IoT - IPv6 INTEGRATION HANDBOOK FOR SMEs

Maria Rita Palattella, Latif Ladid, Sebastien Ziegler
Wolfgang Kastner, Markus Jung, Mario Kofler,
Dejan D. Drajic, Srdjan Krco, Giang Nam, Rafael Marin Perez

IoT - IPv6 integration handbook for SMEs

Maria Rita Palattella, Latif Ladid, Sebastien Ziegler
Wolfgang Kastner, Markus Jung, Mario Kofler,
Dejan D. Drajic, Srdjan Krco, Giang Nam, Rafael Marin Perez[1]

May 19, 2014

[1] Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

Content

Preface **7**

1 Application and Benefits of IPv6 for the IoT **11**
1.1 Introduction to IPv6 11
1.2 Main benefits of IPv6 for the IoT 12
1.3 Integration with the Cloud 16
1.4 Integration with the mobile world 17
1.5 Integration with tags, RFID and NFC 18
1.6 Integration with building automation 19
1.7 IoT Emerging standards and trends 21

2 Application and benefits of IPv6 for SMEs **23**
2.1 Main benefits of IPv6 to a SME 23
2.2 IPv6-based IoT Applications: IoT6 use cases 24
2.2.1 Use Case 1: Smart Office and legacy devices Integration 24
2.2.2 Use Case 2: Safety alert and dynamic routing 25
2.2.3 Use Case 3: Building maintenance 27
2.3 IPv6 Business Case: Mobile phone as a sensing tool 28

3 Practical steps: How to deploy IPv6 in an SME **33**
3.1 How to set up IPv6? 33
3.2 Enabling low-power IPv6-IoT networks with 6LoWPAN 36
3.3 Enabling DNS with IPv6 37
3.3.1 DNS Considerations about Special IPv6 Addresses 38
3.3.2 Recommendations for Service Provisioning Using DNS 38

3.4 Enabling a Mail Server with IPv6 40
3.5 Tunneling for providing IPv6 connectivity 41
3.6 Enabling a web server with IPv6 43
3.7 Enabling Security with IPv6 45
3.7.1 Neighbor discovery threats 46
3.7.2 DHCP related threats 47
3.8 Integrating legacy devices 48

4 Conclusion **53**

5 Glossary **55**

6 References **59**
6.1 Useful web sites and tools 63

Acknowledgements **65**

List of Figures

1 IoT6 Consortium 9
1.1 Product management architecture 18
1.2 Integration of home and building automation technologies 20
2.1 Use Case 1: Smart office presence detection 24
2.2 Use Case 2: Safety alert 25
2.3 Use Case 3: Building maintenance 27
2.4 IPv6 communication between laptop and CoAP Server 29
2.5 IPv6 communication between laptop and MindWave device 31

Preface

In recent years, the Internet has been facing two major revolutions: (i) moving from a human centric network to the Internet of Things (IoT), with more devices connected to the Internet than human beings; and (ii) moving towards the Internet Protocol version 6 (IPv6), with its almost unlimited number of IP addresses.

The present handbook has been written to support SMEs in this transition in order to help them seize a piece of this new emerging market. It has been realized in the framework of a European research project, named *IoT6: Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability*, by a consortium of academic and industrial partners bringing complementary expertise, from Europe and Korea, including:

- Mandat International (Switzerland)
- Ericsson (Serbia)
- Run My Process (France)
- University College of London (UK)
- University of Murcia (Spain)
- Vienna University of Technology (Austria)
- University for Applied Sciences Western Switzerland (Switzerland)
- University of Luxembourg (Luxembourg)
- Korea Advanced Institute of Science and Technology (S.Korea)

The IoT6 research project aimed at exploiting the potential of IPv6 and related standards (6LoWPAN [1], COAP [2], etc.) to overcome current shortcomings and fragmentation of the Internet of Things, in line with the Internet of Things European Research Cluster (IERC) vision and the EC recommendations. Its main challenges

and objectives were to:

1. Research the potential of IPv6 features and related standards to support the future Internet of Things and to overcome its current fragmentation.

2. Design an Open Service Layer to provide mechanisms for discovery, look-up and integration of services offered by Smart Objects to distributed clients and devices connected via IPv6.

3. Explore, based on Service-Oriented Architecture, innovative forms of interactions with:

- Information and intelligence distribution;
- Multi-protocol interoperability with and among heterogeneous devices, including various non-IP based communication protocols;
- Device mobility and mobile phone networks integration;
- Cloud computing integration with Software as a Service (SaaS);
- Tags and Smart Things Information Services (STIS) [3].

Therefore, the main outcomes of IoT6 are recommendations on how to exploit IPv6 features for the IoT and a well-defined IPv6-based Service Oriented Architecture enabling interoperability, mobility, cloud computing and intelligence distribution among heterogeneous smart components, applications and services - including business processes management tools and smart buildings.

IoT6 has demonstrated the high potential of IPv6 for the future IoT. As you will discover in the following sections, IPv6 provides an ideal solution to interconnect unlimited number of heterogeneous smart things. It is also a powerful integrator for the integration of the Internet of Things with Cloud applications and web services.

IoT6 has worked in close cooperation with International Forums (e.g., IPv6 Forum, ITU-T JCA-IoT), standardization bodies (e.g., ETSI, M2M, ETSI), major industries and other research projects (e.g., IoT-A, IoT-I, SEnsei, etc.) with an European and international perspective. We are pleased to share with you part of our acquired knowledge and hope it will be useful for your own developments. For further information do not hesitate to visit our website (*www.iot6.eu*) or to contact us at: *iot6@mandint.org*.

The handbook is organized as follows: Chapter 1 gives an overview of the main benefits that the Internet of Things can gain by using IPv6, and describes some potential

area of applications, such as cloud, mobile word, building automation. Chapter 2 details for the SMEs, the main benefit of integrating IPv6 and IoT. As an example, some of the use (and business) cases developed in the context of the IoT6 project are presented. Chapter 3 provides some practical advices and several technical details about how SMEs can set up IPv6, and in detail, how different services (mail and web server, DNS, etc.) can be enabled with IPv6. Finally, Chapter 4 concludes the book.

Figure 1: IoT6 Consortium

1

Application and Benefits of IPv6 for the IoT

1.1 Introduction to IPv6

Global Internet human users are currently estimated at 2.4 billion and are further projected to climb to 3 billion by 2015. More significantly, the number of Internet connected objects has surpassed the number of connected human beings, and is expected to expend far beyond the human population, with 20 to 50 billion interconnected smart things. Over the last decades, the Internet Protocol version 4 (IPv4) [4] has emerged as the mainstream protocol for networking layer. However, this protocol was not designed for the Internet of Things (IoT) [5] and is inherently limited to about 4 billion addresses. At the global level, IANA has entirely exhausted its IPv4 address allocations on the 3rd February 2011; and two out of five RIRs (Regional Internet Registries) have achieved their address allocation limit in April 2011 by APNIC and in August 2012 by RIPE. The Internet Protocol version 6 (IPv6) [6] has been adopted by IANA and the RIRs to surpass the IPv4 limitations and to address the growing demand. IPv6 provides an almost unlimited (2^{128}) number of unique Internet addresses. It also provides new features enabling an easier configuration of devices, improved security, and enable real peer-to-peer connections, without passing through NAT barriers. All those elements contribute to turn IPv6 into a natural candidate for IoT addressing and networking.

Many devices are already interconnected through the Internet Protocol, including printers, sensors, lightings, healthcare systems, energy meters, video cameras, TVs and heating control systems. The emergence of IPv6-related standards

specifically designed for the IoT, such as 6LoWPAN, CoAP, and CoRE, has enabled highly constrained devices to become natively IP compliant. IPv6 is being referred to in a growing number of IoT and M2M related standards, such as oneM2M or the IEEE 802.15.4g [7] protocol, which will support Advanced Metering Infrastructure (AMI) for smart cities deployments.

1.2 Main benefits of IPv6 for the IoT

Why should the Internet of Things care about IPv6? As described in this section, many answers can be given to this question, and several arguments can be provided for considering IPv6 as a key enabler for the future Internet of Things.
First of all, it has to be noted that *world-wide adoption of IPv6* is just a matter of time. The Internet Protocol is a must and a requirement for any Internet connection. It is the addressing scheme for any data transfer on the web. The limited size of its predecessor, IPv4, has made the transition to IPv6 unavoidable. The Google figures available at [8] reveal an IPv6 adoption rate following an exponential curve, doubling every 9 months. Its universal acceptance will make IPv6 the protocol of choice for interconnecting almost all the smart and heterogeneous devices in the Future Internet. Beyond the exhaustion of the IPv4 address space, and the consequent migration of the Internet to IPv6, it is widely recognized that IPv6 due to its added features is very suitable for the connectivity of distributed IoT components and can provide several advantages to IoT. In the following section, we list the main IPv6 features and their benefit for IoT.

- **Scalability**
  IPv6 provides a highly scalable address scheme with 2^{128} unique addresses representing 3.4×10^{38} addresses. In other words, more than 2 billion addresses per square millimeter of the Earth's surface are available. Thus, it is quite sufficient to address the needs of any present and future communicating device.

- **Enabling the extension of the Internet and the web of things**
  Thanks to its large address space, IPv6 enables the extension of the Internet to any device and service. Experiments have demonstrated the successful use of IPv6 addresses to large scale deployments of sensors in smart buildings, smart cities and even cattle. Moreover, thanks to the CoAP protocol, low-power constrained

devices can behave as web services which are easily accessible and fully compliant with the REST architecture [9].

• **Solving the NAT barrier**
Due to the limits of the IPv4 address space, the current Internet had to adopt a trick to face its unplanned expansion: the Network Address Translation (NAT). The latter enables several users and devices to share the same public IP address. This solution is working but with two main trade-off:

- The NAT users are borrowing and sharing IP addresses with others. Hence, they do not have their own public IP address, which turns them into homeless Internet users. They can access the Internet, but they cannot be accessed from the Internet.

- It breaks the original end-to-end connection and dramatically weakens any authentication process.

Moreover, the management of NATed address space is not straightforward, and it becomes difficult as they become more prevalent. Finally, many applications do not work with NATs, because they require globally reachable addresses.

• **Improving Routing**
IPv6 provides end-to-end connectivity, with a more distributed routing mechanism. The IPv6 protocol makes routing more efficient and hierarchical by reducing the size and complexity of routing tables. Moreover, IPv6 allows Internet Service Providers (ISPs) to aggregate the prefixes of their customers' networks into a single prefix and announce this one prefix to the IPv6 Internet. Finally, fragmentation in IPv6 networks is handled by the source device, rather than the router, using a protocol for discovering the path's maximum transmission unit (MTU).

• **StateLess Address AutoConfiguration (SLAAC)**
IPv6 provides an address self-configuration mechanism (stateless mechanism). The nodes can define their addresses in very autonomous manner. A router sends the prefix of the local link in its router advertisements. A host can generate its own IP address by appending its link-layer (MAC) address, converted into Extended Universal Identifier (EUI) 64-bit format, to the 64 bits of the local link prefix. This enables the

drastic reduction of the configuration effort and cost for managing the system. This stateless autoconfiguration implies that there is no longer any need to configure IP addresses for end systems, even via Dynamic Host Configuration Protocol (DHCP). This allows new equipment to communicate with the network once it is detected; in other words, devices are ready to use on demand (i.e., plug-and-play). Such IPv6 autoconfiguration feature is very useful in IoT networks to allow entire sensor networks to come on-stream.

• **Multicast and Anycast**

Multicast was already supported in IPv4 but the technology was defined and implemented in an ineffective way in practice, and thus, has been hardly used. All the components of the Internet were easily compromised and exposed to risks, by using IPv4 multicast. With IPv6, the use of multicast is much more risk-free thanks to the way of creating the addresses for destinations groups. In fact, IPv6 defines several multicast addresses (i.e., FF01::1) for the Internet auto configuration precedures.

