sensors

Review

Overlay Virtualized Wireless Sensor Networks for Application in Industrial Internet of Things: A Review

Malvin Nkomo [1], Gerhard P. Hancke [1,2,*], Adnan M. Abu-Mahfouz [3,4], Saurabh Sinha [1,5] and Adeiza. J. Onumanyi [3]

[1] Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park, Johannesburg 2006, South Africa; malvin.m.n@ieee.org (M.N.); ssinha@uj.ac.za (S.S.)
[2] Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong, China
[3] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa; A.AbuMahfouz@ieee.org (A.M.A.-M.); adeiza1@yahoo.com (A.J.O.)
[4] Modelling and Digital Science, Council for Scientific and Industrial Research, Pretoria 0001, South Africa
[5] Office of the Deputy Vice-Chancellor: Research and Internationalisation, University of Johannesburg, Johannesburg 2006, South Africa
* Correspondence: gp.hancke@cityu.edu.hk; Tel.: +852-3442-9341

Received: 13 July 2018; Accepted: 8 August 2018; Published: 23 September 2018

Abstract: In recent times, Wireless Sensor Networks (WSNs) are broadly applied in the Industrial Internet of Things (IIoT) in order to enhance the productivity and efficiency of existing and prospective manufacturing industries. In particular, an area of interest that concerns the use of WSNs in IIoT is the concept of sensor network virtualization and overlay networks. Both network virtualization and overlay networks are considered contemporary because they provide the capacity to create services and applications at the edge of existing virtual networks without changing the underlying infrastructure. This capability makes both network virtualization and overlay network services highly beneficial, particularly for the dynamic needs of IIoT based applications such as in smart industry applications, smart city, and smart home applications. Consequently, the study of both WSN virtualization and overlay networks has become highly patronized in the literature, leading to the growth and maturity of the research area. In line with this growth, this paper provides a review of the development made thus far concerning virtualized sensor networks, with emphasis on the application of overlay networks in IIoT. Principally, the process of virtualization in WSN is discussed along with its importance in IIoT applications. Different challenges in WSN are also presented along with possible solutions given by the use of virtualized WSNs. Further details are also presented concerning the use of overlay networks as the next step to supporting virtualization in shared sensor networks. Our discussion closes with an exposition of the existing challenges in the use of virtualized WSN for IIoT applications. In general, because overlay networks will be contributory to the future development and advancement of smart industrial and smart city applications, this review may be considered by researchers as a reference point for those particularly interested in the study of this growing field.

Keywords: Internet of Things; WSN virtualization; overlay WSN; Industrial Internet-of-Things (IIoT)

1. Introduction

The Industrial Internet of Things (IIoT) is a recent and growing research area concerning the application of the Internet of Things (IoT) and related technologies such as Wireless Sensor Networks

Sensors **2018**, *18*, 3215; doi:10.3390/s18103215 www.mdpi.com/journal/sensors

(WSN) in order to improve the productivity/performance of several industrial processes and systems [1]. The IIoT leverages the IoT paradigm to improve connectivity, efficiency, scalability and cost savings of different manufacturing industries and organizations. The IIoT has been recently applied in several important domains including in smart industry applications [2–4], smart manufacturing [5] and smart city applications [6–8].

Despite being contemporary, the success of the IIoT strongly depends on the advances being made in the study of WSNs [9]. This dependence exists because most IIoT deployments are mostly based on the use of sensors, which are components of the WSNs. In this regard, several existing research challenges regarding WSNs are actively under consideration in the literature. One problem is the growing heterogeneity being created by the increased development and deployment of different sensor nodes in WSNs, which prevent the co-existence of different WSN nodes on a shared physical infrastructure. This heterogeneity introduces difficulties including the interoperability problem across different administrative domains, slow deployment rates of WSNs, conflicting goals and economic interests of different WSNs node vendors, and the increasing cost of WSN deployments.

In order to address these challenges, the concept of virtualization in sensor networks has been introduced. Sensor network virtualization is aimed at providing flexibility in the deployment of WSNs, providing cost-effective solutions, improving seamless interoperability and enhancing security and management facilities. In technical terms, a virtualized wireless sensor network (VWSN) is formed by delivering logical connectivity to a subset of collaborating nodes in order to accomplish a specific task or application at a given time. These nodes are grouped based on the physical condition that these nodes may be tracking or monitoring per time. Based on the success of VWSNs, the recent concept of overlay VWSN (OVWSN) has consequently gained similar popularity, particularly for enhancing the performance of WSNs.

The concept of OVWSN, especially within the existing WSN infrastructure, will increase the reuse of different sensor infrastructures. This is possible because virtualization decouples the network infrastructure from the application services and replaces current direct connections with virtual/logical links. These logical links require the application services to choose the optimal node(s) to use in a particular situation, as opposed to the fixed allocation of nodes to a specific application task(s). The physical resources are thus allocated on-the-fly and on-demand using possible dynamic resource allocation techniques. This concept ensures that resources are efficiently utilized and shared between different application tasks.

Current and future deployment of sensor networks will require efficient and powerful devices to be used as sensor nodes. Thus, feature-rich and resource-abundant devices should be available to create a platform for VWSN, such that sensor nodes can share resources among a variety of applications to avoid unnecessary and redundant deployment of nodes. Virtualization allows users (applications and services) to use resources dedicated to them [10,11]. By complementing the concept of virtualization, overlaying further allows the creation of new services at the edge of the existing infrastructure with little or no change to the underlying hardware. A combination of virtualization and overlay services will enable robust, scalable and resilient WSNs. Besides, virtualization addresses the gap of domain-specificity in wireless sensor deployment. Thus, as WSNs become more pervasive and application areas continue to grow exponentially, the need to build OVWSNs over existing infrastructures will undoubtedly improve the implementation of resilient networks that will meet the future demands of WSN.

Thus, following the growth of OVWSN [12,13], including the increased rate of published materials in this regard, it has become necessary to provide an overview of the progress made thus far in this research area. Therefore, this paper provides a detailed review of OVWSNs, beginning with the motivation behind the need for virtualization in WSN. Then, different existing challenges are provided with an emphasis on virtualization as a possible solution. This is followed by an overview of VWSNs, its building blocks and the different types of virtualization platforms and how they comparatively perform. An exposition into OVWSN is provided based on different areas such as its topology, routing, media access and its service delivery. The state-of-the-art with regards to OVWSN is discussed. After that, few design requirements towards improving OVWSN are considered, followed by the presentation of some existing research challenges for future works. By the above, the following contributions are made in this paper:

- A comprehensive comparison of the different techniques used for implementing VWSN is provided under different categories including the various operating systems used, the middleware, and also the virtual machines used.
- The concept of OVWSN is thoroughly discussed as a potential solution to some of the problems encountered in smart industrial applications.
- Peer-to-peer topologies are well contrasted with ring topologies mainly as possible enablers for OVWSN.
- Some research challenges and potential solutions in OVWSN are also discussed for easy comprehension by budding researchers in the OVWSN study area.
- A conceptualization of the design requirements for future OVWSN has also been provided and discussed.

The rest of this paper is presented as follows: Section 2 expounds on the concepts of virtualization in WSN, and an application scenario of a smart industrial application [5] is presented. Section 3 discusses the importance of shared sensors as fundamental to the emergence of virtualized WSN. Sections 4 details the core of the article on overlay virtual wireless sensor networks, while Section 5 discusses the design requirements for overlay virtual WSNs. Section 6 presents the open research areas available, and Section 7 concludes the article.

2. Virtualization in Wireless Sensor Networks

Technically, virtualization is set up by decoupling the application services from the underlying physical network so that the application services are not directly connected to their corresponding network elements, but are rather connected via logical/virtual linkages spread across the entire network. In most traditional schemes, the middleware and virtual machine approaches have been mostly used in WSNs to achieve virtualization at the network and node level, respectively. Virtualized WSN (VWSN) are typically applied in some recent application areas such as in smart industrial applications [14]. Due to the scale of these applications areas, for example in smart industry applications, it has become pertinent for resource sharing to be the most likely approach for deploying WSN services. Sharing of resources with less individual ownership has become the global trend [15,16], thus giving way to more resources being efficiently managed by on-demand platforms. Such shared resource management schemes have been found to be successful, for example with the Uber scheme [17], and with the hospitality scheme using Airbnb [18]. These schemes, some of which depend on sensor network infrastructures, are normally shared by a variety of users and are accessed using different platforms, for example via specialized operating systems, virtual machines, middleware and other sharing platforms [4].

In most application areas, the ubiquity of the deployed sensor nodes places a great demand on the available shared resources. Thus, this growing demand has made the concept of virtualization of these services a highly beneficial technology for enhancing resource sharing among different stakeholders. An application area that may greatly benefit from VWSN is the smart industry application area. An example of the smart industrial concept is shown in Figure 1. It is seen from Figure 1 that by introducing a heterogeneous sensing layer, the network virtualization layer is thus able to provide fewer nodes for deployment in different sensing applications while meeting all the necessary sensing requirements.

In developing resource sharing frameworks, lightweight architectures are normally used to initiate VWSN for deployment in smart industrial applications. Essentially, these resource sharing frameworks normally include a sensing layer, a virtualization layer, cloud services and an end-user access. The sensing layer typically consists of a number of different sensing scenarios under different application domain areas [19]. The concept of VWSN in smart IIoT presented in Figure 1 aims to reduce the number of nodes required for sensing deployment. As such, the gateways are equipped with virtualization layers, which enable several application tasks to run on the gateway instead of on the sensor nodes. Thus, a great deal of load balancing is provided by the resource rich gateway for the sensor nodes.

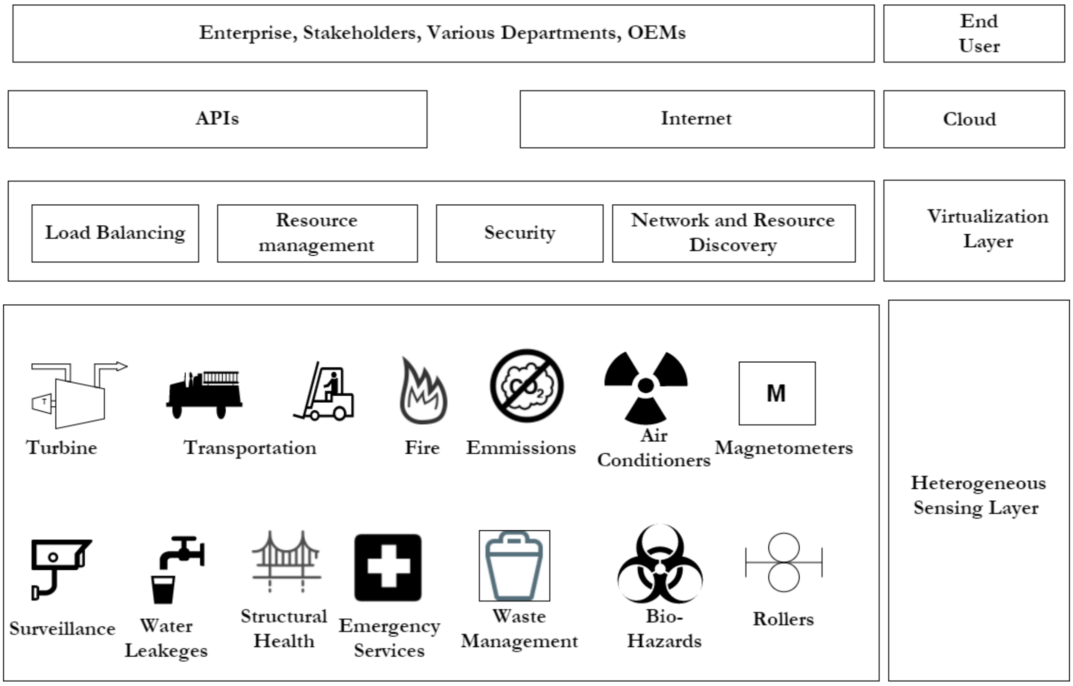

Figure 1. A smart IIoT concept utilizing resource sharing.

The virtualization layer consists of resource rich gateways for the WSN. These gateways consist of a various communication standards based on IEEE 802.15.4 [20], 802.11 [21], 802.3 MAC and sub-GHz open and proprietary standards, which include 6LoWPAN [22], ZigBee [23], Bluetooth Low Energy [24], Wi-Fi, Thread [25], and ZWave [26]. The gateway is also responsible for running multiple instances of the various sensing applications on the node, while giving the illusion that only one application is accessing the gateway at a time. The gateway (or sink) is often able to provide load balancing, resource management, network discovery and able to offer a secure platform for shared infrastructure. Enterprises, governments, municipalities and civilians can thus share the data obtained from the different WSNs. Furthermore, in order to perform a particular sensing function, the number of deployed sensor nodes at the sensing layer will reduce because they are able to share a common aggregated VWSN without changing the underlying hardware network. The also supports improved scalability and reliability of the network [27].

Virtualization can be achieved at the sensor node level or at the network level [5]. Virtualization allows a sensor node to have concomitant access to several application tasks at a time [28]. Virtualization in nodes and on the network level is made possible by using operating systems, middleware and virtual machine approaches. Thus, the state-of-the-art in terms of these various approaches is provided in the next subsections.

2.1. Operating Systems

Operating Systems (OSs) for shared sensor networks are typically designed to alleviate the redundant deployment of sensor nodes in WSNs. We evaluate several OSs that consider resource sharing in their implementation. The design of the architecture for a WSN OS can be considered from a technical and a non-technical perspective. Technical considerations include WSN architecture, modularity, scheduling model, memory allocation, networking, programming model, debugging tools, programming language and hardware abstraction layers. Non-technical issues include documentation, certification, code maturity and licensing. A wide range of OSs for shared sensor systems is evaluated. Our review covers the following OSs: Contiki [29], RIOT [30], TinyOS [31], OpenWSN [32], LiteOS [33], nuttX [34], PAVENET [35], MANTIS [36], FreeRTOS [37], mbed OS, SenSmart [38] and SenSpire [39]. Other OSs that exist but are not covered in the table are eCOS [40], uClinux [41], ChibiOS/RT [42], CoOS [43], nanoRT [44], Nut/OS, ERIKA Enterprise, MansOS [45], NanoQPlus [46], RTEMS, Lorien [47], ThreadX [48], QNX, PikeOS [49] and Nucleus RTOS [50]. A lack of highly-deterministic performance levels are typically exhibited by the platforms provided in Table 1, which is a critical factor to be considered when developing virtual frameworks. Future platforms need to address the lack of real-time performance in OS paradigms. Tiny OS lacks dynamic resource allocation while its counterparts provide dynamic resource management.

