Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques

Article *in* Future Generation Computer Systems · August 2018

DOI: 10.1016/j.future.2018.07.058

CITATIONS
55

READS
5,633

5 authors, including:

Yousaf Bin Zikria
Yeungnam University
105 PUBLICATIONS **1,169** CITATIONS
SEE PROFILE

Heejung Yu
Korea University (Sejong)
118 PUBLICATIONS **1,096** CITATIONS
SEE PROFILE

Muhammad Khalil Afzal
Yeungnam University
59 PUBLICATIONS **1,160** CITATIONS
SEE PROFILE

Mubashir Husain Rehmani
Cork Institute of Technology
199 PUBLICATIONS **6,068** CITATIONS
SEE PROFILE

Future Generation Computer Systems 88 (2018) 699–706

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Editorial

Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques

Yousaf Bin Zikria [a,1], Heejung Yu [a,*], Muhammad Khalil Afzal [b,1], Mubashir Husain Rehmani [c,1], Oliver Hahm [d,1]

[a] Department of Information and Communication Engineering, Yeungnam University, 280 Daehak-Ro, Gyeongsan, Gyeongbuk 38541, Republic of Korea
[b] COMSATS Institute of Information Technology, Pakistan
[c] Waterford Institute of Technology, Ireland
[d] Zuehlke Engineering GmbH, Duesseldorfer Strasse, 65760 Eschborn, Germany

ARTICLE INFO

Keywords:
Internet of Things
Operating systems
Network protocols
Validation

ABSTRACT

By combining energy efficient micro-controllers, low-power radio transceivers, and sensors as well as actuators in so called *smart objects*, we are able to connect the digital cyber world with the physical world as in cyber physical systems. In the vision of the Internet of Things, these smart objects should be seamlessly integrated into the traditional Internet. Typically, smart objects are heavily constrained in terms of computation, memory and energy resources. Furthermore, the commonly used wireless links among smart objects or towards the Internet are typically slow and subject to high packet loss. Such characteristics pose challenges, on one hand in terms of software running on smart objects, and on the other hand in terms of network protocols which smart objects use to communicate. New operating systems, application programming interfaces, frameworks, and middleware have to be designed with consideration of such constraints. In consequence, novel validation methods and experimental tools are needed to study smart object networks in vivo, new software platforms are needed to efficiently operate smart objects, and innovative networking paradigms and protocols are required to interconnect smart objects.

1. Introduction

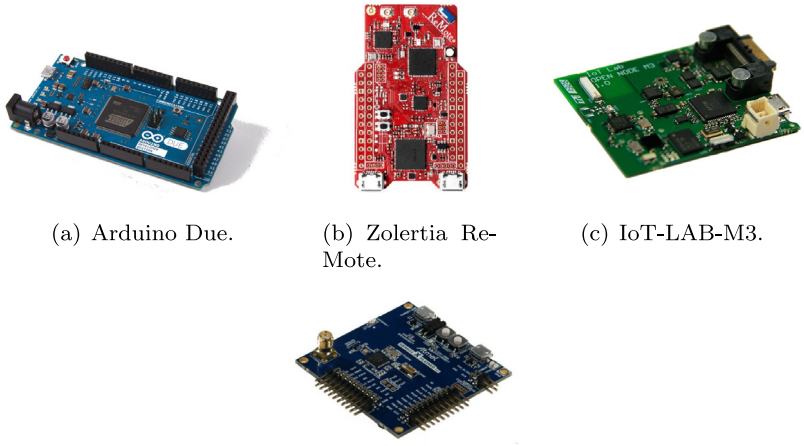
Tiny devices equipped with Microcontrollers (MCUs) and transceivers directly connected to the physical world are often called *smart objects*. The MCUs provide typically memory in the range of kilobytes and usually operate at a speed of some tens of Megahertz. For these devices it is expected that Moore's law will not apply, because Internet of Things (IoT) devices will get rather smaller, cheaper, and more energy-efficient, instead of providing significantly more memory or CPU power [1]. Therefore, in the foreseeable future, low-end IoT devices with a few kilobytes of memory, such as Class 1 and Class 2 devices, are likely to remain predominant in the IoT. The Transceivers often use low-power wireless technologies such as IEEE 802.15.4 or Bluetooth Low Energy. Popular examples of these class of devices include Arduino [2], Zolertia ReMote [3], IoT-LAB M3 nodes [4], OpenMote nodes [5], and TelosB motes [6], some of which are shown in Fig. 1.

The emerging IoT aims to seamlessly integrate these usually resource-constrained, often battery-operated communication devices into the global Internet. As a consequence, the heterogeneity of both hosts and links in the Internet is largely increased. One end of the IoT is composed by these smart objects which directly interact with the physical world, e.g., by controlling engines or sensing the temperature. The other end is composed by powerful servers that can act as the back-end, e.g., by providing a management web interface or a database to store sensor data. Enabling end-to-end connectivity between these devices with a direct coupling to the physical world and services or users in the Internet, not only creates a quantity of new application domains, but also induces a number of new challenges. Smart objects are the key element for many so-called *smart services*. In factory automation, for instance, direct access to physical devices enables much shorter reaction times and collecting much more fine-grained information about the system. In this manner resources can be used more efficiently than in legacy systems where data had to be processed by a central entity or offline. However, the IoT also poses many new challenges for software architectures and network protocols which are required to operate on and between these smart objects. On the one hand, these devices are neither capable of running well-known and mature Operating Systems (OSs) such as Linux or

* Corresponding author.
E-mail addresses: yousafbinzikria@gmail.com (Y.B. Zikria), heejung@yu.ac.kr (H. Yu), khalilafzal@ciitwah.edu.pk (M.K. Afzal), mshrehmani@gmail.com (M.H. Rehmani), oliver.hahm@zuehlke.com (O. Hahm).

