A Distributed Systems Perspective on Industrial IoT

Konrad Iwanicki
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw, Poland
E-mail: iwanicki@mimuw.edu.pl

Abstract—**Industrial Internet of Things (IoT) is frequently mentioned as one of the emerging areas in computing that may have a high potential real-world impact in the coming decade. In this paper, we analyze the challenges posed and opportunities offered by industrial IoT solutions from the distributed systems perspective. We focus on the sensing and actuation layer, which results from the the tight coupling of such solutions with the physical objects they monitor and control. We analyze this layer with respect to interoperability, scalability, and dependability, which are key features of many distributed systems.**

I. Introduction

Starting from 2012, the IEEE Computer Society publishes annually a list of about ten key trends in computing technology that are expected to reach adoption in the following year(s). In particular, the prediction for 2018 includes deep learning, digital currencies, technologies referred to as "blockchain," industrial Internet of Things (IoT), robotics, assisted transportation, augmented and virtual reality, hardware for computing accelerators, artificial intelligence for cybersecurity, and, finally, ethics, laws, and policies for privacy, security, and liability [1]. Interestingly, two years ago, the accuracy of those predictions started being evaluated, with the overall grades so far equal to B+ in 2016 [2] and A- in 2017 [3], which may suggest that the forecasts are at least somewhat reliable. In any case, for 2018, the same technologies appear in a number of similar predictions made by other entities and individuals. Therefore, let us assume that the enumerated trends are indeed what will drive computing in the nearest future.

The trend that has received exceptional recognition in the aforementioned predictions—being listed repeatedly from the beginning—is IoT and the various facets thereof. The core idea behind IoT is embedding tiny, networked, electronic devices into the surrounding physical objects: the things. As a result, those everyday objects gain the ability to interact with their environment and communicate with each other and the Internet. More specifically, each device can observe its object or environment by reading various sensors, analyze the readings locally, possibly to infer some higher-level observations, and finally, send these observations to other nearby devices or the Internet for further processing. Similarly, from the other devices or the Internet, it can receive various commands, based on which it can trigger the object's actuators, thereby changing the object's state. Following this general scheme, the networked physical objects can collaborate directly, that is, without human intervention, in effect making our lives more convenient, healthy, secure, and environment friendly, to name just a few benefits [4]. This huge potential of IoT is well illustrated by estimates from Cisco executives, claiming that 99% of physical objects that may one day join the Internet are or, to be precise, were in 2013, still unconnected [5].

Apart from its potential, what is also special about IoT compared to the other trends listed in the aforementioned prediction is that it is more of a vision than a specific technology. In particular, the other technologies from the prediction are already contributing to IoT or are likely to contribute in the future. To start with, deep learning can be crucial for extracting patterns from the data sensed by the things and interpreting those patterns into actuation decisions that affect the physical world. Digital currencies can facilitate micropayments for the various services offered by the surrounding smart things, while augmented and virtual reality may provide interfaces for those services. Blockchain technologies can play a role in the management of information generated by the things. Robots and vehicles are yet another type of things, whose autonomous operation will likely involve interaction with other smart objects, which makes also robotics and assisted transportation fit elegantly into the IoT vision. Advances in hardware accelerators will in turn be required to deliver the compute power for the IoT analytic components, especially ones utilizing machine learning. Finally, since the so-understood IoT vision poses novel challenges related to privacy and security—including physical safety of humans and actual real-world assets—the application of artificial intelligence in cybersecurity as well as new rules of ethics, laws, and policies will simply become a necessity. All in all, IoT indeed seems as a vision with a large possible impact on computing research agendas.

What is more, the IoT vision has been becoming reality for some time already. Various solutions fitting this vision, which I refer to as IoT solutions/systems/devices/etc., are being deployed in the real-world. In particular, Gartner reports nearly 6.4 billions of IoT devices installed in 2016 [6], a 30% increase compared to 2015 [7], and forecasts a further increase by 243% by 2020. In general, even though other reports present different absolute numbers [8], depending on what is considered an IoT device, there seems to be a common agreement that, in contrast to consumer-oriented IoT gadgets, the adoption of *industrial IoT* solutions has been lagging.