IP multicast is particularly useful in the IoT environment. In large IoT deployments, it allows to distribute one command or set of data to all the devices directly or indirectly on the Wireless Area Network (WAN).

Unlike multicast, anycast, which requests an answer from any entity receiving the packet, was not supported at all in IPv4. Anycast allows to verify if devices or other resources are available in the network. This provides a convenient mechanism for accessing both resource directory and resource depositary entities. It makes the use of alternate resources much easier, thus improving the resilience of the system. Anycast is very useful in Local and Sensor networks. It may be used for IoT resource repositories, security servers and multi-homed gateways.

• **Quality of Service**

The basic IPv6 address structure has provision for several bits to define a given level of Quality of Service (QoS) to be provided while treating packets. For instance, IPv6 is able to use QoS features such as *Diffserv* or *IntServ* to prioritize urgent sensor alarms. This feature, given the advantage that it can offer, has been already exploited. In fact, commercial routers have already been configured to use these bits in IPv6 addresses.

• **Mobility**

IPv6 provides strong features and solutions to support mobility of both end-nodes, and routing nodes (and thus, for resources or agents in IoT applications). In fact, an architecture and deployment provision for mobility were already defined in IPv4. However, the Mobile Internet Protocol (MIP) on which they were based, was not very efficient: each packet had to go via Home Agent using a triangular path. A cut-through technique could have been defined in IPv4 but was not. In IPv6 a new version of MIP, namely MIPv6 has been developed as the Internet standard.

In comparison with MIP, MIPv6 provides less handover latency thanks to several optimizations in the mechanisms: Movement Detection (MD), Duplicate Address Detection (DAD) and Binding Update (BU).

• **Security**

IPv4 was designed without security in mind. Therefore, security in IPv4 communications had to be guaranteed by end-nodes. In other words, the transmission/reception of sensitive data through a secure (encrypted) channel was the responsibility of the application providing the service itself. To overcome such security limitations, new features have been included while designing IPv6. Among the features that support or improve security which can be mentioned are: (i) Introduction of IPSec, designed for IPv6 due to restoration of end to end connectivity, (ii) Mandatory use of IPsec for Mobile IPv6 to secure the return routability, (iii) Large Addressing Space, and (iii) Neighbor Discovery.

By using an Authentication Header (AH) and an Encapsulating Security Payload (ESP) both of which are defined as IPv6 extension headers, IPSec is able to respectively provide authentication, data integrity and confidentiality. With IPv4, the use of IPSec was running into some issues with end to end control because of the use of NAT, and the change of source or destination addresses into the packet when traversing a NAT gateway. With IPv6 every device is globally addressable end-end, and thus, the aforementioned issue was resolved.

By using 128-bits addresses, reconnaissance attacks and port scanning that were relatively simple tasks in IPv4, become practically impossible with IPv6. Finally, the Neighbor Discovery (ND) mechanism, used for router and prefix discovery, together with the StateLess Address AutoConfiguration (SLAAC) feature contribute to make IPv6 more secure than IPv4. ND and SLAAC are both implemented in the Internet Control Message Protocol for IPv6 (ICMPv6).

• **IPv6 version available for low-power devices**
The use of IPv6 for IoT applications has been investigated for many years. One of the main outcomes from research in this area is a compressed version of IPv6 for low power devices, namely I*Pv6 over Low power Wireless Personal Area Networks (6LoWPAN)*. It is a simple and efficient mechanism that allows to shorten the IPv6 address size for constrained devices, while enabling border routers to translate these compressed addresses into regular IPv6 addresses.

• **Fully Internet compliant**
IPv6 is fully Internet compliant. In other words, it is possible to use a global network to develop one's own network of smart things or to interconnect one's own smart things with the rest of the World.

1.3. Integration with the Cloud

The notion of Cloud Computing has been around for some time now in the IT World and has proven to be a big paradigm shift for the whole industry. The core notion is the fact that different resources, hardware as well as software, can be mutualized and eventually sold as services by providers.

Virtualization, a notion that is regularly associated to Cloud computing, is a technical solution that enables Cloud computing: it enables the creation of different resources independently of the underlying hardware. Virtualization is a solution for making better use of hardware by transforming them into more flexible resources to be managed and shared. Therefore, it has been a key factor for the cost decrease that has been observed in the last years.

With the adoption of virtualization, the number of accessible resources has increased and therefore the need for addresses. The adoption of cloud computing and virtualization solutions will, as a consequence, contribute to the need for the whole industry to switch to the IPv6 paradigm.

1.4 Integration with the mobile world

The integration of IPv6-based Internet of Things into mobile phone networks enables the mobile phones to provide ubiquitous access to the smart things connected to the developed IoT6 architecture. In detail, it enables smart things or systems connected to the IoT6 architecture to connect and send messages to the mobile phone, and on the other hand, it enables the architecture to use mobile phones as mobile sensing tools, and retrieve from them information, such as temperature, motion, localization, etc.

Every smarthphone is equipped with a number of embedded sensors, like GPS, microphone, speaker, and a camera. If the data from the phones sensors could be accessed from the Internet, it could be combined and used for obtaining a bigger picture about the surrounding environment. Given the huge amount of smarthphones active in the world, the potential of this solution is huge.

The *Long Term Evolution* (LTE) is a global standard for mobile networks defined by the 3rd Generation Partnership Project (3GPP) for the fourth generation of mobile broadband. LTE network provides connectivity to IoT6 devices and makes them available to the rest of the IPv6 enabled environment in the sense of discovery, access and management. LTE device obtains an IP address from a LTE network. It is up to the operator to decide and define if addresses for LTE devices will be assigned to a separate Access Point Name (APN) or if the LTE devices will be treated as regular LTE users. A good overview of LTE networks commercially launched worldwide can be found in [10].

A mobile phone can be an IoT6 device itself (with its own embedded sensors) or, it can acts as a gateway in the IoT6 architecture, if one or more devices are connected to it. In the latter case, the mobile phone can represent a gateway also for sensors located into devices that do not support IPv6 protocol. For instance, they can be devices connected to the gateway via Bluetooth, Infrared, etc. In order to get all the devices connected via the Internet (i.e., as IoT implies by nature), it is necessary that these non-IPv6 devices get an Internet access over the IPv6 network. To this aim, there should be a gateway able to communicate through IPv6, and at the same time able to connect to a device via non-IPv6 wireless links (e.g., Bluetooth, infrared, etc.) and/or through wired links (e.g., USB connection). Therefore, there are several different approaches for integrating an IPv6-based Internet of Things into mobile phone networks enabling mobile devices to provide access to smart objects as well as to use mobile devices as sensors/actuators.

1.5 Integration with tags, RFID and NFC

RFID technology was initially seen as the prerequisite for the Internet of Things in the early days. Storing Electronic Product Code (EPC) [11] - a global identification system that is developed by GS1 [12], RFID gives physical objects a globally unique ID that distinguishes themselves to others. By equipping physical objects with RFID tags, they can be managed and inventoried automatically by a computer system called EPC Network, which releases companies from heavy human resource requirements. Today, physical objects are not only equipped with tags but are also embedded computing/communication devices. This technology outbreak along with the increasing adoption of IPv6 creates new opportunities for realizing the Internet of Things with better application scenarios and more fruitful services. Accordingly, physical objects can be identified by either tag technologies such as RFID, QR/Bar-code or IPv6 address/domain name. In addition to the object identification, product management can now be aware of a products' condition based on embedded sensing devices remotely in real-time. This is illustrated in Fig. 1.1, which shows a product manufacture process and supply chain that is governed by an EPC Network. The RFID-tagged products also have sensing and communication capability so that their preserving condition is in tight control.

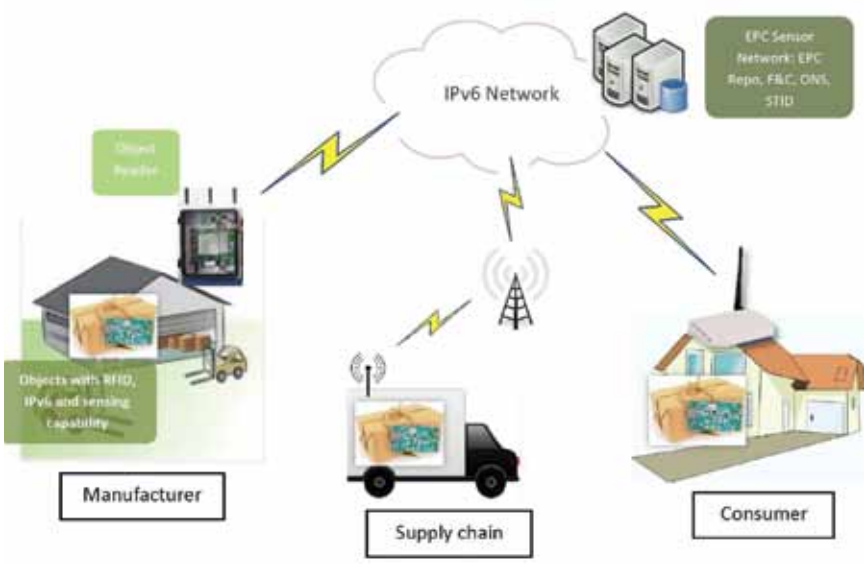

Figure 1.1: Product management architecture

Modern EPC Networks that are able to capture objects' real-time data are called EPC Sensor Network or EPCSN. EPCSN usually consists of an EPC repository where the events generated by objects are kept, a Filter and Collection (F&C) middleware that filters raw data reported by Object Readers and generates EPC events to EPC repository, and Object Readers that capture objects' data directly. In addition, Object Naming Service (ONS) is used to match proper EPC repository with a given object's EPC number. In advanced EPCSN, a Smart Thing Identification component can be deployed to resolve an EPC number into its correspondent IPv6 address (and vice-versa). Therefore, by exploiting this EPCSN architecture, product manufacturers can manage the life cycles of objects in an automatic fashion, the captured data can be used for big data analysis and the end users can leverage the IPv6 connection of the objects to realize advanced smart thing services. Furthermore, through the EPCSN, the authenticity of the objects can be easily verified by querying the repository for all the captured events and data. Thus, the system could prevent fake products to be consumed. One of the trendy technologies used in such EPCSN is the support of Near Field Communication (NFC) in modern smarthphones.
NFC is a subset of RFID technology that operates at a distance of 10cm or less and prices are cheap. By leveraging the NFC tagging technology for physical objects, their identities can be acquired easily through the users' smartphones. Thus, it could enable more elaborate application scenario from the end users' viewpoint.

1.6 Integration with building automation

The integration of wireless and wired home and building automation technologies will be one of the key aspects of the future Internet of Things networks. Currently, there are many available technologies enabling home and building automation, including among others: KNX, BACnet, ZigBee, EnOcean. Each of them has a tailored communication stack and defines custom standards for the different communication layers. In detail, these systems have different physical requirements on the communication media, and custom ways of transmitting frames and exchanging messages. They come with specific ways of addressing devices and entities and the application layer semantics differ and are in some cases adjusted to specific domain. Further, new technologies are emerging in the advent of smart cities and smart grids. For instance, Wireless M-Bus is a protocol which allows to interact with smart meter devices that

provide measurements about the power consumption and the power quality in real-time. The huge heterogeneity is a strong inhibitor for innovative application use cases that rely on various technologies.

IPv6 and the open communication stack developed within the IoT6 research project, addresses this heterogeneity and tries to come up with a unified way for communication. In this case, IPv6 provides a common network layer for global end-to-end connectivity: message exchange based on Web services and open message encoding formats like JSON then allows the creation of interoperable communication interfaces. Within IoT6 a mapping, for a selected number of home and building automation technologies, to the so-called IoT6 stack, based on IPv6, CoAP and JSON protocols, has been defined. Furthermore, at the application layer the adopted information model is based on the Open Building Information Exchange (OBIX) [13] standard, which provides an abstraction for typical features found in home and building automation technologies. In this way, multi-protocol interoperability can be achieved amongst heterogeneous technologies. Beside home and building automation technologies, further information sources like RFID readers and real-time weather data can be integrated based on the proposed communication stack and information model. This integration layer eases and simplifies the creation of innovative application scenarios. An overview of the IoT6 integration of home and building automation technologies is given in Figure 1.2.

