The Changing Computing Paradigm With Internet of Things: A Tutorial Introduction

Sandip Ray
Intel Corporation

Arijit Raychowdhury
Georgia Institute of Technology

Yier Jin
University of Central Florida

Editor's notes:
This Tutorial paper is about the Internet of Things, its applications, challenges, and how it may change the way of computing. Besides a comprehensive introduction, it focuses on two major design constraints, namely, security and power management.

—Jörg Henkel, Karlsruhe Institute of Technology

■ **We are at** the brink of Internet of Things (IoT)—a regime in which we are surrounded by hundreds of billions of smart, connected computing devices identifying, analyzing, and influencing some of our most intimate personal activities. IoT represents one of the fastest growth points in the history of computing, with a projected 50 billion devices by the end of 2020 [1]. The scope of impact of computing in this new era is also pervasive. It encompasses individual, enterprise, automotive, and cloud services, and brings together such diverse areas as security, energy efficiency, physical design, analytics, and software development.

A critical effect of the broad scope of IoT is that it challenges the current separation of topic areas in computing. For example, an IoT device requires energy efficiency, security, interoperability with software applications, etc., and it is difficult today to separate the concerns of one topic when considering the advances in another. To a researcher, this effect is both a challenge and an opportunity. On the one hand, it provides a fertile ground for collaborative, interdisciplinary research. On the other hand, it implies that researchers entering in this

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/MDAT.2016.2526612
Date of publication: 08 February 2016; date of current version: 25 February 2016.

space often have significant ramp-up requirements before they can meaningfully contribute to this domain.

Unfortunately, in spite of its overarching need, and notwithstanding the large body of discussions, debates, and opinion pieces surrounding it over the recent years, there is a surprising dearth of literature to provide an objective, technical survey of the overall computing landscape impacted by IoT. The beginner in this domain is thus faced with the daunting proposition of sifting through either nontechnical op-ed pieces and literature from various companies [2]–[5] or little nuggets of technical results sprinkled across a large number of diverse, often unrelated, conference proceedings and journal volumes to piece together the challenges and progress in this area. Indeed, a considerable confusion still remains on what exactly constitutes IoT: Is it the edge devices only, whether it encompasses the overall infrastructure of devices, routers, and data centers, or whether the ecosystem of software and applications are also included, etc.

In this article, our goal is to bridge this gap by providing a more unified view of IoT. We provide a tutorial introduction to the IoT ecosystem, what it enables, how the industry is moving toward it (as of this writing), and some of the research needs. As concrete illustrations of research needs, we deep-dive into two specific research challenges in IoT: security and energy efficiency. We believe this will give the reader a taste of the challenges in this domain and the kind of collaborative research that can be effective.

The remainder of the article is organized as follows. Section "What is in IoT" introduces the basics of IoT, and provides a flavor of research challenges at a high level. Section "Evolution of the IoT and current applications" recounts some of the history of evolution of IoT, and provides an overview of the current state of the regime. "IoT structure and Web of Things" provides a gentle technical overview of IoT infrastructure, and also introduces the Web of Things (WoT), i.e., the evolution of Internet to support IoT. "Commercial IoT solutions" describes some of the enterprise IoT solutions as of this writing. We then deep-dive on two research areas related to IoT, security (in "Security challenges") and energy (in "Power management challenges"). Note that the article is intended as a first tutorial introduction for a beginning researcher; it should not be treated as a comprehensive survey of the field. The article contains a significant bibliography including both traditional research articles and pointers to various business solutions and white papers, to enable further exploration.

What is in IoT

Before we discuss IoT and the numerous challenges it introduces, it is important to understand what exactly constitutes IoT. Loosely speaking, it refers to physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. This allows objects to be sensed and controlled remotely, creating opportunities for direct integration between the physical and cyber worlds.

While the above loose description is standard, there is still considerable confusion regarding the scope of IoT. The reason is that IoT is not a single well-defined new technology but rather an ecosystem that exploits and expands upon existing environments of connected, embedded devices. From one point of view, there is nothing novel about IoT at all; it is simply "embedded systems with a new name." Another viewpoint emphasizes the size and complexity of new applications of IoT devices—applications in the scale of smart homes to smart cities and beyond, and facilitating automation in all aspects of our life from personal to enterprise. From this point of view, it is one of the "biggest revolutions in history after sliced bread." Furthermore, the notion of what constitutes the primary component of IoT gets distorted depending on the perspective of the persons involved. To some, IoT is all about the design of low-power, connected, embedded devices; to some it is about designing a scalable, connectivity infrastructure; to others it is about smart end-to-end analytics from sensory data coming from a billions of sources, etc.

Given the lack of consensus on what exactly is included within the scope of IoT, in this article we will adopt the following definition:

> The scope of IoT is the computing infrastructure to enable an ecosystem in which there are more "things" connected to the Internet than people.

This definition is adapted from the Cisco white paper on IoT [1]. To understand the explosive growth of computing that the implies, note that in 2003 there were about 500 million computing devices connected to the Internet in a world of about 6.3 billion people, a device-to-people ratio of 0.08; in contrast, in 2020, with a projected 50 billion connected devices in a world of projected population of 7.5 billion, this ratio is approaching 7, a growth of more than 80 times over population growth! Note that these estimates do not take into account rapid advances in Internet or device technology; the numbers presented are based on what is known to be true today. Additionally, the calculation of device-to-human ratio is based on the entire world population, much of which is not yet connected to the Internet; by reducing the population sample to people actually connected to the Internet, the number of connected devices per person rises dramatically, even projected to 2020.

Given the above estimates, the challenge in IoT is not a challenge of a point solution or technology but one of scale, encompassing the entire area of computing. Our computing paradigms and even the Internet were not created with this scale of computing in mind. Enabling IoT requires rethinking of some of the fundamentals of computation and communication paradigms, device fabrication process, software and user experience, security and privacy issues, etc. This also explains why it is so difficult to pinpoint what exactly constitutes IoT and what does not: the silos of individual topics created as computing have matured over the last 70 years and often need to be broken to provide an IoT solution.

What kind of research problems should we be looking at in the IoT space? We will provide more technical answers for a few research directions later in the article, but the following provides a very small sample of the kind of high-level questions that need to be answered.

- Data interoperability: We now have billions of devices generating disparate sensory data, each in its individual formats and languages. These data are transmitted to processing elements such as gateways, data centers, and the cloud for processing and analytics (see below). The processing elements today have to "understand" this disparate and ever-changing data format and languages from millions to billions of devices. A key challenge is to standardize a data language that is independent of the type and form of the sensors involved.
- Low-power device support: Most IoT "edge" devices are low-power, wireless devices. For energy limitations, they must keep low duty cycles (the percentage of time active). On the other hand, much of the Internet that we know today is not optimized for low-power devices: the default assumption is that devices are always active. For example, TCP cannot distinguish between packets dropped due to congestion or packets lost on wireless links.
- Security and privacy: Security in the IoT ecosystem is particularly complex and challenging because with so many connected devices communicating from all around the globe, one expects the communication infrastructure to include some malicious components at all times. The problem is acute since the data being communicated are often sensitive, ranging from highly personalized consumer information such as health and sleep pattern, to trade secrets of the enterprise to state secrets of the government and the military. It is critical to ensure that a malicious agent in the ecosystem cannot infect or destroy the overall communication infrastructure.
- Manufacturing and process: Device manufacturing is heavily diversified given the large amount of IoT manufacturers. The lack of standard IoT design and fabrication processes makes the IoT market largely unregulated. As a result, the selection of hardware platforms and software/firmware stacks are mainly up to the manufacturers' decisions leaving the device compatibility solely at the network level. The diversity also makes it difficult to develop general design solutions, thereby increasing the overall cost for IoT construction.
- Analytics: Data mining is a crucial component of IoT, which helps create meaningful information out of raw data from diverse sensors and helps applications react to its environment, e.g., a smart home can adjust temperature by identifying pattern of occupancy at different times of the day or different seasons of the year. However, traditional data mining algorithms are centralized, requiring all raw data to be

transmitted to a computing server on which the algorithm is executed. However, continuous stream of raw sensor data from billions of devices, if transmitted directly to the cloud for processing, would overwhelm the bandwidth of the Internet and the processing ability of data centers. A critical challenge is to identify how much data to collect and how to aggregate the data at the edge, routers, gateway, and at the cloud. Furthermore, since the cloud is a distributed network of servers, it is critical to identify ways to partition data mining tasks within this network and perform analytics in a distributed manner.

- Software: User experience in the IoT regime critically depends on software developers creating innovative, intelligent, immersive applications of the extensive connectivity infrastructure. However, software developers in the IoT ecosystem form a heterogeneous group, with at least three categories: 1) edge developers coding close to hardware, who develop drivers for various edge devices and create the interface for propagation of sensory data; 2) analytics developers for mining and aggregating data coming out of various sensory sources; and 3) mobile application developers to provide front-end interface. The skills and infrastructure requirements for each of these categories are very different with little commonality in between, e.g., data analytics developers have little familiarity (and often, interest) in how the underlying hardware extracts information, or how the analytics results can be presented to the user in the front end. A key challenge is to provide a programming language, infrastructure, and programming paradigms encompassing developers from such diverse categories.

Many of the research questions cut through the above categories. For example, an obvious issue is the tradeoff between security and analytics requirements, e.g., how to obtain sufficient, relevant data without compromising the user's privacy and security. Indeed, one of the key lessons of research in the IoT regime is that the currently established research silos and topic areas are insufficient to cover the challenges in this space, and cross fertilization and collaboration are key to research progress in this space, perhaps blending some topic areas to the point that the identities of individual subjects become indiscernible.

