9 Evolution of Embedded Internet

The growth of the Internet is the reason behind a new pervasive paradigm in computing and communications. This novel paradigm, named the Internet of things (IoT), is continuous with the concept of smart environments as well as with the deployment of numerous applications in many fields of future life. It utilizes low-cost information that facilitates interactions between the objects themselves in any place and at any time. From a wireless communication perspective, the IoT paradigm is strongly related to the effective utilization of wireless sensor networks (WSNs) and radio-frequency identification (RFID) systems. IoT will provide a wide range of smart applications and services like remote health care monitoring, intelligent transportation, smart distribution, home automation, systematic recycling, and others. Generally, IoT represents intelligent end-to-end systems that enable smart solutions and, as such, it covers a diverse range of technologies including sensing, communications, networking, computing, information processing, and intelligent control technologies. IoT is composed of a large number of nodes. This poses serious scalability requirements on any solution proposed for the IoT. Proposed solutions must be open and interactive with systems based on different technical solutions. This is a complex task given that most IoT nodes will have scarce capacity in terms of both energy and processing capabilities. Mobile management is another research issue that deserves particular investigation. Things will move and the system should be able to locate them at any moment. The major obstacles are related to the number of mobile nodes, which have an effect on scalability. This is another research issue to be addressed regarding traffic characterization that will traverse the IoT. Such traffic will have different properties from present-day Internet traffic. Most of it will be generated and directed to machines that communicate in a different way from humans. Differences in traffic characteristics, along with the energy constraints and the specific features of the IoT communication environment, will be the stimulus for research activities, modeling, and protocols design at both the network and transport layers.

9.1 INTRODUCTION

The term *Internet of things* was coined more than a decade ago by the Auto-ID Labs [1] (the leading global network of academic research laboratories in the field of networked RFID), where in parallel the concept of "ambient intelligence" and "ubiquitous computing" was developed. Since then, there have been considerable developments in both academia and industry, in the United States as well as in Europe and Asia. The developments have primarily been dedicated to applying RFID technology to the logistics value chain. The first trials to establish IoT-like applications for end users have been set up in "future stores" in Germany, Switzerland, and Japan.

The IoT can be considered as a convergence among a number of heterogeneous disciplines. This multidisciplinary domain covers a large number of topics from technical issues (routing protocols, semantic queries), to a mix of technical and societal issues (security, privacy, usability) including social and business themes. Pleasant user experiences are planned in the workplace and public areas as well as in the home environment by embedding computational intelligence into the nearby environment and simplifying human interactions with everyday service.

Overcoming the technical challenges and socioeconomic barriers of wide-scale IoT deployment requires a practical evaluation of corresponding solutions using interdisciplinary, multitechnology, large-scale, and realistic test beds. The test beds aim to design and deploy experimental environments that will allow [2]:

- The technical evaluation of IoT solutions under realistic conditions
- The assessment of the social acceptance of new IoT solutions
- The quantification of service usability and performance with end users in the loop

RFID is one of the key technologies because it not only permits a digital code to be associated with an object in a wireless modality but it also allows its physical status to be captured. RFID tags may be equipped with a large variety of sensors according to different modalities of integration exploiting a broad range of possible functionalities and costs. Active RFID tags use independent power supplies and a microcontroller, and dedicated electronics ensure long operating ranges. Also, it is possible to support high data rates and the greatest versatility in sensor interconnection. The main drawbacks of this solution are its high cost, limited lifetime, and large weight as well as size [3].

Sensor widespread deployment represents significant financial investment and technical achievement. The data they deliver is capable of supporting an almost unlimited set of high-value proposition applications. However, the main problem hampering success is that these sensors are locked into unimodal closed systems. Unlocking valuable sensor data from closed systems is a great task. Access to sensors should be opened such that their data and services can be integrated with data and services available in other information systems [4].

When it comes to wireless sensor technology, a variety of WSN approaches such as ZigBee, and other proprietary solutions have been proposed. If a trillion things are connected through a single open standard interface such as IP, they become transparent as general hosts and servers supporting seamless connectivity, unique addressability, and rich applicability. Because of that, IP-based WSNs have recently gained worldwide attention. The research in this field focuses on how to build a fundamental architecture that enables the IP to be used in a WSN space [5].

Nowadays, there is a clear need to develop a reference architectural model that will allow interoperability among different systems. With respect to the technological roadblocks, there is a need for action in three areas [6]:

a. An architectural reference model for the interoperability of IoT systems, outlining principles and guidelines for the design of its protocols, interfaces, and algorithms
b. Mechanisms for the efficient integration of the architecture into the service layer of a future Internet networking infrastructure
c. A novel resolution infrastructure, allowing scalable lookup and discovery of IoT resources, entities of the real world, and their associations

An emerging category of edge devices that will result in the evolution of the IoT are consumer-centric mobile sensing and computer devices connected to the Internet. They are equipped with various sensing facilities and wireless capabilities that allow producing data and uploading the data to the Internet [7]. Different from the IoT objects that lack computing capabilities, mobile devices have a variety of sensing, computing, and communication facilities. They can either serve as a bridge to other everyday objects, or generate information about the environment themselves. Based on the type of monitored phenomena, these applications can be classified into two categories: personal and community sensing. In personal sensing applications, the phenomena pertains to an individual (e.g., the monitoring of movement patterns of an individual, for personal record-keeping, or health care purposes). Community sensing is also known as participatory or opportunistic sensing. Participatory sensing requires the active involvement of individuals to contribute sensor data related to a large-scale phenomenon. Opportunistic sensing is more autonomous and user involvement is minimal. It means continuous sampling without the explicit action of the user [8]. Taking into account that community sensing spans a wide spectrum of user involvement, with participatory sensing and opportunistic sensing at two ends, the term mobile crowdsensing (MCS), which refers to a broad range of community sensing paradigms, was coined.

Machine-to-machine (M2M) is a promising technology for the development of IoT communications platforms with high potential to enable a wide range of applications in different domains. Providing suitable answers to this issue streaming from IoT platform design requires middleware level solutions to enable seamless interoperability between M2M-based applications and existing Internet-based services. Because of the growing demand for M2M-based services, various standardization bodies, such as 3GPP, the Open Mobile Alliance (OMA), IEEE, and ETSI have promoted various standardization activities on M2M [9]. In brief, M2M enables highly scalable direct communications among wireless heterogeneous terminals called M2M devices, and between M2M devices and control application servers (M2M servers) [10]. The ultimate goal of these standardization activities is to leverage widespread integration of M2M devices without any existing service.

Semantic Web of things [11] is a service infrastructure that makes the development and use of semantic applications involving Internet connected sensors almost as easy as searching and reading a web page. The IoT can be considered as a third ring of the present-day Internet, which already consists of a stable core (routers, servers) and a quickly growing fringe (personal computers and smartphones) [12]. This model is shown in Figure 9.1. As new embedded applications such as smart metering, building and industrial automation, personal sensing, and transportation become IP enabled, they add an entirely new dimension to the Internet with existing possibilities and challenges.

Networking alone does not make the Internet of practical use today. Applications depend on the web architecture using hypertext transfer protocol (HTTP) as well as extensible markup language (XML) to create web services. The interaction model of M2M applications differ greatly from how web services are used today. For example, multicast is commonly required in automation, whereas flow control is needed across the entire M2M network.

Finally, nanotechnology is enabling the development of devices in a scale ranging from one to a few hundred nanometers. At this scale, a nanomachine is defined as the most basic function unit, integrated by nanocomponents, and is able to perform simple tasks like sensing or actuation [13]. Coordination and information sharing among several nanomachines expands the potential applications of individual devices both in terms of

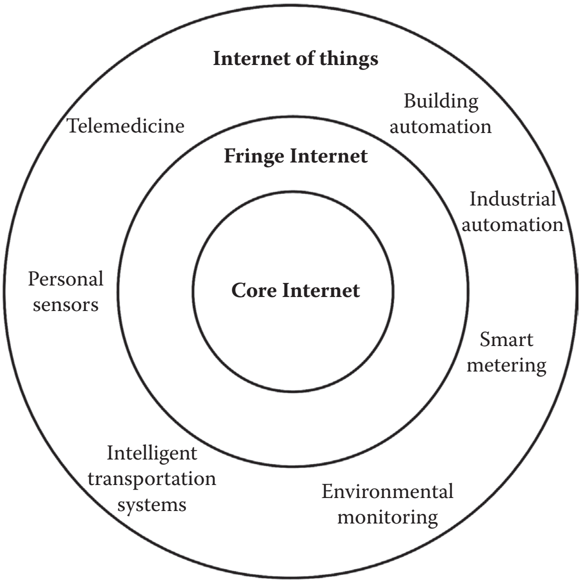


FIGURE 9.1 Ring model for the IoT.

complexity and range of operation [14,15]. The resulting nanonetwork will be able to cover larger areas, to reach new locations in a noninvasive way, and to perform additional in-network processing. The interconnection of nanoscale devices with classic networks and the Internet defines a new networking paradigm, which is referred to as the Internet of nanothings (IoNT) [13]. Two main alternatives for communication in the nanoscale can be envisioned: molecular communication and nanoelectromagnetic communication. Molecular communication is defined as the transmission and reception of electromagnetic (EM) waves from components based on novel nanomaterials. The unique properties observed in these materials will decide the specific bandwidth for electromagnetic emission, the time lag of the emission, and the magnitude of the emitted power for a given input energy.