[1] All authors contributed equally and should be considered as co-first authors.

https://doi.org/10.1016/j.future.2018.07.058
0167-739X/



(a) Arduino Due. (b) Zolertia Re-Mote. (c) IoT-LAB-M3.

(d) Atmel SAM R21.

Fig. 1. Examples of smart objects.

Berkeley Software Distribution (BSD) nor using traditional Internet protocols such as Internet Protocol version 4 (IPv4) or Hyper Text Transfer Protocol (HTTP). On the other hand, providing standardized Application Programming Interfaces (APIs) and protocols is important to ensure interoperability between different vendors as well as compatibility to existing Internet systems. As a consequence various new OSs have be designed to target the particular use cases of IoT scenarios and provide a unified software platform for heterogeneous devices [7].

2. The evolution of the Internet of Things

The first research area focusing on distributed systems composed by heavily resource-constrained devices connected over unreliable, low-power wireless links was Wireless Sensor Networks (WSNs). This research area emerged from the *Smart Dust* research project, which was started in 1998 at the University of California in Berkeley [8]. For almost two decades, a variety of protocols and networking algorithms, addressing the peculiarities and limitations of these systems, has been designed, implemented, and evaluated. A plethora of articles analyzing theoretical models, discussing results from simulations, and testbed driven research has been published [9–11]. At the same time, researchers in this area developed the necessary tools and software components to examine these systems. Along with the implementation of protocols, algorithmic frameworks, simulators and emulators, or experimentation libraries, a variety of OSs and middleware for typical WSN use cases has been implemented [12–14].

Over time proprietary protocol stacks, such as ZigBee or WirelessHART, have emerged as default solutions in commercial scenarios [15]. Typical use cases for WSNs comprise agriculture control, disaster prevention systems, wildlife or structural health monitoring [16]. Convergecast is the typical communication pattern in most of these WSN scenarios where sensor values are forwarded towards a single data sink, e.g., a more powerful gateway node connected to the Internet or a database. Consequently, each WSN deployment is typically controlled by a single entity through the gateway. This entity also has full governance over the network and data inside the WSN. Typically, these networks are deployed once and further nodes are added only to replace malfunctioning ones or to react to changes in the task dynamics [17].

2.1. New hardware and new business models

In the meantime, not only the algorithms and protocols reached a certain level of practicability, but also the hardware in this area has evolved to fulfill the requirements of industrial and commercial deployments:

- MCUs that are not only small and energy efficient enough, but are also available for a very low price.
- Digital radio transceivers that are comparatively simple to program and provide energy-saving features.
- A huge variety of cheap sensors to measure a wide range of physical properties.

However, it is not expected that new generations of hardware in this area will relax computational or memory constraints: IoT devices are rather expected to get smaller, cheaper, and more energy-efficient [1].

The innovations in hardware design and algorithms have enabled the emergence of many new business models. Additional to the traditional WSN deployments, multiple new use cases such as smart building, smart homes, industrial automation, smart metering, and smart grid have been identified for potential business strategies. Moreover, operating areas with very different requirements such as health care, aerospace industry, or city governments became potential beneficiaries of these technologies.

2.2. New opportunities and challenges due to heterogeneity of IoT

All these commercial, industrial, and governmental use cases have also created the requirements for standardization efforts. On the one hand, it has become increasingly important that devices from different vendors are interoperable on various layers of the network stack. On the other hand, a similar need for compatibility has arisen for software components. Gradually, it has become clear that there is a need for natural and seamless interconnection to the worldwide network infrastructure: the vision of the IoT emerged [18]. Consequently, different standardization bodies such as the Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), or Open Mobile Alliance (OMA) have tackled this task and released corresponding protocol specifications. The IETF, for instance, has introduced a set of standards: IPv6 over IEEE 802.15.4 networks (6LoWPAN) [19,20] (an adaptation layer which compacts long IPv6 headers so they fit in short frames typical for sensor networks as in IEEE 802.15.4), RPL [21] (a routing protocol) and CoAP [22] (an application-layer protocol allowing low-power devices to appear as web servers).

These emerging standards also function as a catalyst for more commercial use cases. Companies which refrained from using these new technologies got attracted by the availability of standard

solutions. Networking systems with Internet Protocol (IP) at the narrow waist of the protocol stack has proven to work fine for a large variety of use cases over the last 40 years. Consequently, using IP suite protocols to allow for end-to-end connectivity between low-end devices on the one side and traditional Internet services at the other side enabled a whole set of new business cases.

In contrast to WSNs scenarios, IoT deployments comprise a much larger heterogeneity not only in terms of hardware and link layer technologies, but also in terms of network configurations and applications. In comparison to WSNs where separated networks, which are tailored for one particular use case, are deployed, IoT applications are composed of several components that are supposed to seamlessly work together either locally in an ad-hoc manner between these IoT devices, interconnected through full-fledged wired backbone technologies, or even through cloud services. IoT deployments are furthermore supposed to be deployed for a long and often undetermined time-span. Here, the size of WSN deployments is usually not subject to bigger changes [17], IoT deployments are expected to be gradually extended and updated over time. Therefore, a need for standards, such the Internet protocol suite, is inevitable.

