Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks

Emanuele Toscano, Lucia Lo Bello[1] Senior *Member, IEEE*

***Abstract*—The IEEE 802.15.4 protocol offers great potential for industrial wireless sensor networks, especially when operating in beacon-enabled mode over star or cluster-tree topologies. However, it is known that beacon collisions can undermine the reliability of cluster-tree networks, causing loss of synchronization between nodes and their coordinator. For this reason, this paper proposes a Multichannel Superframe Scheduling (MSS) algorithm, a novel technique that avoids beacon collisions by scheduling superframes over different radio channels, while maintaining the connectivity of all the clusters. The paper describes the MSS algorithm and addresses the advantages it provides over the time-division superframe scheduling. Analytical results are shown which provide a quantitative estimation of how the schedulability space is improved, while simulation results show that the proposed technique increases the number of schedulable clusters and the maximum cluster density. Finally, the paper proves the feasibility of the proposed approach describing a working implementation based on TinyOS.**

***Index Terms*—Wireless Sensor Networks, IEEE 802.15.4, ZigBee, cluster tree, Multichannel Superframe Scheduling.**

I. Introduction and Motivation

The IEEE 802.15.4 standard [4] is generally considered one of the most suitable protocols for Wireless Sensor Networks (WSNs) in the industrial field, as it enables low-rate low-power communications. The IEEE 802.15.4 MAC protocol features two operating modes: a non-beacon-enabled mode, in which nodes access the channel using a classic (unslotted) CSMA/CA mechanism and a beacon-enabled mode in which time is subdivided in superframes, with a slotted CSMA/CA mechanism. When nodes operate in beacon-enabled mode, they subdivide their time into beacon intervals, that are delimited by beacon frames periodically broadcast by each coordinator. Each beacon interval is divided into an active section, called superframe, and an inactive section, during which nodes do not transmit and may enter low-power states. The duration of these sections determines the nodes' duty cycle. When used in the beacon-enabled mode, the IEEE 802.15.4 protocol supports collision-free time slots, called Guaranteed Time Slots (GTS), which can be exploited for transmitting real-time traffic. The allocation of one or more GTSs allows the guarantee of a defined bandwidth and a maximum access delay for a node. The IEEE 802.15.4 protocol supports three different topologies: star, mesh, and cluster-tree. However, the beacon-enabled mode (and hence, support for the Contention-Free Period) only works in star or cluster-tree networks. In the cluster-tree topology, the network comprises multiple coordinators, also called ZigBee Routers (henceforth referred to as "routers"), which establish parent-child relationships so as to form a tree rooted at the PAN coordinator. Moreover, each coordinator periodically generates beacon frames to synchronize the nodes belonging to their cluster (Fig. 1).

The bounded delay capabilities of cluster-tree networks proven in the work [5] enable the use of IEEE 802.15.4 cluster-tree networks to support time-constrained traffic and make it attractive for industrial applications, such as remote sensor/actuator control in production automation and monitoring applications in factory automation. However, if the MAC parameters are not properly configured, serious unreliability problem may arise [6]. One of the most serious problems occurs when the transmission of the beacon frames is not properly synchronized. If not properly scheduled, a beacon frame may collide either with other beacon frames from different coordinators or with data frames from different clusters. Nodes not receiving beacon frames may lose the synchronization with their coordinator and therefore get disconnected from the network. In particular, there are two different types of beacon collisions: direct and indirect. A direct beacon frame collision occurs when two or more coordinators are within the respective transmitting ranges and the transmissions of their beacon frames overlap. An indirect beacon collision may occur when two coordinators are not within their respective radio range but their transmitting ranges intersect. Nodes lying on the intersection may experience indirect beacon frame collision. Collisions may also occur between beacon frames and data frames, when a router transmits its beacon frame during the active period of an adjacent cluster. In cluster-tree topologies therefore the problem of beacon frame collisions is a major issue. In [7] it was shown that although multi-hop beacon-enabled networks are feasible when the beacon order is greater than one, the distribution of coordinators is not very dense and the traffic is low, the non-negligible probability of losing the synchronization may not meet the stringent reliability requirements of typical wireless industrial networks [1]-[3].

The 2006 revision of IEEE 802.15.4 [4] introduces support for time division superframe scheduling, that can be used along with an algorithm to schedule the superframes of a cluster-tree network in a contention-free fashion. However, the standard does not define actual mechanisms to schedule superframes in such a way that beacon collisions are avoided.

In [21], a multichannel approach to schedule the superframes of cluster-tree IEEE 802.15.4 networks over multiple channels on commercial off the shelf (COTS) hardware is presented, together with the Multichannel Superframe Scheduling (MSS) algorithm, that exploits the multichannel approach to allow multiple clusters to schedule their superframes simultaneously on different radio channels. This paper extends the work in [21] in several respects. Firstly, it provides a comprehensive and detailed description of the MSS and a thorough analysis of

1 The authors are with the Department of Electrical, Electronics and Computer Engineering, University of Catania, Italy. emails: {etoscano, llobello}@diit.unict.it

its performance that includes a comparison, both analytical and simulative, between the scheduling capabilities of single-channel and multichannel approaches when used to implement an industrial WSN. Secondly, this work discusses how to implement the multichannel technique here proposed. Finally, this paper demonstrates the feasibility of the proposed approach through a working implementation, i.e., an experimental testbed realized using COTS IEEE 802.15.4 hardware and open-source software.

The paper is organized as follows. Section II addresses related work. Section III explains the rationale behind the multichannel approach, while Section IV provides a detailed description of the MSS algorithm. Section V discusses a suitable channel allocation strategy which enables spatial reuse of the frequencies. Section VI provides a comparison between the time division and the MSS approach in terms of schedulability and scalability, based on both analytic and simulative results. The latter were obtained in realistic scenarios varying both the number and the density of the coordinators. Section VII shows the feasibility of the proposed approach on COTS hardware, providing the description of a working implementation under TinyOS and discussing the experimental results obtained through a real testbed. Finally, Section VIII concludes the paper.

II. Related work

Some works in the literature [24]-[25] proposed mechanisms to avoid beacon frame collisions based on distributed protocols. However these distributed approaches, which are suitable for WSN applications in home and building automation, do not match industrial sensing/control WSNs very well, where a centralized approach may be more appropriate for two main reasons. The first is that the nodes' local knowledge may not be enough to avoid interference between different clusters, as at a given distance nodes may be too far away to successfully communicate with each other, but not far enough to avoid interference. The second reason is that by using such distributed approaches the schedule of a given set of superframes is non-deterministic, as it depends on the arrival order of the beacon requests.

The problem of scheduling multiple IEEE 802.15.4 superframes having different beacon intervals and superframe durations is similar to the Multicycle Scheduling problem, which have been studied in the context of filedbus networks [8]-[9]. In [10], a time division beacon frame scheduling (TDBS) mechanism is presented, where IEEE 802.15.4 superframes are scheduled according to a centralized algorithm run by the PAN coordinator, which is called Superframe Duration Scheduling (SDS). This algorithm is based on the "pinwheel scheduling algorithm" [11], which analyzes the schedulability of a superframe set with different durations and beacon intervals, and provides a schedule if the set is feasible. Time is partitioned into slots, called minor cycles, and a cyclic schedule is produced for a major cycle by allocating the superframes in an increasing order of beacon intervals. An optimization of the time division superframe scheduling problem, with respect to cyclic real-time traffic, is presented in [12], where an algorithm is presented which finds a collision-free superframe schedule that meets end-to-end deadlines of data flows while minimizing the energy consumption of nodes.

While a pure time-division approach solves the beacon frame collision problem, it limits the network scalability in terms of the number of clusters for which a feasible superframe schedule is found, as no parallel communication is allowed unless coordinators are distant enough not to collide. The idea of improving the network performance though the use of multiple radio channels is not new and has also been used in WirelessHART [13], the recent communication standard explicitly designed for industrial applications. Several research works focused on the design of data link layer scheduling in WirelessHART networks [14]-[16] and on how to satisfy the reliability requirements on the routing graphs and construct datalink layer communication schedules on top of them [17]. However, WirelessHART also imposes strict limits on the duration of TDMA slots, which is fixed to 10 ms, therefore it is not as flexible as IEEE 802.15.4 in terms of superframe durations and beacon intervals. A scalable MAC protocol tailored for WSNs is presented in [18], which takes advantage of multifrequency availability to achieve parallel communication while improving energy efficiency. The advantages of multichannel IEEE 802.15.4 communication, in terms of reliability and throughput, is assessed in [19], where two multichannel architectures are presented which make use of the theoretical framework presented in a previous work [20] so as to support predictable medium access, real-time analysis, admission control and transport layer retransmissions. This solution, however, requires custom hardware featuring multiple transceivers in one node.

On the contrary, the Multichannel Superframe Scheduling (MSS) here presented, thanks to its multichannel approach, allows multiple clusters to transmit their superframes simultaneously on different radio channels without requiring custom hardware. In addition, MSS needs only minor changes to the standard 802.15.4 MAC protocol, as it uses the same standard superframe structure as defined in the 2006 revision of IEEE 802.15.4 [4]. The following Section highlights the multichannel approach which MSS builds on.