9.2 MOBILE CROWDSENSING

The term "sensing" is considered ranging from the acquisition of an elementary status of an object (presence or absence with a given, region-localization) to multidimensional description to parameters of a thing with respect to the research environment and other things. Here, the term "thing" is capitalized to highlight the fact that the used tagging of an object is augmented with the physical interaction between the objects and the RFID tag itself to produce more information content [16]. The sensing modalities may fall into the stationary and nonstationary classes. Stationary sensing occurs when the measurement is performed in controlled conditions such as when the mutual position between the reader and the thing remains unchanged during the entire phenomenon to monitor, or when the same position can be replicated exactly in successive readings. On the other hand, we have the case of nonstationary sensing in which interrogation is performed at different times or the object is moving. The eventual change of the reader tag position as well as the change in the environment can be a further unknown of the sensing problem, making data retrieval more difficult. Generally speaking, additional independent data or functionalities are required to manage the sensing [3]. In what follows, we survey existing applications, identify characteristics of MCS, and discuss resource limitations, security and data integration, as well as architecture of MCS applications.

An emerging category of devices at the edge of the Internet are consumer-centric mobile sensing and computing devices such as smartphones, multimedia players, and in-vehicle sensors. These devices will fuel the evaluation of the IoT. In MCS, individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest [7].

MCS applications provide a basis for illustrating various research challenges. They can be classified based on the type of phenomenon being measured or mapped as environmental, infrastructural, or social. In environmental MCS applications, the phenomena are those of the natural environment (measuring various air pollution levels in a city, water levels in rivers, etc.). Applications enable the mapping of various large-scale environmental phenomena by involving the common person. Infrastructural applications involve the measurement of large-scale phenomena related to the public infrastructure (real-time traffic congestion and road conditions, etc.). The third category is social applications, in which individuals share sensed information among themselves. As an example, individuals share their exercise data and compare their exercise levels with the community to improve daily routines. The functioning of typical MCS applications is presented in Figure 9.2.

Raw sensor data are collected on devices and processed by local analytical algorithms to produce adequate data for applications. These data may then be modified to preserve privacy and sent to the back end for aggregation and mining.

9.2.1 MCS Applications Architecture

MCS application has two specific components: one on the device for sensor data collection and propagation, and the second in the back

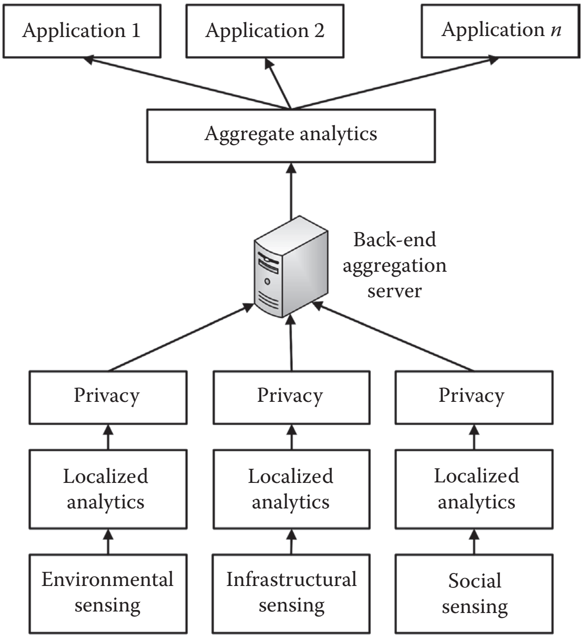


FIGURE 9.2 Function of typical MCS applications.

end or cloud for the analysis of the sensor data to drive the application [7]. This architecture is presented in Figure 9.3. Each application is built from the ground-up and independent from each other. There is no common component even though each application faces a number of common challenges in data collection, resource allocation, and

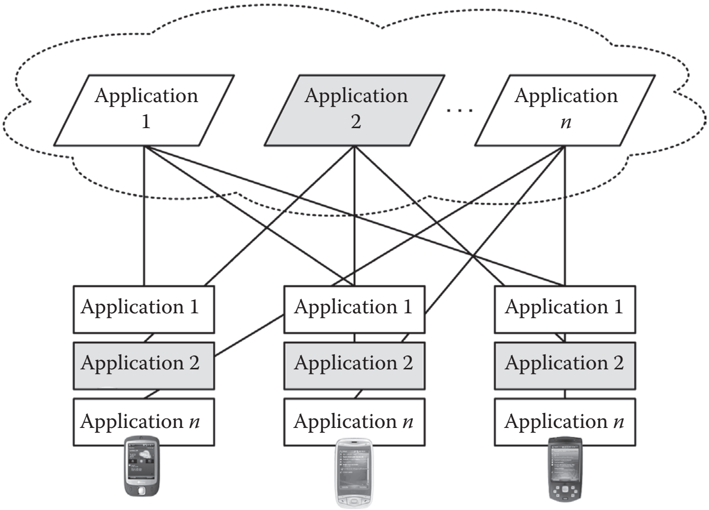


FIGURE 9.3 Architecture of existing MCS applications.

energy conservation. The presented architecture puts some limits to the development and deployment of MCS applications. First of all, the developer has to address challenges in energy, privacy, and data quality in an ad hoc manner. Second, he or she may need to develop different variants of local analytics if that person wants to run the application on heterogeneous devices. This approach is inefficient because there is a high likelihood of duplicating sensing and processing across multiple applications. For example, traffic sensing, air, and noise pollution all require location information, but these applications would each do its own sampling without reusing the same data samples. Also, there is no collaboration or coordination across devices. Namely, devices may not all be needed, especially when the device population is dense. Finally, the current architecture is not scalable because only a small number of applications can be installed on each device due to operating system limitations or user ability to keep track of a large number of applications. The data gathered from societal-scale sensing may overload network and back end server capacities, thus making the current architecture nonscalable.

Unifying architecture could address the current limitations of how MCS applications are developed and deployed. It will satisfy the common needs for different applications. It should allow application developers to specify their data needs in a high-level language. It should identify common data needs across applications to avoid duplicate sensing and processing activities on devices. Next, it should automatically identify the set of devices that can properly provide the desired data, and produce instructions to configure the sensing activities. For a given dynamic change, it should adapt the set of chosen devices and sensing instructions to ensure the desired data quality. Also, a layer that can shield the differences in physical sensor access application APIs and provide the same API upward is necessary. This means that it is possible to reduce some local analytics across different device platforms, assuming these platforms all support a common programming language.

9.2.2 Characteristics of MCS

MCS has a number of unique characteristics that bring both new opportunities and problems. Today's mobile devices have significantly more computing, communication, and storage resources than traditional sensors, and they are usually equipped with multimodality sensing capabilities. These will enable many applications that require resources and sensing modalities. Using these devices, we could potentially build large-scale sensing applications efficiently from the point of cost and time.