There are several issues that have resulted in this lower-than-expected adoption of industrial IoT, some of which stretch beyond computing itself. In this paper, however, we will focus on three crucial ones for which I believe the experience of the distributed systems community may be of paramount value and, symmetrically, which may be of interest to the community, namely *interoperability*, *scalability*, and *dependability*. We start by analyzing how industrial IoT solutions compare to classic distributed systems (Section II). We then discuss some of the challenges posed and the research opportunities offered in the areas of interoperability (Section III), scalability (Section IV) and dependability (Section V) of IoT systems. Finally, we recapitulate major conclusions (Section VI).

II. Distributed Systems in Industrial IoT

Industrial IoT has received a lot of research attention from the networking and embedded systems communities, which has produced some of the compelling solutions that underlie today's deployments. However, as indicated previously, broadening its adoption requires further research contributions, involving the experience of the distributed systems community.

A. Industrial IoT Solutions are Distributed Systems. . .

Adopting a distributed systems perspective is rather natural considering what industrial IoT is about. More specifically, an industrial IoT system is a collection of largely independent interconnected computing elements that monitor or control some physical resources in a way that appears to the users of the system as an operation of a single facility realizing a certain business process. In other words, industrial IoT systems fit perfectly the classic definitions of distributed systems [9].

Among others, this definition emphasizes two aspects. First, it involves components that are interconnected, albeit largely autonomous. Second, it requires these components to appear to the outside world as a single coherent system. This combination implies that the autonomous components have to collaborate one way or the other. The principles and paradigms according to which such a collaboration can be established lie at the core of distributed systems.

What is more, such an approach to industrial IoT—emphasizing inter-component collaboration—complements the approaches taken by the networking and embedded system communities. The networking community typically focuses on methods of interconnecting the components so as to enable efficient and reliable communication. The embedded systems community, in turn, is concerned with the components themselves and their interfaces with the physical objects and the surrounding environment. Although these interests frequently overlap with those of the distributed systems community, they do differ, and hence considering industrial IoT solutions as distributed systems with all their classic challenges has the potential to add value.

B. . . . but Peculiar Ones

However, industrial IoT solutions do differ from the classic distributed systems, notably ones with their key components residing largely in data centers. The differences stem mostly from the tight coupling between such solutions and actual physical objects they monitor or control. These interactions with the physical world give rise to a new logical layer in the popular three-tiered architecture of distributed systems [9]—the sensing and actuation layer (see Fig. 1)—which subsumes the classic user interface layer by providing means for interaction not only with people and other systems but also physical objects. Although industrial IoT significantly impacts also the two other layers, it is the new sensing and actuation layer that we focus on in this paper.

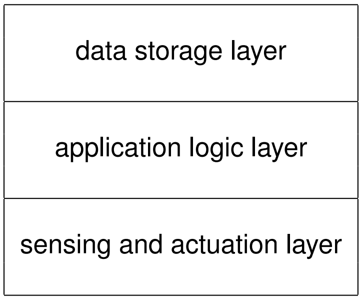

Fig. 1. Logical layering in industrial IoT.

Being responsible for the interactions of the system with the physical objects, the layer comprises the devices embedded into those objects. Because of this embedding, the devices are often constrained in their power supply, form factor, and external wiring. This, in turn, (severely) limits their resources, notably computing power, communication bandwidth, memory, and storage. Moreover, even in a single system, they can be highly heterogeneous. All in all, in terms of computing and networking capabilities, the platforms for the sensing and actuation layer are frequently on the lower extreme of the spectrum, as such diverging significantly from the platforms employed by classic distributed systems.

Likewise, the physical placement of the embedded devices is normally dictated by the specific sensing and actuation points they are to operate at. Depending on the distribution of such points, the number of the devices may be large. Similarly, the area they cover may be wide or, on the contrary, the devices may be densely packed in a small area, which, among others, affects their (wireless) network connectivity.