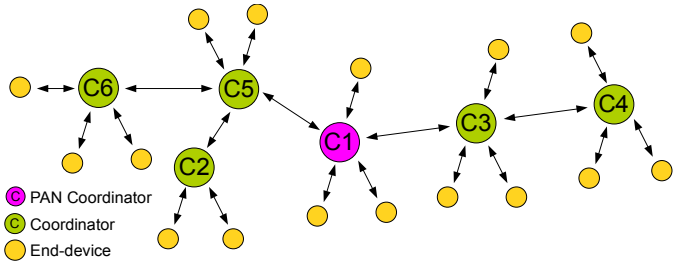

Fig. 1: Network topology.

III. The multichannel approach

The use of multiple channels within the same cluster-tree network is not straightforward, as direct communication between two nodes can take place only if the nodes operate on the same radio channel. For instance, considering the topology in Fig. 1, C3 is the coordinator of a cluster, while also being a member of the cluster coordinated by C1 (the PAN Coordinator). If C1 transmitted its beacon frame on a given radio channel while C3 was scheduling its superframe on a different channel, then C3 would lose the beacon frames from C1. As a result, C3 and C1 would not be able to communicate to each other. The simplest solution to this problem would be to provide C3 (and all the other coordinators) with two different transceivers that can be individually set to two different radio channels, i.e., the

channel of its cluster and that of the parent. Unfortunately, this solution would require custom hardware, as COTS IEEE 802.15.4 modules include a single transceiver. However, as data transmission is performed hop-by-hop, a viable solution to avoid the above mentioned problem is to give C1 and C3 a proper schedule so that, while C1 is transmitting, C3 avoids transmitting but it is still able to receive on C1's channel. On the contrary, C4 may transmit simultaneously with C1 on a different channel, as C4 is not intended to communicate directly with C1. The multichannel approach to avoid beacon frame collisions is based on that consideration. In general, the problem of enabling adjacent clusters to communicate although they use two different channels for their inter-cluster communications can be solved by scheduling adjacent clusters in two alternate timeslices, so that when a coordinator schedules its superframe, its adjacent coordinators are prevented from scheduling theirs. In fact, since in a cluster-tree network there are only two categories of connections, namely parent and children, and since each child has only one parent, it is always possible to partition the network by scheduling beacon frames in two different timeslices. Indeed, all the coordinators which are two hops away may transmit in the same timeslice. For instance, the clusters of the topology in Fig. 1 will be assigned the timeslices as follows: the coordinator C4 will schedule its superframe in the first timeslice (TS1), simultaneously with C1, C2, and C6 but on different radio channels (unless a cluster is so far away that no significant interference may be experienced, as is discussed in Section V). In the following time slice (TS2), C3 and C5 can schedule their superframes. However, the coordinators C4, C2, and C6 remain active and switch to the radio channel used by their parents. This way, they can receive the beacon of their parent as well as communicate with nodes of the parent clusters.

IV. Multichannel Superframe Scheduling

The basic idea behind the Multichannel Superframe Scheduling (MSS) algorithm is to use the maximum superframe duration to determine the timeslice length and to schedule on a different channel all the superframes within the same timeslice. We suppose that each cluster can have a different Beacon Interval (BI) and Superframe Durations (SD), but the configuration of the routers in terms of beacon and superframe order is known a priori and their location is fixed. The MSS algorithm creates a cyclic schedule, which is built for one major cycle, i.e., the least common multiple (LCM) of the beacon intervals of all the clusters, and then is cyclically repeated. Note that, as the beacon intervals supported by the standard are multiple between each other, the LCM always coincides with the greatest BI. The major cycle is divided into smaller time intervals called minor cycles, whose length is given by the greatest common divisor of the beacon intervals of all the clusters and always coincides with the smallest BI. Table I summarizes the notation used in the paper.

The scheduling algorithm takes a superframe set S as input and builds a complete schedule through a sequence of four different steps, as follows:

1. *Cluster Partitioning.* In order to achieve inter-cluster communication while using multiple channels, it is necessary to impose temporal separation between the adjacent clusters. Time separation between adjacent clusters is achieved in MSS by splitting each minor cycle into two timeslices, namely, TS1 and TS2. For this reason, the first step of the algorithm consists in identifying the adjacent clusters and partitioning them into two different subsets, namely S_{T1} and S_{T2}.
2. *Scheduling in the first timeslice*. Scheduling a generic cluster i means assigning it an outgoing radio channel c_i and providing a start time s_{ik} to each of its superframes in the major cycle. Notice that the number of superframes of cluster i in the major cycle N_i depends on the cluster beacon interval and in particular it is equal to B_{maj}/B_i. All the clusters in S_{T1} can be scheduled at the same time but on a different channel, as no direct communication between them is needed. For this reason, the MSS algorithm schedules all the superframes of these clusters starting from the beginning of the major cycle. To improve the network scalability, the outgoing radio channel c_i is selected through a channel selection strategy that provides for spatial reuse of the channels, which is discussed in Section V.
3. *Calculation of the timeslice boundaries*. Once the superframes of the clusters in S_{T1} have been scheduled, it is possible to calculate the length of the timeslices T_m, which are different in each minor cycle and depend on the duration of the superframes scheduled in TS1. It is worth noting that, for the generic minor cycle m, TS1 starts at time $m\,B_{min}$ and ends at time $m\,B_{min}+T_m$, while TS2 starts at time $m\,B_{min}+T_m$ and ends at time $(m+1)\,B_{min}$, when the next minor cycle's TS1 starts.
4. *Scheduling the clusters in S_{T2}.* In the last step, the MSS algorithm tries to allocate the clusters in S_{T2}. For each cluster, the algorithm firstly calculates the starting offset $t_{start,i}$ that must be added to the beginning of the minor cycle in order to avoid time overlapping between superframes allocated in different timeslices. Then it searches for a suitable minor cycle, i.e., a minor cycle whose second timeslice is large enough to fit the whole superframe duration of the given cluster. The channel is selected

S	The original superframe set
S_{Tn}	The subset of S including the superframes to be scheduled in the n^{th} timeslice
D_i	The superframe duration of the i^{th} cluster
B_i	The beacon interval of the i^{th} cluster
C_i	The duty cycle of the i^{th} cluster
B_{maj}	The greatest beacon interval in S (equal to the major cycle)
B_{min}	The smallest beacon interval in S (equal to the minor cycle)
c_i	The channel of the i^{th} cluster
$s_{i\,k}$	The starting offset of the k^{th} superframe of the i^{th} cluster since the beginning of the major cycle
$t_{start\,i}$	The starting offset of the i^{th} cluster since the beginning of each minor cycle
T_m	The duration of the first timeslice in the m^{th} minor cycle
N_i	The num. of superfr. instances of the i^{th} cluster (= BI_{maj} / BI_i)
Π	The space of all the network configurations (BI, n)
Φ	The space of the feasible network configurations (BI, n)
R_{ALG}^{Σ}	The schedulability region of algorithm ALG in the space Σ.

Table I: Summary of notation

Algorithm 1: MSS

```
1:  S_T1 ← {The set superframes having even tree depth}
2:  S_T2 ← {The set superframes having odd tree depth}
3:  foreach Superframe i ∈ S_T1 do
4:      c_i ← getFreeChannel( )
5:      for k from 0 to N_i − 1 do
6:          s_i k ← k B_s.
7:      end for
8:  end for
9:  for m from 0 to B_maj / B_min − 1 do
10:     T_m ← max(B_i scheduled in the m^th minor cycle)
11: end for
12: Sort(S_T2, increasing BI, decreasing SD)
13: foreach Superframe i ∈ S_T2 do
14:     for m from 0 to B_i/B_min − 1 do
15:         for k from 0 to N_i − 1 do
16:             j ← { m + k N_i }
17:             t_start i ← max( t_start i , T_j )
18:         end for
19:         if t_start i + D_i ≤ B_min do
20:             c_i ← getFreeChannel( )
21:             for k from 0 to N_i − 1 do
22:                 s_i k ← m B_min + t_start i + k B_i
23:             end for
24:         end if
25:     end for
26:     if D_s is not scheduled then exit(non feasible)
27: end for
```

Fig. 2: Pseudo-code of the MSS algorithm

using the same strategy as discussed in Section V.

A. Detailed description

In Fig. 2 the MSS algorithm pseudo-code is provided. According to the first step of the algorithm, clusters are partitioned into two subsets, S_{T1} and S_{T2}. The first subset contains the PAN Coordinator and all the clusters that can reach it in an even number of hops, i.e., all the clusters featuring an even tree depth. All the other clusters, i.e., those featuring an odd tree depth, are assigned to the second subset. Lines 3-8 represent the second step of the algorithm, where all the superframes in TS1 are scheduled starting at time zero, but on a different radio channel. The details of the channel selection strategy is discussed in Section V.