TABLE 9.1
Main Characteristics of MCS

MCS Issues	Description	Examples
Cost	There are already millions of user's devices that can collect data, so there is no need to install specialized sensors	Sensors on mobile phones can provide more information about traffic conditions than some specialized sensors [17]
Localized data processing	Depending on the nature of the raw data and the needs of applications, the physical readings from sensors may not be suitable for direct application. Often, raw data processing on the device is needed, producing intermediate results, which are sent to the aggregation server for further processing and application	In a pothole detection application, a local analytic computes spikes from three-axis acceleration sensor data to determine potential potholes [18]
Resource limitations	With different quality and resource consumption trade-offs, different types of data can be used for the same purpose. One of the challenges is leveraging these differences to improve the quality while minimizing resource consumption	Instead of GPS, location data can be provided using Wi-Fi and mobile systems [19], but with decreasing levels of accuracy. This approach represents trade-offs in data quality and accuracy for energy
Security	MCS applications potentially collect sensitive data for individuals. Because of that, privacy is one of the most sensitive subjects for IoT security [20]. The data availability expression has created entities that profile and truck users without their consent	GPS sensor readings can be utilized to inter private information about the individual, such as the routes they take during their daily communities and their home and work locations [21]
Data analyzing	MCS applications rely on analyzing the data from a collection of mobile devices, identifying spatiotemporal patterns. The challenge in identifying patterns from large amounts of data is usually application-specific and involves certain data mining algorithms	Data mining algorithms can provide reports to help prioritize and schedule the repair resources for MCS application in public work maintenance. Such algorithms take as input continuous data streams and identify patterns without the need to first store the data [7]

In traditional sensor networks, the population and the data they can generate are mostly known a priori. Because of that, controlling the data quality is much easier. In MCS, the population of mobile devices, the type of sensor data each can produce, and the quality in terms of accuracy, latency, and confidence can change all the time due to device mobility, variations in their energy levels and communication channels, and device user's preferences [7]. The main characteristics of MCS including description and examples are systematically analyzed in Table 9.1.

9.3 PERSPECTIVE PROTOCOL STACK FOR WIRELESS EMBEDDED INTERNET

The widespread sensor deployment represents significant investment and technical achievements. A crucial problem hampering success is that sensors are typically locked into unimodal closed systems. To solve this problem, sensor connections to the Internet and publishing outputs in well understood machine-processible formats on the web are indispensible.

Internet connectivity requires not only network level integration but also application level integration to enable structured access to sensor data. To enable sensor automatic reasoning, these sensors, their outputs, and their embedding into the real world must be described in a machine-readable format that is compatible with the data formats used to describe existing world knowledge in the Web. Not only must the syntax and semantics of such a description be defined but efficient mechanisms to annotate newly deployed sensors with appropriate descriptions are also required [4]. Finally, the users wish to search for real-world entities by their current state. Such search requests refer not only to the output of sensors but also to further machine-readable information that is available elsewhere in the web. The search engine needs to integrate these different data sources in a seamless manner.

Integrating resource-constrained sensors into the Internet is difficult because traditionally deployed protocols such as HTTP, TCP, or even IP are too complex and resource-demanding. To achieve integration, simple alternatives are required that can easily be converted from/to Internet protocols. Enormous progress has been made at the IETF over the past few years in specifying new protocols that connect smart objects to IP networks. The IETF has formed three WGs that define an adaptation layer (IPv6 in low-power WPAN—6LoWPAN) [22], routing over low-power and lossy networks (ROLL) [23], and a resource-oriented application protocol (Constrained Restful Environments—CoRE) [24]. The stack for wireless embedded Internet based on protocols proposed by corresponding IETF WGs is presented in Figure 9.4.

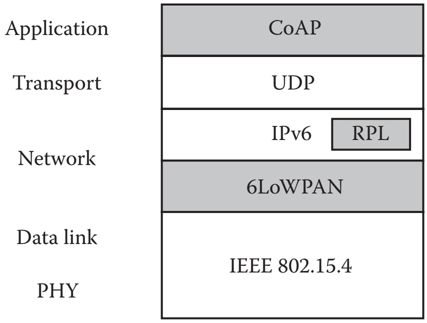


FIGURE 9.4 Perspective protocol stack for wireless embedded Internet.

9.3.1 Adaptation Layer

The adoption of IP by wireless embedded devices is challenging due to several reasons [25]:

- Battery-powered wireless devices require low duty cycles, whereas IP is based on always connected devices
- Multicast is not supported natively in IEEE 802.15.4, but it is essential in many IPv6 operations
- Sometimes, it is difficult to route traffic in multihop wireless mesh networks to achieve the required coverage and cost efficiency
- Low-power wireless networks have low bandwidth (20–250 Kb/s) and frame size (IEEE802.15.4 packets are rather small, 127 bytes maximum at the physical layer, minus MAC/security and adaptation layer overhead). On the other hand, the minimum datagram size that all hosts must be prepared to accept for IPv6 is 1280 bytes. IPv6 requires that every link in the Internet has a maximum transmission unit (MTU) of 1280 bytes or greater. On any link that cannot convey a 1280-byte packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6
- Standard protocols do not perform well in LoWPAN. For example, TCP performs very poorly in wireless networks due to its inability to distinguish between packet losses due to congestion and channel error.

Because IPv6 represents the backbone of NGN [26], the 6LoWPAN WG was chartered to standardize necessary adaptations of this network protocol for systems that use the IEEE 802.15.4 PHY layer, and has defined how to carry IP datagrams over IEEE 802.15.4 links and perform necessary configuration functions to form and maintain an IPv6 subnet. 6LoWPAN represents a lightweight IPv6 adaptation layer allowing sensors to exchange IP packets [27,28]. Core protocols for 6LoWPAN architecture have already been specified and some commercial products have been launched that implement this protocol suite.

9.3.2 Routing over Low-Power and Lossy Networks

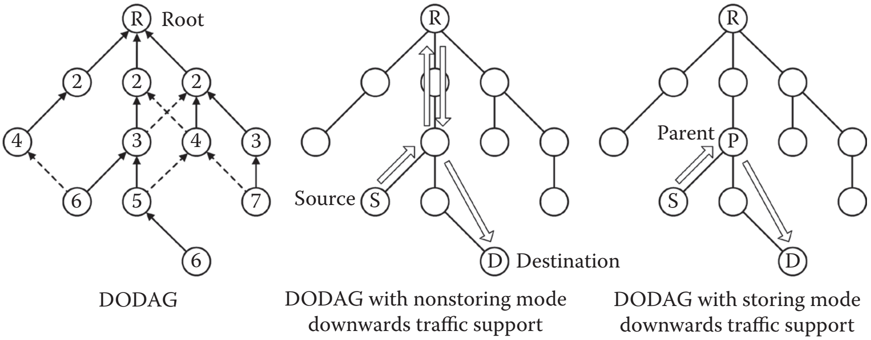
The IETF has significant experience in IP routing and it has specified a number of routing protocols over the past two decades (RIP, OSPF, etc.). On the other hand, routing in networks made of smart objects has unique characteristics. These characteristics led to the formation of a new WG called ROLL, whose objective is to specify a routing protocol for low-power and lossy networks (LLNs), known as RPL [29]. LLNs are formed by smart objects with limited processing power, memory, and energy. Unlike the MANET routing protocols, which perform well for ad hoc networks, RPL is optimized for upstream and downstream routing (to/from a root node), a paradigm appropriate for networks connected to the Internet. This routing protocol is essential for the deployment of the IoT because it enables traffic forwarding between low-power devices and the Internet. It has been designed assuming that the LLN scan comprises up to thousands of nodes interconnected by unstable links. Furthermore, RPL has been designed to operate over a variety of link layers such as IEEE 802.15.4, and it is a typical distance vector IPv6 routing protocol.

A directed acyclic graph (DAG) [29] is a directed graph having the property that all edges are oriented in such a way that no cycles exist. All edges are contained in paths oriented toward and terminating at one or more root nodes (traditionally called sinks in WSNs). RPL routes are optimized for traffic to or from one or more roots (sinks). As a result, RPL uses the DAG topology and is partitioned into one or more destination-oriented DAGs (DODAGs), one DODAG per sink. RPL specifies how to build the DODAG using an objective function. The objective function computes the optimal path according to certain routing metrics and constraints. In this way, DODAGs with different characteristics can be formed. For example, different DODAGs are constructed with the objective to (1) find the best path in terms of link throughput while avoiding battery-operated nodes, or (2) find the optimal path in terms of latency while avoiding nonencrypted links. There can be several objective functions operating at the same node depending on the different path requirements of a given traffic. In this way, it is possible to have multiple DODAGs active at the same time to carry traffic with different requirements.

Example 9.1

RPL specifies local and global repair mechanisms for recomputing routes when an inconsistency is detected or based on administrative decisions [30]. Local repair means detaching a node's sub-DODAG by increasing its rank value. Once a root initiates a global repair event, all the nodes in the DODAG recompute their rank values and reconfigure their parent sets. An example topology of a RPL network together with nonstoring and storing modes is presented in Figure 9.5.

