Virtualization of Wireless Sensor Networks Through MAC Layer Resource Scheduling

Elena Uchiteleva, *Student Member, IEEE*, Abdallah Shami, *Senior Member, IEEE*, and Ahmed Refaey, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a joint throughput and time-resource allocation scheme for the virtualization of IEEE 802.15.4-based wireless sensor networks (WSNs). Virtualization is realized through utilization of the guaranteed time slot (GTS) mechanism of cluster-tree topology to schedule resources on a media access control (MAC) layer. We develop a scheduler that is located in the personal area network (PAN) coordinator and that virtualizes the network into an aggregate of independent profiles, assigning the available resources to each profile with end-to-end (ETE) delay guarantees. The scheduler solves the problem of managing resources available in the network in an optimization framework, taking into consideration the individual profile and sensor requirements. Moreover, it uses the proposed heuristic fair resource allocation (FRA) algorithm to derive the solution in polynomial time. We validate the scheduling performance via discrete event simulation (DES) and compare the proposed FRA algorithm with round robin (RR) and proportionally fair (PF) scheduling algorithms in several scenarios. The proposed scheme demonstrates efficient resource management while maintaining profile isolation in all cases, whereas other algorithms lead to increased latency and lower throughput in the network.**

***Index Terms*— IEEE 802.15.4, resource allocation, scheduling, virtualization, wireless sensor networks, zigbee.**

I. Introduction

Wireless Sensor Network (WSN) virtualization is an emerging technology that targets domain specific and task-oriented networks and enables them to support multiple applications for various domains, thus providing more flexibility, diversity, efficiency and increased manageability. This technology is based on the idea of coexistence of heterogeneous nodes in shared physical sensor substrate (*i.e.* infrastructure), and is a core for the concepts of Internet of Things (IoT), Smart City, Smart Home and Intelligent Transportation Systems (ITS). The traditional WSNs are usually deployed to accommodate only one particular application and are not suitable for reuse by newer applications or sharing by different groups of users/operators [1], [2]. This is inefficient and leads to redundancy when new applications are deployed or functional heterogeneity needs to be implemented. Therefore, the subject of WSNs virtualization is very relevant and important.

Manuscript received September 9, 2016; revised December 5, 2016; accepted December 6, 2016. Date of publication December 22, 2016; date of current version February 7, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by Mircom Technologies Ltd. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ashish Pandharipande.

E. Uchiteleva and A. Shami are with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada (e-mail: euchitel@uwo.ca; ashami2@uwo.ca).

A. Refaey is with the Electrical and Computer Engineering Department, Manhattan College, Bronx, NY 10471 USA, and also with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada (e-mail: ahmed.hussein@manhattan.edu).

Digital Object Identifier 10.1109/JSEN.2016.2643670

Virtualization is defined as the abstraction of physical computing or network resources into logical units dedicated to multiple independent applications and/or users [2], [3]. In our work, we virtualize a WSN's physical resources into logical units, thus creating a pool of resources that can be used by any group or subset of a WSN's nodes dedicated to one application at a time. Each subset is unaware of the underlying virtualization processes or of the existence of other subsets. The set of characteristics that we adopted to describe the WSN virtualization is [4]:

- *Networking technology*, or the attributes of networks that we perform virtualization on. In our case, WSN technology (or more specifically, IEEE 802.15.4).
- *Layer of virtualization*, which specifies the layer of network stack where the virtualization was introduced. In this paper, we consider MAC layer virtualization.
- *Granularity of virtualization*, which defines granularity of the extent to which each logical subset can manage itself. This can be a single virtual node, links between nodes or a whole network. Our case deals with isolated groups of nodes (or *profiles*, as described below).

Current research on WSNs virtualization can be classified into two main categories: node-level virtualization and network-level virtualization. *Node-level virtualization* enables node heterogeneity, which means that a sensor node can essentially become a multipurpose device. This can be achieved using Sensor Operation System-based solutions [5] or using Virtual Machine-/Middleware-based solutions [2]. Although these solutions ensure that the nodes are capable of the concurrent execution of multiple tasks, they do not enable sensors to form isolated *profiles*, or groups of logically connected applications dedicated to different domains of WSN deployment, *e.g.* smart home, health care, industry or agriculture. On the other hand, *network-level virtualization* focuses on domain-segregated sensor groups, introducing the notion of network sharing and sharing management (scheduling), which is of especial interest. For instance, a network of sensors installed in a private home could be simultaneously used for hazard monitoring inside the house (such as smoke or carbon monoxide detection) as well as for health-related purposes by the householders (*e.g.*, a doctor or nurse monitoring the health of elderly patients in the house). Profiles can also be formed in accordance with their priority or importance of the application they support. Consider an example of a ZigBee network installed in an apartment building for its condition, safety and security monitoring. According to Underwriters Laboratories Inc. Standards (UL 864) [6] and National Fire Alarm and Signaling Code (NFPA 72) [7], fire hazards are on a higher priority to be reported than plumbing and security hazards or energy efficiency of the building; therefore, the sensors network in this building can be virtualized into prioritized profiles, each supporting a different application. Packets of profiles with higher priority will be routed to the PAN coordinator quicker than the packets of profiles with lower priority.

Network-level virtualization can be classified into two types: virtual network/overlay-based solutions and cluster-based solutions [2], [8]. In virtual network/overlay-based solutions, logical networks are created on top of an existing physical network to form virtual application-specific groups. References [8]–[12] discuss the concurrent utilization of nodes and the establishment of virtual links (between sensors in different administrative domains); others have investigated the routing of packets through these links to connect other IP-smart objects to the network without use of conventional gateways [13]–[15]. Works [16] and [17] propose architectures to utilize both node-level and network-level virtualization towards realization of IoT concept. Although WSN deployment and connection to the Internet and cloud for virtualization purposes have been extensively discussed, not much detail has been provided about the physical implementation of these schemes or of the deployment of any lower-level standard. Therefore, in this work we target the realization of virtualization concept on lower layers of communication stack.

The design features of virtualization that we focused on in our paper are [4]:

- *Abstraction*. A WSN is abstracted into virtual profiles. The network resources are then divided as logical units between the abstract profiles. Thus, the virtualization can be viewed as the abstraction of physical resources of a network.
- *Flexibility and heterogeneity.* Virtualization on lower layers introduces better flexibility and heterogeneity [4]. For example, resources abstraction allows dismissal of concern about kinds of applications running by profiles. The only attribute which matters is general type and amount of traffic produced by applications. This results in higher flexibility. Moreover, MAC layer virtualization enables facilitation of different upper-layer technologies on the same physical WSN, which increases heterogeneity.
- *Manageability (allocation).* The random nature of the CSMA/CA (the default medium access technique used by the IEEE 802.15.4) [18] may diminish the quality of service (QoS) level in the network. Thereby, a service provider cannot make any guarantees on packet delay, or achieved rates by separate nodes or groups of nodes. Moreover, monopolization by a group of nodes with heavy traffic is probable, which means that there is a possibility of "choking" other less active groups of sensors, which nonetheless might have important information to deliver. Therefore, manageability of a network in terms of dedicated resources is a critical factor in virtualization of a WSN to satisfy the service level agreement (SLA). In this paper, we use the GTS mechanism as the main mechanism for the channel access. This allows us to introduce and attain minimum QoS guarantees for each profile, such as delay and throughput.
- *Isolation.* The manageability of the network introduces the concept of isolation between profiles with respect to resources allocation. This enables coexistence of logically separated and independent subsets of nodes supporting various applications on the same WSN (provided that the network operates within the Admission Control constraints). This facilitates development of network management business models, in which different profiles are managed by separate service providers with minimum service guarantees. The resource virtualization of a network will ensure that the current state of a provider will not affect the states of others.

A. Contribution

In this paper we propose a TDMA-based scheduler for WSNs MAC layer. This scheduler provides a resource management solution for isolated profiles on a shared physical network of sensors, thus enabling the virtualization on a WSN. The purpose of this scheduler is to allocate the available network resources in accordance with the requirements of individual profiles while maximizing total throughput in the network and providing ETE guarantees for critical data.

We investigate the IEEE 802.15.4 standard, which is the most popular standard for layers 1-2 in low-rate, low-power WSNs. Virtualization is achieved using the synchronization mechanism and GTSs supported by the standard. We formulate and solve the problem in a utility-optimization framework, while choosing the class of linear utility functions, which enables coexistence of throughput- and time-resource provisioning to satisfy the requirements of profiles with different types of data: bursty and periodic respectively.

Since virtualization enables the separation of different Service Providers (SPs) both from one another and from the Sensor Infrastructure Providers (SInPs) [3], our work presents the potential for new deployments and management of WSNs, which would introduce new business cases, services and models in the context of sensor networks.

The rest of the paper is organized as follows. In Section II we discuss research available on virtualization in lower communication layers. Section III reviews relevant theoretical background on topology and frame structure of IEEE 802.15.4 and summarizes the assumptions made regarding the framework. Section IV deals with problem formulation. We describe the proposed FRA heuristic algorithm In Section V. Section VI presents simulation results and discussion, followed by the conclusion in Section VII.

II. Related Work

Much of available work on WSNs virtualization on lower layers deals with cluster-based solutions, where physical partition of all nodes is usually applied and each group forms a task-oriented cluster. For example, [19] and its extension [20] discuss dynamically self-organizing tree-based VSNs and their maintenance, as well as inter-VSN and intra-VSN communication. An extensive research has been done on the topic of virtual clusters (VC) formation and scheduling on MAC layer in WSNs towards reducing the network's energy consumption. Thus, several energy-efficient contention-based MAC protocols and their amendments were proposed, such as Sensor MAC (S-MAC) [21], its enhanced version, Timeout MAC (T-MAC) [22], and the modifications for networks with moving nodes: Mobile S-MAC (MS-MAC) [23] and Mobile T-MAC (MT-MAC) [24]. The basic idea is that the nodes form VCs with common local sleeping schedule and use contention-based channel access when they are awake. Some algorithms were proposed to improve these protocols, for instance [25]–[28]. The authors of [28] proposed to use separate channels for inter- and intra- cluster communications and to combine contention-based carrier sense multiple access (CSMA) with contention-free-based time division multiple access (TDMA) periods for urgent data. During the contention-free period each VC assigns time slots to sensors within it with the highest priority, based on priority index calculations. Whereas these works deal with organizing sensors into groups, they target the energy consumption and life-span of a WSN, and not profiles/flows separation and QoS control over them, which are essential for WSN virtualization.

The problem of data flows prioritization on MAC layer has already been addressed by some works, but not in a context of WSN virtualization. Thus, research works in MAC layer of IEEE 802.15.4 WSNs [29] and [30] present models, which facilitate priority-based service differentiation by means of assigning various Backoff Exponent (BE) in CSMA with collision avoidance (CSMA/CA) algorithm to different service classes. In [31] authors use the concept of virtual collision domain and present two algorithms to calculate the collision-free virtual domain and to dynamically adjust the backoff period to serve rate-sensitive data with minimal rate requirements. While these works improve the network's throughput and address the issue of rate requirements, due to use of contention-based channel access the ETE delay cannot be guaranteed, and hence cannot be used for networks with critical data profiles.

