Chapter 1
The IoT Landscape

1.1 What Is IoT?

The Internet of Things (IoT) has become a common news item and marketing trend. Beyond the hype, IoT has emerged as an important technology with applications in many fields. IoT has roots in several earlier technologies: pervasive information systems, sensor networks, and embedded computing. The term *IoT system* more accurately describes the use of this technology than does *Internet of Things*. Most IoT devices are connected together to form purpose-specific systems; they are less frequently used as general-access devices on a worldwide network.

IoT moves beyond pervasive computing and information systems, which concentrated on data. Smart refrigerators are one example of pervasive computing devices. Several products included built-in PCs and allowed users to enter information about the contents of their refrigerator for menu planning. Conceptual devices would automatically scan the refrigerator contents to take care of data entry. The use cases envisioned for these refrigerators are not so far removed from menu planning applications for stand-alone personal computers.

Sensor network research spanned a range of configurations. Many of these were designed for data collection at very low data rates. The collected data would then be sent to servers for processing. Traditional sensor network research did not emphasize in-network processing.

Embedded computing concentrated on either stand-alone devices or tightly coupled networks such as those used in vehicles. Consumer electronics and cyber-physical systems were two major application domains for embedded computing; both emphasized engineered systems with well-defined goals.

Given the wide range of advocates for IoT technology, no single, clear definition of the term has emerged. We can identify several possibilities:

- Internet-enabled physical devices, although many devices don't use the Internet Protocol

D. Serpanos, M. Wolf, *Internet-of-Things (IoT) Systems*,
https://doi.org/10.1007/978-3-319-69715-4_1

- Soft real-time sensor networks
- Dynamic and evolving networks of embedded computing devices

This book is primarily interested in *IoT systems*. We use this term to capture two characteristics. First, the system is designed for one or a set of applications, rather than being an agglomeration of Internet-enabled devices. Second, the IoT system takes into account the dynamics of physical systems. An IoT system may consist primarily of sensors; in some cases it may include a significant number of actuators. In both cases, the goal is to process signals and time-series data.

Interest in the Internet of Things has been spurred by the availability of micro-electromechanical (MEMS) sensors. Integrated accelerometers, gyroscopes, chemical sensors, and other forms of sensor are now widely available. The low cost and power consumption of these sensors enables new applications well beyond those of traditional laboratory or industrial measurement equipment. These sensor applications push IoT systems toward signal processing.

IoT is also enabled by the very low cost of VLSI digital and analog electronics. As we will see, IoT nodes do not rely on state-of-the-art VLSI manufacturing processes. In fact, they are inexpensive because they are able to make use of older manufacturing lines; the lower device counts available in these older technologies are more than sufficient for many IoT systems.

IoT systems must consume very little power. Power consumption is a key factor in total cost of ownership for IoT systems. Achieving the necessary power levels requires careful attention to hardware design, software design, and application algorithms.

Security and safety are key design and operational requirements for IoT systems. As we have argued elsewhere, safety and security are no longer separable problems. The merger of computational and physical systems requires us to merge the previously separate tasks of safe physical system design and secure computer system design.

1.2 Applications

IoT systems are useful in a broad range of applications:

- Industrial systems use sensors to monitor both the industrial processes themselves – the quality of the product – and the state of the equipment. An increasing number of electric motors, for example, include sensors that collect data used to predict impending motor failures.
- Smart buildings use sensors to identify the locations of people as well as the state of the building. That data can be used to control heating/ventilation/air conditioning systems and lighting systems to reduce operating costs. Smart buildings and structures also use sensors to monitor structural health.
- Smart cities use sensors to monitor pedestrian and vehicular traffic and may integrate data from smart buildings.

- Vehicles use networked sensors to monitor the state of the vehicle and provide improved dynamics, reduced fuel consumption, and lower emissions.
- Medical systems connect a wide range of patient monitoring sensors that may be located at the home, in emergency vehicles, the doctor's office, or the hospital.

Use cases help us understand the requirements on an IoT system.

Sensor network The system may act strictly as a data gathering system for a set of sensors.

Alert system Data from sensors may be gathered and analyzed. Alerts are generated when particular criteria are met.

Analysis system Data from sensors is gathered and analyzed, but in this case, the analysis is ongoing. Reports on analytic results may be generated periodically – hourly, daily, *etc.* – or may be continuously updated.

Reactive system Analysis of sensor data may cause actuators to be triggered. We reserve the term *reactive* for systems that don't implement typical control laws.

Control system Sensor data is fed to control algorithms that generate outputs for actuators.

We can identify a class of nonfunctional requirements that apply to many IoT systems. Nonfunctional requirements on the system impose nonfunctional requirements on the components.

Event latency Latency from capture of an event to its destination may not be important for batch-oriented applications but becomes important for online analysis.

Event throughput The rate at which events can be captured, transported, and processed depends on the throughput of the nodes, network bandwidth, and cloud throughput.

Event loss rate and buffer capacity In the absence of strict upper bounds on event production rates, the environment may produce more events in an interval than the system can produce. Event loss rate captures the desired capability, while buffer capacity is a more pragmatic requirement that can be directly tied to component capabilities.