IoT versus embedded systems

Before we end the general overview of IoT, let us answer the question that we brought up at the beginning of this section: "How are IoT devices different from embedded systems?" Embedded computing devices were originally created as dedicated low-power, hardware/software systems targeted to specific applications, e.g., traffic control, biomedical applications, automotive, etc. While some of these devices were connected to the Internet, the connectivity was dictated primarily by the function of the device. There was no expectation that disparate devices from different domains communicate with one another; when they did, it was only a handful of computing devices. IoT devices on the other hand are characterized by billions of such devices connected to the Internet. Apart from the sheer difference in the scale of connectivity, IoT devices have also instigated an urgent need to unify device functionalities across different domains and use cases. Thus, instead of isolated, dedicated embedded devices, the IoT ecosystem is one of devices communicating and sharing data, storage, aggregation, and analysis. This need also drives the requirement for common tools and frameworks for achieving this cooperation.

Evolution of the IoT and current applications

One of the first instantiations of a "thing" connected to the Internet was a modified soda machine at the Carnegie Mellon University in 1982, which could report its drinks inventory as well as the temperature of the drinks stored [6]. In the 1990s, seminal papers by Weiser [7] and Raji [8] solidified the vision of integrating and automating "everything from home appliances to entire factories." In the 2000s, with continuous Internet connectivity becoming more and more accessible, and wearable and portable computing devices becoming affordable, the ability to network embedded devices became cost-effective and attractive for both businesses and personal applications.

The rise of smartphones and tablets starting in the late 2000s provided a significant impetus for the development of applications on sensor-controlled

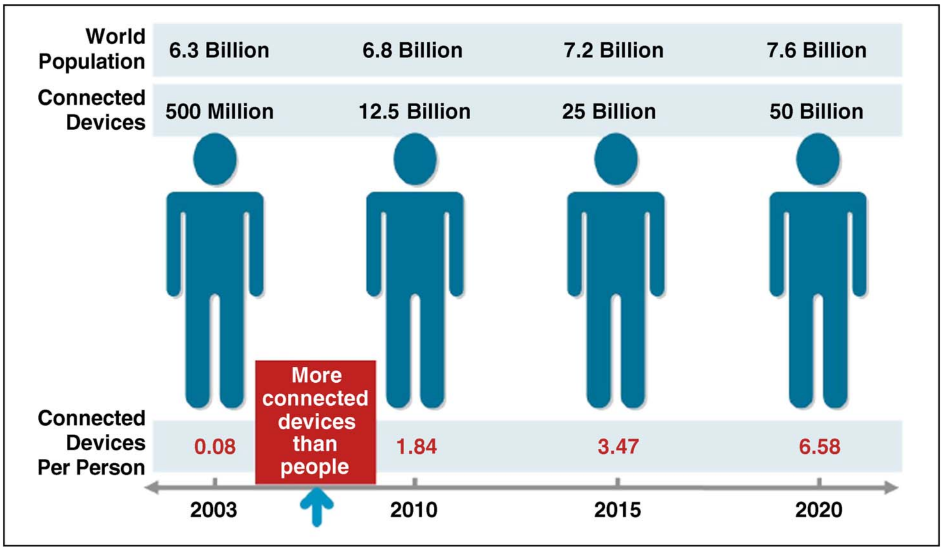

Figure 1. Growth of IoT in the 2000s. Source: [1].

devices. In particular, the advent of third-generation (3G) connectivity on mobile devices, the ability for users to develop customized applications for smartphones and tablets, and the ability of sensor-controlled devices to perform actions in addition to just sensing implied that one could develop customized applications to control and coordinate such devices. The positive spiral has led to a sharply increasing growth of increasingly smarter sensor devices as well as highly sophisticated applications. Figure 1 provides a pictorial representation of this growth. Based on the figure, IoT as defined in this article was initiated sometime between 2008 and 2009, when the number of connected devices surpassed the world population.

Today we have over ten billion devices connected and performing coordinated computations, with a diverse portfolio of current and potential applications [9]. We can divide IoT products today into at least the following three domains.

- Consumer: Consumer IoT solutions today include smart homes [10], vehicles [11], and fitness systems [12], [13]. The goal here is to adapt to the taste and needs of the consumers to add value to the user experience. A smart home may control temperature and lights based on resident moods, time of the day, day of the week, or season of the year; a fitness system may adjust to the user's health and fitness and the amount of sleep the user has had the night before, etc. Note that an IoT solution may be created through an agglomeration of several smaller IoT solutions, e.g., a smart home may be created through combining smart thermostats, washing machines, home security, etc.
- Social: A second critical application area of IoT solutions includes civic, social, and government services. The goal of these solutions is to automate and streamline government and civic infrastructure. Applications in this category include traffic management, emergency services, and environment management [14], [15]. Note that these applications require intelligent adaptation of the system to usage patterns analogous to consumer applications, e.g., a traffic management system may adapt to traffic patterns at different times of the day or different months of the year; in addition, it may be necessary to comprehend a complex supply-chain pipeline, e.g., an IoT application for emergency response must coordinate medical, rescue, and protection services with complex dependencies.
- Enterprise: The third category of IoT applications entails automating enterprise activities. Applications in this category include manufacturing supply chain, transport, etc.

A note on smartness

An over-arching theme in the discussion of any IoT application is the requirement of "smartness" in computing devices. We are already in the world of smartphones, smart watches, smart eyeglasses, smart implants, and so on, and we are looking at applications like smart homes and cities. However, it is worth reflecting upon what can make computing devices qualify to be smart. What makes a fitness tracker smart? A smart fitness tracker "understands" the activity being performed (e.g., running, swimming, walking, sleeping, etc.), anticipates the type of data to be processed for that activity (e.g., number of steps, heart rate, sleep pattern, etc.), and provides feedback and advice to the user relevant to that activity. More generally, a smart computing device is aware of the context in which it receives information from sensors and automatically derives the appropriate response for that context. As we move toward the future, we can speculate that smartness will be increasingly defined by awareness and response to the environment. A smart computing application will anticipate the context, learn from it, and adapt to the user needs.

IoT structure and Web of Things

Figure 2 provides a high-level overview of IoT communications infrastructure. We now discuss the key components of this infrastructure.

- Edge devices: These constitute the "things" component of IoT. They are typically low-power embedded devices containing sensors for different environmental stimuli (e.g., temperature, location, etc.). The goal of these devices is to receive sensory data from the physical objects, and control responses to these objects. As IoT applications get more and more complex, the edge devices often form a subnetwork of their own, communicating through one or more edge routers. The communication protocols among edge devices within today are still home-grown, based on the kind of sensory data captured by the devices, the type of edge routers involved, and the type of communication infrastructure necessary.
- Gateways: Gateways provide the routing infrastructure for communication across different device subnetworks as well as between device networks and the cloud. What makes gateways in the IoT space distinct from traditional gateways in the Internet is the need for local computation. Data transmitted by edge devices are typically subjected to analytics at the cloud; however, given the large amount of data generated from the sensors it is too expensive to transmit all of the raw data to the cloud and perform centralized analytics. Consequently, gateways are responsible for identifying usable information from raw data, performing local analytics, and transmitting the result to the cloud when necessary; correspondingly, they are also often responsible for translating the results of data analytics into control decisions for the edge devices, e.g., if the sensory data are temperature data and the analytics identifies fire, then the mitigating action may be to activate the devices responsible for fire fighting.
- Cloud servers. The cloud servers or data centers are the main computing workhorse in the IoT ecosystem. The cloud provides shared storage, information, and computation power to the IoT ecosystem. For typical IoT applications, the cloud receives sensory data from the edge devices through routers and gateways, and

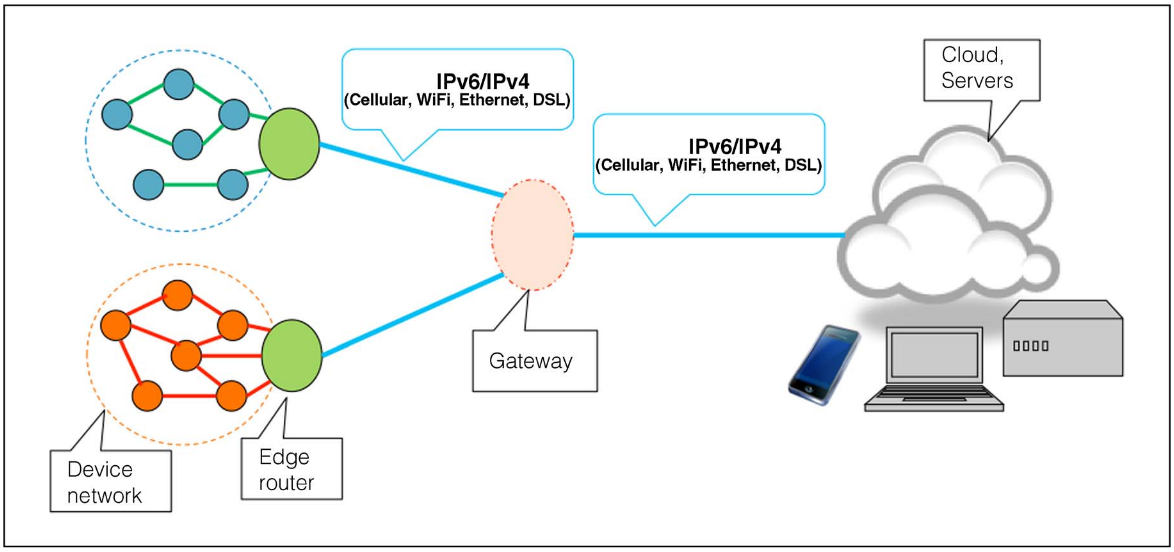

Figure 2. High-level overview of IoT communications infrastructure.

perform extensive analytics to extract useful information. The information generated may be subsequently used to develop mitigating actions, as in case of the fire-fighting example above.

Web of Things

As discussed above, the IoT structure is a connectivity infrastructure for objects or “things” to communicate with one another in the wide ecosystem of billions of devices. The WoT refers to the applications and software architectural styles on top of IoT: the primary concern is to incorporate physical objects into the IoT infrastructure, providing them API, and integrating and reusing them in various IoT applications. The function of WoT in the IoT infrastructure is analogous to that of the application layer in the OSDI stack of the Internet. For example, some IoT scenarios require real-time interaction between things, but application development can involve highly complex interaction of a number of protocols and environments; the goal of the WoT is to provide tools and APIs to facilitate such development. The WoT is built reusing existing Web standards, e.g., JSON, HTTP, REST, Websockets, etc. [16]. Note that these technologies were created primarily for desktop computers; integrating these capabilities for embedded devices, and finally to virtualizing physical objects is an important area of extension.