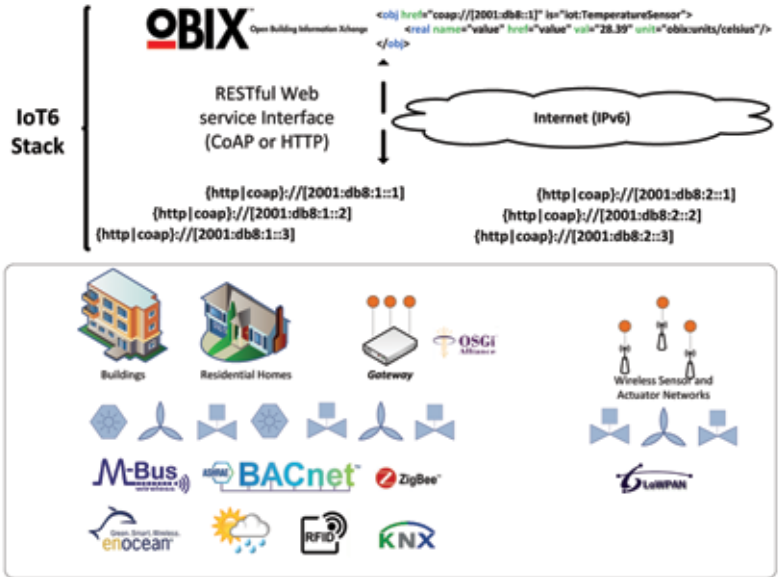

Figure 1.2: Integration of home and building automation technologies

As a practical outcome of the IoT6 research project, a prototype of a Java based integration middleware based on OSGI, coming with a protocol bundle for each technology, has been created and released as an open source project [14] and can support SMEs to tackle the heterogeneity of currently deployed home and building automation technologies. Furthermore, a Contiki based reference implementation of the stack is also provided that illustrates how to deploy the stack on very constrained devices like sensors and actuators operated with micro-controllers and limited computational resources within wireless sensor and actuator networks based on 6LoWPAN. In parallel, IoT6 has tested and researched solutions such as [15], which enable extensive multi-protocol interoperability between IPv6 and all sorts of communication protocols and devices.

1.7 IoT Emerging standards and trends

The IoT requires software architectures that are able to deal with a large amounts of information, queries, and computation, making use of new data processing paradigms, stream processing, filtering, aggregation and data mining, all of this sustained by communication standards such as HyperText Transfer Protocol (HTTP) and Internet Protocol (IP). In contrast, due to the nature of IoT objects, very low power consumptions are required so any object can plug into the Internet while being powered by batteries or through energy-harvesting. Energy is wasted by the transmission of unneeded data, protocol overhead, and non-optimized communication patterns; these need to be taken into account when plugging objects into the Internet.

Existing Internet protocols such as HTTP and Transmission Control Protocol (TCP) are not optimized for very low-power communication, due to both verbose meta-data and headers, and the requirements for reliability through packet acknowledgement at higher layers, which hinders the adaptation of existing protocols to run over that type of networks. In order to interconnect as well as Internet-connect several IoT devices (e.g., RFID, sensors, machines, etc.), *a low power, highly reliable*, and Internet-enabled communication stack is needed.

Starting in 2003, various IEEE and IETF standardization bodies started putting together a framework for the communication protocols of the emerging IoT systems. Specifically, the 6LoWPAN [1], ROLL [16] and CoRE [17] IETF Working Groups have defined protocols at various layers of the Low power and Lossy Net-

works[1] (LLNs) protocol stack, including an IPv6 adaptation layer, 6LoWPAN [1], a routing protocol, RPL [18] and a web transfer protocol, CoAP [2]. This protocol stack so far has been used with IEEE802.15.4 low-power radios [19], whose limitation in mesh-networking conditions has become apparent only recently.

To overcome such limitation, the IEEE802.15.4e standard [20] has been published in 2012 as an amendment to the IEEE802.15.4-2011 Medium Access Control (MAC) protocol. Three different operative modes have been defined in the IEEE802.15.4e standard. Among them, the Timeslotted Channel Hopping (TSCH) mode is the latest generation of ultra-lower power and reliable networking solutions for LLNs. At its core is a medium access technique which uses time synchronization to achieve ultra low-power operation and channel hopping to enable high reliability. Its core technology is similar to the one used in industrial networking technologies such as Wireless HART and ISA100.11a, resulting in comparable performance. However, unlike these industrial protocols, IEEE802.15.4e TSCH focuses on the MAC layer only. This clean layering allows for TSCH to fit under an IPv6 enabled protocol stack for LLNs and IoT applications.

A new Working Group called 6TiSCH [21] has been recently formed within the IETF with the aim to link IEEE802.15.4e TSCH capabilities with prior IETF 6LoWPAN and ROLL standardization efforts and recommendations. Specifically, it aims to (i) define an open standard-based architecture (similar to the one adopted by the IoT6 project), reuse existing protocols when possible, and (ii) face networking and routing issues, among many other challenges. 6TiSCH will highlight best practices, and standardize the missing components to achieve industrial-grade performance in terms of jitter, latency, scalability, reliability and low-power operation for IPv6 over IEEE802.15.4e TSCH.

[1] LLLNs allow to interconnect a large number of resource-constrained devices, forming a wireless mesh network. To be connected to the Internet, a small number of border routers (BRs) usually serve as gateways between each LLN and the Internet. Such LLNs have a wide range of IoT applications, including building and home automation, industrial process control and smart urban environments.

2

Application and benefits of IPv6 for SMEs

2.1 Main benefits of IPv6 to a SME

In today's technology driven environment, to succeed, businesses need to remain up-to-date with the latest developments. The Internet plays a critical role in business operations and as such with the impending transition of IPv4 to IPv6, Small Medium Enterprises (SMEs) need to adapt early to meet the new challenges and reap the benefits.

The present handbook, realized in the context of the IoT6 project, aims to support SMEs in this transition, and thus, help them seize a piece of this new emerging market. Most SMEs rely on their Internet Service Providers (ISPs) for Internet connectivity and should check that they are able to provide access over IPv6 as a matter of urgency. Any hardware or software that is bought off the shelf should be IPv6 ready (even though it may need to be configured). Old office equipment such as routers may not be IPv6 compatible and may need upgrading or even replacing. SMEs should make it a priority to adopt IPv6. *By ensuring that all devices connected to the Internet are compatible with IPv6, SMEs can ensure they stay connected and safeguard the sustainable growth of their business.* A carefully planned and strategically executed implementation of IPv6 will be far less disruptive for an organization than a last-minute, rushed roll-out. Hereafter, we provide some guidelines to SMEs about the main steps to follow for setting up and deploying IPv6.

2.2 IPv6-based IoT Applications: IoT6 use cases

In order to highlight the benefits of an IPv6-based IoT, three different use cases that have been implemented in the context of the IoT6 project, are presented. The use cases demonstrate how it is possible to interconnect different devices, and create interactions between different services. In the first use case, we illustrate the integration of legacy building automation devices into a homogeneous IoT IPv6-based smart office. In the second use case, an advanced scenario regarding building safety is described. And finally, in the last use case, the replacement of a faulty device focusing on building maintenance, is described.

2.2.1 Use Case 1: Smart Office and legacy devices Integration

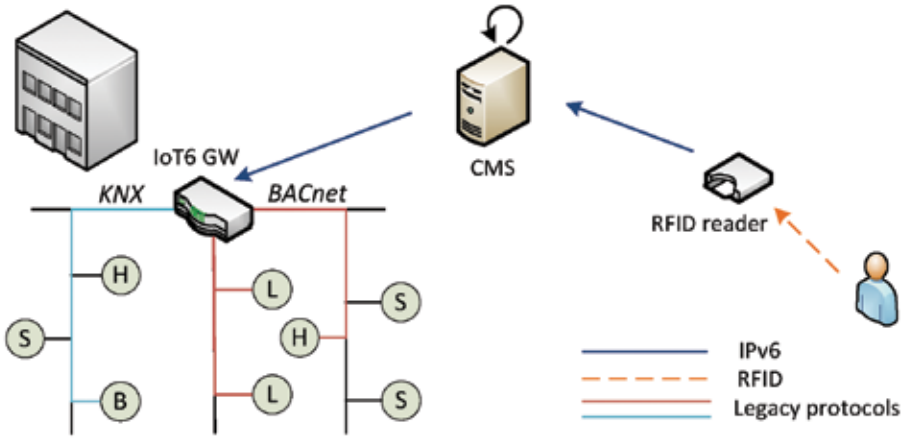

Figure 2.1: Use Case 1: Smart Office presence detection

As there still exists a heterogeneous landscape and large variety of legacy devices and networks in the building automation domain, their integration into the Internet through a single interface is still a challenge to face. However, these challenges can be addressed by IPv6 and the IoT. In the presented Smart Office scenario, several automation devices of different legacy networks (i.e., BACnet, and KNX) are integrated through a gateway which is responsible for translating legacy protocol messages into IPv6 packages and providing a homogeneous view on the underly-

ing heterogeneous networks and associated devices. Figure 2.1 illustrates (i) an IoT6 Gateway (IoT6 GW) integrating several legacy devices, (ii) an IPv6-enabled RFID reader, and (iii) a Control and Monitoring System (CMS) as service in the IoT cloud. The Smart Office use case starts when an employee enters the building and presents his/her RFID badge to the system's RFID reader. As the RFID reader is IPv6-enabled it may directly communicate with the CMS using IPv6. The CMS subsequently chooses the employee's comfort profile for his/her office and sends settings and commands to the IoT6 GW which integrates devices of the particular office into the IoT. The IoT6 GW controls a variety of different devices from heterogeneous building automation networks and masks this heterogeneity by providing a uniform IPv6 interface for all devices. The IoT6 Gateway can later on be used to set user-defined preferences for the employee in his/her office. In the example case, the heating actuator setpoint (H) and two brightness actuators (L) integrated through a BACnet network are adjusted. At the same time, also the position of the sunblind (B) which is controlled via a KNX network is adapted according to user preferences. For the CMS, the idiosyncrasies of the different underlying legacy networks make no difference as the IoT6 Gateway transparently integrates these devices into the IoT in a uniform way.

A similar situation to the one illustrated in Figure 2.1 can be observed when the employee leaves the office building. As soon as the employee provides his/her RFID badge to the RFID reader, the CMS is informed that the employee is about to leave the building. In this case, the CMS can execute an energy-saving rule which turns off all devices in the employee's office. Alternatively, a presence sensor in the office combined with a time-out could be used to detect absence and initiate the energy-saving scenario.

2.2.2 Use Case 2: Safety alert and dynamic routing

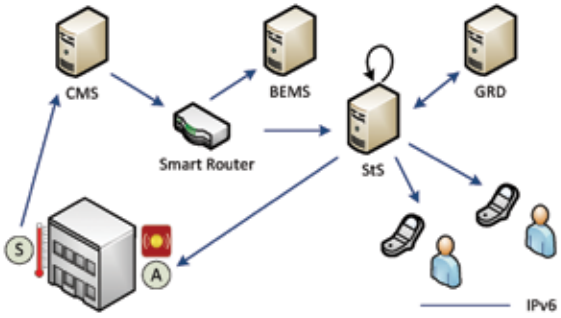

Figure 2.2: Use Case 2: safety alert

The second use case, which is slightly more complex than the first one, is focused on an emergency situation and the capabilities of an IoT architecture to deal with this. As initial setup for this use case, an IPv6-enabled temperature sensor (S) is considered which periodically sends an update of the sensed temperature value to a Control and Management System (CMS) available as service in the IoT. The starting point for this use case is a sensor that reads a temperature which is too high (i.e., outside the boundaries of usual operation). The Control and Management System detects this abnormality and flags the received message as an alert message. It sends the value to a smart router which is an IoT component that according to the type of the message may take different routing decisions. If a normal temperature message is received the smart router sends the temperature messages to a Building Energy Management Server (BEMS) that may be responsible for logging and reporting the energy demand of a building.