Finally, the devices may monitor critical phenomena or control expensive equipment. The operating conditions may involve both low and high temperatures, sometimes in subdiurnal cycles. They may also be exposed to other adverse environmental factors, such as water, dust, vibration, direct sunlight, to name just a few examples. In other words, their responsibilities and operating environments differ significantly from those of the systems deployed in "sterile" data centers.

Together, these peculiarities indicate the importance of the three aspects of the sensing and actuation layer in industrial IoT systems—interoperability, scalability, and dependability—which we elaborate on next. Note that by no means does this list of aspects aim to be complete; neither is their coverage is subsequent sections. Fully covering all challenges that industrial IoT poses and the current state of the art would require far more pages than available in this short paper. The interested reader is encouraged to start, for instance, with one of the available surveys or more recent books. In contrast, the goal of this paper is just to spark interest in some of the problems, which I encountered during my collaborations with several industrial entities, which are recognized as relevant to the entire industrial IoT, and which I believe may be of interest to the distributed systems community.

III. Interoperability in Industrial IoT

The goal of industrial IoT systems is transforming business processes. Although this transformation is often disruptive, it usually does not involve replacing the entire infrastructure. On the contrary, such systems normally complement the infrastructure or even integrate its various existing components. This implies that they have to operate with legacy components, sometimes in ways that were not envisioned by the creators of those components.

Moreover, as mentioned previously, even dedicated IoT-oriented devices can be highly heterogeneous in a single system. Depending on their function, they may vary in their power supply, available resources, networking technology, sensors and actuators attached, and the software they are capable of running, to name just a few examples. Despite such differences, they must interoperate to give an illusion of a single coherent system.

A. *Standardization*

One of the approaches to addressing this problem has been standardization. At the sensing and actuation layer, standardization is especially apparent in communication protocols for IoT devices; yet, with mixed success. In particular, for IEEE 802.15.4, designed for low-power low-data-rate communication, frequently only parts of the standard are used in practice, whereas the other parts are replaced with custom solutions so as to gain an edge over competing system providers. This renders any two different 802.15.4-based solutions likely incompatible already at the link layer. Bluetooth Low Energy aims to address this incompatibility by standardizing communication up to the application layer. This, however, makes it poorly suitable for industrial applications. Moreover, novel standards targeting different communication scenarios appear regularly (e.g., LoRa) while existing ones gain their low-power versions (e.g., Wi-Fi HaLow). Finally, there are many older standards dedicated for industrial applications that do not perfectly fit the Internet protocol stack [10].

Considering the architecture of the Internet, this plethora of low-level standards in theory need not be a problem as long as the various solutions can be integrated with the Internet Protocol [11]. However, in practice, standardizing only the ways IP packets are carried in the frames of those protocols [12], [13] or even how such packets can be routed efficiently [14] is not enough. The end-to-end integration of the various components comprising industrial IoT systems requires application layer protocols, such as the Constrained Application Protocol (CoAP) [15]. Nevertheless, it still does not guarantee interoperability with legacy solutions.

B. *Middleware*

Such integration problems have been addressed for a long time already by the distributed systems community by means of middleware. As numerous experiences from enterprise application integration illustrate, middleware can be an effective approach for ensuring interoperability, even among legacy systems. Therefore, although standardization is important, developing appropriate middleware may also have a large impact on the adoption of industrial IoT. In fact, the aforementioned CoAP is a textbook example of a middleware protocol.

One of the problems, however, apart from the peculiarities of the devices at the sensing and actuation layer of industrial IoT, is that different industries may have different needs with respect to the middleware. Discovering and addressing these needs can be a challenging process.

IV. Scalability in Industrial IoT

Another requirement for many industrial IoT systems is scalability, that is, the ability of a system to perform well when it grows significantly. The importance of designing for scalability is particularly apparent when we consider not only the possible sizes of industrial IoT solutions but also the way they are deployed. Since such solutions often aim at physically transforming business processes, their deployment typically proceeds incrementally for each customer: it starts with one or a few small tests, which are followed by a rollout comprising initially only a part of the target system, and finally, the deployment of remaining parts. Such an approach minimizes disruption to the customer's operation and the risk of a major business interruption. However, for the designers, it means that the system has to tolerate a growth even by several orders of magnitude. What is more, initial overprovisioning may not be an option because it could entail a financial loss for the system provider upon the customer's resignation at intermediate deployment stages. This need for several-orders-of-magnitude scalability is thus particularly challenging considering a common engineering rule of thumb that an order of magnitude change should normally entail a redesign [16].