Solid arrows represent each node's preferred parent (determined from the node's neighbors and their rank values) whereas dotted arrows point to the other nodes in the parent set. To support routing to various destinations within the DODAG, which is the root, RPL uses the destination advertisement object (DAO) message. RPL supports scenarios in which in-network nodes do not have enough memory to store routes to all possible destinations. In this case, the DAO messages, which contain information on the desired parent set of a destination node, are propagated up the DODAG until they reach the root. The root gathers DAOs from all nodes in the DODAG, and uses them to construct "down" routes to various destinations. Data to these advertised destinations is forwarded along a DODAG until it reaches the root, which then attaches a source routing header and sends it back down the DAG. Alternatively, nodes in the DODAG may store next-hops to downstream destinations. However, a key design simplification was not supporting "mixed-mode operation" in which storing and nonstoring nodes coexist because this



DODAG

DODAG with nonstoring mode downwards traffic support

DODAG with storing mode downwards traffic support

FIGURE 9.5 Example of RPL nodes that form a DAG rooted at a destination node supporting multipoint-to-point traffic.

was still considered a research issue; thus, all nodes in a DODAG must either store or not store routes.

9.3.3 Application Protocol

In 2010, the IETF started a new WG, called CoRE [24], with the aim of extending the Web architecture to even the most constrained networks and embedded devices. Today's Web protocols work well between servers and clients running on PCs and handheld devices. However, constrained LLNs often mean high packet loss (5%–10%), frequent topology changes, low throughput (10–20 Kb/s), and useful payload sizes that are often less than 100 bytes. Embedded devices typically depend on cheap embedded microcontrollers, with processors running at several megahertz and limited memory. In addition, the interaction patterns in M2M applications are different, often requiring multicast support, asynchronous transactions, and push rather than pull. The CoRE WG has been chartered to develop a new Web transfer protocol and appropriate security setups for these M2M applications over constrained networks and nodes [31].

The WG is currently completing work on the constrained application protocol (CoAP) [32]. It provides a highlight alternative to HTTP using a binary representation and a subset of HTTP's methods (GET, POST, etc.). In addition, CoAP provides some transport reliability using acknowledgments and retransmissions. For seamless integration, reverse proxies may convert 6LoWPAN to IPv6 and UDP/CoAP to TCP/HTTP so that sensor data can be accessed using these omnipresent protocols. The integration of sensors into the Internet using CoAP/HTTP already enables many applications in which developers query and process data provided by a known set of sensors. A machine-understandable description of sensors and the data they generate are required. Semantic Web technologies fulfill this requirement as they enable machines to understand, process, and interlink data using structured descriptions of service [33]. Linked open data as the framework makes this integration both immediate and meaningful through the inclusion of semantic links into a resource's machine-readable description.

Example 9.2

The use of UDP as transport protocol and the reduction of the packet header size significantly decrease power consumption in IoT. To evaluate the CoAP performance improvement compared with the HTTP, a simple experiment can be performed.

TABLE 9.2
HTTP and CoAP Comparison in Terms of Power Consumption

Protocol	Bytes/Transaction	Power Consumption (mW)
HTTP	1451	1.333
CoAP	154	0.744

A series of web service requests, first, between a CoAP client/server system and then between an HTTP client/server system are generated by Colitti et al. [34]. The CoAP system is based on a previously described protocol stack. Table 9.2 illustrates the results of the comparison between CoAP and HTTP in terms of bytes transferred per transaction and power consumption. It should be noted that the results have been taken in steady state conditions.

An HTTP transaction has a number of bytes nearly 10 times larger than the CoAP transaction. This is a consequence of the significant header compression executed in CoAP. In fact, CoAP uses a short fixed-length compact binary header of 4 bytes and a typical request has a total header of about 10 to 20 bytes. After being encapsulated in the UDP, 6LoWPAN, and MAC layer headers, the CoAP packet can be transferred into a single MAC frame, which has a size of 127 bytes. It is straightforward that the higher number of bytes transferred in an HTTP transaction implies a more intensive activity of the transceiver and CPU and, consequently, higher power consumption. The battery lifetime is unrealistically short in both cases as a consequence of the high number of client requests generated during the experiment. It is worth underlining that the results presented in this example do not exhaustively compare the two protocols. The simple experiment presented is only intended to illustrate how the UDP binding and the header compression introduced in CoAP improve the power consumption of IoT.

9.4 WSNs AND IoT

There have been many research projects in the field of IP-based WSN approach to the IoT from a dedicated IP stack for low–processing-power microprocessors to real deployments. Since the development of a micro-IP (μIP), as an open source TCP/IP stack capable of being used with tiny microcontrollers for smart sensors, a few approaches were carried out on top of TinyOS [35] and ContikiOS [36].

There are a wide range of technologies that will be involved in building the IoT. The enhancement of the communications network infrastructure, through heterogeneous technologies, is essential as well as the adoption of IPv6 to provide a unique address to each thing connected to the network. The technologies that allow the location and identification of physical objects will also be basic in this context. WSNs are able to provide an autonomous and intelligent connection between the physical and virtual worlds. Focusing on this type of network, a particular important challenge is the creation of a secure E2E channel between remote entities [37]. Therefore, it is necessary to allow the elements of a WSN to connect with other entities through the Internet. An increase in research efforts has lead to maturity in this field, yet it seems that there are some gaps that need to be filled.

As for protocols, they should be carefully designed in terms of the tradeoff between interlayer, independence, and optimization. Some protocols may be vertically located through layers for optimization. Other protocols may stay beyond the adoption layer for protocol independence. The sensor networks for an all-IP world (SNAIL) [5] approach to the IoT protocols were designed to comply with IETF RFC 4944 [27]. The SNAIL adaptation layer includes all the necessary frame formats and operations, and is fully compliant with standards related to header compression, addressing, fragmentation and reassembly, and so on. To complete the IPv6 adaptation, it employs important protocols that are not specified in the standard. To obtain an IPv6 address in the start-up process, a bootstrapping network protocol is proposed. In this protocol, autoconfiguration is completed by combining a network prefix. This prefix is obtained from a neighbor discovery (ND) message from a response node with an interface identifier from an association process between two IEEE 802.15.4 nodes. During the bootstrapping process, a joining node registers its information, which includes an IP address, 64-bit extended unique identifier (EUI-64), type of routing protocol, sensors, as well as service for further network management by the gateway. Also, the registration information is used to advertise the gateway’s liveness and the network prefix.

To serve rich Web content and to reduce the overhead of sensor nodes with no sacrifice of interoperability, the distributed resource-based simple web service (DRESS-WS) was designed [5]. It uses HTTP over TCP and distributes traffic to presentation servers and sensor nodes. As for web content, there exists two types of web content: real data and presentation templates. The real data are dynamic, whereas the templates are static and shareable accounting for the high traffic intensity. Only real data are served by nodes that host a web server, whereas the templates are served by presentation servers.

Example 9.3

Consider the DRESS-WS architecture shown in Figure 9.6. It consists of four components: distributed domain name system (DDNS) server, presentation server, gateway, and sensor node. The DDNS server manages the IP address information, which corresponds to a domain name to handle mobility of nodes. The presentation server serves templates, including multimedia and application codes to process the sensed data. The codes enable clients to control the periodic reporting of the sensed data. Each presentation server is assigned to sensor nodes, and each sensor node can have many distributed servers, depending on the manufacture of the node or the type of application in the node. The gateway performs HTTP/TCP/IP header compression and decompression to cover a limited network bandwidth with interoperability. The sensor nodes operate as a web server and serve the sensed data through Web services.

