Extending the Internet of Things to the Future Internet through IPv6 support

Antonio J. Jara[1], Socrates Varakliotis[2], Antonio F. Skarmeta[1], Peter Kirstein[2]

[1]Department of Information and Communications Engineering (DIIC), Computer Science Faculty at the University of Murcia, ES-3100, Spain.

[2]Department of Computer Science University College London, WC1E 6BT, UK

{jara, skarmeta}@um.es; {socrates, kirstein}@cs.ucl.ac.uk

Abstract – Emerging Internet of Things (IoT)/Machine-to-Machine (M2M) systems require a transparent access to information and services through a seamless integration into the Future Internet. This integration exploits infrastructure and services found on the Internet by the IoT. On the one hand, the so-called Web of Things aims for direct Web connectivity by pushing its technology down to devices and smart things. On the other hand, the current and Future Internet offer stable, scalable, extensive, and tested protocols for node and service discovery, mobility, security, and auto-configuration, which are also required for the IoT. In order to integrate the IoT into the Internet, this work adapts, extends, and bridges using IPv6 the existing IoT building blocks (such as solutions from IEEE 802.15.4, BT-LE, RFID) while maintaining backwards compatibility with legacy networked embedded systems from building and industrial automation. Specifically, this work presents an extended Internet stack with a set of adaptation layers from non-IP towards the IPv6-based network layer in order to enable homogeneous access for applications and services.

Keywords – Internet of Things, Network communications, Internetworking, Wireless sensor networks, Backwards Compatibility, System Architecture, IPv6.

1. Introduction

The Internet of Things (IoT) [1] is one of the main applications driving the evolution of the Internet towards the Future Internet. Here sensors, actuators and devices (now called things), are connected to the Internet through gateways and Supervisory Control and Data Acquisition platforms (SCADAs). This Intranet of Things [2] is being extended to smart things [3] with a higher scalability, pervasiveness, and integration into the Internet Core.

The ongoing and future work aims to create an extended Internet of Things. This requires both an architecture and products that allow for the extension of the Internet technologies, in order to reach a homogeneous integration of the Future Internet, Services, People, and Things with the Future Internet of Things, Services and People.

This drive to integrate everything into the Internet Core is motivated by the market wish to have all processes remotely accessible – while at the same time understanding that re-engineering an infrastructure to allow this for each application would be prohibitively costly and time-consuming. Moreover, the current evolution from uniform mass markets, to personalized ones, where the customization and user-specified adaptation is a requirement, makes the sort of uniform infrastructure found in the Internet, imperative. This allows many components to be re-used, and services to be shared, with correspondingly huge economies of scale and shortened implementation times.

The Internet of Things fills the gap between the needs arising from the evolution of the market, information, users, and things, by moving all of them to a common framework, the Internet. This is different from the current approach in such applications, where they are based usually on stand-alone and monolithic solutions designed for a narrow application domain. Users now require more flexibility and freedom. Offering a common framework allows choice among the available manufacturers, suppliers, service providers, delivery options, and payment services. While this obviates the need for standalone or proprietary solutions, it also requires a high level of integration.

This work describes an integrated approach to the Future Internet that supports existing Internet of Thing technologies and extends, and bridges to IPv6, the existing IoT building blocks. The basic network technologies proposed are the use of 6LoWPAN [4]; 6LoWPAN offers IPv6 over IEEE 802.15.4, the EPC codes in RFID [5], and Bluetooth Low Energy (BT-LE). We propose mappings between the three technologies. These involve a novel header compression and adaptation protocol for BT-LE and IEEE 802.15.4 called GLoWBAL IPv6 [6], which presents a better performance for header compression when Global IPv6 addresses are used.

Second, an integration solution has been proposed using a multiprotocol card for maintaining backwards compatibility with legacy, networked, embedded systems. As a specific example, we present the integration of EIB/KNX [7], X10 [8], Control Area Networks Bus (CAN) [9], and digital/analog I/O for buildings and industrial automation. These technologies were integrated into a novel IPv6 addressing structure, in order to achieve integral support of IPv6 throughout the proposed framework, which is complementary to the aforementioned integration of IPv6.

This approach provides a further step for the integration of IPv6 into the IoT; this is part of the coexistence strategy for the management of heterogeneous technologies and architectures, on the way to achieving interoperability across businesses, service providers, and users [10].

2. Related Technologies and Standards

IoT is the main driver for the Future Internet, where IPv6 is the fundamental technology. It is estimated that the Future Internet will number hundreds of billions of *connected things* by 2020. Unlike IPv4, IPv6 can address this number of objects. The IPv6 address space supports 2128 unique addresses (approximately 3.4×1038); specifically, it can offer 1.7×1017 addresses on an area about the size of the tip of your pen. The advantages of the IPv6 integration in the IoT are not limited to a universal addressing space; its main advantages are to offer stable, scalable, extensive, and tested protocols for global end-to-end communications, node/service discovery, mobility, end-to-end security, and relevant features such as stateless addressing auto-configuration, multicast addressing for group operations and its extensibility for application layers with technologies such as Web Services.

The initial step for the integration of a dual IPv4/IPv6 stack in embedded systems was lwIP/uIP [11]. This approach focused mainly on the reduction of the memory requirements and code size for the communication stack. This stack is commonly used for the integration of IPv6 in embedded systems for their communication interfaces such as Ethernet.

Embedded systems and sensor networks have been extended during the last decade or so with wireless technologies. Therefore, IPv6 support is needed also in technologies for Wireless Sensor Networks such as IEEE 802.15.4. The constraint of these networks is not only the limited memory, but also the reduced payload available in view of the low power consumption features desired. For this reason, a new protocol, 6LoWPAN [4], was designed as an adaptation layer to carry IPv6 datagram over the IEEE 802.15.4 link, taking into account the limited bandwidth, memory and energy resources. This adaptation layer has focused on header compression in order to reduce the processing load.

In addition to 6LoWPAN, the GLoWBAL IPv6 protocol [6] presents a lower overhead than 6LoWPAN header and an optimized approach for global communications, since 6LoWPAN has limitations in the compression of global IPv6 addresses. For this reason, GLoWBAL IPv6 is being considered for new wireless technologies with higher payload limitations such as Bluetooth Low Energy.

An important aspect of the Internet is that there is a uniform interface between applications and network services. Hence Web technologies such as HTTP, REST, SOAP, JSON and XML [12], which provide access to resources and services, are being adapted to the IoT. Thereby, the application layer for smart things is being defined via Web Services, thus becoming the Web of Things [13].