In the present case (excessive temperature), however, the smart router detects the priority of the message (alert) and according to the tagging carried out by the CMS forwards the value to a specific Safety Server (StS) which is responsible for handling alert situations. As the StS receives the abnormal value, it first contacts the Global Resource Directory (GRD) to gather information about the location of the sensor. If the StS already has a list of alarm devices with their location, it can compare the location of stored devices with the location of the temperature sensor to directly turn on an IPv6-enabled alarm device (A) in the vicinity of the alert situation. If the StS has no pre-stored alarm devices for the area for which the alarm was reported, it is possible to issue another query to the GRD service requesting alarm devices that are in the vicinity (e.g., found in 15 meters radius) of the alert. Any device capable of signaling an alarm which is found can subsequently be switched on. Furthermore, the StS may have a list of mobile phones of people in charge for alert situations (e.g., fire wardens, system engineers).

In this case, the CMS has to gather information about the current location of the IPv6-enabled mobile phones from the Global Resource Directory through an additional query. After this information is received, the CMS can inform responsible persons near the area of interest about the alert situation via their mobile phones. As Figure 2.2 shows, all communication is handled via IPv6 which emphasizes the diversity of devices and components that may be integrated in the IoT. If legacy devices are involved, either on the sensor or on the actuator side, an IoT6 Gateway can be used for integration as described previously.

2.2.3 Use Case 3: Building maintenance

The third use case is related with building service maintenance. It involves a variety of IoT components and demonstrates how these components in combination with IPv6 communication can be used to detect device failures in a building and investigate as well as fix the cause. In the presented case (Figure 2.3), a number of sensors are connected to the IoT through an IoT6 gateway (IoT6 GW) which assures that all legacy sensors can be accessed in a uniform way through IPv6 communication (as in the Use Case 1). This use case starts with the failure of a legacy component in the subnet controlled by a specific IoT6 gateway, e.g., a temperature sensor. Usually, a Control and Management System (CMS) observes the value of a temperature sensor, for example, to detect safety situations or perform energy reporting (as in Use Case 2). In the case an observed sensor silently fails, a time-out occurs at the CMS, indicating that something is wrong with the device. A message is generated at the Control and Management System and sent to the Maintenance Tool (MaT) for further examination.

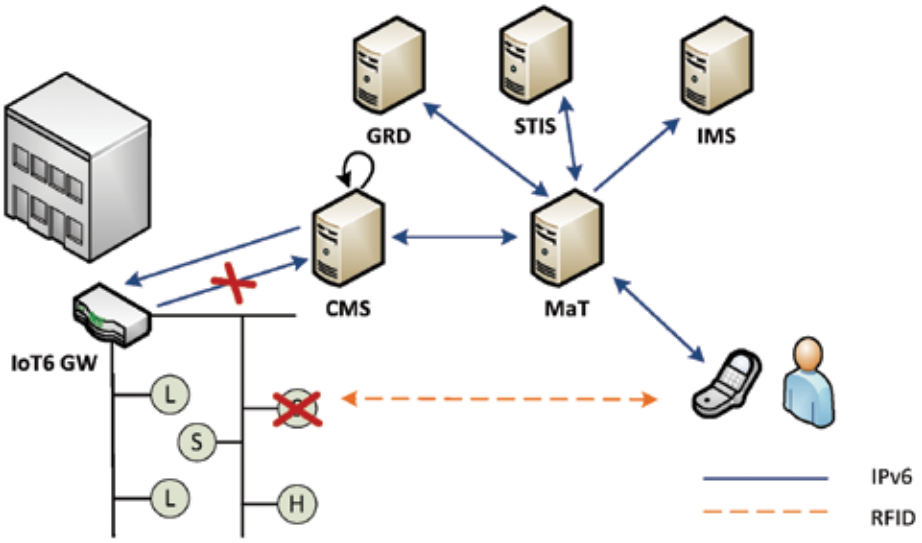

Figure 2.3: Use Case 3: building maintenance

In its simplest form, the Maintenance Tool could also run on a local CMS but presently is pictured as a global service in the IoT cloud. As soon as the MaT gets the message about the failure of a device, it creates an alert ticket and sends out failure notifications to a variety of mobile phones of responsible persons (e.g., system engineers). The group of recipients may again be based on the current location of the mobile phones for which a lookup call to the Global Resource Directory would be necessary (as in Use Case 2). A person associated with one of the contacted mobile phones seeks out the faulty device and uses a maintenance app on the mobile phone to scan its RFID tag. The information is relayed to the MaT which needs to locate the device which is associated to the respective RFID tag. Therefore, it queries the Global Resource Directory (GRD) for the location of the Smart Things Information Service (STIS), a database-like service that keeps associations between RFID tags and devices. The MaT further sends back information to the maintenance app running at the mobile device providing the system engineer with more information about the device. With the help of this information, the engineer has the possibility to run diagnostics on the device. In this case, the CMS further acts as an intermediary between Maintenance Tool and the IoT6 Gateway, accepting and relaying messages from the Maintenance Tool to the IoT6 Gateway. In case the device's defectiveness is confirmed, a replacement order needs to be made. This order can again be performed using the MaT. The information previously retrieved from the STIS may in this case further be used to directly order the spare part from an Inventory Management System (IMS), another service in the IoT. If the address of the IMS is not yet known by the MaT, it has to first again issue a request to the GRD. Alternatively, the IMS may be part of the Maintenance Tool in which case the separation of the two services can be omitted.

2.3 IPv6 Business Case: Mobile phone as a sensing tool

One of the many possible business cases that could be deployed using the proposed architecture for IPv6 end point sourcing is given in this section. Specifically, this case demonstrates how data from the phones sensors could be accessed from the Internet and used for forming the bigger picture about the environment. A smartphone has a number of embedded sensors, like GPS, microphone, speaker, camera, light, etc., that could be used for environmental monitoring. For example, data gathered from many

different sound sensors on phones could provide information about the noise level in different parts of the city in order to form the noise level map.

In observed cases, a mobile device can have its own sensors (embedded) or different sensors can be connected wirelessly, for example via Bluetooth, when the mobile phone acts as a half-gateway for sensors from devices that do not support the IPv6 protocol, allowing them to be accessible via the IPv6 network. A phone, while on the IPv4 mobile network, does not have a static IP address and every time when the phone is switched off and on, it obtains new IP address from the network. On the other hand, if the phone is on the WiFi, through the IPv4 network, port forwarding on the local router must be provided. Here we demonstrate the usage of IPv6 addressing system that enables every IoT (Internet of Things) device to have a unique IP address which facilitates implementation by avoiding port forwarding. Communication with the mobile phones is done over the CoAP protocol, while the Digcovery system is used for the service discovery. The two set-ups are presented. In the first set-up, the smartphone is used as an end point that could be accessed directly through the IPv6 address. The REST CoAP server is used so every sensor could be accessed independently, through its own interface. CoAP is an application layer protocol designed to lower the complexity for the constrained networks but, also, to enable communication over the existing Internet infrastructure. The second set-up shows how a smartphone could be used as a half-gateway for non IP devices. In this way, access to the devices that use Bluetooth or Infrared, is provided. A smartphone application is responsible for registering phone's sensors into a Digcovery directory. Digcovery is introduced as a service discovery system on the IoT6 project. It has a CoAP interface built-in in order to enable communication with the constrained devices on the network edge. On low power devices it is too complicated or impossible to implement the DNS protocol, and the usage of a CoAP for discovery enables development of more distributed systems. It allows end devices to discover the services that they need.

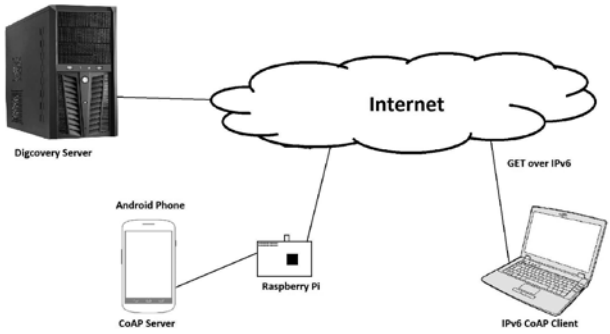

Figure 2.4: IPv6 communication between laptop and CoAP Server.

Another device (a laptop in this case) searches the directory for the required service. After receiving the required description, a client application on the laptop communicates with the phone and collects measurements from the sensors on the phone. An Android based smartphone was used for the implementation of the IPv6 CoAP server, a Raspberry Pi [22] was acting as an IPv6 border router and finally a laptop as an IPv6 client (Fig. 2.4). Raspberry Pi is basically a Linux machine and therefore, it could be set to be a router for the local network enabling Internet access to the local devices. Raspberry Pi is converted to be a WiFi hot spot for the IPv6 network. In this way, IPv6 enabled devices could obtain the IPv6 address through the Raspberry. A full /56 prefix is assigned to a Raspberry, enabling the distribution of IPv6 connectivity to an entire network. A DHCP server is built on the Raspberry which assigns unique IPv6 address to every device that tries to connect with it. A static IPv6 address, accessible from the web is assigned to the Raspberry Pi.

In the second set-up case, shown in Fig. 2.5, access and communication to the external device connected via Bluetooth with the phone is presented. In this set-up, the mobile phone acts as a half-gateway for sensors from devices that do not support IPv6 protocol or, as in this case, do not have an IP stack at all. These devices are connected to the phone via Bluetooth, Infrared, etc. Since IoT means connected devices via the Internet, it is crucial to show how these devices could have an Internet access over the IPv6 network. The role of half-gateway is to communicate through IPv6 but still be able to connect to a device via Bluetooth or Infrared. The mobile phone performs registration of these devices in Digcovery thus allowing their discovery and obtaining measurements. In the full gateway implementation, additional protocol adaptations, security and privacy aspects should be suported. In this setup an Android phone with CoAP Server implemented is used as an IPv6 half-gateway for the Bluetooth enabled device MindWave [23]. The MindWave device is able to read brain wave activity and to send raw measurements to the smartphone. As in the first test case, a Raspberry Pi is set-up as the Border Router for IPv6. A connection between the MindWave and the smartphone is established using Bluetooth. An application installed on the phone communicates over the IPv6 network, reads and processes the EEG (Electro Encephalograph) data from the MindWave and interprets it according to the level of attention and meditation. The same application has a CoAP server that waits for the request from the Internet.

With the presented business cases, we have demonstrated how sensors on mobile phones could be used for environmental monitoring, and how non-IP de-

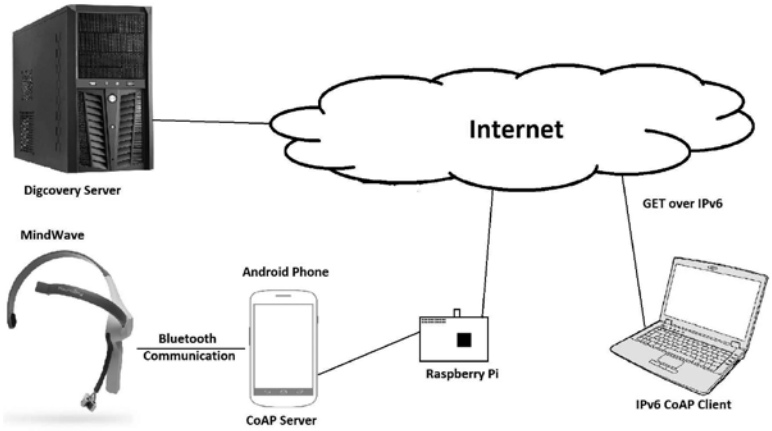

Figure 2.5: IPv6 communication between laptop and MindWave device.

vices could be connected in an IPv6 network. The CoAP protocol is used, since existing protocols on the application layer which operate according to a request-response model are not a good match for low-power, resource constrained devices. CoAP is a light-weight application protocol based on UDP that supports multicast requests, cashing and REST web services between the end-points, and is seen as a future protocol for IoT. Digcovery is a global discovery platform and is used for service discovery. This platform is used to locate the different domains and the wide deployed directories with their different resources. Raspberry Pi is acting as an IPv6 border router for the local network enabling Internet access to the local devices. It is converted to a WiFi hot spot for the IPv6 network, allowing IPv6 enabled devices to obtain IPv6 address through the Raspberry. As mentioned above, Raspberry Pi is operating under the Linux OS but it is also an embedded device with digital GPIO's (General Purpose Input/Output), that provides many opportunities. Raspberry Pi could have any kind of server built-in (CoAP or HTTP) with access to GPIOs with the possibility to control any device connected to the Raspberry. In that way, many of home and office devices could be controlled from the browser, desktop application or even smartphone application.