Dynamic resource allocation is critical in decoupled ecosystems because the entire framework relies on on-demand resource utilization. Resources are not fixed to an application and thus they are shared among multiple application tasks. Virtualization techniques for resource sharing are implemented across the different OS platforms with a view to examine their efficiency levels. Contiki offers a rich networking stack that supports IoT ready protocols. It is imperative for IoT platforms to enable interactions between existing IPv4 networks, IPv6 and IEEE 802.015.4 based protocols [17]. Contiki supports μIP, which is compatible with IPv4, μIP6, IPv6, Rime stack, and with IEEE 802.15.4 related protocols. It is also imperative for the operating systems to support routing protocols for low power and lossy networks (RPL) [51] as they are integral components of the future progression of WSNs. The maintenance of sensor nodes should also be considered as it becomes daunting to maintain these nodes if they do not support over-the-air (OTA) updates. This means that for every firmware upgrade each sensor needs to be physically recalled for programming, which becomes a very difficult exercise to manage, considering the scale of future WSN. The need to support OTA is inextricably bound to the evolution of WSN towards shared ecosystems.

In addition to the specific VWSN OS compared in Table 1, we provide further discussion regarding some well-known OSs particularly considered for VWSN as follows:

(1) SenSmart: SenSmart is a sensor based OS that supports simultaneous application tasks in resource constrained nodes [38]. In order to provide concurrent execution of different application tasks, SenSmart is designed with a stack allocation system that is managed dynamically at run time. This enables an unused stack space to be reclaimed from expired tasks that no longer require it. When a new task is initiated to run, the content of the current task is compressed and saved in a circular buffer for its resumption. This mechanism typically supports the concept of virtualization in WSN as it enables more nodes to access limited system resources as required. SenSmart is an event-driven programming model and thus follows a sense-and-send workflow model. This further supports its use in VWSN. It has been implemented in some hardware platforms including Mica2/MicaZ [38]. However, it is found that SenSmart uses more CPU cycles for same applications than the TinyOS.

(2) RIOT: RIOT is an Internet of Things (IoT) specialized OS designed to support the use of diverse hardware resources in the IoT [30]. Its main aim is to provide real-time multithreading support, ensure a friendly programming model, while providing support for resource-constrained devices using low power consumption transmission technologies. RIOT is still a work in progress with no technical performance comparisons with existing Oss [30]. However, regarding VWSN, RIOT uses a realtime thread-based programming model in which different services are encoded in standard ANSI C/C++ languages to run in parallel. Thus, application tasks are encoded independently of the hardware and software in order to run them on different devices. This is a key feature required for VWSN.

(3) SenSpire: SenSpire is an event-driven and thread-based programming model [39]. SenSpire adopts a multilayer abstraction approach in order to develop networked applications. Regarding VWSN, SenSpire ensures that tasks can be programmed as events or as threads. In this case, event tasks typically have higher priority than thread tasks [39]. This ensures that the OS reacts more to external requests for system resources, thus facilitating broader use of the same system

resource. It has less interrupt latency than the TinyOS, although with more overhead scheduling delay than the MANTIS OS.

(4) PAVENET: This is a thread-based OS for handling issues regarding preemption of multithreaded application tasks [35]. Its use is highly limited to the PIC18 microchip and cannot be deployed on other hardware platforms such as MICAZ. In order to support VWSN, the PAVENET OS supports thread-based programming and the use of C language. It is possible to ensure varying priority levels via the use of programmed multithreaded applications [35]. The main limitation of PAVENET is its lack of portability across diverse hardware platform.

(5) MANTIS: MANTIS is also a thread-based embedded OS that supports concurrent execution on sensor nodes [36]. It is considered for VWSN because it is completely thread-based and typically easy to program without the need to manage low-level details of the stack/memory. The time-sliced multithreading approach ensures that several application tasks can run concurrently without using a run-to-completion model [36].

(6) LiteOS: It is a Unix-like OS particularly considered for sensor nodes [33]. It adopts a hierarchical file system with a command shell that works wirelessly. LiteOS is highly flexible for VWSN because it uses a hybrid programming model that combines both simultaneous execution of application threads and events through a call-back mechanism. Application tasks can be programmed in C language[33]. Installation and the execution of application tasks is very simple and can be accomplished by dynamically copying user applications. It is highly viable for deployment in VWSN.

(7) Contiki: It is one of the most popular OSs for WSN. It provides the concept of protothreads, which combines the concepts of event-driven and thread-based approaches [29]. This allows applications and services to be dynamically uploaded/unloaded wirelessly on sensor nodes. For VWSN, Contiki is highly applicable because it supports multiple applications that are typically independent of the OS and can invariably run on top of it. Applications can be programmed in C language and updated/installed without reinstalling the entire OS.

(8) TinyOS: It is an application-specific, component-based OS that is event-driven and offers a flexible platform for innovation [31]. It is written in a variant of C-language called nesC. It may not necessarily be the most viable for VWSN because it is mainly event-driven. However, efforts are currently underway to create variants that may be suitable for VWSN.

Table 1. A comparison of some of the operating systems of interest.

Platform	**Contiki**	**TinyOS**	**MANTIS**	**OpenWSN**	**LiteOS**
Real-Time	No	No	No	No	No
Hardware Platforms	ESB, TelosB, Tmote Sky	MICA (z)(2), TelosB, Iris, Shimmer	MICA(2)(z), Telos, MANTIS nymph	TelosB, GINA, WSN430, Z1, OpenMoteCC2538	MICAz, IRIS
Virtualization	Serial Execution	Yes	Semaphores	Yes	Synchronization primitives
Static or Dynamic	Dynamic	Static	Dynamic	Dynamic	Dynamic
Network Support	uIP, uIP6, Rime	Active message	Comm	6LoWPAN, RPL, CoAP	Message-based
Simulation	Cooja. MSP-Sim, NetSim	TOSSIM, Viptos, Qualnet	XMOS	Open Visualizer, OpenSim	AVRORA
OTA	Yes	Yes	No	Yes	Yes
Latest build	2.2.1	2.0	1.0 Beta	1.8.02	1.0
Multi-threads	Yes	Tiny-threads	Yes	Yes	Yes
Release date	2004	2000	2005	2011	2008
Concurrent execution	Yes	Yes	Yes	Yes	No
References	[52–54]	[31,55,56]	[36,57,58]	[32,59]	[37]

2.2. Middleware and Virtual Machine-Based Approaches

The middleware approach affords a developer the opportunity to operate at a layer above the host OS, as exemplified in Figure 2. Middleware is intended to satisfy a wide range of system design goals that are a challenge in traditional WSN. Some of the goals are to provide a high QoS [60], ease of programming, efficient resource management [61], scalability [62], agility support, reconfiguration of nodes [63], heterogeneity [64], virtualization [65], interoperability [66], large data sets and multi-radio support [67,68]. Several academic research efforts have been put into modelling and implementing middleware for sensor nodes.

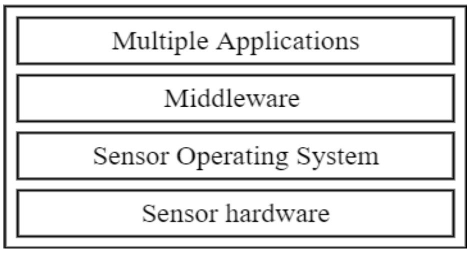

Figure 2. Middleware Approach.

Early efforts in this regard include Impala [69], EnviroTrack [70], Mires [71], Cougar [72], Smart Message [10], MiLAN [62], TinyLime [73], DSWare [74], TinyCubus [75] and TinyDB [76]. These implementations are typically plagued by different challenges, ranging from weak abstraction support and data fusion to a lack of dynamic topologies, rigid programming models, a lack of QoS support and a limited security policy [60,77–79]. Some further research efforts are noted to have addressed the challenges concerning the middleware and VM-based approaches to WSNs. They yielded better efficiencies than the previous implementations. State-of-the-art implementations include among others include Agilla [80], VMStar [81], SenaaS [82], UMADE [83], Squawk VM, Mate [84], Nano-CF [44] and SenShare [85].

The programming model is predominantly threaded and event driven. Event driven paradigms enable energy conservation by only sending data/information when critical threshold conditions have been reached, otherwise the messages will not be critical and energy-worthy of transmission. The threaded model allows for the design of distributed algorithms while enabling the underlying network to conceal the heterogeneous nodes. However, programming overheads are experienced in [69]. There are limited emphases on real-time performance in the surveyed platform. Earlier implementations may not have focused on this aspect, however, both SenShare [85] and Pavenet [86] are noted to have elements of real-time performance evaluations. The dynamic nature of virtualized platforms indicates that highly-deterministic behavior is required for the success of distributed logically connected layers.

A few notable VM based solutions are briefly discussed with regards to VWSN, and summarized in Table 2, as follows:

(1) VMSTAR: It is a Java-based software framework for developing application-specific virtual machines [81]. It supports the sequential and simultaneous use of thread-based applications. For VWSN, VMSTAR does not support the simultaneous use of multi-thread application tasks, instead, it supports only single-threaded Java applications. However, concurrent events can be handled using action listeners [81]. This can be used to identify high priority threads so that expired threads can be relived of system resources to cater for other application tasks.

(2) Squawk: This is also a Java virtual machine that runs on sensor hardware [84]. Different from VMSTAR, Squawk does not require an OS in order to run, instead, all its basic requirements are inbuilt. For VWSN, Squawk adopts a different approach compared to other solutions. First, it provides an application isolation mechanism, which enables multiple application tasks to be treated as Java objects [84]. Thus, applications can have multiple threads, which are managed by the Java Virtual Machine (JVM).

(3) Agilla: It is a mobile agent-based middleware that runs over the TinyOS along with a VM engine to conduct sequential execution of multiple applications [80]. This is normally done in a round robin

manner. For VWSN, Agilla depends on the TinyOS in order to provide simultaneous execution of tasks. It also guarantees this via the mobile agents executed in a round-robin style [80]. However, the difficulty in the programming language adopted by Agilla typically limits its use for VWSN. It adopts a low-level assembly-like language, which can be very difficult to modify or to build upon.

(4) UMADE: UMADE is a mechanism provided to promote fair utilization of resources among multiple contending applications [83]. It is typically built based on the Agilla VM and the TinyOS. For VWSN, UMADE typically uses Agilla for virtualization, while extending Agilla in order to provide dynamic memory management for concurrent applications.

(5) Nano-CF: It is a macro-programming framework for in-network programming and execution of multiple applications in WSN [44]. It adopts a proprietary OS called Nano-RK operating system, which enables several applications to use a common WSN architecture. This makes it suitable for VWSN. For VWSN, it allows independent application developers to write application tasks for a common WSN infrastructure [44]. These application tasks run independently and are not coupled to the sensor OS. It highly suited for data acquisition with sensor nodes having multiple on-board sensors.

2.3. Node Virtualization

The birth of newer sensing application areas has motivated a new ecosystem that requires the sharing of resources from a hardware perspective. The sensor node is the basic unit of a WSN and its efficient utilization is vital to the overall performance of the entire wireless network. The ability to virtualize the application services running on the network produces benefits that are analogous to those of virtualization in computing systems.

Table 2. Comparison table for wireless sensor middleware and VM-based platform.

Platform	SenShare	Pavenet	Agilla	Squawk VM	VMStar
Programming Model	Event-driven	Thread	Tuple-space and mobile agents	Thread	Thread
Real-time Performance	Yes	Yes	No	No	No
Communication Protocols	CTP	Not discussed	Not discussed	6LoWPAN, CTP, LQRP	Not discussed
Decoupling	Yes	No	Yes	No	No
Programming Language	nesC	C	Assembly	J2ME	Java

There are two strategies that have been adopted to address the virtualization of sensor nodes. These are the Sequential and the Simultaneous execution methods [87]. The Sequential execution method [87] is a rather less efficient means of virtualization because the application tasks run in sequence. In simultaneous execution, each process or task is given a time slice or quantum, and context-switching occurs based on the time slices allocated [88]. Sensor node virtualization allows for the running of several application tasks on a single node. This model overcomes the cost imperatives that accompany the redundant deployment of sensor nodes.

Traditionally, sensor nodes are application-specific in a single domain and these sensor nodes are not used for any other application. A new sensor node is deployed in the event of a new sensing application. A major drawback associated with multiple applications sharing a sensor hardware is that the devices have limited resources. The real estate on the current sensor nodes is such that the platforms for virtualization cannot fit into the hardware resources available, particularly in terms of computation, communication and storage capabilities. Advances in processor technologies have yielded more power-efficient processors with larger memory spaces and computing power. Virtualization platforms can therefore be hosted in these nodes. Node virtualization addresses some of these challenges by creating a platform that will enable resource sharing and application management on a single node. Multiple applications can run on the node using a virtualization framework [83]. The basic concept of node virtualization is shown in Figure 3.

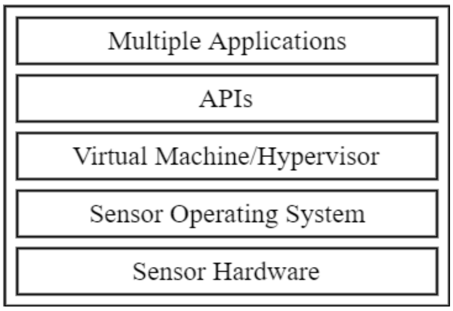

Figure 3. Multiple application execution on a single node [28].

