Protocol and methods for LR-WPANs: (Low-Rate Wireless Personal Networks) a survey on existing standards and interconnection issues

Jacopo Mondi
Corso di Sistemi e Reti Wireless
Laurea magistrale in informatica
mondi@cs.unibo.it

June 7, 2011

Abstract

In last decade, growth in availability and computational power of low cost embedded devices, lead to an increasing demand of communication capabilities for interconnected systems of so-called "smart-objects" .
While intelligence and resources in end devices are constantly growing, a lack of interoperability and standardization is slowing down the real ubiquity realization, imposing the introduction of protocol translation gateways to let heterogeneous technology communicate.

This article firstly analyzes the actual state of art and common methodologies used to realize inter-network connections and the drawbacks imposed by lack of standardization in networks deployment context.
We concentrate on the mainly adopted technologies in the field, starting with the basic foundation standard for modern wireless sensors network *IEEE 802.15.4*, and then analyzing two concurrent approaches to the network and flow control tasks, the *ZigBee stack* and *6LowPAN* standard.

We then present an analysis of how is it possible to make a sensor network interoperable with the existing IP-based infrastructure, in relation to the previously discussed technologies, presenting advantages and drawbacks.

1 Introduction

From the first time until the IP revolution, that in the late 70's standardized the fragmented world of computer networks and made possible the realization of the internet as we currently know it, the same problems are emerging again.

Nowadays we are entering the ubiquity era, that manifest itself in many ways.
The widespread idea of an *internet of things*, capable of coordinate itself and communicate by means of dedicated protocols and methodologies, still lacks an efficient and standardized way to inter-operate with the existing infrastructures.

The emerging protocols and standards involving the "smart-ization" of sensor nodes or network devices are currently provided by OEMs, each one featuring it's own protocol stack and methodologies, that maybe conform to the same standards, but continue to isolate the dedicated network from the rest of the interconnected world.

This emerging requirements lead in this years to the formation of dedicated working groups, academic and industrial initiatives that aim to supersede the actual limitations we have to face when dealing with networked embedded systems.
One remarkable example of that kind of initiatives is the **IPSO Alliance** (IP for Smart Objects) started in late 2008 by *Adam Dunkels* and *J.P Vasseur*, respectively from SICS and CISCO.
As stated in the alliance white paper [1], the current situation is similar to what computer networks looked like about two decades ago, and toady as yesterday, bottlenecks and complexity mainly reside in multi-protocol gateways used to interconnect non standardized networks and the existing infrastructures.

The adoption of IP-based technologies is expected to contribute to the development of the availability and diffusion of wireless sensors networks for several reasons, including (but not only):

- Reuse of existing infrastructures

- IP is based on open and freely available specifications
- A great amount of tool and methodologies has already been developed for IP networks
- IP-based devices could be connected directly to other IP-based networks with no proxies or protocol translation gateways.

For such reasons the introduction of IP (and associated protocols, such as UDP or TCP) is a great prospective for the technological development of wireless sensor networks and for real ubiquity realization.

2 The current scenario

In the context we are analyzing, that involves devices with physical transmission capabilities limited to the usual *Wirelss Personal Area Networks* (WPANs) or to small *Wireless Local Area Networks* (WLANs), the protocol that is gaining even more consideration is currently *ZigBee*, whose success is mainly due to an industrial alliance and to the characteristics of the employed MAC layer (802.15.4).

Other technologies used in WPAN or small WLAN context, such as *Bluetooth* or less known standards such as *HomeRF* (also known as *Firefly*) based on other 802.15 MAC layers, are even more oriented to the cable replacement function, providing higher data rates, but diverging from the emerging requirements that modern sensor networks are imposing to protocol designers and original equipment manufactures.

In contrast to *ZigBee* approach that aims to redefine a dedicated network and flow control layer, in the last years the attention of research and industry has been dedicate to possible solutions to incorporate standard IPv6 technology into LR-WPANs.
Thanks to that efforts new protocols like *6LowPAN* have been standardized and are currently gaining even more attentions.

The design and implementation of the de-facto standards that are imposing themselves in the low power, low range wireless technologies are also slowing moving over the usual 7-layer definition provided by the ISO/OSI defined

network stack[2], and to fully understand why such an approach is being slowly abandoned or, at least heavily modified, an analysis of emerging requirements for LR-WPAN has to be presented.

2.1 LR-WPAN requirements

Since the application context is rapidly evolving, for ubiquitous and mobile networks new requirements are emerging and could be summarized as done by J.A. Gutirrez et Al. [3]:

Emerging Requirements for Sensor Networks and MANETs

- **Power Consumption**:
  For stand-alone RF transceiver battery should last as long as possible to reduce on-site maintenance.

- **Range**:
  Limited available power reduce the possible communication range in the ISM band.

- **Network Topology**:
  Due to low range transmission multi-hop protocol, possibly with low duty cycle activity are required to enable a full peer-to-peer communication among network members.

- **Self-Organization**:
  To reduce installation effort no special on-site configuration has to be done. That means that node forming the network must be able to auto configure addressing, traffic balancing and association.

To archive especially self-organization and power consumption efficiency, the traditional view of MAC layer as a monolithic block serving the network protocol and driving the physical layer, could been decomposed in various sub-layer, defining services for higher levels applications interacting with the lower level functionality.

