FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

A Small Contribution Would Support Us.

Dear Viewer,

Future Vision BIE is a free service and so that any Student/Research Personal **Can Access Free of Cost**.

If you would like to say **thanks**, you can make a **small contribution** to the author of this site.

Contribute whatever you feel this is worth to you. This gives **us support** & to bring **Latest Study Material** to you. After the Contribution Fill out this Form (https://forms.gle/tw3T3bUVpLXL8omX7). To Receive a **Paid E-Course for Free**, from our End within 7 Working Days.

Regards

- K B Hemanth Raj (Admin)

Contribution Methods

UPI ID

1. futurevisionbie@oksbi
2. futurevisionbie@paytm

Account Transfer

Account Holder's Name: K B Hemanth Raj

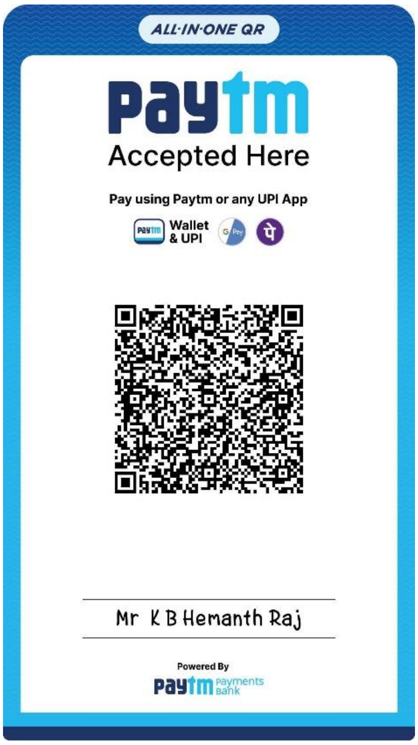
Account Number: 39979402438

IFSC Code: SBIN0003982

MICR Code: 560002017

More Info: https://hemanthrajhemu.github.io/Contribution/

Scan & Pay



Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering & MORE...

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE

IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things

David Hanes, CCIE No. 3491
Gonzalo Salgueiro, CCIE No. 4541
Patrick Grossetete
Robert Barton, CCIE No. 6660, CCDE No. 2013:6
Jerome Henry, CCIE No. 24750

Cisco Press
800 East 96th Street
Indianapolis, Indiana 46240 USA

Contents

Part I

Introduction to IoT

This page intentionally left blank

Chapter 1

What Is IoT?

Imagine a world where just about anything you can think of is online and communicating to other things and people in order to enable new services that enhance our lives. From self-driving drones delivering your grocery order to sensors in your clothing monitoring your health, the world you know is set to undergo a major technological shift forward. This shift is known collectively as the Internet of Things (IoT).

The basic premise and goal of IoT is to "connect the unconnected." This means that objects that are not currently joined to a computer network, namely the Internet, will be connected so that they can communicate and interact with people and other objects. IoT is a technology transition in which devices will allow us to sense and control the physical world by making objects smarter and connecting them through an intelligent network.[1]

When objects and machines can be sensed and controlled remotely across a network, a tighter integration between the physical world and computers is enabled. This allows for improvements in the areas of efficiency, accuracy, automation, and the enablement of advanced applications.

The world of IoT is broad and multifaceted, and you may even find it somewhat complicated at first due to the plethora of components and protocols that it encompasses. Instead of viewing IoT as a single technology domain, it is good to view it as an umbrella of various concepts, protocols, and technologies, all of which are at times somewhat dependent on a particular industry. While the wide array of IoT elements is designed to create numerous benefits in the areas of productivity and automation, at the same time it introduces new challenges, such as scaling the vast numbers of devices and amounts of data that need to be processed.

This chapter seeks to further define IoT and its various elements at a high level. Having this information will prepare you to tackle more in-depth IoT subjects in the following chapters. Specifically, this chapter explores the following topics:

- **Genesis of IoT:** This section highlights IoT's place in the evolution and development of the Internet.
- **IoT and Digitization:** This section details the differences between IoT and digitization and defines a framework for better understanding their relationship.
- **IoT Impact:** This section shares a few high-level scenarios and examples to demonstrate the influence IoT will have on our world.
- **Convergence of IT and OT:** This section explores how IoT is bringing together information technology (IT) and operational technology (OT).
- **IoT Challenges:** This section provides a brief overview of the difficulties involved in transitioning to an IoT-enabled world.

Genesis of IoT

The age of IoT is often said to have started between the years 2008 and 2009. During this time period, the number of devices connected to the Internet eclipsed the world's population. With more "things" connected to the Internet than people in the world, a new age was upon us, and the Internet of Things was born.

The person credited with the creation of the term "Internet of Things" is Kevin Ashton. While working for Procter & Gamble in 1999, Kevin used this phrase to explain a new idea related to linking the company's supply chain to the Internet.

Kevin has subsequently explained that IoT now involves the addition of senses to computers. He was quoted as saying: "In the twentieth century, computers were brains without senses—they only knew what we told them." Computers depended on humans to input data and knowledge through typing, bar codes, and so on. IoT is changing this paradigm; in the twenty-first century, computers are sensing things for themselves.[2]

It is widely accepted that IoT is a major technology shift, but what is its scale and importance? Where does it fit in the evolution of the Internet?

As shown in Figure 1-1, the evolution of the Internet can be categorized into four phases. Each of these phases has had a profound impact on our society and our lives. These four phases are further defined in Table 1-1.

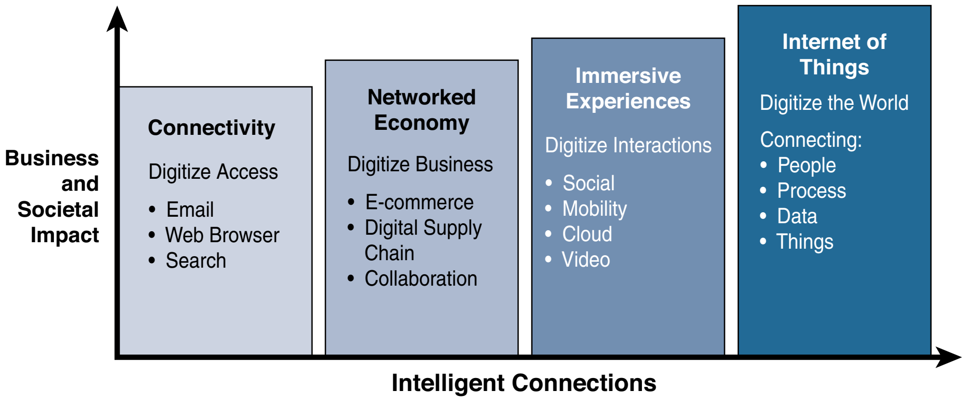


Figure 1-1 *Evolutionary Phases of the Internet*

Table 1-1 *Evolutionary Phases of the Internet*

Internet Phase	Definition
Connectivity (Digitize access)	This phase connected people to email, web services, and search so that information is easily accessed.
Networked Economy (Digitize business)	This phase enabled e-commerce and supply chain enhancements along with collaborative engagement to drive increased efficiency in business processes.
Immersive Experiences (Digitize interactions)	This phase extended the Internet experience to encompass widespread video and social media while always being connected through mobility. More and more applications are moved into the cloud.
Internet of Things (Digitize the world)	This phase is adding connectivity to objects and machines in the world around us to enable new services and experiences. It is connecting the unconnected.

Each of these evolutionary phases builds on the previous one. With each subsequent phase, more value becomes available for businesses, governments, and society in general.

The first phase, Connectivity, began in the mid-1990s. Though it may be hard to remember, or even imagine if you are younger, the world was not always connected as it is today. In the beginning, email and getting on the Internet were luxuries for universities and corporations. Getting the average person online involved dial-up modems, and even basic connectivity often seemed like a small miracle.

Even though connectivity and its speed continued to improve, a saturation point was reached where connectivity was no longer the major challenge. The focus was now on leveraging connectivity for efficiency and profit. This inflection point marked the beginning of the second phase of the Internet evolution, called the Networked Economy.

With the Networked Economy, e-commerce and digitally connected supply chains became the rage, and this caused one of the major disruptions of the past 100 years. Vendors and suppliers became closely interlinked with producers, and online shopping experienced incredible growth. The victims of this shift were traditional brick-and-mortar retailers. The economy itself became more digitally intertwined as suppliers, vendors, and consumers all became more directly connected.

The third phase, Immersive Experiences, is characterized by the emergence of social media, collaboration, and widespread mobility on a variety of devices. Connectivity is now pervasive, using multiple platforms from mobile phones to tablets to laptops and desktop computers. This pervasive connectivity in turn enables communication and collaboration as well as social media across multiple channels, via email, texting, voice, and video. In essence, person-to-person interactions have become digitized.

The latest phase is the Internet of Things. Despite all the talk and media coverage of IoT, in many ways we are just at the beginning of this phase. When you think about the fact that 99% of "things" are still unconnected, you can better understand what this evolutionary phase is all about. Machines and objects in this phase connect with other machines and objects, along with humans. Business and society have already started down this path and are experiencing huge increases in data and knowledge. In turn, this is now leading to previously unrecognized insights, along with increased automation and new process efficiencies. IoT is poised to change our world in new and exciting ways, just as the past Internet phases already have.

IoT and Digitization

IoT and *digitization* are terms that are often used interchangeably. In most contexts, this duality is fine, but there are key differences to be aware of.

At a high level, IoT focuses on connecting "things," such as objects and machines, to a computer network, such as the Internet. IoT is a well-understood term used across the industry as a whole. On the other hand, digitization can mean different things to different people but generally encompasses the connection of "things" with the data they generate and the business insights that result.

For example, in a shopping mall where Wi-Fi location tracking has been deployed, the "things" are the Wi-Fi devices. Wi-Fi location tracking is simply the capability of knowing where a consumer is in a retail environment through his or her smart phone's connection to the retailer's Wi-Fi network. While the value of connecting Wi-Fi devices or "things" to the Internet is obvious and appreciated by shoppers, tracking real-time location of Wi-Fi clients provides a specific business benefit to the mall and shop owners. In this case, it helps the business understand where shoppers tend to congregate and how much time they spend in different parts of a mall or store. Analysis of this data can lead to significant changes to the locations of product displays and advertising, where to place certain types of shops, how much rent to charge, and even where to station security guards.

Note For several years the term *Internet of Everything*, or *IoE*, was used extensively. Over time, the term IoE has been replaced by the term *digitization*. Although technical terms tend to evolve over time, the words *IoE* and *digitization* have roughly the same definition. IoT has always been a part of both, but it is important to note that IoT is a subset of both IoE and digitization.

Digitization, as defined in its simplest form, is the conversion of information into a digital format. Digitization has been happening in one form or another for several decades. For example, the whole photography industry has been digitized. Pretty much everyone has digital cameras these days, either standalone devices or built into their mobile phones. Almost no one buys film and takes it to a retailer to get it developed. The digitization of photography has completely changed our experience when it comes to capturing images.

Other examples of digitization include the video rental industry and transportation. In the past, people went to a store to rent or purchase videotapes or DVDs of movies. With digitization, just about everyone is streaming video content or purchasing movies as downloadable files.

The transportation industry is currently undergoing digitization in the area of taxi services. Businesses such as Uber and Lyft use digital technologies to allow people to get a ride using a mobile phone app. This app identifies the car, the driver, and the fare. The rider then pays the fare by using the app. This digitization is a major disruptive force to companies providing traditional taxi services.

In the context of IoT, digitization brings together things, data, and business process to make networked connections more relevant and valuable. A good example of this that many people can relate to is in the area of home automation with popular products, such as Nest. With Nest, sensors determine your desired climate settings and also tie in other smart objects, such as smoke alarms, video cameras, and various third-party devices. In the past, these devices and the functions they perform were managed and controlled separately and could not provide the holistic experience that is now possible. Nest is just one example of digitization and IoT increasing the relevancy and value of networked, intelligent connections and making a positive impact on our lives.

Companies today look at digitization as a differentiator for their businesses, and IoT is a prime enabler of digitization. Smart objects and increased connectivity drive digitization, and this is one of the main reasons that many companies, countries, and governments are embracing this growing trend.

IoT Impact

Projections on the potential impact of IoT are impressive. About 14 billion, or just 0.06%, of "things" are connected to the Internet today. Cisco Systems predicts that by 2020, this number will reach 50 billion. A UK government report speculates that this number could be even higher, in the range of 100 billion objects connected. Cisco further estimates that these new connections will lead to $19 trillion in profits and cost savings.[3] Figure 1-2 provides a graphical look at the growth in the number of devices being connected.

What these numbers mean is that IoT will fundamentally shift the way people and businesses interact with their surroundings. Managing and monitoring smart objects using real-time connectivity enables a whole new level of data-driven decision making. This in turn results in the optimization of systems and processes and delivers new services that save time for both people and businesses while improving the overall quality of life.

The following examples illustrate some of the benefits of IoT and their impact. These examples will provide you with a high-level view of practical IoT use cases to clearly illustrate how IoT will affect everyday life. For more in-depth use cases, please refer to the chapters in Part III, "IoT in Industry."

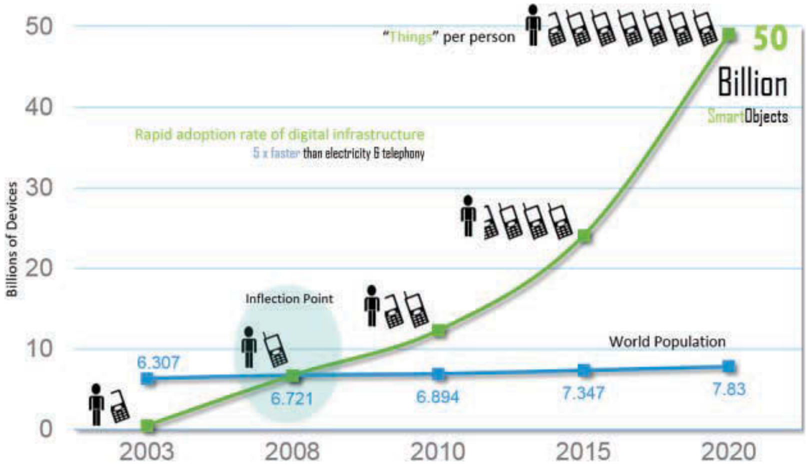


Figure 1-2 *The Rapid Growth in the Number of Devices Connected to the Internet*

Connected Roadways

People have been fantasizing about the self-driving car, or autonomous vehicle, in literature and film for decades. While this fantasy is now becoming a reality with well-known projects like Google's self-driving car, IoT is also a necessary component for implementing a fully connected transportation infrastructure.

IoT is going to allow self-driving vehicles to better interact with the transportation system around them through bidirectional data exchanges while also providing important data to the riders. Self-driving vehicles need always-on, reliable communications and data from other transportation-related sensors to reach their full potential. *Connected roadways* is the term associated with both the driver and driverless cars fully integrating with the surrounding transportation infrastructure. Figure 1-3 shows a self-driving car designed by Google.

Figure 1-3 *Google's Self-Driving Car*

Basic sensors reside in cars already. They monitor oil pressure, tire pressure, temperature, and other operating conditions, and provide data around the core car functions. From behind the steering wheel, the driver can access this data while also controlling the car using equipment such as a steering wheel, pedals, and so on. The need for all this sensory information and control is obvious. The driver must be able to understand, handle, and make critical decisions while concentrating on driving safely. The Internet of Things is replicating this concept on a much larger scale.

Today, we are seeing automobiles produced with thousands of sensors, to measure everything from fuel consumption to location to the entertainment your family is watching during the ride. As automobile manufacturers strive to reinvent the driving experience, these sensors are becoming IP-enabled to allow easy communication with other systems both inside and outside the car. In addition, new sensors and communication technologies are being developed to allow vehicles to "talk" to other vehicles, traffic signals, school zones, and other elements of the transportation infrastructure. We are now starting to realize a truly connected transportation solution.

Most connected roadways solutions focus on resolving today's transportation challenges. These challenges can be classified into the three categories highlighted in Table 1-2.

Table 1-2 *Current Challenges Being Addressed by Connected Roadways*

Challenge	Supporting Data
Safety	According to the US Department of Transportation, 5.6 million crashes were reported in 2012 alone, resulting in more than 33,000 fatalities. IoT and the enablement of connected vehicle technologies will empower drivers with the tools they need to anticipate potential crashes and significantly reduce the number of lives lost each year.

Challenge	Supporting Data
Mobility	More than a billion cars are on the roads worldwide. Connected vehicle mobility applications can enable system operators and drivers to make more informed decisions, which can, in turn, reduce travel delays. Congestion causes 5.5 billion hours of travel delay per year, and reducing travel delays is more critical than ever before. In addition, communication between mass transit, emergency response vehicles, and traffic management infrastructures help optimize the routing of vehicles, further reducing potential delays.
Environment	According to the American Public Transportation Association, each year transit systems can collectively reduce carbon dioxide (CO_2) emissions by 16.2 million metric tons by reducing private vehicle miles. Connected vehicle environmental applications will give all travelers the real-time information they need to make "green" transportation choices.

Sources: Traffic Safety Facts, 2010; National Highway Traffic Safety Administration, June 2012; and WHO Global Status Report on Road Safety, 2013.

By addressing the challenges in Table 1-2, connected roadways will bring many benefits to society. These benefits include reduced traffic jams and urban congestion, decreased casualties and fatalities, increased response time for emergency vehicles, and reduced vehicle emissions.

For example, with IoT-connected roadways, a concept known as Intersection Movement Assist (IMA) is possible. This application warns a driver (or triggers the appropriate response in a self-driving car) when it is not safe to enter an intersection due to a high probability of a collision—perhaps because another car has run a stop sign or strayed into the wrong lane. Thanks to the communications system between the vehicles and the infrastructure, this sort of scenario can be handled quickly and safely. See Figure 1-4 for a graphical representation of IMA.

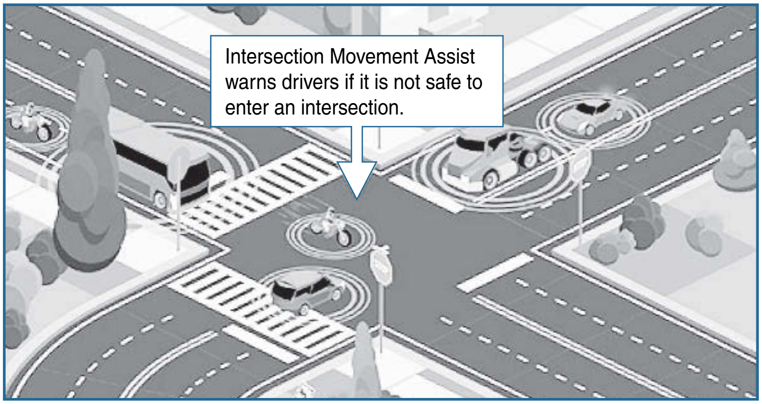


Figure 1-4 *Application of Intersection Movement Assist*

IMA is one of many possible roadway solutions that emerge when we start to integrate IoT with both traditional and self-driving vehicles. Other solutions include automated vehicle tracking, cargo management, and road weather communications.