The third step is represented by lines 9-11. For each minor cycle i, the boundary between the first and the second timeslice, T_i, is defined as the time when the last superframe of each minor cycle ends. The value of T_i corresponds to the greatest superframe duration scheduled in each minor cycle. The fourth step of the algorithm begins at line 12. To schedule the superframes in S_{T2}, these are initially ordered in an increasing order of BI and the ties are broken in a decreasing order of SD. Then the algorithm tries to allocate each superframe in S_{T2} starting from the first minor cycle, but, if there is not enough space for a given superframe in a minor cycle, the algorithm goes to the next minor cycle, and so on for a number of minor cycles that corresponds to the BI of the given superframe

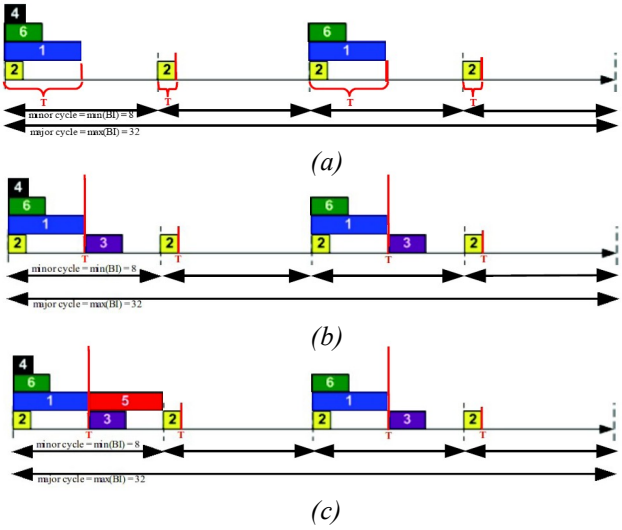

Fig. 3: MSS superframe scheduling.

Coordinator	SD	BI
C1	4	16
C2	1	8
C3	2	16
C4	1	32
C5	4	32
C6	2	16

TABLE II: SET OF SI AND BI VALUES.

(lines 13-27).

In each minor cycle m, the starting time for a superframe is determined by the largest timeslice boundary among those needed by all the superframes of the given cluster in the major cycle (lines 15-18). Analytically, if the coordinator i has multiple superframe instances in the major cycle, and the starting minor cycle is currently m, the starting offset for those superframes is calculated as

$$t_{start\,i}(m) = max\left\{ \bigcup_{i=0}^{N_i-1} T_{m+i\,N_i} \right\}.$$

If there is enough space to allocate the whole superframe duration of the cluster in all the minor cycles, then the superframe is scheduled there (lines 19-24), otherwise, the algorithm goes to the next minor cycle, and so on. If a time is reached when the number of the remaining minor cycles is smaller than the superframe interval, and no space is found that fits the superframe duration, than the algorithm concludes that scheduling is not feasible (line 26).

B. Scheduling example

Fig. 3 shows how the MSS algorithm works in the scenario depicted in Fig. 1, with the superframe set given in Table II. The major cycle is 32, while the minor cycle is 8. According to the MSS algorithm, the clusters in S_{T1} are C2, C1, C6, and C4, while S_{T2} only contains C3 and C5. All the clusters of the first group are allocated together, starting from the time t=0 (Fig. 3a). The timeslice boundary is set as shown in the chart, where it is referred to as T. Now it is possible to schedule the clusters in S_{T2}. The first node to be scheduled is C3. The algorithm starts from the first minor cycle (and the third, according to its BI value). As the greatest timeslice between the first and the

third minor cycle starts at time is 4, we have $t_{start\ 3} + D_3 = 4 + 2 = 6$, which is smaller than 8, so it is possible to allocate the C3. The resulting scheduling is shown in Fig. 3b. The last cluster to allocate is C5. As the beacon interval of C5 is 32, it needs only one minor cycle to be allocated. Again, the algorithm starts the first minor cycle. In this case it is sufficient to check the timeslice boundary of the first minor cycle only, as the beacon interval is 32. Now we have $t_{start\ 5} + D_5 = 4 + 4 = 8$, so the superframe can be scheduled. As a result, the scheduling of the superframe set succeeds, as shown in Fig. 3c.

V. Channel selection strategy

The MSS algorithm schedules on a different channel all the clusters that are on the same timeslice (either TS1 or TS2). For this reason, channel selection is a relevant part of our multichannel scheduling algorithm. There are two main issues to consider when a proper channel has to be selected. The first is network scalability, as the limited number of available channels in IEEE 802.15.4 also limits the number of schedulable clusters. For example, considering a naive channel selection strategy that reserves one channel to each cluster, an MSS-enabled cluster-tree network may comprise up to 32 clusters, 16 in the first timeslice and 16 in the second. This number might not be sufficient for large industrial WSNs, so it is necessary to enable spatial reuse of the channels. The second issue is communication reliability, which may be affected by interference from other clusters of the same network. Interference may not only come from nodes transmitting on the same channel that are not far enough from the receiver, but also from nodes transmitting on the adjacent channel if they are very close. In fact, even though IEEE 802.15.4 channels are not overlapped, recent research [22] shows that some cross-channel interference actually takes place between adjacent channels due to spurious emissions. For these reasons, a channel selection strategy is needed to improve network scalability by enabling channel reuse while limiting interference as much as possible. The mechanism here proposed tries to avoid interference coming from clusters that use the same channel as well as cross-channel interference coming from co-located clusters operating at adjacent channels. The channel selection algorithm is centralized and runs in a network management entity (located either at the PAN Coordinator or outside the network, if the network planning is done offline), which has full knowledge of the network topology, including the location of routers. The information about the routers' location is used to derive the interference relations between the clusters. Although it is quite difficult to accurately predict the signal power in the presence of many obstacles, it is possible to have a conservative estimation of the interfering range, considering that the power received from an interfering node in such an environment is rarely stronger than the power that would be received from a line-of-sight path through free space, with no obstacles between the node and the interferer. The transmitting range in a complex environment is typically smaller than that calculated under the free-space loss model, according to which the received signal power is

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2$$

where P_t is the transmitted signal power, G_r and G_t is the antenna gain of the receiver and the transmitter, respectively, λ is the wavelength of the signal and d is the distance between transmitter and receiver. As a result, it is possible to obtain an estimation of the transmitting range as

$$d_{tx} = \frac{\lambda}{4\pi} \sqrt{\frac{G_r G_t P_t}{10^{S_{dB}/10}}}$$

where S_{dB} is the receiver sensitivity expressed in decibels. The interfering distance, which is the minimum distance after which it is possible to reuse a channel, is generally different from the transmitting range d [27], but is typically related to it. Zhou et al. [28] shows that the interfering range of IEEE 802.15.4 radios is considerably larger (of about 70%) than the transmitting range for weak links. Such measurements are also consistent with the theoretical calculation shown in [31], where the interfering range of 802.11 nodes equipped with ideal omnidirectional antennas is calculated as $d_i = 1.78\ d_{tx}$. However, typical cellular radio architectures adopt larger reuse distances to improve the signal to noise ratio, which can be calculated as $d_u = d_{tx} \sqrt{3C}$, where C is the number of radio channel used in the cellular architecture [29]. Typical values for C are 3, 4, 7, and 9. In an IEEE 802.15.4 network it is possible to realize a cellular architecture with any of these configurations, as 16 non-overlapping radio channels are available.

The proposed algorithm for channel selection first checks whether other superframes have already been allocated at a time that overlaps entirely or partially with the current superframe. If there is no overlapping superframe, the algorithm picks the first free channel. On the other hand, if superframes overlapping in time have been scheduled, the algorithm checks whether some of them are far enough to be scheduled on the same channel, i.e., if the distance from the relevant coordinator is more than d_u. If a suitable superframe is found, the algorithm returns the channel of such a superframe. Otherwise, a new channel is chosen. To limit cross-channel interference, our algorithm first selects the even channels and then, when no more even channels are available, it starts selecting the odd ones. In this way, the allocated channels are not adjacent, unless it is unavoidable.

VI. Comparative Assessments

Here we present a comparison between the time division and the multichannel approach in terms of schedulability and scalability. The comparison is both analytical and simulative. In particular, in Subsection VI-A the schedulability space is analyzed without any restriction on the number of available channels. Then, the advantages of MSS in the realistic case where the number of available channels is limited are shown by a simulative assessment in Subsection VI-B. Simulations will assess the improvements of MSS in terms of number of schedulable clusters and maximum cluster density.

A. Analysis of the schedulability space

The aim of this section is to quantify the improvement of MSS over classical time division superframe scheduling (i.e., SDS), in terms of schedulability. For this reason, the schedulability regions of MSS and SDS are analyzed and compared. Our analysis is performed in three steps. First, a set of assumptions are made under which it is possible to derive sufficient schedulability conditions for both SDS and MSS. This is dealt with in Subsection VI-A1.

Second, the space of the possible network configurations (Π) is defined and the advantages of MSS are analyzed, in terms of increased schedulability region in the domain of Π, which is the subset of Π including only the schedulable network configurations. This is dealt with in Subsection VI-A2.

Finally, a subset of Π is identified as the space of the feasible network configurations (Φ), i.e., the subset of Π which includes only the network configurations that can be setup in real IEEE 802.15.4 networks, and the schedulability analysis is restricted to such a domain in order to derive theoretical results which focus on more realistic IEEE 802.15.4 networks. This is dealt with in Subsection VI-A3.

1) Deriving sufficient schedulability conditions for SDS and MSS.

Here we derive the schedulability condition for SDS which is both necessary and sufficient, and the one for MSS, only sufficient. The first condition is obtained through considerations about the nodes duty cycle, while the second is derived exploiting the results in [21] that were obtained considering the structure of MSS schedules.