Furthermore the pressure imposed by strict requirements on fault-tolerance and fairness in communications between peers, the necessity to assign specific roles to nodes forming the network in order to implement energy conservative devices, is forcing protocol designers to experiment with cross-layer approaches and methodologies.
That redefines the usual 7-layer approach, since the context the designer was referring to 30 years ago, was quite different from the one that is emerging today.

3 IEEE 802.15.4

The *IEEE 802.15.4* MAC layer has become the de-facto standard for LR-WPANs. This protocol has been standardized in 2003 by *IEEE* under the guidelines specified in the *IEEE Std. 802.15.4-2003*[11].
The main feature this standard introduces are, in particular, the low duty cycle necessary for devices to communicate, multiple operation mode (low power, full features), and simple frame structure with optional QoS support. 802.15.4 MAC layer also provides basic mechanism for implementing security and encryption, such as AES-128, ACL modes, Data Encryption, Frame Integrity and Sequential Freshness.

In addition to the usual functions, a LLC layer has been introduced above the MAC, providing optional flow control mechanism and performing basic network administration tasks acting as entry point for upper layers to interact with the MAC layer in a consistent way.

3.1 Physical

802.15.4 supports different physical specifications, with different frequency bands and modulations.
The three different supported bands employ the **868, 915, 2450** Mhz frequencies, with 1, 10 and 16 channel respectively.
All the bands employ the DSSS (*Direct Sequence Spread Spectrum*) technology for the logical channel creation, while the encryption is performed with a *Binary Phase Shift Keying* (B-PSK) technology for the 868/915 bands and

with a *Offset Quadrature Phase Shift Keying* (OQ-PSK) for the 2450 band. That difference in encoding and channel availability lead to different performances, with an archived maximum data rate respectively of *20, 40, 250* Kbps.

3.2 MAC

The MAC supports two different mode of operation for a single device, the *Reduced Functions Device* (RFD) and *Full Functions Device* (FFD).
An RFD device can only act as end-node,while an FFD can act as network coordinator, that provides basic network formation primitives and periodically send beacons to coordinate the end-nodes and the communications.

802.15.4 allows two basic network topology formation, the start topology and the peer-2-peer topology.
Star networks employ a master-slave communication schema, where an FFD acts as the PAN coordinator, driving communications and performing network initiation. All other devices (RFD or FFD) communicate only with the PAN coordinator that eventually route the communication to another node connected to it.
In the peer-2-peer topology, an FFD can communicate directly with another FFD in its coverage range, and can afford messages forwarding for other FFDs, forming a multi-hop network.

The unique PAN coordinator can rule its network in beacon operated mode, or in beacon-less mode, where all the communication are just based on medium contention.
In beacon operated mode, the coordinator synchronizes the other devices with a periodic beacon, and then starts a two phase super-frame structure, which allows the coordinator to sleep for half the super-frame duration.
The active super-frame part is divided in two different sections: the *contention based access period* (CAP) and the *contention free period* (CFP), where *guaranteed time slots* (GTS) are assigned to specific end-nodes.

During the contention based period, all the end nodes that wish to communicate have to perform a CSMA/CA procedure to gain access to the channel, while during the contention free period the coordinator provides a set of GTS to end-nodes which registered their selves during one of the previous CAP periods.

The GTS mechanism guarantees basic Quality of Service implementation and allows node with big amount of data to reserve a fixed amount of the available period to transmit. Communication of coordinator with end-nodes is performed in indirect way.
Coordinator announces pending delivery in beacon messages while end-devices sleep most of the super-frame time and wake up during CAP to request a messages delivery. In a beacon-less network coordinator is up all the time, waiting for incoming messages from end-nodes, that wake up for accessing the channel to periodically poll the coordinator for incoming messages.
If a coordinator has to communicate with other coordinators, it has to act as an end-device, registering for a GTS, or accessing the channel during the CAP.

3.3 LLC

The *Logical Link Controll* (MAC-LLC) sits on top of the low-level MAC, providing multiplexing of transmitted protocols, re-transmission of dropped packets and other flow control services.
802.15.4 modified LLC sub-layer to introduce an interface for accessing lower levels, and to provide a direct access to the MAC from upper layers through a *Service Specific Convergence Sublayer* (SSCS).

3.4 Addresses

802.15.4 supports two types of addresses, *short addresses* (16 bits) and *extended addresses* (64 bits).
Short addresses are used to identify the node inside a specific network and are called *PAN IDs*. A PAN ID is assigned to each node joining the PAN and it is unique in the whole network.
An extended address is assigned to each 802.15.4 compliant node, and globally identifies the device. It is usually called *EUI-64*.
After a node has joined a network, for communicating inside the PAN it can use the short ID in place of the longer extended ID, reducing the overall frame size (48 bits less).