Virtualization can also be achieved through type 1 or type 2 virtualization. Type 1 (also called bare metal) and type 2 (also called hosted) [89] have been used in a myriad of operating systems for WSN virtualization. Node level virtualization can be achieved using sensor operating systems (OS) or virtual machine/middleware- (VM/M) based solutions [90]. These two approaches assume different architectures when realizing virtualization. In the OS approach, the OS handles the multiple services that run on the node. Event-driven and thread programming models are dominant. In the VM/M approach, the VM system operates above the host operating system. This is shown in Figure 4. The approach is split into three: OS-based, middleware and VM-based approaches.

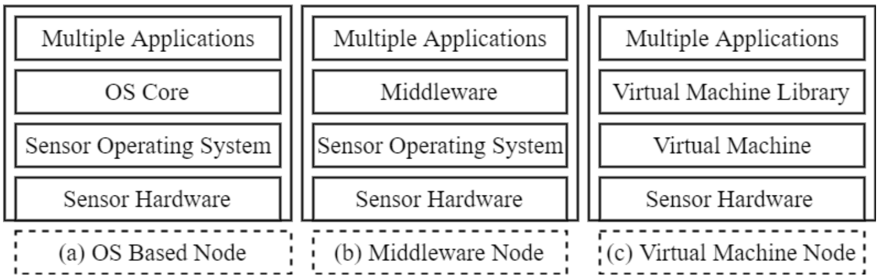

Figure 4 (**a**) OS-based Solution (**b**) Middleware-based solution (**c**) VM-based solution.

Some efforts are noted in Table 3 for improving node virtualization based on the different OSs, the different types of middleware and the different VM-based platforms in use. Some qualitative metrics are presented to compare the different platforms. Examples of these metrics include the type of programming model being adopted by the different platforms, the consideration of resource discovery, the platform type being supported, virtualization level, heterogeneity, platform independence, multi-radio support, programming language and the communication protocols in use. Summarily, our comparison in Table 3 presents a quick view of the viability of different possible platforms available for the effective and efficient virtualization of WSNs.

The following are some of the points of interest noted from the comparison in Table 3:

(1) Event-driven programming model is more prevalently adopted for VWSN than the threaded-driven model. The is prevalence may be because VWSN nodes need to stay in the idle or sleep mode and may only be required to transmit data whenever there is a significant change in the parameter(s) being monitored. This typically makes the event-driven model more power preserving than the threaded mode. Thus, nodes can easily send signals at longer time intervals (for example, in 24 h intervals) in order to inform the network about their continuous existence.

(2) The event driven model is typically slower in execution than the threaded driven model. This makes it quite poor in managing VWSNs in highly dynamic resource environments [9]. On the other hand, a few notable threaded programming models are more capable of resource discovery, for example, the RIOT platform. For this reason, frameworks such as the RIOT platform are typically encouraged

for the implementation of VWSNs.

(3) It is noted in Table 3 that some platforms lack resource discovery and publication services, which are very critical requirements in the management of the entry of new nodes into the network. Thus, platforms or frameworks, which are typically desired for VWSNs should possess dynamic management capabilities for efficient resource distribution in virtual environments.

(4) Most platforms for VWSN are typically OS based. This implies that there is a greater trend towards the use of OS-based solutions than the use of virtual-machine based solutions. This may be attributed to the higher cost of development associated with using VM than OS based solutions.

(5) Based on the examination metrics adopted in Table 3, it is quickly noted that Contiki possesses more desirable characteristics than the other platforms. It notably supports more protocols based on its unique programming model, which enables it to easily combine both the event and threaded driven models. A close competitor to the Contiki platform is the RIOT platform notable for its ability to perform resource discovery. Thus, the Contiki platform may be a more generalized model to adopt in VWSN designs.

(6) Most node level virtualization platforms in Table 3 exhibit a strong sense of heterogeneity and platform independency. For example, multi-radio support is only supported in [29–31], which enables various channels to be utilized leading to a reduction in the network congestion rate.

In addition, it is worth noting that the network stack typically used in most platforms comprises of different IoT-enabled protocols that provide internet connectivity to the nodes. Essentially, these lightweight data exchange protocols for IoT paradigms can be clustered into two principal architectures, namely the broker-based [91] and the bus-based architectures [92]. In the broker-based models, the broker mediates between the publisher and subscriber. The broker is also responsible for storing, forwarding, filtering and prioritizing publishing requests. The protocols include MQTT [93], AMPQ, CoAP [94] and JMS [95]. In the bus model, the client publishes specific content to a defined subscriber without a negotiator. These classes of data exchange protocols include DDS [27], REST and XMPP [96]. Virtualized networks thrive with low payload data exchange protocols. The reduction of the payloads from thousands of bytes from web applications to tens of bytes in an IoT node results in reduced communication overheads between the nodes and the decoupled environment. This configuration leads to an improved network performance as shown in Figure 5.

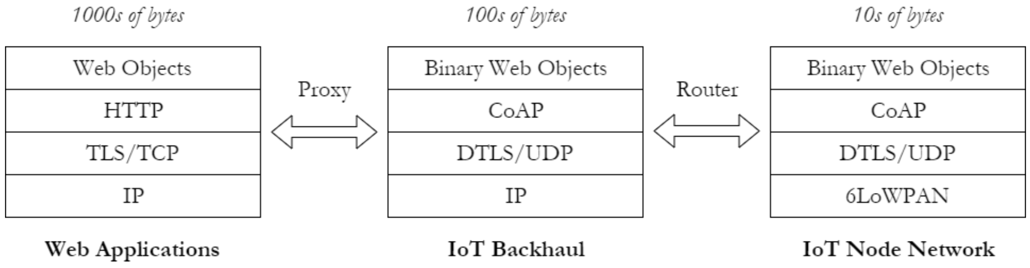

Figure 5. Comparison of web applications and IoT nodes interfaces.

2.4. Network Virtualization

Network virtualization allows for abstraction and sharing of WSN resources, while providing the impression of lone ownership. Network virtualization allows for the formation of dynamic logical groups of sensor nodes in which each group is set to a distinct domain. This leads to the formation of a VWSN, which is a subset of the WSN limited to a specific domain. Two approaches are discussed for achieving network-level virtualization, namely the cluster based and the overlay based approaches. The cluster-based topology achieves network virtualization by grouping logical instances of several sensor nodes from the underlying WSN infrastructure, while the Overlay networks leverages on the existing infrastructure to create a virtual topology at the application layer of the WSN. The cluster based approach is discussed in more details at this point, while the Overlay networks, being the main focus in this paper, are fully discussed in a later section.

In traditional WSNs, sensor nodes are arranged into groups called clusters. The sensors in that group communicate through the cluster head. The role of the cluster head is to communicate the aggregated information to the network sink, which in turn saves energy and bandwidth. This follows the analysis that the cost of conveying a bit of data is higher than processing it. Clustering in VWSN entails grouping logical instances of sensor nodes from the underlying network infrastructure and connecting them via virtual links to create sensor networks that are specific to an application task. The nodes are held together by logical links to form a VWSN.

Clusters are also formed by shared heterogeneous sensor nodes under different application domains. These clusters associate themselves based on their computing, communication, and processing capabilities, as well as based on their energy consumption rates. Rather than deploying private sensor systems in a particular field, these sensor nodes are shared by various end users. An example where clustering finds use is in an industrial surveillance system as part of the IIoT concept. The enterprises, stakeholders, various departments and the OEMs might need to access certain video footages. At the application layer, the relevant stakeholders can have an instance of the service they want to access. Not all end users need to deploy cameras in the respective areas, but they can share the existing infrastructure and each of them have applications running to access the surveillance system as if they are the only one accessing it.

Clustering can be realized through a static topology that offers local control and a dynamic topology that can change its network parameters on the fly. Following the clustering model, aggregated data can be sent to the sink node to reduce the number of nodes involved in the transmission. Clustering algorithms enable energy efficiency in the network and allow for improved scalability [97]. In this case, communication overheads are reduced for both single and multi-hop topologies.

3. Virtualization: A Solution to Some Challenges in WSN

Having presented an overview of VWSN, in this section we consider VWSN in view of some of the existing challenges in WSN. Future IIoT applications will most likely require the sharing of infrastructures as opposed to the lone ownership of the infrastructure by a specific provider. Many sectors of the global economy are shifting towards resource sharing models and the management of sharing is offered using on-demand service platforms. Traditional WSNs are deployed to a specific application area for sensing, communication, actuation, and for computational purposes. With increasing number of sensing applications, sensors might need to access data/information that lie outside the designated network or privileged data/information meant for specific users. Thus, sensor hardware infrastructures can be shared by various applications. Sharing sensor nodes are given access to the underlying hardware while being separate from each other in order to carry out their different roles. It further requires that the application tasks are not fixed to a particular node but are rather dynamically assigned to a node resource whenever they are required. Therefore, an application task is not guaranteed to use the exact sensor node it used in a previous task allocation. Access is granted using a distributed and dynamic key management technique [98] that allows multiple applications to access the underlying hardware. To this effect, the interaction between different applications and sensor nodes are not physical but logical leading to the concept of VWSN.

The existence of different applications that are logically linked to a particular hardware creates different topologies, which influences the security framework of the overall architecture. Thus, the security framework needs to be robust and resilient to enable confidentiality, availability and integrity of the application. In addition to security issues, there are other challenges that exist in WSNs and in the following subsections we present some of these challenges while considering the solutions offered by VWSN. The potential future impact of virtualization on security, scalability, quality of service, fault-tolerance, robustness and heterogeneity is discussed.

Table 3. A comparison of different virtualization platforms.

Platform	Programming Model	Resource Discovery	Type	Heterogeneity	Platform Independence	Multi-radio Support	Programming Language	Protocols
Contiki [99]	Protothreads	No	OS	Yes	Yes	Yes	C	HTTP, COAP, UDP, TCP, RPL, 6LoWPAN
RIOT [30]	Threaded	Yes	OS	Yes	Yes	Yes	ANSI C/C++	6LoWPAN, RPL
TinyOS [31]	Event-driven	No	OS	Yes	Yes	Yes	nesC	6LoWPAN, ZigBee
OpenWSN [32]	State-machine	No	OS	Yes	No	No	C	6LoWPAN, RPL, CoAP
FreeRTOS [37]	Threaded	No	OS	Yes	Yes	No	C	Third-party network stacks
VMStar [81]	Threaded	No	VM	No	No	No	Java	NA
SenaaS [82]	Event-driven	No	VM	Yes	Yes	No	NA	NA
SenSmart [38]	Event-driven	No	OS	Yes	Yes	No	nesC	NA
SenSpire [39]	Event-driven and threaded	No	OS	Yes	Yes	No	CSpire	CSMA, CSMA/CA, B-MAC, X-MAC
Agilla [80]	Tuple space and mobile agent	Yes	VM	Yes	No	No	Assembly-like	NA
LiteOS [33]	Event-driven and Threaded	No	OS	Yes	Yes	No	C	NA
PAVENET [86]	Threaded	No	OS	No	No	No	C	NA
MANTIS [36]	Threaded	No	OS	No	No	No	C	TDMA
UMADE [83]	Event-driven	No	VM	No	No	No	nesC	NA
Squawk VM [83]	Threaded	No	VM	No	Yes	No	J2ME	CTP, 6LoWPAN, AODV, LQRP
Nano-CF [44]	Event-driven	No	VM	Yes	Yes	No	Nano-CL	DSR, TDMA, B-MAC

3.1. Security

The goal of security in any connected and communication system is to ensure authentication of the sender and receiver, confidentiality of the data, integrity of the frame and availability of the network [100]. There is a need to constantly detect and stop attacks in a WSN. Connected IoT devices enable a large number of objects to be connected directly to the internet [101]. This wide exchange of information by connected IoT devices presents an appealing environment that malicious attackers may seek to undermine. Thus, security methods are required as countermeasures. In this regard, cryptographic primitives (public and symmetric key encryption) have been employed to address security threats [102]. The role of the public-key infrastructure (PKI) provides a set of tasks, guidelines and processes. Their function is to generate, allocate, utilize and rescind digital certificates among others functions. Because Symmetric-key encryption [103] method scales at a rapid rate, it thus becomes difficult to store these keys in the sensor node. It also suffers from a poor distribution technique. Public-key cryptography [56] is very complex for implementation in computation and storage limited nodes.

A new WSN architecture was introduced in [104] that discusses authentication of certificate authorities in WSNs using a public key setup. It works by providing an initial trust key between network nodes. With ECC being WSN-ready [105], it creates a platform for authentication via virtual certificates (AVCA) as discussed in [104]. AVCA provides mechanisms to overcome issues with regards to safeguarding many dispersed networks. It fosters simplicity, interoperability, mitigation of denial of service (DoS) attacks and scalability, which is ideal for VWSN.

Considering the evolution of VWSNs, the traditional internet security protocols are too resource intensive for integration into the sensor systems. Thus, lightweight security protocols are essential characteristics of any potential solution. Furthermore, security in WSN is tending towards ensuring trusted execution environments (TEE) in which security solutions are provided even from the physical (PHY) level of the OSI model [106,107]. For example, the ability for hardware partitioning in order to guarantee security suggests that virtual network frameworks can be implemented easily to avoid security attacks. Such end-to-end security approaches are enablers for VWSN. This concept is illustrated in Figure 6.

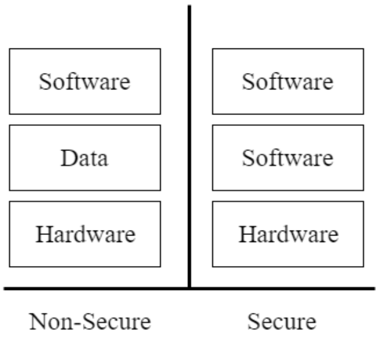

Figure 6. Non-secure and Secure environment.