In DRESS-WS, when a client enters the domain name of a specific node in a common web browser, the browser is connecting to the DDNS server to get the corresponding IP address. While it requests web pages by using the address, the gateway compresses the headers and forwards the compressed packet to the sensor node. For the first access, the node replies with an HTTP redirect toward its presentation server (otherwise, it responds with the requested data). If the client receives this instruction,

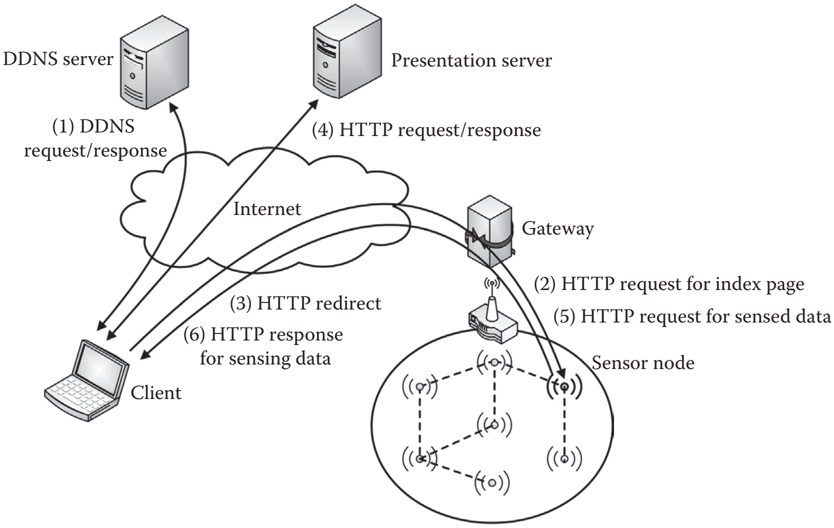

DDNS server
Presentation server
(1) DDNS request/response
(4) HTTP request/response
Internet
Gateway
(2) HTTP request for index page
(5) HTTP request for sensed data
(3) HTTP redirect
(6) HTTP response for sensing data
Sensor node
Client

FIGURE 9.6 DRESS-WS architecture.

the browser automatically requests the template files. After the templates are downloaded, the application codes in the templates work to receive and process the sensed data. Because of that, this architecture enables a resource-constrained device to serve its information with a rich user interface and less overhead. Together with web enablement (previously analyzed), the main issues for WSN and IoT interaction are related to mobility managements, time synchronization, and security concepts.

9.4.1 Mobility Management in Embedded Internet

Mobility management is one of the most important research issues in 6LoWPAN based networks. Because typical mobility protocols are generally targeted for global mobility, they introduce significant network overhead in terms of increasing delay, packet loss, and signaling when mobile nodes change their point of attachment very frequently within small geographical areas. Having that fact in mind, methods for reducing handover delay are therefore essential for the IoT. The mobility protocol should be supported to a lightweight fashion because a thing's mobility behavior directly inherits the characteristics of portable devices.

SNAIL architecture uses a novel mobility management protocol called MARIO, which stands for mobility management protocol to support intra-PAN and inter-PAN handover with route optimization for 6LoWPAN [38]. The design of MARIO is based on MIPv6 and a fast and seamless handover scheme.

In the case of the inter-MARIO handover procedure illustrated in Figure 9.7, when a partner node detects MN's movement, it sends a preconfiguration message with the MN information. The information is stored by foreign agents (FAs) due to resource limitations. The partner node also gives the MN information, such as channel information, about the neighbor PAN. When orphaned, the MN can use the channel information to selectively scan a channel. When the MN associates with the new PAN, the FA performs a surrogate binding update simultaneously with the MN's IP operations, as in the case of the care-of address (CoA) generation. With this operation, the home agent (HA) creates a binding for the home-of address (HoA) of the MN to the FA. After the process of joining the new PAN is completed, the MN sends a binding update to the FA with a binding for the MN's HoA to the MN's CoA. This binding operation brings the MN to the end of the handover procedure. Because of the handover preconfiguration and surrogate binding update, MARIO can reduce the channel scan delay, the layer 2 association delay, and the binding message exchange delay. MARIO also provides an additional benefit to the solution of the route optimization problem.

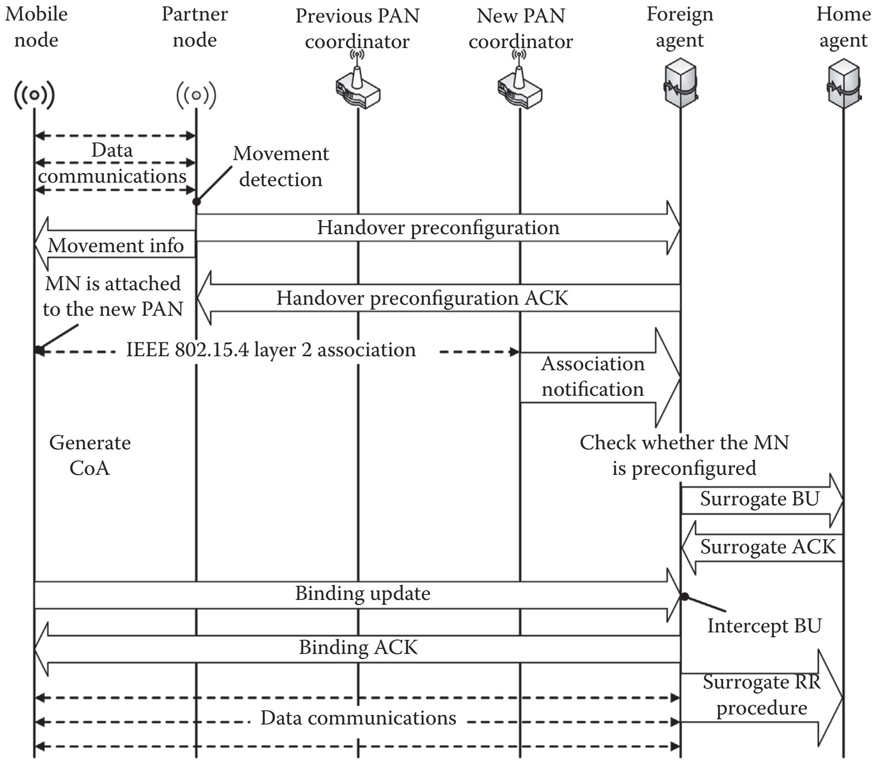


FIGURE 9.7 Inter-MARIO handover procedure.

This procedure does not provide an optimal route from the corresponding node to the MN because packets addressed to the MN may be delivered to the HA, and then it is forwarded to the new location of MN. To solve this problem, which is also known as triangle routing, the MIPv6 standard proposes a route optimization method, which is the return routability (RR) procedure [39].

9.4.2 Global Time Synchronization

Network time protocol (NTP) [40] and simple NTP (SNTP) [41] are the most widely used time protocols on the Internet. They are application-level time protocols based on E2E communication. Also, they are unlikely to be used for resource-limited multihop-based networks because E2E communication for time correction causes substantial overhead. This overhead is due to the necessity for a number of periodic control messages to synchronize from each node.

Several time synchronization protocols and schemes are proposed for WSNs in the open literature [42]. These protocols only focus on clock

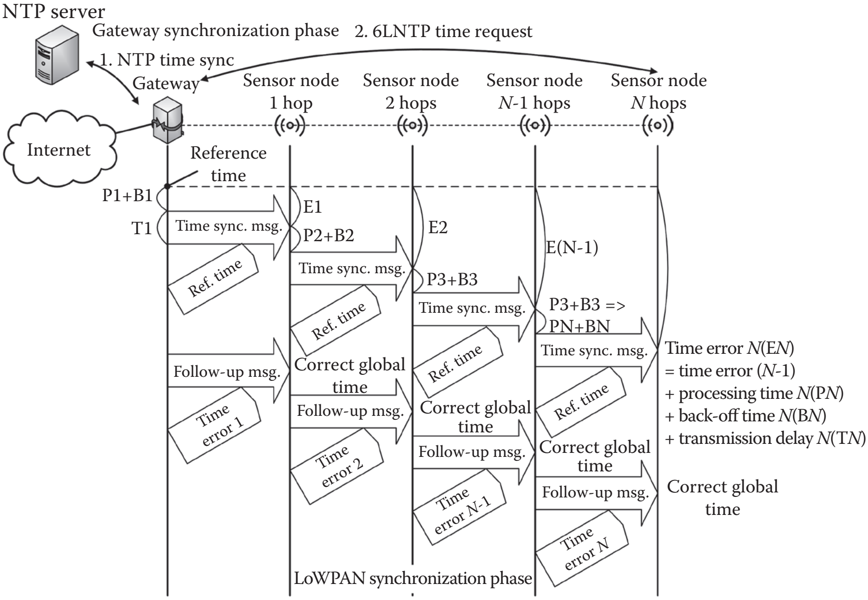


FIGURE 9.8 Multihop time synchronization protocol for IoT.

synchronization, not global time synchronization. This limitation is caused by protocols that are tightly coupled with energy conservation in the MAC layer. Thus, the synchronization process is often executed only between neighbors.