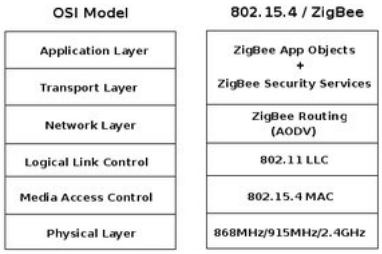

Figure 1: The ISO/OSI and ZigBee stack compared

4 ZigBee

Sometime ZigBee and 802.15.4 are considered to be the same, but conceptually they define really different things.
The ZigBee stack defines additional layers that sit on top of the two 802.15.4 MAC and LLC levels.
Standardization of ZigBee protocol is performed by the *ZigBee Alliance*, an industrial complex composed by many OEMs involved in the wireless device market, that standardize and develop the ZigBee stack.

4.1 ZigBee Standard

The model suggested by *ZigBee alliance*, implemented in commercial ZigBee stacks, reduces the overall complexity of the usual 7 layer stack to a minimal implementation ranging from level 0 (*Physycal*) to level 5 (*Application Support and Security*).

Fig. 1 provides a compared view of the ISO/OSI stack and the one implemented in ZigBee compliant stacks. The decomposition also involves the inner layers, where each OEM implements its (usually proprietary) solutions and features.

Fig 2 illustrate the detailed ZigBee stack, where upper layers are implemented above IEEE 802.15.4 MAC layer. This model complies to the one described by [4], where cross-layering approach is also analyzed in relation to security and reliability concerns.

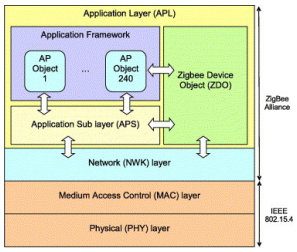

Figure 2: Detailed ZigBee stack layering

The *network layer* (NWK) is in charge of organizing and providing routing over a multihop network, additionally providing network topology management, MAC management, discovery protocol, and security management.

The *application layer* (APL) is intended to provide a framework for distributed application development and communication.
APL comprises the Application Framework, the *zigBee device objects* (ZDO), and the *application sub layer*(APS).
ZDO provides functions that allow services discovery and organization into a distributed application, while APS offers an interface to data and security services to the Application Objects and ZDO.
The Application Framework allows the definition of up to 240 *applications objects* (APO) that are part of the stack, and form the *ZigBee profiles* used to speed up application development and to provide conformity between different producers.

4.2 Network formation and addressing

ZigBee defines three different roles for network nodes, and two different operations level, the usual *Reduced Function Device* (RFD) or *Full Function Device* (FFD), that correspond to the ones in the MAC layer.
A *ZigBee end device* can be a RFD or FFD device, that act as simple communication slave, with no routing nor network initialization capabilities. *ZigBee Router* is an FFD with routing capabilities, while a *ZigBee Coordinator* is an FFD which manage the network topology, that could be the usual star topology, or the more complex tree or mesh.

Network formation happens through the *join procedure*, where a father-son relationship is established.

When a node wants to join a network, the Zigbee network level drives the MAC layer to perform a scan operation. If the desired network identifier (or the identifier dynamically chosen by the network layer) gets found, the MAC layer starts an *association procedure* with the parent node.
The parent node offers a 16-bit identifier to the child (the PAN ID), that will use this ID for all the ongoing communications.

The formation of those father-son relationships forms a tree of nodes, where the *ZigBee coordinator* is in root position, *ZigBee routers* are inner nodes and *ZigBee end devices* are leaves.
This tree-like organization is also used for address assignment, based on the node depth and position.
Given a 16-bit identifier range assigned to each newly joint node, the first address is reserved for the node itself, while the rest is available for future children nodes.

4.3 Routing

The routing algorithm depends on the established network topology.
In the simplest situation, a plain tree-like network, the routing exploits the address assignment schema to determinate if the packet received by the a node (a router or a coordinator) has to be forwarded to one of its child node or to its parent node. This is not the most energy efficient schema, but admits beacon-based routing operations, where the router node emits beacons at fixed intervals, sleeping for the rest of the period. Children nodes communicate in contention based part of the super-frame structure, while parent to child communication is indirect and happens as explained 8 section 3.2.

Mesh based networks require a *routing table* (RT) to be maintained and updated, and do not admit beacon based operations. When trivial routing is not possible (direct routing to a child), the RT is consulted and next hop gets determinated. If no entry for the current destination is available, a routing discovery procedure is performed only if enough energy is available.

Route discovery is performed using a procedure based on the well-known AODV (*Ad-Hoc On Demand Distance Vector*) algorithm[5].
Route discovery is initiated with an *RREQ* request, that is broadcast to the neighbor nodes. The *RREQ* message maintains a weight (based on link qual-