We recall that, according to the IEEE 802.15.4 standard, the duration of BI and the SD depends on two parameters, the Beacon Order (BO) and Superframe Order (SO), according to eq. (1) and (2)

$$B = aBaseSuperframeDuration \cdot 2^{O_I} \quad (1)$$

$$D = aBaseSuperframeDuration \cdot 2^{O_S} \quad (2)$$

where *aBaseSuperframeDuration* is a constant defined in the standard [1] that denotes the number of symbols that form a superframe when SO is 0, while O_S and O_B are the SO and BO values, respectively, with $0 \le O_S \le O_B \le 14$.

The duty cycle (DC) of nodes is

$$C = \frac{D}{B} = 2^{O_S - O_B} = 2^{O_I} \quad (3)$$

where O_I is called an Inactivity Order.

In a pure time division approach such as SDS, in each superframe slots are allocated in an exclusive way and there are no simultaneous communications, so a necessary condition for a superframe set to be schedulable is that the sum of all the duty cycles is lower than one [10], i.e.,

$$\sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \frac{D_i}{B_i} \le 1 \quad (4)$$

where n is the number of clusters in the cluster-tree topology. In fact, when the sum of duty cycles is 1, the utilization of the channel is 100%.

On the other hand, the MSS algorithm simultaneously schedules superframes of different clusters in different channels, therefore there is no need to satisfy formula (4). The possibility of scheduling sets of superframes having a sum of duty cycles greater than one extends the space of schedulability. In our analysis, we initially recall the results on MSS schedulability provided in [21] which focused on the restrictions on the superframe duration and beacon intervals.

As each cluster needs time to communicate with the others, no coordinator with a duty cycle equal to one is allowed. Due to relations (1) and (2), the maximum allowed duty cycle is 0.5. If the IEEE 802.15.4 standard were extended to allow arbitrary duty cycles, the need for two non-overlapping timeslices enabling communication between adjacent clusters would impose the condition expressed in eq. (5)

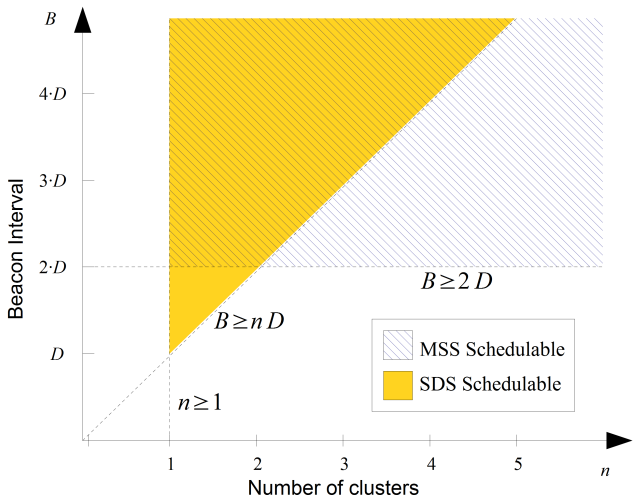

Fig. 4: Schedulability regions of MSS and SDS.

$$\max_{i \in S_{T1}}(C_i) + \max_{j \in S_{T2}}(C_j) \le 1 \quad (5)$$

where S_{T1} and S_{T2} are the sets of superframes to be scheduled in the first and in the second timeslice, respectively. Finally, BI values must be large enough to fit all the superframes in the other timeslice, so eq. (6) must also hold.

$$\max_{i \in S_{T1}}(D_i) < \min_{j \in S_{T2}}(B_j) \quad (6)$$

A more detailed discussion on MSS schedulability, including proof of the above mentioned relations, is given in [21], together with a sufficient schedulability condition for the MSS algorithm that is expressed by eq. (7),

$$\max_{i \in S_{T1}}(D_i) + \max_{j \in S_{T2}}(D_j) \le \min_{k \in S} B_k \quad (7)$$

i.e., the sum of the maximum SD in TS1 and the maximum SD in TS2 must be smaller than the minimum BI.

To compare the schedulability of a single-channel scheduling algorithm such as SDS and the MSS algorithm, let us consider the case in which all the clusters in *S* have the same beacon order and superframe duration. We henceforth refer to this case as a *homogeneous superframe set*. As a matter of fact, this is the only configuration supported by the current IEEE 802.15.4 standard. Under this hypothesis, it is easy to see that eq. (4) is both a necessary and a sufficient condition for a task set in order to be schedulable using SDS [10]. Provided that D and B are the values, common to all the clusters, of the superframe duration and the beacon interval, respectively, eq. (4) can be rewritten as

$$B \ge n\,D. \quad (8)$$

Under the same hypotheses, eq. (7), which provides a sufficient condition for MSS schedulability, can be rewritten as

$$B \ge 2\,D. \quad (9)$$

2) Schedulability analysis in the space of the possible network configurations

The objective of this Subsection is to provide a first comparative result about the schedulability space of MSS and SDS. First, the set of all the possible network configurations is defined. Then, the SDS and MSS-schedulable regions are identified from the schedulability conditions in eq. (8) and (9), respectively. Finally the SDS and MSS-schedulable regions are compared to obtain a quantitative assessment of how MSS

increases schedulability over SDS.
The schedulability space of a superframe scheduling algorithm is the set of all the superframes that are schedulable with the given scheduling algorithm. Under the hypotheses of homogeneous superframes, a set can be specified by using two parameters, i.e., the beacon interval (B) and the number of coordinators (n). Therefore it is possible to define the set of all the possible network configurations as follows:

Definition 1: Π is the set of network configurations $c = (B, n)$ where $B, n \in \mathbb{R}^+$.

If we express the number of clusters in the cluster-tree topology as a function of the minimum beacon interval that must be used to obtain a feasible superframe set, we obtain the schedulability regions of MSS and SDS in the space of all the network configurations in Π, which are indicated as R^{Π}_{MSS} and R^{Π}_{SDS}, respectively. Such regions, directly derived from eq. (8) and (9), are represented in Fig. 4. Here, R^{Π}_{SDS} is the part of the plane delimited by the lines "$n = 1$" and "$B = n\,D$", while R^{Π}_{MSS} is the part of the plane delimited by the lines "$n = 1$" and "$B = 2\,D$".
Both schedulability regions include an infinite number of network configurations, however it is possible to quantify the improvement of MSS over SDS in terms of schedulability space as the ratio between the area of R^{Π}_{MSS} and that of R^{Π}_{SDS}. Intuitively, looking at Fig. 4 it is possible to see that R^{Π}_{MSS} is a rectangle with infinite base and height, while R^{Π}_{SDS} is a triangle with infinite base and height. It is well known that the ratio between the area of a rectangle and that of a triangle, with the same base and height, is 2. This means that MSS increases the schedulability space over SDS by a factor of 2. This result is formally proved in Theorem 1 (see Appendix).

3) Schedulability analysis in the space of the feasible network configurations
The aim of this Subsection is to refine the theoretical result provided in Subsection VI-A2 to provide a comparative assessment of more realistic IEEE 802.15.4 networks. In fact, the space of the network configurations Π contains both feasible and unfeasible network configurations, but we are interested only in the feasible ones. Therefore, this Subsection first defines what a feasible network configuration actually is, then it focuses the analysis on the space of feasible network configurations.
In a real network, the number of clusters n is an integer and, according to the IEEE 802.15.4 standard, the beacon interval has to be a multiple of the superframe duration by a power of two. In a more formal way, it is possible to define the space of the feasible network configurations, Φ, as follows:

Definition 2: Φ is the subset of Π that includes the network configurations $c = (B, n)$ that meet the following conditions:

1. $B \in \mathbb{R}^+$,
2. $n \in \mathbb{N}^+$,
3. $\exists k \in \mathbb{N}^+ \mid B = 2^k \cdot D$.

It is possible to restrict the analysis to the domain of feasible solutions by studying the sets of the feasible network configurations within SDS and MSS schedulability regions.

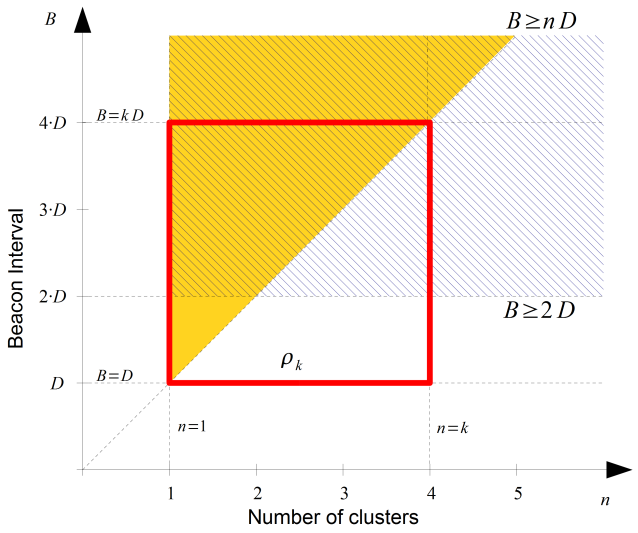

Fig. 5: Graphical representation of ρ_k region for k=4.