This is considered the most generic and homogeneous route to access services from the IoT. Specifically, it is based on a constrained version of RESTFul, denominated CoAP [14]. DNS-SD [15] also defines a description of CoAP Web Services which follow the semantic and naming conventions that describe how services will be represented in DNS records. In addition, the CoRE IETF working group is defining the Web Linking description, termed the Link format, and other protocols for CoAP such as Observe and Blockwise Transfer [14].

The IETF is also in the process of defining light versions of XML and SOAP (specified as EXI [16], and SOAP lightweight [17], respectively). Regarding security, DTLS is being simplified to DTLS for CoAP [18], JSON offers security with JOSE, and ID/Locator split architectures such as HIP in its “diet” version, i.e. HIP DEX [19].

When this entire infrastructure has been provided, it is necessary to apply it to a specific use case to ensure all hangs together.

3. Integration of Things in the IPv6 Stack

The previous section has already presented some adaptation techniques defined for the integration of IPv6 in smart things. However, to account for the degree of heterogeneity encountered in the real world one has to consider how legacy technologies would integrate into the IoT using IPv6. While it would be convenient if the legacy systems were immediately abandoned, such an approach is completely unrealistic. Just as it is clear that there will be a long period of overlap between the use of IPv4 and IPv6 in the Internet, there will also be a lengthy period of overlap between use of legacy systems for application domains, and their transition to systems more tailored to the IoT.

While the use of IPv6 and 6LoWPAN are strongly advocated for the Internet side of the IoT, the integration of legacy, non-IPv6-enabled technologies, require additional mechanisms in order to map the different address spaces to the IPv6 one. These legacy technologies have been tailored to, and are heavily used in, areas such as building and industrial automation. For this reason, this work aims to provide a transparent mechanism for users, devices and control systems to map the different address spaces to a common IPv6 one. Using the proposed integrated Internet stack, with legacy-system-specific translation gateways, every device from each technology will get a common framework based on IPv6 and protocols over IPv6 such as Web Services and any other protocol via TCP/UDP sockets.

For these reasons, a key contribution of this paper is the definition of ***Half-Gateways (HGWs)*** that bridge non-IPv6-enabled technologies with an IPv6-powered environment. This integration operates on top of the Future Internet infrastructure, i.e. IPv6. Thus, users and clients discover and use homogeneous IPv6-based resources.

At the same time, the application domains will continue to use well-known established protocols and already deployed technologies such as KNX/EIB for building automation, and new technologies such as Bluetooth Low Energy, which are not based on IPv6 networks, and other technologies such as RFID and its identifiers, e.g. Electronic Product Code (EPC) [20] and UID [5].

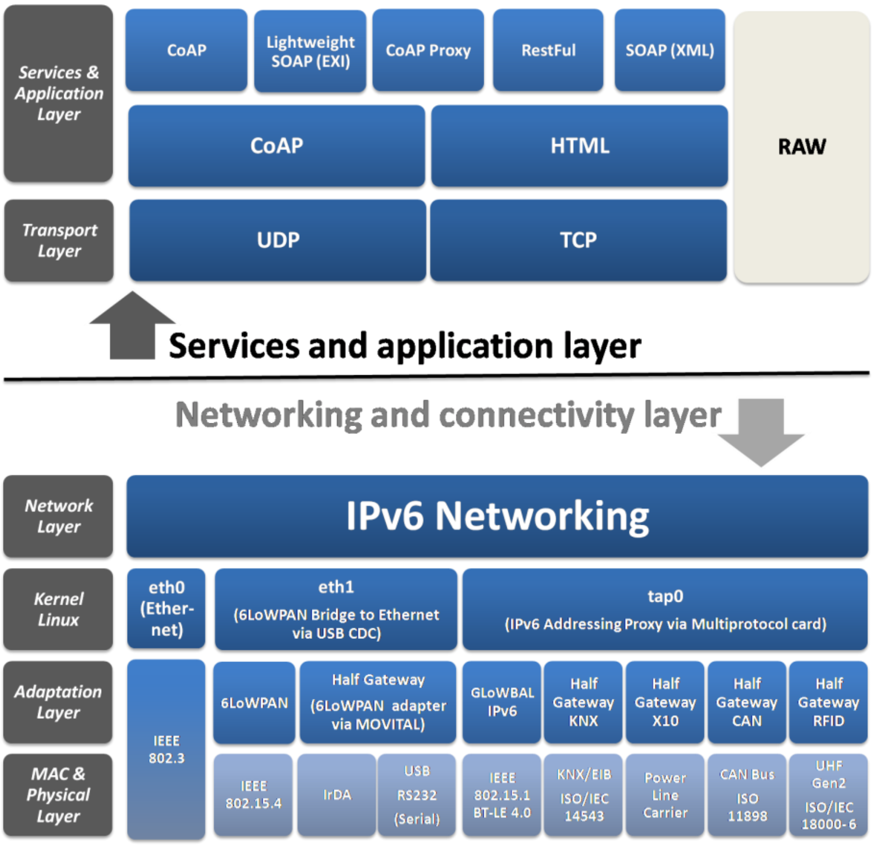


Figure 1. Integration of Things in IPv6 stack.

Our principal goal is to focus on IPv6 network mechanisms in order to homogenize the discovery, access and use of resources through the Internet infrastructure, i.e. through the IPv6 network. This makes services reachable via homogeneous and interoperable technologies. For example, Web Services and the discovery of services should be conducted through network-based Information Systems that are already deployed, such as the Domain Name System with Service Discovery (DNS-SD) [15].

Specifically, Figure 1 presents the technology stack proposed for the full integration of smart things, building automation technologies, RFID tags, and embedded technologies into a homogeneous IPv6 networking layer.

Under the IPv6 layer exists the sub-system adaptation and integration modules. These modules are based on hardware and/or software. The hardware adaptation provides the physical interface between the proprietary or native protocol and our platform. The software module transforms the native functionality into a set of homogeneous Web Services accessible via IPv6.

This architecture provides to the layers above IPv6 an environment for services and applications totally independent of the underlying technology.

Just as in the rest of the Internet, a transport layer is defined; in the case of smart things this is mainly focused on UDP because of its simplicity. For example, 6LoWPAN offers compression for UDP headers; however, implementations exist for TCP in most extended operating systems used for smart things such as Tiny and Contiki OS.

In the application layer, due to the dominance of the aforementioned Web of Things, the main tendency is to offer Web Services based on the Constrained Application Layer for smart things, i.e. CoAP; this is a light version of RESTFul/HTTP, paying attention to the limitations of these devices.

Stacks like those in Figure 1 require adaptation layers at many different levels; thus a stack is being built for the IoT analogous to the existing one for the current Internet. Specifically, a set of gateways, proxies, and border routers provide the adaptation. Some examples are the 6LoWPAN Border Router [4], the CoAP proxy [21], and now the HGWs presented in this work.