With automated vehicle tracking, a vehicle's location is used for notification of arrival times, theft prevention, or highway assistance. Cargo management provides precise positioning of cargo as it is en route so that notification alerts can be sent to a dispatcher and routes can be optimized for congestion and weather. Road weather communications use sensors and data from satellites, roads, and bridges to warn vehicles of dangerous conditions or inclement weather on the current route.

Today's typical road car utilizes more than a million lines of code—and this only scratches the surface of the data potential. As cars continue to become more connected and capable of generating continuous data streams related to location, performance, driver behavior, and much more, the data generation potential of a single car is staggering. It is estimated that a fully connected car will generate more than 25 gigabytes of data per hour, much of which will be sent to the cloud. To put this in perspective, that's equivalent to a dozen HD movies sent to the cloud every hour—by your car! Multiply that by the number of hours a car is driven per year and again by the number of cars on the road, and you see that the amount of connected car data generated, transmitted, and stored in the cloud will be in the zettabytes range per year (more than a billion petabytes per year). Figure 1-5 provides an overview of the sort of sensors and connectivity that you will find in a connected car.

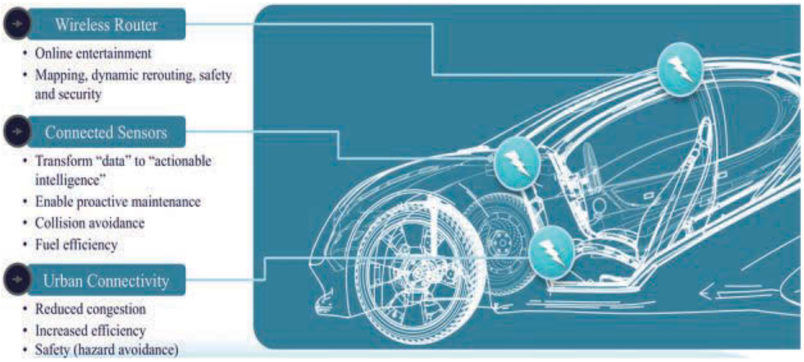


Figure 1-5 *The Connected Car*

Another area where connected roadways are undergoing massive disruption is in how the data generated by a car will be used by third parties. Clearly, the data generated by your car needs to be handled in a secure and reliable way, which means the network needs to be secure, it must provide authentication and verification of the driver and car, and it needs to be highly available. But who will use all this data? Automobile data is extremely useful to a wide range of interested parties. For example, tire companies can collect data related to use and durability of their products in a range of environments in real time. Automobile manufacturers can collect information from sensors to better understand how the cars are being driven, when parts are starting to fail, or whether the car has broken down—details that will help them build better cars in the future. This becomes especially true as autonomous vehicles are introduced, which are sure to be driven in a completely different way than the traditional family car.

In the future, car sensors will be able to interact with third-party applications, such as GPS/maps, to enable dynamic rerouting to avoid traffic, accidents, and other hazards. Similarly, Internet-based entertainment, including music, movies, and other streamings or downloads, can be personalized and customized to optimize a road trip.

This data will also be used for targeted advertising. As GPS navigation systems become more integrated with sensors and wayfinding applications, it will become possible for personalized routing suggestions to be made. For example, if it is known that you prefer a certain coffee shop, through the use of a cloud-based data connector, the navigation system will be able to provide routing suggestions that have you drive your car past the right coffee shop.

All these data opportunities bring into play a new technology: the IoT data broker. Imagine the many different types of data generated by an automobile and the plethora of different parties interested in this data. This poses a significant business opportunity. In a very real sense, the data generated by the car and driver becomes a valuable commodity that can be bought and sold. While the data transmitted from the car will likely go to one initial location in the cloud, from there the data can be separated and sold selectively by the data broker. For example, tire companies will pay for information from sensors related to your tires, but they won't get anything else. While information brokers have been around a long time, the technology used to aggregate and separate the data from connected cars in a secure and governed manner is rapidly developing and will continue to be a major focus of the IoT industry for years to come.

Connected roadways are likely to be one of the biggest growth areas for innovation. Automobiles and the roads they use have seen incredible change over the past century, but the changes ahead of us are going to be just as astonishing. In the past few years alone, we have seen highway systems around the world adopt sophisticated sensors systems that can detect seismic vibrations, car accidents, severe weather conditions, traffic congestion, and more. Recent advancements in roadway fiber-optic sensing technology is now able to record not only how many cars are passing but their speed and type. Due to the many reasons already discussed, connected cars and roadways are early adopters of IoT technology. For a more in-depth discussion of IoT use cases and architectures in the transportation industry, see Chapter 13, "Transportation."

Connected Factory

For years, traditional factories have been operating at a disadvantage, impeded by production environments that are "disconnected" or, at the very least, "strictly gated" to corporate business systems, supply chains, and customers and partners. Managers of these traditional factories are essentially "flying blind" and lack visibility into their operations. These operations are composed of plant floors, front offices, and suppliers operating in independent silos. Consequently, rectifying downtime issues, quality problems, and the root causes of various manufacturing inefficiencies is often difficult.

The main challenges facing manufacturing in a factory environment today include the following:

- Accelerating new product and service introductions to meet customer and market opportunities
- Increasing plant production, quality, and uptime while decreasing cost
- Mitigating unplanned downtime (which wastes, on average, at least 5% of production)
- Securing factories from cyber threats
- Decreasing high cabling and re-cabling costs (up to 60% of deployment costs)
- Improving worker productivity and safety[4]

Adding another level of complication to these challenges is the fact that they often need to be addressed at various levels of the manufacturing business. For example, executive management is looking for new ways to manufacture in a more cost-effective manner while balancing the rising energy and material costs. Product development has time to market as the top priority. Plant managers are entirely focused on gains in plant efficiency and operational agility. The controls and automation department looks after the plant networks, controls, and applications and therefore requires complete visibility into all these systems.

Industrial enterprises around the world are retooling their factories with advanced technologies and architectures to resolve these problems and boost manufacturing flexibility and speed. These improvements help them achieve new levels of overall equipment effectiveness, supply chain responsiveness, and customer satisfaction. A convergence of factory-based operational technologies and architectures with global IT networks is starting to occur, and this is referred to as the *connected factory*.

As with the IoT solutions for the connected roadways previously discussed, there are already large numbers of basic sensors on factory floors. However, with IoT, these sensors not only become more advanced but also attain a new level of connectivity. They are smarter and gain the ability to communicate, mainly using the Internet Protocol (IP) over an Ethernet infrastructure.

In addition to sensors, the devices on the plant floor are becoming smarter in their ability to transmit and receive large quantities of real-time informational and diagnostic data. Ethernet connectivity is becoming pervasive and spreading beyond just the main controllers in a factory to devices such as the robots on the plant floor. In addition, more IP-enabled devices, including video cameras, diagnostic smart objects, and even personal mobile devices, are being added to the manufacturing environment.

For example, a smelting facility extracts metals from their ores. The facility uses both heat and chemicals to decompose the ore, leaving behind the base metal. This is a multistage process, and the data and controls are all accessed via various control rooms in a facility. Operators must go to a control room that is often hundreds of meters away for data and production changes. Hours of operator time are often lost to the multiple trips to the control room needed during a shift. With IoT and a connected factory solution, true "machine-to-people" connections are implemented to bring sensor data directly to operators on the floor via mobile devices. Time is no longer wasted moving back and forth between the control rooms and the plant floor. In addition, because the operators now receive data in real time, decisions can be made immediately to improve production and fix any quality problems.

Another example of a connected factory solution involves a real-time location system (RTLS). An RTLS utilizes small and easily deployed Wi-Fi RFID tags that attach to virtually any material and provide real-time location and status. These tags enable a facility to track production as it happens. These IoT sensors allow components and materials on an assembly line to "talk" to the network. If each assembly line's output is tracked in real time, decisions can be made to speed up or slow production to meet targets, and it is easy to determine how quickly employees are completing the various stages of production. Bottlenecks at any point in production and quality problems are also quickly identified.

While we tend to look at IoT as an evolution of the Internet, it is also sparking an evolution of industry. In 2016 the World Economic Forum referred to the evolution of the Internet and the impact of IoT as the "fourth Industrial Revolution."[5] The first Industrial Revolution occurred in Europe in the late eighteenth century, with the application of steam and water to mechanical production. The second Industrial Revolution, which took place between the early 1870s and the early twentieth century, saw the introduction of the electrical grid and mass production. The third revolution came in the late 1960s/early 1970s, as computers and electronics began to make their mark on manufacturing and other industrial systems. The fourth Industrial Revolution is happening now, and the Internet of Things is driving it. Figure 1-6 summarizes these four Industrial Revolutions as Industry 1.0 through Industry 4.0.

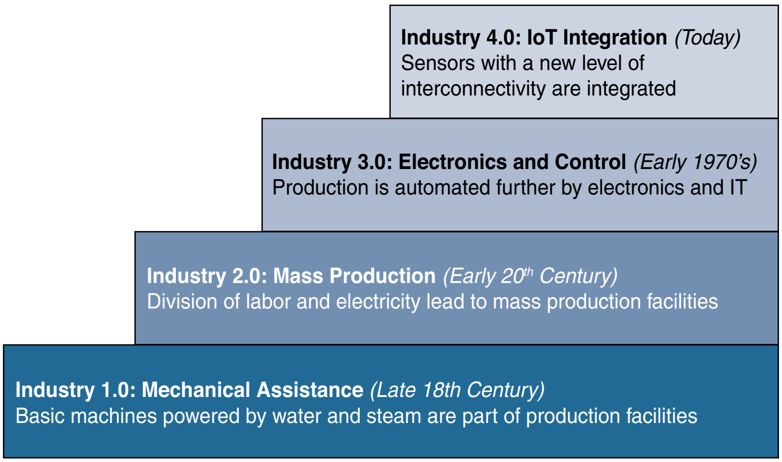


Figure 1-6 *The Four Industrial Revolutions*

The IoT wave of Industry 4.0 takes manufacturing from a purely automated assembly line model of production to a model where the machines are intelligent and communicate with one another. IoT in manufacturing brings with it the opportunity for inserting intelligence into factories. This starts with creating smart objects, which involves embedding sensors, actuators, and controllers into just about everything related to production. Connections tie it all together so that people and machines work together to analyze the data and make intelligent decisions. Eventually this leads to machines predicting failures and self-healing and points to a world where human monitoring and intervention are no longer necessary.

Smart Connected Buildings

Another place IoT is making a disruptive impact is in the smart connected buildings space. In the past several decades, buildings have become increasingly complex, with systems overlaid one upon another, resulting in complex intersections of structural, mechanical, electrical, and IT components. Over time, these operational networks that support the building environment have matured into sophisticated systems; however, for the most part, they are deployed and managed as separate systems that have little to no interaction with each other.

The function of a building is to provide a work environment that keeps the workers comfortable, efficient, and safe. Work areas need to be well lit and kept at a comfortable temperature. To keep workers safe, the fire alarm and suppression system needs to be carefully managed, as do the door and physical security alarm systems. While intelligent systems for modern buildings are being deployed and improved for each of these functions, most of these systems currently run independently of each other—and they rarely take into account where the occupants of the building actually are and how many of them are present in different parts of the building. However, many buildings are beginning to deploy sensors throughout the building to detect occupancy. These tend to be motion sensors or sensors tied to video cameras. Motion detection occupancy sensors work great if everyone is moving around in a crowded room and can automatically shut the lights off when everyone has left, but what if a person in the room is out of sight of the sensor? It is a frustrating matter to be at the mercy of an unintelligent sensor on the wall that wants to turn off the lights on you.

Similarly, sensors are often used to control the heating, ventilation, and air-conditioning (HVAC) system. Temperature sensors are spread throughout the building and are used to influence the building management system's (BMS's) control of air flow into a room.

Another interesting aspect of the smart building is that it makes them easier and cheaper to manage. Considering the massive costs involved in operating such complex structures, not to mention how many people spend their working lives inside a building, managers have become increasingly interested in ways to make buildings more efficient and cheaper to manage. Have you ever heard people complain that they had too little working space in their office, or that the office space wasn't being used efficiently? When people go to their managers and ask for a change to the floor plan, such as asking for an increase in the amount of space they work in, they are often asked to prove their case. But workplace floor efficiency and usage evidence tends to be anecdotal at best. When smart building sensors and occupancy detection are combined with the power of data analytics (discussed in Chapter 7, “Data and Analytics for IoT”), it becomes easy to demonstrate floor plan usage and prove your case. Alternatively, the building manager can use a similar approach to see where the floor is not being used efficiently and use this information to optimize the available space. This has brought about the age of building automation, empowered by IoT.

While many technical solutions exist for looking after building systems, until recently they have all required separate overlay networks, each responsible for its assigned task. In an attempt to connect these systems into a single framework, the building automation system (BAS) has been developed to provide a single management system for the HVAC, lighting, fire alarm, and detection systems, as well as access control. All these systems may support different types of sensors and connections to the BAS. How do you connect them together so the building can be managed in a coherent way? This highlights one of the biggest challenges in IoT, which is discussed throughout this book: the heterogeneity of IoT systems.

Before you can bring together heterogeneous systems, they need to converge at the network layer and support a common services layer that allows application integration. The value of converged networks is well documented. For example, in the early 2000s, Cisco and several other companies championed the convergence of voice and video onto single IP networks that were shared with other IT applications. The economies of scale and operational efficiencies gained were so massive that VoIP and collaboration technologies are now the norm. However, the convergence to IP and a common services framework for buildings has been slower.

For example, the de facto communication protocol responsible for building automation is known as BACnet (Building Automation and Control Network). In a nutshell, the BACnet protocol defines a set of services that allow Ethernet-based communication between building devices such as HVAC, lighting, access control, and fire detection systems. The same building Ethernet switches used for IT may also be used for BACnet. This standardization also makes possible an intersection point to the IP network (which is run by the IT department) through the use of a gateway device. In addition, BACnet/IP has been defined to allow the “things” in the building network to communicate over IP, thus allowing closer consolidation of the building management system on a single network. Figure 1-7 illustrates the conversion of building protocols to IP over time.

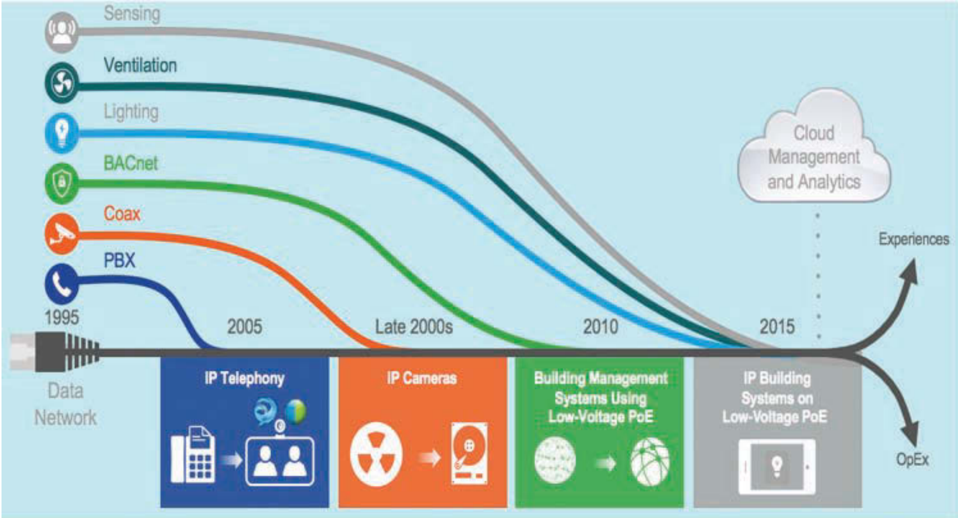


Figure 1-7 *Convergence of Building Technologies to IP*

Another promising IoT technology in the smart connected building, and one that is seeing widespread adoption, is the "digital ceiling." The digital ceiling is more than just a lighting control system. This technology encompasses several of the building's different networks—including lighting, HVAC, blinds, CCTV (closed-circuit television), and security systems—and combines them into a single IP network. Figure 1-8 provides a framework for the digital ceiling.

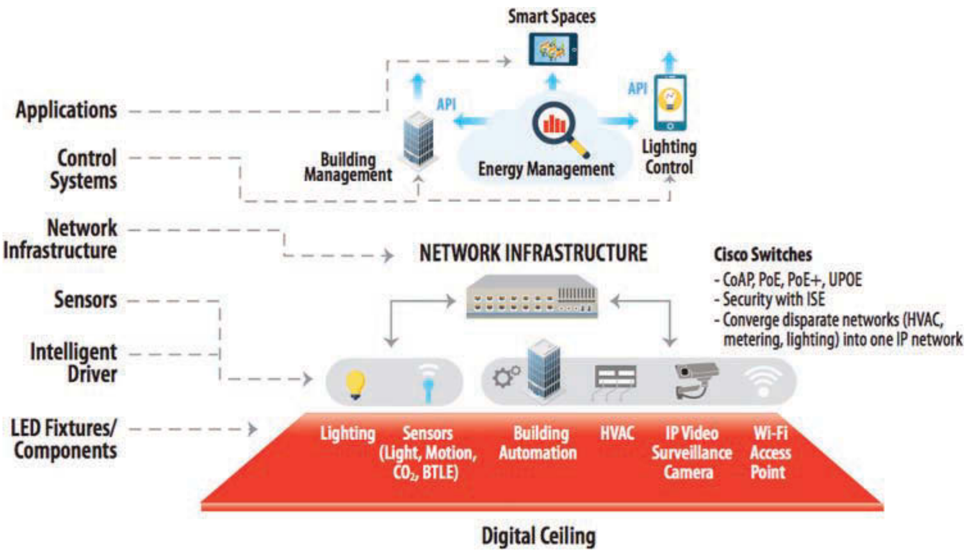


Figure 1-8 *A Framework for the Digital Ceiling*

Central to digital ceiling technology is the lighting system. As you are probably aware, the lighting market is currently going through a major shift toward light-emitting diodes (LEDs). Compared to traditional lighting, LEDs offer lower energy consumption and far longer life. The lower power requirements of LED fixtures allow them to run on Power over Ethernet (PoE), permitting them to be connected to standard network switches.

In a digital ceiling environment, every luminaire or lighting fixture is directly network-attached, providing control and power over the same infrastructure. This transition to LED lighting means that a single converged network is now able to encompasses luminaires that are part of consolidated building management as well as elements managed by the IT network, supporting voice, video, and other data applications.