There are different axes along which scalability of distributed systems can be analyzed. Here, we use a version of a classic taxonomy [17], [9] distinguishing: size scalability, geographic scalability, and administrative scalability.

A. *Size Scalability*

An intuitive definition of scalability is what is referred to as size scalability: a distributed system is said to scale well in size when its performance does not degrade or degrades only slightly when the number of users or machines grows significantly. When size scalability problems occur, the first line of defense is often to provision more resources: memory, storage, computing power, or network bandwidth, depending which is the bottleneck. A more involved approach is to employ replication or partitioning of data and/or computations, so that the load can be divided between multiple components. Only when this fails, is one forced to redesign the system (or abandon it altogether). Such a redesign typically boils down to replacing centralized services or algorithms with decentralized counterparts. However, practice shows that this process is often far from trivial, and thus the best resulting solutions frequently correspond to milestone inventions in distributed systems.

In any case, what many of these scaling techniques assume is that the logical components constituting a distributed system, that is, processes, data stores, or, in general, services, can be instantiated at different physical locations. In other words, they assume that the software architecture of the system is by and large independent of the actual system architecture [18], which, among others, dictates the placement of the various software component instances.

This is not the case for the sensing and actuation layer in industrial IoT, though, where the monitoring and control normally have to be performed at specific physical points in space, required by a particular application. The fixed placement, together with additional constraints on the form factor, power supply, and network connectivity of the devices, render many of the classic scaling techniques impractical in industrial IoT, at least at the sensing and actuation layer. In effect, these functionalities have to be explicitly designed for scalability, in particular, to a large extent being decentralized and close to the sensing and actuation points. An additional opportunity is that the architecture of the devices can often be adapted to best support such custom designs. Examples of such recent approaches involve migrating parts of deep neural networks to low-power devices to exploit the tradeoff between communication and computation [19], [20].

B. Geographic Scalability

Another scalability aspect is geographic scalability, which entails that the performance of a system does not suffer when its components are spread over a larger geographic area. Traditionally, this larger geographic area describes situations in which different components of the system reside in different data centers on different continents. This is because the wide-area network links between such components entail considerable latency of hundreds of milliseconds up to seconds in some cases [21], [22].

To facilitate geographic scalability distributed systems often employ a combination of replication and asynchronous communication and algorithms. In particular, by replicating or caching data close to the users, one can significantly reduce the user-perceived latency of accessing those data [23]. This, however, necessitates geographically-scalable solutions for keeping the replicas consistent. Currently, a common practice is to employ some variant of eventual consistency, preferably in combination with decentralized resolution of potentially conflicting updates [24]. Notably, a compelling approach is to use so-called conflict-free replicated data types (CRDTs) [25], which allow for asynchronous concurrent updates to shared replicated data.

Although such solutions can be deployed at the other two layers of industrial IoT systems, geographic scalability of the sensing and actuation layer requires different approaches. One of the problems is that, in the case of low-power wireless devices, the communication latency is dominated by the number of wireless hops the data has to travel. Since the devices sleep most of the time to conserve energy, a packet may take seconds to be transmitted over few wireless hops [26], [27]. Considering that a radio range of a single device is normally on the order of tens of meters and that some deployments span hundreds of thousands of square meters, the communication latency may be high. What is more, if there are few border routers through which the low-power devices communicate with the machines in the other layers, the devices in proximity of the routers may exhibit a heavy load, which drains their energy. In other words, geographic scalability problems of industrial IoT are directly associated with the physical spaces, in which such systems operate, and occur even at scales much smaller than a continent.