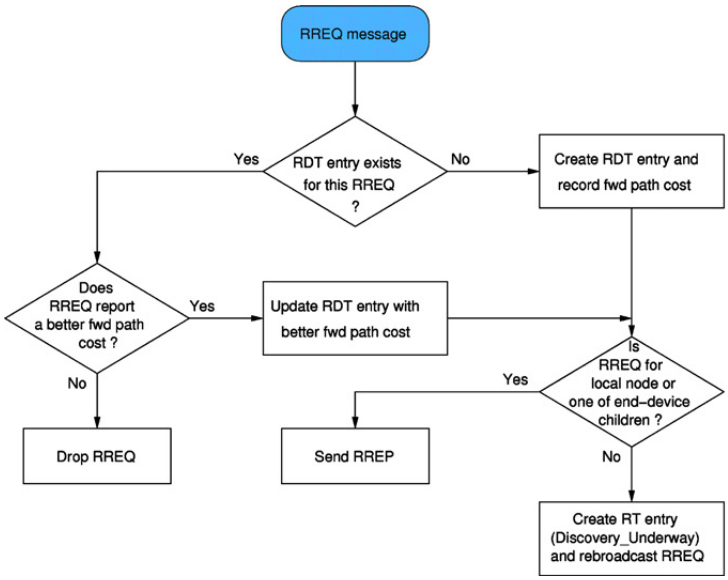

Figure 3: The RREQ algorithm

ity of fixed values) for each traversed connection, and a unique identifier to avoid request doubling.
When a node receives a RREQ request, it consult its routing table, if no entry is found for the destination (or a better path cost is carried along the RREQ) the node issues a new RREQ to all its neighbor. If the RREQ carries a path cost higher that the one stored in the table RREQ is dropped and broadcast stops.

When a node founds the destination is itself or one of its child node, it sends back a RREP with the original request issuer as destinations. Intermediate nodes compares the residual paths cost with the one stored in their RT, and discard or forwards the RREP. When the original nodes receives the RREP back, it stores the next-hop node address in its RT the entry and marks it as 'Active'.
RREQ and RREP procedures are illustrated in **Fig. 3** and **Fig. 4**.

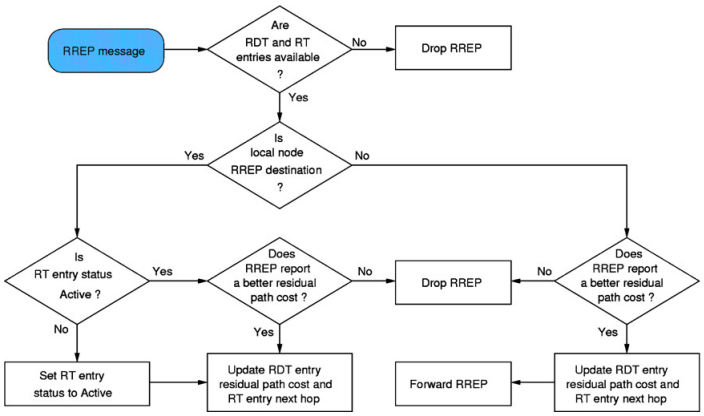

Figure 4: The RREP algorithm

5 IP over LR-PAN

This section presents one major approach, alternative to ZigBee to account network addressing, routing and transport functionality in a low power, low range wireless network based on the 802.15.4 MAC and LLC layers.
Alternatively to ZigBee, this approach aims to solve from the basis the interconnection problem with existing infrastructure, basing all the network and transport functions in the WSN on the standard TCP/IP protocol stack.
For the requirements of address auto-configuration and for the number of involved nodes -that is expected to grow very fast in a context where sensing node ubiquity is a fundamental component- the IP version we are referring to is obviously the version 6 (IPv6) as specified in IETF RFC 2460[12] and 4291 [13].

IP stack for memory constrained devices

The adoption of the IP protocol "as is" cannot be afforded for LR-WPAN for several reasons, some of them superseded by the progress made by the research and some of them still to be fully solved.
Firstly, the TCP/IP stack has always been considered too heavyweight and was though to require too much resources both in terms of code size and run-time memory to fit in a standalone embedded system.
Many existing implementations targeted to memory constrained devices, such as a wireless sensor node, are designed for a single architecture and provided

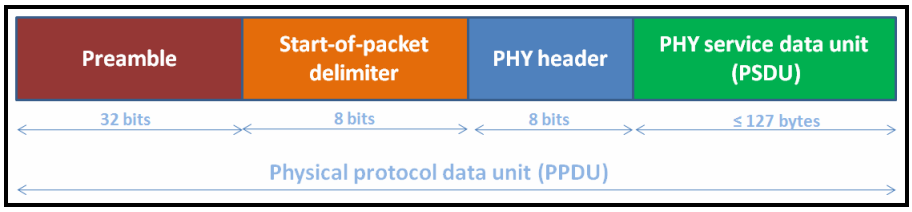

Figure 5: The 802.15.4 Physical protocol data unit

by OEMs, often saving space and memory removing some TCP mechanism such as congestion control (Atmel TCP/IP), targeting a specific application (such as web control or monitoring interfaces) or requiring that the correspondent node is running a full featured IP stack, preventing peer-2-peer communications between nodes.

The **uIP** implementation [7] solved part of the mentioned issues, providing a full-featured TCP-IP stack affordable also for very small and resource constrained devices. The uIP implementation has been ported to many devices an architectures and is available as network stack in the *Contiki operating system*.
Furthermore all the source code and documentation are freely available, under a 3 clause BSD-like license.

5.1 Physical packet size and Fragmentation

Another challenge that has to be faced when considering the introduction of IP protocol in such a context, is the frame size problem, due to the small packet size supported by 802.15.4 MAC layer.