HGWs interface between the Internet and the different proprietary architectures. The properties in each network/architecture must be considered separately; then, the two stacks must be connected together through the HGWs into an IPv6 network.

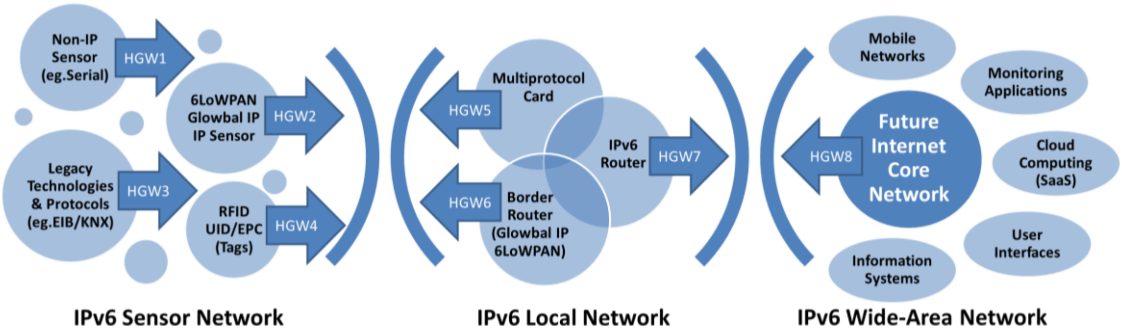


Figure 2. Schematic of the topology envisaged to integrate in IPv6 through half-gateways from the sensor level to the Future Internet Core Network.

Figure 2 illustrates our vision of the integration of the things into the Future Internet Core Network. There are usually three regions.

The left region of the Figure 2 represents the "IPv6 Sensor Network" domain, a term we use to denote a set of technologies including legacy protocols, RFID, non-IP sensors, and 6LoWPAN/GLoWBAL IPv6 sensors. This contains both legacy and IPv6-enabled subsystems that are directly concerned with things. It is here that each subsystem may require a technology-specific HGW in order to adapt/bridge the native protocol with an IPv6-enabled interface. The technologies from the left region (usually called the 'fringe' of the Internet) connect to an "IPv6 Intranet" through different HGWs. Each HGW bridges a specific technology and provides adaptation layers specific to this technology. For that reason, this “IPv6 Intranet” can be seen as an “Application or technology-dependent Internet”, since we need to take into account the specific features from the applications and technologies integrated. This Intranet is located at the central region, which is a domain-specific, but IPv6 technology. This contains the domain-specific operations, and has access to the domain-specific resource databases such as is the case of a multi-protocol card, residential gateway or Border Routers for 6LoWPAN. Section IV presents the technologies proposed to support the HGWs functionality used in this work.

This central region is also composed of the network termination broadband adapter such as xDSL modems and IPv6 routers which link the local and domain-specific networks to the Wide Area Network. Thereby, all access to the things resources are via standard Internet procedures.

The right region represents the Wide Area Network accessed through the Future Internet Core Network. This offers the different technologies and services of the Future Internet, such as monitoring applications, Software as a Service (SaaS) solutions, and any information system.

In summary, this integration of things with IPv6, through the presented communications stack and architecture, offers the aforementioned advantages for interoperability and homogeneity. In addition, it also enables the related IPv6 standards to be used, offering a wide set of tested, open, and extended technologies. Smart things, and solutions based on them, are able to benefit from the standard implementations.

4. Building Blocks of the proposed IPv6 integrated Stack for Smart Things

The integration of things in the IPv6 stack presents several advantages to use standard, tested, open and useful solutions through open platforms such as Linux OS, which offers all the required IPv6-related technologies and protocols. For this reason, in the proposed stack, Figure 1, we have defined an adaptation of the technologies to an interface compatible with the Linux Kernel, i.e., Linux Operating System (OS).

Linux OS allows to use the existing implementations for routing (‘routed’), security policies (‘iptable’), Neighbor Discovery (‘radvd’), DNS-SD (‘bind’) and mDNS (‘avahi’). In addition, this makes the framework more secure and scalable; since it can be configured to contain sub-networks, one for each technology, and allocate a virtual network kernel devices built with virtual interfaces such as tun/tap. Thereby, the different access policies may apply for each virtual interface, thus allowing isolation and independent managing of each sub-network. It can allow the definition of different namespaces for DNS-SD/mDNS [15]. This also allows the upgrade of platform components without requiring reconfiguration of the other components; this makes the system easier to expand and allows the installation of installing software that has already been tested and made available for Linux thus - adding a degree of robustness to the proposed IoT integration framework.

In order to implement the integration of the IoT using IPv6, we consider the platforms presented in Figure 3. Here, one can see various instances of HGWs: some bridging towards legacy and non-IPv6 sensor technologies and some acting as gateways to interconnect to the Future Internet Core Network via the Local Network (both being IPv6-based). The main platform is the multiprotocol card presented in Figure 3.1. It is based on the Atmel ARM9 processor running at 400MHz (32-bit) with 256MB RAM and 256MB NAND memory, which supports Linux OS. It features 6LoWPAN adaptation (with Contiki OS) (point A). It also offers the following interfaces: GPRS from Wavecom (B), Bluetooth from BlueGiga (C), USB 2.0 ports, I/O digital/analog/relays (D), and Ethernet 100Mbps (E).

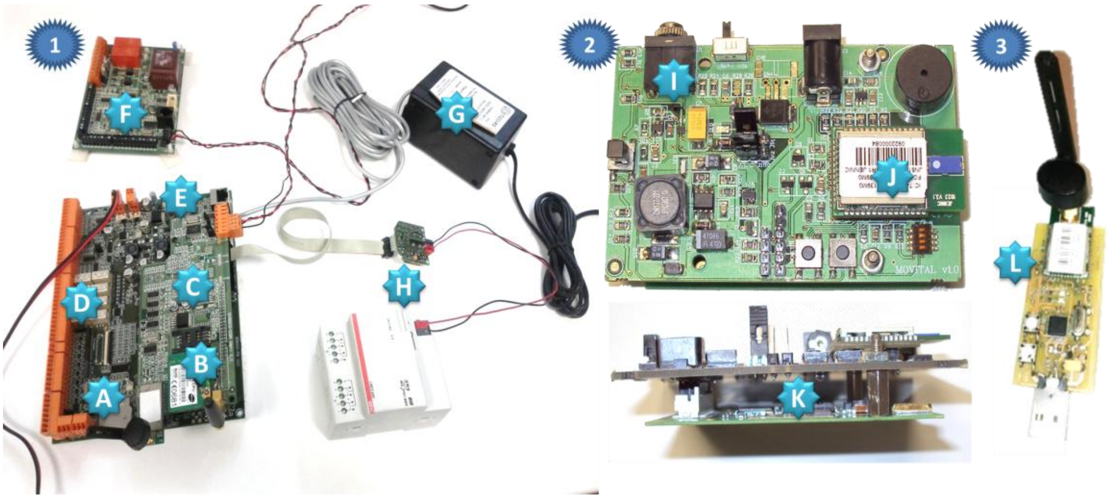


Figure 3. (1) Multiprotocol card with 6LoWPAN (A), GPRS (B), Bluetooth (C), native interfaces (D), Ethernet (E), CAN (F), X10 (G), and KNX (H). (2) Movital adapter with native interfaces (I), 6LoWPAN (J) and RFID (K). (3) 6LoWPAN Bridge based on USB (L) from 6loWPAN to Ethernet.