The next time you look at the ceiling in your office building, count the number of lights. The quantity of lights easily outnumbers the number of physical wired ports—by a hefty margin. Obviously, supporting the larger number of Ethernet ports and density of IP addresses requires some redesign of the network, and it also requires a quiet, fanless PoE-capable switch in the ceiling. That being said, the long-term business case supporting reduced energy costs from LED luminaries versus traditional fluorescent or halogen lights is so significant that the added initial investment in the network is almost inconsequential. The business case for the digital ceiling becomes even stronger when a building is being renovated or a new structure is being built. In these cases, the cost benefit of running CAT 6/5e cables in the ceiling versus plenum-rated electrical wiring to every light is substantial.

The energy savings value of PoE-enabled LED lighting in the ceiling is clear. However, having an IP-enabled sensor device in the ceiling at every point people may be present opens up an entirely new set of possibilities. For example, most modern LED ceiling fixtures support occupancy sensors. These sensors provide high-resolution occupancy data collection, which can be used to turn the lights on and off, and this same data can be combined with advanced analytics to control other systems, such as HVAC and security. Unlike traditional sensors that use rudimentary motion detection, modern lighting sensors integrate a variety of occupancy-sensing technologies, including Bluetooth low energy (BLE) and Wi-Fi. The science here is simple: Because almost every person these days carries a smart device that supports BLE and Wi-Fi, all the sensor has to do is detect BLE or Wi-Fi beacons from a nearby device. When someone walks near a light, the person's location is detected, and the wireless system can send information to control the air flow from the HVAC system into that zone in real time, maximizing the comfort of the office worker. Figure 1-9 shows an example of an occupancy sensor in a digital ceiling light.

Figure 1-9 *An LED Digital Ceiling Light with Occupancy Sensor*
(Photo by Bill MacGowan)

You can begin to imagine the possibilities that IoT smart lighting brings to a workplace setting. Not only does it provide for optimized levels of lighting based on actual occupancy and building usage, it allows granular control of temperature, management of smoke and fire detection, video cameras, and building access control. IoT allows all this to run through a single network, requiring less installation time and a lower total cost of system ownership.

Smart Creatures

When you think about IoT, you probably picture only inanimate objects and machines being connected. However, IoT also provides the ability to connect living things to the Internet. Sensors can be placed on animals and even insects just as easily as on machines, and the benefits can be just as impressive.

One of the most well-known applications of IoT with respect to animals focuses on what is often referred to as the "connected cow." Sparked, a Dutch company, developed a sensor that is placed in a cow's ear. The sensor monitors various health aspects of the cow as well as its location and transmits the data wirelessly for analysis by the farmer.

The data from each of these sensors is approximately 200 MB per year, and you obviously need a network infrastructure to make the connection with the sensors and store the information. Once the data is being collected, however, you get a complete view of the herd, with statistics on every cow. You can learn how environmental factors may be affecting the herd as a whole and about changes in diet. This enables early detection of disease as cows tend to eat less days before they show symptoms. These sensors even allow the detection of pregnancy in cows.

Another application of IoT to organisms involves the placement of sensors on roaches. While the topic of roaches is a little unsettling to many folks, the potential benefits of IoT-enabled roaches could make a life-saving difference in disaster situations.

Researchers at North Carolina State University are working with Madagascar hissing cockroaches in the hopes of helping emergency personnel rescue survivors after a disaster. As shown in Figure 1-10, an electronic backpack attaches to a roach. This backpack communicates with the roach through parts of its body. Low-level electrical pulses to an antenna on one side makes the roach turn to the opposite side because it believes it is encountering an obstacle. The cerci of the roach are sensory organs on the abdomen that detect danger through changing air currents. When the backpack stimulates the cerci, the roach moves forward because it thinks a predator is approaching.

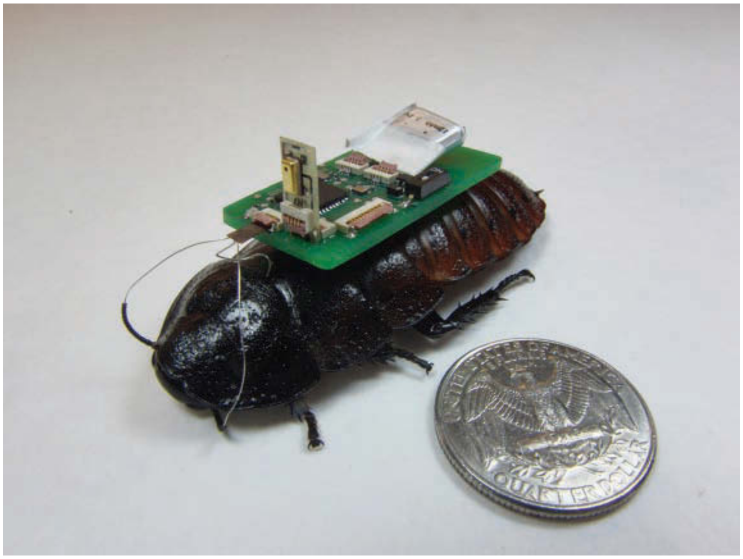


Figure 1-10 *IoT-Enabled Roach Can Assist in Finding Survivors After a Disaster (Photo courtesy of Alper Bozkurt, NC State University)*

The electronic backpack uses wireless communication to a controller and can be "driven" remotely. Imagine a fleet of these roaches being used in a disaster scenario, such as searching for survivors in a collapsed building after an earthquake. The roaches are naturally designed to efficiently move around objects in confined spaces. Technology has also been tested to keep the roaches in the disaster area; it is similar to the invisible fencing that is often used to keep dogs in a yard. The use of roaches in this manner allows for the mapping of spaces that rescue personnel cannot access, which helps search for survivors.

To help with finding a person trapped in the rubble of a collapsed building, the electronic backpack is equipped with directional microphones that allow for the detection of certain sounds and the direction from which they are coming. Software can analyze the sounds to ensure that they are from a person rather than from, say, a leaking pipe. Roaches can then be steered toward the sounds that may indicate people who are trapped. In addition, the microphones provide the ability for rescue personnel to listen in on whatever sounds are detected.

These examples show that IoT often goes beyond just adding sensors and more intelligence to nonliving "things." Living "things" can also be connected to the Internet and this connection can provide important results.

Convergence of IT and OT

Until recently, information technology (IT) and operational technology (OT) have for the most part lived in separate worlds. IT supports connections to the Internet along with related data and technology systems and is focused on the secure flow of data across an organization. OT monitors and controls devices and processes on physical operational systems. These systems include assembly lines, utility distribution networks, production facilities, roadway systems, and many more. Typically, IT did not get involved with the production and logistics of OT environments.

Specifically, the IT organization is responsible for the information systems of a business, such as email, file and print services, databases, and so on. In comparison, OT is responsible for the devices and processes acting on industrial equipment, such as factory machines, meters, actuators, electrical distribution automation devices, SCADA (supervisory control and data acquisition) systems, and so on. Traditionally, OT has used dedicated networks with specialized communications protocols to connect these devices, and these networks have run completely separately from the IT networks.

Management of OT is tied to the lifeblood of a company. For example, if the network connecting the machines in a factory fails, the machines cannot function, and production may come to a standstill, negatively impacting business on the order of millions of dollars. On the other hand, if the email server (run by the IT department) fails for a few hours, it may irritate people, but it is unlikely to impact business at anywhere near the same level. Table 1-3 highlights some of the differences between IT and OT networks and their various challenges.

Table 1-3 *Comparing Operational Technology (OT) and Information Technology (IT)*

Criterion	Industrial OT Network	Enterprise IT Network
Operational focus	Keep the business operating 24x7	Manage the computers, data, and employee communication system in a secure way
Priorities	1. Availability 2. Integrity 3. Security	1. Security 2. Integrity 3. Availability
Types of data	Monitoring, control, and supervisory data	Voice, video, transactional, and bulk data
Security	Controlled physical access to devices	Devices and users authenticated to the network
Implication of failure	OT network disruption directly impacts business	Can be business impacting, depending on industry, but workarounds may be possible

Criterion	Industrial OT Network	Enterprise IT Network
Network upgrades (software or hardware)	Only during operational maintenance windows	Often requires an outage window when workers are not onsite; impact can be mitigated
Security vulnerability	Low: OT networks are isolated and often use proprietary protocols	High: continual patching of hosts is required, and the network is connected to Internet and requires vigilant protection

Source: Maciej Kranz, *IT Is from Venus, OT Is from Mars*, blogs.cisco.com/digital/it-is-from-venus-ot-is-from-mars, July 14, 2015.

With the rise of IoT and standards-based protocols, such as IPv6, the IT and OT worlds are converging or, more accurately, OT is beginning to adopt the network protocols, technology, transport, and methods of the IT organization, and the IT organization is beginning to support the operational requirements used by OT. When IT and OT begin using the same networks, protocols, and processes, there are clear economies of scale. Not only does convergence reduce the amount of capital infrastructure needed but networks become easier to operate, and the flexibility of open standards allows faster growth and adaptability to new technologies.

However, as you can see from Table 1-3, the convergence of IT and OT to a single consolidated network poses several challenges. There are fundamental cultural and priority differences between these two organizations. IoT is forcing these groups to work together, when in the past they have operated rather autonomously. For example, the OT organization is baffled when IT schedules a weekend shutdown to update software without regard to production requirements. On the other hand, the IT group does not understand the prevalence of proprietary or specialized systems and solutions deployed by OT.

Take the case of deploying quality of service (QoS) in a network. When the IT team deploys QoS, voice and video traffic are almost universally treated with the highest level of service. However, when the OT system shares the same network, a very strong argument can be made that the real-time OT traffic should be given a higher priority than even voice because any disruption in the OT network could impact the business.

With the merging of OT and IT, improvements are being made to both systems. OT is looking more toward IT technologies with open standards, such as Ethernet and IP. At the same time, IT is becoming more of a business partner with OT by better understanding business outcomes and operational requirements.

The overall benefit of IT and OT working together is a more efficient and profitable business due to reduced downtime, lower costs through economy of scale, reduced inventory, and improved delivery times. When IT/OT convergence is managed correctly, IoT becomes fully supported by both groups. This provides a “best of both worlds” scenario, where solid industrial control systems reside on an open, integrated, and secure technology foundation.[6]

IoT Challenges

While an IoT-enabled future paints an impressive picture, it does not come without significant challenges. Many parts of IoT have become reality, but certain obstacles need to be overcome for IoT to become ubiquitous throughout industry and our everyday life. Table 1-4 highlights a few of the most significant challenges and problems that IoT is currently facing.

Table 1-4 *IoT Challenges*

Challenge	Description
Scale	While the scale of IT networks can be large, the scale of OT can be several orders of magnitude larger. For example, one large electrical utility in Asia recently began deploying IPv6-based smart meters on its electrical grid. While this utility company has tens of thousands of employees (which can be considered IP nodes in the network), the number of meters in the service area is tens of millions. This means the scale of the network the utility is managing has increased by more than 1,000-fold! Chapter 5, "IP as the IoT Network Layer," explores how new design approaches are being developed to scale IPv6 networks into the millions of devices.
Security	With more "things" becoming connected with other "things" and people, security is an increasingly complex issue for IoT. Your threat surface is now greatly expanded, and if a device gets hacked, its connectivity is a major concern. A compromised device can serve as a launching point to attack other devices and systems. IoT security is also pervasive across just about every facet of IoT. For more information on IoT security, see Chapter 8, "Securing IoT."
Privacy	As sensors become more prolific in our everyday lives, much of the data they gather will be specific to individuals and their activities. This data can range from health information to shopping patterns and transactions at a retail establishment. For businesses, this data has monetary value. Organizations are now discussing who owns this data and how individuals can control whether it is shared and with whom.
Big data and data analytics	IoT and its large number of sensors is going to trigger a deluge of data that must be handled. This data will provide critical information and insights if it can be processed in an efficient manner. The challenge, however, is evaluating massive amounts of data arriving from different sources in various forms and doing so in a timely manner. See Chapter 7 for more information on IoT and the challenges it faces from a big data perspective.

Challenge	Description
Interoperability	As with any other nascent technology, various protocols and architectures are jockeying for market share and standardization within IoT. Some of these protocols and architectures are based on proprietary elements, and others are open. Recent IoT standards are helping minimize this problem, but there are often various protocols and implementations available for IoT networks. The prominent protocols and architectures—especially open, standards-based implementations—are the subject of this book. For more information on IoT architectures, see Chapter 2, "IoT Network Architecture and Design." Chapter 4, "Connecting Smart Objects," Chapter 5, "IP as the IoT Network Layer," and Chapter 6, "Application Protocols for IoT," take a more in-depth look at the protocols that make up IoT.

Summary

This chapter provides an introductory look at the Internet of Things and answers the question "What is IoT?" IoT is about connecting the unconnected, enabling smart objects to communicate with other objects, systems, and people. The end result is an intelligent network that allows more control of the physical world and the enablement of advanced applications.

This chapter also provides a historical look at IoT, along with a current view of IoT as the next evolutionary phase of the Internet. This chapter details a few high-level use cases to show the impact of IoT and some of the ways it will be changing our world.

A number of IoT concepts and terms are defined throughout this chapter. The differences between IoT and digitization are discussed, as well as the convergence between IT and OT. The last section details the challenges faced by IoT.

This chapter should leave you with a clearer understanding of what IoT is all about. In addition, this chapter serves as the foundational block from which you can dive further into IoT in the following chapters.

References

1. Lindsay Hiebert, Public Safety Blog Series-Connecting the Unconnected in Public Safety Response, https://blogs.cisco.com/government/connecting-the-unconnected-in-public-safety-response, October 25, 2013.

2. Arik Gabbai, "Kevin Ashton describes the Internet of Things," *Smithsonian Magazine*, January 2015, www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/.

3. UK Government Chief Scientific Adviser, *The Internet of Things: Making the Most of the Second Digital Revolution*, Accessed December 2016, www.gov.uk/government/uploads/system/uploads/attachment_data/file/389315/14-1230-internet-of-things-review.pdf.

4. Cisco, *The Cisco Connected Factory: Powering a Renaissance in Manufacturing* (white paper), Accessed December 2016, www.cisco.com/c/dam/m/es_la/| internet-of-everything-ioe/industrial/assets/pdfs/cisco-connected-factory.pdf.

5. Klaus Schwab, *The Fourth Industrial Revolution: What It Means, How to Respond*, https://www.weforum.org/agenda/2016/01/ the-fourth-industrial-revolution-what-it-means-and-how-to-respond/

6. Adapted from Rockwell/Cisco presentation, https://salesconnect.cisco.com/#/ content-detail/a5c09760-7260-4870-9019-0fb6a4a98af0 (Login required).

This page intentionally left blank

IoT Network Architecture and Design

Imagine that one day you decide to build a house. You drive over to the local construction supply store and try to figure out what materials you will need. You buy the lumber, nails and screws, cement mix for the foundation, roofing materials, and so on. A truck comes by and drops off all the materials at the site of your future home. You stare at the piles of materials sitting on what you hope will one day become your front lawn and realize you have no idea where to start. Something important is missing: You don't have architectural plans for the new house! Unfortunately, your plans to build a beautiful new home will have to wait until you get the help of an architect.

As most home builders know, even the simplest construction projects require careful planning and an architecture that adheres to certain standards. When projects become more complex, detailed architectural plans are not only a good idea, they are, in most places, required by law.

To successfully complete a construction project, time and effort are required to design each phase, from the foundation to the roof. Your plans must include detailed designs for the electrical, plumbing, heating, and security systems. Strong architectural blueprints (and the required engineering to support them) are necessary in all construction projects, from the simple to the very complex. In the same vein, a computer network should never be built without careful planning, thorough security policies, and adherence to well-understood design practices. Failure to carefully architect a network according to sound design principles will likely result in something that is difficult to scale, manage, adapt to organizational changes, and, worst of all, troubleshoot when things go wrong.

Most CIOs and CTOs understand that the network runs the business. If the network fails, company operations can be seriously impaired. Just as a house must be designed with the strength to withstand potential natural disasters, such as seismic events and hurricanes, information technology (IT) systems need to be designed to withstand "network earthquakes," such as distributed denial of service (DDoS) attacks, future growth requirements, network outages, and even human error. To address these challenges, the art of network architecture has gained tremendous influence in IT organizations over the past two decades. In fact, for many companies, the responsibility of overseeing network architecture is often seen as one of the most senior positions in the IT and operational technology (OT) organizations. For example, the title chief enterprise architect (CEA) has gained so much traction in recent years that the position is often equated to the responsibilities of a CTO, and in many instances, the CEA reports directly to the CEO.

Enterprise IT network architecture has matured significantly over the past two decades and is generally well understood; however, the discipline of IoT network architecture is new and requires a fresh perspective. It is important to note that while some similarities between IT and IoT architectures do exist, for the most part, the challenges and requirements of IoT systems are radically different from those of traditional IT networks. The terminology is also different to the point where IoT networks are often under the umbrella of OT, which is responsible for the management and state of operational systems. In contrast, IT networks are primarily concerned with the infrastructure that transports flows of data, regardless of the data type.

This chapter examines some of the unique challenges posed by IoT networks and how these challenges have driven new architectural models. This chapter explores the following areas:

- **Drivers Behind New Network Architectures:** OT networks drive core industrial business operations. They have unique characteristics and constraints that are not easily supported by traditional IT network architectures.
- **Comparing IoT Architectures:** Several architectures have been published for IoT, including those by ETSI and the IoT World Forum. This section discusses and compares these architectures.
- **A Simplified IoT Architecture:** While several IoT architectures exist, a simplified model is presented in this section to lay a foundation for rest of the material discussed in this book.
- **The Core IoT Functional Stack:** The IoT network must be designed to support its unique requirements and constraints. This section provides an overview of the full networking stack, from sensors all the way to the applications layer.
- **IoT Data Management and Compute Stack:** This section introduces data management, including storage and compute resource models for IoT, and involves edge, fog, and cloud computing.

Drivers Behind New Network Architectures

This chapter begins by comparing how using an architectural blueprint to construct a house is similar to the approach we take when designing a network. Now, imagine an experienced architect who has built residential houses for his whole career. He is an expert in this field and knows exactly what it takes to not only make a house architecturally attractive but also to be functional and livable and meet the construction codes mandated by local government. One day, this architect is asked to take on a new project: Construct a massive stadium that will be a showpiece for the city and which will support a variety of sporting teams, concerts, and community events, and which has a seating capacity of 60,000+.

While the architect has extensive experience in designing homes, those skills will clearly not be enough to meet the demands of this new project. The scale of the stadium is several magnitudes larger, the use is completely different, and the wear and tear will be at a completely different level. The architect needs a new architectural approach that meets the requirements for building the stadium.