Fig 5 shows the *Physical Protocol Unit Data* for 802.15.4 networks.
This unit determinate the maximum size of a packet sent at physical level, where other stack's layer are expected to encapsulate their header and data. *PPDU* is composed of a synchronization header consisting of a 4 byte preamble of binary zeros (for chip and symbol synchronization) and a single byte for the start-of-packet delimiter (SFD) and another one for physical layer header. Remaining 127 bytes are reserved for the payload, that ranges from a 5 byte length for ACK, to 127 for full data packets as specified in section 6.3 of *IEEE Std. 802.15.4-2003* [11].

Given a *Service Data Unit* size of 127 available bytes, 25 bytes are needed for Frame control, sequence number and addressing fields, only 102 octets are available in the MAC payload. But since Link-layer security imposes further overhead, which in the maximum case (21 octets of overhead in the AES-CCM-128 case, versus 9 and 13 for AES-CCM-32 and AES-CCM-64, respectively)[9], the total number of bytes left available for data packets is 81.
Considering that IPv6 header is 40 bytes long, and TCP and UDP impose an additional overhead of 20 and 8 bytes respectively, very few bytes are available for data to be transmitted.

Additionally IPv6 requires all links to support a minimum transfer size (MTU) of 1280 bytes per packet.
Due to the limited available size of the physical protocol unit, a fragmentation and reassembly service is needed. In addition, not all the devices has enough memory and processing power to handle 1280 byte packets, anyway interoperability between nodes has to be guarantee.

5.2 Routing and addressing

When dealing with IP over 802.15.4 MAC/PHY a double addressing schema has to be taken into account. IP maintains its own addressing, while the MAC layer employs the short and extended identifiers.
Those different methods could be employed to define a single identifier for the node.

Low power devices communicating over 802.15.4 MAC layer usually have a very low duty cycle of activity. It is possible that they sleep for a long time, and during this period are unable to transmit or receive data.
This behavior can be problematic when pro-active routing schema are employed in the network, because a node that was awake when the routing tables have been updated, can be sleeping in a successive time, causing the other device to re-issue a routing table update procedure, wasting time and energy.

6 6LowPAN

6LowPAN is a proposal for a set of standardized methods and techniques, for enabling networked embedded device, to use IPv6 as common communication protocol even in LR-WPAN context.
Standard has been created by *IEFT* [10], is open available and promise an implementation model that guarantees interoperability with legacy IP-based network infrastructures and between different devices of different manufacturers.

6LowPAN essentially consist in an adaption layer, that makes possible for IPv6 frames to be transported over 802.15.4 packets implementing fragmentation and reassembly support, adopting compression techniques for headers size reduction and imposing methodologies for IPv6 address mapping upon MAC layer identifiers.

6.1 Dispatch type and headers

6LowPAN header specify an encapsulation stack that involves 4 different header types: *Mesh (L2) addressing, hop-by-hop options (including L2 broadcast/multicast), fragmentation, and payload*. This headers has to appear in the specified order when more than one is used.

In order to distinguish between different headers in use in the current frame, a special encoding is used for the first byte, called *Dispatch Byte*.
It is divided between *Dispatch type*, specified in the first two bytes and *Dispatch header* that is specified in the remaining six.
This byte defines the exact meaning of the data that follow, in order to indicate in which way they have to be handled by the stack.
The 4 different headers can co-exist in the same frame, and have to be combined as specified in section 5 of [10]

The initial dispatch byte composed by dispatch type and dispatch header is defined by the specifications in order to leave space for future enhancements and extensions. At the moment the only meaningful combinations are:

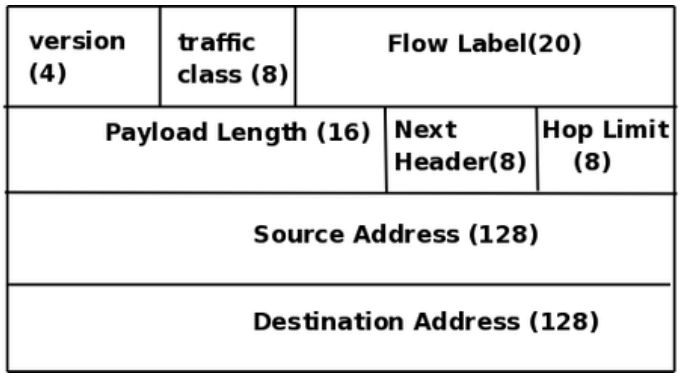

Figure 6: The uncompressed IPv6 header

00 xxxxxx	NALP	*Not a low pan frame*
01 000001	IPv6	*Uncompressed IPv6 header follows*
01 000010	LOWPAN_HC1	*HC1 compressed IPv6 header follows*
01 010000	LOWPAN_BC0	*BC0 broadcast*
01 111111	ESC	*Additional dispatch bytes follow*
10 xxxxxx	MESH	*Mesh header*
11 000xxx	FRAG	*Fragmentation header (first)*
11 100xxx	FRAGN	*Fragmentation header (subsequent)*

6.2 Header compression

Fig. 6 shows an uncompressed plain IPv6 header, as specified by [12], section 3. The plain header is 40 bytes long, and introduces support for dasy-chaining of additional header components, traffic flow labeling and integrated security. The header format has been simplified respect the IPv4 header, but its size imposes the definition of reduction techniques for reasons exposed at page 14, section 5.1.