In a sensor node chipset, a non-secure world might be defined as a hardware subset consisting of memory regions, caches and specific devices. Thus, non-trusted software can be limited to an environment that prevents access to, or even knowledge of the additional hardware required to support the architecture in the secure world. TEE offers a secure and locked execution of acute applications in virtualisation environments. Furthermore, system-wide security can also be provided by integrating the trusted region into the processor that interconnects the system peripherals [108,109]. The switch between the normal and the secured worlds is performed in hardware, thus eliminating the need for a hypervisor/VMM that has processing overhead. This yields real-time performance as well as lower power operation [110].

3.2. Scalability

With the addition of extra hardware units to an ever expanding network, the network architecture is required to dynamically adjust its parameters to accommodate the new units without causing any performance degradation. A network is scalable if it can dynamically handle the extra load injected into the system and yet perform at an optimal throughput. It is well known that the performance rate of most networks degrades as the load increases. Thus, this could be attributed to the constricted connection between the application services and the hardware.

3.3. Quality of Service (QoS)

Quality of service (QoS) [111] describes the quality metrics of a wireless sensor. These metrics include the hop time of a message from one node to the other, unbalanced network traffic, network dynamics, data redundancy and energy balance, amongst others. The parameters used to configure a network typically serve to define the bounds within which the network operates in. Usually, a network's quality is derived from the optimal performance of the network and its applications. QoS can be categorized as application-oriented QoS and network-oriented QoS [112]. Application-oriented QoS is concerned with the functions of the service tasks. The metrics of the application services include the locus of radio scope, number of active agents and the exactitude of parameters in the application. Network-specific QoS focuses on the fine grain details of the networking profile in order to ensure that minimal or no packet losses occur during the transmission or reception process. These abstractions normally inject latency and reduce reliability in the network. Virtual links in VWSN may address these challenges by mitigating the balance between the application-oriented QoS performance metrics and the network-oriented metrics. The ability to dynamically assign network components at runtime for application layer usage enables VWSN to contribute to the QoS improvement in WSN. Logical links also bring cohesion to the balance of traffic in WSN.

3.4. Fault-Tolerance

With the distributed nature of most WSNs, critical information bearing nodes could fail leading to an underperformance of an entire network. The failure of such nodes typically in a cluster-based topology must not affect the overall performance and operation of the network. Thus, the network architecture should have some means of self-healing and being able to recover from such setbacks. Dynamic and efficient routing algorithms should be able to reconfigure the network on runtime to accommodate the failing nodes without causing significant degradation to the network throughput. Virtual connections employed in VWSN will enable isolation of physical nodes that fail and other nodes can be assigned to the application task. A virtual approach will considerably reduce the effects of failing nodes.

3.5. Robustness

Resilient [113] networks are key to supporting federated sensor networks. Large-scale sensor interaction is prone to failure and these failing nodes tend to impact the network performance. A robust architecture is key to reducing these effects. The need to acclimatize to variations in the network configuration is relevant to creating robust networks. Network elements are created in modules for ease of adaptation, maintenance and isolation in the event of attacks. By considering the rate of increase in the number of application tasks in WSN, it is evident that tight coupling of services may not be the most suitable option owing to its restriction of the network from adapting to its changing sensing environment. Therefore, the decoupling of application tasks from their services and hardware and connecting them via virtual/logical links will foster a robust structure capable of adapting to the changing sensor environment.

3.6. Heterogeneity

The different application areas of sensor networks require that the nodes should have different physical capabilities. Some nodes have better computation and communication power [64]. Heterogeneous deployments tend to assume cluster-based [114] topologies, which require that the more powerful node should act as the leader of a certain cluster of nodes. The members of each cluster then communicate to the rest of the network via that cluster only. It is in this sense that the nodes deployed in the field for various sensing applications are heterogeneous in nature. The overall network topology and the architecture require that such provision should be made for this condition. The decoupling of sensor nodes from their respective applications implies that the virtual task manager may as well choose any node that optimally performs a certain application task without the need to use high-end nodes for simple tasks. Virtualization will in the long run promote the efficient utilization of resources.

4. Virtualization in WSN and the Concept of Overlay Networks

The internet was first developed as an overlay network to the traditional telephony system. It used the existing telephone infrastructure to render its services. Thus, overlays have a special history in the evolution of most technological services. Similarly, with the recent ubiquity of WSNs, it has become important to innovate over existing infrastructures in order to use these infrastructures to promote resource sharing. Essentially, Overlay networks are important because they enable transmission of data between end devices without the need to change the underlying hardware. The overlay approach is thus an appealing approach to achieving virtualization in WSN.

Olariu [115] in an early effort recognized that overlaying a virtual network over existing resources is an effective approach to deploying massive scale WSNs. Most applications and services that run on the overlay network normally execute their tasks concurrently in order to achieve resource sharing. Virtualization at the overlay layer decouples the applications and services from the existing hardware in order to give an illusion that each application task has sole control of the underlying hardware resource, while in effect it is being shared by multiple application tasks. Overlays operate at the edge of the network. They foster a distributed architecture thereby eliminating a single point of failure, which can improve network reliability and management [116].

Two approaches have been presented for the implementation of overlay VWSN (OVWSN), with the first been by Khan [12] in which a gate-to-overlay approach was used and the second by Xun [117] in which a gate to skip graph [118] approach was used. Khan introduced a layered design shown in Figure 7 for WSN virtualization using a gate to overlay layer. The entities in the overlay layer are grouped into logical/virtual groups that are assigned to a specific application. To address some of the shortcomings, the architecture proposed in [12] was further extended to enable interactions between VWSN Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) for the dynamic provision of different applications and services [12]. A base ontology concept was used to build a distributed and highly-deterministic annotation of raw sensor node data in order to allow semantic applications on the VWSN. An empirical based ontology for management and creation of VWSN was discussed in [119].

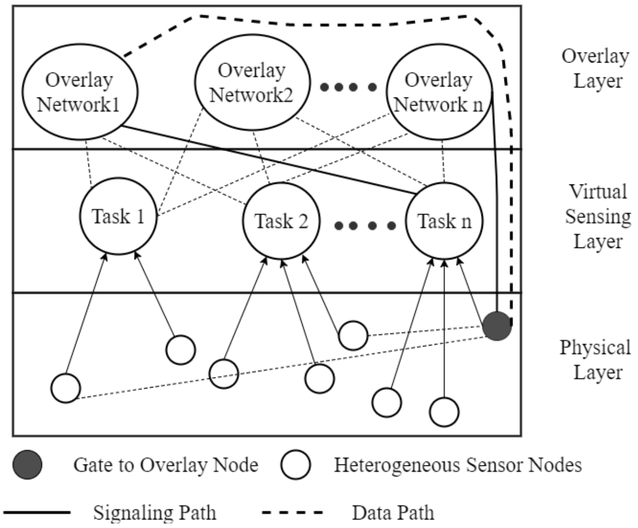

Figure 7. Gate to overheard virtualization technique.

An advantage of the platform is that it provides scalability in all layers. This is key for future deployments since it does not require re-deployment of the entire network in the event of new agents. The tasks in the virtual layer are loosely coupled with the agents in the overlay network, which can be expanded to multiple networks. The resource-weak nodes at the sensing layer are boosted to a higher capability using the gate-to-overlay node. There are two dedicated communication channels for signaling and data. The data channel allows for IoT-ready lightweight data exchange protocols between the interface layers. The signal path relays the network management and control related parameters, such as initialization and service management. The gate to overlay approach serves as an enabler for legacy and resource constrained nodes to participate in the virtual network area without a major infrastructural change. The drawback with this approach could possibly lie in the complexity of building the gate-to-overlay networks and the congestion and energy requirements of the node to meet the requirements of being a gateway. Dynamic formation of virtual sensing nodes strongly relies on the underlying sensing layer and their capability may be limited, hence the overload on the gate-to-overlay node.

Xun [117] on the other hand introduced a hierarchy–based overlay network approach leveraging on the gate-to-skip graph method as a possible approach to large scale distributed overlay networks. Earlier implementations of this approach yielded a self-aware sensing environment capable of identifying surrounding sensor nodes that join the network. This technique yielded a self-consolidating and distributed virtual sensor network. The entities of the network are used to extract the sensing resources that help to determine their fitness of participating in the overlay network. Figure 8 shows a generic approach of a hierarchy-based gate-to-skip configuration. This configuration embodies the idea that centralized control of resource management will tend to be a weakness in the system. Thus, a peer-to-peer interactive agent extension will better conceal the difference between the physical resources and the link varied in the WSN through the overlay layer. This introduces a query-like approach that sniffs the characteristics of each node in order to identify the node's suitability to participate in the overlay network.

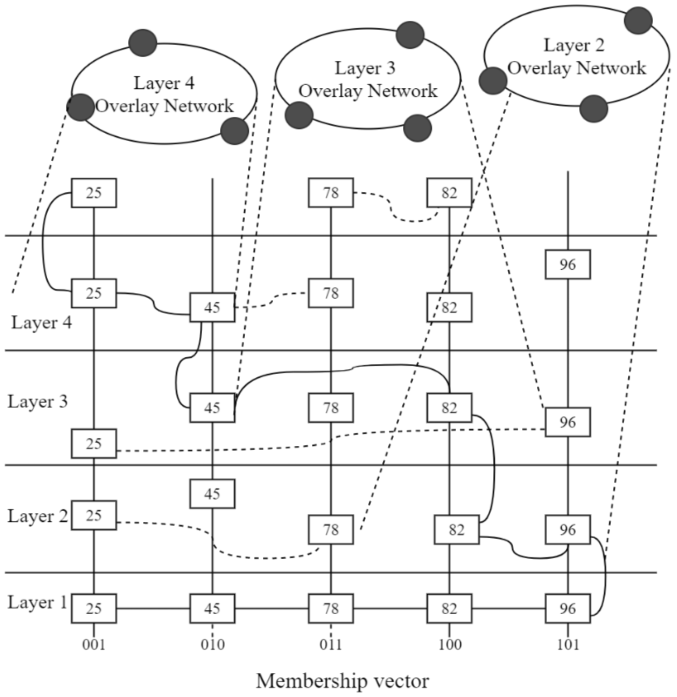

Figure 8. General Architecture of a Hierarchical Skip Graph Overlay Network.

Nevertheless, this approach lacks a rich resource discovery capability, which affects its usefulness in large-scale federated networks. One reason for this lack may be because the resource discovery technique is range based. Thus, in the next subsections, we discuss some of the key areas in the overall architecture of overlay wireless sensor networks, which include the topology, routing, media access, service discovery, resource allocation and the utilization requirements.

4.1. Topology

Fei in [120] introduced the design considerations to be considered for an overlay topology in a robust overlay WSN. The goal of the topology is to for the network to remain resilient under any condition ranging from the network payload, security, communication and computation. Its design consideration addresses critical factors that influence the direction of the architecture. Shuhei [121] also noted that a peer to peer overlay topology can act as both a client and server, thereby eliminating the level of overheads associated with the large volume of information transferred between the two terminals in the network. An overlay network with a server-less architecture introduces performance degradation due to the increase in the round-trip-time of a query sent to the network. This is because it needs to retrieve a specific request as it searches every node for the required information. Multilayer overlay topologies have a potential for alleviating this burden by introducing an alternate layer of the network. The proposed topology lacks a heuristic approach for heterogeneous networks that are the characteristics of WSN. Two topologies are discussed, peer-to-peer and ring topologies.

4.2. Peer-to-Peer Topology

The peer-to-peer topology is more prevalent than the ring topology in deploying overlay networks in virtualized WSNs. Peer-to-peer interactions are influenced by the mutual aid of the individual nodes that form the network. This is achieved by trusted entities that do not inject malicious activities into the network. Peer-to-peer topologies also improve network performance but further introduce restricted network competences and high failure recovery times. Their implementation lacks robustness, ease of diagnosis and it has limited privacy and security [122,123].

Figure 9 shows a general representation of an overlaid peer-to-peer topology in a VWSN model with a distributed overlay network. The virtual nodes in the overlay have interactions with nodes from different WSNs and also between themselves.

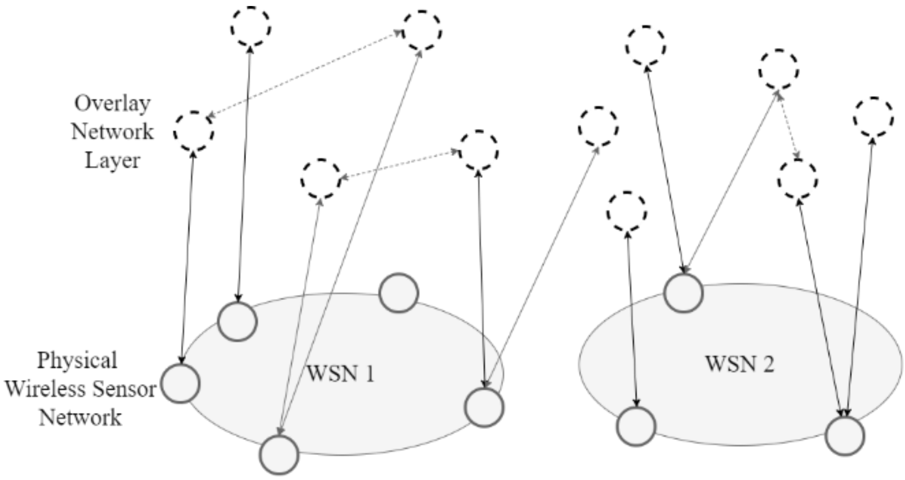

Figure 9. Peer-to-peer overlay topology.

Amy et al., in [124] proposed a structured overlay network built with in-network processing in view. The architecture considers the heterogeneity of the service providers, along with an overlay global framework that spreads across the overlay structure and stream processing for packet handling. Such architecture is critical for future networks as it offers resilient frameworks to enable resource sharing. Peer-to-peer interactions are further discussed in [125] to demonstrate the need for redundant communication paths in overlay networks to achieve resilience against nodes that can drop or be attacked via DoS [126]. The architecture of [125] further improves the availability of the network through a self-healing implementation.