We end this section with a discussion of one key constraints for IoT, the need for a large network address space to handle the surge of connected devices. Communication over the Internet happens through internet protocol (IP), and requires a distinct address for the communicating devices. The dominant IP today is IPv4, which was introduced by ARPANET in 1983 [17]. Unfortunately, IPv4 uses 32-b addresses, which permit only 4.3 billion unique addresses making it unsuitable for IoT. This has led to the development and deployment of a successor protocol, IPv6 [18], which uses 128-b addresses. Furthermore, IPv6 permits hierarchical address allocation methods, and facilities supporting device mobility and security. On the other hand, IPv4 and IPv6 are not designed to be interoperable, making a smooth transition in the scale of the Internet a challenge. This has led to the creation of several transition mechanisms that permit communication between IPv4 and IPv6 hosts [19], [20]. Finally, IPv6 is difficult to use for low-power devices, e.g., it assumes that the device will be continuously connected, while low-power devices must reduce the amount of active time to conserve energy. To address this, another protocol called IPv6 over low-power wireless personal area networks (6LoWPAN) [21], [22] is conceived for low-power devices. In particular, it defines header compression and encapsulation mechanisms for IPv6 packets; this permits their transmission over IEEE 802.15.4 networks that allows Internet connectivity at low data rates suitable for low-power devices.

Commercial IoT solutions

Given the importance and growth potential of IoT, several companies and enterprises are joining the “bandwagon” of creating an IoT solution. Depending on the business interest of the enterprise, different solutions focus on different categories of the IoT space. In this section, we provide a brief overview of some of the current enterprise solutions. The goal here is not to provide a comprehensive detail of every enterprise but to provide a flavor of the different business angles related to IoT.

- Apple Homekit [23]: Apple Homekit tries to provide consumers with secure, convenient control over smart devices. It focuses on providing seamless integration with products already in the Apple ecosystem, ensuring ease of use on their mobile devices running their iOS operating system, and providing the user with an intuitive way to control the smart devices within their home.
- Cisco Fog Computing IOx [24]: Cisco’s IoT approach is focused on making it a software platform. Their idea is to view smart devices as extensions of their fog computing system, rather than emphasizing the devices, and targets a commercial audience instead of mainstream consumers. A key target of the solution is to add more smartness to the routers, gateways, and communicating infrastructure, to enable analytics and efficient data processing at source. To that end, the fog computing platform emphasizes the ability to process large amounts of data, which happens to be produced by a family of smart devices.

- Google NEST [25]: Nest attempts to add more automation to smart devices. Its distinguishing feature is providing an API to build algorithms that work off of the smart device data to allow them to do things with less user input, such as a thermostat changing the temperature based on living patterns or a carbon monoxide sensor triggering an alert when the levels reach a certain point. In addition they provide a set of algorithms that will try to customize the smart device configuration on a per-user basis.
- IBM Node-RED, Bluemix [26]: Node-Red is IBM's foray into IoT software. Their goal is to provide an easy way to create general processing and storage pipelines. Their approach exposes a UI that visualizes how data will flow from a series of information sources, such as Twitter or smart devices, and then passes through processing nodes until they are stored in some form of storage now. This makes the topology of their setup explicit, while still abstracting away some of the lower level details.
- Intel IoT platform [3]: Intel aims to provide every single component for the infrastructure of the IoT. Their approach is to create a platform that includes all the software and hardware needed to utilize smart devices, starting with smart devices built from a reference stack that Intel already provided, and including their own servers to collect and process data from them. Their foray into the IoT market seems to target developers more than consumers, unlike some of these other approaches.
- Microsoft nitrogen.io [27]: The nitrogen.io platform attempts to help create smart device front ends easily. To do so, Microsoft provides both server and client Node.js libraries that may be used to create connections from the smart devices to Microsoft's Azure cloud servers. Their approach is to provide a simple API that will allow their users to develop their own applications, and just provide a simple way to deal with the difficulties of connecting from the smart devices to the cloud.

Security challenges

The wide deployment of IoT on diverse application domains creates significant implications to security and privacy. From the consumer perspective, it is clear that an environment in which we are surrounded by computing devices collecting data on some of our most intimate activities—often for the express purpose of comprehending our behavior pattern through analytics—would lead to serious concerns regarding privacy. Similar concerns are valid for IoT solutions for enterprise or government: an IoT infrastructure automating an enterprise supply-chain pipeline could easily include exploitable back doors or Trojans. The situation is exacerbated by the fact that IoT solution development itself includes several players with a complex supply-chain ecosystem, which makes it vulnerable to attacks at different phases of its development. For the IoT regime to be successful it is critical for us to identify the security issues in this space and create protection and mitigation strategies.

It is of course worth noting that security has been a topic of critical interest since the early days of computing, with significant fundamental research dating back to the 1970s and the 1980s [28]–[30]. Indeed, even in the world of embedded and mobile systems prior to IoT, a significant fragment of the system development lifecycle is devoted to developing security architecture, validating security objectives, and ensuring that functionality is preserved in the presence of security constraints [31]–[33]. However, IoT has several unique characteristics which make it difficult to directly apply many of the security mitigation approaches developed for general-purpose and embedded computing systems [34], [35]. Note that IoT security is a nascent and highly active research topic, and many of the issues and challenges are unexplored; the challenges discussed here should only be treated as providing a flavor of the problems we need to solve rather than a comprehensive compendium of research topics in the area.

Security-affecting factors for IoT

To understand why many of the traditional security technologies cannot be directly employed for IoT, we need to comprehend some of the unique characteristics of the IoT space that influence security. Not all these characteristics make security enforcement harder: some in fact facilitate development of protection. We divide the IoT security issues into the following three categories: 1) factors that make security more challenging for IoT; 2) factors that facilitate security assurance; and

3) miscellaneous factors that influence security and make the problem different from traditional embedded and mobile security.

The following factors are included in category 1). Note that many of these factors arise through interplay of security with other features of IoT, e.g., need for smart analytics, scale, and heterogeneity.

- Exascale distribution: One of the biggest factors making security challenging is the problem of scale, diversity, and customization. IoT involves billions of devices deployed in diverse scenarios through enterprise, government, and consumer markets. For many of these applications, the high priority properties are reliability, functionality, and responsiveness. Consequently, security countermeasures must take these factors into consideration, e.g., it is not possible to develop a mitigation response against a security attack by simply shutting off system functionality. Furthermore, a key appeal for IoT devices is the potential for customization. This, together with the differing security needs for different enterprise deployments, makes it difficult to develop a generic security recipe across the devices even within a single deployed IoT system even if the systems provide the same functionality, e.g., an automotive supply-chain management system would typically be customized for different manufacturers and locations even if the overall functionality is the same; each customization would likely induce a different security requirement. Addressing scalability in security architecture in the presence of customization is a critical challenge.
- Heterogeneous platforms and connections: Recall from commercial IoT solutions that we now have different enterprise IoT solutions, often competing with one another. This introduces a significant heterogeneity in a deployed system, e.g., is it possible for iOS devices to be connected to a NEST device through Apple Homekit? In fact, enterprise solutions today acknowledge the need for supporting heterogeneous platforms and connections. Products compatible with HomeKit include surveillance cameras, smart door locks, thermostats, grilling thermometers, lights, and garage door controllers; similarly, Amazon's AWS IoT platform [36] allows the user to easily connect devices to the cloud and to other devices using HTTP and MQTT [37]. The Amazon's AWS IoT platform also supports other industry-standard and custom protocols enabled by cross-protocol communications [36]. The diversity of IoT device platforms and the performance-oriented networking protocols make it much more challenging to protect modern IoT devices. Given that there are not any standardized security solutions available, it is mostly up to the device manufacturers (or platform developers) to determine if their platforms are properly protected.
- IoT resources: Different from traditional embedded systems which have extremely limited onboard resources, IoT devices are normally equipped with medium- to high-end processor cores which are powerful enough to execute malicious payloads. Ironically, while the available onboard resources are capable of executing malicious payloads, they are often not powerful enough to apply sophisticated protection schemes developed for general computing systems. In other words, general attack models may apply to IoT while related defense solutions may not work. As a result, many of the design vulnerabilities existing in general computing systems, e.g., ROP attacks, buffer overflow, etc., can also be leveraged by attackers to compromise IoT devices. Meanwhile, existing countermeasures may not be applied due to resource constraints, requiring more efficient solutions dedicated for IoT platforms.
- Long device life: IoTs represent a break from traditional computing in the requirement for a long device life. Ever since the beginning of computing, the lives of computing devices have increasingly shortened. Today, desktops and laptops have an average life of a couple of years and mobile devices (e.g., smartphones, tablets, watches, etc.) have a life of less than a year before they are replaced or become obsolete. In contrast, IoT devices require a much longer life, going from ten to as much as 30 years. For example, a smart automotive may be on road for a decade after introduction, although the controls and software are expected to be patched and upgraded at regular intervals. Where this affects security is that many security solutions are not developed with such a long life in mind, e.g., encryption algorithm implementations have a

life-span of five to seven years. The impact of this is that 1) we need different algorithms for security mitigation, with the longevity integrated into design from the ground up; and 2) facilities are necessary for effective on-field patching of security implementations.

The above factors pose significant challenges to effective security solutions in the IoT ecosystem. However, IoTs do provide a few unique characteristics to help support security development.