The difference between IT and IoT networks is much like the difference between residential architecture and stadium architecture. While traditional network architectures for IT have served us well for many years, they are not well suited to the complex requirements of IoT. Chapter 1, "What Is IoT?" introduces some of the differences between IT and OT, as well as some of the inherent challenges posed by IoT. These differences and challenges are driving fundamentally new architectures for IoT systems.

The key difference between IT and IoT is the data. While IT systems are mostly concerned with reliable and continuous support of business applications such as email, web, databases, CRM systems, and so on, IoT is all about the data generated by sensors and how that data is used. The essence of IoT architectures thus involves how the data is transported, collected, analyzed, and ultimately acted upon.

Table 2-1 takes a closer look at some of the differences between IT and IoT networks, with a focus on the IoT requirements that are driving new network architectures, and considers what adjustments are needed.

Table 2-1 *IoT Architectural Drivers*

Challenge	Description	IoT Architectural Change Required
Scale	The massive scale of IoT endpoints (sensors) is far beyond that of typical IT networks.	The IPv4 address space has reached exhaustion and is unable to meet IoT's scalability requirements. Scale can be met only by using IPv6. IT networks continue to use IPv4 through features like Network Address Translation (NAT).
Security	IoT devices, especially those on wireless sensor networks (WSNs), are often physically exposed to the world.	Security is required at every level of the IoT network. Every IoT endpoint node on the network must be part of the overall security strategy and must support device-level authentication and link encryption. It must also be easy to deploy with some type of zero-touch deployment model.

Challenge	Description	IoT Architectural Change Required
Devices and networks constrained by power, CPU, memory, and link speed	Due to the massive scale and longer distances, the networks are often constrained, lossy, and capable of supporting only minimal data rates (tens of bps to hundreds of Kbps).	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The network is also constrained, meaning modifications need to be made to traditional network-layer transport mechanisms.
The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing network bottlenecks and slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT network, from the edge to the cloud. In traditional IT networks, analytics and applications typically run only in the cloud.
Support for legacy devices	An IoT network often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, and IoT networks need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet and IP.
The need for data to be analyzed in real time	Whereas traditional IT networks perform scheduled batch processing of data, IoT data needs to be analyzed and responded to in real-time.	Analytics software needs to be positioned closer to the edge and should support real-time streaming analytics. Traditional IT analytics software (such as relational databases or even Hadoop), are better suited to batch-level analytics that occur after the fact.

The following sections expand on the requirements driving specific architectural changes for IoT.

Scale

The scale of a typical IT network is on the order of several thousand devices—typically printers, mobile wireless devices, laptops, servers, and so on. The traditional three-layer campus networking model, supporting access, distribution, and core (with subarchitectures for WAN, Wi-Fi, data center, etc.), is well understood. But now consider what happens when the scale of a network goes from a few thousand endpoints to a few million. How many IT engineers have ever designed a network that is intended to support millions of routable IP endpoints? This kind of scale has only previously been seen by the Tier 1 service providers. IoT introduces a model where an average-sized utility, factory, transportation system, or city could easily be asked to support a network of this scale. Based on scale requirements of this order, IPv6 is the natural foundation for the IoT network layer.

Security

It has often been said that if World War III breaks out, it will be fought in cyberspace. We have already seen evidence of targeted malicious attacks using vulnerabilities in networked machines, such as the outbreak of the Stuxnet worm, which specifically affected Siemens programmable logic controller (PLC) systems.

The frequency and impact of cyber attacks in recent years has increased dramatically. Protecting corporate data from intrusion and theft is one of the main functions of the IT department. IT departments go to great lengths to protect servers, applications, and the network, setting up defense-in-depth models with layers of security designed to protect the cyber crown jewels of the corporation. However, despite all the efforts mustered to protect networks and data, hackers still find ways to penetrate trusted networks. In IT networks, the first line of defense is often the perimeter firewall. It would be unthinkable to position critical IT endpoints outside the firewall, visible to anyone who cared to look. However, IoT endpoints are often located in wireless sensor networks that use unlicensed spectrum and are not only visible to the world through a spectrum analyzer but often physically accessible and widely distributed in the field.

As more OT systems become connected to IP networks, their capabilities increase, but so does their potential vulnerability. For example, at 3:30 p.m. on December 23, 2015, the Ukrainian power grid experienced an unprecedented cyber attack that affected approximately 225,000 customers. This attack wasn't simply carried out by a group of opportunistic thieves; it was a sophisticated, well-planned assault on the Ukrainian power grid that targeted the SCADA (supervisory control and data acquisition) system, which governs communication to grid automation devices.

Traditional models of IT security are simply not designed for the new attack vectors introduced by highly dispersed IoT systems. IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques that understand the behavior of industrial protocols and can respond to attacks on critical infrastructure. For optimum security, IoT systems must:

- Be able to identify and authenticate all entities involved in the IoT service (that is, gateways, endpoint devices, home networks, roaming networks, service platforms)
- Ensure that all user data shared between the endpoint device and back-end applications is encrypted
- Comply with local data protection legislation so that all data is protected and stored correctly
- Utilize an IoT connectivity management platform and establish rules-based security policies so immediate action can be taken if anomalous behavior is detected from connected devices
- Take a holistic, network-level approach to security

See Chapter 8, "Securing IoT," for more information on IoT security.

Constrained Devices and Networks

Most IoT sensors are designed for a single job, and they are typically small and inexpensive. This means they often have limited power, CPU, and memory, and they transmit only when there is something important. Because of the massive scale of these devices and the large, uncontrolled environments where they are usually deployed, the networks that provide connectivity also tend to be very lossy and support very low data rates. This is a completely different situation from IT networks, which enjoy multi-gigabit connection speeds and endpoints with powerful CPUs. If an IT network has performance constraints, the solution is simple: Upgrade to a faster network. If too many devices are on one VLAN and are impacting performance, you can simply carve out a new VLAN and continue to scale as much as you need. However, this approach cannot meet the constrained nature of IoT systems. IoT requires a new breed of connectivity technologies that meet both the scale and constraint limitations. For more detailed information on constrained devices and networks, see Chapter 5, "IP as the IoT Network Layer."

Data

IoT devices generate a mountain of data. In general, most IT shops don't really care much about the unstructured chatty data generated by devices on the network. However, in IoT the data is like gold, as it is what enables businesses to deliver new IoT services that enhance the customer experience, reduce cost, and deliver new revenue opportunities. Although most IoT-generated data is unstructured, the insights it provides through analytics can revolutionize processes and create new business models. Imagine a smart city with a few hundred thousand smart streetlights, all connected through an IoT network. Although most of the information communicated between the lighting network modules and the control center is of little interest to anyone, patterns in this data can yield extremely useful insights that can help predict when lights need to be replaced or whether they can be turned on or off at certain times, thus saving operational expense. However, when all this data is combined, it can become difficult to manage and analyze effectively. Therefore, unlike IT networks, IoT systems are designed to stagger data consumption throughout the architecture, both to filter and reduce unnecessary data going upstream and to provide the fastest possible response to devices when necessary.

Legacy Device Support

Supporting legacy devices in an IT organization is not usually a big problem. If someone's computer or operating system is outdated, she simply upgrades. If someone is using a mobile device with an outdated Wi-Fi standard, such as 802.11b or 802.11g, you can simply deny him access to the wireless network, and he will be forced to upgrade. In OT systems, end devices are likely to be on the network for a very long time—sometimes decades. As IoT networks are deployed, they need to support the older devices already present on the network, as well as devices with new capabilities. In many cases, legacy devices are so old that they don't even support IP. For example, a factory may replace machines only once every 20 years—or perhaps even longer! It does not want to upgrade multi-million-dollar machines just so it can connect them to a network for better visibility and control. However, many of these legacy machines might support older protocols, such as serial interfaces, and use RS-232. In this case, the IoT network must either be capable of some type of protocol translation or use a gateway device to connect these legacy endpoints to the IoT network. Chapter 6, "Application Protocols for IoT," takes a closer look at the transport of legacy IoT protocols.

Comparing IoT Architectures

The aforementioned challenges and requirements of IoT systems have driven a whole new discipline of network architecture. In the past several years, architectural standards and frameworks have emerged to address the challenge of designing massive-scale IoT networks.

The foundational concept in all these architectures is supporting data, process, and the functions that endpoint devices perform. Two of the best-known architectures are those supported by oneM2M and the IoT World Forum (IoTWF), discussed in the following sections.

The oneM2M IoT Standardized Architecture

In an effort to standardize the rapidly growing field of machine-to-machine (M2M) communications, the European Telecommunications Standards Institute (ETSI) created the M2M Technical Committee in 2008. The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications and devices. Over time, the scope has expanded to include the Internet of Things.

Other related bodies also began to create similar M2M architectures, and a common standard for M2M became necessary. Recognizing this need, in 2012 ETSI and 13 other founding members launched oneM2M as a global initiative designed to promote efficient M2M communication systems and IoT. The goal of oneM2M is to create a common services layer, which can be readily embedded in field devices to allow communication with application servers.[1] oneM2M's framework focuses on IoT services, applications, and platforms. These include smart metering applications, smart grid, smart city automation, e-health, and connected vehicles.

One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods. By developing a horizontal platform architecture, oneM2M is developing standards that allow interoperability at all levels of the IoT stack. For example, you might want to automate your HVAC system by connecting it with wireless temperature sensors spread throughout your office. You decide to deploy sensors that use LoRaWAN technology (discussed in Chapter 4, "Connecting Smart Objects"). The problem is that the LoRaWAN network and the BACnet system that your HVAC and BMS run on are completely different systems and have no natural connection point. This is where the oneM2M common services architecture comes in. oneM2M's horizontal framework and RESTful APIs allow the LoRaWAN system to interface with the building management system over an IoT network, thus promoting end-to-end IoT communications in a consistent way, no matter how heterogeneous the networks.

Figure 2-1 illustrates the oneM2M IoT architecture.

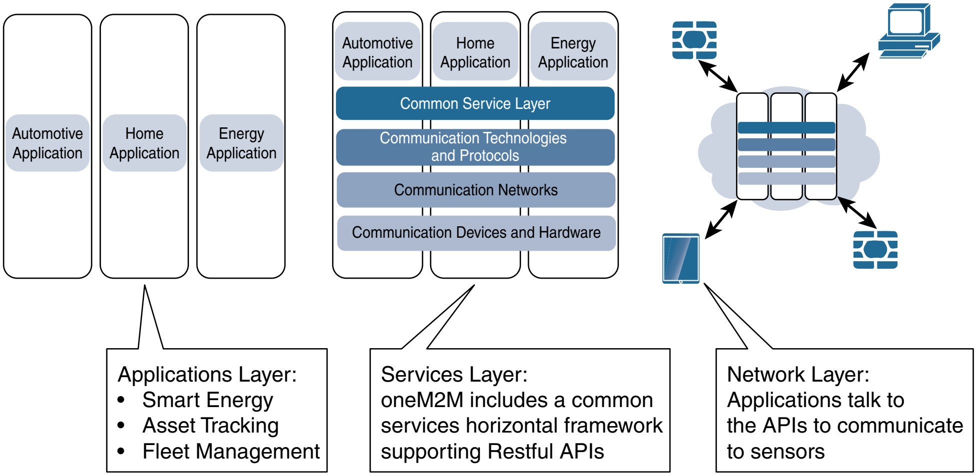


Figure 2-1 *The Main Elements of the oneM2M IoT Architecture*

The oneM2M architecture divides IoT functions into three major domains: the application layer, the services layer, and the network layer. While this architecture may seem simple and somewhat generic at first glance, it is very rich and promotes interoperability through IT-friendly APIs and supports a wide range of IoT technologies. Let's examine each of these domains in turn:

- **Applications layer:** The oneM2M architecture gives major attention to connectivity between devices and their applications. This domain includes the application-layer protocols and attempts to standardize northbound API definitions for interaction with business intelligence (BI) systems. Applications tend to be industry-specific and have their own sets of data models, and thus they are shown as vertical entities.

- **Services layer:** This layer is shown as a horizontal framework across the vertical industry applications. At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware. Examples include backhaul communications via cellular, MPLS networks, VPNs, and so on. Riding on top is the common services layer. This conceptual layer adds APIs and middleware supporting third-party services and applications. One of the stated goals of oneM2M is to "develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software nodes, and rely upon connecting the myriad of devices in the field area network to M2M application servers, which typically reside in a cloud or data center." A critical objective of oneM2M is to attract and actively involve organizations from M2M-related business domains, including telematics and intelligent transportation, healthcare, utility, industrial automation, and smart home applications, to name just a few.[2]

- **Network layer:** This is the communication domain for the IoT devices and endpoints. It includes the devices themselves and the communications network that links them. Embodiments of this communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to-multipoint systems, such as IEEE 801.11ah. Also included are wired device connections, such as IEEE 1901 power line communications. Chapter 4 provides more details on these connectivity technologies.

  In many cases, the smart (and sometimes not-so-smart) devices communicate with each other. In other cases, machine-to-machine communication is not necessary, and the devices simply communicate through a field area network (FAN) to use-case-specific apps in the IoT application domain. Therefore, the device domain also includes the gateway device, which provides communications up into the core network and acts as a demarcation point between the device and network domains.

Technical Specifications and Technical Reports published by oneM2M covering IoT functional architecture and other aspects can be found at www.onem2m.org.

The IoT World Forum (IoTWF) Standardized Architecture

In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven-layer IoT architectural reference model. While various IoT reference models exist, the one put forth by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access. It provides a succinct way of visualizing IoT from a technical perspective. Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Figure 2-2 details the IoT Reference Model published by the IoTWF.

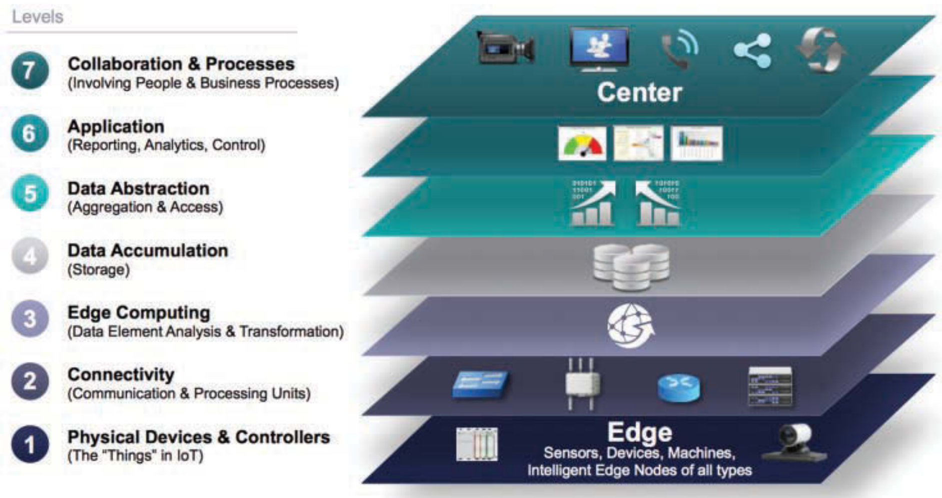


Figure 2-2 *IoT Reference Model Published by the IoT World Forum*

As shown in Figure 2-2, the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes. In general, data travels up the stack, originating from the edge, and goes northbound to the center. Using this reference model, we are able to achieve the following:

- Decompose the IoT problem into smaller parts
- Identify different technologies at each layer and how they relate to one another
- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

The following sections look more closely at each of the seven layers of the IoT Reference Model.

Layer 1: Physical Devices and Controllers Layer

The first layer of the IoT Reference Model is the physical devices and controllers layer. This layer is home to the "things" in the Internet of Things, including the various endpoint devices and sensors that send and receive information. The size of these "things" can range from almost microscopic sensors to giant machines in a factory. Their primary function is generating data and being capable of being queried and/or controlled over a network.

Layer 2: Connectivity Layer

In the second layer of the IoT Reference Model, the focus is on connectivity. The most important function of this IoT layer is the reliable and timely transmission of data. More specifically, this includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).

As you may notice, the connectivity layer encompasses all networking elements of IoT and doesn't really distinguish between the last-mile network (the network between the sensor/endpoint and the IoT gateway, discussed later in this chapter), gateway, and backhaul networks. Functions of the connectivity layer are detailed in Figure 2-3.

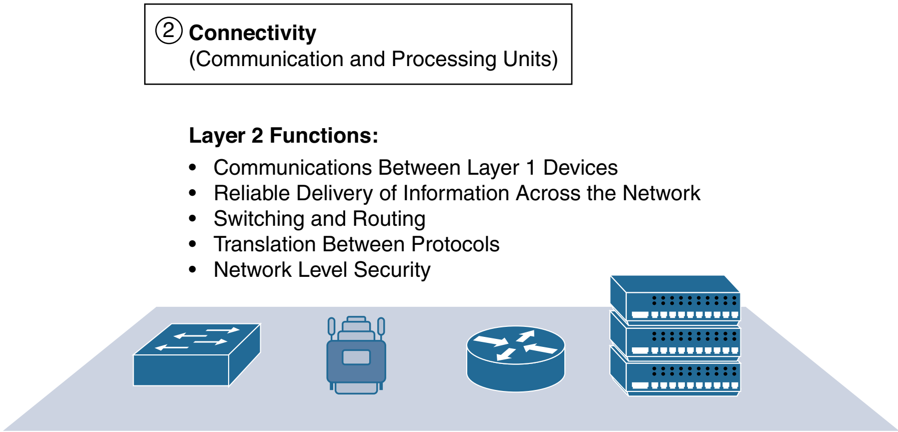


Figure 2-3 *IoT Reference Model Connectivity Layer Functions*

Layer 3: Edge Computing Layer

Edge computing is the role of Layer 3. Edge computing is often referred to as the "fog" layer and is discussed in the section "Fog Computing," later in this chapter. At this layer, the emphasis is on data reduction and converting network data flows into information that is ready for storage and processing by higher layers. One of the basic principles of this reference model is that information processing is initiated as early and as close to the edge of the network as possible. Figure 2-4 highlights the functions handled by Layer 3 of the IoT Reference Model.