Peer-to-peer topologies in overlay networks are further reinforced in [127] using a distributed model to connect end-users and gateways in mobile ad-hoc networks. Overlay networks are extended to support internet-of-things devices in a smart industrial application using a robust message exchange service for various city-wide applications such as seismic data and water leakages in the industrial distribution system [113]. The heterogeneity of the application areas in a smart industrial ecosystem renders an overlay approach very critical in addressing the scale and ubiquity of the sensor systems. It remains that the different implementation of peer-to-peer topologies presents a challenge with the vast designs that do not allowing interoperability of the different frameworks. A reference design will enable a standardized topology that can foster interoperability.

4.3. Ring Topology

Ring topology overlay networks are presented in [128] with varying in-network architectures, which have stubs along the ring. Figure 10 shows a general ring topology in overlay WSNs. The rings consist of passive and active elements. The topology is defined by the active nodes while the passive nodes only act as a means of relaying messages between the active nodes. The passive nodes are not always used and can be bypassed. This model has nodes that are static. The advantages of a ring topology is that it can support mutual exclusion and group management [129]. The topology is advantageous because the signal strength is enhanced as messages hop from one node to the other. The nodes have equal access to the available resources and no sink node is required for central control.

The drawback of the ring topology is that it may suffer from redundant data transfers on each loop, thereby draining the energy reserves of the nodes. Furthermore, the topology is prone to single points of failures, which affects the performance of the network due to the looping effect. Figure 8 shows the active and passive elements in an overlay ring topology. The main network is shown using alphabetic characters and stubs are shown using digits.

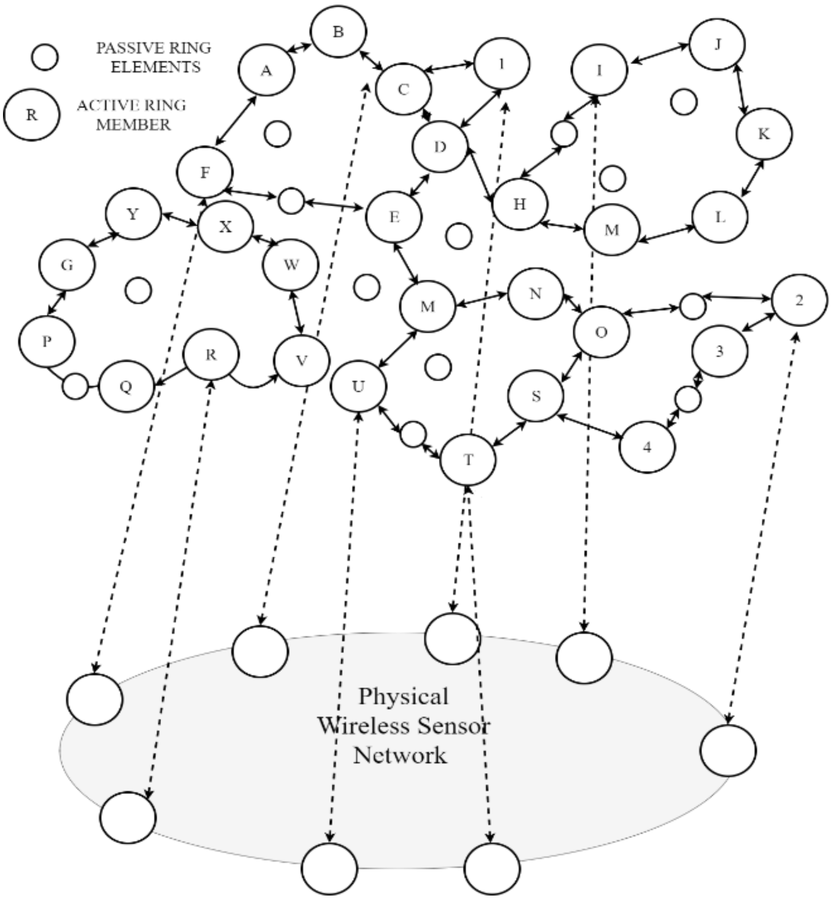

Figure 10. Ring Overlay Topology.

4.4. Routing

Routing is a pivotal aspect of a WSN architecture. Overlay networks need to conserve energy in their choice of routing profiles. The decentralized nature and dense deployment of these virtual networks means that data needs to be transferred from the node to the sink in the shortest possible path while being energy-aware. ScatterPastry is a peer-to-peer routing protocol for overlay networks that uses distributed Hash tables [130,131]. However, it lacks discovery techniques that could possibly increase the shelf life of the network.

A greedy perimeter technique was presented in [132] that addresses the use of small chunks of data per node to relay messages in a network to enhance the shelf-life of the network. Data-driven routing algorithms typically improve the robustness of the network as depicted in [133]. The minimum connected dominating set (MCDS) is another protocol used in robust overlay networks to achieve less energy consumption rates as compared to the shortest path related protocols [134].

4.5. Media Access

Spectrum access techniques in overlay networks are typically distinguished between the time frequency and the time division channels approach [135]. Likewise, channel access for overlay networks can be achieved via the use of intelligent techniques. As VWSNs continue to scale, the communication channel access will thrive using intelligent mechanisms for the management of access channels used in the overlay services. Chiti et al., in [136] proposed a versatile channel selector for overlay networks. Channel selection methods in traditional WSNs typically include time slotted channel hopping [137]. The use of cognitive radio methods uses an energy detector to select the communication channel [116]. Similarly, Game theory techniques also use channel selection methods [138,139]. This reduces the probability of channel access collisions by overlay applications and services. However, the game theory method may become inefficient as the network scales up.

4.6. Service Discovery

Services in WSN could be thought of as any object that sources information or manages a resource on behalf of another. Service discovery in overlaid virtual WSN could be centralized or decentralized [140–142] owing to the network architecture and by the purpose of the network. A service centric model for overlay VWSNs is proposed in [142] and contrasted with a network centric model prevalent in traditional WSN service discovery techniques. The service centric model was shown to yield improved congestion and better energy management as compared to the network centric model. Broadcast mechanisms are also discussed in [143–145], application layer protocols are described in [146,147], while reinforcement learning is presented in [148] in order to design robust frameworks for the OVWSN.

4.7. Resource Allocation and Utilization

Carmen et al., in [149] discussed about a framework for resource allocation in a virtual sensor network. As the number of applications increased, it was shown that resource allocation complexity emerges. This was driven by the fact that there is limited real-estate in the sensor node for computation, communication and energy management. The need to re-use the existing infrastructure has given rise to novel paradigms that address resource allocation and utilization. Sharief et al., in [150] also discussed about a three tier approach to efficient utilization. However, the implementation in [150] is collocated near the edge resource of the network, which poses a challenge for its use in shared sensor networks.

4.8. Summary

The overlay approach to VWSNs may still be considered as being recent, and for now are largely driven by peer-to-peer interactions. These interactions foster a server-less approach to data exchange between nodes. Resource limited nodes are thus capable of participating in virtual overlay networks by adding a gate-to-overlay node to enable buffering of the resource-limited node by the capable node. It is further noted that the hierarchy based skip graph overlay may be a favorable alternative to the modeling of the overlay network. It was also noted that the peer-to-peer topology for overlay design is more widely adopted than the ring and other topologies. This is because they are considered to be trusted counterparts that do not inject malicious activities into the network. Further notes include the fact that routing protocols for overlay networks may need to address the dynamic and loose couplings associated with overlay networks, while ScatterPastry and the distributed harsh table are quite applicable in peer-to-peer interactions. With the physical disconnect between the overlay layer and the underlying infrastructure, service discoveries have to be dynamic in order to manage the assets in the ecosystem. Finally, it was noted that the resource assignment in the VWSN is a volatile process and may thus be assigned based on-demand.

5. Overlay Virtualized WSN: Some Design Requirements

The use of Overlay networks in shared sensor networks will require the deployment of resource rich hardware. According to Moore [151], the processors are getting more efficient and faster. Thus, sensor nodes may soon be less limited by the available resources at the hardware level. Advances in silicon fabrication will also enable sensor nodes to be resource rich and energy efficient. Following this, it is envisaged that elastic network designs will ensure that routing protocols are capable of catering for the demands of virtual network implementations. As more multiple applications concurrently access the available resources, there is need to have a robust resource management and utilization framework. This framework will be able to adjust to the changing demands of the network. The co-existence of resource rich and constrained nodes will entail a high level of heterogeneity support.

Khan et al., in [12] addressed the challenge of heterogeneity support by adding a gate-to-overlay node. This node assists other resource limited nodes to participate in the overlay network via cooperation. Such an implementation introduces complexities in the architecture. The application services need to be isolated to ensure that the entities of the next application does not intrude into the next application without relevant authorization. This is done in conjunction with the security framework that needs to be put in place to mitigate the risks of concurrent execution of applications. Nodes that enter the network at runtime will be managed through the same security framework.

As the network scales on the physical side, the overlay architecture should be able to adjust dynamically to the underlying hardware. Connections between the overlay network and the physical network are not permanent are normally established on-demand. Consequently, their reliability becomes a function of the QoS.

Over-the-air (OTA) firmware updates remain a crucial aspect of the design requirement, as they enable wireless reprogramming. Generally, sensors should not be recalled from the field in order to conduct a firmware upgrade. Thus, a modular design will provide an ease of programming OTA. Modularity will enhance the system design and provide pluggable units that make maintenance and development manageable and efficient.

Furthermore, network interfaces on the application side should be IIoT compliant [152] by being ready and lightweight with a robust interoperability context. In this regard, resource discovery plays a pivotal role in the overall architecture of an OVWSN. Concurrent resource requests from different applications demand that the discovery techniques be highly efficient and flexible. The creation of the overlay topology should be fault tolerant and immune to single-point-of-failure problems. OpenThread is a potential solution to the self-healing demands of network topologies offering a mesh topology.

Following the above notes on some design requirements for OVWSN, the following summary is provided:

(1) *Multiple application execution*—The loose coupling between the application tasks in the overlay network and the underlying infrastructure enables dynamic resource sharing and for several applications to utilize same hardware with the illusion of lone ownership, which forms the core of shared sensor systems.

(2) *Robust and IoT ready routing protocols*—Several lightweight routing protocols exist in the IoT domain that are poised to take the lead in framing the standard protocols for data exchange in WSN. A few protocols like CoAP [153] have exhibited dynamic response to their operation in virtual frameworks as well as in overlay networks.

(3) *Resource-rich nodes*—Recent advances in silicon technologies have produced a rise in the number resource rich sensor nodes being used for WSN. With the increase in the demand for shared sensor systems, it is imperative that resource rich nodes allow for energy efficiency, yet producing high throughput. Sensing applications such as in video, seismic, terrestrial and volcanic activities suggest that high- end devices are needed to process these application areas. Consequently, multi-core embedded systems [151] are noted to have contributed to the progress in resource-rich physical layers. The aggregation of data before it is sent to the sink node is also required for these nodes to avoid redundant data being sent to the node. Only significant data/information is sent, which could signify a change in status from the previous value.

6. Overlay Virtualized WSN: Open Research Challenges

The ubiquity of sensor networks has given rise to the need to build overlay services and virtualize sensor nodes to reduce redundant deployment of nodes. With increase in the demand for more resources, there is urgent need to shrink the cost of deploying sensor nodes and to share the available hardware resources that these networks utilize. The starting point to limiting the increasing cost of hardware deployment is to improve sensor node sharing. The scale of sensor node deployment has led to the need to reuse the deployed hardware for new application areas. With virtualization allowing for the sharing of resources, applications, services and infrastructure, several problems and challenges arise. A few of these challenges in overlay WSN are presented as follows:

6.1. Real-Time Performance

The introduction of virtualization framework into the node's OS creates overheads that limit the performance of the OS. It is observed in Table 3 that PAVENET and SenShare are proponents involved in providing real-time performance in their technologies. Other models were noted to lack such capabilities. Thus, it is imperative to move towards near-real-time performance.

6.2. Advanced Node Virtualization

With the boom in the need to share sensor resources and build overlay networks on top of them, opportunities exist for deeper virtualization of discrete sensor layers like the MAC layer and routing layers. The efforts in [154] has proposed a service provisioning platform for cutting-edge middleware design that is aimed at a scalable, sustainable and secure virtualization platform, but no further work has been undertaken to realize this concept.

6.3. Publication and Discovery

Virtualized environments normally experience the complexities associated with implementing service discovery and publication. Virtual environments are available on demand and are terminated when they are no longer needed. This creates a need for dynamic management of resources used in the virtual environment. However, in conventional WSNs, a resource location and discovery service [155] coupled with CoAP provides for the standard peer-to-peer connection that does not rely on federated services for resources discovery. An implementation of this nature will address the challenges faced with publication and discovery in virtualized nodes.

6.4. Simulation Tools

The large deployment scale of sensor networks makes it difficult for new protocols and algorithms to be tested in typical physical network implementations. Simulation environments are thus needed to test and optimize the proposed algorithms and protocols. Simulators such as COOJA [156], XMOS and AVRORA [157] offer reliable and acceptable performance for WSN preliminary design. Efforts have been conceived to address aspects of simulators in virtual sensor systems in [158,159]. However, these are discrete approaches addressing aspects of the virtual environment. An integrated simulation framework offering end-to-end services is presented in [160]. This is platform-independent and comprises of a variety of different network APIs to provide a common communication paradigm. It also has the ability to split application requirements into events, time, query and data sections. It remains pertinent to create scalable and robust platforms to cater for the on-demand behaviour of virtual sensors.

6.5. Task and Sensor Node Assignment

A correlation exists between the types of sensors to be selected to perform a certain task in a WSN. This is based on the current task to be performed and the future sensing applications in which the node might be involved. A sensor-mission assignment framework is proposed for wireless sensor systems that use energy-conscious allocation algorithms [161]. The need to extend this allocation scheme to multiple applications that share a single node will enable shared sensor systems to efficiently and optimally assign nodes to their tasks.

6.6. Evolution of the Framework

Dynamic and evolutionary platforms are required for future virtual nodes. In recent years, WSNs and related technologies have been evolving towards the use of IoT [162]. Sensor nodes need to further evolve towards the use of overlay virtual nodes. Future sensor nodes need a framework that can adjust to new sensing environments without being reprogrammed. This could possibly be introduced through the use of artificial intelligent ecosystems in sensor nodes [163]. Future sensor node virtualization networks are expected to embrace machine learning [164] and deep learning [165] techniques in order to adapt to different sensing environments without the need for any hardware upgrade.