- Flexible hardware platform construction: The creation of highly diversified IoT hardware platforms is becoming possible due to the low cost of hardware components and open-source designs in hardware domains. On the other hand, general computing systems are often built on top of hardware platforms and processors which the designers can rarely customize. The wide usage of instruction set architecture (ISA), including MIPS [38], RISC-V [39], and ARM [40], makes it possible for IoT manufacturers to design and customize hardware platforms. As a consequence, hardware-supported cybersecurity protection schemes become popular solutions which can achieve high-efficiency and low-performance overhead for resource constraint IoT devices [41]–[44].
- Open-source systems for patch development: Open-source software programs are widely used in IoT development. In fact, the usage of open-source programs facilitates the quick development of software stack of IoT devices. The usage of similar open-source programs in various devices may bring similar vulnerabilities to a large set of different devices. However, the similarity often provides the opportunity to develop security patches through the support of the whole community. While the lack of security experts may prevent IoT manufacturers from developing security patches promptly, these manufacturers will benefit from the patches developed by the open-source community for their own devices, thus reducing the cost. A leading example is the security patches for Android systems developed by the cybersecurity community led by Google [45]. These efforts largely alleviate the work burden on smartphone manufacturers to quickly solve security threats.
- Faster patch installation: One of the main appeals of the IoT is that these systems are able to be upgraded remotely. Instead of having someone on site to upgrade a device, manufacturers are able to send updates over the air (OTA). This eliminates the need of replacing devices that utilize nonvolatile memory for firmware. In addition, this cuts down on infrastructure costs and increases reliability. Supported by the cloud, the update and patch distribution can be done in a centralized way so that users do not need to be involved. Some devices also require mandatory security updating such that all devices will be updated without known vulnerabilities available for attackers. This strategy is quicker and more efficient than the traditional strategy of upgrading computing systems where users have to be heavily involved.

In addition to the above, there are features of IoT that must be taken into consideration. These factors do not necessarily complicate or facilitate the solution, but must be accounted for carefully both to maximize potential benefits and avoid security loopholes.

- Always connected: One unique property of IoT is that all devices will be connected to the network constantly. This always-connected property enables easy patch installation but also opens the door for remote attacks. Though the channels for these updates are convenient, their security must be considered. Devices sometimes may not implement proper identification of updates, allowing arbitrary or even malicious updates to be accepted by the device, thereby compromising it. Possible countermeasures include signing firmware updates and utilizing a secure channel for server–device communication. These approaches can prevent attackers from intercepting an update, or determining when an update is occurring. They will also allow the system to verify that the firmware is from the legitimate manufacturers.
- Devices never connected before: An important feature of IoT is that it enables communication among physical objects that had never been connected before. For example, it might enable the car to “talk” to the refrigerator to determine the quantity of beer! It is critical for the security

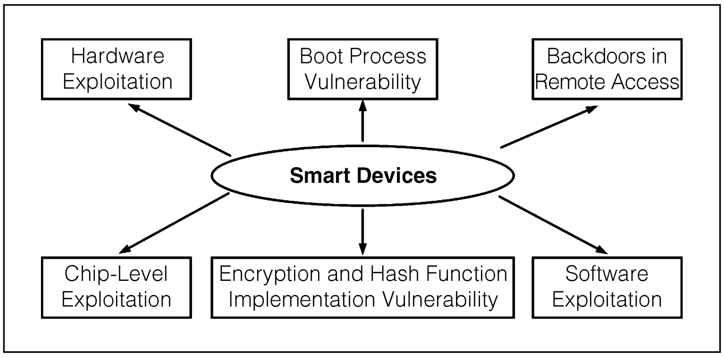
Hardware Exploitation
Boot Process Vulnerability
Backdoors in Remote Access
Smart Devices
Chip-Level Exploitation
Encryption and Hash Function Implementation Vulnerability
Software Exploitation

Figure 3. Security vulnerabilities in smart devices.

designer to think through the potential scenarios arising out of such communications, which were unforeseen before the IoT era. For example, perhaps it is possible to hack the refrigerator and identify how much a person drinks, thereby violating privacy issues. Identifying and mitigating such attacks within the context of billions of devices is a challenge.

- Cloud configuration: Almost all commercial IoT and many of the industrial IoT are supported by the cloud to provide services and to collect user information. It is obvious that the cloud makes the deployment of IoT much easier and also enables customized services. At the same time, cloud security will be closely related to IoT security given the mutual trust and data shared among the cloud and IoT devices. Cloud security is itself a large topic and will not be covered in this article [46], [47].
- Pairing and authentication: The large amount of IoT devices also complicates the device pairing and authentication procedures. It is very important that the pairing process is secure. Otherwise, the communication channels among device-to-device and device-to-router will be intercepted. Meanwhile, designers also need to consider power consumption given the frequency with which IoT devices pair and authenticate. For example, Bluetooth low power (BLE) was developed for low-power pairing and communication [48]. However, original implementations of BLE were proven to be insecure [49]. Low-power implementations will be inherently less secure because there is less power driving them. As a result, many devices utilize BLE for low-priority information transfer and high-power Bluetooth for more intensive information transfer, i.e., firmware updates [50].

Attacks on IoT devices

How can an IoT device be attacked? We categorize threats to smart devices into six types. A full taxonomy of these security threats is shown in Figure 3. Note that our categorization may be incomplete.

- Boot process exploitation: The boot sequence is one of the main targets where attackers try to bypass system level protection methods. During the boot process, many of the high-level protection mechanisms are not linked, installed, or enabled. Therefore, the protection of the boot process becomes critical for smart device security.
- Hardware exploitation: Hardware level exploitation is a newly proposed attack vector to smart devices given most security protection methods are located at the software or firmware level. Further, in order to increase a device's testability after manufacturing, debugging interfaces are widely installed in modern devices, however most of them are not protected leaving hardware backdoors open for exploitation.
- Chip-level exploitation: Chip-level exploitation of integrated circuits, including semi-invasive and invasive intrusions have become a serious threat to smart devices recently given that a trusted boot sequence is always starting from a trusted on-chip asset. Encryption/decryption keys and other sensitive information is stored on-chip which, for a long time, was considered a secure means of storage. Newly developed invasive methods may reveal valuable assets stored in the chip and compromise any protocols utilizing the secret information.
- Encryption and hash function implementations: Side-channel attacks, along with other physical information-based cryptoanalysis methods, have been threatening encryption and hashing algorithms which are otherwise proven to be mathematically secure and robust. Improper implementations and weak encryption algorithms are other security vulnerabilities present in modern encryption and hash functions which play a key role in device communication and authentication.

- Network and remote access channels: Smart devices are often equipped with remotely accessible channels for communication and debugging postmanufacturing. Remote access also makes OTA firmware upgrading a possibility. However, the remote access channels may be compromised, leaving attackers a channel to remotely obtain the status of the device or even control the device.
- Software exploitation: Software-level vulnerabilities of smart devices are similar to their counterparts in traditional embedded systems and general computing systems. Because the whole software stack of smart devices is derived from the general computing domain, any software vulnerabilities found in the general computing area will also apply to smart devices. Therefore, software patches are required frequently to update smart devices against known software-level attacks.

Finally, we note that attacks on devices is only a small part of potential security attacks in an IoT ecosystem. It is possible to attack the routers, gateways, and data centers, and any combination.

Power management challenges

Four decades of continuous scaling and the indisputable triumph of Moore's law have enabled a plethora of low-power computation and communication devices. Due to the mobile and sometimes standalone nature of these devices, powering them poses a new paradigm in power delivery and management solutions. The increasing demand for features and intelligence in IoT devices further exacerbates the problem. Fine-grained spatio–temporal power gating and clock gating are already in widespread use in the industry. However, we are now faced with decreasing die sizes, lower decoupling capacitance, multiple chip and platform power states, and an increasing number of power grids and migrating hotspots. Consequently, delivering power efficiently is a critical design challenge [51]–[53]. Further, the workload being executed on these devices demonstrates a huge variation in terms of both voltage and current. High-performance modes with operating supplies close to 1 V need to be supported along with near-theshold-voltage (NTV) operation. Research has started in earnest to explore novel circuits and control strategies in integrated direct current to direct current (dc–dc) converters and voltage regulators that can support such large dynamic ranges with the ability to make power state transitions in a few clock cycles. On the other hand, to improve the battery life and due to the increased momentum in the field of ambient energy harvesting, opportunistic energy harvesters are also becoming a reality. This adds another dimension to the challenge, as energy harvesters are variable and sporadic sources of power. We need to provide a platform and interface circuits for optimum power transfer at minimum losses. Traditional power delivery networks (PDNs) designed for servers, desktops and high end mobile phones are based upon worst case load condition. This approach is targeted for performance and therefore is rendered inefficient in the IoT world where power efficiency continues to play an ever-increasing role. Worst case designs are agnostic toward the wide scale variations both of load circuits (digital, analog, and RF) as well as energy sources. Therefore, it is critical to reevaluate and modify the strategy for designing PDNs for IoT devices. Adaptive and reconfigurable designs for components close to both source and load can be a viable and energy-efficient solution.

Power delivery architecture

Figure 4 shows a typical power flow architecture for an IoT device. The architecture consists of three important stages: the source, the PDN, and the load. In general, the source is a rechargeable battery. Over the last few years there has been a resurgence of energy harvesting devices. This has been facilitated by two factors: 1) the energy conversion efficiency of the harvesting transducers are increasing at a rapid pace; and 2) load circuits, particularly for IoT devices are demanding lower and lower power thereby narrowing the gap between the supply (harvesters) and the demand (load). Some of the important energy harvesters include photovoltaic, vibrational, thermoelectric, and wireless energy scavengers. The load consists of variety of circuits and components depending upon the application. Digital circuits could include CPU, GPU, memory, accelerators, audio and video processing blocks, etc. There are also a number of analog and RF blocks that are quintessential for a connected world. In between the energy sources