3

Practical steps: How to deploy IPv6 in an SME

3.1 How to set up IPv6?

Many small businesses operate their own networks, either for customer use or for internal operations. The extent to which their network needs to be upgraded will depend on their specific setup. To ensure that their business is not adversely affected, they should start by making sure they have an answer to each of the following questions:

- *Are you and your IT staff aware of IPv6?*
- *Is your Internet Service Provider ready to provide IPv6 connectivity?*
- *Is your own network equipment IPv6 compatible, and if not, what steps are necessary to make it IPv6 compatible?*
- *Have you considered IPv6-readiness in your technology upgrade cycle?*
- *Have you assessed IPv6-related risk and security and put the necessary measures in place?*

Based on the answers to these questions, SMEs will be able to plan the steps they need to take to ensure that IPv6 adoption does not see their business left behind. While everyone's IT networks are set up differently, one of the most common approaches to begin IPv6 deployment is to use the dual stack method. The latter implements IPv4 and IPv6 software either independently or in a hybrid form, allowing both IPv4 and IPv6 networks to work concurrently.

Where are you now? The first step is to assess where the SMEs are in relation to IPv6. Their Internet Service Provider may already offer IPv6 connectivity, they may already be operating hardware and software that is compatible with IPv6, and using IPv6 may be as simple as flipping a switch. It is more likely though, that deploying IPv6 will involve at least some investment of time and resources. The following checklist is a rough guide to how they may wish to proceed:

- *Appoint a project manager*
- *Talk to your Internet Service Provider*
- *Identify the network components that will need to be changed or upgraded*
- *Identify the training needs for team and project manager*
- *Determine costs of new hardware and software*
- *Select suppliers (possibly the same as you have today) and consultants*
- *Draft a project plan and start implementation*

Talk to your ISP. Most businesses rely on an Internet Service Provider (ISP) for their connection to the Internet. Their own network's IPv6 requirements and deployment schedule will be contingent upon their upstream provider's IPv6 deployment, so it is important that they understand what their ISP can provide and when. Some questions that they might ask their service provider are:

- *Do you currently provide IPv6 connectivity?*
- *If not, when do you plan to deploy IPv6 on your customer networks? What is your deployment timeline?*
- *When will our website be available over IPv6?*
- *Do you provide customers with IPv6-compatible modems, or other devices necessary for connecting over IPv6?*
- *Why have you not provided information for your customers on IPv6 and the requirements from the clients' side to be ready?*

Once an SME has agreed upon a plan of action with its ISP, they need to consider:(i) physical technology, (ii) software compatibility, and (iii) training.

IPv6-compatible Physical Technology. An essential part of making an SME's business IPv6 ready is to ensure that their equipment is compatible with the next generation of IP addressing. The first step is to carry out an IT audit to identify which pieces of equipment (routers, servers and other hardware) need upgrading or even replacing. Their hardware vendor(s) should be able to help them with this process, and advise them on how to make the necessary changes. It may require a significant amount of time and effort to convert all elements of their IT infrastructure, so they may want to consider a staged deployment.

There are 4 checkpoints to ensure that devices, security measures, network and other equipment are IPv6 compliant:

1. Internet-enabled devices: It is necessary to check if the currently used version of the operating system is up-to-date with IPv6 standards (the software manual should contain that information). In general, most current operating systems, including Linux, Windows Vista or above, and Apple Mac OS X or above, are IPv6-enabled by default.

2. Network devices: The devices used to connect to the Internet, such as routers, wireless modems, as well as the Customer Premises Equipment (CPE), including the device that is provided by your ISP, such as ADSL modem, VDSL modem and cable modem, must be IPv6-ready. SMEs should check with their software vendors and their ISP to make sure their devices' versions are able to manage IPv6 web and network traffic, as IPv6 firewalls might accidentally filter out important information.

3. ISP: The SMEs connection will be IPv6-enabled if their ISP is offering an IPv6 service. If they are not sure, they have to check directly with their ISP.

4. Hosting services: It has to be checked if equipment of the hosting service providers, including web and email servers, are IPv6 ready.

The IPv6 Forum is running the *IPv6 Ready Logo Program* to certify products that comply with IPv6 standards. It is highly recommended to buy IPv6-enabled products that bear the "IPv6 Ready" logo. Moreover, the IPv6 Forum maintains a list of IPv6-ready equipment that is useful to check, before preparing the shopping list of new devices to buy.

3.2 Enabling low-power IPv6-IoT networks with 6LoWPAN

In the Internet, a packet passes through many different interconnected networks on its way from source to destination. Thus, considering the link layer technology of each traversed network, there needs to be an "IP-over-X" specification to define how to transport IP packets. In many cases, to map the services required by the IP layer on the services provided by the lower layer (i.e, the link layer), the "IP-over-X" specification can introduce a (sub)layer of its own, often called adaptation layer [24]. Following the same strategy, in the process of shaping the IoT world, the IETF IPv6 over Low power WPAN (6LoWPAN) Working Group began in 2007 to work on specifications for transmitting IPv6 over IEEE 802.15.4 networks.

Typically, Low power WPANs are characterized by: small packet sizes, support for addresses with different lengths, low bandwidth, star and mesh topologies, battery supplied devices, low cost, large number of devices, unknown node positions, high unreliability, and long idle periods during which communications interfaces are turned off to save energy.

Given the aforementioned features, it is clear that the adoption of IPv6 on top of a Low power WPAN is not straightforward, but poses strong requirements for the optimization of this adaptation layer. For instance, due to the IPv6 default minimum MTU size (i.e., 1280 bytes), a non-fragmented IPv6 packet would be too large to fit in an IEEE 802.15.4 frame. Moreover, the overhead due to the 40 bytes long IPv6 header would waste the scarce bandwidth available at the PHY layer. For these reasons, the 6LoWPAN working group has devoted huge efforts in defining an effective adaptation layer in [25], [1]. Further issues encompass the auto-configuration of IPv6 addresses [26], the compliance with the recommendation on supporting linklayer subnet broadcast in shared networks [27] , the reduction of routing and management overhead, the adoption of lightweight application protocols (or novel data encoding techniques), and the support for security mechanisms (i.e., confidentiality and integrity protection, device bootstrapping, key establishment and management).

Routing issues are very challenging for 6LoWPAN, given the low-power and lossy radio-links, the battery supplied nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions should take into account the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms [24]. An effective solution is being developed by the IETF "Rout-

ing Over Low power and Lossy (ROLL) networks" working group. Recently, the IETF ROLL has proposed the leading IPv6 Routing Protocol for Low power and Lossy Networks (LLNs), RPL, based on a gradient based approach [28], [18]. RPL can support a wide variety of different link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with host or router devices with very limited resources, as in building/home automation, industrial environments, and urban applications [29] [30] [31] [32]. It is able to quickly build up network routes, to distribute routing knowledge among nodes, and to adapt the topology in a very efficient way. For these characteristics, it is suitable also for smart grid communications [33].

The introduction of the IETF 6LoWPAN protocol family has been instrumental in connecting the low power radios to the Internet and the work of IETF ROLL allowed suitable routing protocols to achieve universal connectivity. Indeed, both WGs enabled IPv6 connectivity which is a great asset in guaranteeing global reachability, true scalability, and reliable security.

3.3 Enabling DNS with IPv6

Domain Name System (DNS) has been designed to present a single, globally unique name space [34]. This property should be still maintained, when migrating from IPv4 to IPv6. The IP version used to transport the DNS queries and responses is independent of the DNS records used for representing the queried address [1]. Specifically, AAAA records can be queried over IPv4, and A records over IPv6. The DNS servers must not make any assumptions about what data to return for Answer and Authority sections based on the underlying transport used in a query. It is fundamental to avoid IPv4/IPv6 Name Space Fragmentation. To this aim, if some parts of DNS are only visible using IPv4 (or IPv6) transport, the best practice/recommendation is to always keep at least one authoritative server IPv4-enabled and IPv6- enabled, and to ensure that recursive DNS servers support IPv4 and IPv6.

[1] Note that in the forward zones IPv6 addresses are represented using AAAA records. In the reverse zones, IPv6 addresses are represented using PTR records in the nibble format under the ip6.arpa.tree [35], [36], [37]

3.3.1 DNS Considerations about Special IPv6 Addresses

While designing DNS with IPv6, some considerations about "special IPv6 addresses" have to be taken into account. The following IPv6 address types are considered special:

• *Limited-Scope Addresses.* The IPv6 addressing architecture [38] includes two kinds of local-use addresses: link-local (fe80::/10) and site-local (fec0::/10). The site-local addresses have been deprecated [39] and should never be published in the DNS. Link-local addresses should never be published in DNS (whether in forward or reverse tree), because they have only local (to the connected link) significance [40].

• *Temporary Addresses.* Temporary addresses defined in RFC 3041 [41] (sometimes called "privacy addresses") use a random number as the interface identifier. The DNS AAAA records should always be updated to reflect the current address assigned.

• *6to4 Addresses.* 6to4 [42] specifies an automatic tunneling mechanism that maps a public IPv4 address V4ADDR to an IPv6 prefix 2002:V4ADDR::/48. [43] aims to design an autonomous reverse delegation system that anyone being capable of communicating using a specific 6to4 address would be able to set up a reverse delegation to the corresponding 6to4 prefix. This could be deployed by, e.g., Regional Internet Registries (RIRs). This is a practical solution, but may have some scalability concerns.

• *Other Transition Mechanisms.* 6to4 is mentioned as a case of an IPv6 transition mechanism requiring special considerations. In general, mechanisms that include a special prefix may need a custom solution; otherwise, for example, when IPv4 address is embedded as the suffix or not embedded at all, special solutions are likely not needed.

3.3.2 Recommendations for Service Provisioning Using DNS

When names are added in the DNS to facilitate a service, there are several general guidelines to consider to be able to do it as smoothly as possible.

Use of Service Names instead of Node Names

It makes sense to keep information about separate services logically separate in the

DNS by using a different DNS hostname for each service. There are several reasons for doing this and they are as follows:

- Additional flexibility and ease for migration of (only a part of) services from one node to another,
- Configuring different properties (e.g., Time to Live (TTL)) for each service, and
- Deciding separately for each service whether or not to publish the IPv6 addresses (in cases where some services are more IPv6-ready than others).

Using SRV records [44] would avoid these problems. Unfortunately, those are not sufficiently widely used to be applicable in most cases. Hence, an operation technique is to use service names instead of node names (or "hostnames"). Note that this operational technique is not specific to IPv6. For example, assume a node named "pobox.example.com" provides both SMTP and IMAP service. Instead of configuring the MX records to point at "pobox.example.com", and configuring the mail clients to look up the mail via IMAP from "pobox.example.com", one could use, e.g., "smtp.example.com" for SMTP (for both message submission and mail relaying between SMTP servers) and "imap.example.com" for IMAP. Note that in the specific case of SMTP relaying, the server itself must typically also be configured to know all its names to ensure that loops do not occur. DNS can provide a layer of indirection between service names and where the service actually is, and using which addresses. Obviously, when wanting to reach a specific node, one should use the hostname rather than a service name.