6.7. Abstraction Support

Heterogeneous implementations are typical in most WSNs. The different nature of hardware systems presents a challenge to the architecture of the operating system and its related packages. Different hardware vendors require different software routines in order to emulate the underlying hardware. This introduces complexity into the software design. A unified abstraction layer supporting heterogeneous platforms will reduce the complexity of the architecture and number of subroutines.

6.8. Energy Efficiency

Ensuring that sensor nodes are energy efficient is essential to the success of WSN deployments. While it is evident that future nodes will have more resourceful processors, energy-aware communication protocols and routing algorithms are necessary for the longevity of nodes in the field. Radio duty-cycling has been in the forefront of most energy conservation techniques, but it might not be the optimal approach to large scale overlay virtual networks. Thus, better energy efficient techniques and algorithms are required for OVWSNs. In addition, it is envisaged that energy efficient methods for resource management in OVWSN will be a point of concern in future designs. One notable effort in this regard can be found in [166] in which authors proposed a joint failure recovery, fault prevention and energy efficient resource management architecture for software defined networks (SDNs). In this regard, the energy consumption and reliability of selected paths are optimized, while maintaining the required Quality of Service (QoS) of the network. This work typically highlights some possible future research solutions that can be considered towards the realization of a more energy efficient OVWSN. Another recent work worth noting can be found in [167] in which authors considered the case of joint energy efficiency, QoS-aware path allocation and Virtual Network Function (VNF) placement for SDN. In this regard, authors propose a novel resource (re)allocation architecture to improve energy efficiency in SDN based networks. It is shown that the proposed solution achieves near optimal solutions particularly in terms of the execution time for real-life network deployment. Such effective solutions as in [167] will find future importance and application in OVWSN.

6.9. Security, Resource Management and Allocation

Security, resource management, as well as computational latency remain areas of concern in the quest to virtualize sensor nodes. The OSs, VMs and the middleware built for sensor node virtualization face the challenge of assigning and arraying applications in shared networks. Resource management seeks to enable efficient utilization of the underlying hardware that gathers data from the environment. The ubiquity of future sensor nodes and their connectivity to the internet will introduce a security and privacy component that needs to be aligned to the existing and future internet security layers. Lightweight security schemes need to be adopted that will provide the same integrity, confidentiality and availability as provided by the mainstream internet security framework.

Virtualized platforms at the sensor node will enable efficient resource utilization and conservation of the number of deployed nodes. Thus, dynamic frameworks are relevant for adaptive WSNs, which are networks that will be able to handle many applications on a single node. This will require that open communication standards are embedded in the network stack. Dynamic allocation and management of resources will be crucial in these future implementations. Efficient resource management and scheduling protocols for resource reservation and session management will ensure that real-time performance is guaranteed on the virtualized platforms.

6.10. Process Scheduling

Efficient process scheduling algorithms for sensing activities are also vital to minimizing energy losses in OVWSNs and are therefore important in the virtualization of sensor nodes. Large-scale federated sensor network platforms are potential considerations that will usher in the virtualization of sensor nodes. Connectivity between heterogeneous virtual sensor nodes will play a pivotal role in this paradigm and this is an area for further investigation.

6.11. Challenges from Other Emerging Technologies

There are several other emerging technologies designed with the aim to enhance the IIoT. An example of such a technology is Fog Computing (FC). In particular, FC has been recently applied in WSN with the goal to improve the performance of the IIoT. An extensive review was presented in [160] with regards to the merger between FC and the IoT. One challenge noted in [160] relates to the handling of the unpredictable large volume of data generated by different IoT based applications. It was noted that proximate Fog and remote Cloud data centres may present possible solutions to this problem. These are concepts much related to the development and use of VWSN. An energy efficient algorithm for fog-supported WSN was proposed in [161]. It was shown that the algorithm improved the network lifetime of WSNs by 40% compared to other known algorithms. Such algorithms can be further developed for inclusion in the OVWSN concept with the aim to improve the IIoT. A Fog supported smart city network architecture termed FOCAN was proposed for management of applications in the IoT environment in [162]. The FOCAN architecture presents as an interesting framework for application in OVWSNs. It is envisaged to further address the high energy consumption rates being experienced in the use of WSN in different IIoT applications, particularly those involving the use of the OVWSN. These new emerging technologies have a promising future in the application of OVWSN in IIoT. Finally, as Industrial IoT starts to included new emerging concepts, such as networks for multimedia and big data [168,169], OVWSN will also need to adapt to allow for QoS and high volume traffic.

7. Conclusions

It is envisaged that the use of sensor systems in the industrial internet of things will drive the next level of industrial productivity. These sensors will initiate tasks and communicate with other equipment in order to lower operational costs, prevent accidents and failures during operation and potentially take action in dangerous scenarios. One key technology that will significantly contribute to this vision is the overlay virtualized wireless sensor networks (OVWSN). The deployment of OVWSN will advance the use of shared sensor system towards reducing the redundant deployment of sensor nodes for different sensing applications. Overlay networks are vital to the success of the IIoT as they will provide a platform for the reuse of existing infrastructures in order to offer robust and dynamic services. Given the ever-growing study of OVWSN, we have in this paper presented a review of VWSN and the use of overlay networks concerning their application in IIoT. The various forms of shared sensor techniques were discussed. The concept of overlay services at the edge of the existing virtual wireless sensor network was also reviewed. It is noted that the efficient utilization of WSN resources is vital and the efficient management of resources will guarantee the real-time performance of the virtualized platforms. In general, because overlay networks will be instrumental in the future development and advancement of smart industrial and smart city applications, this review may be considered by researchers as a reference point for those particularly interested in the study of this growing field.

Author Contributions: M.N. was the main contributor responsibly for content collection, document drafting and comparative methodology. G.P.H., A.M.A.-M. and S.S. advised and supervised the first authors, contributing possible technical ideas and assisting to prepare the document. A.J.O. assisted with drafting some content and doing final editing.

Acknowledgments: This work was partially supported by a grant from City University of Hong Kong (Project No. 9678131).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khan, I.; Belqasmi, F.; Glitho, R.; Crespi, N.; Morrow, M.; Polakos, P. Wireless sensor network virtualization: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 553–576.
2. Ogbodo, E.U.; Dorrell, D.; Abu-Mahfouz, A.M. Cognitive radio based sensor network in smart grid: Architectures, applications and communication technologies. *IEEE Access* **2017**, *5*, 19084–19098.
3. Daugherty, P.; Banerjee, P.; Negm, W.; Alter, A.E. Driving Unconventional Growth through the Industrial Internet of Things. *Accenture Technol.* **2014**, *1*, 1–20.
4. Potter, C.H.; Hancke, G.P.; Silva, B.J. Machine-to-Machine: Possible applications in industrial networks. In Proceedings of the 2013 IEEE International Conference on Industrial Technology (ICIT), Cape Town, South Africa, 25–28 February 2013; pp. 1321–1326.
5. Sanchez, L.; Muñoz, L.; Galache, J.A.; Sotres, P.; Santana, J.R.; Gutierrez, V.; Ramdhany, R.; Gluhak, A.; Krco, S.; Theodoridis, E.; et al. SmartSantander: IoT experimentation over a smart city testbed. *Comput. Netw.* **2014**, *61*, 217–238.
6. Ntuli, N.; Abu-Mahfouz, A.M. A simple security architecture for smart water management system. *Procedia Comput. Sci.* **2016**, *83*, 1164–1169.
7. Silva, B.; Hancke, G.P. IR-UWB-Based Non-Line-of-Sight Identification in Harsh Environments: Principles and Challenges. *IEEE Trans. Ind. Inform.* **2016**, *12*, 1188–1195.
8. Michelin. *The MEMS Evolution3*. Available online: https://www.michelin.com/eng/content/.../ June2015-MICHELIN-MEMS3-ENG.pdf (accessed on 1 June 2015)
9. Abu-Mahfouz, A.M.; Hancke, G.P. Localised information fusion techniques for location discovery in wireless sensor networks. *Int. J. Sens. Netw.* **2018**, *26*, 12–25.
10. Khalid, Z.; Fisal, N.; Rozaini, M. A survey of middleware for sensor and network virtualization. *Sensors* **2014**, *14*, 24046–24097.
11. Bila, N.; Wright, E.J.; de Lara, E.; Joshi, K.; Lagar-Cavilla, H.A.; Park, E.; Goel, A.; Hiltunen, M.; Satyanarayanan, M. Energy-oriented partial desktop virtual machine migration. *ACM Trans. Comput. Syst.* **2015**, *33*, 1–51.
12. Khan, I.; Belqasmi, F.; Glitho, R.; Crespi, N. A multi-layer architecture for wireless sensor network virtualization. In Proceedings of the 6th Joint IFIP Wireless and Mobile Networking Conference, Dubai, UAE, 23–25 April 2013; pp. 1–4.
13. Khan, I.; Belqasmi, F.; Glitho, R. Wireless sensor network virtualization: Early architecture and research perspectives. *IEEE Netw.* **2015**, *29*, 104–112.
14. Gungor, V.C.; Lu, B.; Hancke, G.P. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3557–3564.
15. Hamari, J.; Sjöklint, M.; Ukkonen, A. The sharing economy: Why people participate in collaborative consumption. *J. Assoc. Inf. Sci. Technol.* **2015**, *3*, 80–90.
16. Cohen, B.; Kietzmann, J. Ride on! Mobility business models for the sharing economy. *Org. Environ.* **2014**, *27*, 279–296.
17. Cannon, S.; Summers, L.H. How Uber and the sharing economy can win over regulators. *J. Bus. Ventur.* **2012**, *27*, 544–558.
18. Zervas, G.; Proserpio, D.; Byers, J.W. The rise of the sharing economy: Estimating the impact of Airbnb on the Hotel Industry. *J. Mark. Res.* **2017**, *54*, 687–705.
19. Akpakwu, G.A.; Silva, B.J.; Hancke, G.P.; Abu-Mahfouz, A.M. A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. *IEEE Access* **2018**, *6*, 3619–3647.
20. Farooq, M.O.; Kunz, T. Contiki-based IEEE 802.15.4 node's throughput and wireless channel utilization analysis. *IFIP Wirel. Days* **2012**, 3–5, doi:10.1109/WD.2012.6402828.
21. Sud, S.; Want, R.; Pering, T.; Lyons, K.; Rosario, B.; Gong, M.X. Dynamic migration of computation through virtualization of the mobile platform. *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng.* **2012**, *17*, 206–215.
22. Wang, X.; Mu, Y.; Member, S. Addressing and privacy support for 6LoWPAN. *IEEE Sens. J.* **2015**, *15*, 5193–5201.
23. Kumar, A.; Hancke, G.P. A zigbee-based animal health monitoring system. *IEEE Sens. J.* **2015**, *15*, 610–617.
24. Gomez, C.; Oller, J.; Paradells, J. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors* **2012**, *12*, 11734–11753.

25. Gazis, V.; Görtz, M.; Huber, M.; Leonardi, A.; Mathioudakis, K.; Wiesmaier, A.; Zeiger, F.; Vasilomanolakis, E. A Survey of Technologies for the Internet of Things. In Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015.
26. Fouladi, B.; Ghanoun, S. Security Evaluation of the Z-Wave Wireless Protocol. *Black Hat* **2013**, *24*, 1–2.
27. Kang, W.; Kapitanova, K.; Son, S. RDDS: A real-time data distribution service for cyber-physical systems. *IEEE Trans. Ind. Inform.* **2012**, *8*, 393–405.
28. Khan, I. Cloud-Based Cost-Efficient Application and Service Provisioning in Virtualized Wireless Sensor Networks. Ph.D. Thesis, Institut Mines-Telecom, Telecom SudParis, Evry, France, 2015.
29. Dunkels, A.; Grönvall, B.; Voigt, T. Contiki—A lightweight and flexible operating system for tiny networked sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16–18 November 2004; pp. 455–462.
30. Baccelli, E.; Hahm, O.; Matthias, W.; Schmidt, T.C. RIOT OS: Towards an OS for the Internet of Things. In Proceedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; pp. 79–80.
31. Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Whitehouse, K.; Woo, A.; Gay, D.; Hill, J.; Welsh, M.; Brewer, E.; et al. TinyOS: An operating system for wireless sensor networks. In *Ambient Intelligence*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 115–148.
32. Wei, Z.; Jian, W.; Huan, G.; Songli, Z.; Wen, T.; Jin, Z. OpenWSN, an layer first component architecture and package for sensor networks. In Proceedings of the IET Conference on Wireless, Mobile and Sensor Networks 2007 (CCWMSN07), Shanghai, China, 12–14 December 2007; pp. 1047–1050.
33. Cao, Q.; Abdelzaher, T.; Stankovic, J.; He, T. The LiteOS operating system: Towards Unix-like abstractions for wireless sensor networks. In Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008), St. Louis, MO, USA, 22–24 April 2008; pp. 233–244.
34. Lopez, S.D. Low Power Embedded Software Optimization for the NuttX RTOS. Master's Thesis, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, 2013.
35. Hill, J.L. System Architecture for Wireless Sensor Networks. Ph.D. Thesis, University of California, Berkeley, CA, USA, 2003.
36. Abrach, H.; Bhatti, S.; Carlson, J.; Dai, H.; Rose, J.; Sheth, A.; Shucker, B.; Deng, J.; Han, R. MANTIS: System support for multimodAl NeTworks of in-situ sensors. In Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, San Diego, CA, USA, 19 September 2003; pp. 50–59.
37. Stangaciu, C.S.; Micea, M.V.; Cretu, V.I. Hard real-time execution environment extension for FreeRTOS. In Proceedings of the 2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings, Timisoara, Romania, 16–18 October 2014; pp. 124–129.
38. Chu, R.; Gu, L.; Liu, Y.; Li, M.; Lu, X. SenSmart: Adaptive stack management for multitasking sensor networks. *IEEE Trans. Comput.* **2013**, *62*, 137–150.
39. Dong, W.; Chen, C.; Liu, X.; Liu, Y.; Bu, J.; Zheng, K. SenSpire OS: A predictable, flexible, and efficient operating system for wireless sensor networks. *IEEE Trans. Comput.* **2011**, *60*, 1788–1801.
40. Lohmann, D.; Scheler, F.; Tartler, R.; Spinczyk, O.; Schröder-Preikschat, W. A quantitative analysis of aspects in the eCos kernel. *ACM SIGOPS Oper. Syst. Rev.* **2006**, *40*, 191.
41. Hyok-Sung, C.; Hee-Chui, Y. Context Switching and IPC Performance Comparison between uClinux and Linux on the ARM9 based Processor. In Proceedings of the Samsung Technology Conference, Seoul, Korea, 3–5 September 2005; pp. 1–4.
42. Berger, C.; al Mamun, M.A.; Hansson, J. COTS-Architecture with a Real-Time OS for a Self-Driving Miniature Vehicle. In Proceedings of the 2nd Workshop Architecting Safety in Collaborative Mobile Systems, Toulouse, France, 25 September 2013; pp. 1–12.
43. Mazmanov, D. A Device Message Bus for Sensor Networks. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2010.
44. Gupta, V.; Kim, J.; Pandya, A.; Lakshmanan, K.; Rajkumar, R.; Tovar, E. Nano-CF: A coordination framework for macro-programming in Wireless Sensor Networks. In Proceedings of the 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Salt Lake City, UT, USA, 27–30 June 2011; pp. 467–475.
45. Strazdins, G.; Elsts, A.; Selavo, L. MansOS. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10), Zürich, Switzerland, 3–5 November 2010; pp. 427–428.