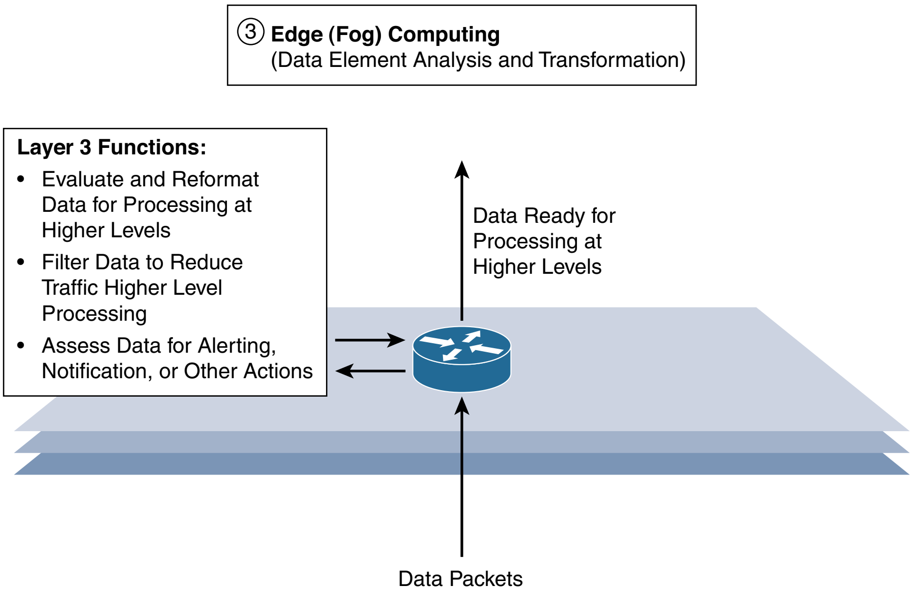


Figure 2-4 *IoT Reference Model Layer 3 Functions*

Another important function that occurs at Layer 3 is the evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer. This also allows for data to be reformatted or decoded, making additional processing by other systems easier. Thus, a critical function is assessing the data to see if predefined thresholds are crossed and any action or alerts need to be sent.

Upper Layers: Layers 4–7

The upper layers deal with handling and processing the IoT data generated by the bottom layer. For the sake of completeness, Layers 4–7 of the IoT Reference Model are summarized in Table 2-2.

Table 2-2 *Summary of Layers 4–7 of the IoTWF Reference Model*

IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

IT and OT Responsibilities in the IoT Reference Model

An interesting aspect of visualizing an IoT architecture this way is that you can start to organize responsibilities along IT and OT lines. Figure 2-5 illustrates a natural demarcation point between IT and OT in the IoT Reference Model framework.

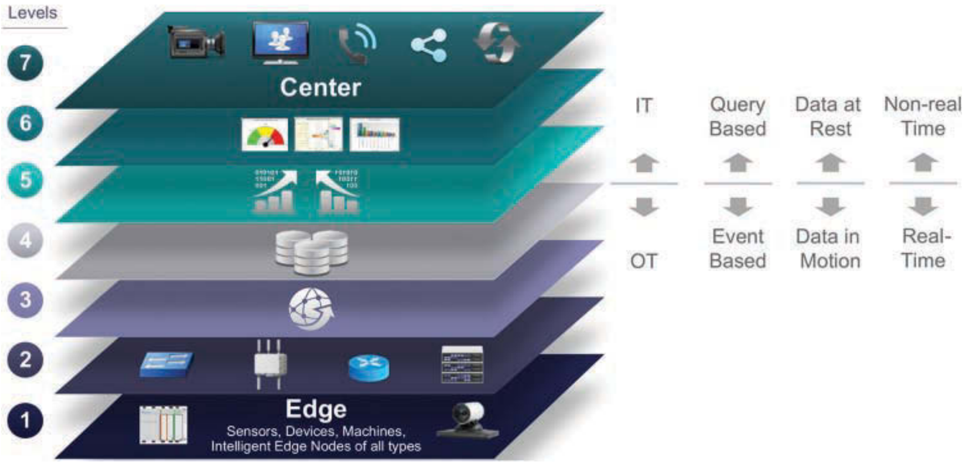


Figure 2-5 *IoT Reference Model Separation of IT and OT*

As demonstrated in Figure 2-5, IoT systems have to cross several boundaries beyond just the functional layers. The bottom of the stack is generally in the domain of OT. For an industry like oil and gas, this includes sensors and devices connected to pipelines, oil rigs, refinery machinery, and so on. The top of the stack is in the IT area and includes things like the servers, databases, and applications, all of which run on a part of the network controlled by IT. In the past, OT and IT have generally been very independent and had little need to even talk to each other. IoT is changing that paradigm.

At the bottom, in the OT layers, the devices generate real-time data at their own rate—sometimes vast amounts on a daily basis. Not only does this result in a huge amount of data transiting the IoT network, but the sheer volume of data suggests that applications at the top layer will be able to ingest that much data at the rate required. To meet this requirement, data has to be buffered or stored at certain points within the IoT stack. Layering data management in this way throughout the stack helps the top four layers handle data at their own speed.

As a result, the real-time "data in motion" close to the edge has to be organized and stored so that it becomes "data at rest" for the applications in the IT tiers. The IT and OT organizations need to work together for overall data management.

Additional IoT Reference Models

In addition to the two IoT reference models already presented in this chapter, several other reference models exist. These models are endorsed by various organizations and standards bodies and are often specific to certain industries or IoT applications. Table 2-3 highlights these additional IoT reference models.

Table 2-3 *Alternative IoT Reference Models*

IoT Reference Model	Description
Purdue Model for Control Hierarchy	The Purdue Model for Control Hierarchy (see www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.pdf) is a common and well-understood model that segments devices and equipment into hierarchical levels and functions. It is used as the basis for ISA-95 for control hierarchy, and in turn for the IEC-62443 (formerly ISA-99) cyber security standard. It has been used as a base for many IoT-related models and standards across industry. The Purdue Model's application to IoT is discussed in detail in Chapter 9, "Manufacturing," and in Chapter 10, "Oil & Gas."
Industrial Internet Reference Architecture (IIRA) by Industrial Internet Consortium (IIC)	The IIRA is a standards-based open architecture for Industrial Internet Systems (IISs). To maximize its value, the IIRA has broad industry applicability to drive interoperability, to map applicable technologies, and to guide technology and standard development. The description and representation of the architecture are generic and at a high level of abstraction to support the requisite broad industry applicability. The IIRA distills and abstracts common characteristics, features and patterns from use cases well understood at this time, predominantly those that have been defined in the IIC. For more information, see www.iiconsortium.org/IIRA.htm.
Internet of Things–Architecture (IoT-A)	IoT-A created an IoT architectural reference model and defined an initial set of key building blocks that are foundational in fostering the emerging Internet of Things. Using an experimental paradigm, IoT-A combined top-down reasoning about architectural principles and design guidelines with simulation and prototyping in exploring the technical consequences of architectural design choices. For more information, see https://vdivde-it.de/en.

A Simplified IoT Architecture

Although considerable differences exist between the aforementioned reference models, they each approach IoT from a layered perspective, allowing development of technology and standards somewhat independently at each level or domain. The commonality between these frameworks is that they all recognize the interconnection of the IoT endpoint devices to a network that transports the data where it is ultimately used by applications, whether at the data center, in the cloud, or at various management points throughout the stack.

It is not the intention of this book to promote or endorse any one specific IoT architectural framework. In fact, it can be noted that IoT architectures may differ somewhat depending on the industry use case or technology being deployed, and each has merit in solving the IoT heterogeneity problem discussed earlier. Thus, in this book we present an IoT framework that highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network. This framework is presented as two parallel stacks: The IoT Data Management and Compute Stack and the Core IoT Functional Stack. Reducing the framework down to a pair of three-layer stacks in no way suggests that the model lacks the detail necessary to develop a sophisticated IoT strategy. Rather, the intention is to simplify the IoT architecture into its most basic building blocks and then to use it as a foundation to understand key design and deployment principles that are applied to industry-specific use cases. All the layers of more complex models are still covered, but they are grouped here in functional blocks that are easy to understand. Figure 2-6 illustrates the simplified IoT model presented in this book.

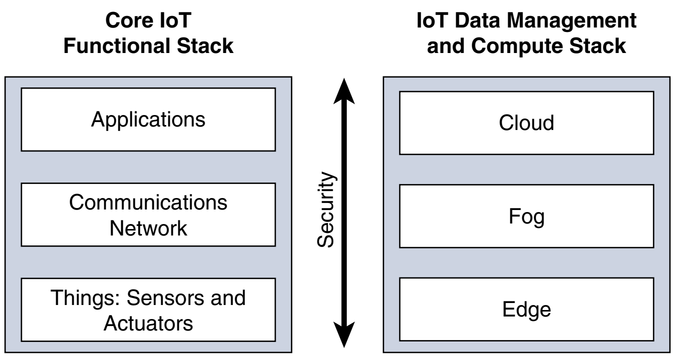


Figure 2-6 *Simplified IoT Architecture*

Nearly every published IoT model includes core layers similar to those shown on the left side of Figure 2-6, including "things," a communications network, and applications. However, unlike other models, the framework presented here separates the core IoT and data management into parallel and aligned stacks, allowing you to carefully examine the functions of both the network and the applications at each stage of a complex IoT system. This separation gives you better visibility into the functions of each layer.

The presentation of the Core IoT Functional Stack in three layers is meant to simplify your understanding of the IoT architecture into its most foundational building blocks. Of course, such a simple architecture needs to be expanded on. The network communications layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies. Consider for a moment the heterogeneity of IoT sensors and the many different ways that exist to connect them to a network. The network communications layer needs to consolidate these together, offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.

Many of the last-mile technologies used in IoT are chosen to meet the specific requirements of the endpoints and are unlikely to ever be seen in the IT domain. However, the network between the gateway and the data center is composed mostly of traditional technologies that experienced IT professionals would quickly recognize. These include tunneling and VPN technologies, IP-based quality of service (QoS), conventional Layer 3 routing protocols such as BGP and IP-PIM, and security capabilities such as encryption, access control lists (ACLs), and firewalls.

Unlike with most IT networks, the applications and analytics layer of IoT doesn't necessarily exist only in the data center or in the cloud. Due to the unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system. In the model presented in this book, data management is aligned with each of the three layers of the Core IoT Functional Stack. The three data management layers are the edge layer (data management within the sensors themselves), the fog layer (data management in the gateways and transit network), and the cloud layer (data management in the cloud or central data center). The IoT Data Management and Compute Stack is examined in greater detail later in this chapter. Figure 2-7 highlights an expanded view of the IoT architecture presented in this book.

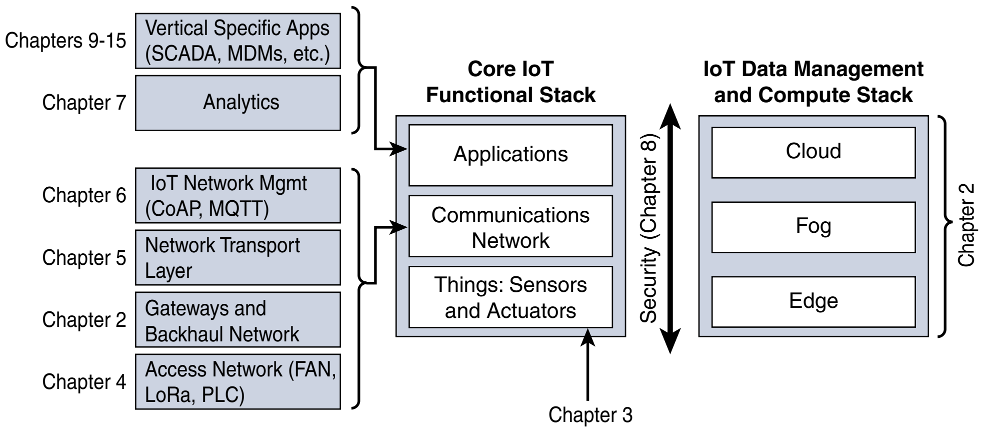


Figure 2-7 *Expanded View of the Simplified IoT Architecture*

As shown in Figure 2-7, the Core IoT Functional Stack can be expanded into sublayers containing greater detail and specific network functions. For example, the communications layer is broken down into four separate sublayers: the access network, gateways and backhaul, IP transport, and operations and management sublayers.

The applications layer of IoT networks is quite different from the application layer of a typical enterprise network. Instead of simply using business applications, IoT often involves a strong big data analytics component. One message that is stressed throughout this book is that IoT is not just about the control of IoT devices but, rather, the useful insights gained from the data generated by those devices. Thus, the applications layer typically has both analytics and industry-specific IoT control system components.

You will notice that security is central to the entire architecture, both from network connectivity and data management perspectives. The chapters in Part II, "Engineering IoT Networks," discuss security at each layer. Chapter 8 is dedicated to the subject of securing IoT systems. The industry chapters in Part III, "IoT in Industry," highlight how lessons learned in Parts I, "Introduction to IoT," and II can be applied to specific industries. Each of the Part III chapters examines the issue of IoT security for a particular sector.

The architectural framework presented in Figure 2-7 reflects the flow of the chapters in this book. To help navigate your way through this book, chapter numbers are highlighted next to the various layers of the stack.

The remainder of this chapter provides a high-level examination of each layer of this model and lays the foundation for a detailed examination of the technologies involved at each layer presented in Part II, and it gives you the tools you need to understand how these technologies are applied in key industries in Part III.

The Core IoT Functional Stack

IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services. These objects are "smart" because they use a combination of contextual information and configured goals to perform actions. These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform. In this case, the management platform can be used to process data collected from the smart object and also guide the behavior of the smart object. From an architectural standpoint, several components have to work together for an IoT network to be operational:

- **"Things" layer:** At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.
- **Communications network layer:** When smart objects are not self-contained, they need to communicate with an external system. In many cases, this communication uses a wireless technology. This layer has four sublayers:
  - **Access network sublayer:** The last mile of the IoT network is the access network. This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network may also be wired.

- **Gateways and backhaul network sublayer:** A common communication system organizes multiple smart objects in a given area around a common gateway. The gateway communicates directly with the smart objects. The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a headend central station where the information is processed. This information exchange is a Layer 7 (application) function, which is the reason this object is called a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.
- **Network transport sublayer:** For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.
- **IoT network management sublayer:** Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP and MQTT.

- **Application and analytics layer:** At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

The following sections examine these elements and help you architect your IoT communication network.

Layer 1: Things: Sensors and Actuators Layer

Most IoT networks start from the object, or "thing," that needs to be connected. Chapter 3, "Smart Objects: The 'Things' in IoT," provides more in-depth information about smart objects. From an architectural standpoint, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures. There are myriad ways to classify smart objects. One architectural classification could be:

- **Battery-powered or power-connected:** This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source. Battery-powered things can be moved more easily than line-powered objects. However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.
- **Mobile or static:** This classification is based on whether the "thing" should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor). The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

- **Low or high reporting frequency:** This classification is based on how often the object should report monitored parameters. A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

- **Simple or rich data:** This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others. Richer data typically drives higher power consumption. This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric. A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

- **Report range:** This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most. The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen. If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary. By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away.

- **Object density per cell:** This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway. An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.

From a network architectural standpoint, your initial task is to determine which technology should be used to allow smart objects to communicate. This determination depends on the way the “things” are classified. However, some industries (such as manufacturing and utilities) may include objects in various categories, matching different needs. Figure 2-8 provides some examples of applications matching the combination of mobility and throughput requirements.

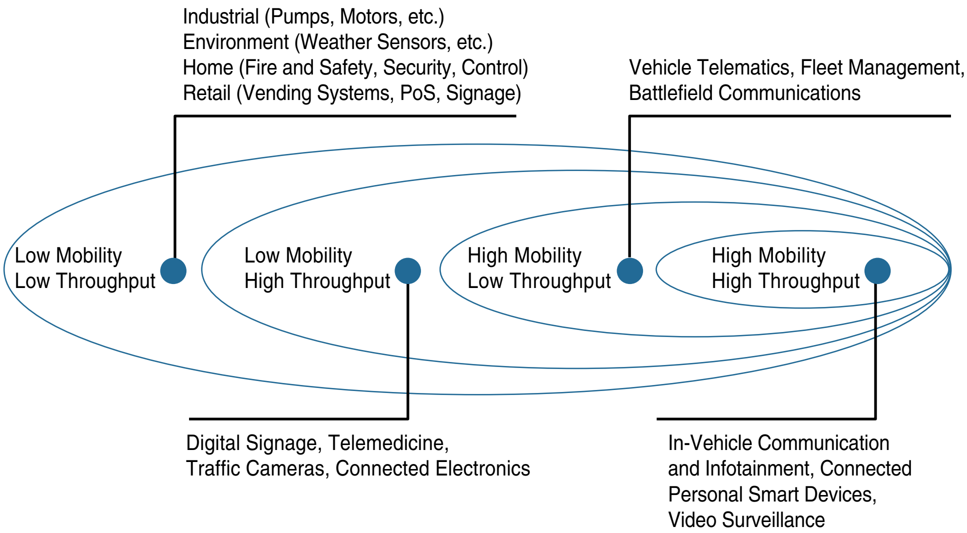


Figure 2-8 *Example of Sensor Applications Based on Mobility and Throughput*

The categories used to classify things can influence other parameters and can also influence one another. For example, a battery-operated highly mobile object (like a heart rate monitor, for example) likely has a small form factor. A small sensor is easier to move or integrate into its environment. At the same time, a small and highly mobile smart object is unlikely to require a large antenna and a powerful power source. This constraint will limit the transmission range and, therefore, the type of network protocol available for its connections. The criticality of data may also influence the form factor and, therefore, the architecture. For example, a missing monthly report from an asphalt moisture sensor may simply flag an indicator for sensor (or battery) replacement. A multi-mirror gyroscope report missing for more than 100 ms may render the entire system unstable or unusable. These sensors either need to have a constant source of power (resulting in limited mobility) or need to be easily accessible for battery replacement (resulting in limited transmission range). A first step in designing an IoT network is to examine the requirements in terms of mobility and data transmission (how much data, how often).

Layer 2: Communications Network Layer

Once you have determined the influence of the smart object form factor over its transmission capabilities (transmission range, data volume and frequency, sensor density and mobility), you are ready to connect the object and communicate.

Compute and network assets used in IoT can be very different from those in IT environments. The difference in the physical form factors between devices used by IT and OT is obvious even to the most casual of observers. What typically drives this is the physical environment in which the devices are deployed. What may not be as inherently obvious, however, is their operational differences. The operational differences must be understood in order to apply the correct handling to secure the target assets.

Temperature variances are an easily understood metric. The cause for the variance is easily attributed to external weather forces and internal operating conditions. Remote external locations, such as those associated with mineral extraction or pipeline equipment can span from the heat of the Arabian Gulf to the cold of the Alaskan North Slope. Controls near the furnaces of a steel mill obviously require heat tolerance, and controls for cold food storage require the opposite. In some cases, these controls must handle extreme fluctuations as well. These extremes can be seen within a single deployment. For example, portions of the Tehachapi, California, wind farms are located in the Mojave Desert, while others are at an altitude of 1800 m in the surrounding mountains. As you can imagine, the wide variance in temperature takes a special piece of hardware that is capable of withstanding such harsh environments.

Humidity fluctuations can impact the long-term success of a system as well. Well heads residing in the delta of the Niger River will see very different conditions from those in the middle of the Arabian Desert. In some conditions, the systems could be exposed to direct liquid contact such as may be found with outdoor wireless devices or marine condition deployments.