Separate vs. the Same Service Names for IPv4 and IPv6

The service naming can be achieved in basically two ways: when a service is named "service.example.com" for IPv4, the IPv6-enabled service could either be added to "service.example.com" or added separately under a different name, e.g., in a sub-domain like "service.ipv6.example.com". These two methods have different features. Using a different name allows for easier service piloting, minimizing the disturbance to the "regular" users of IPv4 service; however, the service would not be transparent, without the user/application explicitly finding it and asking for it, which would be a disadvantage in most cases. When the different name is under a sub-domain, and the services are deployed within a restricted network (e.g., inside an enterprise), at

least to a degree, it is possible to see the services transparently, by modifying the DNS search path; however, this is a suboptimal solution. Using the same service name is the "long-term" solution, but may degrade performance for those clients whose IPv6 performance is lower than IPv4.

In most cases, it makes sense to pilot or test a service using separate service names, and move to the use of the same name when confident enough that the service level will not degrade for the users unaware of IPv6.

3.4 Enabling a Mail Server with IPv6

Enabling IPv6 on mail servers is both easy and useful to do. The largest email server (gmail.com) has been using IPv6 since 2012 without any problem. Actually, mail servers will be dual-stack for many years to come, and thus, be able to send and receive emails over IPv4 and IPv6 by using the same protocols suite: SMTP, IMAP and POP (hopefully secured by the use of TLS encryption and authentication). Given that email messages are not really interactive, there is no real issue regarding the choice of IPv4 or IPv6 for latency. In the worst case, there will be a difference of a tenth of a second that is meaningless and invisible. All email systems support a dual-stack deployment, from the open source postfix and send mail to Microsoft Exchange. The 2013 version of the latter even refuses to install on a system where IPv6 has been disabled. Depending on the specific system in use, the IPv6 may be not enabled by default. But, the configuration to support IPv6 is quite easy. For instance, for postfix, a single file (*/etc/postf ix/mail.cf*) should be updated by adding/changing a single line (*inetprotocols = all*). As soon as IPv6 support is enabled on an email server, all sent emails will be sent over IPv6 or IPv4 depending on the protocol version supported by the receiving or relaying parties.

But, how does an email server decide to send over IPv6 or over the legacy IPv4? In all cases, the email address, user@example.org, is exactly the same because they are independent of the underlying transfer protocol. The answer lies in the Domain Name System (DNS) which is used to find the IP address associated to an email domain (i.e., domain @example.org. in the considered example). There is a specific DNS request to enquire about the IP addresses of a mail server, and its Mail Exchange (MX), i.e., "Which email server do I need to contact in order to send an email?". If the associated email server has an IPv6 address, then this address will be used; respectively,

if the associated email server has also an IPv4 address, then this IPv4 address can also be used. While sending email over IPv6 can be on by default as soon as the email server has IPv6 connectivity, there is also another operation to be done in order to receive email over IPv6, i.e., to update the DNS.

However, nowadays email is not only about sending and receiving messages; it is also to ensure virus-free and spam-free content! Because, all messages (good and bad) can be sent either over IPv6 or over the legacy IPv4, spam and viruses are obviously also sent and received over IPv6. The good news is that all anti-virus systems also work over IPv6 as they do over IPv4. The same applies for anti-spam systems (both commercial and open source). The only temporary caveat is that in 2014 some anti-spam systems rely on the sender IP address to give a spam probability. This reputation is simply a database or a registry giving a spam score to IP addresses and while the IPv4 database is well established for many years, the IPv6 one is still being built and is currently and momentarily less reliable. This issue will be slowly fixed by itself as more email messages are exchanged over IPv6, the more reliable the database will become. Finally, to build a secure email system a message authentication or a signature to prove the origin of an email message is needed. Application level signatures (PGP, Outlook, etc.) are independent of the transfer protocol and works as well for IPv4 as for IPv6. The network level origin authentication (SPF, DKIM, DMARC, etc.) relies on information in the DNS system, and the DNS entries for @example.org, must also be updated.

3.5 Tunneling for providing IPv6 connectivity

In many situations, using native IPv6 for connecting devices is not an option. This does not prevent the use of IPv6 for the actual device communication but there will be a need for a tunneling solution to provide an IPv6 overlay. There is a multitude of different tunneling based transition protocols to choose from and depending on the specific use case different protocols will have different pros and cons. This means that there is not one single way of providing IPv6 for IoT using tunneling instead there is a toolbox of solutions available to pick from and the preferred solution will vary from case to case.

Two widely deployed tunneling protocols are 6to4 [42] and 6RD [45]. The two protocols are very similar and provide a peer to peer capable tunneling overlay

for IPv6 for connecting a large number of devices. It is done by encapsulating IPv6 in IPv4 and embedding the IPv4 address in the IPv6 address. The only issue with 6to4 and 6RD is that they require a public IPv4 address to function, which means that it is not suitable in scenarios where it is included in devices that will be deployed behind a NAT. Thus, in order to use 6to4 or 6RD in a residential user's home network, the home router/NAT would have to be upgraded to support the protocol. This would enable IPv6 for all devices within the home network, and not just the IoT devices. Unlike 6to4, 6RD creates a private tunneling overlay. This mitigates the biggest concern with using 6to4, i.e., the asymmetrical routing of traffic that goes to and from 6to4 devices and native IPv6 devices. The traffic relies on relays that are different depending on which direction the traffic is going and there is no way of managing which relays will be used. That means performance will not be predictable and might vary depending on which native IPv6 device the 6to4 device is communicating with. Since 6RD creates a private domain the network administrator can control where the traffic is routed in the different directions, and even though it is still asymmetric, it is predictable. To the outside world, the 6RD device will look like native IPv6 device unlike the 6to4 ones which use a distinct IPv6 prefix to identify themselves. It should be noted that 6to4 can be used with no additional infrastructure today, as it is a global transition protocol, while 6RD requires a full deployment in order to function, which might be a limiting factor for a small IoT deployment. Even though 6to4 together with 6RD are the prevailing tunneling transition protocols, they might not be the most useful in many IoT scenarios as devices are likely to be behind a NAT. Instead, protocols like Teredo [46] and TSP [47] might be better suited for providing IPv6 when deploying devices in a multitude of networks.

Teredo provides a peer to peer capable overlay, very much like 6to4, but since it is designed to function behind NATs it is more complex. It requires a Teredo server for discovery of the connection even when the communication takes place between Teredo hosts and relay connected with native IPv6 hosts. As Teredo encapsulates the IPv6 traffic in UDP IPv4, it works with the majority of NATs, but unfortunately not all. Teredo is being slowly removed from Microsoft Windows due to complaints about it creating network security issues (by creating public connections when behind a NAT) and thus, it is no longer considered a preferred IPv6 migration protocol. Despite that, Teredo can be an excellent tool for deploying devices in a multitude of networks and providing connectivity for them. For instance, Xbox One [48], uses a dedicated Teredo deployment to provide peer-to-peer connectivity between the dif-

ferent consoles. If the Xbox One has a functioning IPv6 connection it will switch to native IPv6, otherwise it uses Teredo for its connection. The same approach could be taken for an IoT deployment that mainly communicate in a closed group.

TSP and other tunneling protocols without peer-to-peer support, such as L2TP Softwire [49], are always a safe alternative to the more dynamic peer-to-peer tunneling protocols. TSP provides a point to multipoint tunneling solutions that works in any network scenario (unless it is explicitly blocked). The main downside is that all traffic needs to pass through a single point, the tunnel server. In some use cases where the traffic volume is not big, this might not be an issue, but it can become a problem if the traffic volume is high and a lot of the traffic is peer-to-peer. It is important to note that if most of the traffic goes from the devices to the Internet, the demand on the tunnel servers will be equivalent to the demand on the relays in the other scenarios. Although when using 6to4 and Teredo, you must rely on the relays provided by other parties. TSP and L2TP do not come with a public infrastructure and will require deployment of servers to function, in the same way as 6RD. This is also why they are more predictable than the other tunneling options as the whole solution is managed and controlled by one entity.

As there is no obvious choice of tunneling solution for providing IPv6 for IoT, the choice must be based on the specific use case. Pros and cons of the different protocols have to be weighted in order to decide on which solution to use. In some cases, a combination of protocols might be preferential. 6to4 and 6RD provides a great performing peer-to-peer capable solution when public IPv4 is available. Teredo gives you a peer-to-peer capable alternative behind NAT and, finally TSP and L2TP provides a reliable tunneling option for any type of network but without peer-to-peer support.

3.6 Enabling a web server with IPv6

Enabling IPv6 on a Website is a very simple process, assuming the underlying operating system supports IPv6 and the user has IPv6 (either tunneled or direct) on his/her service.

Virtually all web server software supports IPv6 today. The top 5 web servers (Apache, Nginx, MS IIS, LiteSpeed and Google Servers) all have supported IPv6 for years. Most Operating Systems used to host Web Servers have also supported

IPv6 for years, including Linux, FreeBSD and its derivatives, real AT&T Unix, and Windows Server, since 2008 (for those that are still using Windows Server 2003, it is definitely time to upgrade since its IPv6 support is based on Windows XP, and it is not very good).

Most web development languages (PHP, ASP.NET and Java being the leaders) not only support IPv6, in most cases once the underlying web server is running IPv6, virtually all web apps written for IPv4 just work with IPv6 with no modifications required. For example, the term "IPv6" cannot be found anywhere on www.joomla.org, yet Joomla (which is written in PHP) is 100% functional over IPv6.

There are some colo and web hosting facilities that can host websites over both IPv4 and IPv6. It is enough to search for "IPv6 web hosting" to find them. It is highly recommended to use them. For $600 a month it is possible to get a 72U rack, with guaranteed power and 100 Mbit symmetric dual stack service (including a /24 block of IPv4 and a /48 block of IPv6). Just hosting dual stack websites is usually about the same price as IPv4-only.

It is possible to publish the necessary A and AAAA records in DNS even on authoritative DNS servers that have only IPv4 connectivity. However, it is recommend using dual stack DNS. There are a number of hosted DNS services that provide support for IPv6 (e.g., GoDaddy hosted DNS). A domain registrar that allow to register both IPv4 and IPv6 addresses of the DNS servers with the TLD servers should be used [2].

It is also possible to "cheat" by deploying a reverse translating web proxy in front of a legacy IPv4-only website (which can even be in a legacy IPv4-only network) that can "up-convert" the legacy site into full dual stack glory. Such a reverse proxy can be built using Apache or Nginx, or purchased as an appliance designed to do just this, with very simple configuration. The reverse proxy itself must have access to both IPv4 and IPv6, but can be deployed in a dual stack colo. Typically, IPv4 goes directly to the real web server, but IPv6 is directed to the reverse proxy, which down-converts to IPv4 and sends it to the legacy server, then up-converts the response to IPv6 which is returned to the IPv6 browser. This is a really quick and easy way to make a website dual stack, with very little effort or cost. It can be fully done (from start to end) in about one day. All IPv6 traffic will appear to the application to be coming from the IPv4 address of the reverse proxy. It is also possible to make an IPv6-only web server

[2] GoDaddy provides this for most TLDs they support

available also over IPv4 using a reverse proxy. In this case, the IPv6 goes directly to the real web server, and IPv4 connections get routed to the reverse proxy. There is a do-it-yourself project to deploy a dual stack web server on *www.sixscape.com*, using open source software and free tunneled service from Hurricane Electric. This includes the minor changes in web server configuration required to add IPv6 support for Apache.

3.7 Enabling Security with IPv6

The IPv6 protocol has been designed to address some of the security problems found in IPv4, but it is has to be noticed that not all the security issues have been solved, and thus appropriate countermeasures should be taken.