As eq. (4) is both a necessary and sufficient condition for an homogeneous superframe set in order to be SDS schedulable, the SDS schedulability region in Φ is given by the subset of R^{Π}_{SDS} which contains only feasible network configurations. Formally, we have:

$$R^{\Phi}_{SDS} = R^{\Pi}_{SDS} \cap \Phi. \tag{10}$$

On the other hand, in the case of MSS, eq. (9) provides a sufficient but not necessary schedulability condition. In fact, if we consider the trivial network configuration made up of just one superframe having a BI equal to its SD, i.e., $c_0 = (D, 1)$, the algorithm shown in Fig. 2 is able to find a feasible schedule, even though this set does not meet condition (9). As a result, we can conclude that

$$R^{\Phi}_{MSS} \supseteq \left(R^{\Pi}_{MSS} \cap \Phi\right) \cup c_0. \tag{11}$$

It is possible to prove that all the SDS schedulable configurations in Φ are also schedulable with MSS (this result is proven in Theorem 2, see Appendix). It follows that the MSS schedulability region is wider than the SDS schedulability region also in the Φ space. However, further analysis is needed to determine to what extent the MSS schedulability region is bigger. For this purpose, it is possible to calculate the discrete number of feasible configurations within a finite region of the space Φ that meet conditions (8) and (9), respectively.
In particular, for each finite region $\rho \subseteq \Pi$, the subset of the SDS schedulability region, defined as $R^{\rho}_{SDS} = R^{\Phi}_{SDS} \cap \rho$, contains a finite number of network configurations. Analogously, the subset of the MSS schedulability region, defined as $R^{\rho}_{MSS} = R^{\Phi}_{MSS} \cap \rho$, has a finite cardinality. Therefore, it is possible to analyse the finite number of feasible network configurations within a generic square region ρ_k that are schedulable under SDS and MSS and then extend the region ρ_k so as to comprise all the feasible network configurations.
In particular, we define the region ρ_k as

$$\rho_k = \{(B, n) \in \Pi \mid D \le B \le k\,D,\ 1 \le n \le k\}, \tag{12}$$

with $k \ge 1$. For example, for k=4 we obtain the region depicted in Fig. 5. It is possible to prove that the number of SDS-schedulable configurations in a generic ρ_k region is

$$|R^{\rho_k}_{SDS}| = 2^{\lfloor \log_2(k) \rfloor + 1} - 1, \tag{13}$$

while the number of MSS-schedulable configurations in ρ_k is

$$|R_{MSS}^{\rho_k}| = k\lfloor \log_2(k) \rfloor. \quad (14)$$

The above results are proven in Lemma 1 and Lemma 2, respectively (see Appendix).

Once the number of SDS and MSS-schedulable configurations in a generic finite region of Π has been calculated, it is possible to quantify the improvement of MSS over SDS in terms of schedulability space as the ratio between the cardinality of $R_{MSS}^{\rho_k}$ and that of $R_{SDS}^{\rho_k}$ as ρ_k approaches Φ, i.e., when k approaches infinity.

Intuitively, when k increases, eq. (13) grows approximately as k, while eq. (14) grows approximately as $k\log(k)$. From this consideration it is possible to prove that the MSS algorithm extends the schedulability region of the SDS algorithm in the space Φ as much as a function that grows like $\log(n)$ as the number of clusters n approaches infinity. Formal proof of this result is given in Theorem 3 (see Appendix).

The analysis described above does not pose limits on the maximum number of channels used by MSS, i.e., it is possible to allocate a virtually infinite number of channels. Even though in real networks the number of channels is finite, the assumption of unlimited channels is not very far from reality, thanks to the channel allocation strategy described in Section V which implements spatial reuse. In fact, three non-overlapping channels are enough to deploy a multi-channel network using the classic honeycomb arrangement. The IEEE 802.15.4 protocol provides 16 non-overlapping channels, so the maximum cluster density under which the MSS can allocate a virtually unlimited number of channels is 5.3 times higher than that of a network using the classic honeycomb arrangement.

B. Simulative Assessment

To assess the advantages of MSS in the realistic case where the number of available channels is limited and the assumption of homogeneous superframe sets is released, we set up a simulative assessment and compared the SDS and MSS schedulability in the same scenarios. Two different assessments were made. The first evaluates the scalability of MSS as the number of the clusters increases. The second evaluates the schedulability of MSS as the density of the IEEE 802.15.4 coordinators increases.

1) Impact of the coordinator number

In the first assessment, the number of coordinators is increased while network density is kept constant. To maintain constant network density while increasing the number of coordinators (N), the ratio between N and the playground area (A) has been fixed to

$$\frac{N}{A} = \frac{2\pi}{d^2\sqrt{27}}, \quad (15)$$

which is the value suggested in [30] as the minimum node density required to monitor a generic area, as a function of the node range (d). However, note that in this paper we are considering the density of coordinators, in place of that of sensor nodes. The number of nodes associated to each coordinator is not relevant in this context, as our purpose here is to assess the ability to schedule a growing set of superframes. The locations of the IEEE 802.15.4 coordinators were set randomly by using

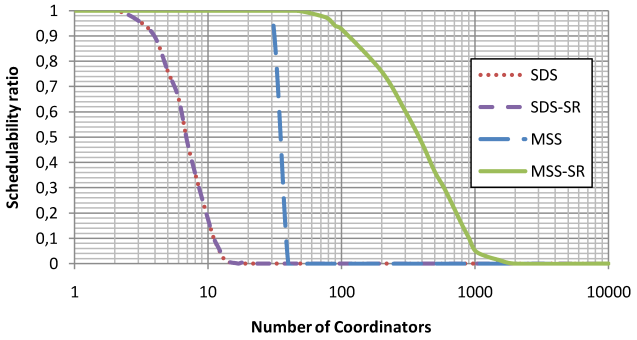

Fig. 6: Schedulability ratio of MSS and SDS vs number of coordinators.

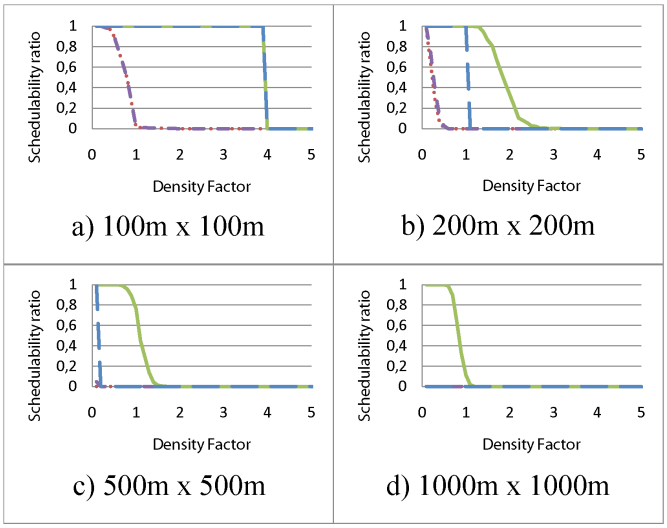

Fig. 7: Schedulability ratio of MSS and SDS vs density.

the GENSEN topology generator [30] with the "one-by-one" deployment strategy, which simulates an operator that deploys the sensor nodes independently from each other. The beacon orders are randomly distributed between 3 and 6, while the superframe orders are randomly distributed between 0 and 2, so that beacon intervals are always longer than superframe durations. The use of random locations, BIs and SDs, allowed us to automatically generate and test a large number of scenarios and therefore to provide statistically significant results. The transmission range (d) of nodes is set to 40 m, while the channel reuse distance (d_u) is set to $2\sqrt{3}d$, approximately 145 m, which is a safe, realistic value, used in cellular architectures that use four different radio channels. The SDS and MSS algorithms are run in the resulting superframe sets in both the cases where spatial reuse of the channels is disabled and enabled. When spatial reuse is enabled, the channels are selected according to what is described in Section V, whereas when spatial reuse is disabled the getFreeChannel function (see the MSS pseudocode in Fig. 2) always returns the next unused channel for the current timeslice.

Forty different scenarios were realized, with the number of coordinators ranging from 1 to 10000. For each scenario, 500 runs were analyzed. The results of our simulations are summarized in Fig. 6, where the schedulability ratio, i.e., the ratio of the runs that were found to be schedulable to the total number of runs using each algorithm, is plotted as a function of the number of coordinators. Here, the cases where spatial reuse is enabled are reported as separate scheduling algorithms and have the suffix -SR, i.e., SDS-SR and MSS-SR. The simulation results show that, as was expected, MSS outperforms

SDS, in both the cases with and without spatial reuse. Moreover, Fig. 6 shows that the results obtained by SDS and SDS-SR are nearly the same, as the graphs overlap. Neither SDS nor SDS-SR were able to schedule scenarios with more than 19 nodes. Some benefit of spatial reuse in SDS is found when the number of coordinators ranges from 10 to 19, but the increase in schedulability ratio is below 1%, so it is negligible in Fig. 6. As a result, in our scenarios, SDS-SR is outperformed also by MSS without spatial reuse, as the latter is able to find a schedule for all the superframe sets until the number of coordinators exceeds 32. In fact, with 16 channels used in TS1 and 16 in TS2, the maximum number of clusters that MSS can schedule without channel reuse is 32. However, spatial reuse provides a considerable advantage to MSS-SR. Our simulations show that about 93% of the runs with 100 coordinators and 76% of the runs with 200 coordinators, respectively, were schedulable. Then, the schedulability ratio is decreased by approximately 10% for each hundred of coordinators added to the network. The reason why spatial reuse is much more effective in MSS than in SDS is that MSS performs a loose time division, as one superframe is scheduled within each timeslice on a given channel. Therefore in MSS it is more likely that coordinators that are further than d_u between each other, can be scheduled in parallel even though they have a different BO and SO value. As a result, the advantage of MSS-SR over SDS-SR in terms of schedulability ratio is of nearly two orders of magnitude.