This, however, allows for revisiting the previous approaches to addressing geographic scalability problems. For instance, it turns our that by employing highly synchronous end-to-end communication involving tight coordination of multiple devices, one can minimize the end-to-end latency [28], [29], [30]. Similarly, by utilizing in-network aggregation [31] in combination with on-demand pulling of data from the devices [30], it is possible to alleviate the effects of the heavy load in the vicinity of border routers. Likewise, by exploiting parallelism, one can improve the efficiency of border router failure detection by orders of magnitude [32]. These are just a few examples but there are many other aspects of geographic scalability that yet remain open for industrial IoT.

C. Administrative Scalability

As the final scalability axis, we consider administrative scalability, which describes how well a system operates when managed by multiple parties. Administrative scalability is notoriously difficult to achieve in distributed systems [9] but there are examples of successful solutions, such as DNS [33] or PlanetLab [34]. Moreover, the advent of cloud computing, with its virtualization technologies, has allowed for alleviating many of the administrative scalability problems.

However, in the case of industrial IoT, cloud computing is not the solution for the problems at the sensing and actuation layer. Sensors and actuators managed by different entities can be sharing the same physical space. Consider, for instance, complex projects, such as construction sites, where many entities collaborate to achieve a common goal. The distributed systems utilized by those entities may share the sensing and actuation points, or even the sensors and actuators themselves. What is more, they will likely compete for resources, notably wireless communication channels. Considering the wealth of technologies that can be used for implementing industrial IoT systems, sharing the same communication bands is itself a considerable issue [35], [36]. It is, nevertheless, just one among the plethora of problems stemming from the need for administrative scalability.

V. Dependability in Industrial IoT

The last aspect we discuss in this paper is dependability. It is arguably the most frequently quoted requirement for industrial IoT systems because these solutions can have direct impact on the surrounding physical world, and hence our lives and health. There are many facets of dependability. Again, for our analysis, we employ a classic definition that requires dependable systems to be: reliable, safe, available, maintainable, and secure [9], [37].

A. Reliability

Reliability describes to what extent a system guarantees to operate correctly, that is, as prescribed in its specification. It can be quantified, for instance, by measuring the mean time to failure or the inverse thereof, that is, the frequency of failures. The measurement can regard the individual components comprising the system or the system as a whole.

Reliability is of primary concern in many distributed systems as the failure of a large system typically entails a substantial financial loss for its provider. There are essentially two orthogonal approaches to achieving a high reliability.

The first is to try maximizing the reliability of the individual components. This should preferably be done both at the hardware and software layer. From the perspective of this paper, especially the possibilities offered by software engineering are of major interest. Although there are many existing techniques that can be applied to improve software reliability, in the case of the sensing and actuation layer in industrial IoT systems, the possibilities are usually limited. This is because of the resource constraints of micro-devices at this layer, which essentially preclude many runtime techniques, like virtualization, sandboxing, or even more advanced monitoring. At the same time, however, the resource constraints open new opportunities for pre-deployment techniques. More specifically, because of the resource constraints, the software images for micro-devices are several orders of magnitude smaller than the software bases at the other layers. This facilitates analyzing various aspects of (parts of) such images statically on normal or even high-performance machines before deployments, so as to improve their reliability [38], [39], [40], [41].

The second approach to achieving a high reliability is to assume that individual components are unreliable and maximize the reliability of component groups by employing redundancy. There are essentially three types of redundancy that can be applied: information redundancy, time redundancy, and physical redundancy [42], [9]. Again, the sensing and actuation layer in industrial IoT systems limits the extent to which redundancy can be applied. Information redundancy, typically utilized for communication, is limited by the resource constraints of the devices. Time redundancy is sometimes at odds with soft-realtime requirements of such systems. Finally, physical redundancy may be severely restricted in certain applications in which sensing and actuation points are precisely defined. In other words, employing redundancy in industrial IoT can be challenging.

B. Safety

Related to reliability is safety, which guarantees that when a failure does occur, nothing catastrophic will happen. The concept of safety has been borrowed from concurrent programing and distributed algorithms. It is nevertheless crucial in industrial IoT systems that control actual physical processes, especially those in which people or infrastructure may be at risk. Ensuring safety in such applications is far from trivial and requires a careful, often formalized analysis of all possible operational scenarios. In practice, full safety can rarely be ensured and the goal is just to minimize the risk of calamities.