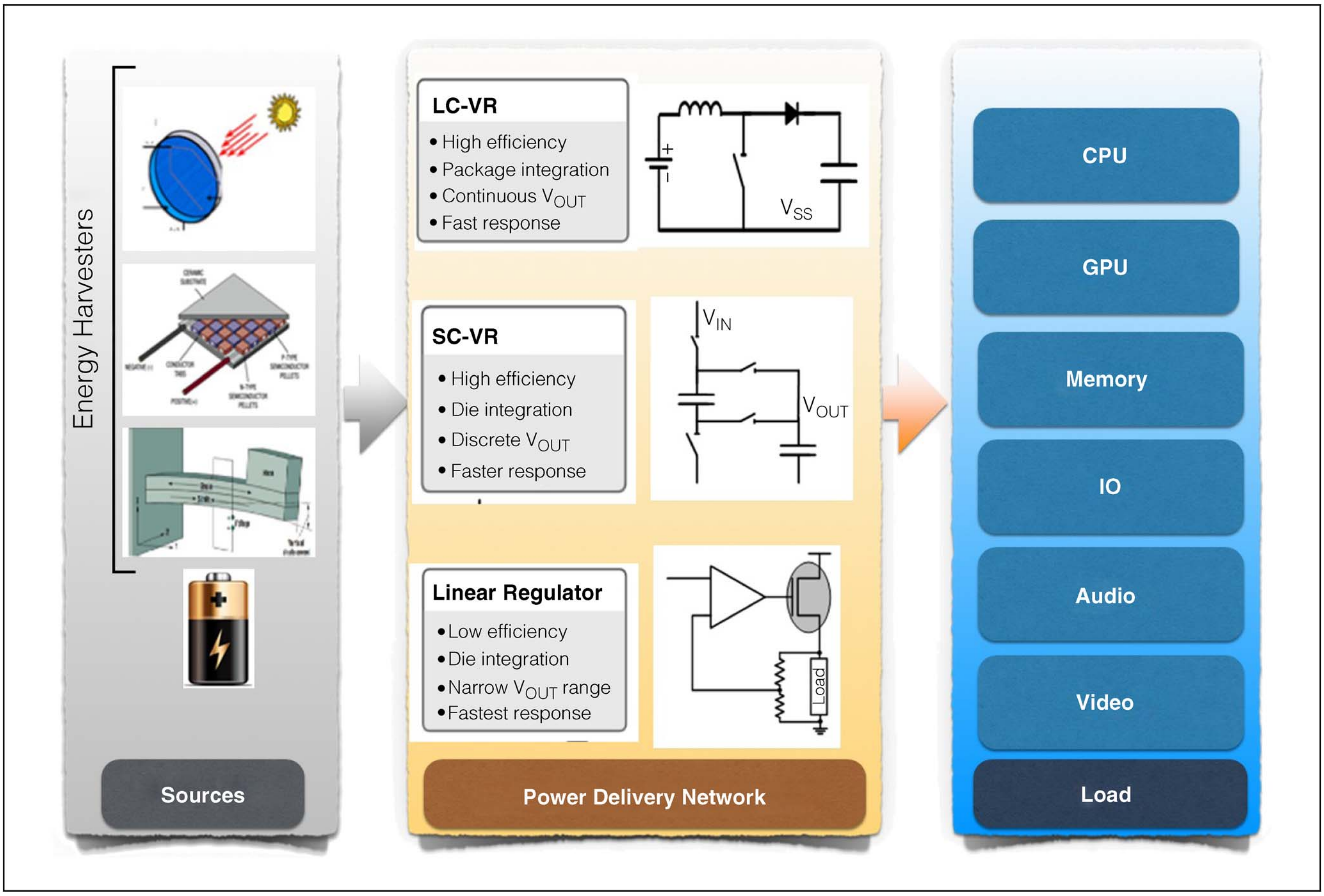

Figure 4. PDN for a typical IoT device showing the sources and load circuits and the components of dc–dc converters and regulators.

and the loads, we have a PDN whose primary task is to provide stability of the supply voltages, high power efficiency, excellent load and line regulation as well as maximum power transfer from the source to the load in a highly dynamic and ever-changing environment.

Components of the PDN

Before going into the details of adaptive designs suited for large dynamic ranges, let us briefly discuss the various PDN components in a representative design, as shown in Figure 5. These components can be classified broadly into switching dc–dc converters and integrated linear regulators.

- Switching dc–dc converters: Switched inductor (SL) [54]–[57] or switched capacitor (SC) converters [51]–[53], [58] are primarily used to step down high input voltage coming from secondary batteries to levels compatible with complementary metal–oxide–semiconductor (CMOS) logic. The converters in general feed to a single voltage regulator in case there is only one power domain or can provide voltage and current to several voltage regulators in case of multiple voltage domains. Such converters operate in a feedback loop to stabilize the output voltage independent of the load current. Although potentially both the SL and SC converters can be implemented on-die, each of these has its own unique challenge. SC converters have been demonstrated with on-die integrated capacitors, but they typically suffer from low capacitance density. Technology-circuit co-design has been explored to enable competitive SC converters. However, due to their limited applicability and cost (in terms of silicon area) commercially available designs typically employ SL converters. In SL converters, the inductor is either on the printed circuit board (PCB) or in

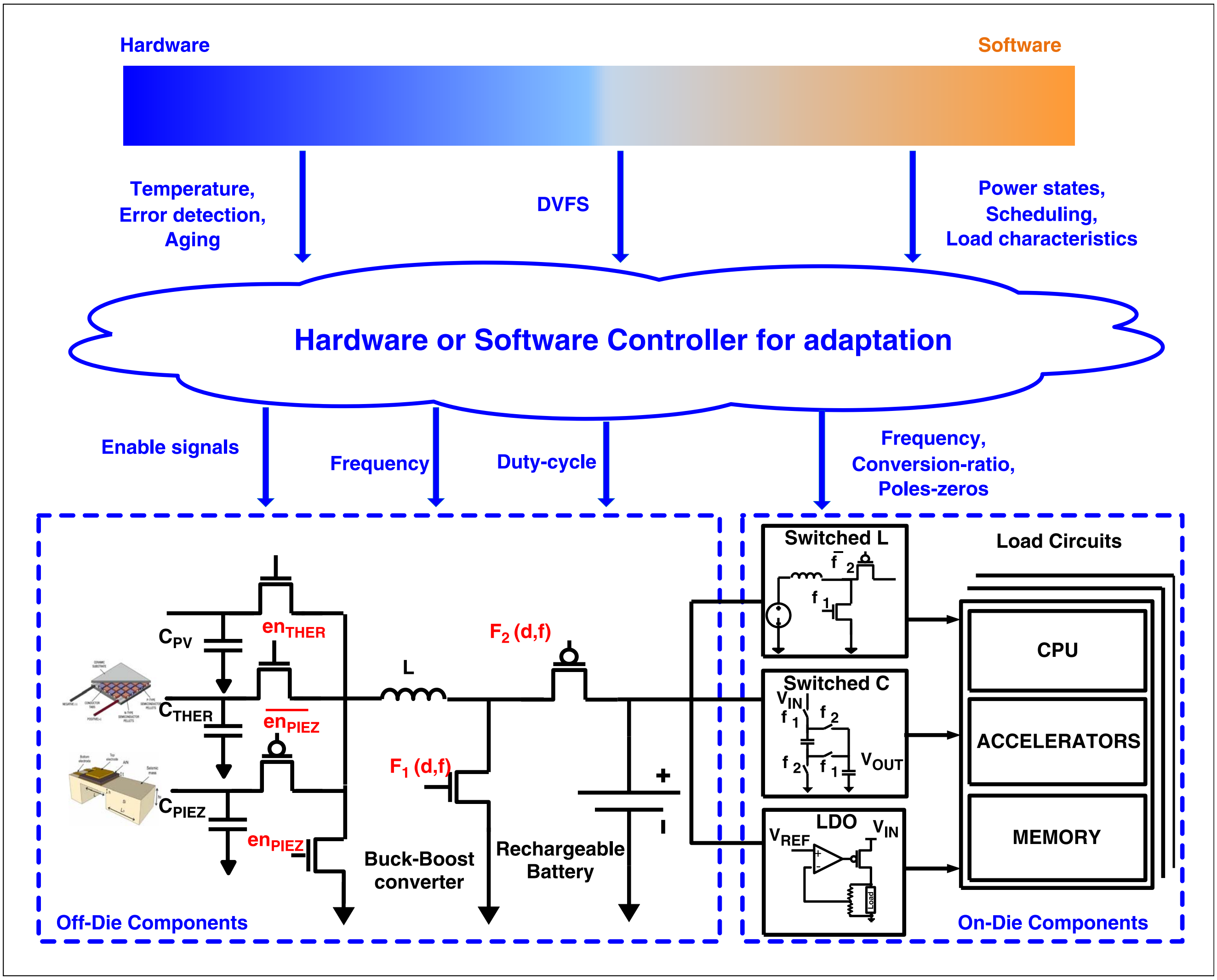

Figure 5. Hardware- and software-driven adaptation of various control knobs in the PDN can increase energy efficiency and system stability.

the package and provides high power density. It should be noted that SL converters when converting from the battery voltage (3.3–5 V) to CMOS compatible levels (≈ 1 V), operate in a buck or step-down mode. When operated to boost the voltage level (typically from a harvester) to either supply the load or to charge a secondary battery, SL converters operate as boost converters. Extensive work has been done for both boost and buck topologies and they continue to be the mainstays of the PDN.

- Point of load-voltage regulators: The output of switching inverters having ripple, although advanced, techniques such as time-interleaving and multiphased designs can alleviate a part of the problem. To supply a constant voltage to the load, and suppress any ripple, linear voltage regulators at the point of load (PoL) are the most popular design choices [59]–[62]. As we move into a domain of ultrafine-grained spatio–temporal power management, linear regulators continue to be distributed across the die. Linear regulators provide regulation by dynamically changing the resistance of an active series resistor to maintain a constant voltage across the load. Hence, they are inherently lossy and can only be as efficient as V_{OUT}/V_{IN} where V_{OUT} and V_{IN} are the output and input voltages of the linear regulator, respectively. An important class of efficient linear regulators are low-dropout regulators (LDOs) whose drop-out voltage ($V_{\mathrm{IN}} - V_{\mathrm{OUT}}$) can be as low as 50 mV. The last couple of decades have seen continuous improvement in the design, implementation, and integration of analog linear regulators (including LDOs). They exhibit high load/line regulation, high bandwidth as well as high power supply rejection. However, with continuous lowering of V_{IN}, analog linear regulators are losing their ranges of application. Current research focuses on supplementing analog linear regulators with synthesizable, process and voltage scalable, all-digital linear regulators that have been demonstrated to enable fast response at extremely low controller currents. Both analog and digital loops have also been incorporated to provide high bandwidth as well as high energy efficiency.