3. IoT use cases

The IoT comprises a wide range of professional use cases [23–25]. In contrast to WSN scenarios, these use cases comprise many different communication and traffic patterns [26]. Therefore, it is important to take a closer look at some use cases for IoT applications in order to understand the variety of scenarios and derive the particular requirements. Typical use cases for IoT applications are as follows.

- **Industrial Automation**
  Steel mills, oil refineries, chemical industries, or power plants are examples for industrial settings where complex monitoring and management processes occur. Sensing data such as temperature, pressure, vibration, or tank fill levels are used to control actuators and coordinate production stages often by thousands of nodes.
  The particular requirements of these applications for industrial networks in contrast to the traditional Internet, have lead to completely decoupled development of technologies, protocols, and standards. While the Internet is built to interconnect billions of heterogeneous devices communicating globally large amounts of data, an industrial network is typically deployed within a factory floor, typically connecting hundreds or thousands of devices. In many cases, the amount of traffic and content data in industrial applications are not very large, but reliability, dependability, and deterministic latency are often mandated.
- **Building Automation and Heating, Ventilation, and Air Conditioning (HVAC)**
  Nowadays, many buildings are equipped with a variety of controllers connected to electronic or pneumatic elements. These building management systems consist of a network of sensors, actuators, controllers, and user interface devices that interoperate to provide a safe and comfortable environment while constraining energy costs. Typical applications in this area are HVAC, room lighting, window shades, solar loads, physical security, fire detection, and elevator or lift systems. They can often be found in government facilities, pharmaceutical manufacturing facilities, hospitals, or office buildings ranging in size from 10 000 to more than 100 000 m^2. The variety of sensors range from common temperature, lighting, and humidity sensors to specialized air flow and pressure or CO_2 sensors.
- **Smart Metering, Advanced Metering Infrastructure (AMI)**
  The demand for electricity is continuously increasing,[2] while the electrical infrastructure is mostly outdated and not equipped for automated analysis or fast response time. Additional challenges are caused by a growing population, global climate change, equipment failures, energy storage problems, and resilience problems. Consequently, a new smart grid infrastructure is urgently needed to address these challenges. A smart grid is an infrastructure aiming for improved efficiency, reliability, and safety. Additionally, it should enable a smooth integration of renewable and alternative energy sources. These properties should be achieved through automated control technologies. Renewable energy generators need to be effectively integrated by enhanced sensing and communication capabilities.
- **Mobile Health**
  In mobile health applications the goal is typically to monitor a patient with various devices [27]. These devices are equipped with one or more sensors to measure, for example, the blood pressure or the heart rate. Typical applications are hearing aids, drug dosage, Electrocardiogram (ECG), Electroencephalogram (EEG), temperature, respiration, or glucose monitoring. Mobile health applications can help to reduce the exploding costs for health care, caused by an aging population and sedentary life style. At the same time, they can help to improve the quality of living for sick or handicapped people.

4. Requirements, constraints, and challenges for software, network protocols, and validation techniques

4.1. IoT software

Low-end IoT devices are typically very constrained in terms of resources including energy, CPU, and memory capacity. As a consequence of these constraints and general requirements for IoT use cases, the software for these devices has to fulfill the following requirements:

- **Energy Efficiency**
  Many IoT devices will run on batteries or other constrained energy sources. Energy efficiency is also required to effectively support a massive number of IoT devices that are expected to be deployed. IoT hardware including MCUs, radio transceivers and sensors provides features to operate in an energy-efficient manner. Hence, it is crucial for IoT software to leverage the low power modes of the hardware and put the device to a sleep mode for as long as possible.
- **Small Memory Footprint**
  Compared to other connected machines, IoT devices are much more resource-constrained, especially in terms of memory. One of the requirements for IoT software is thus to fit within such a memory constraint. While PCs, smartphones, tablets, or laptops provide Giga- or Terabytes of memory, IoT devices typically provide a few kilobytes of memory, i.e. a million times less. In order to fit within a memory footprint constraint, IoT application designers must be provided with a set of optimized libraries (potentially cross-layer) with common IoT functionality and efficient data structures.
- **Support for Heterogeneous Hardware**
  While the diversity of hardware and protocols used in today's Internet is relatively small from an architectural perspective,

[2] The U.S. Department of Energy reported that the demand and consumption of electricity increased by 2.5% annually over the last twenty years.

the degree of heterogeneity explodes in the IoT. The wide variety of use cases [28–32] leads to the development of a wide variety of hardware and communication technologies. IoT devices are based on various MCU architectures and families, including 8-bit (e.g., Intel 8051/52, Atmel AVR), 16-bit (e.g. TI MSP430), 32-bit (ARM7, ARM Cortex-M, MIPS32, and even x86) architectures. 64-bit architectures might also appear in the future [33].