Some other works dealt with TDMA-like scheduling and showed its advantages over the contention-based scheduling. Authors of [32] introduced a platform for development of cross-layer protocols in multi-hop mobile ad hoc networks (MANET), which virtualizes the network on MAC layer in TDMA fashion to allow fair comparison between various network protocol stacks (virtual protocols) in almost identical propagation environment. Each virtual protocol has an access to a virtual link layer and virtual time. In this case, the TDMA scheme had an advantage of preventing interference between different stacks globally. In [33] a multimode hybrid MAC protocol (MH-MAC) was introduced. This protocol can switch between asynchronous and synchronous modes, with or without contention, to support WSNs with heterogeneous nodes producing infrequent bursts of data. It was shown that the data bursts are handled better by the contention-free synchronous mode than by the asynchronous contention-based one.

Time Division Multiplexing (TDM) is a popular technique for isolation between virtual interfaces on MAC layer in other wireless technologies [34]. For example, in Wireless Local Area Network (WLAN) virtualization approach, called *virtual WiFi* [35], the time domain multiplexing was used to separate between multiple virtual MACs of different virtual machines to enable their sharing of the same physical network without interference. In [36] the TDM was implemented to multiplex virtual access point interfaces to one physical collision-free interface. The authors of [37] adopted TDM to isolate between embedded Virtual Networks (VN) for multicasting in a Wireless Mesh Network (WMN). The TDM-based link virtualization was also used in works [38]–[40] dealing with WLAN virtualization. Others used frequency-division multiplexing (FDM) which virtualizes the transmission in frequency domain [41]–[43].

Virtualization of cellular networks, such as Worldwide Interoperability for Microwave Access (WiMax), includes bandwidth-based (rate) and resource-based (spectrum or time slots) scheduling on MAC layer in each orthogonal frequency division multiplexing (OFDM) frame [34], [44]. The scheduling depends on the slice (or profile) type and the utility functions at the base station which were agreed between the mobile network operator (MNO) and the service provider (SP) [45]–[47]. In Long-Term Evolution (LTE) systems the resources are scheduled as Physical Resource Blocks (PRBs), which are chunks of spectrum and are the smallest units that the LTE MAC layer can allocate to a user [48]. The PRBs are allocated based on pre-defined criteria, *e.g.* data rates, power, bandwidth, traffic load, interference, channel conditions or a combination of them [34], [49]–[52]. Virtualization on a WSN, nonetheless, is a challenging task, because these networks are very restricted in their available resources as compared to other wireless networks, especially in available bandwidth. Usually, only one narrow-band channel is allowed for the transmission in a whole PAN; therefore, the frequency diversity or multiplexing cannot be exploited in this case. However, though resource scheduling and simultaneous isolation maintenance are difficult to achieve, the contention-free TDMA schemes show a lot of potential for the realization of WSNs sharing and virtualization.

III. Theoretical Background and Assumptions

A. Topology and Resources

The IEEE 802.15.4 enables three basic types of network topology: star, mesh and cluster-tree [18], [53]. In *star network topology*, all devices are directly connected to the PAN coordinator and can communicate with other devices on the network strictly through the coordinator. In *mesh network topology*, each node can directly communicate with any other node that is within radio range [54]. The data is routed in ad-hoc fashion through several "hops" to the destination. *Cluster-tree network*

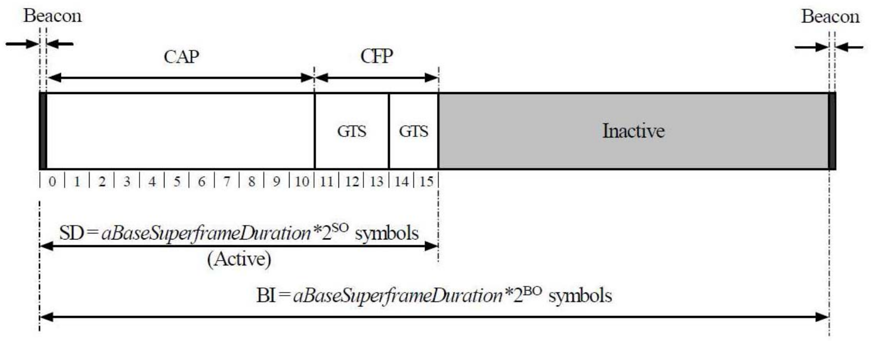
Beacon
CAP
CFP
GTS
GTS
Inactive
Beacon
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
SD = aBaseSuperframeDuration*2^SO symbols (Active)
BI = aBaseSuperframeDuration*2^BO symbols

Fig. 1. The Superframe structure of IEEE 802.15.4 [18].

topology is a special case involving a mesh network with only one possible routing path between two nodes [54]. The cluster-tree routing protocol is much lighter than the mesh routing one. Moreover, IEEE 802.15.4's beacon-enabled mode introduces a synchronization mechanism. This mode allows ETE delay guarantees to be made on a per-cluster basis using GTS in the superframe contention-free period (CFP).

Topology plays a very important role in virtualization. Although mesh networks are more robust, they are not able to support resource management due to pure CSMA/CA protocol employed; they also waste bandwidth and energy because of overheads and redundant routing. On the other hand, cluster-tree topologies support synchronization, which can be effectively used for isolating groups and sharing resources between them.

B. IEEE 802.15.4 Beacon-Enabled Mode and the Superframe Structure

The IEEE 802.15.4 MAC sub-layer supports two modes: beacon-enabled and non-beacon-enabled. In the non-beacon-enabled mode, nodes use an unsynchronized random channel access protocol, which is unable to give any time or resource guarantees. In contrast, the beacon-enabled mode uses periodical beacon frames (superframes) to synchronize the nodes with the PAN coordinator. According to the standard [18], the superframe structure of the beacon-enabled mode is defined by the coordinator and is bounded by two beacons; it may also include an inactive period, (see Fig. 1). The active portion of the superframe is divided into 16 equal slots, within which transmission is allowed, and contains a contention access period (CAP) and a CFP. During the CAP, nodes must compete for the channel (through a slotted CSMA/CA random access mechanism), whereas in the CFP they are allocated GTSs during which they can transmit without any contention.

The beacon is transmitted during the first slot of a frame. The GTSs form the CFP, which always appears at the end of the active portion of the superframe and starts immediately following the CAP, as shown in Fig. 1. The coordinator can allocate up to seven GTSs, and a GTS may occupy more than one slot period [18]. However, a minimum CAP portion of 440 symbols (1760 bits) should remain for contention-based access of other network devices or for new devices wishing to join the network. If no GTSs are allocated, the CAP lasts for the whole active period of a frame.

The beacon order (BO) attribute in Fig. 1 describes the interval at which the coordinator shall transmit its beacon frames, and the superframe order (SO) attribute describes the length of the active part of the superframe. These are related to the beacon interval (BI) and superframe duration (SD) as follows:

$$SD = 3840 \cdot 2^{SO}\ [bits] = 15.36 \cdot 10^{-3} \cdot 2^{SO}\ [sec], \quad (1)$$

$$BI = 3840 \cdot 2^{BO}\ [bits] = 15.36 \cdot 10^{-3} \cdot 2^{BO}\ [sec], \quad (2)$$

where $0 \leq SO \leq BO \leq 14$ must hold (see Fig. 1).

The MAC layer requires a finite amount of time to process the received data from physical layer (PHY). Therefore, any successive frames should be separated by at least one inter-frame spacing (IFS) period [18].

C. Assumptions for the Framework

We assume a single WSN with one PAN coordinator with the remaining devices being either routers or end devices. We also assume a cluster-tree topology of unity depth, where all end devices and routers are direct children of the coordinator. This topology inherits the simplicity of a star topology, while at the same time being enhanced by the synchronization mechanism (the superframe structure). The beacon frame is considered to have only an active period (an extension to the case with both active and inactive periods is straightforward). The upper layers can be defined by any appropriate standard which operates on those layers, *e.g.* ZigBee.

In order to fulfil the *flexibility and heterogeneity* feature of virtualization, we rather assume a generic representation of data produced by a network than a set of particular running applications. Thus, we consider only two main and most widely adopted types of data generated by the network (and, respectively, profiles generating these data): periodical data [17], [31]–[33], [55]–[59], and bursty data [28], [33], [60]–[66]. These two data types and their combination generally describe any kind of application that can run on a network. Periodical data is generated by sensors at predefined time instances and has a known volume. This can include profiles monitoring the environment or the status of a home. In contrast, profiles, that generate bursty data, require elevated throughputs at random (often rare) time instances, keeping quiet during other times. One good example involves municipal authorities monitoring an accident or disaster (for example, tracking a bush fire or investigating an automobile collision).

Our goal is to develop a MAC-layer scheduler that will utilize the same network infrastructure to accommodate both types of profiles. Approach of resource allocation based on requirements of various data classes (traffic) was investigated in wired and mobile networks and was shown to be effective for scheduling in networks hosting applications with different needs. Therefore, it is a promising solution for heterogeneous WSNs as well, where the profiles may have different demands for resource allocation based on the type of data they produce [44], [67], [68]. Thus, we can separate resource provisioning into two logical types: time-slot provisioning and throughput provisioning. Time-slot provisioning is aimed at periodical profiles which require guaranteed time slots each frame (or every several frames) during which they can send periodical data to the sink. These time slots are allocated during the CFP of the frame. Throughput provisioning is aimed towards bursty profiles which require throughput guarantees during random events, which can be achieved by allocating the remaining time slots. In this way, bursty data devices benefit from increased throughput during the events, whereas the periodical data devices benefit from guaranteed and uninterrupted transmission during the CFP of each scheduling period. Akin division of a network into logical profiles and resource scheduling based on these profiles requirements fulfills the virtualization characteristic of *abstraction*, discussed in Section I.

We assume that the Admission Control is performed by the Sensor Virtualization Network Service Provider (SVNSP), and the cumulative reserved resources of all profiles, as well as generated traffic do not exceed the total bandwidth of the network.

IV. Problem Formulation

A. Optimization Framework

Let us consider two groups of profiles P_b and P_p. The profiles in P_b generate bursty type of data and require minimum throughput guarantees, whereas the profiles in P_p generate periodical data and require minimum time resource guarantees. We also assume that the beacon frame consists of minimal CAP and that the rest of the frame is the CFP. For the sake of simplicity, we assume unacknowledged packet transmission with no security setting.