Less obvious are the operating extremes related to kinetic forces. Shock and vibration needs vary based on the deployment scenario. In some cases, the focus is on low-amplitude but constant vibrations, as may be expected on a bushing-mounted manufacturing system. In other cases, it could be a sudden acceleration or deceleration, such as may be experienced in peak ground acceleration of an earthquake or an impact on a mobile system such as high-speed rail or heavy-duty earth moving equipment.

Solid particulates can also impact the gear. Most IT environments must contend with dust build-up that can become highly concentrated due to the effect of cooling fans. In less-controlled IT environments, that phenomenon can be accelerated due to higher concentrations of particulates. A deterrent to particulate build-up is to use fanless cooling, which necessitates a higher surface area, as is the case with heat transfer fins.

Hazardous location design may also cause corrosive impact to the equipment. Caustic materials can impact connections over which power or communications travel. Furthermore, they can result in reduced thermal efficiency by potentially coating the heat transfer surfaces.

In some scenarios, the concern is not how the environment can impact the equipment but how the equipment can impact the environment. For example, in a scenario in which volatile gases may be present, spark suppression is a critical design criterion.

There is another class of device differentiators related to the external connectivity of the device for mounting or industrial function. Device mounting is one obvious difference between OT and IT environments. While there are rack mount environments in some industrial spaces, they are more frequently found among IT type assets. Within industrial environments, many compute and communication assets are placed within an enclosed space, such as a control cabinet where they will be vertically mounted on a DIN (Deutsches Institut für Normung) rail inside. In other scenarios, the devices might be mounted horizontally directly on a wall or on a fence.

In contrast to most IT-based systems, industrial compute systems often transmit their state or receive inputs from external devices through an alarm channel. These may drive an indicator light (stack lights) to display the status of a process element from afar. This same element can also receive inputs to initiate actions within the system itself.

Power supplies in OT systems are also frequently different from those commonly seen on standard IT equipment. A wider range of power variations are common attributes of industrial compute components. DC power sources are also common in many environments. Given the criticality of many systems, it is often required that redundant power supplies be built into the device itself. Extraneous power supplies, especially those not inherently mounted, are frowned upon, given the potential for accidental unplugging. In some utility cases, the system must be able to handle brief power outages and still continue to operate.

Access Network Sublayer

There is a direct relationship between the IoT network technology you choose and the type of connectivity topology this technology allows. Each technology was designed with a certain number of use cases in mind (what to connect, where to connect, how much data to transport at what interval and over what distance). These use cases determined the frequency band that was expected to be most suitable, the frame structure matching the expected data pattern (packet size and communication intervals), and the possible topologies that these use cases illustrate.

As IoT continues to grow exponentially, you will encounter a wide variety of applications and special use cases. For each of them, an access technology will be required. IoT sometimes reuses existing access technologies whose characteristics match more or less closely the IoT use case requirements. Whereas some access technologies were developed specifically for IoT use cases, others were not.

One key parameter determining the choice of access technology is the range between the smart object and the information collector. Figure 2-9 lists some access technologies you may encounter in the IoT world and the expected transmission distances.

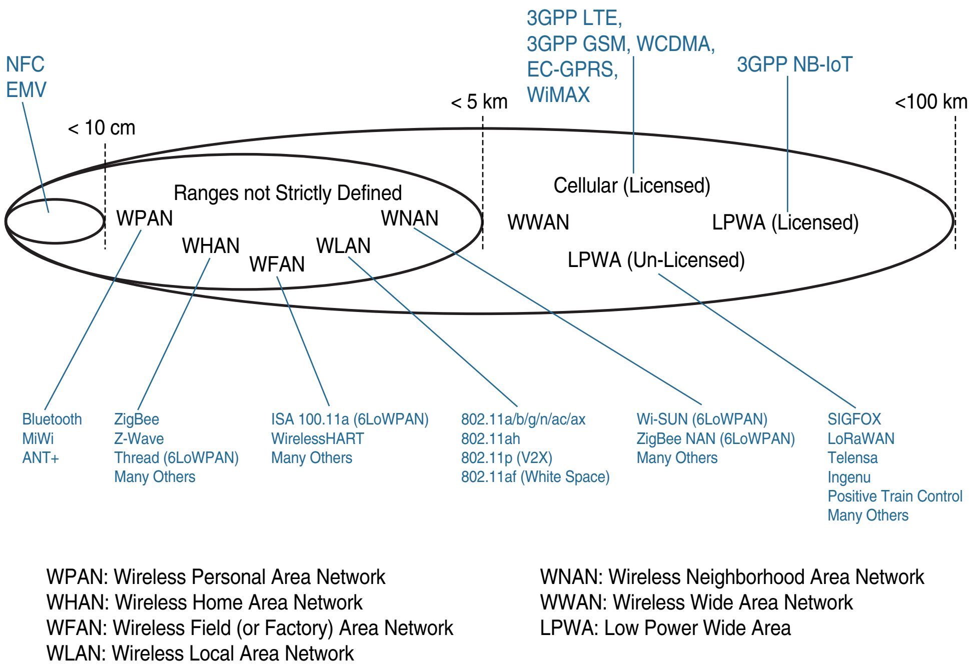


Figure 2-9 *Access Technologies and Distances*

Note that the ranges in Figure 2-9 are inclusive. For example, cellular is indicated for transmissions beyond 5 km, but you could achieve a successful cellular transmission at shorter range (for example, 100 m). By contrast, ZigBee is expected to be efficient over a range of a few tens of meters, but you would not expect a successful ZigBee transmission over a range of 10 km.

Range estimates are grouped by category names that illustrate the environment or the vertical where data collection over that range is expected. Common groups are as follows:

- **PAN (personal area network):** Scale of a few meters. This is the personal space around a person. A common wireless technology for this scale is Bluetooth.
- **HAN (home area network):** Scale of a few tens of meters. At this scale, common wireless technologies for IoT include ZigBee and Bluetooth Low Energy (BLE).
- **NAN (neighborhood area network):** Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data is collected.
- **FAN (field area network):** Scale of several tens of meters to several hundred meters. FAN typically refers to an outdoor area larger than a single group of house units. The FAN is often seen as "open space" (and therefore not secured and not controlled). A FAN is sometimes viewed as a group of NANs, but some verticals see the FAN as a group of HANs or a group of smaller outdoor cells. As you can see, FAN and NAN may sometimes be used interchangeably. In most cases, the vertical context is clear enough to determine the grouping hierarchy.

- **LAN (local area network):** Scale of up to 100 m. This term is very common in networking, and it is therefore also commonly used in the IoT space when standard networking technologies (such as Ethernet or IEEE 802.11) are used. Other networking classifications, such as MAN (metropolitan area network, with a range of up to a few kilometers) and WAN (wide area network, with a range of more than a few kilometers), are also commonly used.

Note that for all these places in the IoT network, a "W" can be added to specifically indicate wireless technologies used in that space. For example, HomePlug is a wired technology found in a HAN environment, but a HAN is often referred to as a WHAN (wireless home area network) when a wireless technology, like ZigBee, is used in that space.

Similar achievable distances do not mean similar protocols and similar characteristics. Each protocol uses a specific frame format and transmission technique over a specific frequency (or band). These characteristics introduce additional differences. For example, Figure 2-10 demonstrates four technologies representing WHAN to WLAN ranges and compares the throughput and range that can be achieved in each case. Figure 2-10 supposes that the sensor uses the same frame size, transmit power, and antenna gain. The slope of throughput degradation as distance increases varies vastly from one technology to the other. This difference limits the amount of data throughput that each technology can achieve as the distance from the sensor to the receiver increases.

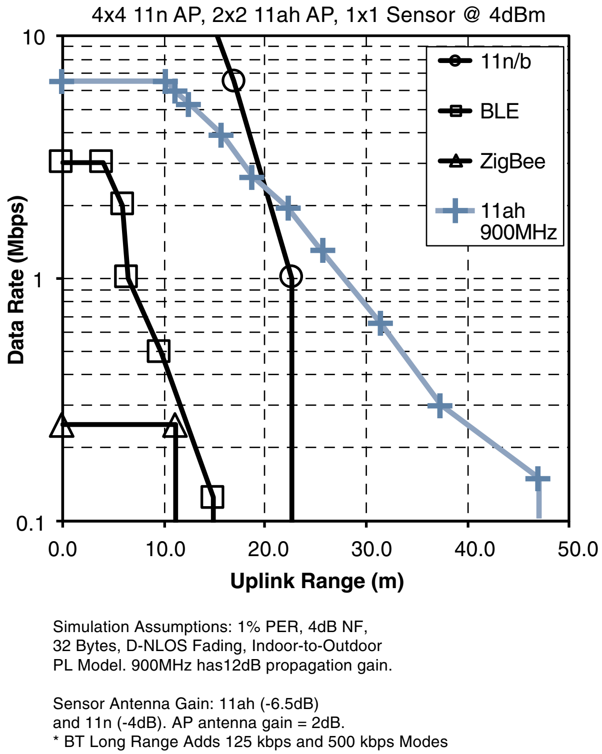


Figure 2-10 *Range Versus Throughput for Four WHAN to WLAN Technologies*

Increasing the throughput and achievable distance typically comes with an increase in power consumption. Therefore, after determining the smart object requirements (in terms of mobility and data transfer), a second step is to determine the target quantity of objects in a single collection cell, based on the transmission range and throughput required. This parameter in turn determines the size of the cell.

It may be tempting to simply choose the technology with the longest range and highest throughput. However, the cost of the technology is a third determining factor. Figure 2-11 combines cost, range, power consumption, and typical available bandwidth for common IoT access technologies.

The amount of data to carry over a given time period along with correlated power consumption (driving possible limitations in mobility and range) determines the wireless cell size and structure.

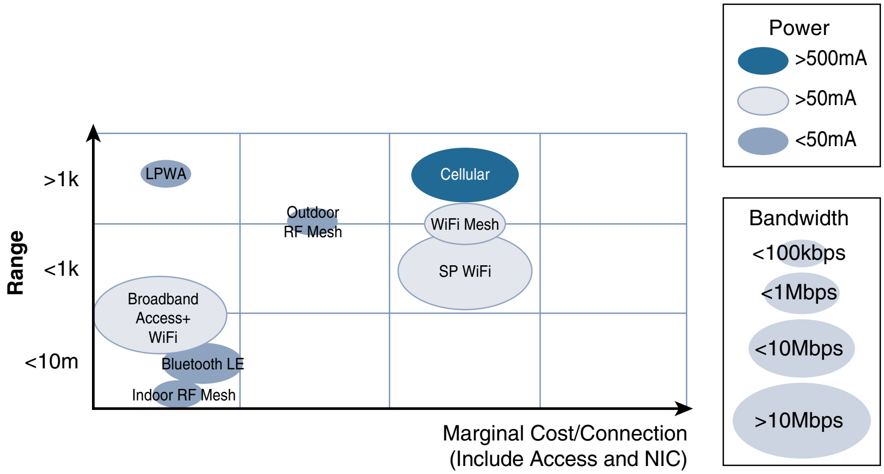


Figure 2-11 *Comparison Between Common Last-Mile Technologies in Terms of Range Versus Cost, Power, and Bandwidth*

Similar ranges also do not mean similar topologies. Some technologies offer flexible connectivity structure to extend communication possibilities:

- **Point-to-point topologies:** These topologies allow one point to communicate with another point. This topology in its strictest sense is uncommon for IoT access, as it would imply that a single object can communicate only with a single gateway. However, several technologies are referred to as "point-to-point" when each object establishes an individual session with the gateway. The "point-to-point" concept, in that case, often refers to the communication structure more than the physical topology.

- **Point-to-multipoint topologies:** These topologies allow one point to communicate with more than one other point. Most IoT technologies where one or more than one gateways communicate with multiple smart objects are in this category. However, depending on the features available on each communicating mode, several subtypes need to be considered. A particularity of IoT networks is that some nodes (for example, sensors) support both data collection and forwarding functions, while some other nodes (for example, some gateways) collect the smart object data, sometimes instruct the sensor to perform specific operations, and also interface with other networks or possibly other gateways. For this reason, some technologies categorize the nodes based on the functions (described by a protocol) they implement.

An example of a technology that categorizes nodes based on their function is IEEE 802.15.4, which is covered in depth in Chapter 4. Although 802.15.4 is used as an example in this section, the same principles may apply to many other technologies. Applications leveraging IEEE 802.15.4 commonly rely on the concept of an end device (a sensor) collecting data and transmitting the data to a collector. Sensors need to be small and are often mobile (or movable). When mobile, these sensors are therefore commonly battery operated.

To form a network, a device needs to connect with another device. When both devices fully implement the protocol stack functions, they can form a peer-to-peer network. However, in many cases, one of the devices collects data from the others. For example, in a house, temperature sensors may be deployed in each room or each zone of the house, and they may communicate with a central point where temperature is displayed and controlled. A room sensor does not need to communicate with another room sensor. In that case, the control point is at the center of the network. The network forms a star topology, with the control point at the hub and the sensors at the spokes.

In such a configuration, the central point can be in charge of the overall network coordination, taking care of the beacon transmissions and connection to each sensor. In the IEEE 802.15.4 standard, the central point is called a *coordinator* for the network. With this type of deployment, each sensor is not intended to do anything other than communicate with the coordinator in a master/slave type of relationship. The sensor can implement a subset of protocol functions to perform just a specialized part (communication with the coordinator). Such a device is called a reduced-function device (RFD). An RFD cannot be a coordinator. An RFD also cannot implement direct communications to another RFD.

The coordinator that implements the full network functions is called, by contrast, a full-function device (FFD). An FFD can communicate directly with another FFD or with more than one FFD, forming multiple peer-to-peer connections. Topologies where each FFD has a unique path to another FFD are called cluster tree topologies. FFDs in the cluster tree may have RFDs, resulting in a cluster star topology. Figure 2-12 illustrates these topologies.

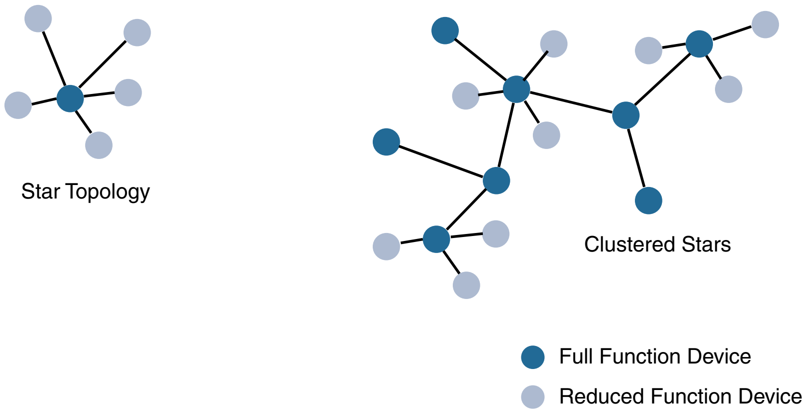


Figure 2-12 *Star and Clustered Star Topologies*

Other point-to-multipoint technologies allow a node to have more than one path to another node, forming a mesh topology. This redundancy means that each node can communicate with more than just one other node. This communication can be used to directly exchange information between nodes (the receiver directly consumes the information received) or to extend the range of the communication link. In this case, an intermediate node acts as a relay between two other nodes. These two other nodes would not be able to communicate successfully directly while respecting the constraints of power and modulation dictated by the PHY layer protocol. Range extension typically comes at the price of slower communications (as intermediate nodes need to spend time relaying other nodes' messages). An example of a technology that implements a mesh topology is Wi-Fi mesh.

Another property of mesh networks is redundancy. The disappearance of one node does not necessarily interrupt network communications. Data may still be relayed through other nodes to reach the intended destination.

Figure 2-13 shows a mesh topology. Nodes A and D are too far apart to communicate directly. In this case, communication can be relayed through nodes B or C. Node B may be used as the primary relay. However, the loss of node B does not prevent the communication between nodes A and D. Here, communication is rerouted through another node, node C.

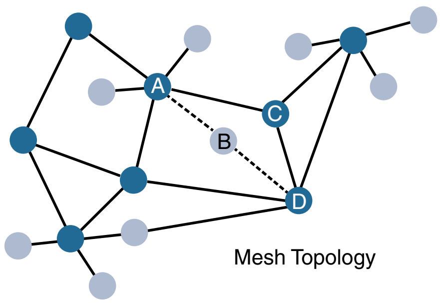


Figure 2-13 *Mesh Topology*

Note Figure 2-13 shows a partial mesh topology, where a node can communicate with more than one other node, but not all nodes communicate directly with all other nodes. In a full mesh topology each node communicates with each other node. In the topology shown in Figure 2-13, which has 17 nodes, a full mesh structure would mean that each node would have 16 connections (one to each other node). Full mesh structures are computationally expensive (as each node needs to maintain a connection to each other node). In the IoT space, full mesh deployments are uncommon. In most cases, information has to travel to a target destination rather than being directly distributed to all other nodes. Full mesh topologies also limit the acceptable distance between nodes (as all nodes must be in range of all other nodes).

Note Do not confuse *topology* and *range*. Topology describes the organization of the nodes, while range is dictated by factors such as the frequency or operation, the signal structure, and operational bandwidth. For example, both IEEE 802.15.4 and LoRaWAN implement star topologies, but the range of IEEE 802.15.4 is a few tens of meters, while LoRaWAN can achieve a successful signal over many kilometers. The bandwidth and signal structure (modulation) are very different. Figure 2-11 helps you compare the use cases and implementation considerations (range, cost, available bandwidth) for common IoT access technologies. Chapter 4 describes in detail IEEE 802.15.4, LTE, LoRaWAN, and other competing IoT technologies. However, keep in mind that many technologies that were not initially designed for IoT usage can be leveraged by specific applications. For example, remote sites may need to leverage satellite communications when standard IoT wireless technologies cannot achieve the range required. Also, the adoption of technology can vary widely over time, based on use cases, technology maturity, and other factors. For example, cellular technologies were initially designed for voice communications. The burst of data traffic that accompanied the explosion of mobile devices in the early 2000s brought the development of enhanced standards for cellular communications (with LTE). In turn, this enhancement allowed LTE to grow rapidly as a major connection technology for FANs.

Gateways and Backhaul Sublayer

Data collected from a smart object may need to be forwarded to a central station where data is processed. As this station is often in a different location from the smart object, data directly received from the sensor through an access technology needs to be forwarded to another medium (the backhaul) and transported to the central station. The gateway is in charge of this inter-medium communication.