2) Impact of the coordinator density

The second assessment evaluates how the network density affects schedulability of a superframe set under both SDS and MSS, considering both the cases where spatial reuse is enabled and disabled. In this assessment, the number of coordinators and the playground size are varied in order to test different network densities. In terms of the coordinator transmitting range and placement, we used the same settings as in the first assessment. However, while in the first assessment the playground size was varied so as to keep the density constant, now the playground is still a square, but four different side lengths were selected, i.e., 100, 200, 500, and 1000 meters. The network density is varied by multiplying the number of coordinators, derived from eq. (15), by a rational number, which we call a density factor D_f. This means that, in this second assessment, the number of coordinators is calculated as $N = 2\pi A D_f / (d^2\sqrt{27})$. When D_f is 1, the density is the same as in the first assessment. To assess how the coordinator density affects schedulability, the D_f parameter has been varied from 0.1 to 5 with steps of 0.1. In this way, a wide range of coordinator densities is tested. For each D_f value, 500 different runs were executed. The results in terms of schedulability ratio are presented in Fig. 7, for the four different playgrounds having side lengths of 100, 200, 500, and 1000 m, respectively. The first thing that can be noted is that both the coordinator density and the playground size impact on the schedulability of the tested algorithms and that, to maintain the same schedulability ratio, the density factor must be decreased as the playground size increases. For example, if we look at the highest D_f value for which the schedulability ratio of MSS-SR is higher than 0.9, it is possible to see that this value is 3.9, 1.4, 0.8, and 0.6 when the side length is 100, 200, 500, and 1000 m, respectively. Moreover, it is easy to see that SDS-SR always performs worse than MSS-SR. In the same example, if we look at the highest D_f value for which the schedulability ratio of SDS-SR is higher than 0.9, we obtain the values 0.4 and 0.1 when the side length is 100 and 200 m, respectively. For bigger playground areas, SDS-SR obtained very low schedulability ratios even with the smallest density factor of 0.1. Also in this second assessment there is little difference between SDS and SDS-SR, no matter the playground size. On the other hand, the difference between MSS and MSS-SR performance becomes more marked as the playground size increases. In fact, in the 100×100 scenario there is no difference between them, as the reuse distance is as big as the maximum distance between coordinators. On the contrary, in the 200×200 scenario there is already a noticeable advantage of MSS-SR over MSS, which becomes even more obvious in the larger scenarios. In the 1000×1000 scenario, indeed, all the algorithms except MSS-SR are unable to schedule any of the tested superframe sets.

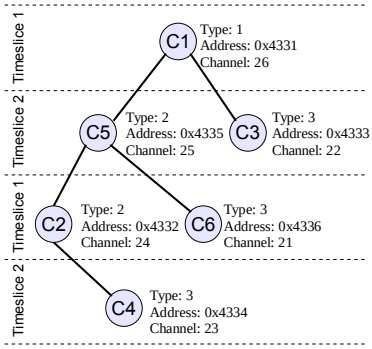

Fig. 8: Network scenario

VII. Experimental Testbed

In order to show the feasibility of the proposed approach using COTS hardware, we implemented it on the well-known TinyOS operating system [33] and tested it using TelosB modules from Crossbow [34].

Implementing the multichannel approach needs only minor changes to the standard IEEE 802.15.4 MAC protocol. In fact, it is possible to use the same superframe structure as defined by the 2006 version of the standard [4], where, for coordinators that are not the PAN coordinator, two different active periods are defined, namely the Incoming and the Outgoing Superframes. In the first the coordinator receives beacons from its parent coordinator, while in the latter the coordinator transmits its own beacon frames. At the MAC layer, the only modification required to implement the multichannel approach is to store two different radio channels, namely the Outgoing and the Incoming Radio Channels and switch to the other channel before the start of the incoming and the outgoing superframe respectively. If the two radio channels coincide, nodes behave exactly as in the standard protocol. As the superframe structure is unchanged, the proposed approach is backward compatible with the standard IEEE 802.15.4 protocol. For nodes not intended to work as coordinators, standard modules can be used.

Our implementation is based on TKN15.4 [35], an open-source, platform-independent IEEE 802.15.4-2006 MAC implementation for the 2.1 release of the TinyOS. To support the MSS we modified the two modules that manage transmission

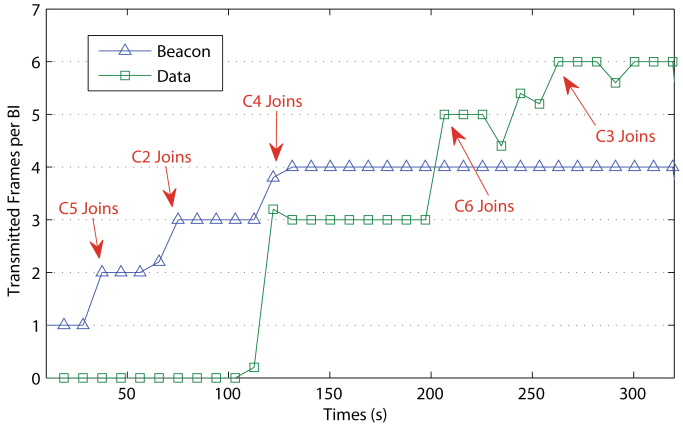

Fig. 9: Temporal trace of the experiment: beacon and data frames per BI

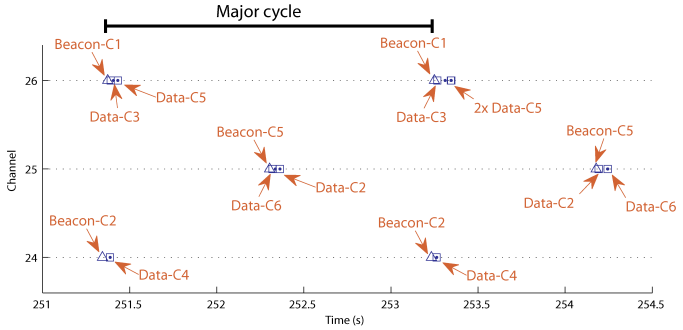

Fig. 10: Temporal trace of the experiment: steady-state network operations

and reception of beacon frames. The first module was modified so as to change the radio channel to the incoming channel before sending the beacon frame, while the latter was modified so as to go back to the outgoing channel when the node is preparing to receive the beacon from the parent coordinator.

As cluster-tree topologies involve multi-hop communications, nodes must be provided with a network layer in charge of data forwarding. As the ZigBee protocol stack [26] already supports a tree routing mechanism it is possible to implement the setup phase of a multichannel network at the application layer, on top of the ZigBee stack. In this way, the association phase remains compliant with the ZigBee and the IEEE 802.15.4 specifications. More details on the association phase can be found in [32]. In our testbed, we also wrote the modules that implement the upper layers, implementing, at the application layer, only the functionalities needed for our purposes.

A. Testbed setup

Three different applications were developed, which identify three types of nodes. Type 1 represents the PAN coordinator, which only schedules the outgoing superframe. Type 2 nodes represent coordinators which receive data during the outgoing superframe and forward the packets later in the incoming superframe. Type 3 nodes are similar to type 2, but they also produce data during their incoming superframe. In particular, a Type 3 node produces a data packet every time it receives a beacon from its coordinator. Using those software modules, we deployed a cluster-tree network composed of six nodes, each hosting a different cluster. The network topology and the configuration of nodes are shown in Fig. 8. In order to make our results easier to examine, we used a different radio channel for each outgoing superframe, and the same superframe durations and beacon intervals for all the clusters. We set BO=7 and SO=6 for each coordinator, which result in a BI of 122880 symbols (corresponding to 1.966 s) and an SD of 61440 symbols (corresponding to 0.983 s). Notice that for such a scenario no feasible schedule can be found using the time division approach, as the sum of the duty cycles of all the coordinators is 3, while using the multichannel approach and the MSS scheduling algorithm a feasible schedule for these superframes was found. In our experiments we performed offline scheduling of the clusters and hardcoded the information about time offset and channel of the outgoing superframe in the coordinators. According to the MSS algorithm, both the major and the minor cycles are equal to the unique BI value. Moreover, the timeslice boundary is equal to the unique SD value. As a result, the *startTime* of all the coordinators of type 2 and 3 is set to 61440 symbols. In order to verify that nodes in the cluster-tree network behave correctly, we used an additional four TelosB modules working in promiscuous mode, each sniffing packets on a different channel (from 23 to 26). In this way, it was possible to observe the transmissions which occur in parallel on the various channels. All the sniffers were connected to a single PC, which was in charge of recording the sniffed packets in a log file and applying a timestamp to each of them. In the end, the whole sequence of transmissions was analyzed by importing the saved log files in a Matlab workspace and extracting the relevant information.