Maximum power transfer from energy sources

Secondary batteries supply power at a constant voltage. However, energy harvesters exhibit widely varying output voltage, current, power, and impedance levels [63]–[67]. The table below captures the typical energy harvesters and the ranges of output voltages and currents of Thevenin Equivalent sources.

Type of Energy Harvester	Open Circuit Voltage (V_{OL})	Short Circuit Current (I_{SC})
Photovoltaic	0.58-0.62 V	0.9-8.6 A
Thermoelectric	0.42-2.75 V	0.5-2.4 mA
Vibrational	4-12 V	0.1-0.29 mA

Figure 6 shows a typical boost converter that interfaces with an energy harvester and serves as the maximum power extractor. Let us consider a typical design to illustrate the key design challenges and solutions [68]–[70]. In order to operate the boost converter at low power, the discontinuous conduction mode is typically chosen [68]. During the first switching phase ϕ, the inductor charges to maximum inductor current for time t_1 and in the next phase ϕ_2 it discharges down to 0 for time t_2. After that there is a dead period where the inductor does not conduct any current as both the paths are cut off. In such a case, the input resistance of the harvester (equivalent to the driving point impedance) can be modeled as

$$R_{\mathrm{IN}} = \frac{V_{\mathrm{EH}}}{I_{\mathrm{IN}}} = \frac{2L}{t_1 * (t_1 + t_2) * f} = \frac{2L}{t_1^2 * f} \qquad (1)$$

where f is the switching frequency and L is the inductance. The last part of the equation assumes that $t_2 \ll t_1$ which is generally true because the harvester may require a boosting ratio as high as 10×. The boosting ratio is given by

$$\text{boosting ratio} = \frac{V_{\mathrm{OUT}}}{V_{\mathrm{IN}}} = \frac{t_1}{(t_1 + t_2)}. \qquad (2)$$

This would also hold for buck-boost converters because the harvester provides power only in the first phase and charges the inductor. Similar analysis can be carried out for continuous conduction mode which is typical for higher power ratings.

To extract the maximum power from a harvester, the looking-in impedance (resistance) R_{IN} offered by the boost converter needs to be equal to the internal resistance of the energy harvester. This is critical when we are using multiple energy harvesters because different harvesters will have different output impedances. While a static network would not be able to handle multiple energy harvesters, an adaptive PDN can utilize the boost converter (or buck-boost converter) and offer a suitable impedance by changing either the switching frequency or the time interval. Now consider the case where the same energy harvester operates at different voltage levels due to static and dynamic variations. When faced with different voltage levels the designer can choose to allow the boosting ratio to remain constant in a static design, but this can have significant implications on the system efficiency. This is particularly true when the dc–dc converter is followed by a linear regulator with a large dropout. The theoretical maximum efficiency an LDO can provide is the ratio of regulated voltage and the input voltage of the LDO and is given by

$$\eta = \frac{V_{\mathrm{OUT}}}{V_{\mathrm{IN}}} \cdot \frac{I_{\mathrm{LOAD}}}{I_{\mathrm{LOAD}} + I_{\mathrm{CONTROLLER}}}. \qquad (3)$$

The second ratio (the ratio of currents) is called the current efficiency of the linear regulator. Let us consider an example. As shown in Figure 6, we consider a thermoelectric energy harvester with a static dc–dc converter. The design assumes worst case corner so the assumption is based on the fact that the thermoelectric energy harvester is producing a low open circuit voltage of ≈0.05 V. However, during nominal usage, the output voltage could assume that the load current delivered by the harvester is constant. If the boosting ratio of the dc–dc boost converter is kept constant and the voltage demand on the load circuit is unchanged, any extra voltage generated by the boost converter must be dropped across the linear regulator. This leads to significant energy loss in the system. Alternatively, an adaptive design would be able to adjust the boosting ratio to minimize losses across the PDN. Our calculations show that when the harvester delivers high power and voltage the system-level power efficiency drops to 23. 75% and in nominal mode of operation it is slightly lower than 50%. Hence, in a worst case, static design cannot fully comprehend the opportunities presented by a dynamic environment and an adaptive PDN can reduce loss of pessimism in design. One simple circuit level solution to improve this drastic drop in energy efficiency is to change the boosting ratio by changing t_2 and keeping $t_1 + t_2$ constant. The control circuitry required would have a feedback system which constantly monitors the output voltage and keeps increasing t_2 until the boosting ratio meets a target. Details of such a design are left for interested readers for further study. Alternative design styles can also be adopted, as long as adaptation to the dynamic environment is sensed and the PDN adjusted based on the current supply and demand constraints. Utilizing this technique would ensure that high energy efficiency is maintained across the large dynamic range of V_{EH}. Summary of typical analytical results is shown in Figure 7.

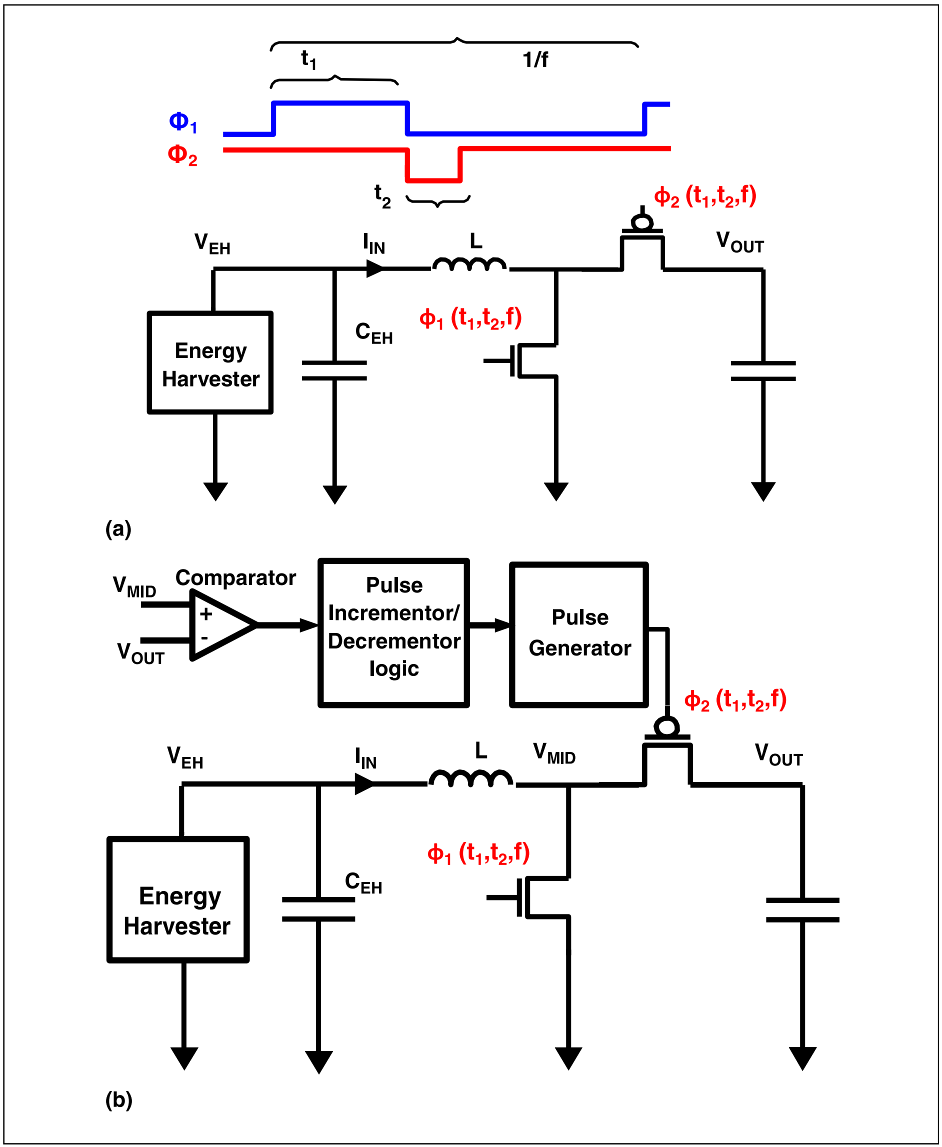

Figure 6. (a) A static switched inductor boost converter. (b) An adaptive design for maximum power transfer from an energy harvesting source.

Dynamic point-of-load regulators for varying load currents

To achieve high power efficiency without sacrificing performance the loads circuits employ dynamic voltage and frequency scaling (DVFS). Also, fine-grained clock and power gating are used to minimize power consumption in the unused domains of the load circuit. Further, changes in temperature and aging can cause changes in the load current ranges both spatially as well as in time. Considering that the IoT devices would need to be economically designed, low-cost packaging adds variations and further increases the load current ranges. A typical core in a normal processor can demonstrate current variations from a few milliamperes to a few microamperes. A simple smart watch, for example, can demonstrate current

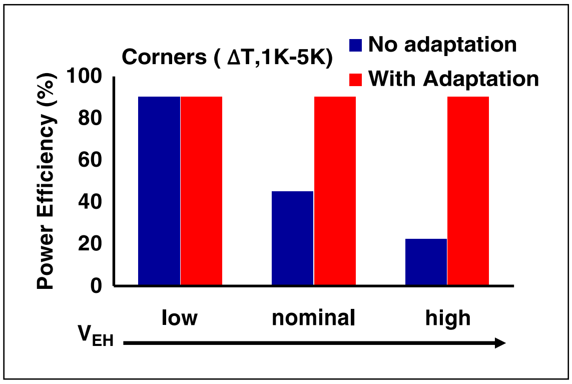

Figure 7. Power efficiency of the PDN for different harvesting voltages, both with and without adaptation.

ranges from 50 to 100 mA to a few microamperes. So modern PDNs not only need to supply voltages across multiple operating voltages but also need to supply large ranges of current with high stability, low ripple, fast response times, and low line/load regulations.