- **Network Connectivity**
  The ability to communicate and interact at a local and global scale, is the key feature for IoT. IoT devices are thus typically equipped with at least one network interface. Communication techniques used in the IoT encompass not only a wide variety of low-power radio technologies, e.g., IEEE 802.15.4, Bluetooth/BLE, DASH7, and EnOcean, but also various wired technologies, e.g., Power Line Communication (PLC), Ethernet, and several bus systems. The combination of (i) having to support multiple link layer technologies and (ii) having to communicate with other Internet hosts, leads to the use of network stacks based on IP protocols directly on IoT devices [34–36].
- **Interoperability**
  The tremendous amount of IoT use cases and their vastly diverging requirements make it impossible to be covered by single software project. Hence, it becomes inevitable to (i) reuse existing software components as much as possible and (ii) provide the well-defined interfaces in order to ease integration of third-party software components. Consequently, it is desirable for an IoT OS to comply with the well-known system and programming language standards such as POSIX or ANSI C. Moreover, the OS needs to support mechanisms to use third-party code similar to package systems in Linux distributions or ports in BSD.
- **Real-Time Capabilities**
  Precise timing, and timely execution are crucial in various IoT use-cases, e.g., smart health applications such as body area networks (BAN) with pacemakers providing wireless monitoring and control [37,38], Vehicular Ad-Hoc Network (VANET) and other scenarios including actuators and/or robots in industrial automation contexts. An OS that can fulfill timely execution requirements is called a Real-Time Operating System (RTOS), and is designed to guarantee worst-case execution time and worst-case interrupt latency. Therefore, another requirement for a generic OS for the IoT is to be an RTOS, which typically implies that kernel functions have to operate with deterministic run-time. Tasks have to meet certain deadlines in order to work correctly.
- **Security and Safety**
  The unified IoT platform makes physical objects accessible to applications across organizations and domains. On one hand, some IoT systems are part of critical infrastructure or industrial systems with life safety implications [39]. On the other hand, since they are connected to the Internet, IoT devices are in general expected to meet high security and privacy standards. Thus, a requirement (and challenge) for IoT OSs is to provide the necessary mechanisms, e.g., cryptographic libraries and security protocols, while retaining flexibility and usability. Last but not least, because software with a certain degree of complexity can never be expected to be 100% bug-free and security standards evolve (driven by various stake holders such as industry, government, consumers etc.), it is crucial to provide mechanisms for software updates on already-deployed IoT devices and to use open source as much as possible [40].

4.2. IoT network protocols

The term Low-power and Lossy Network (LLN) was proposed in the context of the IETF ROLL working group and formally defined in RFC7102 [41] as follows:

> Typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4 or low-power Wi-Fi. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (HVAC, lighting, access control, fire), connected home, health care, environmental monitoring, urban sensor networks, energy management, assets tracking, and refrigeration.

In the IETF context, the terms of "Low-power Wireless Area Networks", "Networks of Resource Constrained Nodes", and "Constrained Node Networks"[3] are used in a similar sense.

The common properties of these networks are as follow.

- **Low Bandwidth**
  Highly energy-efficient transceivers, slow MCUs, and wide coverage of IoT networks lead to very low data rates in the range of 200 kbit/s or even significantly below.
- **Small Packet Sizes**
  Short packet size is inevitable due to collected data size by IoT devices, constraints on memory and energy and the need to share the wireless medium among a massive number of nodes.
- **High Packet Loss Rate**
  Due to the low-power transmissions and general challenges of wireless communication, such as a large number of contending IoT transmitters, interference especially in unlicensed band, and multipath fading, low packet loss rate cannot be guaranteed as in the other wireless networks like cellular systems.
- **Omni-directional Transmissions**
  Most of the time IoT radio transceivers transmit in an omnidirectional manner with a single antennas. Multicast (if available) and unicast can only be filtered after reception of the destination address field. This can easily lead to congestion on the link layer.

It can be observed that these properties are partially induced by the wireless medium, partially by the need for maximum energy efficiency, and partly by the constraints of the nodes. Despite all these limitations and constraints, many of the IoT use cases aim for reliability and resource availability.

As a consequence of these properties in combination with the general requirements of IoT use cases, we can derive the following requirements for the design of IoT network protocols:

- Interoperability
- Energy Efficiency
- Reduced Control and Data Traffic
- Bounded Latency
- Robustness
- Security and Privacy

4.3. Validation techniques for IoT

For a long time the use and deployment of many well-known standards have been considered infeasible for low-end embedded

[3] More precisely, LLNs could be described as the combination of a Constrained Node Network and a Constrained Network.

devices both in the system side and in the networking side. Even with the development of the first IP stacks for these systems in the early days of WSNs, it was considered rather a scientific gimmick than a realistic use case. However, with the latest evolution of IoT systems and increasing relevance in commercial and governmental projects, the needs for standards have been increasing. Interoperability and reusability are mandatory. Interoperability events, like the so-called *plugtests* organized by European Telecommunications Standards Institute (ETSI) play an important role in networking standardization. These events are important for both, the specification developers and the implementers. While the latter can verify the interoperability of their implementations, the specification developers can check the unambiguity of their documents.

Furthermore, conducting experiment-driven research is inevitable to study software and protocol design for IoT use cases. Experiments can be used to verify (or disprove) perceived insights gained from theoretic models or simulations. Moreover, results from these experiments can serve to generate valid input parameters for model or simulation driven research.

Many different tools are required for experiment-driven research on IoT systems. Besides the implementation of the examined approach itself, a software platform (e.g. an OS) to operate the IoT devices, frameworks or middleware software, and tools to schedule, execute, control, and evaluate the experiment are necessary.

5. Brief review of articles in special issue

In this special issue, we have published papers in the domain of IoT, Cyber-physical Systems, Industry 4.0, Smart Home, and Building Automation. The various issues in IoT systems, networks and applications have been covered in the papers.

IoT is comprised of heterogeneous communication and network technologies. These includes short and long range communications, static and dynamic topologies, low-rate and high-rate communications. However, the coexistence of these technologies and the network resource management are still challenging issues in IoT networks. Software Defined Networking (SDN) makes the network easy to configure and deploy networks with the main advantage of separating between control and data plans. Bendouda et al. [42] have handled the heterogeneity of wireless network technologies in the IoT context by proposing a new hybrid and programmable architecture-based on SDN paradigm with the aim to reduce the overhead related to network control mechanism. In order to make the proposed architecture adaptable to different network technologies, this paper has introduced three levels of controls: principal, secondary, and local controllers. Connected Dominating Set (CDS) and fuzzy set approaches are used to select local controller. 36% improvement is shown by the proposed algorithm.

