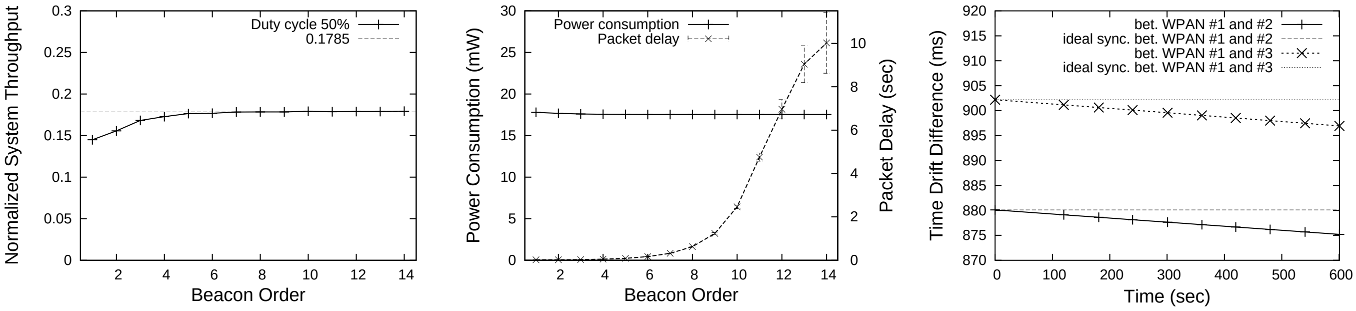

Figure 4. (a) Normalized saturation throughput, (b) power consumption and delay, and (c) different time drifts among three PAN coordinators

BO group.

2. *The incoming WPAN's BO and SO can be reduced for the coexistence while maintaining the original duty cycle.*

3. *If the incoming WPAN causes severe degradation to the previous WPANs, it should not start operation.*

As mentioned in the previous section, LC-scheduler excludes many candidate ψ's in Eq. (4) before its minimization process in Eq. (6). Since VCS best tries to equalize coexisting WPAN's BOs in the same logical channel to benefit as the ideal case, it is obvious that the incoming WPAN's superframe needs to be stuck to other superframes in any superframe schedulers using VCS.

4.2 Virtual Channel Selector (VCS)

Based on the principles in the previous subsection, we devise *Virtual Channel Selector (VCS)*. This works as the virtual channel manager combined with one of the superframe scheduling algorithms. The main idea of VCS is to categorize the incoming WPANs' BOs into a few groups in order to achieve the advantage of the ideal case even with limited logical channels.

VCS needs two parameters, namely, the predetermined set Φ for logical channel grouping and BO adjustment and the threshold p_{thr} for the self-admission control. The set Φ contains BOs frequently used by co-located WPANs. By using the elements in the set Φ as the boundary values for grouping, possible BOs from 0 to 14 can be categorized into a few BO groups. On the other hand, the threshold p_{thr} is used for the third principle in the previous subsection.

In VCS, at first, each logical channel is classified into one of 3 types by investigating the mostly-used BO, $BO_{mostly-used}$, in the channel: (1) Public Channel (PC), (2) Dedicated Channel for a certain BO ϕ_i (DC-ϕ_i), where ϕ_i indicates the *i*-th element of Φ with the elements sorted in the ascending order, and (3) Empty Channel (EC) as follows:

$$\begin{cases} \text{PC} & BO_{mostly-used} < \phi_0, \\ \text{DC-}\phi_1 & \phi_1 \le BO_{mostly-used} < \phi_2, \\ & \vdots \\ \text{DC-}\phi_n & \phi_n \le BO_{mostly-used}. \end{cases} \tag{7}$$

There might be multiple BOs, which are used the most, in one logical channel to be classified. In such a case, if the candidate BOs are smaller than ϕ_0, the channel is marked as Public Channel (PC). This is because PCs are also utilized by WPANs, which cannot find a suitable logical channel to start their operations. Otherwise, VCS checks whether all of the candidate BOs can be classified into the same DC-ϕ_i, i.e., whether they are within the range from ϕ_i to ϕ_{i+1} or not. If they can be, the logical channel under classification can be identified as DC-ϕ_i. If not, the channel is set to PC.