The scheduler allocates minimum required resources for profiles in P_b and P_p, while the remaining GTSs are used to provide extra throughput to decrease delay in the network. These allocations are variable and are updated by the scheduler every scheduling period, based on individual profiles' requirements, which satisfies the *manageability* characteristic of underlined resource virtualization process. The frame is composed of three main parts: T_{cap}, the minimum CAP; T_{cfp1}, the CFP for bursty data-type profiles; and T_{cfp2}, the CFP for periodical data-type profiles. Thus:

$$T_{Fr} = T_{cap} + T_{cfp1} + T_{cfp2}. \tag{3}$$

TABLE I

Summary of All Parameters Used in the Analytical Model

Param.	Description
P_b	Group of profiles that require throughput based provisioning
P_p	Group of profiles that require time slots based provisioning
S_p	Number of sensors in a profile $p \in P_b, P_p$
B	Transmission rate of the network in a specific band in kb/s
T_{Fr}	Duration of one frame in seconds
T_{cap}	Duration of the minimum CAP in seconds per frame
T_{cfp1}	Duration of CFP allocated to profiles p in P_b per frame in seconds
T_{cfp2}	Duration of CFP allocated to profiles p in P_p per frame in seconds
T_{cfp}	Total duration of CFP per frame in seconds
T_{ifs}	The IFS time in seconds
T_{slot}	Duration of 1 time slot in seconds
N	Scheduling period: the frequency of scheduling update in frames
$q_{(i,p)}$	Queueing size of a sensor i in a profile $p \in P_b, P_p$
q_p	Cumulative (across sensors) queue of a profile $p \in P_b, P_p$
$St_{(i,p)}$	State of the buffer of sensor i in a profile $p \in P_b, P_p$
St_p	Cumulative (across sensors) status of the profile $p \in P_b, P_p$
Q	Sensor's buffer capacity in packets (same for all sensors)
S_{thr}	The buffer state threshold which triggers the minimum reserved throughput allocation, r_p^{rsv}
r_p^{rsv}	The minimum guaranteed throughput that the scheduler attempts to provide to a profile $p \in P_b$ per one scheduling period
r_p	The cumulative throughput in kb/s achieved by a profile $p \in P_b$ till any instant of time
d_p	Number of bits transmitted during one scheduling period from a profile $p \in P_b$ to the coordinator
$U_p(r_p)$	The utility for the profile $p \in P_b$ if throughput of r_p was achieved by this profile
R_b	Overall throughput of P_b
t_p^{rsv}	Minimum reserved amount of time slots per period allocated to a profile $p \in P_p$ per one scheduling period
t_p	Achieved cumulative amount of time slots for a profile $p \in P_p$ till any instant of time
$V_p(t_p)$	The utility function for the profile $p \in P_p$ if amount of t_p slots was allocated to this profile

Next, we can define the total contention-free period per frame:

$$T_{cfp} = T_{cfp1} + T_{cfp2} = T_{Fr} - T_{cap}. \tag{4}$$

The summary of all parameters is given in Table I.

We set r_p^{rsv} and r_p as the minimum reserved cumulative throughput and cumulative throughput achieved by a profile p during a scheduling period, respectively. Thus, we can define $U_p(r_p)$ to be the utility for the profile p if it achieves a throughput of r_p. Let us assume converge-cast for P_b (*i.e.*, data is forwarded from profiles to the coordinator). If a profile p in P_b transmits d_p bits during M frames (M is an arbitrary number), the effective throughput R_b for the group P_b can be expressed as follows (see Fig. 2):

$$R_b = \frac{\sum\limits_{p \in P_b} d_p}{M(T_{Fr} + T_{ifs})} = \sum_{p \in P_b} r_p, \tag{5}$$

where T_{ifs} is the duration of IFS period, r_p is the cumulative throughput achieved by a profile p until time instant $t = M \cdot (T_{Fr} + T_{ifs})$ (the scheduler's update interval). On the other hand, the overall amount of data transmitted by P_b to the coordinator during this period is given by:

$$\sum_{p \in P_b} d_p = MB \cdot T_{cfp1}, \tag{6}$$

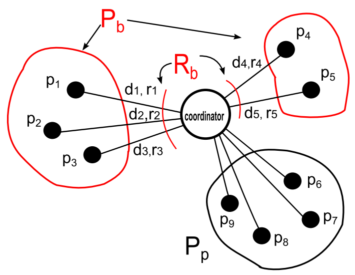

Fig. 2. Example of groups P_b and P_p, throughput of group P_b.

where B is the transmission rate of an IEEE 802.15.4 network within a particular band. In the 2.4-GHz band this rate equals 250 kb/s [18]. We assume that the signal to noise ratio (SNR) is high and does not affect the transmission rate B. From (5) and (6), we can derive the required duration of the CFP per frame for the bursty data profiles as a function of cumulative throughput:

$$T_{cfp1} = \frac{T_{Fr} + T_{ifs}}{B} \sum_{p \in P_b} r_p. \tag{7}$$

Now, let us assume a profile p in P_p. The quantities t_p^{rsv} and t_p represent the minimum reserved cumulative amount of time resources (or time slots) and the achieved cumulative amount of time resources by a profile p during a scheduling period, respectively. We also denote $V_p(t_p)$ the utility function for a profile p if the amount t_p of time resources was allocated to this profile. Note that t_p^{rsv} and t_p are whole numbers, as a fraction of a slot cannot be allocated. There are different kinds of utility functions available in the literature which suit different business models: for example, see [44]. We choose the case of weighted linear utility functions and consider two separate functions, U_p and V_p to accommodate both types of profiles, assumed in Section III-C:

$$U_p(r_p) = w_p r_p, \quad V_p(t_p) = v_p t_p. \tag{8}$$

Here w_p and v_p are the weights of profiles' utility and are deterministic functions of profiles' cumulative queues q_p, *i.e.* $w_p = f(q_p)|_{p \in P_b}$ and $v_p = f(q_p)|_{p \in P_p}$. The assignment of these weights is arbitrary and depends on the business model being used by the particular service provider. These weights can also reflect the price of the service in accordance with the reserved throughput and time resources per profile. We discuss the weight-calculation algorithm for our case in Section IV-B. The objective of the scheduler for one scheduling period, or N frames, can be formulated as follows:

$$\textbf{maximize} \sum_{p \in P_b} U_p(r_p) + \frac{B \cdot T_{slot}}{N(T_{fr} + T_{ifs})} \sum_{p \in P_p} V_p(t_p), \tag{9}$$

$$\textbf{subject to } r_p \geq r_p^{rsv} \quad if \quad p \in P_b, \tag{10}$$

$$t_p \geq t_p^{rsv} \quad if \quad p \in P_p, \tag{11}$$

$$\sum_{p \in P_b} r_p + B \frac{T_{slot}}{N(T_{fr} + T_{ifs})} \sum_{p \in P_p} t_p \leq B \frac{T_{cfp}}{T_{fr} + T_{ifs}}, \tag{12}$$

where t_p are whole numbers and $r_p \geq 0$. The constraint in (12) is the total resource constraint per N frames, and T_{slot} is a slot duration. The right-hand sides of the constraints are input parameters and have known values at the start of each scheduling period.

The admission control condition, which is a sufficient condition to ensure the feasibility of the problem (9), guarantees that the reserved resources do not exceed the maximum effective bandwidth of the system:

$$\sum_{p \in P_b} r_p^{rsv} + B \frac{T_{slot}}{N(T_{fr} + T_{ifs})} \sum_{p \in P_p} t_p^{rsv} \leq B \frac{T_{cfp}}{(T_{fr} + T_{ifs})}. \tag{13}$$

The problem (9) is the mixed-integer linear programming (MILP), and can therefore be effectively solved by a Greedy Algorithm [52], [69]–[71]. In Section V-A, we propose a reduced-complexity, suboptimal Fair Resource Allocation (FRA) algorithm, which effectively solves the above optimization problem in polynomial time.

The above resource sharing scheme is aimed at fulfilling the final important virtualization characteristic of *isolation* between profiles, which will be validated via simulation in Section VI.

B. Buffer Status Feedback

As was mentioned in Section IV-A, we assume that the weights of utility functions $U_p(r_p)$ and $V_p(t_p)$ in (8) depend on cumulative queueing sizes (across the sensors) of individual profiles, *i.e.* $w_p = f_1(q_p)|_{p \in P_b}$ and $v_p = f_2(q_p)|_{p \in P_p}$. We are interested in giving the privilege of extra allocated resources to the profiles with longer queues to avoid buffer overflow. This could be done using the adaptive weight allocation in the utility functions. Since the problem we solve involves maximization of a linear function, a larger weight will cause the algorithm to allocate a greater amount of resources to the corresponding profile. Let us assume that all sensors' buffers have capacity of Q packets and that the length of a buffer is divided into four states. The state $St_{(i,p)}$ corresponds to the state of a sensor i in a profile p with queueing size of $q_{(i,p)}$ packets:

$$St_{(i,p)}(q_{(i,p)}) = \begin{cases} 1, & 0 \leq q_{(i,p)} \leq \frac{1}{4}Q; \\ 2, & \frac{1}{4}Q < q_{(i,p)} \leq \frac{1}{2}Q; \\ 3, & \frac{1}{2}Q < q_{(i,p)} \leq \frac{3}{4}Q; \\ 4, & \frac{3}{4}Q < q_{(i,p)} \leq Q, \end{cases} \tag{14}$$

for $\forall i \in S_p$ and $\forall p \in P_b, P_p$, where S_p is the total number of sensors in a profile p. The state can be represented by two bits and can be incorporated into the packet's header or into the data payload. Each scheduling period, a profile updates the scheduler with its current cumulative state:

$$St_p = \sum_{i=1}^{S_p} St_{(i,p)}, \quad \forall p \in P_b, P_p. \tag{15}$$

The scheduler also recalculates corresponding normalized weights each scheduling period as follows:

$$w_p(St_p) = \frac{St_p}{\tilde{S}}, \quad for\ \forall p \in P_b, \tag{16}$$

$$v_p(St_p) = \frac{St_p}{\tilde{S}}, \quad for\ \forall p \in P_p, \tag{17}$$

where $\tilde{S}$ is an L^1 norm, or $\tilde{S} = \sum_{p \in P_b, P_p} St_p$. Note, that all weights sum up to unity, *i.e.* $\sum_{p \in P_b} w_p + \sum_{p \in P_p} v_p = 1$. Thus, weights of profiles are functions of their cumulative statuses, which reflects queuing sizes of sensors in individual profiles. As shown in Section IV-A, the weights are incorporated into the objective function, and the scheduler allocates extra resources every N frames based on this information. Profiles with larger weights will be allocated more extra resources (in addition to the reserved ones). The granularity of buffer states can be variable and depends on the trade-off between the length of the header and the desired fairness and efficiency of the scheduler. If more states are to be considered, the analysis is the same as above.

C. Reserved Throughput Allocation With Buffer Feedback

Bursty profiles produce increased amounts of data only when random events occur, during which they require their throughput guarantees. Otherwise, they produce very little or no data at all. For efficiency, the reserved resources should be allocated only in case of an event. Hence, the scheduler needs to be able to detect events in bursty profiles. When some bursty profiles do not indicate an event, the amount of their assigned resources can be lower than that reserved without jeopardizing their performance, thereby allowing other profiles to benefit from extra resources.