First of all, since most organizations cannot change all their networks to IPv6 overnight, IPv6 will be gradually deployed while IPv4 is supported for legacy clients and services. This presents a challenge, since a dual protocol environment increases the complexity and potentially also security risk.

The U.S. Department of Commerce's National Institute of Standards and Technology has made its Guidelines for the Secure Deployment of IPv6 and it downloadable to the public.

Here are some of the best practices that SMEs should take in building and maintaining secure IPv6 networks:

- Use standard, static addresses for critical systems;
- Ensure adequate filtering capabilities for IPv6;
- Filter internal-use IPv6 addresses at border routers;
- Block all IPv6 traffic on IPv4-only networks;
- Filter unnecessary services at the firewall level;
- Pay close attention to the security aspects of inter-protocol transition mechanisms.

In addition, it is critical to note that IPv6 cannot prevent certain attacks such as: viruses and malicious code, brute-force attacks and password guessing, Denial of Service (DoS) attacks, spamming, phishing, among others. In the following, we outline some of the main threats, and potential countermeasures that should be taken.

3.7.1 Neighbor discovery threats

Faked advertisement. The neighbor discovery protocol allows a host to find another host at the link layer. It is equivalent to the ARP protocol for IPv4. So, when a node A wants to connect to a node B, it has to send a Neighbor Solicitation message which should be replied by node B. However, any attacker on the same link can send a forged reply to spoof the IPv6 address of node B. Thus, the attacker can act as the man in the middle. This can cause serious issues in particular when node B is a router.

Counter-measures:

- SEND (Secure Neighbor Discovery) relies on a cryptographic approach to avoid spoofing but is difficult to deploy as every nodes has to integrate it [50].

- NDPMon (http://ndpmon.sourceforge.net/) monitors neighbor discovery protocol messages to detect misbehavior. Acting as a monitor, it can be instantiated on a single node.

- The use of encryption, like a PKI is recommended to prevent information leakage. By doing so, the attacker is unable to decrypt the messages which will be finally forwarded from node A to node B.

Router (Distributed) Denial of Service. Assuming the attacker knows the prefix announced by the router, he may send a lot of packets towards existing or non existing IP addresses (using a single or multiple machines like a bootnet) in this prefix forcing the router to trigger the neighbor discovery protocol to locate them at the link level. By doing this, resources of the router are consumed and so it might not serve properly its other tasks, in particular auto-configuration and outing.

Counter-measures:

- Limit neigbor discovery messages exchanged to keep enough resources for other tasks.
- Monitoring such a behavior using an IDS.

Faked Neighbor Solicitation. When a host sends a neighbor solicitation message, it includes its own link layer address which so allows the other hosts to cache it. However, an attacker can also send many neighbor solicitations with different address which will thus creates a lot of false entries in the cache of the other hosts.

Counter-measures:

- Monitor when a host pretends to be a different addresses (or at least too many).

IPv6 addresses DoS using DAD. The DAD protocol allows a host to verify that the chosen address by the autoconfiguration is not used yet by another host. To do so, host A has to send an neighbor solicitation message with this address and wait if there is a reply. Even if not used by any other ones, an attacker can reply to the neighbor solicitation such that A cannot be configured. This can be repeated for all neighbor solicitation messages to prevent all hosts of the network to be properly configured.

Counter-measures:

- The attack is quite aggressive when the attacker replies on all neighbor solicitation which thus can be easily discovered by an IDS.
- The SEND protocol can be used [50].

3.7.2 DHCP related threats

If using stateless address auto-configuration is not possible, DHCP has been adapted to work with IPv6 (DHCPv6). Attacking such a service may completely disturb IP address allocation.

Impersonation of DHCP server. As DHCP servers are usually not authenticated, an attacker can impersonate it and therefore, highly impacts the configuration of the hosts, for example the default DNS servers or the gateway.

Counter-measures:

- Use authentication

Information gathering. As DHCP client are usually not authenticated, an attacker can contact the DHCP server to obtain information about the configuration like the DNS server. Although this is not seen as an attack in itself, this helps the attacker to retrieve useful information in order to plan an attack.

Counter-measures:

- Use authentication
- Guarantee a high level of security on the different hosts to prevent such cases of attacks.

DoS Attacks. DoS or DdoS attacks represent a vast category of threats including, in particular, flooding attacks which are easy to perform by sending huge volume of messages.

Counter-measure:

- Monitor the network
  - Firewalling: A standard firewall like ip6tables with some add-ons (xtables-addon) allows to rapidly filter bad behavior or in the advised case only allowing predefined traffic (block by default).
  - More advanced monitoring systems like IDS are encouraged to be used. In IoT6, we extended MaM which is an open source tool which provides the basic block for monitoring the network in a scalable way. Metrics can be then derived from it.

3.8 Integrating legacy devices

One of the challenges for the future Internet of Things is related to its inherent heterogeneity. Hundreds of communication protocols have emerged to address specific requirements. Interconnection of things implies to deal with huge amount of different technologies and then with their different protocols. Some technologies were developed with IP capabilities; others used different networking technologies, with open or proprietary buses. Over time, part of those protocols may move towards IP. However, existing systems are likely to remain and quite a number of communication protocols will keep their specific bus technology. The integration of heterogeneous

Internet of Things components faces several challenges, including:

- Integrating non-IP-based communication protocols into an IP-based environment
- Integrating together communication protocols using different application layers.

During this time, different solutions have been researched and developed:

1. Bridges and gateways
   The first and most natural integration scheme has been to develop bridges and gateways enabling the translation of a communication protocol into another one. It enables the integration of distinct protocols into IPv6 and vice-versa. Such gateways usually provide a clear IP-based API to communicate with the devices and its specific communication protocol.

2. IP adaptation
   Several communication protocols have moved a step farther by developing IP-based versions of their own protocols. This option has been largely developed in the building automation domain, with protocols such as the KNX Association, which has standardized a KNX IP version of its standard.

3. Universal Device Gateway
   The Universal Device Gateway (UDG) [15] is a multi-protocol control and monitoring system developed by a research project initiated in Switzerland in 2006. It aimed at integrating heterogeneous communication protocols into IPv6. The UDG control and monitoring system enables cross protocol interoperability. It demonstrated the potential of IPv6 to support the integration among various communication protocols and devices, such as KNX, X10, ZigBee, GSM/GPRS, Bluetooth, and RFID tags. It provides connected device with a unique IPv6 address that serves as unique identifier for that object, regardless of its native communication protocol. It has been used in several research projects, including IoT6, where it has been used among others as an IPv6 and CoAP proxy for all kinds of devices.

4. IoT6 stack and IoTSyS
   IoT6 has defined a clear IoT6 stack based on IPv6 (or 6LoWPAN in constrained networks), CoAP, JSON and oBIX. This stack has been used to interface and

integrate the various IoT6 components together. In order to test the integration of legacy protocols, the IoT6 research project has developed IoTSyS, a prototype of a Java based integration middleware abstracting the low level protocol details through the IoT6 stack to allow the communications with the other components of the IoT6 framework and vice-versa [51]. This prototype was used to test and demonstrate the interconnectivity among several protocols such as BACnet, KNX, ZigBee, etc.

IoT6 has confirmed the capacity of those various approaches to integrate heterogeneous communication protocols and devices together through IPv6. While traditional approaches require multiplying the number of bridges for each couple of communication protocols, the two latter solutions enable a simplification of the network extension to additional standards. Moreover, they are easily portable and deployable in constrained environments.

5. IPv6 Address mapping

Beyond the interconnection and interoperability mechanisms, another issue has been addressed by IoT6: the possibility to map IPv6 addresses on top of other addressing schemes, from non-IP communication protocols. Part of the challenge of integrating legacy technologies into an IPv6 network is represented by devising a mechanism for stateless auto configuration of such devices. Indeed such a mechanism would ensure that a number of properties of the mapping hold, such as, for example:

- Consistency: a host should get the same IPv6 address every time it connects to a same legacy network. This feature might be particularly important for devices which are not always "on", or which are not permanently connected
- Local Uniqueness: for devices which have an IPv6 address with a same network part, the host part should be unique for each host. This property avoids address's conflicts within a same subnet.
- Uniqueness within the whole Internet: coherently with the IoT vision, the host part of an IPv6 address associated to a host should be unique within the whole Internet.

This effort within the IoT6 project has produced a proposal for a new standard for IPv6 address mapping of non-IP-based communication protocols, currently

in the form of an IETF draft. The proposed solution named 6TONon-IP provides a clear specification of a mapping mechanism which tries to maximize the satisfaction of the properties mentioned. The gateways provide through the IPv6 address mapping solution, the IPv6 addresses to the objects they manage, using a semantic to identify and differentiate the protocols. Two solutions were deployed to address this challenge and each one designed its own internal semantics.

4

Conclusion

The Internet of Things is a vivid sector with still many innovations to come. However, by analyzing the recent evolution, we can clearly anticipate a strong convergence between the Internet of Thing and IPv6. As highlighted by the handbook, IPv6 provides an ideal solution to provide each and every smart thing with its own public address, which is fully Internet compliant. IPv6 will provide the Internet of Things with a strong scalability and interoperability enabler, coupled with a secured and reliable technology. The handbook has summarized the current state of the art and sketched some promising potentialities. We hope all readers will find it useful and interesting.

It is important to highlight that this handbook is the result of a collective work and effort. It gathers different and complementary views on this topic. We thank all those who have contributed to its writing, editing and realization, as well as the European Commission, which is actively supporting the research on the Internet of Things, including through the IoT6 project.

The IoT6 consortium will continue researching and working on this topic in the coming years. If you are interested to work with us on any practical IPv6-based deployment of the Internet of Things and/or smart cities, feel free to contact us at:

IoT6 Research project
c/o Mandat International
iot6@mandint.org
http://www.iot6.eu
Sebastien Ziegler,
IoT6 Project Coordinator

Finally, we wish you a successful journey across the upcoming IPv6- enabled Internet of Things and its unlimited application domains.

5

Glossary

In order to help the reader, we provide a non-exhaustive list of the (main) acronyms used in the book.

Name	Description
3GPP	3rd Generation partnership project
6LoWPAN	Low Power adaption layer over wireless networks
6RD	IPv6 Rapid Deployment
6to4	Internet Protocol Version 6 to Version 4
6TONon-IP	IPv6 mapping to non-IP protocols
AH	Authentication Header
AMI	Advanced Metering Infrastructure
APN	Access Point Name
APNIC	Asia Pacific Network Information Centre
BACnet	Data Communication Protocol for Control Networks
BEMS	Building Energy Management Server
CMS	Control and Monitoring System
COAP	Internet of Things European Research Cluster
CoRE	Constrained RESTful Environments
CPE	Customer Premises Equipment
DHCP	Dynamic Host Configuration Protocol
DHCP	Dynamic Host Configuration Protocol
DiffServ	Packet Prioritisation at entry
DNS	Domain Name System

DoS	Denial of Service
EEG	Electro Encephalograph
EnOcean	Energy-harvesting wireless sensor technology
EPC	Electronic Product Code
EPCSN	EPC Sensor Network
ESP	Encapsulating Security Payload
ETSI	European Telecommunications Standards Institute
EUI64	Extended Universal Identifier of 64-bit format
GRD	Global Resource Directory
HC	Header Compression for LoWPANs
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMPv6	Internet Control Message Protocol for IPv6
IEEE	Institute of Electrical and Electronics Engineers
IERC	Internet of Things European Research Cluster
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IMS	Inventory Management System
IntServ	Packet Prioritisation through Network Throughput
IoT	Internet of Things
IoTSyS	Internet of Things System for Vienna University
IPSec	Security Protocol for IPv6
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISPs	Internet Service Providers
ITU-T	International Union for Telecommunication Standardization
JCA-IoT	Joint Coordination Activity on Internet of Things
JSON	JavaScript Object Notation
KNX	OSI-based network communications protocol for buildings
L2TP	Layer 2 Tunneling Protocol
LLN	Low power and Lossy Networks
LTE	Long Term Evolution
M2M	Machine to machine
MAC	Medium Acess Control
MANEMO	Mobile Ad-hoc Network Mobility

MaT	Maintenance Tool
MIP	Mobile Internet Protocol
MTU	Maximum Transmission Unit
NAT	Network address translation
ND	Neighbor Discovery
NFC	Near Field Communication
OBIX	Open Building Information Exchange
OSGI	Open Services Gateway Initiative
POP	Post Office Protocol
QoS	Quality of Service
QR	Quick Response Code
RD	Resource Discovery
REST	Representational state transfer
RFID	Radio-frequency identification
RIRs	Regional Internet Registries
ROLL	Routing Over Low power and Lossy
SaaS	Software as a Service (SaaS)
SLAAC	StateLess Address AutoConfiguration
SME	Small Medium Enterprise
SMTP	Simple Mail Transfer Protocol
STIS	Tags and Smart Things Information Services
StS	Safety Server
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDG	Universal Device Gateway
Webservices	HTTP
WPAN	Wide Personal Area Network
ZigBee	Low-cost, low-power, wireless network standard

6

References

Bibliography

[1] E. J. Hui and P. Thubert, "RFC6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-based networks," http://tools.ietf.org/html/rfc6282.