Reflecting the anticipated applications shown in Table 1, we assume the predetermined set Φ of BOs for the channel classification including two elements, i.e., 6 and 12. Not shown in Table 1, the SOs would be chosen under the consideration of the WPAN size: the number of child devices and the depth of the multihop network. In conducting a BO adjustment, even though the BO of 2 from BOs of 0 to 5 would be dominant among those BOs, the adjustment of BO to 2 is useless since such an adjustment causes performance loss to the network. Moreover, WPANs with BO smaller than 6 tend to have a high duty cycle, which enervates the virtual channel creation. On the other hand, with the BO of 7 to 11 or 13 to 14, the BO adjustment is possibly done so as to reduce the order to the extent of 6 or 12.

After the channel classification and BO adjustment, VCS attempts to select the logical channel and the time offset for the superframe. If the incoming WPAN successfully finds DC-ϕ_i including its BO, VCS calls the superframe scheduling algorithm to get the proper time offset in that logical channel. Even though DC-ϕ_i for the incoming WPAN's BO exists, a returned $H(\psi)$ smaller than p_{thr} causes a self-admission failure. In such a case, if an EC is available, it takes a privilege to operate in the EC.

If an admission failure occurs due to a small $H(\psi)$ and an EC does not exist or the incoming WPAN has BO less than the least BO in Φ, VCS calls the superframe scheduling algorithms for PCs and search the available time offsets by comparing p_{thr} with the returned $H(\psi)$. Before calling, the candidate PCs for searching should be sorted in the ascending order by the number of WPAN's child devices. Only if all of the above search attempts fail, the incoming WPAN achieves a privilege to operate in the EC if available. In the case that an admission failure occurs and no EC exists, the incoming WPAN not able to find a proper PC cannot start its operation.

With a given p_{thr}, VCS conducts a self-admission control as mentioned above. From Eq. (6), p_{thr} can be obtained by

$$p_{thr} = q \cdot 2^{SO_{n_p} - BO_{n_p}} \cdot p_c(N_{n_p} + N_{ex}), \qquad (8)$$

where q is a tolerable overlap portion to an incoming WPAN's superframe length, and N_{ex} is the number of expected additional contending devices in the overlapped period. By choosing both values for the given application, p_{thr} can be calculated. Note that, for two heuristics, N_{ex} is only effective when it is a multiple of N_{fix}, the assumed number of every WPAN's child devices.

p_{thr} is also useful to reflect a WPAN's characteristic into a WPAN starting process. This parameter implies how much the WPAN is tolerant of extra performance loss caused by superframe overlaps, thus showing demanded reliability of the network, and how much it urges to start its operation. The parameter may be determined according to quality of service (QoS) requirement, resident time, and so on of the application.

5 Inter-WPAN Synchronization

Throughout this paper, we implicitly ground on the assumption that every WPAN has a precise timer for the scheduling of its radio on/off events. However, this cannot be true for practical WPAN devices. Generally, the oscillator drift caused by thermal instability entails the inaccuracy of the timer [8]. As a result, each WPAN experiences a different time drift. To see the degree of this time drift, we conduct a simple experiment with IEEE 802.15.4 testbeds from Korwin [9].

Fig. 4(c), in which BOs of three coordinators are equally set to six, shows the severeness of the problem. Although the devices in one WPAN can synchronize to each other through the reception of the periodic beacons, PAN coordinators coexisting in the same logical channel show different time drifts to each other; the difference between WPAN #2 and #3 seems ignorable, but, as the time goes, it would become critical. Accordingly, the cooperation for the synchronization is necessary for a practical virtual channel management.

We propose *Inter-WPAN Synchronization (IWS)* to prevent the asynchronous time drift of the beacon transmission in coexisting WPANs. The key idea is to activate the beacon tracking functionality of a PAN coordinator, which is originally designed to be used only for non-PAN coordinator devices, so as to obtain the reference time of its own beacon transmission. Since in IEEE 802.15.4 each device has a capability of both tracking a coordinator's beacon and transmitting its own beacon as a coordinator, a simple modification to manage the received beacon enables the time synchronization among the operating WPANs in the same logical channel; when a PAN coordinator receives a beacon of the specified WPAN operating in the same logical channel, it simply reflects the time stamp of the beacon to the transmission time of its own beacon, and drops it to avoid a further processing. For this, each PAN coordinator should know which WPAN has to be synchronized with itself. For this purpose, PAN Identifier (PAN ID) can be used. All incoming WPANs should choose a larger PAN ID than any of the previously used ones. Then, by tracking the beacons of the WPAN with the least PAN ID number, the incoming WPAN can ensure that it shares the same time reference with all the coexisting WPANs in its transmission range.