Following this line of reasoning, in many applications that are not life-critical, safety need not be considered only binary: it can be continuous to some extent. Consider, for instance, HVAC systems in office buildings. They normally have two requirements: ensuring comfort for the building occupants and saving electrical energy. When it comes to comfort, the (soft) safety margins may vary, depending on who occupies a given space at a given time. What is more, the system may deliberately violate these margins to minimize energy consumption. Finally, the revenue the system provider receives (or the penalties the provider has to pay) can be made dependent on the comfort and energy savings. The ways of achieving this are another interesting issue. In general, safety in industrial IoT poses a number of truly hard problems.

C. Availability

Availability is another aspect of dependability that describes the readiness of a system for usage: a highly available system is virtually always delivering its functionality, irrespective of occurring failures but perhaps with a degraded performance. Availability thus corresponds to the concept of liveness in concurrent programming and distributed algorithms. It is also related to reliability but different: a system can be highly reliable (when operating) but not necessarily constantly available, and vice versa [9].

Availability is normally achieved by means of physical redundancy, so that when a replica of a component fails, the other replicas can still deliver the functionality of the component. To this end, however, replication requires nonblocking decentralized algorithms with weak consistency guarantees, for instance, utilizing the aforementioned eventual consistency and decentralized conflict resolution; otherwise, the system cannot ensure availability upon network partitions, as formalized by Brewer's CAP theorem [43].

The theorem is also relevant for industrial IoT systems, many of which should be always on. In particular, in the presence of network partitions, like a lack of network connectivity with the Internet or between the various layers, such systems must at least guarantee safety. Preferably, however, they should continue offering their functionality, possibly within a limited scope, so that the core processes they control can proceed undisturbed. Despite these requirements, however, partition tolerance of the protocols at the sensing and actuation layer of industrial IoT systems has still received relatively little research attention [44] and thus likely leaves room for novel contributions.

D. Maintainability

Somewhat related to availability is maintainability, defined as the ease of changing the configuration of the system or replacing its failed components. Maintainability helps achieving availability and reduces the operational costs of a system. Ideally, systems should thus be self-organizing, self-managing, self-healing, and, in general, self-$\star$ but practice shows that there are tradeoffs to be made.

At the sensing and actuation layer of industrial IoT, maintainability did receive considerable attention but only in some aspects. For instance, even though networking protocols are largely self-organized, they often require expertise when configured for individual deployments [45]. In contrast, little work has been done on automated diagnosis of sensing and actuation components.

E. Security

Finally, a dependable system must also be secure; otherwise, arbitrary faults can be injected into it, violating the system designers' basic assumptions. There have been volumes of complaints on the poor recognition of this fact by IoT designers, notably of consumer-oriented products. The famous quote circulating on the Web arguably best summarizes this situation:

The "s" in "IoT" stands for security.

In industrial IoT, the recognition of the importance of security is somewhat better. Nevertheless, securing IoT systems, notably the sensing and actuation layer, still poses a number of open challenges, many of which stem from the resource constraints of microdevices. For example, although networking standards for such devices do include provisions for a range of secure modes [14], they are hardly implemented [46]. Furthermore, our knowledge of novel security threats that industrial IoT systems pose is still in the process of (slowly) building up. All in all, security in industrial IoT is an important yet difficult topic, which is reflected in the aforementioned predictions by IEEE Computer Society, explicitly mentioning security as one of the key trends.

VI. Conclusions

To sum up, from the distributed systems perspective, industrial IoT systems indeed pose a number of challenges. They have to interoperate with existing infrastructures and integrate highly heterogeneous hardware-software platforms. They also have to be prepared to scale a few orders of magnitude in size, diameter, and/or density, as well as to enable management by different entities. Moreover, they have to be dependable, that is, reliable, safe, available, maintainable, and secure, all at the same time. What is also important is that this list of requirements is by no means complete.