[2] C. Bormann, A. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," in *Internet Computing*, IEEE, 2012, pp. 62–67.

[3] N. K. Giang, S. Kim, D. Kim, M. Jung, and W. Kastner, "Extending the EPCIS with Building Automation Systems: a New Information System for the Internet of Things," in *International Workshop on Extending Seamlessly to the Internet of Things (esIoT, Birmingham, UK)*, 2014.

[4] Darpa, "RFC791: Internet Protocol," 1981, http://tools.ietf.org/html/rfc791.

[5] H. Chaouchi, The Internet of Things: Connecting Objects. John Wiley & Sons, 2010.

[6] S. Deering and R. Hinden, "RFC2460: Internet Protocol, version 6 (IPv6)," 1998, http://tools.ietf.org/html/rfc2460.

[7] IEEE, "802.15.4g-2012 - IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Net- works (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks," http://standards.ieee.org/findstds/standard/802.15.4g-2012.html.

[8] Google, "Google Statistic: IPv6 Adoption Rate by Country," http://www.google.com/intl/en/ipv6/statistics.html.

[9] Fielding and R. Thomas, *Architectural Styles and the Design of Network-Based Software Architectures.* University of California, Irvine, 2000.

[10] LteWorld, "LTE Operators," http://lteworld.org/operator.

[11] GS1, "EPC Global EPC Tag Data Standard (TDS) v1.6," 2011.

[12]——, "EPC global network," http://www.gs1.org/.

[13] OASIS, "oBIX: Open Building Information Exchange," http://www.obix.org/.

[14] "IoTSys available online at:," http://code.google.com/p/iotsys.

[15] DeviceGateway, "Universal Device Gateway," http://www.devicegateway.com.

[16] IETF, "Routing Over Low Power and Lossy Networks (roll)," http://datatracker.ietf.org/wg/roll/.

[17] ——, "Constrained RESTful Environments (core)," http://datatracker.ietf.org/wg/core/.

[18] E. T. Winter, E. P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," http://tools.ietf.org/html/rfc6550.

[19] IEEE, "802.15.4-2011 -IEEE Standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2011.

[20] ——, "802.15.4e-2012 - IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," https://standards.ieee.org/findstds/standard/802.15.4e-2012.html.

[21] IETF, "6TiSCH: "IPv6 over the TSCH mode of IEEE 802.15.4e", "http://datatracker.ietf.org/wg/6tisch/.

[22] www.raspberrypi.org, "Raspberry Pi Computer," http://www.raspberrypi.org/.

[23] NeuroSky, "MindWave," http://store.neurosky.com/products/mindwave1.

[24] Z. Shelby and C. Bormann, 6LoWPAN: *The Wireless Embedded Internet)*. Willey, 2009.

[25] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 networks," 2007, http://tools.ietf.org/html/rfc4944.

[26] M. Crawford, "Transmission of IPv6 Packets over Ethernet Networks," 1998, http://tools.ietf.org/html/rfc2464.

[27] P. Karn, C. Bormann, G. Fairhurst, D. Grossman, R. Ludwig, J.Mahdavi, G. Montenegro, J. Touch, and L. Wood, "Advice for Internet Subnetwork Designers," 2004, http://tools.ietf.org/html/rfc3819.

[28] K. Jeonggil, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, "Connecting Low-Power and Lossy Networks to the Internet," IEEE *Communications Magazine*, vol. 49, pp. 96–101, 2011.

[29] J. Martocci, "Building Automation Routing Requirements in Low- Power and Lossy Networks," 2010, http://tools.ietf.org/html/rfc5867.

[30] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks," 2010, http://tools.ietf.org/html/rfc5826.

[31] K. S. J. Pister and P. Thubert, "Industrial Routing Re- quirements in Low-Power and Lossy Networks," 2009, http://tools.ietf.org/html/rfc5673.

[32] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks," 2009, http://tools.ietf.org/html/rfc5548.

[33] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi, "The Deployment of a Smart Monitoring System Using Wireless Sensor and Actuator Networks," in *Proceeding of the First IEEE International Smart Grid Communications Conference, SmartGridComm,* 2010.

[34] Internet Architecture Board, "IAB Technical Comment on the Unique DNS Root," 2000, http://tools.ietf.org/html/rfc2826.

[35] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS Extensions to Support IP Version 6," 2003, http://tools.ietf.org/html/rfc3596.

[36] R. Bush, A. Durand, B. Fink, O. Gudmundsson, and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)," 2002, http://tools.ietf.org/html/rfc3363.

[37] R. Bush, "Delegation of IP6.ARPA," 2001, http://tools.ietf.org/html/rfc3152.

[38] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," 2006, http://tools.ietf.org/html/rfc4291.

[39] C. Huitema and B. Carpenter, "Deprecating Site Local Addresses," 2004, http://tools.ietf.org/html/rfc3879.

[40] A. Durand and T. Chown, "To Publish, or not to Publish, that is the Question," 2005.

[41] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," 2007, http://tools.ietf.org/html/rfc4941

[42] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," 2001, http://tools.ietf.org/html/rfc3056.

[43] G. Huston, "6to4 Reverse DNS Delegation Specification," 2005.

[44] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for Specifying the Location of Services (DNS SRV)," 2000, http://tools.ietf.org/html/rfc2782.

[45] R. Despres, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)," 2010, http://tools.ietf.org/html/rfc5569.

[46] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," 2006, http://tools.ietf.org/html/rfc4380.

[47] M. Blanchet, Viagenie, and F. Parent, "IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)," 2010, http://tools.ietf.org/html/rfc5572.

[48] Microsoft, "Xbox One Technical Details," 2012, http://download.microsoft.com/download/A/C/4/AC4484B8-AA16- 446F-86F8-BDFC498F8732/Xbox One Technical Details.docx.

[49] B. Storer, E. C. Pignataro, M. D. Santos, E. B. Stevant, L. Toutain, and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)," 2009, http://tools.ietf.org/html/rfc5571.

[50] S. Hogg and E. Vyncke, "IPv6 Security," *Cisco Press*, 2009.

[51] M. Jung, J. Weidinger, C. Reinisch, W. Kastner, C. Crettaz, A. Olivieri, and Y. Bocchi, "A Transparent IPv6 Multi-Protocol Gateway to Inte- grate Building Automation Systems in the Internet of Things," in *Pro- ceedings of the IEEE International Conference on Internet of Things* (*iThings 2012, Besancon, France*), 2012.

6.1 Useful web sites and tools

IoT6 website: *www.iot6.eu*
6 Deploy: *www.6deploy.eu*
Bluetooth technology: *www.bluetooth.com*
Copper CoAP for Firefox: *http://people.inf.ethz.ch/mkovatsc/copper.php*
Device Gateway – multi protocol control and monitoring system: *www.devicegateway.com*
DNS-SD: *www.dns-sd.org*
ETSI: *www.etsi.org*
European Commission page on IoT: *https://ec.europa.eu/digital-agenda/en/internet-things*
European Commission page on IPv6: *http://ec.europa.eu/information_society/policy/ipv6/index_en.htm*
FI-WARE: *www.fi-ware.org*
Fosstrak, open source EPCSN implementation: *www.fosstrak.org*
Google Statistic: IPv6 Adoption Rate by Country: *http://www.google.com/intl/en/ipv6/statistics.html.*
GS1 Global: *www.gs1.org*
Hobnet: *www.hobnet-project.eu*
ICANN: *www.icann.org*
IETF - 6lo: *https://datatracker.ietf.org/wg/6lo/charter/*
IETF - 6LoWPAN: *http://tools.ietf.org/search/rfc6282*
IETF - 6TiSCH: *http://datatracker.ietf.org/wg/6tisch/charter/*
IETF - CoAP: *https://datatracker.ietf.org/doc/draft-ietf-core-coap/*
IERC – European Research Cluster on IoT: *www.internet-of-things-research.eu*
Internet Society: www.internetsociety.org
IoT-A (Intranet of Things-Architecture) project: *www.iot-a.eu*
IoT Blog: *www.webofthings.org*
IoT Conference: *www.iot-conference.org*
IoT Forum: *www.iotforum.org*
IoT Lab: *www.iotlab.eu*
IoT platforms: *https://xively.com, https://thingspeak.com, https://sen.se*
IoT stack: *www.evrythng.com*
IoT OS: *www.contiki-os.org*
IoT Sys: *http://code.google.com/p/iotsys*

IPSO Alliance: *www.ipso-alliance.org*
IPv6 Forum: *www.ipv6forum.com*
IPv6 Traffic and Mobile Networks Stats by Cisco: *http://6lab.cisco.com/stats/*
IPv6 Observatory: *www.ipv6observatory.eu*
KNX Association: *www.knx.org*
Open Building Information Exchange: *www.obix.org*
Open source DNS-SD implementation: *http://jmdns.sourceforge.net*
Raspberry Pi Computer: *www.raspberrypi.org*
Rifidi open source RFID middle ware: *www.transcends.co*
Rifidi RFID emulator and other tools: *www.rifidi.org*
Run My Process SaaS Cloud: *www.runmyprocess.com*
Tracking the IoT: *http://postscapes.com*
Turn It IPv6: *www.turnitipv6.com*
Web services for devices: *http://ws4d.e-technik.uni-rostock.de*
ZigBee Alliance: *www.zigbee.org*

Acknowledgements

We would like to thank the people who have collaborated with us on this book. Special acknowledgment goes to Bill Manning, Staff Researcher at USC ISI [1], Eric Vyncke, Co-chair of the IPv6 Council of Belgium, Distinguished System Engineer at Cisco Systems [2], Adjunct Professor at the University of Liege, and winner of the Jim Bound IPv6 Deployment Award 2014, Mikael Lind, CTO at gogo6 [3], and Lawrence E. Hughes, CTO, Sixscape Communications Ltd. Given their great experience and deep knowledge of the IPv6 technology, their contribution has provided tremendous value to the book.

A special thanks goes also to Srdjan Krco for having been always ready to give his help for this work, and for having taken care of the layout of the book's cover, and the entire print out process.

We would like to thank our colleagues in the IoT6 consortium that have provided their support, during the preparation of the manuscript Jerome Francois, Lou Fedon, Sebastien Gaide, Alex C. Olivieri, Gianluca Rizzo, Antonio Skarmeta, Cedric Crettaz, and Peter Kirstein.

Finally, we would like to thank Martin Potts, and Alicia Higa for their dedication and effort in reviewing the final manuscript.

[1] http://icannwiki.com/index.php/BillManning

[2] http://www.ipv6council.be/

[3] https://www.linkedin.com/in/mikaelipv6

The IoT6 project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n. 288445.

For further information:
IoT6 Research project
c/o Mandat International
iot6@mandint.org
http://www.iot6.eu

Sebastien Ziegler
IoT6 Project Coordinator