Pervasive sensing (PS) is known as collecting huge amount of data from smart instruments and relaying this data to a base-station which gathers and analyzes the data. PS is one of the great examples that use low cost sensory devices in mobile devices. Al-Turjman [43] has proposed a framework for PS in smart cities based on an IoT architectural model that integrates heterogeneous networks and data sources to support the smart grid project.

IoT systems have huge volume of data which is collected from anywhere at any time. It raises issues of privacy in medical systems or daily living environments. It has been observed that the end-devices with weaker identity have limited computational capabilities. However, the weakest type of devices in IoT system, like a smart bulb or smoke detector, is powered by batteries; they do not support any conventional security mechanisms. Privacy is an essential task and higher privacy demands usually tend to require weaker identity. Furthermore, security tends to demand strong identity, especially in authentication processes. This motivates developers to develop a privacy-preserving and accountable authentication protocol for IoT end-devices. Wang [44] proposed privacy-preserving and accountable authentication protocols for IoT devices with weaker identity. Construction of short group signature and Shamir's secret sharing scheme has been implemented as well. However, this protocol enables end-devices with weaker identity can still be authenticated with the aid of an edge layer which helps end devices. In the authentication processes, their essential identity information (strong identity) will not be disclosed and it achieves privacy-preserving. Moreover, if an end-device with weaker identity interrupts critical instruction, the edge layer will help to trace, block, and handle it. This property is called accountability. Security properties of the proposed protocol have been analyzed in the context of six typical attacks. Experiments results and performance analysis show that the proposed protocol is feasible in practice.

In [45], novel solutions have been introduced to create and deliver smart cities more effectively. Cost reduction, safe environment, convenient and friendly applications can be accessed through a system that can take advantages of all the technology's capabilities. IoT offers new types of services to improve daily life. Consequently, other recently developed technologies such as Big Data, Cloud Computing and Monitoring can be integrated with IoT. In this proposed system, Plageras et al. have investigated four advanced technologies to find and combine their common operations. After the smart city border concept, they propose new systems for collecting and managing sensor data in a smart building that works in IoT environments. Data is taken from a remote control and remote device management (mobile) networks. The proposed solution can be used to collect and synthesize sensor data in a smart, energy-efficient building. Multiple sensors are installed in smart building, to get a better monitoring. Plageras et al. used Contiki-OS Cooja to validate the system. The proposed solution is effective in reducing the energy consumption.

Today's electrical grids are currently transformed into Smart Grid (SG). Therefore, SG has become a topic of intensive research, development, and deployment over the last few years. Harsh and complex SG environments pose great challenges to guarantee reliable communication for wireless sensor based SG applications due to equipment noise, electromagnetic interference and multipath fading effects in SG environments. To address these challenges, Faheem and Gungor [46] have proposed a novel multi-mobile sinks-based quality of service aware data gathering protocol for wireless sensors based SG applications. The proposed scheme significantly improves the quality of service performance metrics for wireless sensor network based SG applications and is suitable for both sparse and large network deployment. The extensive performance evaluations show that the protocol has successfully achieved its defined goals in terms of packet delivery ratio, packet error rate, end-to-end delay, throughput, memory utilization, control message overhead and energy efficiency compared to existing routing schemes.

The health-care has been considered as one of the key applications of IoT. Wearable IoT devices with medical sensors gather and deliver a large amount of data, i.e., Big Data. To meet requirements of health-care Big Data systems, Manogaran et al. [47] have proposed a new architecture for the implementation of IoT to store and process scalable sensor data for health-care applications. The proposed architecture consists of two sub architectures, i.e., Meta Fog-Redirection (MF-R) and Grouping and Choosing (GC) architectures. The MF-R architecture uses Big Data technologies for collection and storage of the data generated from sensor devices. The GC architecture includes a key management scheme to protect the data in the cloud against unauthorized access, and the data

categorization function (Sensitive, Critical and Normal) for providing security services.

Modern smart cities use a number of sensors, which are used for applications to store and analyze large amounts of data in real time. The study of [48] related to the use of data collected from local energy management systems. Current Home Energy Management Systems (HEMS) aims for energy efficiency, however, residents typically value indoor comfort as well. Predicted Mean Vote (PMV) is known as the short-term indoor comfort which is used by HEMS to collect various data. The proposed solution determines residents' indoor comfort preferences using PMV from environmental and physical data and selects suitable information from this collected data to be displayed on a web page for users. The proposed system was examined in three homes in Japan for 12 days during winter. The proposed system reduced electricity consumption by 5.15% and increased the comfort satisfaction by 42.3%. Furthermore, qualitative assessment of indoor comfort increased by 16.4%. Providing users with information selected according to their PMV preferences was more effective in reducing electricity consumption and increasing indoor comfort than providing them with random information. The proposed system collected data using networked sensors, however, in the future HEMS data may be stored in a central database operated by a smart city or a system itself. The great potential of HEMS data is to reduce electricity consumption and maintain indoor comfort may contribute to the quality of life, which is one of the fundamental idea behind the smart city.