To ensure the consistency and precision of information, the IoT requires globally synchronized time. A multihop time synchronization protocol called 6LoWPAN NTP (6LNTP) was proposed by Hong et al. [5]. This protocol has two phases (Figure 9.8):

1. Gateway synchronization phase, which involves the use of the NTP, and
2. LoWPAN synchronization phase, which adjusts the time to the reference time from the gateway

In the first phase, the 6LoWPAN gateway corrects the time by exchanging request and response packets with an NTP server. Whereas in the second phase, the nodes synchronize their time with the gateway that synchronized with an NTP server.

9.4.3 Security Issues in Embedded Internet

For WSN to become a part of the IoT, it is necessary to consider from the adaptation of existing network standards to the creation of interoperable protocols and the development of supporting mechanisms for composable services. One of the challenges is security, mainly because it is not possible to directly apply existing Internet-centric security mechanisms due to the nature of WSN. The relevant security challenges are related to the integration of WSN within the IoT. Although these challenges are tightly related to WSN, they can also be applicable to other relevant technologies of the IoT (embedded systems, MCS, etc.). Even if a WSN itself is protected with its own security mechanisms (e.g., using the link-layer security), the public nature of the Internet requires the existence of secure communication protocols for protecting the communications between two peers.

Sensor nodes can make use of the 6LoWPAN protocol to interact with IPv6-based networks. They are powerful enough to implement symmetric key cryptography standards such as AES-128 [43]. Due to the power constraints and limited computational capabilities of the nodes, there is currently no explicit support for the IPsec protocol suite in 6LoWPAN. As a consequence, it is necessary to study how other mechanisms can be used to create an E2E secure channel.

The creation of secure channels is just one of the steps in the creation of a securely integrated WSN [37]. To avoid unauthorized users from accessing the functions of the WSN, authentication and authorization mechanisms must be developed. Also, we need to create suitable and scalable identification mechanisms that can provide "unique identifiers" and "virtual identifiers" to all the different network elements. Finally, we have to take into account the survivability problem of IP-based WSN.

Other important challenges in this particular field are the integration of security mechanisms and data privacy [44]. The security of the IoT from a global perspective, regarding information, must be considered. Even if different wireless technologies are secure by their own, their integration will surely generate new security requirements that must be fulfilled. Also, it is necessary to analyze how the security mechanisms that protect one single technology will be able to coexist and interact with each other.

9.5 M2M COMMUNICATIONS AND EMBEDDED INTERNET

M2M communications is a new technology that provides the networks the ability to bring smart services to users. It is viewed as one of the next frontiers in wireless communications [45]. Different from the traditional human to human communications for which the current wireless networks are designed and optimized, the M2M concept is seen as a form of data communications between entities that do not necessarily need any form of human implication. It is different from current communication models in the sense that it involves new or different market scenarios, low cost and low effort, a potentially very large number of communicating terminals, and small and infrequent traffic transmission per terminal [46]. As for the industry, it has already been working on providing M2M communications and smart services offerings across a wide variety of market segments, including health care, manufacturing, utilities, distribution, and consumer products.

Because this concept brings very different requirements, and the number of communication devices may increase quickly, industry members have proposed enhanced wireless access networks for M2M communications. The topic of M2M communications attracted the attention of standardization bodies such as 3GPP LTE, whose objectives are looking into potential requirements to facilitate improvements in this field, and more efficient use of radio interface and network resources.

Advanced wireless multimedia networks are ready to deliver broadband data service at a significantly lower cost than in the past, thanks to diffusion standardization. These networks offer many of the features necessary to enable M2M services in the future embedded Internet [47]. Ubiquitous Wi-Fi and mobile systems, together with P2P communication, further extend the coverage of wireless networks whereas significantly reducing cost per bit transferred. For the wireless industry, there are also profound economic motivations for M2M agile implementation. The set of potential revenue-generating services includes M2M, cloud computing, and application stores.

M2M represents a future in which billions to trillions of everyday objects and the surrounding environment are connected and managed through a range of devices, networks, and cloud-based servers. The essential components to this IoT vision are the following [47]:

- A continuum of devices from low-cost/low-power to compute-rich/high-performance
- Ultrascalable connectivity, and
- Cloud-based mass device management and services

In the M2M market, large numbers of devices are expected to be embedded, requiring low prices and power consumption. On the other hand, the important challenge is to enable low-cost connectivity that addresses not only the massive network scale but also the vastly diverse requirements dictated by the device continuum. Finally, the vision of the future is for no one device acting alone, but many devices acting together. Thus, centralized decision making and management of many devices within the cloud will become an essential value of the IoT vision.

9.5.1 M2M System Architecture

There exists a need for a cost-effective, scalable M2M solution that will support a variety of applications and devices. The expected increase in M2M

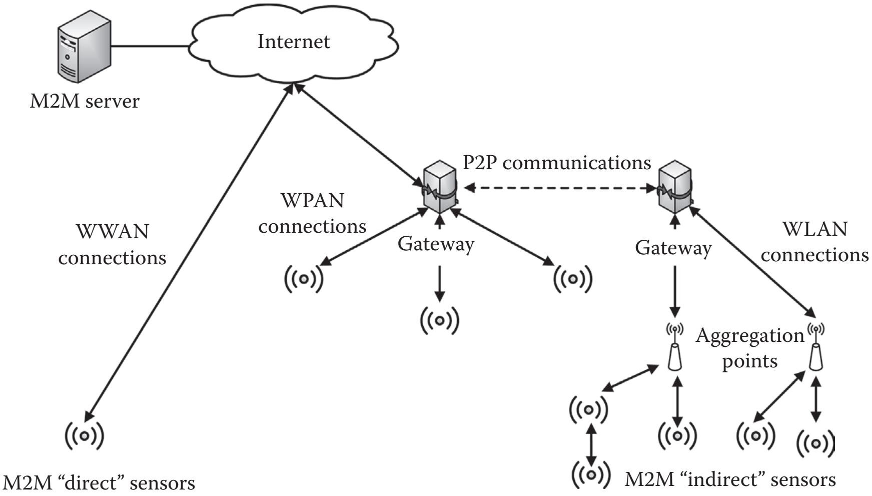


FIGURE 9.9 A high-level M2M system architecture.

devices poses a network capacity concern. A reasonable solution offers hierarchical network architectures. Multiple connectivity options are available to connect M2M devices to a server and to each other. When many devices are limited in range due to cost/size/power constraints, hierarchical deployments that provide reliable and efficient interworking between multiple communication protocols will be needed. A high-level M2M system architecture based on wireless communications is shown in Figure 9.9.

The M2M devices can be connected to the M2M server directly through a WWAN connection or an M2M gateway (aggregation point). Here, the gateway represents a smart M2M device that collects and processes data from simpler M2M devices and manages their operation. Connecting through a gateway is desirable when devices are sensitive to cost, power, or location. These devices can communicate using some lower-cost wireless interfaces (e.g., IEEE 802.11 or IEEE 802.15).

Many M2M applications will require connectivity between end devices. P2P connectivity can be supported in this architecture at various levels of hierarchy, depending on QoS requirements and the type of content. M2M systems need to be able to detect unusual events, such as changes in device location and device malfunction, and support appropriate levels of authentication for M2M devices and gateways. Enhanced monitoring and security may require changes to the network entry/re-entry procedure.

9.5.2 M2M Standardization

Most standard bodies are taking a phased approach because the requirements and applications are still evolving. Fundamental M2M features are standardized and enabled quickly, with optimizations expected in later phases as the market grows.

In the first phase, only enhancements that require software changes (e.g., MAC modifications) are enabled. In the later phases, more extensive modifications to the PHY and MAC are expected. This will accommodate advanced requirements such as those for the M2M gateways. M2M is dependent on various technologies across multiple industries. Thus, the required scope of standardization is significantly greater than that of any traditional standards development. Unique challenges related to the M2M, which must be resolved by the wireless standards bodies, include the following [48]:

- A much larger number of devices need to be supported in a M2M network than an H2H. Optimizations are needed to avoid network congestion and system overload.
- Traffic patterns of M2M devices are quite different from those of H2H networks. M2M devices might frequently access the network, only to transmit small bursts of data.
- Many types of M2M devices, running various applications with different characteristics and requirements, all need to be supported.
- As many M2M devices are fixed devices, resource management and allocation for low-mobility devices need to be optimized.
- As M2M devices may be deployed without human supervision, advanced mechanisms for security and antivandalism need to be supported.
- It's crucial for network operators to be able to offer M2M services and devices at a low cost level for mass-market acceptance.
- Enhancements for minimizing battery power usage are important for low-power M2M devices.
- Other challenges exist, including subscription management and billing.