Service latency and throughput Ultimately, events will be processed by services. We can also specify the latency and throughput for services.

Reliability and availability Since IoT systems are distributed, reliability is more likely to be specified over parts of the network rather than reliability of the complete system. Availability is commonly used to describe distributed systems.

Service lifetime IoT systems are often expected to have longer lifetimes than we expect for PC systems. The lifetime of the system or a subset of the system may be considerably longer than that of a component, particularly if the system uses redundant sensors and other components.

1.3 Architectures

A key aspect of IoT is *event-driven or aperiodic sampling*. Traditional digital signal processing and control assume periodic samples resulting in time-series data. However, time series consume too much power at the nodes and too much bandwidth on the network. Not all applications are amenable to aperiodic data acquisition.

Constraints on power and bandwidth also encourage distributed computing over sensor events. Relatively small processors can perform useful processing on many data streams. Recognizing interesting events using edge processing reduces the amount of network bandwidth consumed; it also reduces power consumption since wireless communication requires large amounts of power. Cloud computing-(centralized servers) or fog computing (servers closer to the edge) can be used to perform further processing on those extracted events.

1.4 Wireless Networks

Wireless networks are integral to IoT systems. Wireless network connections simplify installation and operation of wireless networks.

However, wireless networks introduce some important problems and restrictions. Radio communication requires more power than does wired communication. Some of the wireless networks used in today's IoT devices were designed for other purposes, such as telephony and multimedia. As a result, they are not optimized for event-driven communication and consume significant amounts of power in the communications protocol.

One of the ironies of IoT is that many edge devices and their wireless networks don't operate on the Internet Protocol (IP). IP introduces significant overhead with an extra level of packetization and associated processing. Many IoT devices avoid IP and rely on upstream nodes to provide them with an Internet presence.

IoT networks are typically run by noncomputer experts. IoT wireless networks must be easy to deploy and relatively self-managing.

1.5 Devices

The characteristics of event-driven systems allow IoT nodes to be relatively simple. The realities of low-power operation also push nodes toward relatively low levels of integration.

VLSI technology and Moore's law are key factors in the rise of IoT systems because they allow nodes to be manufactured extremely cheaply. Very small chips can provide enough computation, memory, and networking for useful IoT node

functions. In contrast to traditional microprocessor and consumer electronic applications, where chip areas range around or even higher, chips of several square millimeters are large enough for many IoT node devices.

1.6 Security and Privacy

Security has finally been recognized as an essential requirement for all types of computer systems, including IoT systems. However, many IoT systems are much less secure than typical Windows/Mac/Linux systems. IoT security problems stem from a range of causes: inadequate security features in hardware, poorly designed software with a range of vulnerabilities, default passwords, and other security design errors.

Insecure IoT nodes create problems for the security of the entire IoT system. Because nodes typically have lifetimes of several years, the large installed base of insecure devices will create security problems for some time to come.

Insecure IoT systems also cause security problems for the rest of the Internet. IoT devices are plentiful; insecure IoT nodes are ideally suited to denial-of-service attacks. The Dyn attack [Sch16] is one example of an IoT-based attack on traditional Internet infrastructure.

Privacy is related to security but requires specific measures at the application, network, and device levels. Not only must user data be protected from outright theft, but the network needs to be designed so that less-private data cannot easily be used to infer more private data.

1.7 Event-Driven Systems

We believe that the *event* is a fundamental data type in IoT systems and that *event-driven systems* are an important structuring technique for IoT. Many of the building block technologies used for IoT today show some holdover from traditional, transaction-oriented systems. Event processing pushes us to treat time as a first-class concept and to consider the relationship between events in event sequences.

We use the term *event* more broadly then do simulation engineers. We consider events as *time-value sets*. Event-driven system simulation is widely used for modeling a wide range of engineering systems. In that context, an event is generally used to mean a change in the state of a variable. Given the decentralized nature of IoT systems, we are willing to consider stuttering – the repetition of an event value – as part of the event model. We also use events to model sampled data and time-series data. We believe that all these uses of the term event can be unified to create rich system structures.

1.8 This Book

The rest of this book describes a range of topics in IoT systems in more detail:

- Chapter 2 studies IoT system architectures, including wireless networks.
- Chapter 3 considers VLSI IoT devices. It describes the relationship between cost of ownership, power consumption, and duty cycle.
- Chapter 4 introduces analysis methods for event-driven IoT systems. These analysis methods allow us to study the memory requirements implied by event communication and processing.
- Chapter 5 describes the Industrial Internet of Things and applications of IoT systems in smart energy systems.
- Chapter 6 studies security and safety issues in IoT systems. Computer and cyber-physical system security is closely tied to safety in sensor and closed-loop control systems.
- Chapter 7 describes fuzz testing, a technique for testing the security of IoT systems. Bugs and crashes can provide exploits for attackers; fuzz testing is designed to help identify such problems.

Reference

[Sch16] Schneier, B. (2016, October 22). DDoS attacks against Dyn. *Schneier on Security.* https://www.schneier.com/blog/archives/2016/10/ddos_attacks_ag.html