In IoT networks with massive number of smart objects, enormous data is generated from the sensors and lots of critical decisions are taken with this data. Hence, the accuracy, correctness and integrity of the data are the inevitable requirements for IoT. Previously, outliers regarded as malfunctioning sensors but an outlier may be an important event that should not be neglected. Therefore, Nesa et al. [49] have proposed the outlier detection method based on a sequence learning approach. The three layer architecture of the outlier detection and the efficient non-parametric supervised scheme, which evaluates initial records and processes based on Influential Relative Grade (IRG) and their Relative Mass Functions (RMF), have been proposed. For performance verification, simulation tests have been performed with three different categories of datasets, and shown high accuracy of error and event detection performance.

Telemedicine has been significant because of its ability to provide health-care services to remote locations. Therefore, telemedicine implies concepts of WSNs and the IoT. In [50], IoT based Health Prescription Assistant (HPA) model has been proposed based on telemedicine. This system attains a goal for each patient to follow the doctor's recommendations properly. Moreover, it designs a security system that ensures user authentication and protected access to resources and services. Furthermore, an access control mechanism is implemented to prevent unauthorized access to medical devices. The system requires verification for authentication. After it is successful, the user will be issued an authorization Security Access Token (SAT) ticket by a medical IoT device which is cryptographically protected to guard against forgery. SAT grants the access to medical IoT devices and their services and/or resources. The system determines that the protected access to IoT services and resources can be achieved by adopting a context-aware capability-based access control model. Furthermore, this system depicts a two-phase process comprising of an authentication phase and an SAT-generation phase. The proposed system provides a synopsis of the security component such as the identity provider, the registration authority, and the IoT service provider. The experimental results show that the delegation-based SAT verification approach outperforms the distributed approach in terms of both communication and computation latency. Furthermore, the paper uncovered that the delegation-based SAT verification approach is appropriate for heavy resource constrained medical IoT devices. According to the results, this approach is more energy-efficient than the distributed approach. A prototype of the proposed system has been implemented to experimentally analyze and compare the resource efficiency of different SAT verification approaches in terms of various performance metrics, including computation and communication overhead.

As Big Data is exponentially growing, the needs for efficient storage, processing, and retrieval of massive data sets generated from different applications have continuously increased. The biggest challenge for the research community is handling of these datasets due to the involved heterogeneity in their formats. It has been observed that Probabilistic Data Structures (PDS) are suitable for large-scale data processing, approximate predictions, fast retrieval and unstructured data storage. A real-time in-stream data demands time-bound query output in a single pass. Therefore, a solution [51] has been proposed as Accommodative Bloom Filter (ABF). ABF is a variant of the scalable bloom filter, where the insertion of bulk data is done by using the addition of new filters vertically. Data generated from various sensors has been considered for experimental purposes where query processing is done at two levels to improve the accuracy and reduce the search time. It has been found that insertion and search time complexity of ABF does not increase with the number of elements. Furthermore, the results indicate that ABF outperforms the existing variants of Bloom filters in terms of false positive rates and query complexity, especially when dealing with in-stream data. Theoretical and experimental results demonstrate that the proposed ABF shows better performance and scalability with fewer storage requirements and less query time.

In IoT networks, security is one of the most significant issues in addition to handling of the massive number of connected devices and resource allocation to these devices. In the IoT applications including smart city and smart grid, public information requiring high level of security has been delivered. To prevent eavesdropping of such critical information, therefore, security of the network should be guaranteed. Especially for IoT devices with the constraints of resources, the performance of data transfer to the cloud as well as end-to-end security should be robust and reliable. In [52], Mukherjee et al. have developed an end-to-end security middleware between edge devices and a core cloud of IoT application systems. The proposed middleware in the cloud-fog communication platform is based on a flexible security scheme tailored to application needs. Additionally, the intermittent security copes with unreliable network connections.

Named Data Networking (NDN) has been considered as a promising information-centric networking architecture due to its content-driven networking operations, e.g., in-network caching, receiver-driven, multicast, and name-based routing features. Therefore, the NDN architecture has been considered as a network layer solution to cope with the IoT evolution challenges. Meddeb et al. [53] have shown the potential of NDN to support the IoT systems and focused on the producer mobility issue. In detail, the Adaptive Forwarding Based Link Recovery for Mobility Support (AFIRM) algorithm, which is a content-driven, adaptive forwarding, and fully distributed approach, has been proposed to handle the producer mobility issue in NDN/IoT networks. Through simulations, the feasibility of the proposed approach has been verified. Furthermore, the proposed algorithm outperforms other relevant solutions such as flooding and the Content-driven Bloom Filter Based Routing Algorithm (COBRA) algorithms and it can reduce the packet loss due to the mobility of sensors as well as the signaling cost.

6. Conclusion

Twelve papers in this special issue reflect state-of-art research trends in IoT systems and applications by highlighting key challenges and proposing novel advanced architectures. This special issue covered various topics including IoT applications, network architectures, big data and so on. The guest editors would like to show our appreciation to all the authors and reviewers for their constructive and valuable contributions to this special issue. We would like also to thank Peter Sloot, Editor-in-Chief, for his invaluable help and productive advice in preparing this special issue.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2017R1D1A1B03030757) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