A possible way to track events in bursty profiles is by tracking their cumulative queuing sizes. Profiles that are experiencing an event naturally have more queuing packets in their buffers, and, the opposite, profiles that are not experiencing an event have a smaller number of queuing packets. Assuming that the scheduler is updated about the buffer status of each profile, as described in Section IV-B, we propose a simple scheme for reserved resource allocation based on event detection. In this method, we define the thresholding value for a profile's cumulative buffer status, below which the "no event" state is assumed and the reserved throughput can be reduced. If a profile's status is above the threshold, the "event" state is assumed, and the reserved throughput is set to the initial r_p^{rsv}. Thus, the cumulative status of all sensors in a profile defines whether this profile is to attempt to achieve the reserved throughput. For example, let us assume four profiles p_1, p_2, p_3, and p_4 in P_b. Let us also assume that at the start of a certain scheduling period cumulative queueing statuses of these profiles appear as shown in Fig. 3. The quantities St_{p_1}, St_{p_2}, St_{p_3}, and St_{p_4} represent the cumulative statuses of these profiles (at the start of a scheduling interval) respectively. The value S_{thr} is the buffer state threshold that triggers the minimum reserved throughput allocation in bursty profiles, r_p^{rsv}. If an event occurs, the queuing size in a profile buffer will exceed S_{thr}, and the initially reserved resources will be allocated to this profile. Otherwise, the allocated resources will be reduced, resulting in better utilization.

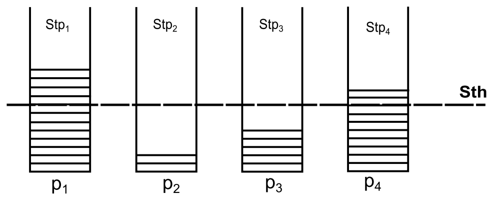

Fig. 3. An example of cumulative buffer statuses of profiles p_1, p_2, p_3, $p_4 \in P_b$.

The reserved throughput constraint of the optimization problem in (10) should be modified to reflect bursty profiles' statuses. To do so, we define an Indicator function $I_{\tilde{y}}^t(St_p)$ as follows:

$$I_{\tilde{y}}^t(St_p) = \begin{cases} 1, & St_p \in \tilde{y}; \\ \frac{1}{2} I_{\tilde{y}}^{t-1}(St_p), & St_p \notin \tilde{y}. \end{cases} \tag{18}$$

Here $\tilde{y}$ is a subset of all cumulative profiles statuses St_p equal to or greater than S_{thr}, and t is the current scheduling time instance. We can then redefine the cumulative throughput:

$$\tilde{r}_p^{rsv} = \max\{r_p^{rsv} I_{\tilde{y}}^t(St_p), r_{min}\}, \tag{19}$$

where r_{min} is the minimum threshold of allocated resources to ensure that a profile is not cut completely. When St_p of a profile p is greater than the threshold S_{thr}, the Indicator function equals 1 and the reserved throughput r_p^{rsv} is assigned to this profile; otherwise, the the Indicator function is halved and the guaranteed throughput during the CFP is reduced with each new scheduling period until it reaches a predefined minimum value r_{min}. The overall optimization problem takes the form of:

$$\textbf{maximize} \sum_{p \in P_b} U_p(r_p) + \frac{BT_{slot}}{N(T_{fr} + T_{ifs})} \sum_{p \in P_p} V_p(t_p), \tag{20}$$

$$\begin{aligned} \textbf{subject to}\ & r_p \geq \tilde{r}_p^{rsv}\ if\ \ p \in P_b, \qquad (21) \\ & t_p \geq t_p^{rsv}\ if\ \ p \in P_p, \\ & \sum_{p \in P_b} r_p + B\frac{T_{slot}}{N(T_{fr} + T_{ifs})} \sum_{p \in P_p} t_p \\ & \leq B\frac{T_{cfp}}{T_{fr} + T_{ifs}}, \qquad (22) \end{aligned}$$

where t_p are whole numbers, $r_p \geq 0$ and $\tilde{r}_p^{rsv}$ is defined in (19). The Indicator function $I_{\tilde{y}}^t(St_p)$ equals 1 for the first scheduling decision and is then updated each period. Note, that the threshold S_{thr} can differ from profile to profile, allowing more flexibility in the modeling. However, in simulations presented in Section VI, we set this threshold value as a constant for all sensors to simplify matters.

V. Solution by Greedy Algorithm

The optimization problem formulated in Section IV is an MILP and can be solved using the brunch-and-bound technique. This method finds the global optimum of any linear programming (LP) problem by enumerating the points in a subproblem's feasible region [72]. Because this algorithm searches for the global optima, it is computationally complex and cannot be used in real-time scheduling. We propose a reduced-complexity heuristic FRA algorithm, which belongs to the family of Greedy Algorithms, in order to solve the optimization problem in (20). This algorithm is suboptimal, but distributes resources according to each profile's weight (see Section IV-B) while ensuring that the minimum reserved requirements are met. First, the algorithm performs the reserved resource allocation. Then, the remaining resources are allocated proportionally according to each profile's weight. As seen in the previous section, weights are functions of the buffer state; thus, profiles with longer queues will benefit by being allocated more extra resources. This efficiently prevents buffer overflow while preserving the concept of network virtualization. In addition, profiles with small queues, or no queue, are still allocated a fraction of extra resources, thus ensuring that they can transmit any newly arriving packets and reducing the overall delay.

A. Fair Resource Allocation Algorithm

The FRA algorithm works using the following procedure:

1) Assign the minimum reserved throughput, $\tilde{r}_p^{rsv}$, and time slots, t_p^{rsv}, to all profiles
2) Find the amount of remaining resources from the total resources condition in (22)
3) Distribute the remaining resources proportionally between profiles in P_b, P_p (according to each profile's utility function weight)
4) Check whether or not the resources left over (from rounding when calculating the amount of slots to satisfy a specific throughput) can be expressed as a whole number of time slots, and, if yes, assign these resources to one of the strongest profiles in P_b, P_p
5) Calculate the revenue function F from (20)

As we can see, the worst case scenario complexity of this algorithm is polynomial and equals $\mathcal{O}(2(P+1))$, where $P = size\{P_b\} + size\{P_p\}$.

B. FRA Versus Optimal Solution Analysis

To evaluate the gap and the tradeoff between optimal and suboptimal solutions, we used the Optimization Toolbox of MATLAB on Lenovo ThinkPad machine with Intel® Core™ i7 processor, 2.20 GHz CPU and 8 GB RAM. The optimal solution was found using the branch and bound algorithm. The random values of weights w_p and v_p were assigned and the problem was solved by both algorithms for the same setting. Then the average was taken over 1000 solutions. The average value of the objective function and the average time to run both algorithms are plotted in Fig. 4 and Fig. 5 as functions of increasing number of profiles. In Fig. 4 we can observe a constant gap of about 10 Kb/s between the solution found by FRA and the optimal solution. On the other hand, Fig. 5 shows that the FRA algorithm runs about 200 times faster than the optimal solution. This is the desired feature for the WSN nodes with limited energy and memory resources.

```
1: **Step 1:**
2: **for** $\forall p$: $p \in P_b, P_p$ **do**
3:     $\tilde{r}_p = \tilde{r}_p^{rsv}, \forall p \in P_b$;   $\tilde{t}_p = t_p^{rsv}, \forall p \in P_p$;
4: **end for**
5: **Step 2:**
6: $D_r = NT_{cfp} - (\sum_{p \in P_b} \frac{\tilde{r}_p}{C} + T_{slot} \sum_{p \in P_p} \tilde{t}_p)$;   $C = \frac{B}{N(T_{fr}+T_{ifs})}$;
7: $D_t = \lfloor \frac{D_r}{t_{slot}} \rfloor$, number of available time slots;
8: **Step 3:**
9: Initialization of variables for tracking assigned extra resources
10: $S_t = 0$;   $S_r = 0$;
11: Calculation and allocation of extra resources
12: **for** $\forall p : p \in P_b$ **do**
13:     $\Delta t = \lfloor D_t \cdot w_p \rfloor$, number of extra slots per profile $p$ in $P_b$
14:     $r_p = \tilde{r}_p + C\Delta t$, assign the extra resources to every profile in $P_b$
15:     $S_r = S_r + \Delta t$, keep track of assigned resources for **Step 4**
16: **end for**
17: **for** $\forall p : p \in P_p$ **do**
18:     $\Delta t = \lfloor D_t \cdot v_p \rfloor$, number of extra slots per profile $p$ in $P_p$
19:     $t_p = \tilde{t}_p + \Delta t$, assign the extra resources to every profile in $P_p$
20:     $S_t = S_t + \Delta t$, keep track of assigned resources for **Step 4**
21: **end for**
22: **Step 4:**
23: $D_t^{rsd} = D_t - (S_t + S_r)$, check whether the residue is greater than or equal to 1 slot
24: **if** $D_t^{rsd} \geq 1$ **then**
25:     Assign $D_t^{rsd}$ to one of the strongest profiles $p \in P_b, P_p$:
26:     **if** $p \in P_b$ **then**
27:         $r_p = r_p + CD_t^{rsd}$;
28:     **end if**
29:     **if** $p \in P_p$ **then**
30:         $t_p = t_p + D_t^{rsd}$;
31:     **end if**
32: **end if**
33: **Step 5:**
34: $F = \sum_{p \in P_b} U_p(r_p) + CT_{slot} \sum_{p \in P_p} V_p(t_p)$, update the objective function
```

C. Round Robin and Proportionally Fair Algorithms

For comparison purposes, we have considered two additional algorithms: the Round Robin (RR) algorithm and the Proportionally Fair (PF) algorithm. RR-based scheduling is one of the most basic and popular schemes in computing

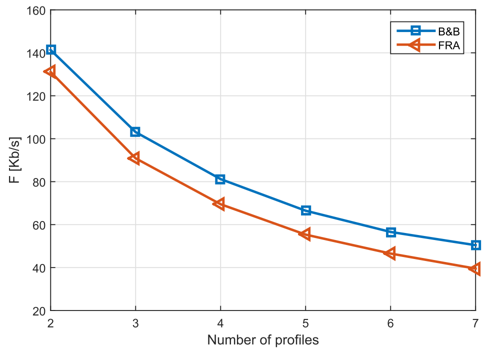

Fig. 4. Gap between optimal and suboptimal solutions.

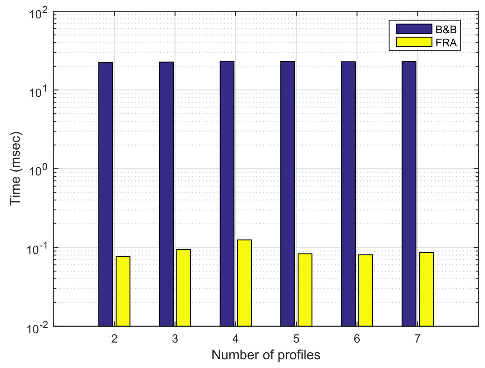

Fig. 5. Time to run the optimal and suboptimal solutions.

and networking due to its simplicity [73], for example see works [74]–[77]. It assigns equal portions of resources to all profiles in circular order with no priority. This scheduling is able to virtualize the network into profiles, but it is not flexible in terms of resource allocation.

PF scheduling technique is widely used in wireless mobile communications and is very well studied in the literature [78]–[80]. It exploits the multiuser diversity to maximize system throughput over independent temporal channel fluctuations, thus maintaining fairness among network users or flows. This algorithm keeps track of the average throughput $T_k[n]$ for each flow in an exponentially weighted window of length W [73]. In scheduling time n, the scheduler assigns the resources to the user k^* with the largest ratio $\frac{R_k[n]}{T_k[n]}$, where $R_k[n]$ is the user's channel rate, which indicates channel quality. Thus, each user is scheduled when its channel is good and at the same time the algorithm is fair in the long-term. It was shown in [73] and [81] that for a large window, W, this is also equivalent to maximizing logarithmic utility function $\sum_{k=1}^{K} log(T_k)$, where T_k is the long-term throughput average of a user k.