Collaboration among standards organizations across different industries is essential. The M2M community recognized this need, and joint efforts and collaborations among standards bodies are increasing. The current status of global M2M standards development is presented in Table 9.3.

Besides developing open interfaces and standard system architectures, M2M ecosystems also need to establish a set of common software and hardware platforms to substantially reduce development costs and improve time to market. Most of the existing proprietary vertical M2M solutions have difficulty scaling.

TABLE 9.3
Status of M2M Standards Development

Standards Development Organization (Project)	M2M Development
3GPP (Release 11)	Requirements and network optimization for features such as low power, congestion and overload control, identifiers, addressing, subscription control, and security Network improvements for M2M communication, network selection and steering, service requirements, and optimizations
ETSI (M2M network architecture)	Functional and behavioral requirements of each network element to provide an E2E view
GSM Alliance (GSM operation for M2M)	Define a set of GSM-based embedded modules that address operational issues, such as module design, radio interface, remote management, provisioning and authentication, and basic element costs
IEEE 802.16p	Optimize radio interface for low power, mass device transmission, small bursts, and device authentication. M2M gateway, cooperative M2M networks, and advanced M2M features
IEEE 802.11	Update radio interface to enable use of subgigahertz spectrum
IEEE 802.15.4	Radio interface optimization for smart grid networks
WiMAX Forum (network system architecture specification)	Define usages, deployment models, functional requirements based on IEEE 802.16 protocols, and performance guidelines for end-to-end M2M system
Wi-Fi Alliance (smart grid task group)	Promote the adoption of Wi-Fi within the smart grid through marketing initiatives, government and industry engagement, and technical/certification programs
Open Mobile Alliance (device manageability)	Define requirements for the gateway-managed object
Telecommunications Industry Association (M2M SW architecture TR50)	Develop and maintain access interface standards for monitoring and bidirectional communication of events and information between smart devices and other devices, applications, or networks

9.5.3 IP Multimedia Subsystem and M2M Integration

In the next few years, IoT services and applications are likely to become an integral part of everyday life. Basic technologies that leverage seamless interaction between unconventional artifacts have already been developed and play a relevant role in different application domains (e.g., tracking and tracing, vehicular telematics, health care, remote maintenance, and control).

Early experimentation on the heterogeneous communication concept in IP multimedia subsystem (IMS) networks has shown that consolidation of strategies for harmonized data handling can greatly assist liberalization of rich content IP communication and guarantee QoS, integrity of mobility management, as well as uniformity of service charging [49].

IMS seems to be a natural solution to simplify the integration of M2M applications in a wider municipal service ecosystem. On the one hand, IMS represents an excellent integration framework that permits us to take full advantage of legacy solutions. On the other hand, IMS makes it possible to effectively interact with internal M2M mechanisms and coordinate M2M device communications to encompass relevant aspects of communication management. Also, the adoption of IMS simplifies the service management process, as involved technical staff requires only limited training to administer and maintain a new IMS-based system [50].

The need to seamlessly integrate the system with a broader service architecture suggested the adoption of IMS as the basic communication management support. IMS provides ubiquitous and production-level support for the development of novel services that are easy to integrate with existing wireless infrastructures, and further decrease costs. As a consequence, IMS is particularly well suited to realizing advanced management platforms that are able to integrate different infrastructures and service components according to specific application domain requirements.

Example 9.4

An example of IMS-enabled M2M-based management system is presented in Figure 9.10. This system consists of three main domains and is applicable in road traffic management [50].

Each retractable bollard hosts an M2M device that participates in the M2M device domain. The second domain is the network domain, which enables communication between the M2M device and the M2M server over the mobile operator network. This domain consists of the most important 3GPP evolved packet system nodes and IMS components such as

- Home subscriber server (HSS) is the database storing authentication data and profiles for clients, ranging from M2M devices to IMS-enabled clients.

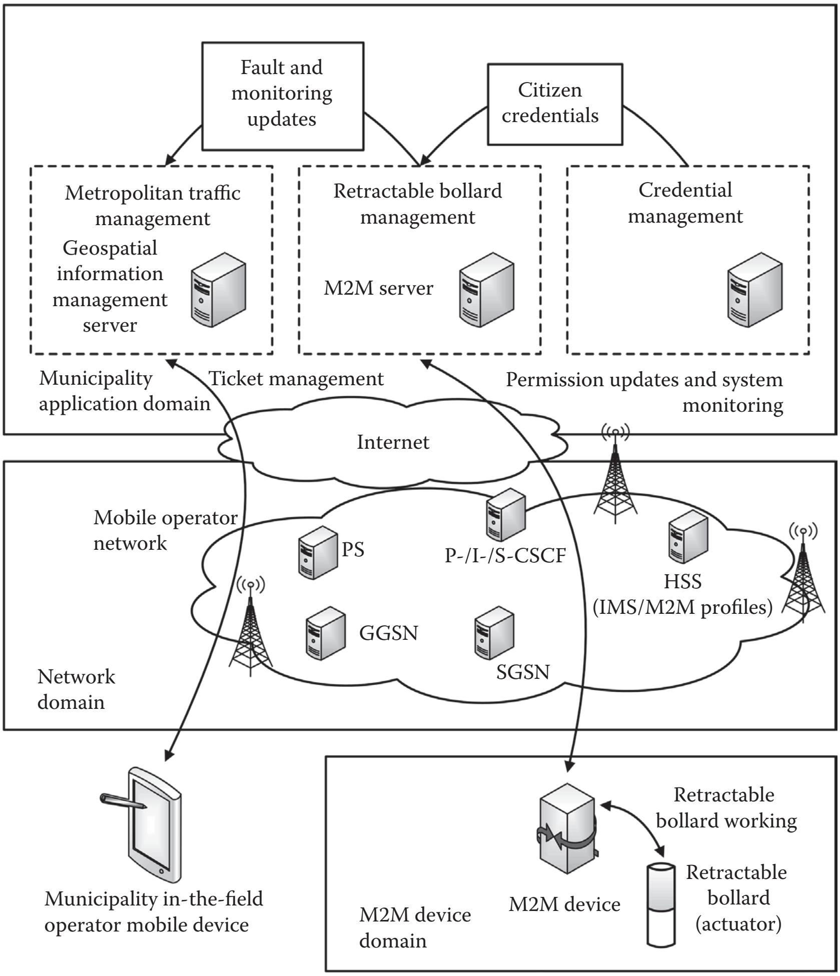

Fault and monitoring updates
Citizen credentials
Metropolitan traffic management
Geospatial information management server
Retractable bollard management
M2M server
Credential management
Municipality application domain
Ticket management
Permission updates and system monitoring
Internet
Mobile operator network
PS
P-/I-/S-CSCF
HSS (IMS/M2M profiles)
GGSN
SGSN
Network domain
Municipality in-the-field operator mobile device
Retractable bollard working
Retractable bollard (actuator)
M2M device
M2M device domain

FIGURE 9.10 IMS-enabled M2M-based road traffic management system.

- Proxy-/interrogating-/serving-call session control functions (P-/I-/S-CSCF) core entities of IMS that realize several main functions, including localization, routing SIP messages, associating an IMS client with its S-CSCF (as indicated within the client profile), and modifying the routing of specific types of SIP messages to applications servers depending on filters/triggers specified by client profiles maintained by the HSS.
- Presence service (PS) that, following a publish/subscribe model, allows users and hardware/software components to publish data to interested entities previously subscribed to the IMS PS server.

- Serving GPRS support node (SGSN) acts as a local mobility anchor node.
- Gateway GPRS support node (GGSN) acts as an interface between the mobile operator network and different packet data networks.

Detailed architecture and functions of IMS components are analyzed by Bakmaz et al. [51]. In the third domain (municipality application domain), the M2M server represents the service integration core component. That is, it interacts with M2M devices over IMS, provides suitable support to authorize citizens to access restricted areas by interacting with a credential management server to obtain currently applicable citizen's credentials, and interacts with the traffic management system.

9.6 NANONETWORKS AND IoT

Nanotechnology promises new solutions for many applications in the biomedical, industrial, and homeland security fields as well as in consumer and industrial goods [13]. Nanomachines can be defined as the most basic functional units, integrated by nanocomponents and able to perform simple tasks such as sensing or actuation. Coordination and information sharing among several nanomachines will expand the potential applications of individual devices both in terms of complexity and range of operation. Traditional communication technologies are not suitable for nanonetworks mainly due to the size and power consumption of transceivers, receivers, and other components. The use of molecules, instead of electromagnetic or acoustic waves, to encode and transmit the information represents a new communication paradigm that demands novel solutions such as molecular transceivers, channel models, or protocols for nanonetworks [14,15].