6 Simulation

6.1 Simulation Setup

In this section, we evaluate the proposed schemes via computer simulations. Most of all, a realistic model for the beacon order (BO) distribution of incoming WPANs is definitely required since the proposed scheme highly depends on it. Based on the target applications shown in Table 1, a distribution as shown in Fig. 5(a) is considered throughout the simulations. For superframe order (SO), we apply different rules for each BO. When a BO is less than 4, SO is set to 0. If BO is larger than 3 and less than 6, we set SO to 1. In other cases, a randomly chosen number between 0 and $(BO - 2)$ is set to SO. Indeed, SO is related to the size of the WPAN: the number of child devices and the depth of the network represented by an average hop count to child devices.

Basically, most of simulations are conducted with four available logical channels and $q = 0.3$ to see how different schedulers make decisions for ψ when superframe overlap is partially allowed. Additionally, BO_{limit} is set to 10 to mitigate the memory space and computing power requirement. The number of child devices is randomly selected

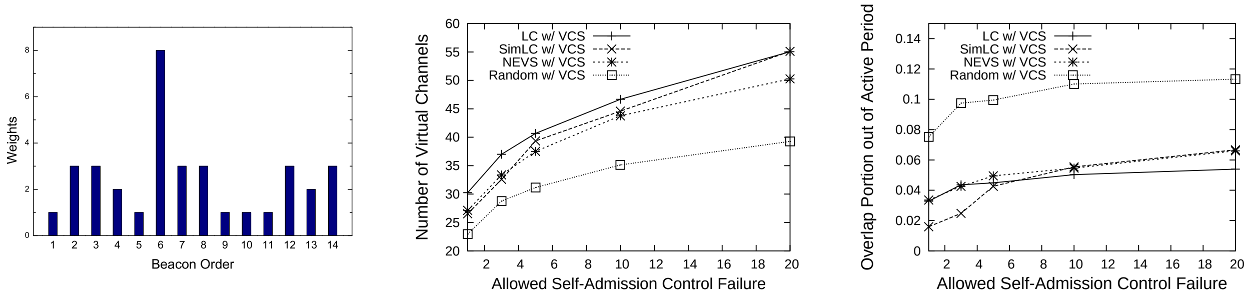

Figure 5. (a) BO distribution weights, (b) virtual channels and (c) overlap portion vs. self-admission control limit

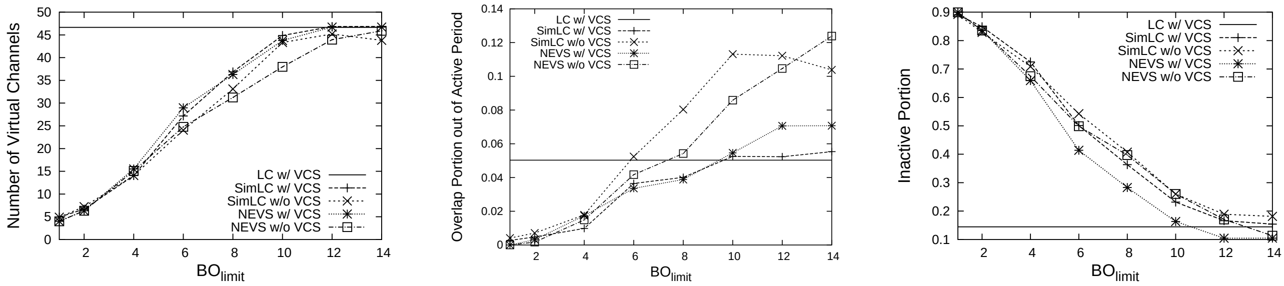

Figure 6. (a) Virtual channels, (b) overlap portion and (c) inactive portion vs. BO_{limit}

from 3 to 20, and two heuristics set their N_{fix} to 10. In order to emulate the coexisting situation with IEEE 802.11b WLANs, four logical channels are used. Additionally, for a comparison, a random scheduler, which randomly chooses time offset ψ, is also considered. However, this scheduler does not represent the IEEE 802.15.4 standard.[6] Instead, it can show the performance of the simplest scheduler which practices our virtual channel concept with self-admission control.[7] All simulation results are averaged with 100 different runs under IWS activation.