This multiprotocol card supports, through its extension interfaces (serial RS232 and SPI ports), the technologies for industrial and building automation. More specifically, the extension interfaces support Control Area Networks (CANs) (F), X10 (G) and European Industrial Bus (EIB)/Konnex (KNX) (H). To put these into the context of Figure 2, one can see how the previous technologies represent HGW2 from the IPv6 Sensor network and HGW5 from the IPv6 Intranet points of view.

The following subsection describes the technologies and protocols proposed for the HGWs and some of the adaptations carried out for the different technologies, which define the building blocks of the presented stack.

4.1 6LoWPAN

6LoWPAN defines an adaptation protocol based on header compression for IPv6 datagrams. More specifically, 6LoWPAN defines one reduced header format for IP (IPv6) and one reduced format for UDP. 6LoWPAN is one of the most important technologies for the integration of IPv6 in smart objects based on Wireless Sensor Networks with low power, limited bandwidth and reduced memory capabilities.

In order to bridge 6LoWPAN subsystems in our framework with the Kernel Linux and IPv6, a USBNet bridge based on CDC-ECM (Ethernet Networking Control Model) has been built, (see Figure 3.3) along with a USB module (L). This way the 6LoWPAN-bridge defines an Ethernet Interface making access to the 6LoWPAN-connected devices transparent. Specifically, 6LoWPAN bridge carries out the translation from the 6LoWPAN header of the packets received via the WPAN interface to the IPv6/UDP headers for the packets transmitted through the Ethernet Interface, and vice versa, in a transparent way. Thereby, it also allows to use the existing Linux protocols such as ‘radvd’ for neighbor discovery, in order to announce the prefix of the network assigned to the 6LoWPAN. Comparing this to Figure 2, this feature corresponds to the functionality of HGW6.

An ancillary component to this 6LoWPAN bridge is the Movital, wireless and personal device, (see Figure 3.2), which is an adapter used for the integration of specific devices such as clinical sensors in personalized health solutions [22]. This connects legacy technologies through USB, Serial RS232, or IrDA, this is the functionality of HGW1 (I). Movital also offers an interface based on RFID of High Frequency (HF) for the user interaction with other users and objects through contactless identification, with the module from Skyetek, i.e. HGW4 (K), and finally the communication through 6LoWPAN, i.e. HGW3 (J).

The 6LoWPAN modules are based on the Jennic JN5139 module. This is an OpenRISC 32-bit processor, which supports IEEE 802.15.4, ZigBee-Pro, 6LoWPAN, and GLoWBAL IPv6. In addition, a port that supports Contiki OS has been built for them. Finally, this also implements an advanced cryptography stack based on Elliptic Curves in order to offer authentication, integrity and confidentiality. These are highly relevant to the IoT as attested in [18].

4.2 GLoWBAL IPv6

The compression headers originally defined for 6LoWPAN in RFC4944 were insufficient for many practical uses of IPv6 with smart things. This was because, the header compression method proposed in RFC4944 was primarily conceived to serve effectively unicast communications in local and personal communication scopes, where IPv6 addresses carry the link-local prefix and an Interface Identifier (IID) directly derived from IEEE 802.15.4 addresses. In this case, both addresses may be completely elided. When global communications are considered end-to-end, including smart things addressable by IPv6, the existing mechanism proved inefficient. To resolve the issue a new encoding format was standardized in the revised RFC6282, to improve the compression of Unique Local, Global, and multicast IPv6 addresses. This new encoding format is based on shared state within contexts. Although usable, the RFC6282 method yields header overheads of 26 bytes; while it does better than the 41 bytes required by RFC4944, it can still be inefficient considering the 102 bytes available for a LoWPAN frame (127 bytes less the 25 bytes from the MAC layer header).

For this reason, GLoWBAL IPv6 [6] has been proposed to optimize global addressing involving LoWPANs. This has the further advantage that it provides efficient addressing and integration to both IEEE 802.15.4 sensor devices, which have no native support for 6LoWPAN, and also to other technologies which do not cater for IPv6 communication capability into their stacks.

Take for example a smart device with a Bluetooth Low Energy interface, such as a smart phone. Usually, a Smart Phone would offer Internet connectivity through its GPRS/GSM network interface. GLoWBAL IPv6 fills the IPv6 addressing requirement for any smart thing connected to the smart phone’s Bluetooth Low Energy network (compare Figure 2) by acting as the mapping protocol between the Local Network and the wide-area network using appropriately constructed IPv6 addresses. Thereby, this smart phone can efficiently enable with IPv6 through GLoWBAL IPv6 to the smart things connected to it through its Bluetooth Low Energy interface.

GLoWBAL IPv6 defines an Access Address/Identifier (AAID), and an AAID-IPv6 address translation mechanism for different technologies, in order to adapt any device to the IoT architecture with IPv6. In this respect, AAID simplifies IPv6 communication parameters (source and destination addresses/ports, originally 36 bytes long) to a single 4-byte communication identifier augmented by one byte for the ‘Dispatch’ header field, totaling 5 bytes for the GLoWBAL IPv6 header. Thus, the IPv6/UDP headers are significantly reduced. This mechanism achieves an efficient frame format for global communications in networks that do not have native support for IPv6.

The implementation of the GLoWBAL IPv6 mechanism is done in a gateway dedicated to carrying out the translation from AAID to IPv6 and vice versa. In reality the gateway is a software module built over the smart phone or the multiprotocol hardware platform depicted in Figure 3.1, which utilizes the mentioned virtual network kernel devices such as tun/tap. The device operates at layer 2 and simulates Ethernet frames. As such, it can exchange frames with the Future Internet core network (right side of the Figure 2).