[1] L. Mirani, Chip-makers are betting that Moore's law won't matter in the Internet of Things, 2014. http://qz.com/218514.
[2] Arduino Due. URL http://arduino.cc/en/Main/arduinoBoardDue.
[3] Zolertia, Z1 Datasheet. URL http://www.zolertia.com/.
[4] IoT-LAB: Very large scale open wireless sensor network testbed, 2016. https://www.iot-lab.info/hardware/m3/. URL https://www.iot-lab.info/hardware/m3/.
[5] OpenMote, OpenMote-CC2538. URL http://www.openmote.com/hardware/openmote-cc2538-en.html.
[6] MoteIV Corporation, Telos — Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module, Datasheet. URL http://www.willow.co.uk/html/telosb_mote_platform.php.
[7] O. Hahm, E. Baccelli, H. Petersen, N. Tsiftes, Operating systems for low-end devices in the Internet of Things: a survey, IEEE Internet Things J. 3 (5) (2016) 720–734. http://dx.doi.org/10.1109/JIOT.2015.2505901.
[8] J.M. Kahn, R.H. Katz, K.S. Pister, Next century challenges: mobile networking for Smart Dust, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM Press, 1999, pp. 271–278.
[9] H. Karl, A. Willig, Protocols And Architectures for Wireless Sensor Networks, John Wiley & Sons, 2007.
[10] S. Khan, A.-S.K. Pathan, N.A. Alrajeh, Wireless Sensor Networks: Current Status and Future Trends, CRC Press, 2016.
[11] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660. http://dx.doi.org/10.1016/j.future.2013.01.010, URL http://www.sciencedirect.com/science/article/pii/S0167739X13000241.
[12] M. Moubarak, M.K. Watfa, Embedded operating systems in wireless sensor networks, in: Guide to Wireless Sensor Networks, Springer, 2009, pp. 323–346.
[13] P. Rawat, K.D. Singh, H. Chaouchi, J.M. Bonnin, Wireless sensor networks: a survey on recent developments and potential synergies, J. Supercomput. 68 (1) (2014) 1–48.
[14] Y.B. Zikria, R. Alil, R. Bajracharya, H. Yu, S.W. Kim, IoT theoratical to practical: Contiki-os and Zolertia remote, Far East J. Electron. Commun. 17 (4) (2017) 915–921. http://dx.doi.org/10.17654/EC017040915, URL http://dx.doi.org/10.17654/EC017040915.
[15] V.C. Gungor, G.P. Hancke, Industrial wireless sensor networks: Challenges, design principles, and technical approaches, IEEE Trans. Ind. Electron. 56 (10) (2009) 4258–4265.
[16] I.F. Akyildiz, M.C. Vuran, Wireless Sensor Networks, John Wiley & Sons, 2010.
[17] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[18] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, J. Vasseur, M. Durvy, A. Terzis, A. Dunkels, D. Culler, Beyond interoperability – Pushing the performance of sensor network IP stacks, in: Conference on Embedded Networked Sensor Systems, SenSys, ACM, 2011.
[19] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944 (Proposed Standard), Sep. 2007. URL http://www.ietf.org/rfc/rfc4944.txt.
[20] J. Hui, P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, RFC 6282 (Proposed Standard), Sep. 2011. URL http://www.ietf.org/rfc/rfc6282.txt.
[21] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550 (Proposed Standard), Mar. 2012. URL http://www.ietf.org/rfc/rfc6550.txt.
[22] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol (CoAP), RFC 7252 (Proposed Standard), Jun. 2014. URL http://www.ietf.org/rfc/rfc7252.txt.
[23] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660. http://dx.doi.org/10.1016/j.future.2013.01.010, URL http://www.sciencedirect.com/science/article/pii/S0167739X13000241.
[24] H. Yu, H. Lee, H. Jeon, What is 5G? Emerging 5G mobile services and network requirements, Sustainability 9 (10) (2017) 1848. http://dx.doi.org/10.3390/su9101848, URL http://www.mdpi.com/2071-1050/9/10/1848.
[25] T. Kim, C. Ramos, S. Mohammed, Smart City and IoT, Future Gener. Comput. Syst. 76 (2017) 159–162. http://dx.doi.org/10.1016/j.future.2017.03.034, URL http://www.sciencedirect.com/science/article/pii/S0167739X17305253.
[26] H. Tschofenig, J. Arkko, D. Thaler, D. McPherson, Architectural considerations in smart object networking, RFC 7452 (Informational), Mar. 2015. URL http://www.ietf.org/rfc/rfc7452.txt.
[27] A. Pantelopoulos, N.G. Bourbakis, A survey on wearable sensor-based systems for health monitoring and prognosis, IEEE Trans. Syst. Man Cybern. C 40 (1) (2010) 1–12. http://dx.doi.org/10.1109/TSMCC.2009.2032660.
[28] M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing requirements for urban low-power and Lossy networks, RFC 5548 (Informational), May 2009. URL http://www.ietf.org/rfc/rfc5548.txt.
[29] J. Martocci, P.D. Mil, N. Riou, W. Vermeylen, Building automation routing requirements in low-power and Lossy networks, RFC 5867 (Informational), Jun. 2010. URL http://www.ietf.org/rfc/rfc5867.txt.
[30] K. Pister, P. Thubert, S. Dwars, T. Phinney, Industrial Routing requirements in low-power and Lossy networks, RFC 5673 (Informational), Oct. 2009. URL http://www.ietf.org/rfc/rfc5673.txt.
[31] A. Brandt, J. Buron, G. Porcu, Home automation routing requirements in low-power and Lossy networks, RFC 5826 (Informational), Apr. 2010. URL http://www.ietf.org/rfc/rfc5826.txt.
[32] K. Rose, S. Eldridge, L. Chapin, The Internet of Things: An Overview, Oct. 2015. URL https://www.internetsociety.org/resources/doc/2015/iot-overview.
[33] S. Evanczuk, The most-popular MCUs ever, Aug. 2013. URL https://web.archive.org/web/20140703062337/http://edn.com/electronics-blogs/systems-interface/4419922/Slideshow--The-most-popular-MCUs-ever.
[34] M.R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L.A. Grieco, G. Boggia, M. Dohler, Standardized protocol stack for the internet of (important) things, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1389–1406.
[35] Y.B. Zikria, M.K. Afzal, F. Ishmanov, S.W. Kim, H. Yu, A survey on routing protocols supported by the Contiki Internet of things operating system, Future Gener. Comput. Syst. 82 (2018) 200–219. http://dx.doi.org/10.1016/j.future.2017.12.045, URL http://www.sciencedirect.com/science/article/pii/S0167739X17324299.
[36] A. Musaddiq, Y.B. Zikria, O. Hahm, H. Yu, A.K. Bashir, S.W. Kim, A survey on resource management in IoT operating systems, IEEE Access 6 (2018) 8459–8482. http://dx.doi.org/10.1109/ACCESS.2018.2808324.
[37] A. Milenković, C. Otto, E. Jovanov, Wireless sensor networks for personal health monitoring: Issues and an implementation, Comput. Commun. 29 (13) (2006) 2521–2533.
[38] B. Hughes, R. Meier, R. Cunningham, V. Cahill, Towards real-time middleware for vehicular ad hoc networks, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET'04, ACM Press, 2004, pp. 95–96. http://dx.doi.org/10.1145/1023875.1023894, URL http://doi.acm.org/10.1145/1023875.1023894.
[39] K.A. Stouffer, J.A. Falco, K.A. Scarfone, SP 800-82 Guide To Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations Such As Programmable Logic Controllers (PLC), Tech. Rep, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2011.
[40] J.-H. Hoepman, B. Jacobs, Increased security through open source, Commun. ACM 50 (1) (2007) 79–83. http://dx.doi.org/10.1145/1188913.1188921, URL http://doi.acm.org/10.1145/1188913.1188921.
[41] J. Vasseur, Terms used in routing for low-power and Lossy networks, RFC 7102 (Informational), Jan. 2014. URL http://www.ietf.org/rfc/rfc7102.txt.
[42] D. Bendouda, A. Rachedi, H. Haffaf, Programmable architecture based on software defined network for Internet of Things: connected dominated sets approach, Future Gener. Comput. Syst. 80 (2018) 188–197. http://dx.doi.org/10.1016/j.future.2017.09.070, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314590.