6.2 Simulation Result

To measure the achievable virtual channels with given parameters, we first have to determine how many WPANs would try to start operation until the end of each simulation run. As the number of such trials increases, the possibility to encounter WPANs whose BO is large and SO is small enough to "squeeze" more virtual channels becomes high, even though many WPANs' trials keep failing until those encounters. Therefore, the sum of the self-admission control failure counts is necessarily checked to prevent meaningless trials to squeeze more virtual channels. By limiting this value, namely, self-admission limit, we can apply a proper number of attempts to start a WPAN in order to obtain considerable results.

Observing the created virtual channels while varying the self-admission limit in Fig. 5(b), the virtual channels are augmented as the limit increases. In addition, Fig. 5(c) shows the overlap percentage of active periods. From these figures, we choose the self-admission limit of 10 for the further simulations. Larger value than 10 causes relatively constant overlap portion while successfully squeezing more virtual channels, which hardly gives meaningful results.

In addition, BO_{limit} for SimLC-scheduler and NEVS should be properly set to reduce the required memory space and time for scanning this space. Fig. 6(a) depicts the number of created virtual channels with respect to the BO_{limit} value. We learn from this figure that SimLC-scheduler with $BO_{limit} = 10$ or 12 shows slight performance loss in spite of the considerably reduced length of the observing channel occupancy function. Consequently, the memory size and required searching space are lessened to one fourth or sixteenth of that of LC-scheduler, respectively. A similar tendency is observed for NEVS. Even with its simplicity, it

[6] The number of achievable channels of standard 802.15.4 is obviously equal to the number of given logical channels.

[7] Without self-admission control, random scheduler cannot give meaningful results on the number of achievable virtual channels since it has no constraint to create them.

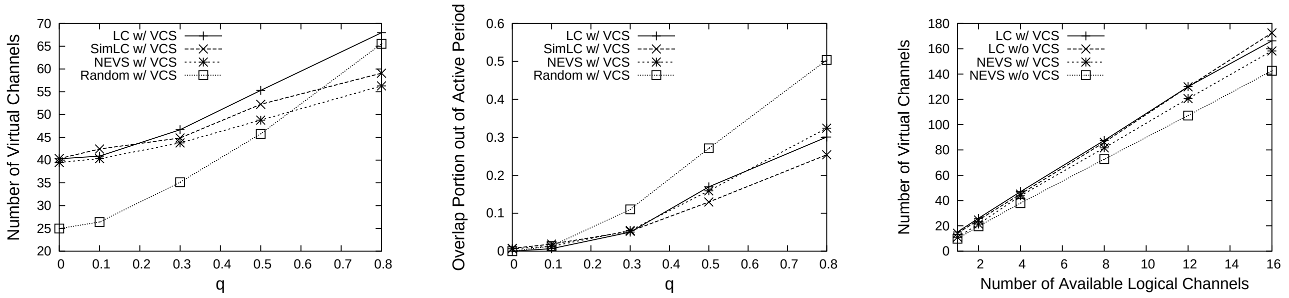

Figure 7. (a) Virtual channels and (b) overlap portion vs. q for p_{thr}, and (c) virtual channels vs. logical channels with self-admission limit