In most cases, the smart objects are static or mobile within a limited area. The gateway is often static. However, some IoT technologies do not apply this model. For example, dedicated short-range communication (DSRC) allows vehicle-to-vehicle and vehicle-to-infrastructure communication. In this model, the smart object's position relative to the gateway is static. The car includes sensors and one gateway. Communication between the sensors and the gateway may involve wired or wireless technologies. Sensors may also be integrated into the road infrastructure and connect over a wired or wireless technology to a gateway on the side of the road. A wireless technology (DSRC operates in the upper 5 GHz range) is used for backhaul communication, peer-to-peer, or mesh communication between vehicles.

In the DSRC case, the entire "sensor field" is moving along with the gateway, but the general principles of IoT networking remain the same. The range at which DSRC can communicate is limited. Similarly, for all other IoT architectures, the choice of a backhaul technology depends on the communication distance and also on the amount of data that needs to be forwarded. When the smart object's operation is controlled from a local site, and when the environment is stable (for example, factory or oil and gas field), Ethernet can be used as a backhaul. In unstable or changing environments (for example, open mines) where cables cannot safely be run, a wireless technology is used. Wi-Fi is common in this case, often with multiple hops between the sensor field and the operation center. Mesh is a common topology to allow communication flexibility in this type of dynamic environment.

However, throughput decreases as node-to-node distance increases, and it also decreases as the number of hops increases. In a typical Wi-Fi mesh network, throughput halves for each additional hop. Some technologies, like 802.11ah, implement Wi-Fi in a lower band (lower than 1 GHz instead of 2.4 GHz/5 GHz for classical Wi-Fi) with special provisions adapted to IoT, to achieve a longer range (up to about 2 km). Beyond that range, other technologies are needed.

WiMAX (802.16) is an example of a longer-range technology. WiMAX can achieve ranges of up to 50 kilometers with rates of up to 70 Mbps. Obviously, you cannot achieve maximum rate at maximum range; you could expect up to 70 Mbps at short range and 2 to 3 Mbps at maximum range. 802.16d (also called Fixed WiMAX) describes the backhaul implementation of the protocol. Improvements to this aspect have been published (802.16.1), but most WiMAX networks still implement a variation of 802.16d. 802.16 can operate in unlicensed bands, but its backhaul function is often deployed in more-reliable licensed bands, where interferences from other systems are better controlled.

As licensed bands imply the payment of a usability fee, other cellular technologies also grew as competitive solutions for the backhaul part to achieve similar range. The choice of WiMAX or a cellular technology depends on the vertical and the location (local preferences, local costs). Chapter 4 offers an in-depth look at the most commonly deployed protocols for this segment, and Table 2-4 compares the main solutions from an architectural angle.

Table 2-4 *Architectural Considerations for WiMAX and Cellular Technologies*

Technology	Type and Range	Architectural Characteristics
Ethernet	Wired, 100 m max	Requires a cable per sensor/sensor group; adapted to static sensor position in a stable environment; range is limited; link is very reliable
Wi-Fi (2.4 GHz, 5 GHz)	Wireless, 100 m (multipoint) to a few kilometers (P2P)	Can connect multiple clients (typically fewer than 200) to a single AP; range is limited; adapted to cases where client power is not an issue (continuous power or client battery recharged easily); large bandwidth available, but interference from other systems likely; AP needs a cable
802.11ah (HaloW, Wi-Fi in sub-1 GHz)	Wireless, 1.5 km (multipoint), 10 km (P2P)	Can connect a large number of clients (up to 6000 per AP); longer range than traditional Wi-Fi; power efficient; limited bandwidth; low adoption; and cost may be an issue
WiMAX (802.16)	Wireless, several kilometers (last mile), up to 50 km (backhaul)	Can connect a large number of clients; large bandwidth available in licensed spectrum (fee-based); reduced bandwidth in license-free spectrum (interferences from other systems likely); adoption varies on location
Cellular (for example, LTE)	Wireless, several kilometers	Can connect a large number of clients; large bandwidth available; licensed spectrum (interference-free; license-based)

Network Transport Sublayer

The previous section describes a hierarchical communication architecture in which a series of smart objects report to a gateway that conveys the reported data over another medium and up to a central station. However, practical implementations are often flexible, with multiple transversal communication paths. For example, consider the case of IoT for the energy grid. Your house may have a meter that reports the energy consumption to a gateway over a wireless technology. Other houses in your neighborhood (NAN) make the same report, likely to one or several gateways. The data to be transported is small and the interval is large (for example, four times per hour), resulting in a low-mobility, low-throughput type of data structure, with transmission distances up to a mile. Several technologies (such as 802.11ah, 802.15.4, or LPWA) can be used for this collection segment. Other neighborhoods may also connect the same way, thus forming a FAN.

For example, the power utility's headend application server may be regional, and the gateway may relay to a wired or wireless backhaul technology. The structure appears to be hierarchical. Practically, however, this IoT system may achieve more than basic upstream reporting. If your power consumption becomes unusually high, the utility headend application server may need on-demand reporting from your meter at short intervals to follow the consumption trend. From a standard vertical push model, the transport structure changes and becomes bidirectional (downstream pull model instead of upstream push).

Distribution automation (DA) also allows your meter to communicate with neighboring meters or other devices in the electrical distribution grid. With such communication, consumption load balancing may be optimized. For example, your air conditioning pulses fresh air at regular intervals. With DA, your neighbor's AC starts pulsing when your system pauses; in this way, the air in both houses is kept fresh, but the energy consumed from the network is stable instead of spiking up and down with uncoordinated start and stop points. Here again, the transport model changes. From a vertical structure, you are now changing to a possible mesh structure with multiple peer-to-peer exchanges.

Similarly, your smart meter may communicate with your house appliances to evaluate their type and energy demand. With this scheme, your washing machine can be turned on in times of lower consumption from other systems, such as at night, while power to your home theater system will never be deprived, always turning on when you need it. Once the system learns your consumption pattern, charging of your electric car can start and stop at intervals to achieve the same overnight charge without creating spikes in energy demand. When these functions appear, the transport model changes again. A mesh system may appear at the scale of the house. More commonly, a partial mesh appears, with some central nodes connecting to multiple other nodes. Data may flow locally, or it may have to be orchestrated by a central application to coordinate the power budget between houses.

In this smart system, your car's charging system is connected to your energy account. As you plug into a public charging station, your car logs into the system to be identified and uniquely links to your account. At regular intervals, the central system may need to query all the charging stations for status update. The transport structure loses its vertical organization a bit more in this model, as you may be connecting from anywhere. In a managed environment, the headend system needs to upgrade the software on your meter, just as appliance vendors may need to update your oven or washing machine smart energy software. From a bottom-up data transport flow, you now implement top-down data flows.

This communication structure thus may involve peer-to-peer (for example, meter to meter), point-to-point (meter to headend station), point-to-multipoint (gateway or headend to multiple meters), unicast and multicast communications (software update to one or multiple systems). In a multitenant environment (for example, electricity and gas consumption management), different systems may use the same communication pathways. This communication occurs over multiple media (for example, power lines inside your house or a short-range wireless system like indoor Wi-Fi and/or ZigBee), a longer-range wireless system to the gateway, and yet another wireless or wired medium for backhaul transmission.

To allow for such communication structure, a network protocol with specific characteristics needs to be implemented. The protocol needs to be open and standard-based to accommodate multiple industries and multiple media. Scalability (to accommodate thousands or millions of sensors in a single network) and security are also common requirements. IP is a protocol that matches all these requirements. The advantages of IP are covered in depth in Chapter 5.

The flexibility of IP allows this protocol to be embedded in objects of very different natures, exchanging information over very different media, including low-power, lossy, and low-bandwidth networks. For example, RFC 2464 describes how an IPv6 packet gets encapsulated over an Ethernet frame and is also used for IEEE 802.11 Wi-Fi. Similarly, the IETF 6LoWPAN working group specifies how IPv6 packets are carried efficiently over lossy networks, forming an "adaption layer" for IPv6, primarily for IoT networks. Chapter 4 provides more details on 6LoWPAN and its capabilities.

Finally, the transport layer protocols built above IP (UDP and TCP) can easily be leveraged to decide whether the network should control the data packet delivery (with TCP) or whether the control task should be left to the application (UDP). UDP is a much lighter and faster protocol than TCP. However, it does not guarantee packet delivery. Both TCP and UDP can be secured with TLS/SSL (TCP) or DTLS (UDP). Chapter 6 takes a closer look at TCP and UDP for IoT networks.

IoT Network Management Sublayer

IP, TCP, and UDP bring connectivity to IoT networks. Upper-layer protocols need to take care of data transmission between the smart objects and other systems. Multiple protocols have been leveraged or created to solve IoT data communication problems. Some networks rely on a push model (that is, a sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (that is, an application queries the sensor over the network), and multiple hybrid approaches are also possible.

Following the IP logic, some IoT implementers have suggested HTTP for the data transfer phase. After all, HTTP has a client and server component. The sensor could use the client part to establish a connection to the IoT central application (the server), and then data can be exchanged. You can find HTTP in some IoT applications, but HTTP is something of a fat protocol and was not designed to operate in constrained environments with low memory, low power, low bandwidth, and a high rate of packet failure. Despite these limitations, other web-derived protocols have been suggested for the IoT space. One example is WebSocket. WebSocket is part of the HTML5 specification, and provides a simple bidirectional connection over a single connection. Some IoT solutions use WebSocket to manage the connection between the smart object and an external application. WebSocket is often combined with other protocols, such as MQTT (described shortly) to handle the IoT-specific part of the communication.

With the same logic of reusing well-known methods, Extensible Messaging and Presence Protocol (XMPP) was created. XMPP is based on instant messaging and presence. It allows the exchange of data between two or more systems and supports presence and contact list maintenance. It can also handle publish/subscribe, making it a good choice for distribution of information to multiple devices. A limitation of XMPP is its reliance on TCP, which may force subscribers to maintain open sessions to other systems and may be a limitation for memory-constrained objects.

To respond to the limits of web-based protocols, another protocol was created by the IETF Constrained Restful Environments (CoRE) working group: Constrained Application

Protocol (CoAP). CoAP uses some methods similar to those of HTTP (such as Get, Post, Put, and Delete) but implements a shorter list, thus limiting the size of the header. CoAP also runs on UDP (whereas HTTP typically uses TCP). CoAP also adds a feature that is lacking in HTTP and very useful for IoT: observation. Observation allows the streaming of state changes as they occur, without requiring the receiver to query for these changes.

Another common IoT protocol utilized in these middle to upper layers is Message Queue Telemetry Transport (MQTT). MQTT uses a broker-based architecture. The sensor can be set to be an MQTT publisher (publishes a piece of information), the application that needs to receive the information can be set as the MQTT subscriber, and any intermediary system can be set as a broker to relay the information between the publisher and the subscriber(s). MQTT runs over TCP. A consequence of the reliance on TCP is that an MQTT client typically holds a connection open to the broker at all times. This may be a limiting factor in environments where loss is high or where computing resources are limited.

Chapter 6 examines in more detail the various IoT application protocols, including CoAP and MQTT. From an architectural standpoint, you need to determine the requirements of your application protocol. Relying on TCP implies maintaining sessions between endpoints. The advantage of reliability comes with the cost of memory and processing resources consumed for session awareness. Relying on UDP delegates the control to the upper layers. You also need to determine the requirements for QoS with different priority levels between the various messages. Finally, you need to evaluate the security of the IoT application protocol to balance the level of security provided against the overhead required. Chapter 8 describes how to evaluate the security aspect of IoT networks.

Layer 3: Applications and Analytics Layer

Once connected to a network, your smart objects exchange information with other systems. As soon as your IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications that make use of the information exchanged with the smart objects.

Analytics Versus Control Applications

Multiple applications can help increase the efficiency of an IoT network. Each application collects data and provides a range of functions based on analyzing the collected data. It can be difficult to compare the features offered. Chapter 7, “Data and Analytics for IoT,” provides an in-depth analysis of the various application families. From an architectural standpoint, one basic classification can be as follows:

- **Analytics application:** This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed. The display can be about any aspect of the IoT network, from historical reports, statistics, or trends to individual system states. The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

- **Control application:** This type of application controls the behavior of the smart object or the behavior of an object related to the smart object. For example, a pressure sensor may be connected to a pump. A control application increases the pump speed when the connected sensor detects a drop in pressure. Control applications are very useful for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object, either because the configured changes are too complex to fit into the local system or because the configured changes rely on parameters that include elements outside the IoT object.

  An example of control system architecture is SCADA. SCADA was developed as a universal method to access remote systems and send instructions. One example where SCADA is widely used is in the control and monitoring of remote terminal units (RTUs) on the electrical distribution grid.

Many advanced IoT applications include both analytics and control modules. In most cases, data is collected from the smart objects and processed in the analytics module. The result of this processing may be used to modify the behavior of smart objects or systems related to the smart objects. The control module is used to convey the instructions for behavioral changes. When evaluating an IoT data and analytics application, you need to determine the relative depth of the control part needed for your use case and match it against the type of analytics provided.

Data Versus Network Analytics

Analytics is a general term that describes processing information to make sense of collected data. In the world of IoT, a possible classification of the analytics function is as follows:

- **Data analytics:** This type of analytics processes the data collected by smart objects and combines it to provide an intelligent view related to the IoT system. At a very basic level, a dashboard can display an alarm when a weight sensor detects that a shelf is empty in a store. In a more complex case, temperature, pressure, wind, humidity, and light levels collected from thousands of sensors may be combined and then processed to determine the likelihood of a storm and its possible path. In this case, data processing can be very complex and may combine multiple changing values over complex algorithms. Data analytics can also monitor the IoT system itself. For example, a machine or robot in a factory can report data about its own movements. This data can be used by an analytics application to report degradation in the movement speeds, which may be indicative of a need to service the robot before a part breaks.

- **Network analytics:** Most IoT systems are built around smart objects connected to the network. A loss or degradation in connectivity is likely to affect the efficiency of the system. Such a loss can have dramatic effects. For example, open mines use wireless networks to automatically pilot dump trucks. A lasting loss of connectivity may result in an accident or degradation of operations efficiency (automated dump trucks typically stop upon connectivity loss). On a more minor scale, loss of connectivity means that data stops being fed to your data analytics platform, and the system stops making intelligent analyses of the IoT system. A similar consequence is that the control module cannot modify local object behaviors anymore.

Most analytics applications employ both data and network analytics modules. When architecting an IoT system, you need to evaluate the need for each one. Network analytics is necessary for connected systems. However, the depth of analysis depends on your use cases. A basic connectivity view may be enough if the smart objects report occasional status, without expectation for immediate action based on this report. Detailed analysis and trending about network performance are needed if the central application is expected to pilot in near-real-time connected systems.

Data analytics is a wider space with a larger gray area (in terms of needs) than network analytics. Basic systems analytics can provide views of the system state and state trend analysis. More advanced systems can refine the type of data collected and display additional information about the system. The type of collected data and processing varies widely with the use case.

Data Analytics Versus Business Benefits

Data analytics is undoubtedly a field where the value of IoT is booming. Almost any object can be connected, and multiple types of sensors can be installed on a given object. Collecting and interpreting the data generated by these devices is where the value of IoT is realized.

From an architectural standpoint, you can define static IoT networks where a clear list of elements to monitor and analytics to perform are determined. Such static systems are common in industrial environments where the IoT charter is about providing a clear view of the state of the operation. However, a smarter architectural choice may be to allow for an open system where the network is engineered to be flexible enough that other sensors may be added in the future, and where both upstream and downstream operations are allowed. This flexibility allows for additional processing of the existing sensors and also deeper and more efficient interaction with the connected objects. This enhanced data processing can result in new added value for businesses that are not envisioned at the time when the system is initially deployed.

An example of a flexible analytics and control application is Cisco Jasper, which provides a turnkey cloud-based platform for IoT management and monetization. Consider the case of vending machines deployed throughout a city. At a basic level, these machines can be connected, and sensors can be deployed to report when a machine is in an error state. A repair person can be sent to address the issue when such a state is identified. This type of alert is a time saver and avoids the need for the repair team to tour all the machines in turn when only one may be malfunctioning.

This alert system may also avoid delay between the time when a machine goes into the error state and the time when a repair team visits the machine location. With a static platform, this use case is limited to this type of alert. With a flexible platform like Cisco Jasper, new applications may be imagined and developed over time. For example, the machine sensors can be improved to also report when an item is sold. The central application can then be enhanced to process this information and analyze what item is most sold, in what location, at what times. This new view of the machines may allow for an optimization of the items to sell in machines in a given area. Systems may be implemented to adapt the goods to time, season, or location—or many other parameters that may have been analyzed. In short, architecting open systems opens the possibility for new applications.

Smart Services

The ability to use IoT to improve operations is often termed "smart services." This term is generic, and in many cases the term is used but its meaning is often stretched to include one form of service or another where an additional level of intelligence is provided.

Fundamentally, smart services use IoT and aim for efficiency. For example, sensors can be installed on equipment to ensure ongoing conformance with regulations or safety requirements. This angle of efficiency can take multiple forms, from presence sensors in hazardous areas to weight threshold violation detectors on trucks.

Smart services can also be used to measure the efficiency of machines by detecting machine output, speed, or other forms of usage evaluation. Entire operations can be optimized with IoT. In hospitality, for example, presence and motion sensors can evaluate the number of guests in a lobby and redirect personnel accordingly. The same type of action can be taken in a store where a customer is detected as staying longer than the typical amount of time in front of a shelf. Personnel can be deployed to provide assistance. Movement of people and objects on factory floors can be analyzed to optimize the production flow.

Smart services can be integrated into an IoT system. For example, sensors can be integrated in a light bulb. A sensor can turn a light on or off based on the presence of a human in the room. An even smarter system can communicate with other systems in the house, learn the human movement pattern, and anticipate the presence of a human, turning on the light just before the person enters the room. An even smarter system can use smarter sensors that analyze multiple parameters to detect human mood and modify accordingly the light color to adapt to the learned preferences, or to convey either a more relaxing or a more dynamic environment.

Light bulbs are a simple example. By connecting to other systems in the house, efficiencies can be coordinated. For example, the house entry alarm system or the heating system can coordinate with the presence detector in a light bulb to adapt to detected changes. The alarm system can disable volumetric movement alarms in zones where a known person is detected. The heating system can adapt the temperature to human presence or detected personal preferences.

Similar efficiency can be extended to larger systems than a house. For example, smart grid applications can coordinate the energy consumption between houses to regulate the energy demand from the grid. We already mentioned that your washing machine may be turned on at night when the energy demand for heating and cooling is lower. Just as your air conditioning pulses can be coordinated with your neighbor's, your washing machine cycles can be coordinated with the appliances in your house and in the neighborhood to smooth the energy demand spikes on the grid.