46. Park, S.; Kim, J.W.; Shin, K.Y.; Kim, D. A nano operating system for wireless sensor networks. *Int. Conf. Adv. Commun. Technol.* **2006**, *1*, 345–348.
47. Porter, B.; Coulson, G. Lorien: A pure dynamic component-based operating system for wireless sensor networks. In Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks (MidSens '09), Urbana Champaign, IL, USA, 1 December 2009; pp. 7–12.
48. Kamboh, A.M.; Krishnamurthy, A.H.; Vallabhaneni, J.K.K. Demonstration of Multitasking using ThreadX RTOS on Microblaze and PowerPC. *Tech. Rep.* **2006**, *1*, 1–12.
49. Kuz, E.I.; Petters, S.M. *MIKES 2007: First International Workshop on MicroKernels for Embedded Systems*; NICTA Publisher: Sydney, Australia, 2007.
50. Hahm, O.; Baccelli, E.; Petersen, H.; Tsiftes, N. Operating systems for low-end devices in the internet of things: A survey. *IEEE Internet Things J.* **2015**, *3*, 720–734.
51. Tripathi, J.; de Olive, J.C.; Vasseur, J.P. A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks. In Proceedings of the 2010 44th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 17–19 March 2010; pp. 1–6.
52. Oikonomou, G.; Phillips, I. Experiences from porting the Contiki operating system to a popular hardware platform. In Proceedings of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS'11, Barcelona, Spain, 27–29 June 2011.
53. Guan, L.; Kuladinithi, K.; Potsch, T; Goerg, C. A deeper understanding of interoperability between tinyrpl and contikirpl. In Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.
54. Dunkels, A.; Eriksson, J.; Finne, N.; Osterlind, F.; Tsiftes, N.; Abeille, J.; Durvy, M. Low-power IPv6 for the Internet of Things. In Proceedings of the 2012 Ninth International Conference on Networked Sensing (INSS), Antwerp, Belgium, 11–14 June 2012; pp. 1–6.
55. Amjad, M.; Sharif, M.; Afzal, M.K.; Kim, S.W. TinyOS-New Trends, Comparative Views, and Supported Sensing Applications: A Review. *IEEE Sens. J.* **2016**, *16*, 2865–2889.
56. Malan, D.J.; Welsh, M.; Smith, M.D. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In Proceedings of the 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004, Santa Clara, CA, USA, 4–7 October 2004.
57. Abrach, H.; Bhatti, S.; Carlson, J.; Dai, H.; Rose, J.; Sheth, A.; Shucker, B.; Deng, J.; Han, R. Mantis-system supports for multimodal Networks on In-Situ sensors. In Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 336–337.
58. Mozumdar, M.M.R.; Lavagno, L.; Vanzago, L. A comparison of software platforms for wireless sensor networks. *ACM Trans. Embed. Comput. Syst.* **2009**, *8*, 1–23.
59. Stanislowski, D.; Vilajosana, X.; Wang, Q.; Watteyne, T.; Pister, K.S.J. Adaptive synchronization in IEEE802.15.4e networks. *IEEE Trans. Ind. Inform.* **2014**, *10*, 795–802.
60. Hua, N.; Yu, N.; Guo, Y. Research on service oriented and middleware based active QoS infrastructure of wireless sensor networks. In Proceedings of the I-SPAN 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks, Kaohsiung, Taiwan, 14–16 December 2009; pp. 208–213.
61. Shah, K.; Kumar, M. Resource Management in Wireless Sensor Networks using Collective Intelligence. In Proceedings of the 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Sydney, Australia, 15–18 December 2008; pp. 423–428.
62. Hadim, S.; Mohamed, N. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE Distrib. Syst. Online* **2006**, *7*, 1–23.
63. He, D.; Chen, C.; Chan, S.; Bu, J. SDRP: A secure and distributed reprogramming protocol for wireless sensor networks. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4155–4163.
64. Fazio, M.; Paone, M.; Puliafito, A.; Villari, M. Heterogeneous sensors become homogeneous things in smart cities. In Proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Palermo, Italy, 4–6 July 2012; pp. 775–780.
65. Chowdhury, N.M.M.K.; Boutaba, R. A survey of network virtualization. *Comput. Netw.* **2010**, *54*, 862–876.
66. Sarakis, L.; Zahariadis, T.; Leligou, H.-C.; Dohler, M. A framework for service provisioning in virtual sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2012**, *2012*, 135.
67. Canfeng, C.; Jian, M. MEMOSEN: Multi-radio enabled MObile wireless SEnsor network. In Proceedings of

the 20th International Conference on Advanced Information Networking and Applications (AINA'06), Vienna, Austria, 18–20 April 2006; Volume 2, pp. 291–295.

68. Zhao, Z.; Yang, G.H.; Liu, Q.; Li, V.O.K.; Cui, L. EASITEST: A multi-radio testbed for heterogeneous wireless sensor networks. In Proceedings of the IET Conference on Wireless Sensor Network, Beijing, China, 15–17 November 2010; pp. 104–108.
69. Martonosi, M.; Liu, T. Impala: A middleware system for managing autonomic, parallel sensor systems. *ACM SIGPLAN Not.* **2003**, *38*, 107–118.
70. Abdelzaher, T.; Blum, B.; Cao, Q.; Chen, Y.; Evans, D.; George, J.; George, S.; Gu, L.; He, T.; Krishnamurthy, S.; et al. Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In Proceedings of the 24th International Conference on Distributed Computing Systems, Tokyo, Japan, 23–24 March 2004; pp. 582–589.
71. Souto, E.; Guimarães, G.; Vasconcelos, G.; Vieira, M.; Rosa, N.; Ferraz, C.; Kelner, J. Mires: A publish/subscribe middleware for sensor networks. *Pers. Ubiquitous Comput.* **2006**, *10*, 37–44.
72. Hermann, C.; Dargie, W. Senceive: A middleware for a wireless sensor network. In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), Okinawa, Japan, 25–28 March 2008; pp. 612–619.
73. Curino, C.; Giani, M.; Giorgetta, M.; Giusti, A.; Murphy, A.L.; Pietro, G. Mobile data collection in sensor networks: The TinyLime middleware. *Pervasive Mob. Comput.* **2005**, *1*, 446–469.
74. Li, S.; Lin, Y.; Son, S.H.; Stankovic, J.A.; Wei, Y. Event detection services using data service middleware in distributed sensor networks. *Telecommun. Syst.* **2004**, *26*, 351–368.
75. Marron, P.J.; Lachenmann, A.; Minder, D.; Hähner, J.; Sauter, R.; Rothermel, K. TinyCubus: A flexible and adaptive framework for sensor networks. In Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN 2005), Istanbul, Turkey, 2 February 2005; pp. 278–289.
76. Madden, S.R.; Franklin, M.J.; Hellerstein, J.M.; Hong, W. TinyDB: An acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.* **2005**, *30*, 122–173.
77. Molla, M.M.; Ahamed, S.I. A survey of middleware for sensor network and challenges. In Proceedings of the International Conference on Parallel Processing Workshops, ICPP, Columbus, OH, USA, 14–18 August 2006; pp. 223–228.
78. Zhai, X.; Jing, H.; Vladimirova, T. Multi-sensor data fusion in Wireless Sensor Networks for Planetary Exploration. In Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Leicester, UK, 14–18 July 2014; pp. 188–195.
79. Sen, J. A survey on wireless sensor network security. *Int. J. Comput. Appl.* **2010**, *3*, 43–49.
80. Fok, C.-L.; Roman, G.-C.; Lu, C. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Trans. Auton. Adapt. Syst.* **2009**, *4*, 1–26.
81. Koshy, J.; Pandey, R. VMSTAR: Synthesizing scalable runtime environments for sensor networks. In Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05), San Diego, CA, USA, 2–4 November 2005; pp. 243–254.
82. Alam, S.; Chowdhury, M.M.R.; Noll, J. SenaaS: An event-driven sensor virtualization approach for Internet of Things cloud. In Proceedings of the 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications, Suzhou, China, 25–26 November 2010; pp. 1–6.
83. Bhattacharya, S.; Saifullah, A.; Lu, C.; Roman, G.C. Multi-application deployment in shared sensor networks based on quality of monitoring. In Proceedings of the 2010 16th IEEE Real-Time and Embedded Technology and Applications Symposium, Stockholm, Sweden, 12–15 April 2010.
84. Levis, P.; Culler, D. Maté: A tiny virtual machine for sensor networks. *Science* **2002**, *30*, 85–95.
85. Leontiadis, I.; Efstratiou, C.; Mascolo, C.; Crowcroft, J. *SenShare: Transforming Sensor Networks into Multi-Application Sensing Infrastructures*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 65–81.
86. Saruwatari, S.; Suzuki, M.; Morikawa, H. PAVENET OS: A compact hard real-time operating system for precise sampling in wireless sensor networks. *SICE J. Control* **2012**, *5*, 24–33.
87. Mottola, L.; Mottola, L. Programming wireless sensor networks: From Physical to Logical Neighborhoods. In Proceedings of the First International Conference on Integrated Internet ad hoc *and Sensor* Networks (InterSense '06), Nice, France, 30–31 May 2006; p. 299.
88. Kasten, O. A state-based programming model for wireless sensor networks. In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications, White Plains, NY, USA, 19–23 March 2007; pp. 261–266.

89. Popek, G.J.; Goldberg, R.P. Formal requirements for virtualizable third generation architectures. *Commun. ACM* **1974**, *17*, 412–421.
90. Islam, M.M.; Huh, E.-N. Virtualization in Wireless Sensor Network: Challenges and Opportunities. *J. Netw.* **2012**, *7*, 412–418.
91. IBM. *MQTT for Sensor Networks (MQTT-SN) Protocol Specification*; IBM: Armonk, NY, USA, 2013; p. 28.
92. Yazar, D.; Dunkels, A. Efficient application integration in IP-based sensor networks. In Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, Berkeley, CA, USA, 4–6 November 2009; pp. 43–48.
93. Thangavel, D.; Ma, X.; Valera, A.; Tan, H.X.; Tan, C.K.Y. Performance evaluation of MQTT and CoAP via a common middleware. In Proceedings of the IEEE ISSNIP 2014—2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Singapore, 21–24 April 2014.
94. Kovatsch, M.; Duquennoy, S.; Dunkels, A. A low-power CoAP for Contiki. In Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, Valencia, Spain, 17–22 October 2011; pp. 855–860.
95. Haase, K. *Java Message Service API Tutorial*; Sun Microsystems, Inc.: California, USA, 2002; pp. 1–278.
96. Saint-Andre, P. *Request for Comments: 6120 Extensible Messaging and Presence Protocol (XMPP): Core*; Internet Engineering Task Force (IETF) RFC 6120; IETF: Fremont, CA, USA, 2011; pp. 1–211, doi:10.17487/RFC6120.
97. Akkaya, K.; Younis, M. Energy and QoS aware routing in wireless sensor networks. *Cluster Comput.* **2005**, *8*, 179–188.
98. Alcaraz, C.; Lopez, J.; Roman, R.; Chen, H.H. Selecting key management schemes for WSN applications. *Comput. Secur.* **2012**, *31*, 956–966.
99. Österlind, F. A sensor network simulator for the Contiki OS. *SICS Res. Rep.* **2006**, *5, 1-40*.
100. Onel, T.; Onur, E.; Ersoy, C.; Delic, H. Wireless Sensor Networks for Security: Issues and Challenges. In *Advances in Sensing with Security Applications*; Springer: Dordrecht, The Netherlands, 2006; pp. 95–119.
101. Hennebert, C.; Santos, J.D. Security protocols and privacy issues into 6LoWPAN Stack: A Synthesis. *IEEE Internet Things J.* **2014**, *1*, 384–398.
102. Louw, J.; Niezen, G.; Ramotsoela, T.D.; Abu-Mahfouz, A.M. A key distribution scheme using elliptic curve cryptography in wireless sensor networks. In Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, 19–21 July 2016; pp. 1166–1170.
103. Senekane, M.; Mafu, M.; Petruccione, F. Six-State Symmetric Quantum Key Distribution Protocol. *J. Quantum Inf. Sci.* **2015**, *5*, 33–40.
104. Holohan, E.; Schukat, M. Authentication using virtual certificate authorities: A new security paradigm for wireless sensor networks. In Proceedings of the 2010 Ninth IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, 15–17 July 2010; pp. 92–99.
105. Fan, J.; Sakiyama, K.; Verbauwhede, I. Elliptic curve cryptography on embedded multicore systems. *Des. Autom. Embed. Syst.* **2008**, *12*, 231–242.
106. Arfaoui, G.; Gharout, S.; Traore, J. Trusted execution environments: A look under the hood. In Proceedings of the 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud 2014), Oxford, UK, 8–11 April 2014.
107. Palattella, M.R.; Accettura, N.; Vilajosana, X.; Watteyne, T.; Grieco, L.A.; Boggia, G.; Dohler, M. Standardized protocol stack for the internet of (important) things. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1389–1406.
108. Abu-Mahfouz, A.M.; Hancke, G.P. Evaluating ALWadHA for providing secure localisation for wireless sensor networks. In Proceedings of the 2013 AFRICON, Pointe-Aux-Piments, Mauritius, 9–12 September 2013; pp. 1–5.
109. Abu-Mahfouz, A.M.; Hancke, G.P. ALWadHA localization algorithm: Yet more energy efficient. *IEEE Access* **2017**, *5*, 6661–6667.
110. Mattison, P.E. Method and Apparatus for Protecting Flash Memory. U.S. Patent 5,778,070, 7 July 1998.
111. Trakadas, P.; Leligou, H.; Zahariadis, T.; Karkazis, P.; Sarakis, L. Managing QoS for Future Internet Applications over Virtual Sensor Networks. *Future Internet* **2013**, *7858*, 52–63.
112. Li, Y.; Chen, C.S.; Song, Y.; Wang, Z.; Li, Y.; Chen, C.S.; Song, Y.; Wang, Z.; Qos, R. Real-time QoS support in wireless sensor networks: A survey To cite this version. *IFAC Proc. Vol.* **2007**, *40*, 373–380.
113. Benson, K.E.; Han, Q.; Kim, K.; Nguyen, P.; Venkatasubramanian, N. Resilient overlays for IoT-based community infrastructure communications. In Proceedings of the 2016 IEEE First International Conference

on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 152–163.