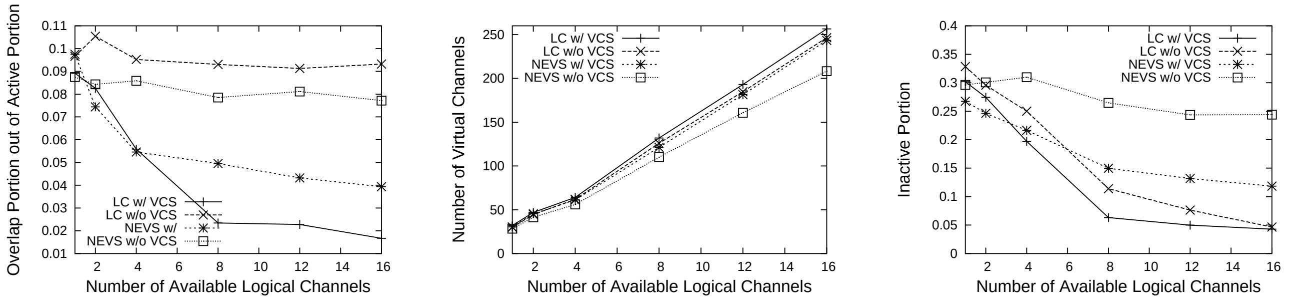

Figure 8. (a) Overlap portion vs. logical channels with self-admission limit and (b) virtual channels and (c) inactive portion vs. logical channels without self-admission limit

works close to SimLC-scheduler with each BO_{limit}. Once compared with NEVS without VCS, we also notice that the VCS plays the main role for NEVS to perform as good as SimLC-scheduler.

Despite less complexity and memory requirement, using BO_{limit} less than 14, as shown in Fig. 6(b), causes undesirable extra overlaps among coexisting WPANs in the same logical channel at the cost of each heuristics' good performance on the number of achievable virtual channels. From $BO_{limit} = 10$, the incurred overlap portion of two heuristics with VCS exceeds that of LC-scheduler. Without VCS, the overlap portion increases significantly despite a similar number of created virtual channels; the overlap portion of SimLC-scheduler and NEVS without VCS exceeds that of two schedulers with VCS at $BO_{limit} = 10$ by 120% and 60%, respectively.

In Fig. 6(c), it is shown that the performance loss caused by a smaller BO_{limit} does not cause a poorer channel utilization in terms of the average inactive portion in logical channels. Again, as seen in Fig. 6(a), $BO_{limit} = 10$ or 12 can be a good approximation for SimLC-scheduler and NEVS. From this fact, we set BO_{limit} of subsequent simulations to 10. Note that, especially without VCS, two heuristics show less channel utilization than those with VCS while causing a more overlap portion as shown in Fig. 6(b). Hence, we can claim that the SimLC-scheduler and NEVS with $BO_{limit} = 10$ or 12 is very powerful and practical with the aid of VCS. ne more thing that can be observed from Fig. 6 is that SimLC-scheduler outperforms NEVS; SimLC-scheduler leaves more inactive period while creating almost the same number of virtual channels. However, it obviously requires more running time.

In Fig. 7(a), we observe the number of possible virtual channels under given q (used in Eq. (8)). In spite of its simplified features, SimLC-scheduler performs almost the same as LC-scheduler with low q in terms of rendered virtual channels. Also, NEVS shows a similar coexistence capability with low q value. At the same time, the random scheduler shows a poor capability to make virtual channels with low q value, yet the capability approaches to that of LC-scheduler and even exceeds that of SimLC-scheduler as q increases. From this, we can see that, if WPANs allow more overlaps of superframes, random scheduler can benefit from its innateness to spread out superframes. In contrast, LC-scheduler, SimLC-scheduler and NEVS tend to aggregate superframes of WPANs. However, highly overlapping superframes are hardly demanded by any applications, and hence, the trend that the random scheduler's performance approach to LC-scheduler's seems useless.

Intuitively, due to $BO_{limit} = 10$, SimLC-scheduler and NEVS are expected to cause more overlaps even with the same created virtual channels as that of LC-scheduler. Nevertheless, in Fig. 7(b), it turns out that the two heuristics do not significantly incur more overlap portion than LC-scheduler. Thus, we can say that the information of other WPAN's child devices for LC-scheduler does not severely impact the performance of the scheduler. Hence, SimLC-scheduler and NEVS are practically valuable despite its ignorance of such information.

To clarify the role of VCS more specifically, the achievable virtual channels with respect to the given logical channels are further investigated in Fig. 7(c). (SimLC-scheduler not shown in the figure follows very similar trends of LC-scheduler). Unlike our expectations, the adoption of VCS does not seem to be beneficial enough for coexisting capability enhancement for LC-scheduler. However, the observations on incurred overlap portion in Fig. 8(a) indicate that VCS facilitates more collision-relaxed virtual channel creations. For NEVS, it is clearly demonstrated that VCS contributes to boost up NEVS's coexistence capability.