B. Results description and discussion

Fig. 9 shows the average number of beacon and data frames transmitted in each BI, as seen by the four sniffers. In our experiment, the first node to be switched on was C1, so at the beginning there is one beacon frame per BI. Then, at about time 25 s, we switched on C5, which started sending beacons on the second timeslice on channel 25 after the association phase. This can be seen in Fig. 9, as the number of transmitted beacons per BI increases to 2. Similarly, it is possible to see that C2 joined the network at time 65 s. Among the type 3 nodes, the first to be switched on was C4 (at about time 112 s). As soon as C4 was associated with C2, it started transmitting data packets on channel 24. In particular, one data packet is transmitted by C4 in each beacon interval. These packets are then forwarded by C2 and C5 in the respective incoming superframes. Moreover, as C4 is also a coordinator, it also starts transmitting its beacons in its outgoing superframe on channel 23. As a result, in Fig. 9 it is possible to see that the number of transmitted beacons per BI increases to 4 and the number of data packets from 0 directly goes to 3. When C6 joins the network, the number of transmitted data packets increases to 5, as also C6 transmits one data frame per beacon interval, but these frames have to be forwarded by C5. However, although C6 was also transmitting beacon frames we cannot not see them in Fig. 9, as no sniffer was present in channel 21. Similarly, it is possible to see an increased number of transmitted data frames when C3 joins, but the number of beacons is still 4 per BI, because no sniffer was present in channel 22.

In Fig. 10, the temporal trace obtained once all the nodes were active and the network reached a steady-state is shown. In the same figure (where all the channels containing only beacon frames, i.e., channels 21-23, are omitted) data packets are represented as squares labeled with the names of their source nodes, while each beacon is represented as a triangle labeled with the coordinator name. It is possible to see that C2 receives data from C4 in the first timeslice (time 251.3 *s*) on

channel 24, C5 then receives data from both C6 and C2 in the second timeslice (time 252.3 *s*) on channel 25 and finally, in the next first-timeslice (time 253.3 *s*), C1 receives two packets from C5 (containing data originated from C4 and C6, respectively) and a packet containing the data from C3 on channel 26. This pattern of transmissions repeated cyclically untill the end of the experiment. We also found that beacons on the different channels always remained perfectly aligned. In fact, as nodes keep tracking the beacons sent from their parents, they are able to maintain the synchronization of all the superframes.
This experiment shows that nodes behave exactly as they are supposed to do according to the MSS algorithm, therefore providing evidence for the feasibility of the implementation on COTS hardware of the approach proposed in this paper.

VIII. Conclusions

The paper described the Multichannel Superframe Scheduling algorithm for cluster-tree IEEE 802.15.4/ZigBee networks and analyzed the advantages that it offers over standard time division approaches. Theoretical results showed that, compared to traditional time division superframe scheduling, the MSS schedulability space is increased by a factor of 2 in the space of all the possible network configurations and by a factor of $\log(n)$ in the space of the network configurations compliant with the IEEE 802.15.4 standard. Simulative results obtained in realistic scenarios confirmed that the proposed technique outperforms standard time division approaches. The MSS performance with varying coordinator number and density were also addressed. To show the feasibility of the proposed approach on COTS hardware, the experimental results of a working implementation based on the open source TinyOS run on a real testbed were provided. Future work will deal with further enhancements, such as the combination of time and frequency division superframe scheduling.

References

[1] F. De Pellegrini, D. Miorandi, S. Vitturi, A. Zanella, "On the use of wireless networks at low level of factory automation systems," IEEE Transactions on Industrial Informatics, vol. 2, no. 2, 2006, pp. 129-143.

[2] H. Korber, H. Wattar G. Scholl, "Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications," IEEE Transactions on Industrial Informatics, vol. 3, no. 2, 2007, pp. 111-119.

[3] D. Dzung, C. Apneseth, J. Endresen, J. Frey, "Design and implementation of a real-time wireless sensor/actuator communication system," Proc. of the 10th IEEE Conf. on Emerging Technologies and Factory Automation, ETFA 2005, pp. 10 pp. 433-442, 2005.

[4] IEEE 802.15.4 Standard (2006) Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Technology, IEEE-SA Standards Board, 2006.

[5] A. Koubâa, M. Alves, E. Tovar, "Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks," Proc. of the 27th IEEE Real-Time Systems Symposium (RTSS'06), Brazil, 2006, pp. 412-421.

[6] G. Anastasi, M. Conti, M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," *IEEE Trans. on Ind. Informat.,* vol.7, no.1, pp.52-65, Feb. 2011

[7] J. Ha, W.H. Kwon, J..J. Kim, Y.H. Kim, Y.H. Shin, "Feasibility analysis and implementation of the IEEE 802.15.4 multi-hop beacon-enabled network." Proc. of the 15th Joint Conf. on Comm.& Inf, Jun 2005.

[8] P. Raja, G. Noubir, "Static and Dynamic Polling Mechanisms for FieldBus Networks," ACM SIGOPS Operating Systems Review, vol. 27, no. 3, pp. 34-45, July 1993.

[9] S. Cavalieri, S. Monforte, A. Corsaro, G. Scapellato, "Multicycle Polling Scheduling Algorithms for FieldBus Networks," Journal of Real-Time Systems, vol. 25, no. 2-3, pp. 157 – 185, Sept.-Oct. 2003.

[10] A. Koubâa, A. Cunha, M. Alves, E. Tovar, "TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks," Real-Time Systems, vol. 40, no. 3, Dec 2008, pp. 321-354.

[11] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel, "The pinwheel: A real-time scheduling problem," in Proc. 22nd Hawaii Int. Conf. Syst. Sci. (HICSS), Jan. 1989, pp. 693–702.

[12] Z. Hanzalek, P. Jurcik, "Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee," IEEE Transactions on Industrial Informatics, vol.6, no.3, pp.438-450, Aug. 2010.

[13] IEC 62591 Ed.1.0: Industrial communication networks - Wireless communication network and communication profiles - WirelessHART (FDIS).

[14] G. Fiore, V. Ercoli, A.J. Isaksson, K. Landernäs, and M.D. Di Benedetto, "Multihop multi-channel scheduling for wireless control in WirelessHART networks," in IEEE Conf. on Emerging Technologies and Factory Automation (ETFA), 2009.

[15] A. Saifullah, You Xu, Chenyang Lu, and Yixin Chen, "Real-Time Scheduling for WirelessHART Networks," in 31st IEEE Real-Time Systems Symposium (RTSS), 2010, pp. 150-159.

[16] P. Soldati, H. Zhang and M. Johansson, Deadline-constrained transmission scheduling and data evacuation in wirelessHART networks, Proceedings of ECC, Sept. 2009.

[17] Song Han, Xiuming Zhu, Deji Chen, Aloysius K. Mok, Mark Nixon, "Reliable and Real-time Communication in Industrial Wireless Mesh Networks" to appear in the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011.

[18] G. Zhou, Y. Wu, T. Yan, T. He, C. Huang, J. A. Stankovic, and T. F. Abdelzaher. "A multifrequency MAC specially designed for wireless sensor network applications," ACM Trans. Embed. Comput. Syst., vol. 9, no. 4, pp.1-41, April 2010.

[19] K. Kunert, M. Jonsson, E. Uhlemann, "Exploiting time and frequency diversity in IEEE 802.15.4 industrial networks for enhanced reliability and throughput," Proc. 15th IEEE Intl Conf. on Emerging Technologies and Factory Automation (ETFA '10), Spain, Sept. 13-16, 2010.

[20] M. Jonsson and K. Kunert, "Towards Reliable Wireless Industrial Communication with Real-Time Guarantees," IEEE Transactions on Industrial Informatics, vol. 5, no. 4, pp. 429-442, Nov. 2009.

[21] E. Toscano, L. Lo Bello, "A Multichannel Approach to Avoid Beacon Collisions in IEEE 802.15.4 Cluster-Tree Industrial Networks." Proc. of the 14th IEEE Intl Conf. on Emerging Technologies and Factory Automation (ETFA), Spain, Sept. 2009.

[22] L. Lo Bello, E., Toscano, "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, May 2009

[23] [online], http://grouper.ieee.org/groups/802/15/pub/TG4b.html

[24] Zou You-Min, Sun Mao-Heng, Ran Peng, "An Enhanced Scheme for the IEEE 802.15.4 Multi-Hop Network," Intl Conf on Wireless Communications, Networking and Mobile Computing, WiCOM 2006, pp. 1-4.

[25] P. Muthukumaran, R. Spinar, K. Murray, et al. "Enabling low power multi-hop personal area sensor networks." In: Proc. of the 10th Intl Symp. on Wireless Personal Multimedia ommuications, India, 2007.

[26] ZigBee Alliance, "ZigBee Specification", Doc. 053474r17, Jan 2008.

[27] G. Gamba, F. Tramarin, A. Willig, "Retransmission Strategies for Cyclic Polling Over Wireless Channels in the Presence of Interference," IEEE Transact. on Ind. Informat., vol.6, no.3, pp.405-415, Aug. 2010

[28] G. Zhou, T. He, J. A. Stankovic, e T. Abdelzaher, "RID: radio interference detection in wireless sensor networks," in Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, pp. 891-901, 2005.

[29] I. Katzela and M. Naghshineh, Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey, IEEE Personal Communications, vol. 3, no. 3, June 1996, pp. 10–31.

[30] T. Camilo, J. Sà Silva, A. Rodrigues, and F. Boavida, "GENSEN: a topology generator for real wireless sensor networks deployment", In Proc. of the 5th IFIP Intl Conf. on Software technologies for embedded and ubiquitous systems (SEUS'07), pp. 436-445, 2007.