In this light, the already immense and yet growing real-world interest in industrial IoT is an excellent opportunity for novel contributions in distributed systems. To this end, however, we—as the academic community—must establish tighter collaborations with various industrial partners. Only through such collaborations will we be able to discover the specific problems they are facing and the constraints that they must respect when deploying IoT solutions in their business processes. Such collaborations have the potential for a significant impact, both scientific and practical one.

Acknowledgments

This work was supported by the National Center for Research and Development (NCBR) in Poland under grant no. LIDER/434/L-6/14/NCBR/2015.

References

[1] IEEE Computer Society, "Top 10 technology trends for 2018: IEEE Computer Society predicts the future of tech," https://www.computer.org/web/pressroom/top-technology-trends-2018, December 2017.

[2] IEEE Computer Society, "IEEE Computer Society grades its 2016 technology predictions – gets a B+," https://www.computer.org/web/pressroom/2016-tech-grades, December 2016.

[3] IEEE Computer Society, "IEEE Computer Society's report card grades 2017 technology predictions: Overall score is A-," https://www.computer.org/web/pressroom/report-card-2017-trends, November 2017.

[4] J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*, 1st ed. Elsevier, April 2014.

[5] R. Soderbery, "How many things are currently connected to the "internet of things" (iot)?" https://www.forbes.com/sites/quora/2013/01/07/how-many-things-are-currently-connected-to-the-internet-of-things-iot, January 2013.

[6] R. van der Meulen, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," https://www.gartner.com/newsroom/id/3165317, February 2017.

[7] R. van der Meulen, "Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015," https://www.gartner.com/newsroom/id/3165317, November 2015.

[8] J. Howell, "The Industrial Internet of Things is here, but widespread adoption remains elusive," https://ihsmarkit.com/research-analysis/the-industrial-internet-of-things-is-here.html, November 2017.

[9] M. van Steen and A. S. Tanenbaum, *Distributed Systems*, 3rd ed. CreateSpace Independent Publishing Platform, February 2017, no. 978-1543057386.

[10] B. Drury, *Control Techniques Drives and Controls Handbook*, 2nd ed. Institution of Engineering and Technology, July 2009, no. 978-1849190138.

[11] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*. Morgan Kaufmann Publishers Inc., 2010.

[12] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," RFC 4944, 2007.

[13] J. W. Hui and D. E. Culler, "Ip is dead, long live ip for wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 15–28.

[14] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, "RPL: IPv6 routing protocol for low-power and lossy networks," RFC 6550, March 2012.

[15] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," RFC 7252, 2014.

[16] J. H. Saltzer and M. F. Kaashoek, *Principles of Computer System Design: An Introduction*, 1st ed. Morgan Kaufmann, July 2009, no. 978-0123749574.

[17] B. C. Neuman, "Scale in distributed systems," in *Readings in Distributed Computing Systems*. IEEE Computer Society Press, 1994, pp. 463–489.

[18] L. Bass, , P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd ed. Addison-Wesley Professional, December 2008.

[19] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar, "DeepX: A software accelerator for low-power deep learning inference on mobile devices," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, ser. IPSN '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 23:1–23:12.

[20] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, “DeepIoT: Compressing deep neural network structures for sensing systems with a compressor-critic framework,” in *SenSys ’17: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, November 2017.

[21] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, “Broadband Internet performance: A view from the gateway,” in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM ’11. New York, NY, USA: ACM, 2011, pp. 134–145.

[22] T. Høiland-Jørgensen, B. Ahlgren, P. Hurtig, and A. Brunstrom, “Measuring latency variation in the internet,” in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT ’16. New York, NY, USA: ACM, 2016, pp. 473–480.

[23] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. van Steen, “Replication for web hosting systems,” *ACM Computing Surveys*, vol. 36, no. 3, pp. 291–334, September 2004.

[24] S. Burckhardt, “Principles of eventual consistency,” *Foundations and Trends in Programming Languages*, vol. 1, no. 1-2, pp. 1–150, 2014.

[25] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, “Conflict-free replicated data types,” in *Stabilization, Safety, and Security of Distributed Systems*, X. Défago, F. Petit, and V. Villain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 386–400.

[26] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle mac with scheduled channel polling,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, ser. SenSys ’06. New York, NY, USA: ACM, 2006, pp. 321–334.