On-chip integrated linear regulators, operated in low dropout modes (LDO VRs) are widely used to provide consistent and well-regulated voltage to the load circuits [51], [59]–[61], [71], [72]. On-chip LDO VRs play a critical role by stepping down the noisy input voltage and providing a clean supply to the loads. The LDOs are fast and their integration at the chip level is well understood. However, the LDO power efficiency is dependent on two crucial components, as shown in (3). It is directly related to the ratio of V_{OUT} to V_{IN}. So the higher is the dropout ($V_{\mathrm{OUT}} - V_{\mathrm{IN}}$), the lower is the maximum efficiency possible. Another component that adds to this is the current efficiency. At higher load currents the controller current of the digital LDOs is insignificant and it is nearly equal to one. However, during the light load scenarios, the controller current can become a significant portion of the overall current and lower the net efficiency. This renders the traditional static design of LDOs pessimistic. Current research addresses this problem by allowing the control loop of an LDO to adapt and change itself depending on the load current and operating conditions. Apart from the loss of efficiency due to the extremely large current ranges, another important design criterion is the loop stability. The LDO loop needs to be stable amidst large changes in the load current. It can be qualitatively understood that changes in the load current would result in changes in the output pole position of the LDO loop (a larger load current would push the output pole to a higher frequency). Even for a well-compensated loop, it is often difficult to achieve high phase/gain margin for a 100x change in the load current (and hence the load pole position). This requires adaptation to changes in the loop characteristics depending on the output current and the output pole position. This removes pessimism, addresses the issues of worst case design, and can guarantee high efficiency and stability across a 50–100x load current range.

Research has started to address critical efficiency and stability issues in linear regulators. Both analog regulators as well as fully digital regulators are being studied. By adapting the passive components of a feedback network with varying load current conditions, target stability has been demonstrated in analog regulators [73]. Conversely, digital regulators have been demonstrated with stable response over a large voltage and current dynamic ranges where the sampling frequency has been allowed to adapt to the operating conditions. A typical example has been shown in Figure 8 and interested readers are referred to [61] for further reading.

This ensures a placement of the closed-loop system poles in the region of stability and fast transient response has been reported across a wide range of operation. As we move into an era of ultralow-power IoT devices, designers are expected to take advantage of closed-loop control in power distribution and management and engineer systems where computation is performed both "just in time" and "with just enough energy." We need further improvements in PDN technologies, better computer-aided design (CAD) tools, and hardware designs that will perform co-optimization of PDN components with the load and sources. This will allow improvements in system level efficiencies, robustness toward variations, and tolerance towards low-cost, variation-prone package components.

We have provided an overview of IoT to help readers have a systematic view of how it has evolved and where we are going in this space. More importantly, through our introduction of the IoT developing trend as well as the specific topics

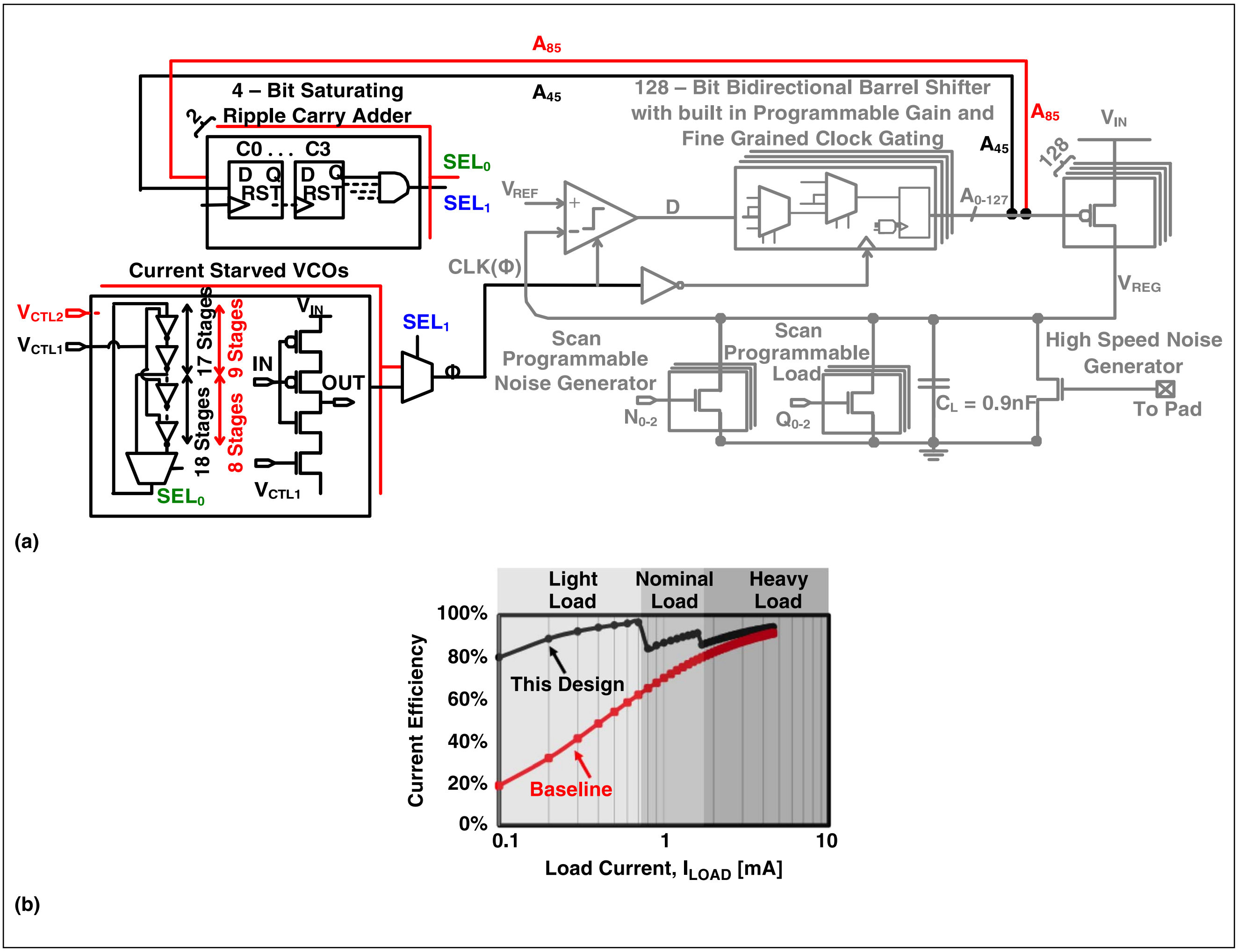

Figure 8. (a) An all-digital adaptive LDO. (b) Current efficiency of an adaptive LDO *vis-a-vis* a baseline LDO showing improved current efficiency across a wide dynamic range [61].

in IoT security and IoT energy efficiency, we have outlined some of the critical growth and research areas in IoT development, both from academic and industrial views. We hope more researchers and engineers can join this area and address the goals of high computation efficiency, high security, low cost, and easy deployment. Our objective for this article has been to serve as the starting point to help build such IoT systems.

Although this article covers a wide spectrum of topics related to IoT, we have only scratched the surface of challenges and research opportunities. The research questions cross-cut several computing disciplines, including programming languages, computer architecture, physical designs, security, algorithms, and analytics. There are clear indications that innovation in this area would come from cross fertilization and blending of different areas. To researchers looking for cross-collaborative, multidisciplinary research topics these are exciting times. On the other hand, research progresses smoothly only when problems can be clearly compartmentalized into topics which can be individually explored. The fact that we cannot do that for this area suggests both the complexity of the subject and also perhaps its relative immaturity. Perhaps only by systematic, interdisciplinary collaboration among researchers and practitioners we will develop critical abstractions that permit definition of effective topic silos. ■

■ References

[1] D. Evans, “The Internet of Things—How the next evolution of the internet is changing everything,” Cisco Internet Business Solutions Group (IBSG), white paper, 2011.

[2] B. Edson, “Creating the internet of your things,” Microsoft Corporation, 2014.

[3] Intel Corporation, “The Internet of Things (IoT) starts with Intel inside.” [Online]. Available: http://www.intel.com/iot/

[4] IBM, “Watson Internet of Things.” [Online]. Available: http://www.ibm.com/internet-of-things/

[5] R. Chitkara and W. Ballhaus, “The Internet of Things: The next growth engine for the semiconductor industry,” PricewaterhouseCoopers, 2015.

[6] “Internet of Things done wrong stifles innovation,” *InformationWeek*, 2014.

[7] M. Weiser, “The computer for the 21st century,” *Sci. Amer.*, vol. 265, no. 3, pp. 2014, 1991.

[8] R. Raji, “Smart networks for control,” *IEEE Spectrum*, vol. 31, no. 6, pp. 49–55, Jun. 1994.

[9] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future Internet: The Internet of Things architecture, possible applications and key challenges,” in *Proc. 10th Int. Conf. Front. Inf. Technol.*, 2012, pp. 257–260.

[10] [Online]. Available: http://www.smarthome.com/

[11] F. Bonomi, “The smart and connected vehicle and the Internet of Things.” [Online]. Available: http://tf.nist.gov/seminars/WSTS/PDFs/1-0_Cisco_FBonomi_ConnectedVehicles.pdf

[12] [Online]. Available: http://www.fitbit.com/

[13] [Online]. Available: http://www.mybasis.com/

[14] M. Ersue, D. Romascanu, and J. Schoenwaelder, “Management of networks with constrained devices: Use cases,” IETF Internet Draft, 2014.

[15] S. Li, H. Wong, T. Xu, and G. Zhou, “Application study on internet of things in environment protection field,” in *Informatics in Control, Automation and Robotics*, Lecture Notes in Electrical Engineering. Berlin, Germany: Springer-Verlag, 2011, vol. 133, pp. 99–106.

[16] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, “From the Internet of Things to the Web of Things: Resource-oriented architecture and best practices,” in *Architecting the Internet of Things,* D. Uckelmann, M. Harrison, and F. Michahelles, Eds., New York, NY, USA: Springer-Verlag, 2011.

[17] *Internet Protocol, Version 4*, 1981, IETF RFC 791.

[18] *Internet Protocol, Version 6*, 1998, IETF RFC 2460.

[19] *IPv6 Addressing of IPv4/IPv6 Translators*, 2010, IETF RFC 6052.