In WSNs with stationary nodes, however, channels do not exhibit severe fading and are usually modeled as line-of-sight (LoS). Therefore, to adapt this algorithm to our case, we utilized temporal queuing fluctuations in the network. The scheduler first assigns the minimum reserved resources to all profiles as in (21). Then, the extra resources in a scheduling time n are allocated to the profile p^* with the largest $\frac{St_p[n]}{T_p[n]}$, where $St_p[n]$ is current cumulative state, defined in (15) and $T_p[n]$ is the exponentially weighted low-pass filter, which is updated as follows:

$$T_p[n+1] = \begin{cases} \left(1-\frac{1}{W}\right) T_p[n] + \left(\frac{1}{W}\right) St_p[n], & p = p^*; \\ \left(1-\frac{1}{W}\right) T_p[n], & p \neq p^*. \end{cases} \quad (23)$$

This way, the algorithm gives the priority to profiles with high ratio of cumulative queue to long-term throughput, while preserving the minimum reserved resources allocation.

VI. Simulation

A. Discrete Event Simulator

A discrete stochastic simulation model can be used to simulate behavior of a network. DESs are used to model stochastic systems as they evolve in time by changing the state variable at discrete time intervals [72]. The high-level architecture of the simulator that we developed is shown in Fig. 6. It is built of four main stages (or units): Initialization, Central Scheduler, Internal Scheduler, and DES. Whereas the first two stages are generalized for all profiles, the last two stages are determined by the number of profiles and of sensors, respectively. Thus, if we have P profiles in total ($P = size\{P_b\} + size\{P_p\}$), then the simulator will have P internal schedulers (one per profile), and if we have an overall number of K sensors in the system, we will have K DES blocks (one per sensor).

1) Initialization Unit: During the initialization stage, the sensor objects are created and assigned to profiles, the MAC parameters of the frame are set, and the DES parameters are initialized.

2) Central Scheduler Unit: The central scheduler unit provides the decision on amount of resources to be allocated to each profile for one scheduling period (N frames). Type of decision depends on the algorithm we choose to use. After the Central Scheduling Unit obtains a decision for N frames, it allocates slots to each profile for the current frame and returns the allocation to each profile's Internal Scheduler. This allocation is updated each frame.

3) Internal Scheduler Units: There are P internal scheduler units, one for each profile, that perform flow control, *i.e.* manage individual throughput of each sensor by strategically distributing resources available within the same profile. Each frame, the Internal Schedulers receive a decision from the Central Scheduler about which slots corresponding profiles can use to transmit their packets. Furthermore, the Internal Schedulers divide available slots between sensors in the profile they belong to, based on each individual sensor's queuing size (or individual reported buffer status). Sensors with higher statuses are granted a larger portion of available resources. This is done by assigning a weight to each sensor in accordance with its status:

$$u_{(i,p)} = \frac{St_{(i,p)}}{St_p}, \quad \forall i = 1, \ldots, S_p, \quad \forall p \in P_b, P_p, \quad (24)$$

where $u_{(i,p)}$ is the weight of sensor i in profile p, $St_{(i,p)}$ is the reported status of sensor i in profile p, and St_p is the cumulative status of profile p. Thus, number of slots

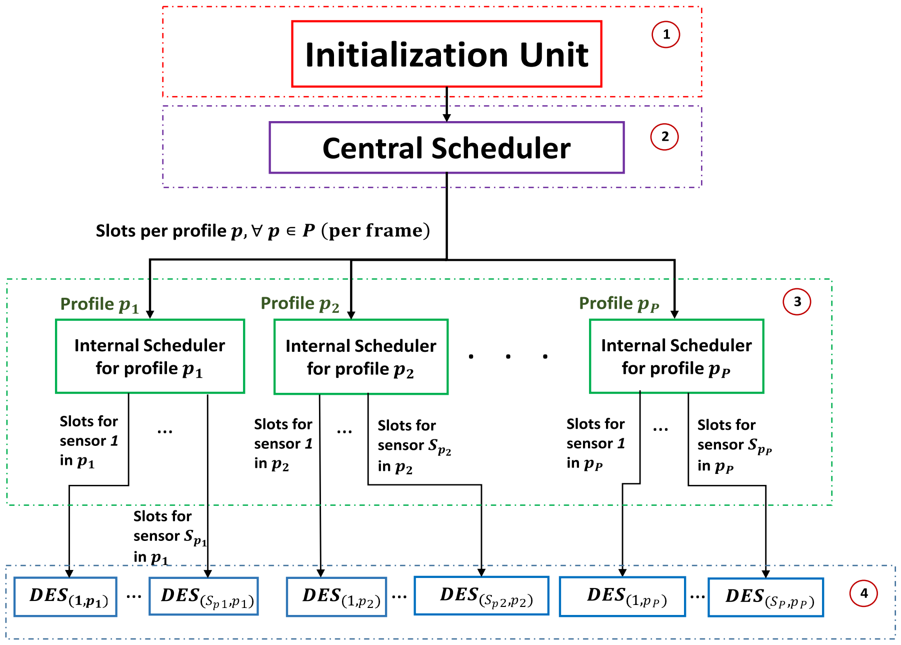
Initialization Unit
Central Scheduler
Slots per profile p, $\forall p \in P$ (per frame)
Profile p_1
Profile p_2
Profile p_P
Internal Scheduler for profile p_1
Internal Scheduler for profile p_2
Internal Scheduler for profile p_P
Slots for sensor 1 in p_1
Slots for sensor S_{p_1} in p_1
Slots for sensor 1 in p_2
Slots for sensor S_{p_2} in p_2
Slots for sensor 1 in p_P
Slots for sensor S_{p_P} in p_P
$DES_{(1,p_1)}$
$DES_{(S_{p1},p_1)}$
$DES_{(1,p_2)}$
$DES_{(S_{p2},p_2)}$
$DES_{(1,p_P)}$
$DES_{(S_P,p_P)}$
1
2
3
4

Fig. 6. DES simulator, general architecture.

assigned to each sensor in the current frame is $\lfloor n_p u_{(i,p)} \rfloor$. Note, $\sum_{i=1}^{S_p} u_{(i,p)} = 1, \forall p \in P_b, P_p$. If there is a residue due to rounding, it is divided in biased round-robin manner between sensors (*i.e.*, higher-status sensors are the first to receive extra slots).

4) DES Units: The DES units emulate the transmission of packets from sensors to the coordinator over the network and track statistics. Note, that in each slot there is only one DES unit running, corresponding to an active sensor. Each unit updates the state of its sensor and advances the Global Time (if there is an event happening during this slot).

B. FRA Versus RR and PF Algorithms

In this subsection, we validate by a simulation the effectiveness our resource management scheme with use of FRA scheduling algorithm. We also compare it to the performance of other popular algorithms. Finally, we aim to verify that the proposed virtualization technique fulfills the main virtualization characteristics that we stated in Section I: abstraction, flexibility and heterogeneity, manageability and isolation.

We set up an arbitrary scenario in which the network is abstracted into three bursty profiles, $p_1, p_2, p_3 \in P_b$, with five sensors each, and two periodical profiles, $p_4, p_5 \in P_p$ with two sensors each. Each of the bursty profiles is experiencing an event and produces large number of random packets, whereas the periodical profiles produce a small number of packets each second. The sensors in profiles p_1, p_2, and p_3 generate random packets according to a Poisson distribution, with parameters λ_1, λ_2, and λ_3 respectively (sensors within the same profile p have an equal rate of λ_p each). The aggregate packet arrival rate for P_b is:

$$\lambda_{P_b} = S_{p_1}\lambda_1 + S_{p_2}\lambda_2 + S_{p_3}\lambda_3 = 5 \cdot (\lambda_1 + \lambda_2 + \lambda_3). \quad (25)$$

Profiles in P_p produce periodical packets with rates ρ_1 and ρ_2 packets per second (p/s). Total rate for the group P_p is:

$$\rho_{P_p} = S_{p_4}\rho_1 + S_{p_5}\rho_2 = 2 \cdot (\rho_1 + \rho_2). \quad (26)$$

The superframe is defined by parameters $BO = SO = 4$ (see Fig. 1), with the beacon and the minimum CAP occupying only the first slot of each frame. We assume the beacon frame of length $L = 1016$ bits without security overhead and with short addressing in use. The scheduling period is $N = 10$ frames. The minimum reserved throughput and number of time slots are determined arbitrary and set to be at least the aggregate average amount of data generated across a profile or the required minimal number of time slots needed to transmit this data (note, that the reserved resources are per one scheduling period). The IFS is assumed to be zero for simplicity.

The scenario assumes that bursty profiles are experiencing an event for the entire duration of the simulation, and hence $I_{\bar{y}}^t = \{1, 1, 1\}$ (Section IV-C) throughout the simulation. Buffer sizes of all sensors are assumed to be infinite. Therefore, the buffer size value Q is abstract in this case and represents the reference point for the weight calculations rather than the actual physical size of the end device buffer. For the following simulations, it is set to 10 packets. All buffers exceeding this value are assumed to be in state 4. The length of the simulation is 30 seconds and statistics are collected over 1000 realizations. The simulation parameters are summarized in Table II.