It can be expected that one of the factors that the adoption of VCS does not impressively increase virtual channels is the self-admission control scheme in VCS. Since we limit the number of total self-admission control failures in all the simulations above, the schemes with VCS are more likely to cause an early cessation of the simulations.

To confirm this, we offer many WPANs' trials to squeeze virtual channels with $q = 0$ and observe the achievable virtual channels. This is depicted in Fig. 8(b); with 8 available logical channels, LC-scheduler with VCS achieves 7 extra virtual channels compared to LC-scheduler without VCS. This means that, without any communication overhead, VCS makes 7 more virtual channels for additional applications. Even with one logical channel, the small enhancement is observed due to the BO and SO adjustment mechanism of VCS. SimLC-scheduler is omitted in these figures since it shows almost the same tendency as LC-scheduler. For NEVS, VCS significantly improves coexisting capability by achieving 12 extra virtual channels with given 8 logical channels. Thus, we can conclude that VCS can function to render more virtual channels if WPANs with proper BO and SO are provided.

In addition, we study an average inactive portion of all logical channels in Fig. 8(c), and ensure that VCS efficiently manages the logical channels. Especially with a few available logical channels, VCS can reduce the inactive portion by about 5% more. By grouping similar BOs in the same logical channel, VCS can reduce futile inactive portion so that more virtual channels can be created.

7 Conclusion

In this paper, we propose a novel concept of Virtual Channel in order to enhance the coexistence capability of IEEE 802.15.4 LR-WPANs. To efficiently implement this concept, the virtual channel management is developed, which comprises (1) one of the superframe scheduling algorithms including LC-scheduler, SimLC-scheduler and NEVS, and (2) Virtual Channel Selector (VCS). Moreover, a practical time synchronization problem among coexisting WPANs is resolved by the proposed Inter-WPAN Synchronization (IWS). According to the simulation results, the number of coexisting WPANs can be significantly increased for each logical channel. Thus, the proposed schemes make the 802.15.4 practically more attractive by making lots of LR-WPAN applications operate simultaneously even with the interference from other ISM band-based wireless networks, especially, IEEE 802.11b WLANs.

References

[1] IEEE Std 802.15.4, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specification for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4-2003, Dec. 2003.

[2] Youngkyu Choi *el at.*, "Enhancement of a WLAN-based Internet Service in Korea," in *Proc. ACM WMASH'03*, Sep. 2003.

[3] Soo Young Shin *et al.*, "Packet Error Rate Analysis of IEEE 802.15.4 under IEEE 802.11b Interference," in *Proc. WWIC'2005*, Xanthi, Greece, May 2005.

[4] Ivan Howitt and Jose A. Gutierrez, "IEEE 802.15.4 low rate wireless personal area network coexistence issues," in *Proc. WCNC'03*, March 2003.

[5] "ZigBee Alliance," http://www.zigbee.org, Online Link.

[6] IEEE 802.15.4-REVb/D1, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specification for Low Rate Wireless Personal Area Networks (LR-WPANs), Jan. 2005.

[7] Tae Rim Park, Tae Hyun Kim, Jae Young Choi, Sunghyun Choi, and Wook Hyun Kwon, "Throughput and Energy Consumption Analysis of IEEE 802.15.4 Slotted CSMA/CA," *IEE Electronics Letters*, vol. 41, issue 18, Sept. 2005.

[8] Jeremy Eric Elson, "Time Synchronization in Wireless Sensor Networks," Ph.D dissertation, Univ. of California, L.A., 2003.

[9] "Korea Wireless Networks (Korwin)," http://www.korwin.co.kr/home/, Online Link.

[10] Hao Zhu and Guohong Cao, "A Power-Aware and QoS-Aware Service Model on Wireless Networks," in *Proc. IEEE INFOCOM'04, Mar. 2004.*

[11] R. Krashinsky and H. Balakrishnan. "Minimizing Energy for Wireless Web Access with Bounded Slowdown," in *Proc. ACM MobiCom'02,* Sept. 2002.

[12] A. Grilo *et al.*, " A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks," *IEEE Wireless Communications*, vol. 10, pp. 36–43, June 2003.