4.3 IPv6 Addressing Proxy

The current situation in industrial and building automation is a rather fragmented set of technologies. Each technology comes with a set of fit-for-purpose sensors and their respective application environments with lack of efficient interoperability among them. Some associations of manufactures have been formed to build common technology frameworks, e.g. Konnex (KNX) for building automation. While such de facto standards enjoy widespread adoption to date, this does not discourage use of other relevant protocols such as the emerging ZigBee and the older X10. Due to this fragmentation, various IoT initiatives are considering a shift towards a common access and communication framework based on IPv6. Adoption of the Internet Protocol implies that addressing of devices in each legacy technology needs to be redefined. With our work we aim to provide a transparent mechanism for the users and devices to map the different addressing spaces from each legacy technology to a common IPv6 addressing space.

The challenge for IPv6 here is to embrace all existing native addressing schemes, so that the features and functional specifications of existing devices in the legacy building automation networks are maintained. To this respect, we have proposed IPv6 mappings for each native addressing scheme by use of an IPv6 Addressing Proxy which handles the translations between an IPv6 address and its corresponding technology addressing, i.e. the native addressing depending on the technology.

Figure 4 describes the mapping hierarchy from the multiprotocol card presented in the Figure 3. This mapping is managed by a set of tun/tap virtual network kernel device as in the GLoWBAL IPv6 implementation. The IPv6 Addressing Proxy requires information for each technology, this has been implemented for legacy networks, such as the EIB/Konnex, CAN, X10, and it is currently being adapted for BACNet and DALI networks.

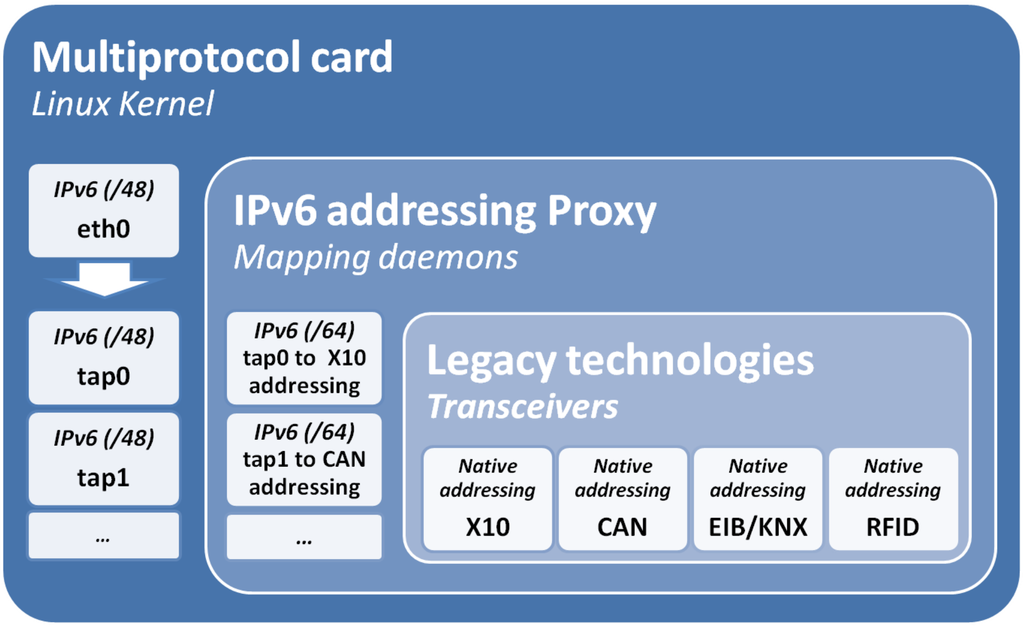


Figure 4. IPv6 addressing Proxy integration in the Multiprotocol card.

Figure 5 presents an example of the mapping technique for the native addressing from EIB/KNX based on logical and physical addressing. Logical addressing is focused on the organization in groups/families, and physical addressing is more focused on the location of the device. These native addressing fields, presented in the right of the Figure 4, have been mapped into an IPv6 address structure directly in conjunction with the properties specific to each technology.

Mapping allows that a system can easily locate and identify the device in a multi-protocol card platform such as the one we presented in Section III (Figure 3.1).

The mapping is carried out in the lowest 48 bits of the “Host ID” half of an IPv6 address, leaving the highest 16 bits for application-level sub-networking. In addition, a network prefix of 48 bits has also been assumed; this arrangement leaves 8 bits for sub-networking of the technology (*LINE ID*) and another 8 bits for sub-networking of the ‘group’ (*TECH ID*). Thus access policies can be managed and a more scalable management of the different technologies is enabled.

Native addressing for EIB/KNX is based on the concept of physical and logical addresses. Physical addresses are defined by ‘lines’. The lines are grouped by ‘areas’, and finally, areas connect to the backbone of the network. Logical addresses are used to associate a group of devices with similar functionality.

The proposed mapping takes into account the definition of physical and logical addresses, in order to make the mapping easier. Therefore, the various bits of the EIB/KNX addresses maintain their semantics and they are extended with new features.