The average (across sensors) delay probability density function (pdf) of packets in bursty profiles is shown in Fig. 7, where we can clearly see that the delay diverges with the RR algorithm. The reason for this is that the effective maximum throughput of the network is 186 Kb/s (due to the minimum CAP period and slots which are not the multiple

TABLE II
SIMULATION PARAMETERS. RR VS. FRA SCENARIO

Parameter	Value
P_p	profiles $p_1 - p_3$
P_b	profiles p_4, p_5
$S_{p_1} = S_{p_2} = S_{p_3}$	$5\ sensors$
$S_{p_4} = S_{p_5}$	$2\ sensors$
λ_1	$10\ p/s$ $(10.16\ Kb/s)$
λ_2	$14\ p/s$ $(14.24\ Kb/s)$
λ_3	$10\ p/s$ $(10.16\ Kb/s)$
λ_{P_b}	$170\ p/s$ $(172.72\ Kb/s)$
ρ_1	$1\ p/s$ $(1.016\ Kb/s)$
ρ_2	$2\ p/s$ $(2.032\ Kb/s)$
ρ_{P_p}	$6\ p/s$ $(6.096\ Kb/s)$
r_p^{rsv}	$\{51, 72, 51\}\ Kb/s$
t_p^{rsv}	$\{2, 3\}\ slots$
L	$1016\ bits$
$BO = SO$	4
T_{slot}	$480\ octets$ $(0.0154\ sec)$
T_{frame}	$7680\ octets$ $(0.246\ sec)$
T_{ifs}	$0\ sec$
N	$10\ frames$
Q	$10\ packets$ $(10.16\ Kb)$

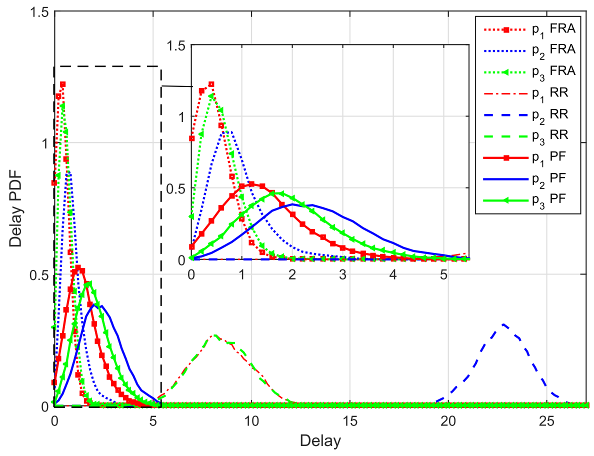

Fig. 7. The average delay pdf of bursty profiles $p_1 - p_3$. RR, PF and FRA algorithms.

of packet length). With equivalent resource allocation by the RR Algorithm between five profiles, the maximum throughput per profile is 37.2 Kb/s. Moreover, the profiles that have no packets to send are still allocated resources by this algorithm, which cannot be used by other profiles. Longer simulations will result in greater divergence, because the system is unstable, as the bursty profiles produce traffic at a rate higher than 37.2 Kb/s per profile, the maximum amount supported by the RR algorithm. The PF algorithm performs better in terms of delay, however, as seen from zoomed Fig. 7, the delay is greater than one scheduling period (2.46 sec). The reason is that while the PF algorithm favors the profiles with longest queues, the long-term fairness drives allocating extra slots to periodical profiles with smaller throughputs, that may not need extra resources. This results in less effectiveness and worse performance. The FRA algorithm, on the other hand, divides extra resources between the profiles proportionally to their need. Therefore, in this case the maximum delay is

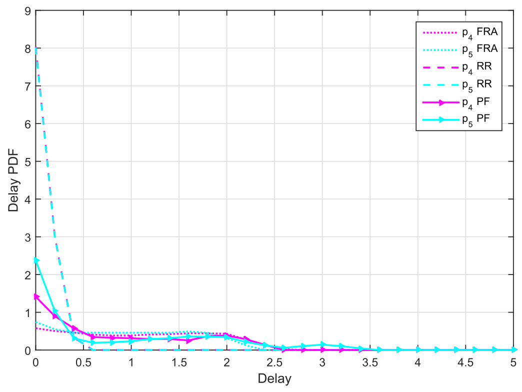

Fig. 8. The average delay pdf of periodical profiles $p_4 - p_5$. RR, PF and FRA algorithms.

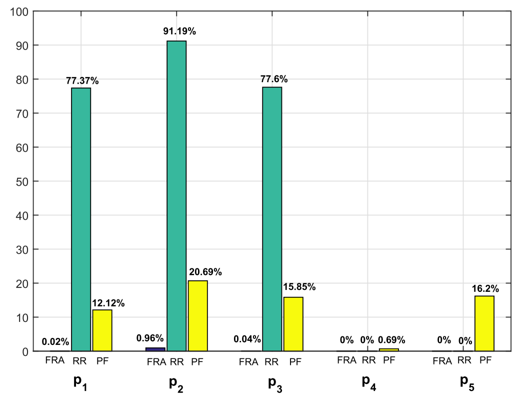

Fig. 9. Percentage of packets with ETE delay longer than one scheduling period in all profiles $p_1 - p_5$. RR, PF and FRA algorithms.

smaller than one scheduling period. Fig. 8 demonstrates, that in periodical profiles, the RR algorithm exhibits shorter delays than the FRA algorithm, where the maximum delay does not exceed the duration of one scheduling period. This is the case, because the number of slots reserved to profiles p_4 and p_5 is relatively small; therefore, when the FRA algorithm is in use, any sensor, that was not granted a slot in the current period, needs to wait till the next scheduling period, in which it will have a priority because of its waiting packets. Again, the delay is bound by one scheduling period time; reserving more slots would reduce the delay. At the same time, the RR algorithm divides all resources equally between all profiles, resulting in allocation of a surplus of resources to profiles with small amounts of generated data. PF algorithm behaves similarly to FRA with slightly larger number of packets having shorter delays. The percentage of packets with ETE delay longer than one scheduling period in all profiles is shown in Fig. 9. In this figure, we can observe that in bursty profiles under the same traffic conditions, number of packets with excessive delay is very high in RR and moderately high in PF cases respectively. In FRA case, it is below 1% in p_2 and almost zero in p_1 and p_3. In periodical profiles, p_4 and p_5, this percentage is almost zero for FRA and RR algorithms, because the produced periodical traffic is low, and is about 16% in p_5 with PF algorithm. All average achieved throughputs during

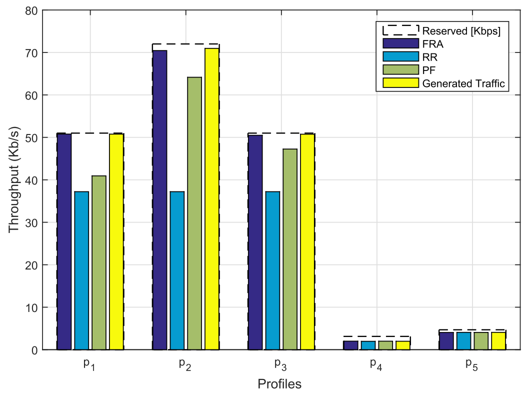

Fig. 10. The average achieved throughputs during one scheduling period vs. reserved resources and generated traffic. RR, PF and FRA algorithms.

one scheduling period are shown in Fig. 10. Here we can see the throughput with the RR algorithm at 37.2 Kb/s, less than the generated traffic during this time, for all bursty profiles. With the PF algorithm the throughput is higher, but still is lower than the traffic. This is due to ineffective management of the resources, as the PF algorithm, again, tries to be fair to periodical profiles that do not need extra resource allocation. The throughput with the FRA algorithm attains the amount of produced traffic. In periodical profiles, the achieved throughputs equal to generated traffic for all algorithms. Combined with predictable delays, this serves as a good example of resource management for WSN virtualization. This scenario illustrates the flexibility of the FRA algorithm in managing resources, which proves feasibility of the idea of network virtualization through MAC layer. Various heterogeneous applications using the network are abstracted into profiles and are supported by variable resource allocation in accordance with their requirements. As our simulations showed, each profile can be effectively segregated from other profiles with bounded ETE delay, thus fulfilling the isolation characteristic of the virtualization process.

C. Scenario With an Oversaturated Profile

In the second scenario, we evaluate the ability of our scheduling mechanism to redirect the available resources from profiles with little or no traffic to profiles with increased amounts of traffic. Namely, we test the event-detection scheme discussed in Section IV-C. This scheme is designed to increase the efficiency of resource utilization, resulting in shorter ETE delays, which is aimed at making the virtualization process less sensible by individual profiles.

Let us assume that there are three profiles in the network: two bursty profiles and one periodical. The first bursty profile is monitoring an event and thus generating many packets; the other bursty profile is quiet, generating only rare packets. The third profile is periodical and generates small amount of data. The simulation setting is given in Table III (if a parameter is not mentioned in this table, it has the same value as in Table II). We also assume that the majority of all resources is reserved to the bursty profiles: $r_p^{rsv} = \{102, 102\}\ Kb/s$, and the periodical profile reserves only one slot per scheduling period, *i.e.* $t_p^{rsv} = \{1\}$. It can be seen that the profile p_1

TABLE III
SIMULATION PARAMETERS. SCENARIO WITH AN OVERSATURATED PROFILE

Parameter	**Value**
P_b	profiles p_1, p_2
P_p	profile p_3
S_{p_1}	5 $sensors$
$S_{p_2} = S_{p_3}$	1 $sensor$
λ_1	30 (30.48 Kb/s)
λ_2	0.1 p/s (0.1016 Kb/s)
λ_{P_b}	150.1 p/s (152.4 Kb/s)
ρ_1	0.1 p/s (0.1016 Kb/s)
ρ_{P_p}	0.1 p/s (0.1016 Kb/s)
Q	10 $packets$ (10.16 Kb)
r_{min}	1.5622 Kb
r_p^{rsv}	$\{102, 102\}\ Kb/s$
t_p^{rsv}	$\{1\}\ slot$

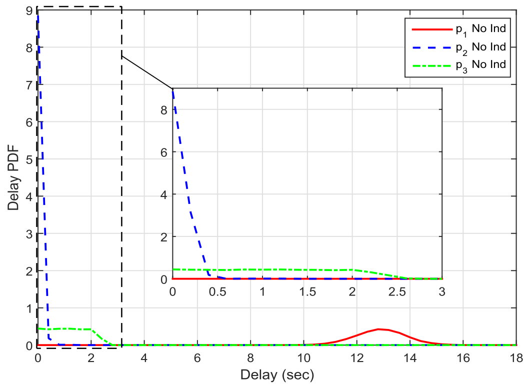

Fig. 11. The average delay pdf of profiles $p_1 - p_3$ with no event detection mechanism. Scenario with an oversaturated profile.

is oversaturated, *i.e.* produces more traffic than its reserved throughput (102 Kb/s reserved versus 152.4 Kb/s generated on average). We run the simulation twice: once without the event detection algorithm from Section IV-C and once with it. The minimum assigned throughput r_{min} is set to 1.56 Kb/s, which is equivalent to one assigned slot per period to ensure that a profile without an event is not completely cut from the network and can send any rare packet within one scheduling period. We expect that, in the first case, delay in profile p_1 will diverge due to a lack of assigned resources. The available resources are occupied, but not used, by profile p_2, resulting in continuously increasing delays in profile p_1. However, in the second case, the system should be able to detect "no events" in profile p_2 and reduce the reserved resources allocated to it, allowing the first profile to use them efficiently. The FRA algorithm is used for scheduling. The average delay (across sensors in each profile) pdf is shown in Fig. 11–12. As we expected, in the case with no indication mechanism, the delay of the oversaturated profile p_1 diverges, confirming the inefficiency of the schedule (Fig. 11). At the same time, the delay for profile p_1 with the event indication mechanism is much shorter than one scheduling period, see Fig. 12. Again, longer simulations will result in greater divergence because of instability of the system. Profiles p_2 and p_3 experience delays within one scheduling period, as they produce data within their reserved quotas. Using the event detection mechanism, all delays are within one scheduling interval, which means that

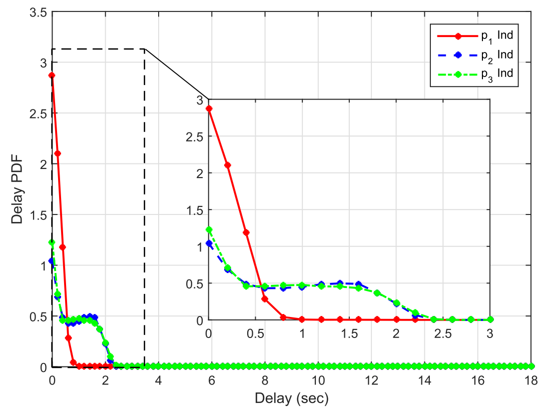

Fig. 12. The average delay pdf of profiles $p_1 - p_3$ with event detection mechanism. Scenario with an oversaturated profile.