Efficiency also applies to M2M communications. In mining environments, vehicles can communicate to regulate the flows between drills, draglines, bulldozers, and dump trucks, for example, making sure that a dump truck is always available when a bulldozer needs it. In smart cities, vehicles communicate. A traffic jam is detected and anticipated automatically by public transportation, and the system can temporarily reroute buses or regulate the number of buses servicing a specific line based on traffic and customer quantity, instantaneous or learned over trending.

Part III of this book provides detailed examples of how IoT is shaping specific industries. The lessons learned are always that architecting open IoT systems allows for increased efficiency over time. New applications and possibilities for an IoT system will appear in the upcoming years. When building an IoT network, you should make sure to keep the system open for the possibility of new smart objects and more traffic on the system.

IoT Data Management and Compute Stack

One of the key messages in the first two chapters of this book is that the massive scale of IoT networks is fundamentally driving new architectures. For instance, Figure 1-2 in Chapter 1 illustrates how the "things" connected to the Internet are continuing to grow exponentially, with a prediction by Cisco that by 2020 there will be more than 50 billion devices connected to some form of an IP network. Clearly, traditional IT networks are not prepared for this magnitude of network devices. However, beyond the network architecture itself, consider the data that is generated by these devices. If the number of devices is beyond conventional numbers, surely the data generated by these devices must also be of serious concern.

In fact, the data generated by IoT sensors is one of the single biggest challenges in building an IoT system. In the case of modern IT networks, the data sourced by a computer or server is typically generated by the client/server communications model, and it serves the needs of the application. In sensor networks, the vast majority of data generated is unstructured and of very little use on its own. For example, the majority of data generated by a smart meter is nothing more than polling data; the communications system simply determines whether a network connection to the meter is still active. This data on its own is of very little value. The real value of a smart meter is the metering data read by the meter management system (MMS). However, if you look at the raw polling data from a different perspective, the information can be very useful. For example, a utility may have millions of meters covering its entire service area. If whole sections of the smart grid start to show an interruption of connectivity to the meters, this data can be analyzed and combined with other sources of data, such as weather reports and electrical demand in the grid, to provide a complete picture of what is happening. This information can help determine whether the loss of connection to the meters is truly a loss of power or whether some other problem has developed in the grid. Moreover, analytics of this data can help the utility quickly determine the extent of the service outage and repair the disruption in a timely fashion.

In most cases, the processing location is outside the smart object. A natural location for this processing activity is the cloud. Smart objects need to connect to the cloud, and data processing is centralized. One advantage of this model is simplicity. Objects just need to connect to a central cloud application. That application has visibility over all the IoT nodes and can process all the analytics needed today and in the future.

However, this model also has limitations. As data volume, the variety of objects connecting to the network, and the need for more efficiency increase, new requirements appear, and those requirements tend to bring the need for data analysis closer to the IoT system. These new requirements include the following:

- **Minimizing latency:** Milliseconds matter for many types of industrial systems, such as when you are trying to prevent manufacturing line shutdowns or restore electrical service. Analyzing data close to the device that collected the data can make a difference between averting disaster and a cascading system failure.
- **Conserving network bandwidth:** Offshore oil rigs generate 500 GB of data weekly. Commercial jets generate 10 TB for every 30 minutes of flight. It is not practical to transport vast amounts of data from thousands or hundreds of thousands of edge devices to the cloud. Nor is it necessary because many critical analyses do not require cloud-scale processing and storage.
- **Increasing local efficiency:** Collecting and securing data across a wide geographic area with different environmental conditions may not be useful. The environmental conditions in one area will trigger a local response independent from the conditions of another site hundreds of miles away. Analyzing both areas in the same cloud system may not be necessary for immediate efficiency.

An important design consideration, therefore, is how to design an IoT network to manage this volume of data in an efficient way such that the data can be quickly analyzed and lead to business benefits. The volume of data generated by IoT devices can be so great that it can easily overrun the capabilities of the headend system in the data center or the cloud. For example, it has been observed that a moderately sized smart meter network of 1 million meters will generate close to 1 billion data points each day (including meter reads and other instrumentation data), resulting in 1 TB of data. For an IT organization that is not prepared to contend with this volume of data storage and real-time analysis, this creates a whole new challenge.

The volume of data also introduces questions about bandwidth management. As the massive amount of IoT data begins to funnel into the data center, does the network have the capacity to sustain this volume of traffic? Does the application server have the ability to ingest, store, and analyze the vast quantity of data that is coming in? This is sometimes referred to as the "impedance mismatch" of the data generated by the IoT system and the management application's ability to deal with that data.

As illustrated in Figure 2-14, data management in traditional IT systems is very simple. The endpoints (laptops, printers, IP phones, and so on) communicate over an IP core network to servers in the data center or cloud. Data is generally stored in the data center, and the physical links from access to core are typically high bandwidth, meaning access to IT data is quick.

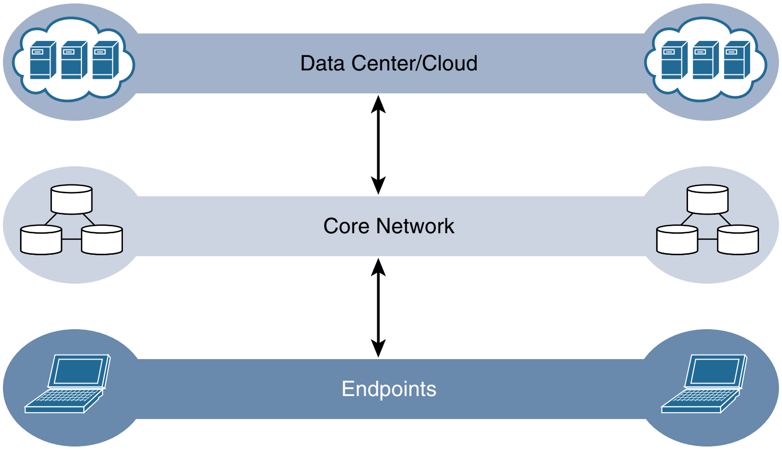


Figure 2-14 *The Traditional IT Cloud Computing Model*

IoT systems function differently. Several data-related problems need to be addressed:

- Bandwidth in last-mile IoT networks is very limited. When dealing with thousands/millions of devices, available bandwidth may be on order of tens of Kbps per device or even less.
- Latency can be very high. Instead of dealing with latency in the milliseconds range, large IoT networks often introduce latency of hundreds to thousands of milliseconds.
- Network backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links. Backhaul links can also be expensive if a per-byte data usage model is necessary.
- The volume of data transmitted over the backhaul can be high, and much of the data may not really be that interesting (such as simple polling messages).
- Big data is getting bigger. The concept of storing and analyzing all sensor data in the cloud is impractical. The sheer volume of data generated makes real-time analysis and response to the data almost impossible.

Fog Computing

The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible. The best-known embodiment of edge services in IoT is fog computing. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways. Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.

Note The concept of fog was first developed by Flavio Bonomi and Rodolfo Milito of Cisco Systems. In the world of IoT, fog gets its name from a relative comparison to computing in the cloud layer. Just as clouds exist in the sky, fog rests near the ground. In the same way, the intention of fog computing is to place resources as close to the ground—that is, the IoT devices—as possible. An interesting side note is that the term "fog" was actually coined by Ginny Nichols, Rodolfo's wife. Although not working directly in IoT, she had an excellent grasp of what her husband was developing and was able to quickly draw the comparison between cloud and edge computing. One day she made the suggestion of simply calling it the "fog layer." The name stuck.

An advantage of this structure is that the fog node allows intelligence gathering (such as analytics) and control from the closest possible point, and in doing so, it allows better performance over constrained networks. In one sense, this introduces a new layer to the traditional IT computing model, one that is often referred to as the "fog layer." Figure 2-15 shows the placement of the fog layer in the IoT Data Management and Compute Stack.

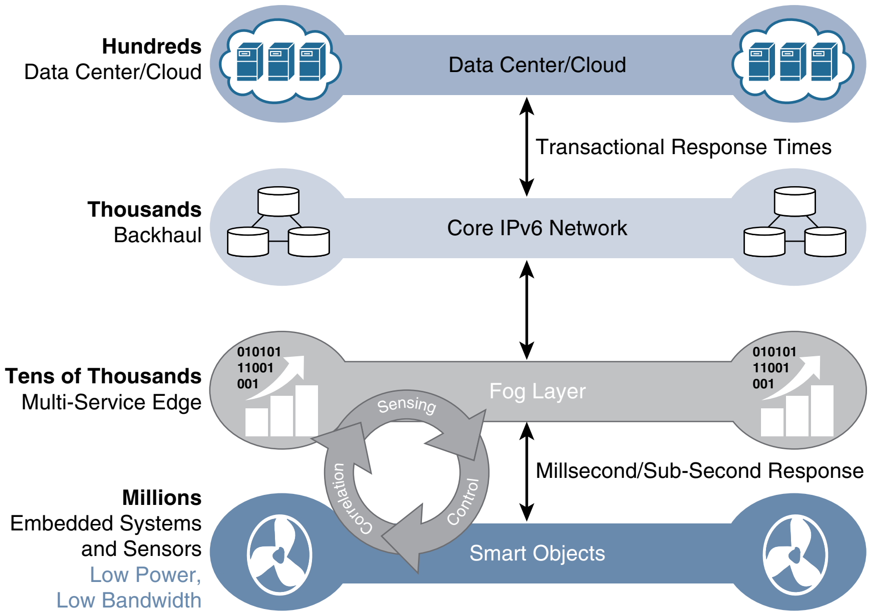


Figure 2-15 *The IoT Data Management and Compute Stack with Fog Computing*

Fog services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible. One significant advantage of this is that the fog node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors. For example, there might be a fog router on an oil derrick that is monitoring all the sensor activity at that location. Because the fog node is able to analyze information from all the sensors on that derrick, it can provide contextual analysis of the messages it is receiving and may decide to send back only the relevant information over the backhaul network to the cloud. In this way, it is performing distributed analytics such that the volume of data sent upstream is greatly reduced and is much more useful to application and analytics servers residing in the cloud.

In addition, having contextual awareness gives fog nodes the ability to react to events in the IoT network much more quickly than in the traditional IT compute model, which would likely incur greater latency and have slower response times. The fog layer thus provides a distributed edge control loop capability, where devices can be monitored, controlled, and analyzed in real time without the need to wait for communication from the central analytics and application servers in the cloud.

The value of this model is clear. For example, tire pressure sensors on a large truck in an open-pit mine might continually report measurements all day long. There may be only minor pressure changes that are well within tolerance limits, making continual reporting to the cloud unnecessary. Is it really useful to continually send such data back to the cloud over a potentially expensive backhaul connection? With a fog node on the truck, it is possible to not only measure the pressure of all tires at once but also combine this data with information coming from other sensors in the engine, hydraulics, and so on. With this approach, the fog node sends alert data upstream only if an actual problem is beginning to occur on the truck that affects operational efficiency.

IoT fog computing enables data to be preprocessed and correlated with other inputs to produce relevant information. This data can then be used as real-time, actionable knowledge by IoT-enabled applications. Longer term, this data can be used to gain a deeper understanding of network behavior and systems for the purpose of developing proactive policies, processes, and responses.

Fog applications are as diverse as the Internet of Things itself. What they have in common is data reduction—monitoring or analyzing real-time data from network-connected things and then initiating an action, such as locking a door, changing equipment settings, applying the brakes on a train, zooming a video camera, opening a valve in response to a pressure reading, creating a bar chart, or sending an alert to a technician to make a preventive repair.

The defining characteristic of fog computing are as follows:

- **Contextual location awareness and low latency:** The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.
- **Geographic distribution:** In sharp contrast to the more centralized cloud, the services and applications targeted by the fog nodes demand widely distributed deployments.
- **Deployment near IoT endpoints:** Fog nodes are typically deployed in the presence of a large number of IoT endpoints. For example, typical metering deployments often see 3000 to 4000 nodes per gateway router, which also functions as the fog computing node.

- **Wireless communication between the fog and the IoT endpoint:** Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.
- **Use for real-time interactions:** Important fog applications involve real-time interactions rather than batch processing. Preprocessing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.

Edge Computing

Fog computing solutions are being adopted by many industries, and efforts to develop distributed applications and analytics tools are being introduced at an accelerating pace. The natural place for a fog node is in the network device that sits closest to the IoT endpoints, and these nodes are typically spread throughout an IoT network. However, in recent years, the concept of IoT computing has been pushed even further to the edge, and in some cases it now resides directly in the sensors and IoT devices.

Note Edge computing is also sometimes called "mist" computing. If clouds exist in the sky, and fog sits near the ground, then mist is what actually sits on the ground. Thus, the concept of mist is to extend fog to the furthest point possible, right into the IoT endpoint device itself.

IoT devices and sensors often have constrained resources, however, as compute capabilities increase. Some new classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions. For example, consider a water sensor on a fire hydrant. While a fog node sitting on an electrical pole in the distribution network may have an excellent view of all the fire hydrants in a local neighborhood, a node on each hydrant would have clear view of a water pressure drop on its own line and would be able to quickly generate an alert of a localized problem. The fog node, on the other hand, would have a wider view and would be able to ascertain whether the problem was more than just localized but was affecting the entire area. Another example is in the use of smart meters. Edge compute–capable meters are able to communicate with each other to share information on small subsets of the electrical distribution grid to monitor localized power quality and consumption, and they can inform a fog node of events that may pertain to only tiny sections of the grid. Models such as these help ensure the highest quality of power delivery to customers.

The Hierarchy of Edge, Fog, and Cloud

It is important to stress that edge or fog computing in no way replaces the cloud. Rather, they complement each other, and many use cases actually require strong cooperation between layers. In the same way that lower courts do not replace the supreme court of a country, edge and fog computing layers simply act as a first line of defense for filtering, analyzing, and otherwise managing data endpoints. This saves the cloud from being queried by each and every node for each event.

This model suggests a hierarchical organization of network, compute, and data storage resources. At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer. As data needs to be sent to the cloud, the latency becomes higher. The advantage of this hierarchy is that a response to events from resources close to the end device is fast and can result in immediate benefits, while still having deeper compute resources available in the cloud when necessary.

It is important to note that the heterogeneity of IoT devices also means a heterogeneity of edge and fog computing resources. While cloud resources are expected to be homogenous, it is fair to expect that in many cases both edge and fog resources will use different operating systems, have different CPU and data storage capabilities, and have different energy consumption profiles. Edge and fog thus require an abstraction layer that allows applications to communicate with one another. The abstraction layer exposes a common set of APIs for monitoring, provisioning, and controlling the physical resources in a standardized way. The abstraction layer also requires a mechanism to support virtualization, with the ability to run multiple operating systems or service containers on physical devices to support multitenancy and application consistency across the IoT system. Definition of a common communications services framework is being addressed by groups such as oneM2M, discussed earlier. Figure 2-16 illustrates the hierarchical nature of edge, fog, and cloud computing across an IoT system.

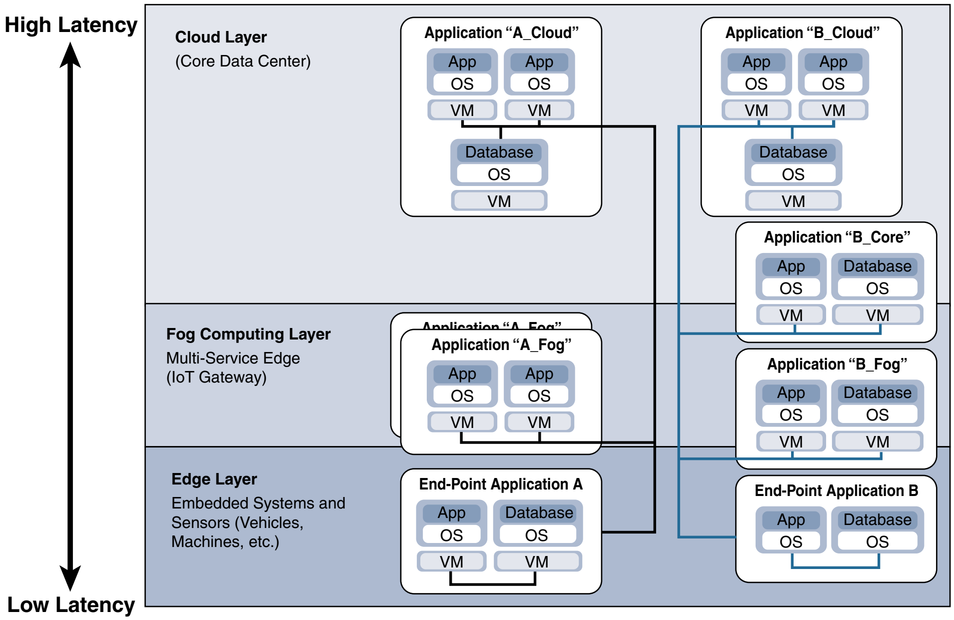


Figure 2-16 *Distributed Compute and Data Management Across an IoT System*

From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices. The fog IoT application then directs different types of data to the optimal place for analysis:

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage. For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of data to the cloud for historical analysis and storage.

In summary, when architecting an IoT network, you should consider the amount of data to be analyzed and the time sensitivity of this data. Understanding these factors will help you decide whether cloud computing is enough or whether edge or fog computing would improve your system efficiency. Fog computing accelerates awareness and response to events by eliminating a round trip to the cloud for analysis. It avoids the need for costly bandwidth additions by offloading gigabytes of network traffic from the core network. It also protects sensitive IoT data by analyzing it inside company walls.

Summary

The requirements of IoT systems are driving new architectures that address the scale, constraints, and data management aspects of IoT. To address these needs, several IoT-specific reference models have arisen, including the oneM2M IoT model and the IoT World Forum's IoT Reference Model. The commonalities between these models are the interaction of IoT devices, the network that connects them, and the applications that manage the endpoints.

This book presents an IoT framework that uses aspects of these various models and applies them to specific industry use cases. This chapter presents a model based on common concepts in these architectures that breaks the IoT layers into a simplified architecture incorporating two parallel stacks: the Core IoT Functional Stack and the IoT Data Management and Compute Stack. This architecture sets the format for the chapters that follow in this book.

The Core IoT Functional Stack has three layers: the IoT sensors and actuators, networking components, and applications and analytics layers. The networking components and applications layers involve several sublayers corresponding to different parts of the overall IoT system.

The IoT Data Management and Compute Stack deals with how and where data is filtered, aggregated, stored, and analyzed. In traditional IT models, this occurs in the cloud or the data center. However, due to the unique requirements of IoT, data management is distributed as close to the edge as possible, including the edge and fog layers.

References

1. ETSI, *oneM2M*, Accessed December 2016, www.etsi.org/about/what-we-do/global-collaboration/onem2m.

2. oneM2M, *oneM2M Technical Specification*, Accessed December 2016, ftp.onem2m.org/Deliverables/20140801_Candidate%20Release/TS-0002-Requirements-V-2014-08.pdf.

This page intentionally left blank