[27] Y. Sun, O. Gurewitz, and D. B. Johnson, “Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks,” in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys ’08. New York, NY, USA: ACM, 2008, pp. 1–14.

[28] P. Dutta, D. Culler, and S. Shenker, “Procrastination might lead to a longer and more useful life,” in *Proceedings of the Sixth Workshop on Hot Topics in Networking (HotNets-VI)*, November 2007.

[29] N. Burri, P. von Rickenbach, and R. Wattenhofer, “Dozer: Ultra-low power data gathering in sensor networks,” in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, ser. IPSN ’07. New York, NY, USA: ACM, 2007, pp. 450–459.

[30] R. Musaloiu-E., C. J. M. Liang, and A. Terzis, “Koala: Ultra-low power data retrieval in wireless sensor networks,” in *2008 International Conference on Information Processing in Sensor Networks (IPSN 2008)*, April 2008, pp. 421–432.

[31] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tinydb: An acquisitional query processing system for sensor networks,” *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, Mar. 2005.

[32] K. Iwanicki, “RNFD: Routing-layer detection of DODAG (root) node failures in low-power wireless networks,” in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2016, pp. 1–12.

[33] P. Albitz and C. Liu, *DNS and BIND*, 5th ed. O’Reilly Media, February 2009.

[34] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, “Operating system support for planetary-scale network services,” in *Proceedings of the First USENIX Symposium on Networked Systems Design and Implementation (NSDI ’04)*, March 2004.

[35] R. Natarajan, P. Zand, and M. Nabi, “Analysis of coexistence between IEEE 802.15.4, BLE and IEEE 802.11 in the 2.4 GHz ISM band,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, October 2016, pp. 6025–6032.

[36] U. Wetzker, I. Splitt, M. Zimmerling, C. A. Boano, and K. RÃ¶mer, “Troubleshooting wireless coexistence problems in the industrial internet of things,” in *IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC)*, Aug 2016, pp. 98–98.

[37] H. Kopetz and P. Veríssimo, “Real time and dependability concepts,” in *Distributed Systems*, 2nd ed., S. Mullender, Ed. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1993, pp. 411–446.

[38] P. Li and J. Regehr, “T-Check: Bug finding for sensor networks,” in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN ’10. New York, NY, USA: ACM, 2010, pp. 174–185.

[39] R. Sasnauskas, O. Landsiedel, M. H. Alizai, C. Weise, S. Kowalewski, and K. Wehrle, “KleeNet: Discovering insidious interaction bugs in wireless sensor networks before deployment,” in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN ’10. New York, NY, USA: ACM, 2010, pp. 186–196.

[40] K. Iwanicki, P. Horban, P. Glazar, and K. Strzelecki, “Bringing modern unit testing techniques to sensornets,” *ACM Trans. Sen. Netw.*, vol. 11, no. 2, pp. 25:1–25:41, August 2014.

[41] M. Ciszewski and K. Iwanicki, “Efficient automated code partitioning for microcontrollers with switchable memory banks,” *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 4, pp. 114:1–114:26, May 2017.

[42] B. W. Johnson, “An introduction to the design and analysis of fault-tolerant systems,” in *Fault-tolerant Computer System Design*, D. K. Pradhan, Ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996, pp. 1–87.

[43] E. Brewer, “CAP twelve years later: How the “rules” have changed,” *IEEE Computer*, vol. 45, no. 2, pp. 23–29, February 2012.

[44] A. Paszkowska and K. Iwanicki, “The IPv6 routing protocol for low-power and lossy networks (RPL) under network partitions,” in *EWSN 2018: Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*, February 2018, pp. 90–101.

[45] A. Paszkowska and K. Iwanicki, “On designing provably correct DODAG formation criteria for the IPv6 routing protocol for low-power and lossy networks (RPL),” in *DCOSS 2018: Proceedings of the 14th International Conference on Distributed Computing in Sensor Systems*. IEEE, June 2018.

[46] H.-S. Kim, J. Ko, D. Culler, and J. Paek, “Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey,” *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.