[20] *Connection of IPv6 Domains via IPv4 Clouds*, 2001, IETF RFC 4944.

[21] G. Mulligan, “The 6lowpan architecture,” in *Proc. 4th Workshop Embedded Netw. Sensors*, 2007, pp. 78–82.

[22] *Transmission of IPv6 Packets Over IEEE 802.15.4 Networks*, 2007, IETF RFC 3056.

[23] Apple HomeKit. [Online]. Available: http://www.apple.com/ios/homekit/

[24] J. Burt, “Cisco moving apps to the network edge for Internet of Things,” *eWeek*, 2014. [Online]. Available: http://www.eweek.com/networking/cisco-moving-apps-to-the-network-edge-for-internet-of-things.html

[25] Google Nest. [Online]. Available: https://nest.com/

[26] Node-RED. [Online]. Available: http://nodered.org/

[27] Nitrogen. [Online]. Available: http://nitrogen.io/

[28] J. Goguen, and J. Meseguer, “Security policies and security models,” in *Proc. IEEE Symp. Security Privacy*, 1982, pp. 11–20.

[29] J. Rushby, “Noninterference, transitivity, channel-control security policies,” SRI, Tech. Rep., 1992.

[30] S. J. Greenwald, “Discussion topic: What is the old security paradigm,” in *Proc. Workshop New Security Paradigms*, 1998, pp. 107–118.

[31] S. Ray, J. Yang, A. Basak, and S. Bhunia, “Correctness and security at odds: Post-silicon validation of modern SoC designs,” in *Proc. 52nd Int. ACM/EDAC/IEEE Design Autom. Conf.*, 2015, DOI: 10.1145/2744769.2754896.

[32] A. Basak, S. Bhunia, and S. Ray, “A flexible architecture for systematic implementation of SoC security policies,” in *Proc. 34th Int. Conf. Comput.-Aided Design*, 2015, pp. 536–543.

[33] S. Ray and Y. Jin, “Security policy enforcement in modern SoC designs,” in *Proc. 34th Int. Conf. Comput.-Aided Design*, 2015, pp. 435–530.

[34] J. Wurm, O. Arias, K. Hoang, A.-R. Sadeght, and Y. Jin, “Security analysis on consumer and industrial IoT devices,” in *Proc. 21st Asia South Pacific Design Autom. Conf.*, 2016, pp. 519–524.

[35] O. Arias, J. Wurm, K. Hoang, and Y. Jin, “Privacy and security in internet of things and wearable devices,” *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 99–109, 2015.

[36] AWS IoT. [Online]. Available: https://aws.amazon.com/iot/

[37] MQ Telemetry Transport (MQTT). [Online]. Available: http://mqtt.org/

[38] G. Kane and J. Heinrich, *MIPS RISC Architectures* Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[39] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic, “The Risc-v instruction set manual, Volume I: Base user-level ISA,” Electr. Eng. Comput. Sci. Dept., Univ. California Berkeley, Tech. Rep. UCB/EECS-2011-62, 2011.

[40] D. Seal, *ARM Architecture Reference Manual* New York, NY, USA: Pearson, 2001.

[41] ARM Limited, “Building a secure system using trustzone technology,” 2009.

[42] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, “Using innovative instructions to create trustworthy software solutions,” in *Proc. 2nd Int. Workshop Hardware Architect. Support Security Privacy*, 2013, DOI: 10.1145/2487726.2488370.

[43] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, “Innovative technology for CPU based attestation and sealing,” in *Proc. 2nd Int. Workshop Hardware Architect. Support Security Privacy*, 2013, pp. 1–7.

[44] L. Davi et al., “HAFIX: Hardware-assisted flow integrity extension,” in *Proc. 52nd Annu. Design Autom. Conf.*, 2015, pp. 74:1–74:6.

[45] The Android mobile OS by Google. [Online]. Available: https://www.android.com/

[46] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance.* New York, NY, USA: O’Reilly Media, 2009.

[47] B. Kandukuri, V. Paturi, and A. Rakshit, “Cloud security issues,” in *Proc. IEEE Int. Conf. Services Comput.*, 2009, pp. 517–520.

[48] Bluetooth Low Energy. [Online]. Available: https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy

[49] M. Ryan, “Bluetooth: With low energy comes low security,” in *Proc. 7th USENIX Conf. Offensive Technol.*, 2013. [Online]. Available: https://www.usenix.org/conference/woot13/workshop-program/presentation/Ryan

[50] Bluetooth Core Specification. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification

[51] H.-P. Le, J. Crossley, S. R. Sanders, and E. Alon, “A sub-ns response fully integrated battery-connected switched-capacitor voltage regulator delivering 0.19 w/mm ^{2} at 73% efficiency,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 372–373.

[52] M. Ang, R. Salem, and A. Taylor, “An on-chip voltage regulator using switched decoupling capacitors,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2000, pp. 438–439.

[53] B. Axelrod, Y. Berkovich, and A. Ioinovici, “Switched-capacitor/switched-inductor structures for getting transformerless hybrid dc-dc PWM converters,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 687–696, 2008.

[54] N. Sturcken et al., “A 2.5 d integrated voltage regulator using coupled-magnetic-core inductors on silicon interposer delivering 10.8 a/mm ^{2},” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2012, pp. 400–402.

[55] N. Kurd et al., “Haswell: A family of IA 22 nm processors,” *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 49–58, 2015.

[56] E. Burton et al., “Fivr—Fully integrated voltage regulators on 4th generation Intel Core SoCs,” in *Proc. 29th Annu. IEEE Appl. Power Electron. Conf. Expo.*, 2014, pp. 432–439.

[57] G. Patounakis, Y. W. Li, and K. L. Shepard, “A fully integrated on-chip dc-dc conversion and power management system,” *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 443–451, 2004.

[58] K. Ngo and R. Webster, “Steady-state analysis and design of a switched-capacitor dc-dc converter,” in *23rd Annu. IEEE Power Electron. Specialists Conf. Record*, 1992, pp. 378–385.

[59] S. B. Nasir and A. Raychowdhury, “On limit cycle oscillations in discrete-time digital linear regulators,” in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2015, pp. 371–376.

[60] S. Gangopadhyay, S. B. Nasir, and A. Raychowdhury, “Integrated power management in IoT devices under wide dynamic ranges of operation,” in *Proc. 52nd Annu. Design Autom. Conf.*, 2015, pp. 149.

[61] S. Bin Nasir, S. Gangopadhyay, and A. Raychowdhury, “A 0.13 μm fully digital low-dropout regulator with adaptive control and reduced dynamic stability for ultra-wide dynamic range,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2015, DOI: 10.1109/ISSCC.2015.7062944.

[62] R. J. Milliken, J. Silva-Martínez, and E. Sánchez-Sinencio, “Full on-chip CMOS low-dropout voltage regulator,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 9, pp. 1879–1890, Sep. 2007.

[63] S. P. Beeby, M. J. Tudor, and N. White, “Energy harvesting vibration sources for microsystems applications,” *Meas. Sci. Technol.*, vol. 17, no. 12, pp. R175–R195, 2006.

[64] G. K. Ottman et al., “Adaptive piezoelectric energy harvesting circuit for wireless remote power supply,” *IEEE Trans. Power Electron.*, vol. 17, no. 5, pp. 669–676, 2002.

[65] G. K. Ottman et al., “Optimized piezoelectric energy harvesting circuit using step-down converter in discontinuous conduction mode,” *IEEE Trans. Power Electron.*, vol. 18, no. 2, pp. 696–703, 2003.

[66] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.

[67] N. Vlachopoulos, P. Liska, J. Augustynski, and M. Grätzel, “Very efficient visible light energy harvesting and conversion by spectral sensitization of high surface area polycrystalline titanium dioxide films,” *J. Amer. Chem. Soc.*, vol. 110, no. 4, pp. 1216–1220, 1988.

[68] Y. K. Ramadass and A. P. Chandrakasan, “A batteryless thermoelectric energy-harvesting interface circuit with 35 mv startup voltage,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2010, pp. 486–487.

[69] A. Chandrakasan et al., “Design considerations for distributed microsensor systems,” in *Proc. IEEE Custom Integr. Circuits*, 1999, pp. 279–286.

[70] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, “Energy harvesting from human and machine motion for wireless electronic devices,” *Proc. IEEE*, vol. 96, no. 9, pp. 1457–1486, Sep. 2008.

[71] S. Gangopadhyay, D. Somasekhar, J. W. Tschanz, and A. Raychowdhury, “A 32 nm embedded, fully-digital, phase-locked low dropout regulator for fine grained power management in digital circuits,” *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2684–2693, Nov. 2014.

[72] A. Raychowdhury, S. B. Nasir, and S. Gangopadhyay, “The role of adaptation and resiliency in computation and power management,” in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2014, pp. 74–79.

[73] I. Vaisband et al., “Distributed LDO regulators in a 28 nm power delivery system,” *Analog Integr. Circuits Signal Process.*, vol. 83, no. 3, pp. 295–309, 2015.

Sandip Ray is a Research Scientist at Strategic CAD Labs, Intel Corporation, Hillsboro, OR, USA. His research focuses on developing trustworthy embedded, mobile, and Internet-of-Things (IoT) systems through a synergy of architecture, synthesis, and validation techniques. Ray has a PhD from the University of Texas at Austin, Austin, TX, USA. He is a Senior Member of the IEEE and professional member of the Association for Computing Machinery (ACM).

Yier Jin is an Assistant Professor at the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL, USA. His research focuses on trusted embedded systems, hardware IP protection, and hardware/software codesign for security. Jin has a PhD in electrical engineering from Yale University, New Haven, CT, USA (2012). He is a member of the IEEE and the Association for Computing Machinery (ACM).

Arijit Raychowdhury is currently the ON Semiconductor Junior Associate Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include low-power digital and mixed-signal circuit design. Raychowdhury has a PhD in electrical and computer engineering from Purdue University, West Lafayette, IN, USA (2007).

■ Direct questions and comments about this article to Yier Jin, Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA.; yier.jin@eecs.ucf.edu.