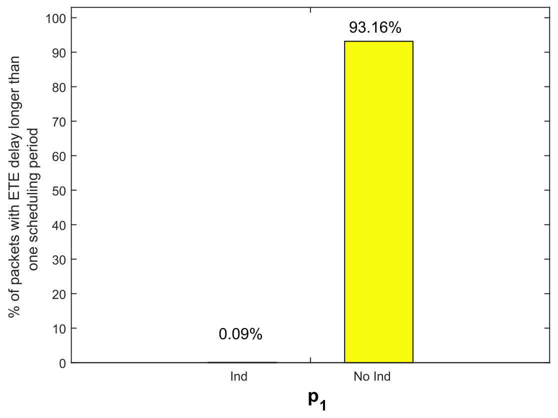

Fig. 13. Percentage of packets with ETE delay longer than one scheduling period in bursty profile p_1. Scenario with an oversaturated profile.

the system successfully detects the absence of an event in p_2 and reduces its reserved throughput to the preset minimum, allowing the busy profile p_1 to use the available resources.

The percentage of packets with ETE delay longer than one scheduling period in the profile with high traffic, p_1, is shown in Fig. 13. When the event detection mechanism is in use, this percentage is nearly zero due to effective sharing of the resources, whereas without the event detection almost all packets reach the coordinator with excessive delay. In low traffic profiles, p_2 and p_3, this percentage is zero in both cases, and hence is not shown in the figure.

The attained average throughputs during one scheduling period are shown in Fig. 14, where we can observe that the throughput equals the minimum amount of reserved resources for profile p_1 in the case with no event detection and achieves its traffic amount in the case with event detection. This shows that profile p_1 was granted about 30% extra resources not being used by profile p_2 due to the lack of an event. Should an event occur in profile p_2, the system would detect it through the buffer status feedback, setting its indicator function to 1 and assigning the initially reserved $r_{p_2}^{rsv}$ resources. We also can see that even though profile p_1 reserved fewer resources than the traffic it produced, using the event detection algorithm it was able to achieve greater throughputs while allowing the quiet profiles to send their data in timely manner (within one scheduling period).

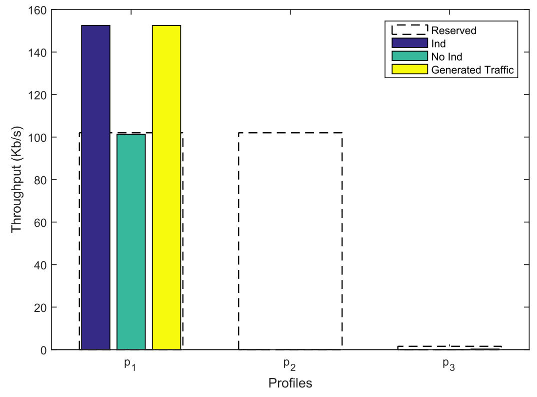

Fig. 14. The achieved average profiles' throughputs per one scheduling period vs. reserved resources and generated traffic. Scenario with an oversaturated profile.

VII. Conclusion

In this paper, we considered the approach to bridge the gap of WSNs virtualization on lower layers of communication stack. Our main goal was to realize the virtualization concept through abstraction of physical resources into logical units and managing their allocation to different profiles, supporting various applications. In order to satisfy heterogeneity and flexibility requirements, we assumed that all profiles can be categorized into two general groups: profiles producing periodical data and profiles producing bursty data. Each group has different resource requirements.

We solved the resource-sharing problem in a linear optimization framework, making sure that each individual profile's requirements were satisfied. Furthermore, we developed a two-level scheduling mechanism that comprises the top-level main scheduler, which assigns resources to each profile, and a sequence of lower-level internal schedulers, that control flows inside each profile by assigning resources to sensors within the profile. We also proposed a reduced-complexity heuristic algorithm, which is used by the general scheduler to obtain a decision for each scheduling period. Moreover, an event detection mechanism was developed to increase the efficiency of resource-sharing and the flexibility of the scheduling.

The simulation included two scenarios to validate the proposed virtualization scheme and its ability to isolate profiles as well as its effectiveness. The results show that the scheduler, based on the proposed algorithm, manages resource allocation effectively and keeps the profiles independent of each other, thus proving the feasibility of our solution for virtualization on the MAC layer. Moreover, the event detection mechanism showed good performance in a case, in which one of the profiles was oversaturated. The mechanism allowed this profile to "borrow" resources not being used by another profile, which prevented the divergence of end-to-end delay, increasing the effectiveness of resource utilization in the underlying virtualization process and making it less sensible by individual profiles.

References

[1] M. M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. Huh, "A survey on virtualization of wireless sensor networks," *IEEE Sensors J.*, vol. 12, no. 2, pp. 2175–2207, Feb. 2012.

[2] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surveys Tut.*, vol. 18, no. 1, pp. 553–576, 1st Quart., 2015.

[3] C. M. De Farias *et al.*, "A systematic review of shared sensor networks," *ACM Comput. Surv.*, vol. 48, no. 4, May 2016, art. no. 51.

[4] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010.

[5] P. Levis *et al.*, *Ambient Intelligence*. Berlin, Germany: Springer, 2005.

[6] Underwriters Laboratories (UL) Inc. Standards, (2015). [Online]. Available: http://ul.com/

[7] National Fire Protection Assosiation (NFPA), (2016). [Online]. Available: http://www.nfpa.org

[8] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: Early architecture and research perspectives," *IEEE Netw.*, vol. 29, no. 3, pp. 104–112, May/Jun. 2015.

[9] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in *Proc. 6th Joint IFIP Wireless Mobile Netw. Conf. (WMNC)*, Apr. 2013, pp. 1–4.

[10] A. Muneeb and K. Langendoen, "A case of peer-to-peer network overlays in sensor networks,"in *Proc. Int. Workshop Wireless Sensor Netw. Archit. (WWSNA)*, 2007, pp. 56–61.

[11] A.-B. Al-Mamou and H. Labiod, "ScatterPastry: An overlay routing using a DHT over wireless sensor networks," in *Proc. Int. Conf. Intell. Pervasive Comput. (IPC)*, Oct. 2007, pp. 274–279.

[12] E. De Poorter, E. Troubleyn, I. Moerman, and P. Demeester, "IDRA: A flexible system architecture for next generation wireless sensor networks," *Wireless Netw.*, vol. 17, no. 6, pp. 1423–1440, Aug. 2011.

[13] G. Fersi, W. Louati, and M. B. Jemaa, "Distributed hash table-based routing and data management in wireless sensor networks: A survey," *Wireless Netw.*, vol. 19, no. 2, pp. 219–236, Feb. 2013.

[14] I. Ishaq, J. Hoebeke, I. Moerman, and P. Demeester, "Internet of Things virtual networks: Bringing network virtualization to resource-constrained devices," in *Proc. Int. Conf. Green Comput. Commun. (GreenCom)*, Nov. 2012, pp. 293–300.

[15] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comp. Syst.*, vol. 18, no. 3, pp. 263–297, Aug. 2000.

[16] I. Khan, F. Z. Errounda, S. Yangui, R. Glitho, and N. Crespi, "Getting virtualized wireless sensor networks' IaaS ready for PaaS," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, 2015, pp. 224–229.

[17] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, "SenShare: Transforming sensor networks into multi-application sensing infrastructures," in *Proc. 9th Eur. Conf. Wireless Sensor Netw. (EWSN)*, Feb. 2012, pp. 65–81.

[18] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Std 802.15.4, 2011.

[19] H. M. N. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "Cluster tree based self organization of virtual sensor networks," in *Proc. IEEE GLOBECOM Workshops*, Nov. 2008, pp. 1–6.

[20] H. M. N. D. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "A top-down clustering and cluster-tree-based routing scheme for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2011, pp. 1–17, May 2011, Art. no. 940751.

[21] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc. INFOCOM*, Jun. 2002, pp. 1567–1576.

[22] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, Nov. 2003, pp. 171–180.

[23] H. Pham and S. Jha, "An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC)," in *Proc. IEEE Int. Conf. Mobile Ad-hoc Sensor Syst.*, Oct. 2004, pp. 558–560.

[24] M. Zareei, A. K. M. M. Islam, A. Zeb, S. Baharun, and S. Komaki, "Mobility-aware timeout medium access control protocol for wireless sensor networks," *Int. J. Electron. Commun.*, vol. 68, no. 10, pp. 1000–1006, Oct. 2014.

[25] S. Ghosh, P. Veeraraghavan, S. Singh, and L. Zhang, "Performance of a wireless sensor network MAC protocol with a global sleep schedule," *Int. J. Multimedia Ubiquitous Eng.*, vol. 4, no. 2, pp. 99–114, 2009.

[26] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. 18th Int. Parallel Distrib. Process. Symp.*, Apr. 2004, p. 224.

[27] H. Malik, E. Shakshuki, and T. Sheltami, *Agent Based Analytical Model for Energy Consumption Among Border Nodes in Wireless Sensor Networks*. Berlin, Germany: Springer-Verlag, 2008.

[28] S. Li, T. Zhang, and D. Qian, "Deployment oriented role destined integrated protocol for event-driven wireless sensor networks," in *Proc. IEEE 14th Int. Conf. Comput. Sci. Eng. (CSE)*, Aug. 2011, pp. 387–391.

[29] E. Kim, M. Kim, S. Youm, S. Choi, and C.-H. Kang, "Multi-level service differentiation scheme for the IEEE 802.15.4 sensor networks," in *Proc. Int. Conf. Embedded Ubiquitous Comput. (EUC)*, 2005, pp. 693–703.

[30] E.-J. Kim, M. Kim, S.-K. Youm, S. Choi, and C.-H. Kang, "Priority-based service differentiation scheme for IEEE 802.15.4 sensor networks," *Int. J. Electron. Commun.*, vol. 61, no. 2, pp. 69–81, Feb. 2007.

[31] C. Na and Y. Yang, "MRSD: Multirate-based service differentiation for the IEEE 802.15.4 wireless sensor network," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2013, pp. 1–6.

[32] G. Noubir, W. Qian, B. Thapa, and Y. Wang, "Experimentation-oriented platform for development and evaluation of MANET cross-layer protocols," *Ad Hoc Netw.*, vol. 7, no. 2, pp. 443–459, Mar. 2008.

[33] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A wireless sensor MAC protocol for bursty data traffic," in *Proc. 18th Annu. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2007, pp. 1–5.

[34] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tut.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[35] L. Xia *et al.*, "Virtual WiFi: Bring virtualization from wired to wireless," in *Proc. 7th ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ. (VEE)*, Mar. 2011, pp. 181–192.

[36] T. Nagai and H. Shigeno, "A framework of AP aggregation using virtualization for high density WLANs," in *Proc. 3rd Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Nov. 2011, pp. 350–355.

[37] P. Lv, Z. Cai, J. Xu, and M. Xu, "Multicast service-oriented virtual network embedding in wireless mesh networks," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 375–377, Mar. 2012.