The principles that drove the design of the LOWPAN_HC1 compression schema have been:

- *Omit any header fields that can be calculated from the context, send the remaining fields unmodified*
- *Nodes do not have to maintain compression state (stateless compres-*

sion)

- *Support (almost) arbitrary combinations of compressed /uncompressed header fields*

That means that information such as *traffic class* or *flow labeling* are always 0 unless specified. In addition all fields such as *protocol version* or *frame length* that are fixed or can be inferred from lower layers can be omitted, while the only field that cannot be compressed is the *Hop Limit* counter. All the fields that need to be carried in-line (uncompressed) follows the compressed header, and the specific bit combination in the HC1_header will be set accordingly.

After the (eventual) uncompressed fields the only headers that can follow are UDP, ICMP or TCP. headers
A specific compression algorithm, called LOWPAN_HC2 takes care of transport layer headers compression. HC2 is conceptually the same as HC1, as it tries to leave unspecified all the information that can be inferred or calculated. The only uncompressed field is the checksum, that has to be carried in-line.

6.3 Mesh header

IEEE 802.15.4 admit mesh routing and mesh network, but it does not provide any method to directly support so.
Instead all the work is remanded to a specific mesh routing algorithm that populates routing tables and defines the route to a node to another. In this way, two communicating nodes do not need to be directly connected, but need a *multi-hop* method to let the message flow from an *originator* to a *final destination*.

A node that wishes to communicate with another one which is not directly connected to, has to insert the appropriate *Mesh dispatch header* and insert its own link-layer address as the *originator address* and the corresponding node's link-layer address as *final destination address*. It then sets its address as 802.15.4 source address and next-hop node address as 802.15.4 destination address, the packet is then transmitted to the forwarder.
A special counter, called *Hops left* is inserted and decremented each forwarding until it becomes zero. When this happens the packet is discarded.

A node receiving a frame with mesh header set has to examine the *final destination* address to determinate the next-hop node. If the *final destination* is itself the frame's payload is consumed, otherwise the packet is forwarded only if the decremented *hops left* field is not zero.

6.4 Fragmentation Header

For the several reasons explained in section 5.1, fragmentation of IPv6 frames is very likely to happen, and a specific method for correctly handle this situation is provided by the use of *fragmentation dispatch* and *fragmentation header*.

Since a fragmented packet has to be reassembled on the destination side, all the fragments of the same packet are marked with the same *datagram tag*, that must be the same in all the frame's parts. In the initial packet the overall frame size is specified in the *datagram size* field.

All the fragments, except the first, must contain a *datagram offset* that specify in what sequence the fragments has to be reassembled. Receiver tries to reassemble all the incoming frames, until the initially specified *datagram size* is reached.
If a disconnection event happens, all the on-going frame reassembly has to be discarded and all the incoming buffer flushed.

6.5 Stateless address auto-configuration

Section 5.2 specify why a suitable methodology for cross-layer address auto-configuration is desirable for a LowPAN protocol.

All the 802.15.4 compliant devices are provided with an extended address called *EUI-64*. This address globally identifies the device and can be used for 6LowPAN address formation. The network address can be composed starting from *EUI-64* exactly in the same way IPv6 address are composed with Ethernet addresses.

If a node wishes to use the 16 bit short address, a pseudo 48-bits long address is composed using the PAN-ID in the following way:

16_BIT_PAN:16_ZERO_BITS:16_BIT_SHORT_ADDRESS

Again, as specified by [14], the global network address is composed staring from this 48 bit address in the usual way.
When this address is used, the *Universal /Local* bit must be set to zero to indicate that this is not a globally valid address.

7 Interoperability, Internetworking and Interconnection

The real challenge, for the yet-to-come sensor networks, can be predicted being their ability to transparently integrate into the current infrastructure, that is totally based on, and regulated by, the TCP/IP protocol stack.
The existing infrastructure have to be considered, not only as the main interface that have to be used to interact with the sensor network, but also as a backbone to be exploited for creation and deploy of geographically distributed networks of sensors and devices.

The digital information highways created in these years, used by billions of 'regular' devices during our everyday life, have to be opened to small, cheap and smart systems, able to organize them-selves and to operate as autonomous entities.
They will not only communicate with other nodes in the same WSN, but have to be accessible from everywhere, must be able to exchange information in a point-to-point fashion, and will require everyday a more direct, more simplified way for accessing the digitalized world that has already been created and is everyday expanding.

The promise of full interoperability and transparent networking between this new citizens of the network, and the previously installed infrastructure is the next evolving step that has not yet been totally accomplished.
To supersede this situation different solutions have been proposed, and this evolving pressure lead to standardization of protocol such as 6LoWPAN that have been designed for lowering the required effort for a full interconnection realization.
Some other protocols, such as ZigBee, are born under a different approach, and today the standardization entities are trying to correct the road, but

currently, the available solutions still suffer from the original design issues.

7.1 ZigBee extension devices

In the previous sections the analysis made about the ZigBee stack and related applications has been accomplished considering the single, isolated network context, where the WSN had to be considered as a standalone entity with no meaning to communicate with the external world or to exploit the existing infrastructure. The problem of the isolation of the single WSN has been affronted also by the ZigBee alliance, purposing the following device taxonomy [6] to archive full interconnection between IP networks and ZigBee, or at least exploit the existing infrastructure for expanding the single WSN.