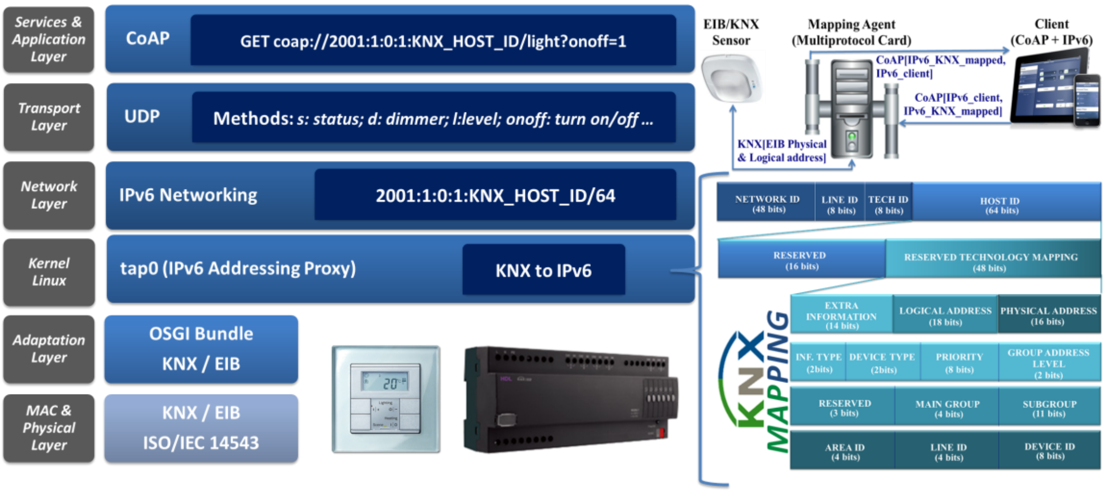

Services & Application Layer
CoAP
GET coap://2001:1:0:1:KNX_HOST_ID/light?onoff=1
Transport Layer
UDP
Methods: s: status; d: dimmer; l:level; onoff: turn on/off ...
Network Layer
IPv6 Networking
2001:1:0:1:KNX_HOST_ID/64
Kernel Linux
tap0 (IPv6 Addressing Proxy)
KNX to IPv6
Adaptation Layer
OSGI Bundle KNX / EIB
MAC & Physical Layer
KNX / EIB ISO/IEC 14543
EIB/KNX Sensor
Mapping Agent (Multiprotocol Card)
Client (CoAP + IPv6)
CoAP[IPv6_KNX_mapped, IPv6_client]
CoAP[IPv6_client, IPv6_KNX_mapped]
KNX[EIB Physical & Logical address]
NETWORK ID (48 bits)
LINE ID (8 bits)
TECH ID (8 bits)
HOST ID (64 bits)
RESERVED (16 bits)
RESERVED TECHNOLOGY MAPPING (48 bits)
KNX MAPPING
EXTRA INFORMATION (14 bits)
LOGICAL ADDRESS (18 bits)
PHYSICAL ADDRESS (16 bits)
INF. TYPE (2bits)
DEVICE TYPE (2bits)
PRIORITY (8 bits)
GROUP ADDRESS LEVEL (2 bits)
RESERVED (3 bits)
MAIN GROUP (4 bits)
SUBGROUP (11 bits)
AREA ID (4 bits)
LINE ID (4 bits)
DEVICE ID (8 bits)

Figure 5. Left: Internet integration stack instance for KNX integration. Right: KNX mapping from native addressing to IPv6 and translation process.

Figure 5 depicts how the physical address structure is maintained in the lower (least significant) 16 bits. These fields contain information about the device identifier, line and area. In addition, the ‘Extra Information’ field is defined in the remaining 14 bits. These have been further sub-divided into 4 fields:

- The ‘Information Type ID’ is an identifier of the information's type that the device is able to manage. It can differentiate between 4 different types of information, which are defined by the following field the “Device type”,

- The Device Types for EIB/KNX are sensor, actuator, line coupler, and area coupler.

- The Priority field, which matches with the priority field of the EIB message, for future Quality of Service use

- The Group Address Level to indicate the logical level addressing. In our case this field is always set to Level 2.

4.4 Other technologies

6LoWPAN has been presented as suitable for integrating smart things into IPv6. We now propose GLoWBAL IPv6 and an IPv6 addressing Proxy as an optimization for 6LoWPAN and to enable IPv6 to be used with new programmable technologies such as Bluetooth Low Energy. In addition, the IPv6 addressing Proxy is now proposed also for the integration of technologies which are not programmable; here a proxy is needed in order to translate from the assigned IPv6 address for each end device to the native addressing defined by the legacy technology. This new technology provides a solution that is applicable for any current addressing scheme. Moreover, it can be considered also for the current translations schemes from non-IPv6 addressing to an IPv6 one for the identifiers from RFID technologies, Digital Objects Identifiers

(DOIs) and Universal Identifiers (UID). For example, the protocol and identifier deployed most widely is the Electronic Product Code (EPC) from EPCGlobal. EPC codes are 96-bits unique codes. An architecture has been proposed similar to that of the Internet for the management of the EPC; it consists of EPC Information Systems and a global Object Name Server (ONS), which can be seen as the equivalent to the DNS.

A mapping between EPC and IPv6 is needed in order to integrate EPC over IPv6. This integration is justified, since EPC is not a unique standard for products identification.

A Unique Identifier (UID) has been defined, which consists of a 40-bit identifier hard-coded by the manufacturer to ensure it is really unique. This uniqueness property is being considered by the pharmaceutical industry, because it satisfies its requirements for offering an efficient, trustable, and safely traceable solution.

Finally, the integration of the Digital Objects Identifier (DOIs) should be considered because of its extended use in physical things such as books and movies.

5. Discussion and Conclusions

The key contribution of this paper is the proposal of a set of technologies for the extension of legacy technology addressing to the IPv6 address space. This will allow the management of all things around us and access to their information independently of the technology used to convey this information.

For this purpose an integration stack and appropriate hardware platforms have been proposed. In addition, the address space integration has been supplemented by the Half-Gateways, which bridge legacy technologies to the IPv6 world, either at the network layer, or at the application layer as required. We have instantiated this integration with concrete examples for EIB/KNX, X10, CAN, Bluetooth Low Energy, and IEEE 802.15.4. The proposed technology is not limited to the above; additional legacy technologies, protocols and identifiers living in the fringe of the Internet (thus making up the IoT) can be considered such as the DOI, EPC and UID as mentioned. Table 1 summarizes the main features of the existing and the proposed solutions.

Protocol/ Feature	Code Size Optimized (low memory req.)	Header Size Optimized (overhead level from IPv6 header)	Communication Stack independent	Feasible for legacy technologies (application level editable)	Feasible for proprietary technologies (non editable)	Require Border Router or Gateway	IPv6 address managed by end-device
IPv6 (Base)	✗	✗ ●●●●●	✗	✗	✗	✓	✓
lwIP	✓	✗ ●●●●●	✗	✗	✗	✓	✓
uIP	✓	✗ ●●●●●	✗	✗	✗	✓	✓
6LoWPAN (RFC 4944)	✓	✓ ●●●○○	✗	✗	✗	✗	✓
6LoWPAN (RFC 6282)	✓	✓ ●●○○○	✗	✗	✗	✗	✓
GLoWBAL IPv6	✓	✓ ●○○○○	✓	✓	✗	✗	✓
IPv6 addressing Proxy	✓	✓ ○○○○○	✓	✓	✓	✗	✗

Table 1. Comparison of different protocols for the integration of IPv6 in constrained and legacy technologies.