[38] G. Smith, A. Chaturvedi, A. Mishra, and S. Banerjee, "Wireless virtualization on commodity 802.11 hardware," in *Proc. 2nd ACM Int. Workshop Wireless Netw. Testbeds, Experim. Eval. Characterization (WinTECH)*, Sep. 2007, pp. 75–82.

[39] S. Perez, J. M. Cabero, and E. Miguel, "Virtualization of the wireless medium: A simulation-based study," in *Proc. IEEE 69th Veh. Technol. Conf. (VTC)*, Apr. 2009, pp. 1–5.

[40] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, "SplitAP: Leveraging wireless network virtualization for flexible sharing of WLANs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–6.

[41] A. Leivadeas, C. Papagianni, E. Paraskevas, G. Androulidakis, and S. Papavassiliou, "An Architecture for virtual network embedding in wireless systems," in *Proc. 1st Int. Symp. Netw. Cloud Comput. Appl. (NCCA)*, Nov. 2011, pp. 62–68.

[42] T. Magedanz, A. Gavras, H. T. Nguyen, and J. S. Chase, *Testbeds and Research Infrastructures*. Berlin, Germany: Springer-Verlag, May 2010.

[43] S. Singhal, G. Hadjichristofi, I. Seskar, and D. Raychaudhuri, "Evaluation of UML based wireless network virtualization," in *Proc. IEEE Next Generation Internet Netw.*, Apr. 28–30, 2008, pp. 223–230.

[44] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[45] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "CellSlice: Cellular wireless resource slicing for active RAN sharing," in *Proc. 5th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2013, pp. 1–10.

[46] X. Lu, K. Yang, Y. Liu, D. Zhou, and S. Liu, "An elastic resource allocation algorithm enabling wireless network virtualization," *Wireless Commun. Mobile Comput.*, vol. 15, no. 2, pp. 295–308, Feb. 2015.

[47] X. Lu, K. Yang, and H. Zhang, "An elastic sub-carrier and power allocation algorithm enabling wireless network virtualization," *Wireless Pers. Commun.*, vol. 75, no. 4, pp. 1827–1849, Apr. 2013.

[48] Y. Zaki, L. Zhao, and C. G. A. Timm-Giel, "LTE mobile network virtualization: Exploiting multiplexing and multi-user diversity gain," *Mobile Netw. Appl.*, vol. 16, no. 4, pp. 424–432, Aug. 2011.

[49] Y. Zaki, "Future mobile communications," *LTE Optimization and Mobile Network Virtualization*. Wiesbaden, Germany: Springer-Verlag, 2013.

[50] L. Zhao, M. Li, Y. Zaki, A. Timm-Giel, and C. Görg, "LTE virtualization: From theoretical gain to practical solution," in *Proc. 23rd Int. Teletraffic Congr. (ITC)*, Sep. 2011, pp. 71–78.

[51] M. Li *et al.*, "Investigation of network virtualization and load balancing techniques in LTE networks ," in *Proc. IEEE VTC*, May 2012, pp. 1–5.

[52] M. Kalil, A. Shami, and A. Al-Dweik, "QoS-aware power-efficient scheduler for LTE uplink," *IEEE Trans. Mobile Comput.*, vol. 14, no. 8, pp. 1672–1685, Aug. 2015.

[53] F. Cuomo, S. D. Luna, P. Todorova, and T. Suihko, "Topology formation in ieee 802.15.4: Cluster-tree characterization," in *Proc. 6th Ann. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2008, pp. 276–281.

[54] S. Tennina *et al.*, *IEEE 802.15.4 and ZigBee as Enabling Technologies for Low-Power Wireless Systems with Quality of Service Constraints*. Springer, 2013.

[55] S. Nguyen-Xuan, S. Oh, and A. Sunshin, "EE-MAC: Energy efficient-medium access control for periodic applications in border surveillance wireless sensor networks," in *Proc. 16th Int. Conf. Adv. Commun. Technol.*, Feb. 2014, pp. 693–697.

[56] S. Wijetunge, U. Gunawardana, and R. Liyanapathirana, "Data transmission reliability of IEEE 802.15.4 based wireless sensor networks with synchronised periodic data," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Jun. 2012, pp. 619–624.

[57] D. D. Guglielmo, G. Anastasi, and M. Conti, "A localized de-synchronization algorithm for periodic data reporting in IEEE 802.15.4 WSNs," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2012, pp. 000605–000610.

[58] B. Mohamed, R. Houria, and E. Tahar, "A comprehensive performance study of the contention access period of the slotted IEEE 802.15.4," in *Proc. Int. Conf. Adv. Wireless Inf. Commun. Technol. (AWICT)*, 2015, pp. 306–311.

[59] N.-C. Liang, P. Chen, T. Sun, G. Yang, L.-J. Chen, and M. Gerla, "Impact of node heterogeneity in zigbee mesh network routing," in *Proc. Int. Conf. Syst., Man*, 2006, pp. 187–191.

[60] A. M. Mohamed and A. F. Agamy, "Performance behaviour of WSN with bursty traffic," in *Proc. 8th Int. Conf. Informat. Syst. (INFOS)*, 2012, pp. NW-40–NW-45.

[61] S. Ray, I. Demirkol, and W. Heinzelman, "Supporting bursty traffic in wireless sensor networks through a distributed advertisement-based TDMA protocol (ATMA)," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 959–974, 2013.

[62] G. Z. Papadopoulos, A. Gallais, T. Noel, V. Kotsiou, and P. Chatzimisios, "Enhancing ContikiMAC for bursty traffic in mobile sensor networks," in *Proc. IEEE Sensors*, Nov. 2014, pp. 257–260.

[63] A. Koubâa, M. Alves, E. Tovar, and A. Cunba, "An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: Theory and practice," *Real-Time Syst.*, vol. 39, no. 1, pp. 169–204, 2007.

[64] P. Jurcik, A. Koubâa, M. Alves, E. Tovar, and Z. Hanzálek, "A simulation model for the IEEE 802.15.4 protocol: Delay/throughput evaluation of the GTS mechanism," in *Proc. 15th Int. Symp. Modeling, Anal., Simulution Comput. Telecommun. Syst. (MASCOTS)*, Oct. 2007, pp. 109–116.

[65] P. Jurcik, R. Severino, A. Koubâa, M. Alves, and E. Tovar, "Real-Time communications over cluster-tree sensor networks with mobile sink behaviour," in *Proc. 14th IEEE Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, Oct. 2008, pp. 401–412.

[66] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. New York, NY, USA: Springer, 2001.

[67] I. Matta and A. U. Shankar, "Type-of-service routing in datagram delivery systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1411–1425, Oct. 1995.

[68] J. He, R. Zhang-shen, Y. Li, C. Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically adaptive virtual networks for a customized Internet," in *Proc. ACM CoNEXT Conf. (CoNEXT)*, 2008, pp. 1–12.

[69] T. T. Pham, H. H. Nguyen, and H. D. Tuan, "Relay assignment for max-min capacity in cooperative wireless networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2387–2394, Jun. 2012.

[70] J. Tang, B. Mumey, K. Zhubayev, and R. S. Wolff, "Leveraging cooperative, channel and multiuser diversities for efficient resource allocation in wireless relay networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1789–1797, Oct. 2012.

[71] N. M. Do, C.-H. Hsu, and N. Venkatasubramanian, "Video dissemination over hybrid cellular and Ad Hoc networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 274–286, Feb. 2014.

[72] W. L. Winston, *Operations Research: Applications and Algorithms*, 4th ed. Boston, MA, USA: Cengage Learning, 2004.

[73] D. Tse, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[74] S. S. Kanhere, H. Sethu, and A. B. Parekh, "Fair and efficient packet scheduling using Elastic Round Robin," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 3, pp. 324–336, Mar. 2002.

[75] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 7, pp. 1024–1039, Sep. 1991.

[76] Y. Xu, H. Su, Y.-J. Pan, Z.-G. Wu, and W. Xu, "Stability analysis of networked control systems with round-robin scheduling and packet dropouts," *J. Franklin Inst.*, vol. 350, p. 2013–2027, Oct. 2013.

[77] K. Liu, E. Fridman, and L. Hetel, "Stability and $L_2 L_2$-gain analysis of networked control systems under round-robin scheduling: A time-delay approach," *Syst. Control Lett.*, vol. 61, no. 5, pp. 666–675, 2012.

[78] E. Liu and K. K. Leung, "Proportional fair scheduling: Analytical insight under rayleigh fading environment," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2008, pp. 1883–1888.

[79] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, 1997.

[80] S. B. Lee, I. Pefkianakis, A. Meyerson, and S. Xu, "Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink," in *Proc. INFOCOM*, Apr. 2009, pp. 2611–2615.

[81] H. Kushner and P. Whiting, "Asymptotic properties of proportional-fair sharing algorithms: Extensions of the algorithm," in *Proc. 40th Allerton Conf. Commun., Control Comput.*, 2002, pp. 1532–1541.

Elena Uchiteleva (S'13) received the B.Sc. degree in electrical engineering from the Technion - Israel Institute of Technology, Haifa, Israel, in 2010, and the M.E.Sc. degree in electrical and computer engineering from Western University, London, ON, Canada, in 2013. In 2014, she joined the Optimized Computing and Communications Lab, Western University, where she is currently pursuing the Ph.D. degree in electrical and computer engineering. Her research interests include wireless communications, wireless sensor networks, data networks, and virtualization of wireless networks. She is also the Regional Coordinator of the IEEE Canada Women in Engineering (WIE) Affinity Group and the Past Chair of the IEEE WIE London Section Group.

Abdallah Shami (SM'09) received the B.E. degree in electrical and computer engineering from Lebanese University, Beirut, Lebanon, in 1997, and the Ph.D. degree in electrical engineering from the Graduate School and University Center, City University of New York, New York, NY, USA, in 2002. Since 2004, he has been with Western University, Canada, where he is currently a Professor with the Department of Electrical and Computer Engineering. His current research interests are in the area of network-based cloud computing and wireless/data networking. He has chaired key symposia of the IEEE GLOBECOM, the IEEE ICC, the IEEE ICNC, and the ICCIT. He is the Chair of the IEEE Communications Society Technical Committee on Communications Software. He is currently an Associate Editor of the IEEE *Communications Surveys and Tutorials*, the *IET Communications Journal*, and the *Journal of Wireless Communications and Mobile Computing* (Wiley).

Ahmed Refaey (SM'15) received the B.Sc. and M.Sc. degrees from Alexandria University, Egypt, in 2003 and 2005, respectively, and the Ph.D. degree from Laval University, Quebec, Canada, in 2011. He is an Assistant Professor at Manhattan College as well as an Adjunct Research Professor at The Western University. Previously, his positions included: a Senior Systems Architect, R&D group, Mircom Group of Companies from 2012 to 2016 and Professional Researcher at the LRTS Lab, Laval University, in the field of wireless communications from 2007 to 2011. Prior to joining Laval University, he was a System/Core Network Engineer leading a team of junior engineers and technicians in the telecom field in the three large companies of Fujitsu, Vodafone and Alcatel-Lucent.