114. Karthikeyan, T.; Audithan, S. An enhanced adaptive re-clustering protocol in wireless sensor network. In Proceedings of the 2nd International Conference on Current Trends in Engineering and Technology (ICCTET). Tamil Nadu, India, 8 July 2014; pp. 418–422.
115. Olariu, S.; Xu, Q. A simple and robust virtual infrastructure for massively deployed wireless sensor networks. *Comput. Commun.* **2005**, *28*, 1505–1516.
116. Ndiaye, M.; Hancke, G.P; Abu-Mahfouz, A.M. Software Defined Networking for Improved Wireless Sensor Network Management: A Survey. *Sensors* **2017**, *17*, 1031.
117. Shao, X.; Jibiki, M.; Teranishi, Y.; Nishinaga, N. Hierarchy-aware skip graph for sensing resource discoveries on large-scale sensor overlay networks. *Comput. Commun.* **2015**, *70*, 15–27.
118. Shao, X.; Jibiki, M.; Teranishi, Y.; Nishinaga, N. A low cost hierarchy-awareness extension of skip graph for world-wide range retrievals. In Proceedings of the 2014 IEEE 38th Annual Computer Software and Applications Conference, Vasteras, Sweden, 21–25 July 2014; pp. 438–445.
119. Jafrin, R.; Khan, I.; Sahoo, J.; Glitho, R. A framework for ontology provisioning in Virtualized Wireless Sensor Networks. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 743–749.
120. Xing, F.; Wang, W. On the resilient overlay topology formation in multi-hop wireless networks. In Proceedings of the 6th International IFIP-TC6 Conference on Ad Hoc Sensors Networks, Wireless Networks, Next Generation Internet (NETWORKING'07), Atlanta, GA, USA, 14–18 May 2007; pp. 1–12.
121. Kajimoto, S.; Asaka, T. Topology construction method for multi-overlay sensor networks. In Proceedings of the 2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS), Hiroshima, Japan, 25–27 September 2013; pp. 1–3.
122. Matos, R.; Sargento, S.; Hummel, K.A.; Hess, A.; Tutschku, K.; de Meer, H. Context-based wireless mesh networks: A case for network virtualization. *Telecommun. Syst.* **2012**, *51*, 259–272.
123. Akyildiz, I.F.; Wang, X.; Wang, W. Wireless mesh networks: A survey. *Comput. Netw.* **2005**, *47*, 445–487.
124. Babay, A.; Danilov, C.; Lane, J.; Miskin-amir, M.; Obenshain, D.; Schultz, J.; Stanton, J.; Tantillo, T.; Amir, Y. Structured Overlay Networks for a New Generation of Internet Services. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 1771–1779.
125. Heck, H.; Kieselmann, O.; Wacker, A. Evaluating connection resilience for the overlay network Kademlia. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2581–2584.
126. Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. *Computer* **2012**, *35*, 54–62.
127. Hormati, M.; Belqasmi, F.; Khendek, F.; Glitho, R. Overlay interconnection for end-user applications and wireless sensor networks in MANETs. In Proceedings of the 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), Paris, France, 22–24 July 2015.
128. Banerjee, A.; King, C.; Member, S. Building Ring-Like Overlays on Wireless Ad Hoc and Sensor Networks. In Proceedings of the 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Vancouver, BC, Canada, 9–12 October 2009; Volume 20, pp. 1553–1566.
129. Amgoth, T.; Jana, P.K. Efficient overlay construction for wireless sensor networks. *Wirel. Pers. Commun.* **2016**, *86*, 959–973.
130. Al-Mamou, A.A.B.; Labiod, H. ScatterPastry: An overlay routing using a DHT over wireless sensor networks. In Proceedings of the International Conference on Intelligent Pervasive Computing (IPC 2007), Jeju Island, Korea, 11–13 October 2007; pp. 274–279.
131. Thanh, V.V.; Chan, H.N.; Viet, B.P.; Huu, T.N. A survey of routing using DHTs over Wireless Sensor Networks. *Technology* **2009**, *9*, 978–981.
132. Karp, B.; Kung, H. GPSR: Greedy Perimeter Stateless Routing for wireless networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (ACM MobiCom), Boston, MA, USA, 6–11 August 2000; pp. 243–254.
133. Shenker, S.; Ratnasamy, S.; Karp, B. Data-centric storage in sensornets. *ACM SIGCOMM* **2003**, *33*, 137–142.
134. Shaikot, S. Energy aware routing in high capacity overlays in wireless sensor networks. In Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, Doha, Qatar, 31 March–4 April 2008 2008; Volume 11, pp. 276–283.

135. Liang, W.; Ng, S.X.; Hanzo, L. Cooperative overlay spectrum access in cognitive radio networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1924–1944.
136. Chiti, F.; Fantacci, R.; Tani, A. Performance evaluation of an adaptive channel allocation technique for cognitive wireless sensor networks. *IEEE Trans. Veh. Technol.* **2007**, *66*, 5351–5363.
137. Hahm, O.; Adjih, C.; Baccelli, E.; Schmidt, T.C.; Wählisch, M. A Case for Time Slotted Channel Hopping for ICN in the IoT. *arXiv* **2016**, arXiv:1602.08591.
138. Zhang, H.; Jiang, C.; Beaulieu, N.C.; He, S.; Chu, X. Cooperative bargaining resource allocation for cognitive small cell networks. In Proceedings of the 2014 IEEE Global Communications Conference (GLOBECOM 2014), Austin, TX, USA, 8–12 December 2014; pp. 3080–3085.
139. Valli, R.; Dananjayan, P. Transmission power control in virtual MIMO wireless sensor network using game theoretic approach. *Adv. Netw. Commun.* **2011**, *132*, 535–545.
140. Mayer, S.; Guinard, D. An extensible discovery service for smart things. In Proceedings of the 2nd International Workshop on Web of Things (WoT 2011), San Francisco, CA, USA, 12 June 2011; pp. 7–12.
141. McIlraith, S.A.; Son, T.C.; Zeng, H. Semantic Web services. *Intell. Syst. IEEE* **2001**, *16*, 46–53.
142. Munir, S.A.; Dongliang, X.; Canfeng, C.; Ma, J. Virtual overlay for service classification & discovery in wireless sensor networks introduction. *Discovery* **2009**, *2*, 991–996.
143. Van Luu, H.; Tang, X. On the construction of rings overlay for robust data collection in wireless sensor networks. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference (WCNC), Sydney, Australia, 18–21 April 2010.
144. Chowdhury, N.M.M.K.; Boutaba, R. Network virtualization: State of the art and research challenges. *IEEE Commun. Mag.* **2009**, *47*, 20–26.
145. Jannotti, J.; Gifford, D.K.; Johnson, K.L.; Kaashoek, M.F.; O'Toole, J.W., Jr. Overcast: Reliable multicasting with on overlay network. In Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation, San Diego, CA, USA, 22–25 October 2000; p. 14.
146. Chaudet, C.; Demeure, I. A model to evaluate brokers overlays for Publish/Subscribe in Wireless Sensor Networks. In Proceedings of the 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Courmayeur, Italy, 9–11 January 2012; pp. 71–74.
147. Chaudet, C.; Costagliola, N.; Demeure, I.; Ktari, S.; Tardieu, S. Building an efficient overlay for publish/subscribe in wireless sensor networks. In Proceedings of the 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Seoul, Korea, 18–21 June 2012; Volume 1, pp. 362–370.
148. Singsanga, S.; Hattagam, W.; Tat, E.H. Packet forwarding in overlay wireless sensor networks using NashQ reinforcement learning. In Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, Australia, 7–10 December 2010; pp. 85–90.
149. Delgado, C.; Gállego, J.R.; Canales, M.; Ortín, J.; Bousnina, S.; Cesana, M. On optimal resource allocation in virtual sensor networks. *Ad Hoc Netw.* **2016**, *50*, 23–40.
150. Oteafy, S.M.A.; Hassanein, H.S. Towards a global IoT: Resource re-utilization in WSNs. In Proceedings of the 2012 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 30 January–2 February 2012; pp. 617–622.
151. Munir, A.; Gordon-Ross, A.; Ranka, S. Multi-core embedded wireless sensor networks: Architecture and applications. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 1553–1562.
152. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376.
153. Raza, S.; Trabalza, D.; Voigt, T. 6LoWPAN compressed DTLS for CoAP. In Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS 2012), Hangzhou, China, 16–18 May 2012.
154. Navarro, M.; Antonucci, M.; Sarakis, L.; Zahariadis, T. VITRO architecture: Bringing virtualization to WSN world. In Proceedings of the 2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS), Valencia, Spain, 17–22 October 2011; pp. 831–836.
155. Mäenpää, J.; Bolonio, J.; Loreto, S. Using RELOAD and CoAP for wide area sensor and actuator networking. *EURASIP J. Wirel. Commun. Netw.* **2012**, *2012*, 121.
156. Kugler, P.; Nordhus, P.; Eskofier, B. Shimmer, Cooja and Contiki: A new toolset for the simulation of on-node signal processing algorithms. In Proceedings of the 2013 IEEE International Conference on Body Sensor Networks, Cambridge, MA, USA, 6–9 May 2013.

157. Titzer, B.L.; Lee, D.K.; Palsberg, J. Avrora: Scalable sensor network simulation with precise timing. In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN), Boise, ID, USA, 15 April 2005; pp. 477–482.
158. Hsieh, C.M.; Wang, Z.; Henkel, J. Dance: distributed application-aware node configuration engine in shared reconfigurable sensor networks. In Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, Grenoble, France, 25–27 March 2013; pp. 839–842.
159. Gupta, V.; Pereira, N.; Tovar, E.; Rajkumar, R.R. A harmony of sensors: Achieving determinism in multi-application sensor networks. In Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014; pp. 299–300.
160. Haghighi, M. Dynamic data storage estimation for multiple concurrent applications using probability distribution modeling in WSNs. *J. Adv. Comput. Netw.* **2013**, *1*, 254–259.
161. Rowaihy, H.; Johnson, M.P.; Liu, O.; Bar-Noy, A.; Brown, T.; la Porta, T. Sensor-mission assignment in wireless sensor networks. *ACM Trans. Sens. Netw.* **2010**, *6*, 1–33.
162. Mainetti, L.; Patrono, L.; Vilei, A. Evolution of wireless sensor networks towards the Internet of Things: A survey. In Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 15–17 September 2011; pp. 1–6.
163. Gubbi, J.; Buyya, R.; Marusic, S. Internet of Things (IoT): A vision, Architectural Elements and Future Direction. In Proceedings of the International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Bournemouth, UK, 9–11 October 2006; pp. 1–19.
164. Di, M.; Joo, E.M. A survey of machine learning in wireless sensor networks. In Proceedings of the 2007 6th International Conference on Information, Communications & Signal Processing, Singapore, 10–13 December 2007; pp. 1–5.
165. Förster, A.; Forster, A. Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey. In Proceedings of the 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Melbourne, Australia, 3–6 December 2007; pp. 365–370.
166. Tajiki, M.M.; Shojafar, M.; Akbari, B.; Salsano, S.; Conti, M.; Singhal, M. Joint failure recovery, fault prevention, and energy-efficient resource management for real-time SFC in fog-supported SDN. *arXiv* **2018**, arXiv:1807.00324.
167. Tajiki, M.M.; Salsano, S.; Shojafar, M.; Chiaraviglio, L.; Akbari, B. Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining. *arXiv* **2017**, arXiv:1710.02611.
168. Cheng, B.; Cui, L.; Jia, W.; Zhao, W.; Hancke, G.P. Multiple region of interest coverage in camera sensor networks for tele-intensive care units. *IEEE Trans. Ind. Inform.* **2016**, *12*, 2331–2341.
169. Cheng, B.; Zhang, J.; Hancke, G.P.; Karnouskos, S.; Colombo, A.W. Industrial Cyberphysical systems: Realizing cloud-based big data infrastructures. *IEEE Ind. Electron. Mag.* **2018**, *12*, 25–35.