- **ZigBee Gateways**:
  Allow the interconnection of the ZigBee network and IP devices through an abstracted interface on the GateWay side. The gateway could perform translations to other industrial or commercial protocol, for data storage and analysis.

- **ZigBee Bridge**:
  ZigBee bridges, or *ZigBee Expansion Devices* (ZED), aims to interconnect different ZigBee networks through an IP based infrastructure. The ZED do not perform any kind of bi-directional translation between the protocols, but just deliver information to one network to another.

An example of a *ZigBee gateway* design and implementation can be found in [15] by Wang, et al. Their solution propose a gateway structure for the realization of IPv6-802.3 /ZigBee-802.15.4 networking, but concepts are extendable to other technologies such as IPv6-802.11 etc.
The proposed solution create and overlay network employing of a double ZigBee-IPv6 multicast addressing schema. After the network unification has been accomplished, a method for header rewriting based on *IP Switching*[16] algorithm has been implemented

7.2 IP-NET

IP-NET is a proprietary solution developed and employed by *Hellicom Inc* in some commercial products. This solution features a double stack approach, where 6LoWPAN and ZigBee are laid out on the same MAC layer.
Anyway this solution does not present a real interconnection possibility, since only one of the two stack can be used at the same time.

7.3 ZigBee-IP

ZigBee alliance published in late 2010, 20 slides where it was announcing the definition (or a proposal) of the next ZigBee specifications. ZigBee-IP is defined a 'super-specification' that aims to incorporate other standards and define an interoperability set between them, instead of specifying its own network and transport layers as done with ZigBee PRO specifications. The standard's basic components will be:

- IEEE 802.15.4-2006 MAC/PHY
- IETF 6lowpan-hc adaptation layer
- IETF 6lowpan-nd neighbor discovery
- IPv6 network layer
- TCP/UDP transport layer
- IETF ROLL RPL routing
- PANA/EAP/EAP-TLS/TLS security

If this will really be the next ZigBee protocol specification, it will represent a dramatic shift of prospective by ZigBee alliance, that will adopt *IEFT* and *IEEE* standards 'as is', and will instead concentrate on their co-existence and management.
Currently no official informations are available, but a proposed stack could hypothetically be as described by **Fig 7.3**.