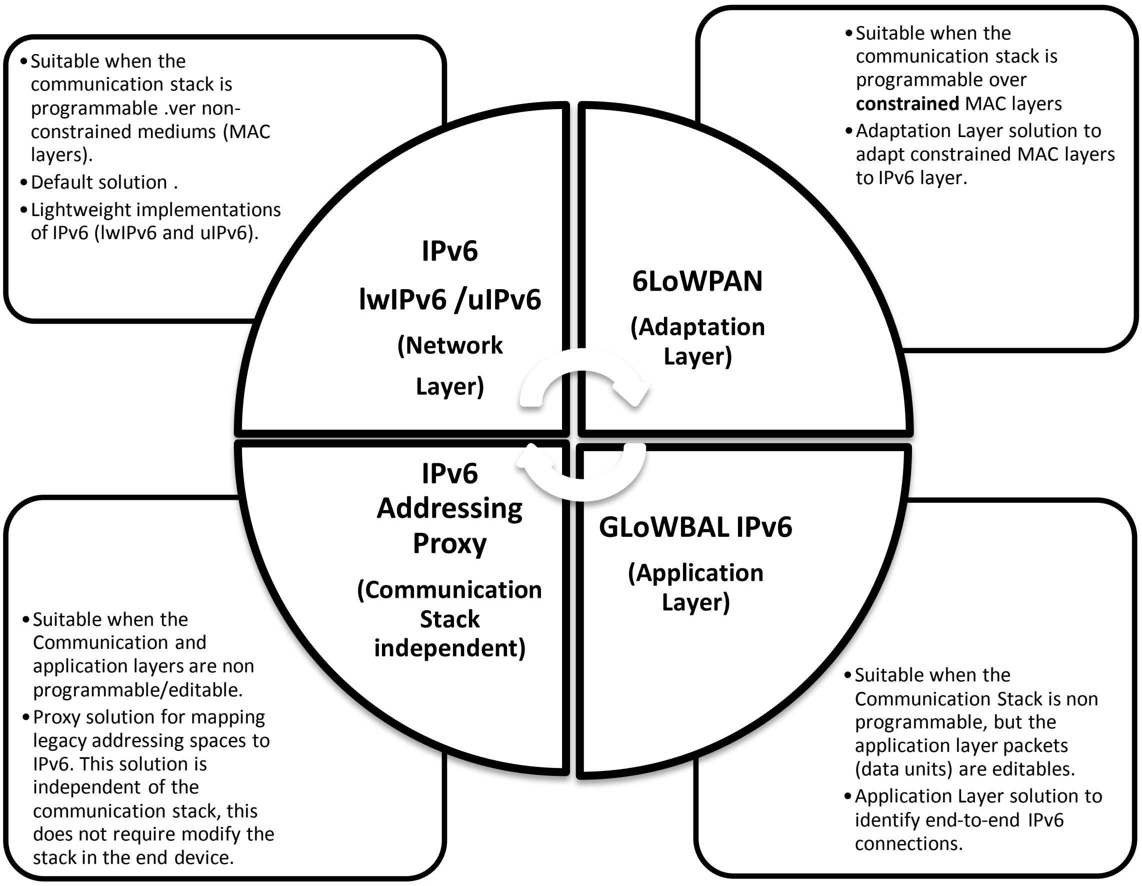
Table 1 shows that lwIP and uIP have mainly focused on the reduction of the code footprint, since the stack was defined for wired technologies such as embedded systems with Ethernet Interfaces. For a wireless medium with constrained frame size, 6LoWPAN with header compression mechanism presents a high processing load. For this reason, we propose GLoWBAL IPv6. This presents a reduced overload, based on the reduced overhead from GLoWBAL IPv6 header in relation with the overhead from IPv6 header and even 6LoWPAN header. In addition, GLoWBAL IPv6 would allow the integration in the application layer (payload) of the AAID in order to make it compatible with solutions, which are programmable at the application level but not are able to be modified in the communication stack. Therefore, it is communication stack independent, which is useful for the IPv6 integration over already deployed networks based on closed stacks over technologies such as the Bluetooth Low Energy and IEEE 802.15.4.

Since not all the technologies and solutions are able to be programmed at the application layer, we have defined an IPv6 addressing Proxy, which offers compatibility with proprietary technologies and native address spaces. The main advantage of GLoWBAL IPv6 with respect to the IPv6 addressing Proxy is that with GLoWBAL IPv6 the end device is aware about IPv6, while in the IPv6 addressing Proxy, the end-device is totally agnostic about the version from IP.

The second key contribution is the ability to integrate into the Internet legacy technologies and devices with a new rich range of IPv6-enabled services. This way both real and virtual objects can interoperate and communicate by way of IPv6 end-to-end, if their corresponding services so wish. Thus, homogeneous, transparent and scalable access to devices and services can be achieved. From the Web of Things approach, the CoAP/RestFul methods could be applied directly to the end-device, increasing the scalability of the solution, offering flexibility and allowing for extension of the ubiquitous concept with mobility and global interoperability. This is just what back-end services like Software-as- a-Service would like to see.

Such as presented in the Table 1, the main problem from the proposed adaptations to integrate Internet in constrained devices such as 6LoWPAN, GLoWBAL IPv6, and IPv6 addressing Proxy is that these nodes require of 6LoWPAN Border Routers or gateways to adapt from the lightweight version of the protocol headers and protocols to the common one, in order to interoperate with the rest of the IPv6-enabled entities.

Figure 6 summarizes the suitability of each one of the presented solutions in terms of constrained level for the Medium Access Control (MAC) layer, and the programmability of the different layers from the communication stack.



•Suitable when the communication stack is programmable .ver non-constrained mediums (MAC layers).
•Default solution .
•Lightweight implementations of IPv6 (lwIPv6 and uIPv6).
•Suitable when the communication stack is programmable over **constrained** MAC layers
•Adaptation Layer solution to adapt constrained MAC layers to IPv6 layer.
IPv6
lwIPv6 /uIPv6
(Network Layer)
6LoWPAN
(Adaptation Layer)
IPv6 Addressing Proxy
(Communication Stack independent)
GLoWBAL IPv6
(Application Layer)
•Suitable when the Communication and application layers are non programmable/editable.
•Proxy solution for mapping legacy addressing spaces to IPv6. This solution is independent of the communication stack, this does not require modify the stack in the end device.
•Suitable when the Communication Stack is non programmable, but the application layer packets (data units) are editables.
•Application Layer solution to identify end-to-end IPv6 connections.

Figure 6. Classification of the presented solutions in function of the communication layer upgraded.

Our vision is that in the current Internet of Things world everything can be discovered through global resource directories, distributed as desired. These directories would be based on Internet technologies such as the example of mDNS/DNS, and accessed in a homogeneous way through Web Services technologies over IPv6 such as HTTP and CoAP at the application level. This will be complemented with SenML over JSON, RDF, or EXI for the semantic description such as those defined by SPITFIRE [24].

Ongoing work is focused on offering mobility and multi-homing support for the GLoWBAL IPv6 protocol and the IPv6 addressing Proxy. We also need to extend our technology to multiple addressing proxies – themselves accessed by relevant protocols (e.g. using IPv6 Anycast). We believe that applications in mobile environments and the integrated in devices such as smart phones with multiple communications interfaces should not depend on any particular IPv6 sub-network. For this reason, we consider an extension of the application protocols for smart things; this should include session management to define security associations, manage mobility and multi-homing support for the open sessions from each smart object. Presumably the architecture and gateways should address this device context regarding the open sessions such as the synchronization of the AAID among the different GLoWBAL IPv6 gateways.