Nanonetworks do not represent a simple extension of traditional communication networks at the nanoscale. They provide a complete new communication concept in which most of the processes are inspired by biological systems. The main differences between traditional communication networks and nanonetworks enabled by molecular communication are evaluated by Hiyama et al. [52] and summarized Table 9.4.

The interconnection of nanoscale devices with existing communication networks and, ultimately, the Internet defines a new networking paradigm that is further referred to as the IoNT. The interconnection of nanomachines with existing communication networks and eventually the Internet requires the development of new network architectures.

TABLE 9.4
Comparison of Traditional Communication Networks and Nanonetworks Enabled by Molecular Communication

	Communication	
	Traditional	**Molecular**
Signal type	EM waves	Chemical
Propagation speed	Speed of light	Extremely low
Medium conditions	Wired: Almost immune Wireless: Affects communication	Affects communication
Noise	Electromagnetic fields and signals	Particles and molecules in medium
Encoded information	Multimedia	Phenomena, chemical states, or processes
Energy consumption	High	Low

9.6.1 Components of IoNT Architecture

Regardless of the final application, the following components such as nanonodes, nanorouters, nano-microinterface devices, and gateways can be identified (Figure 9.11).

Nanonodes are the simplest and smallest nanomachines. They can perform simple computation, have limited memory, and can only transmit over very short distances, mainly because of their reduced energy and limited communication capabilities. Good examples are biological nanosensor

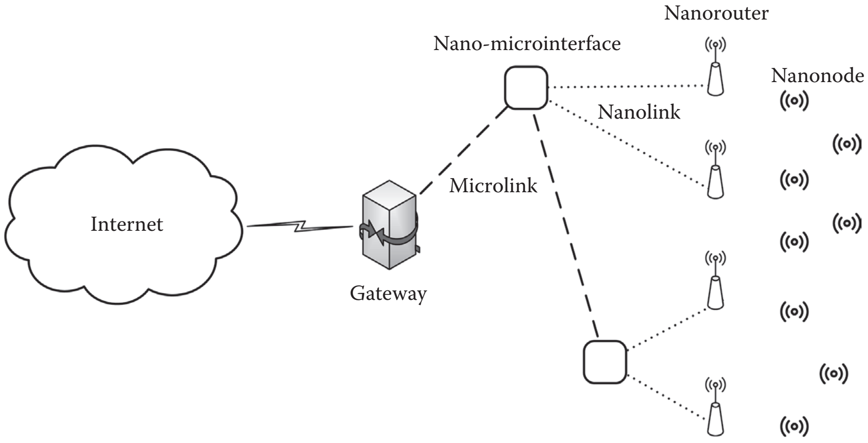


FIGURE 9.11 Network architecture for IoNT.

nodes inside the human body and nanomachines with communication capabilities integrated in all types of things such as books, keys, and so on.

Nanorouters as a type of nanodevice have comparatively larger computational resources than nanonodes and are suitable for aggregating information coming from underlying nanomachines. Also, nanorouters can control the behavior of nanonodes by exchanging very simple control commands such as on/off, standby, and so on. However, this increase in capabilities involves an increase in their size, and this makes their deployment more invasive.

Nano-microinterface devices are able to aggregate the information coming from nanorouters, to convey it to the microscale, and vice versa. They can be contemplated as hybrid devices able both to communicate in the nanoscale using the aforementioned nanocommunication techniques and to use classic communication paradigms in conventional communication networks.

In this architecture, a gateway enables the remote control of the entire system over the Internet. For example, in an intrabody network scenario [13], a smartphone can forward the information it receives from a nano-microinterface to the health care provider. On the other hand, in the interconnected office, a modem-router can provide this functionality.

9.6.2 Nanonetworks Communication

The IoNT communication concept begins at the networking of several nanomachines. Nanonetworks are not downscaled networks. There are several properties stemming from the nanoscale that require us to totally rethink well-established networking concepts. The main challenges from the communication perspective are presented in a bottom-up fashion, by starting from the physical nanoscale issues affecting a single nanomachine up to the nanonetworking protocols [13,53]. Figure 9.12 shows the design flow for the development of nanonetworks.

Communication at the nanoscale is strongly determined by the operating frequency band of future nanotransceivers and nanoantennas. Graphene-based nanoantennas have been proposed as prospective solutions for nanoscale communications [54]. The wave propagation velocity in graphene can be up to one hundred times below the speed of light. As a result, the resonant frequency of nanoantennas can be up to two orders of magnitude below that of nanoantennas built with noncarbon materials. The use of EM waves in the megahertz range can initially be more appealing than emission in the terahertz band, provided that by transmitting at lower frequencies, nanomachines can communicate over longer distances. However, the energy efficiency of the process to mechanically generate EM waves in a nanodevice is predictably very low [55].

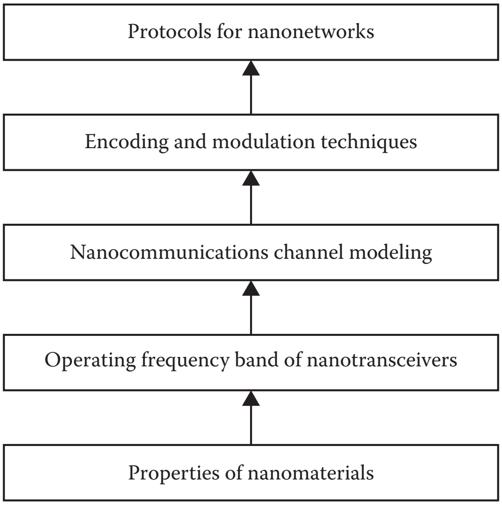

Protocols for nanonetworks
Encoding and modulation techniques
Nanocommunications channel modeling
Operating frequency band of nanotransceivers
Properties of nanomaterials

FIGURE 9.12 The design flow for nanonetworks communication.

9.7 CONCLUDING REMARKS

An emerging category of devices at the edge of the Internet are consumer-centric mobile sensing and computing devices. In a category of applications termed MCS, individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest.

Existing semantic sensor web technologies enable the integration of sensors into the Web. It was difficult to foresee the wealth of current web applications back when the Web was first created, yet now we have seen how widely adopted the Web has become. Likewise, it is difficult to predict how people will come to use the semantic Web of things. Using sensor data is clearly beneficial, because then integration with knowledge from arbitrary services is possible. For example, sensor data can be linked to geographic data, user-generated data, scientific data, and so on. A strong indicator of whether this line of development will be successful in the long run is also provided by the exponential growing amount of linked data.

An important and big step toward the IoT would be to facilitate suitable IP-based WSN technologies to support the network of things. An increase in research efforts has led to maturity in this field, yet there seems to be gaps to be filled because of the focus on how to adopt the IP to the space of things. Considering the effect of billions of new internetworked devices, the emerging IETF protocol stack for IoT should be the cornerstone for research in this field.

M2M communications in the context of the mobile Internet has been a subject of intense discussions over the past 2 years. Some people see it as the next technology revolution after the computer and the Internet. As for M2M, it presents both challenges and opportunities to the industry. Although there are significant business and economic motivations for operators and equipment manufacturers to invest in future generations of M2M services, the highly fragmented markets risk the forecasted growth of M2M markets. Two things are needed for the embedded Internet vision to materialize: the development of new technologies that scale with the growth of M2M markets, and a broad standardization effort in system interfaces, network architecture, and implementation platforms.

Although the IoT concept has been around for several years now, there are still many crucial issues that have not been solved, including heterogeneity, scalability, security, and others. The complexity of these technical issues, especially in view of the resource-constrained nature of many components and of the use of wireless communications, calls for a unified architectural view that is able to address them in a coherent fashion.

Nanotechnology is enabling the development of advanced devices that are able to generate, process, and transmit multimedia content at the nanoscale. The wireless interconnection of pervasively deployed nanodevices with all sorts of devices and, ultimately, the Internet will enable a new networking paradigm, known as the IoNTs. This new concept will have a great effect on almost every field, starting from telemedicine to military purposes. To enable communication among nanomachines, it is necessary to readjust traditional communications and to define new alternatives stemming from the nature of the nanoscale. Although nanohardware is still being developed, the definition of new encoding and modulation for nanomachines, and the development of nanonetworking structures and protocols are major scientific research issues.