[43] F. Al-Turjman, Mobile couriers' selection for the smart-grid in smart-cities' pervasive sensing, Future Gener. Comput. Syst. 82 (2018) 327–341. http://dx.doi.org/10.1016/j.future.2017.09.033, URL http://www.sciencedirect.com/science/article/pii/S0167739X17310737.

[44] Z. Wang, A privacy-preserving and accountable authentication protocol for IoT end-devices with weaker identity, Future Gener. Comput. Syst. 82 (2018) 342–348. http://dx.doi.org/10.1016/j.future.2017.09.042, URL http://www.sciencedirect.com/science/article/pii/S0167739X17307495.

[45] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B. Gupta, Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings, Future Gener. Comput. Syst. 82 (2018) 349–357. http://dx.doi.org/10.1016/j.future.2017.09.082, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314127.

[46] M. Faheem, V. Gungor, MQRP: Mobile sinks-based QoS-aware data gathering protocol for wireless sensor networks-based smart grid applications in the context of industry 4.0-based on internet of things, Future Gener. Comput. Syst. 82 (2018) 358–374. http://dx.doi.org/10.1016/j.future.2017.10.009, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314541.

[47] G. Manogaran, R. Varatharajan, D. Lopez, P.M. Kumar, R. Sundarasekar, C. Thota, A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system, Future Gener. Comput. Syst. 82 (2018) 375–387. http://dx.doi.org/10.1016/j.future.2017.10.045, URL http://www.sciencedirect.com/science/article/pii/S0167739X17305149.

[48] K. Matsui, An information provision system to promote energy conservation and maintain indoor comfort in smart homes using sensed data by IoT sensors, Future Gener. Comput. Syst. 82 (2018) 388–394. http://dx.doi.org/10.1016/j.future.2017.10.043, URL http://www.sciencedirect.com/science/article/pii/S0167739X17313559.

[49] N. Nesa, T. Ghosh, I. Banerjee, Non-parametric sequence-based learning approach for outlier detection in IoT, Future Gener. Comput. Syst. 82 (2018) 412–421. http://dx.doi.org/10.1016/j.future.2017.11.021, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314474.

[50] M. Hossain, S.R. Islam, F. Ali, K.-S. Kwak, R. Hasan, An Internet of Things-based health prescription assistant and its security system design, Future Gener. Comput. Syst. 82 (2018) 422–439. http://dx.doi.org/10.1016/j.future.2017.11.020, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314085.

[51] A. Singh, S. Garg, S. Batra, N. Kumar, J.J. Rodrigues, Bloom filter based optimization scheme for massive data handling in IoT environment, Future Gener. Comput. Syst. 82 (2018) 440–449. http://dx.doi.org/10.1016/j.future.2017.12.016, URL http://www.sciencedirect.com/science/article/pii/S0167739X17314516.

[52] B. Mukherjee, S. Wang, W. Lu, R.L. Neupane, D. Dunn, Y. Ren, Q. Su, P. Calyam, Flexible IoT security middleware for end-to-end cloud-fog communication, Future Gener. Comput. Syst. http://dx.doi.org/10.1016/j.future.2017.12.031, URL http://www.sciencedirect.com/science/article/pii/S0167739X17311470.

[53] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, S. Gannouni, AFIRM: adaptive forwarding based link recovery for mobility support in NDN/IoT networks, Future Gener. Comput. Syst. (2018).