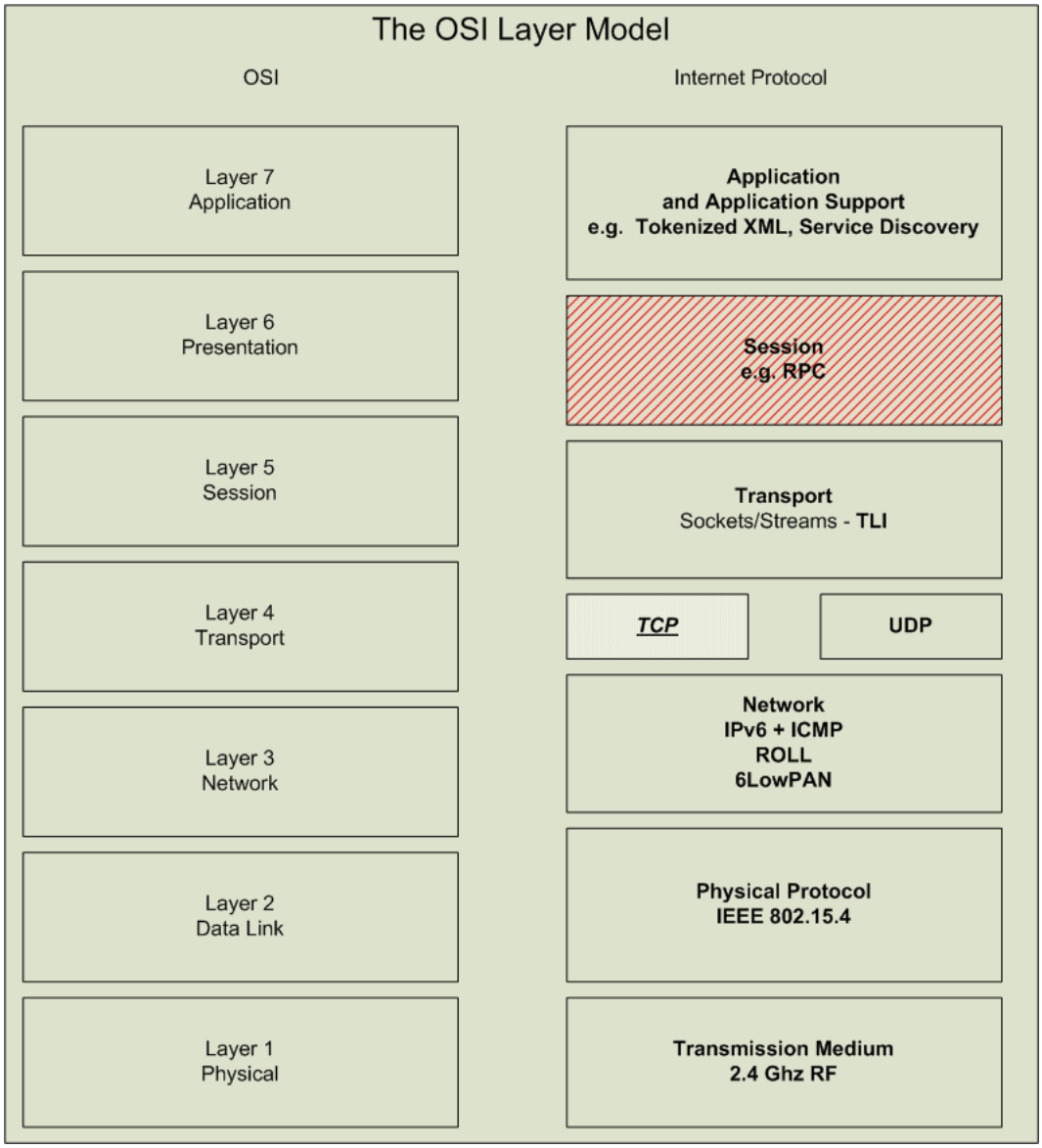

Figure 7: The proposed ZigBee-IP stack

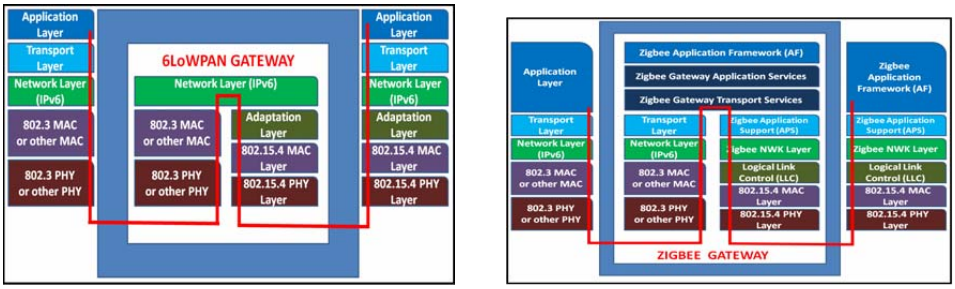

Figure 8: IP-6LowPAN gateway and IP-ZigBee gateway architectures

7.4 IP-6LowPAN

6LowPAN has been developed for being natively interoperable with traditional IPv6 based infrastructure, so there are no special needs for header rewriting or overlay network creation.
Anyway, a light adaption layer has to be deployed under the network layer (IPv6).
This adaption layer has to take care of header compression and un-compression, and has to fragment the packets that are traveling to the 6LowPAN network, and reassemble the one that are traveling in the opposite direction.

Furthermore, the adaption layer is only involved with sub-L3 mechanism, and does not need to perform any manipulation at higher levels.
A comparison between an IP-6LowPAN gateway and a IP-ZigBee gateway is proposed in figure 8.

References

[1] IP for Smart Objects. A. Dunkels, J.P. Vasseur. The IPSO White Paper

[2] IEEE Transactions on Communications, Hubert Zimmermann, vol. 28, no. 4, April 1980, pp. 425 432

[3] Low-rate wireless personal area networks,Jos A. Gutirrez, Edgar H. Callaway, Raymond Barrett, ISBN 0-7381-3557-3

[4] Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards, Paolo Baronti, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, Y. Fun Hu, ELESEVIER, December 2006

[5] Ad hoc On-Demand Distance Vector (AODV) Routing. C. Perkins , E. Belding-Royer, S. Das. (July 2003), IETF. RFC 3561.

[6] Gateways: Beyond the Sensor Network, Patrick Kinney, Kinney Consulting LLC, ZigBee Alliance, 2004

[7] Full TCP/IP for 8-Bit Architectures, Adam Dunkels, Swedish Institute of Computer Science

[8] IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, N. Kushalnagar, G. Montenegro, C. Schumacher, August 2007, IEFT RFC 4919

[9] Interconnection between 802.15.4 Devices and IPv6: Implications and Existing Approaches, Md. Sakhawat Hossen, A. F. M. Sultanul Kabir, Razib Hayat Khan and Abdullah Azfar, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1, January 2010

[10] Transmission of IPv6 Packets over IEEE 802.15.4 Networks, G. Montenegro, N. Kushalnagar,J. Hui, D. Culler, September 2007, IEFT RFC 4944.

[11] IEEE Computer Society, "IEEE Std. 802.15.4-2003", October 2003.

[12] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", IEFT RFC 2460, December 1998.

[13] R.Hinden and S. Deering, "IP Version 6 Addressing Architecture", IEFT RFC 4291, February 2006

[14] Transmission of IPv6 Packets over Ethernet Networks, M. Crawford, IEFT RFC 2464.

[15] Internetworking Between ZigBee/802.15.4 and IPv6/802.3 Network, Reen-Cheng Wang Ruay-Shiung Chang Han-Chieh Chao

[16] Newman, P., Lyon, T., and Minshall, G. Flow Labelled IP: A Connectionless Approach to ATM, In Proceeding of IEEE Infocom 1996, (San Francisco, CA, USA, March 24-28, 1996), 3, 1251-1260.

[17] The ZigBee IP Stack IPv6-based stack for 802.15.4 networks, R. Cragie, ZigBee alliance, 2010