The final conclusion is that smart phones, personal data terminals, and other mobile computing devices are still far from what a Future Internet of Things will require to connect services, people, and things. But, full IPv6 integration is the first step towards this destination. As next steps one envisages support for mobility, multi-homing, discovery techniques, and management solutions in order to make things more autonomous and to enable a communications era based on the **Future Internet of Things, Services and People.**

Acknowledgements

The authors would like to thank the European Project "Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability (IoT6)" from the FP7 with the grant agreement no: 288445, and the Spanish ministry, for education, social politics and sport, for sponsoring this research activity with the grant FPU program (AP2009-3981). Finally, this research has been partially carried out by the Intelligent Systems and Networks group, of the University of Murcia, Espinardo, Spain, awarded for its excellence as a research group in the frame of the Spanish "Plan de Ciencia y Tecnologia de la Region de Murcia" from the "Fundacion Seneca" (04552/GERM/06).

References

[1] Atzori, L.; Iera, A.; Morabito. G.; *The Internet of Things: A survey*. Computer Networks Vol. 54, No. 15, pp. 2787-2805, 2010.

[2] Zorzi, M.; Gluhak, A.; Lange, S.; Bassi, A.; *From today's INTRAnet of things to a future INTERnet of things: a wireless-and mobility-related view*, IEEE Wireless Communications, Vol.17, No.6, pp.44-51, 2010.

[3] Kortuem, G.; Kawsar, F.; Fitton, D.; Sundramoorthy, V.; *Smart objects as building blocks for the Internet of things,* Internet Computing, IEEE, Vol.14, No.1, pp.44-51, 2010.

[4] Shelby, Z.; Bormann, C.; *6LoWPAN: The Wireless Embedded Internet*, Wiley, ISBN: 978-0-470-74799-5, 2009.

[5] Welbourne, E.; Battle, L.; Cole, G.; Gould, K.; Rector, K.; Raymer, S.; Balazinska, M.; Borriello, G.; *Building the Internet of Things Using RFID: The RFID Ecosystem Experience*, Internet Computing, IEEE , Vol.13, No.3, pp.48-55, 2009.

[6] Jara, A. J.; Zamora, M.A.; Skarmeta, A.F..; *GLoWBAL IP: an adaptive and transparent IPv6 integration in the Internet of Things*, Mobile Information Systems, Vol. 8, N. 3, pp. 177-197, 2012.

[7] Merz, T. Hansemann, C. Hubner, *Building Automation: Comm. Systems with EIB/KNX, LON und BACnet*, Springer, Series on Signals and Communication Technology. ISBN.978-3-540-88828-4, 2009.

[8] Zamora, M.A.; Santa, J.; Skarmeta, A.F.G.; *An integral and networked Home Automation solution for indoor Ambient Intelligence*, IEEE Pervasive Computing, Vol. 9, pp. 66-77, 2010.

[9] Farsi, M.; Ratcliff, K.; Barbosa, M. *An overview of controller area network*, Computing & Control Engineering Journal, vol.10, no.3, pp.113-120, 1999.

[10] Uckelmann, D.; Harrison, M.; Michahelles, F.; *Architecting the Internet of Things*, Springer, ISBN 978-3-642-19156-5, 2011.

[11] Adam Dunkels, *Full TCP/IP for 8-Bit Architectures*. In Proceedings of the first international conference on mobile applications, systems and services (MOBISYS 2003), San Francisco, May 2003.

[12] Gurram, R.; Mo, B.; Gueldemeister, R. *A Web Based Mashup Platform for Enterprise 2.0*, Web Information Systems Engineering, Lecture Notes in Computer Science, pp. 144-151, Vol. 5176, 2008.

[13] Guinard, D.; Trifa, V.; Karnouskos, S.; Spiess, P: Savio, D; *Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services*, Services Computing, IEEE Trans., Vol.3, No.3, pp.223-235, 2010.

[14] Shelby, Z.; *Embedded web services*, Wireless Communications, IEEE , vol.17, no.6, pp.52-57, December 2010.

[15] Jara, A.J.; Martinez-Julia, P.; Skarmeta A.F.: *Light-weight multicast DNS and DNS-SD (lmDNS-SD): IPv6-based resource and service discovery for the Web of Things*, International Workshop on Extending Seamlessly to the Internet of Things, 2012.

[16] Castellani, A.P.; Gheda, M.; Bui, N.; Rossi, M.; Zorzi, M.; *Web Services for the Internet of Things through CoAP and EXI*, Communications Workshops (ICC), 2011 IEEE International Conference on, 5-9 June 2011.

[17] Carballido, B.; Pesch, D.; De Paz Alberola, R.; Fedor, S.; and Boubekeur, M.; *Constrained Application Protocol for Low Power Embedded Networks: A Survey*, International Workshop on Extending Seamlessly to the Internet of Things (esIoT), IEEE proceedings, 978-0-7695-4684-1, 2012.

[18] Roman, R.; Najera, P.; Lopez, J.; *Securing the Internet of Things*, Computer , vol.44, no.9, pp.51-58, Sept. 2011.

[19] Moskowitz, R.; *HIP Diet EXchange (DEX)*, IETF draft-moskowitz-hip-rg-dex (work in progress), Internet-Draft, Internet Engineering Task Force (IETF), 2012.

[20] Sang-Do Lee; Myung-Ki Shin; Hyoung-Jun Kim; *EPC vs. IPv6 mapping mechanism*, Advanced Communication Technology, The 9th International Conference on, Vol. 2, 2007

[21] Colitti, W.; Steenhaut, K.; De Caro, N.; Buta, B.; Dobrota, V.; *REST Enabled Wireless Sensor Networks for Seamless Integration with Web Applications*, Mobile Adhoc and Sensor Systems (MASS), IEEE 8th International Conference on, pp.867-872, 17-22 Oct. 2011.

[22] Jara, A. J.; Zamora, M.A.; Skarmeta, A.F.G.; *An internet of things–based personal device for diabetes therapy management in ambient assisted living (AAL)*, Personal and Ubiquitous Computing, Vol. 15, no. 4, pp. 431-440, 2011.

[23] Tatipamula, M.; Grossetete, P.; Esaki, H.; *IPv6 integration and coexistence strategies for next-generation networks*, Communications Magazine, IEEE, vol.42, no.1, pp. 88- 96, Jan 2004.

[24] Pfisterer, D.; Romer, K.; Bimschas, D.; Kleine, O.; Mietz, R.; Cuong Truong; Hasemann, H.; Kröller, A.; Pagel, M.; Hauswirth, M.; Karnstedt, M.; Leggieri, M.; Passant, A.; Richardson, R.; *SPITFIRE: toward a semantic web of things*, Communications Magazine, IEEE , Vol.49, No.11, pp.40-48, 2011.