[31] Kaixin Xu, M. Gerla, e Sang Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," in IEEE Global Telecommunications Conference (GLOBECOM), vol. 1, pp. 72-76, 2002.

[32] [online] http://www.diit.unict.it/users/llobello/TR-201101.pdf

[33] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, e D. Culler, "TinyOS: An Operating System for Sensor Networks," Ambient Intelligence, 2005, 115-148.

[34] [online],http://www.xbow.com/Products/Product_pdf_files/Wireless_pd

f/TelosB_Datasheet.pdf
[35] J. Hauer, TKN15.4: An IEEE 802.15.4 MAC Implementation for TinyOS 2, Telecommunicat. Networks Group, Tech. Univ. Berlin, 2009.
[36] X. Wang and T. Berger, "Spatial channel reuse in wireless sensor networks," *Wireless Networks,* vol. 14, no. 2, pp. 133-146, Mar. 2008.

APPENDIX

In this Appendix formal proof of the theoretical results intuitively described in Section VI-A is derived.

In the first theorem, the extent to which MSS increases the schedulability space in the domain of all the possible network configurations is derived.

Theorem 1

Let *S* be a homogeneous superframe set. The MSS algorithm extends the schedulability region of the SDS algorithm in the space Π by a factor of 2.

Proof:

Consider the schedulability regions in Π depicted in Fig. 4.

Note that, for each *BI* > *SD*, the triangular surface of the network configurations space whose vertexes are (1, *SD*), (1, *BI*), and (*n*, *n*·*SD*) is schedulable using the SDS algorithm, due to eq. (8). The area of such a triangle is

$$A_{SDS}=(BI-SD)\frac{(n-1)}{2}. \tag{16}$$

Similarly, for each *BI* > *SD*, the rectangular surface of the network configurations space whose vertexes are (*1*, *2*·*SD*) (2, *n*), (1, *BI*), and (*BI*, *n*) is schedulable using the MSS algorithm, due to eq. (9). The area of such a rectangle is

$$A_{MSS}=(BI-2\mathrm{SD})(n-2). \tag{17}$$

Now it is possible to see that, for indefinitely large values of *BI* and *n*, A_{SDS} and A_{MSS} converge to R^{Π}_{SDS} and R^{Π}_{MSS}, respectively. As a result, it is possible to calculate the ratio between the surface of the schedulability regions of MSS and SDS evaluating the limit of the ratio between eq. (16) and (17) as *BI* becomes infinite:

$$\lim_{BI\to+\infty}\frac{2(BI-2\mathrm{SD})(n-2)}{(BI-SD)(n-1)}=2\frac{n-2}{n-1}, \tag{18}$$

and then evaluating the limit of eq. (18) as *n* approaches infinity:

$$\frac{R^{\Pi}_{MSS}}{R^{\Pi}_{SDS}}=\lim_{n\to+\infty}2\frac{n-2}{n-1}=2. \tag{19}$$

Eq. (19) proves the theorem. □

The second theorem proves that every SDS-schedulable network configuration in the domain of the feasible network configurations is also MSS-schedulable.

Theorem 2

Let *S* be a homogeneous superframe set. In the space of the feasible network configurations Φ, the MSS schedulability region R^{Φ}_{MSS} is a superset of the SDS schedulability region R^{Φ}_{SDS}.

Proof:

The proof is made by contradiction. Suppose a network configuration in Φ exists, that is schedulable with SDS but not with MSS. Then a feasible configuration $c_f\in R^{\Phi}_{SDS}$ must exist which does not belong to R^{Φ}_{MSS}.

For $n<2$, there is only one feasible configuration in R^{Φ}_{SDS}, which is $c_0=(SD,1)$. But $c_0\in R^{\Phi}_{MSS}$ by definition, as shown in eq. (11). This contradicts our hypothesis that $c_f\in R^{\Phi}_{SDS}$ but $c_f\notin R^{\Phi}_{MSS}$.

For $n\geq 2$, the beacon interval of c_f must be greater than $n\cdot SD$ in order to be in R^{Φ}_{SDS}, due to the SDS schedulability condition given by eq. (8). But, as we are considering $n\geq 2$, then also the MSS schedulability condition given by eq. (9) is true. Therefore also this configuration contradicts our hypothesis that $c_f\in R^{\Phi}_{SDS}$ but $c_f\notin R^{\Phi}_{MSS}$. □

In the following lemmas, the number of feasible network configurations which are also SDS- and MSS-schedulable, respectively, is derived. In other words, Lemma 1 and 2 provide a formal proof for eq. (13) and (14), respectively.

Lemma 1: Let *S* be a homogeneous superframe set and let ρ_k be the region defined as in eq. (12). The cardinality of the subset of the SDS schedulability region $R^{\rho_k}_{SDS}$ is

$$|R^{\rho_k}_{SDS}|=2^{\lfloor\log_2(k)\rfloor+1}-1$$

Proof:

First of all, notice that $R^{\rho_k}_{SDS}=R^{\Pi}_{SDS}\cap(\Phi\cap\rho_k)$.

Now, consider a generic $k\geq 1$. The set of configurations in $\Phi\cap\rho_k$ contains only the *BI* values which are a multiple of SD by a power of two, i.e., $SD, 2SD, 4SD, \cdots, 2^{\lfloor\log_2(k)\rfloor}SD$. The overall number of BI values in $\Phi\cap\rho_k$ is therefore $\lfloor\log_2(k)\rfloor$.

For a generic beacon interval *BI* in $\Phi\cap\rho_k$, the values of *n* that satisfy eq. (8) are those from 1 to *BI*/*SD*. As all the BI values can be written as $2^i\cdot SD$, it follows that the cardinality of the subset of the SDS schedulability region $R^{\rho_k}_{SDS}$ is

$$|R^{\rho_k}_{SDS}|=\sum_{i=0}^{\lfloor\log_2(k)\rfloor}2^i=2^{\lfloor\log_2(k)\rfloor+1}-1, \tag{20}$$

which proves the lemma. □

Lemma 2: Let *S* be a homogeneous superframe set and let ρ_k be the region defined as in eq. (12). The cardinality of the subset of the MSS schedulability region $R^{\rho_k}_{MSS}$ is

$$|R^{\rho_k}_{MSS}|=k\lfloor\log_2(k)\rfloor$$

Proof:

First of all, notice that $R^{\rho_k}_{MSS}=R^{\Pi}_{MSS}\cap(\Phi\cap\rho_k)$.

As shown in Lemma 1, the overall number of BI values in $\Phi\cap\rho_k$ is $\lfloor\log_2(k)\rfloor$. Eq. (9) is true for every *BI* > 2, therefore the values of *n* in $\Phi\cap\rho_k$ that satisfy eq. (8) are those from 1 to *k*. It follows that the cardinality of the subset of the MSS schedulability region is $|R^{\rho_k}_{MSS}|=k\lfloor\log_2(k)\rfloor$, which proves the lemma. □

Finally, the third theorem proves that, in the domain of the feasible network configurations, the advantage in terms of schedulability space of MSS over SDS grows as the logarithm of the number of coordinators in the network.

Theorem 3

Let *S* be a homogeneous superframe set. The following statement is true:

$$\frac{|R_{MSS}^{\Phi}|}{|R_{SDS}^{\Phi}|}=O(\log(n)). \tag{21}$$

Proof:

For every $n > 0$, we can calculate from eq. (13) the cardinality of the SDS schedulable subset of $(\Phi \cap \rho_n)$. Analogously, we can calculate from (14) the cardinality of the MSS schedulable subset of $(\Phi \cap \rho_n)$.

We define $r(n)$ as the ratio between $|R_{SDS}^{\rho_n}|$ and $|R_{MSS}^{\rho_n}|$, i.e.,

$$r(n)=\frac{n|\log_2(n)|}{2^{|\log_2(n)|+1}-1}. \tag{22}$$

Note that, as n approaches infinity, $|R_{SDS}^{\rho_n}|$ and $|R_{MSS}^{\rho_n}|$ converge to $|R_{SDS}^{\Phi}|$ and $|R_{MSS}^{\Phi}|$, respectively. However, the limit of $r(n)$ as n becomes infinity does not exist. In fact, $r(n)$ always oscillates because of the floor functions, so it does not converge to any defined value.

Nevertheless, it is possible to see that $r(n)$ has infinite local minima which are located at $n=2^i, i=2,3,4,\cdots$. In fact, while the numerator of formula (22) is strictly increasing in the interval $(2^i,2^{i+1})$, the denominator is constant. Moreover, we have:

$$\lim_{n\to 2^i+} r(n)=\frac{2^i\cdot i}{2i-1} \tag{23}$$

$$\lim_{n\to 2^i-} r(n)=\frac{2^i\cdot(i-1)}{2i-3}, \tag{24}$$

and it is possible to prove that the right-hand limit in formula (23) is smaller than the left-hand limit in formula (24) if $i > 3/2$.

It follows that, if we define $r_{inf}(n)$ as

$$r_{inf}(n)=\frac{n\log_2(n)}{2n-1}, \tag{25}$$

we have $r_{inf}(n)\le r(n)$ for each $n\ge 2$.

As n approaches infinity, we have

$$\frac{|R_{MSS}^{\Phi}|}{|R_{SDS}^{\Phi}|}=r(n)\ge r_{inf}(n)=O(\log(n)), \tag{26}$$

which proves the theorem. □