

# The 6LoWPAN Architecture

Geoff Mulligan  
Proto6  
+1 719 593 2992  
geoff@proto6.com

6LoWPAN Working Group  
Internet Engineering Task Force  
6lowpan@lists.ietf.org

## ABSTRACT

6LoWPAN is a protocol definition to enable IPv6 packets to be carried on top of low power wireless networks, specifically IEEE 802.15.4. The concept was born from the idea that the Internet Protocol could and should be applied to even the smallest of devices. The initial goal was to define an adaptation layer – “IP over Foo” to deal with the requirements imposed by IPv6, such as the increased address sizes and the 1280 byte MTU. The final design takes the concepts used in IPv6 to create a set of headers that allow for the efficient encoding of large IPv6 addresses/headers into a smaller compressed header – sometimes as small as just 4 bytes, while at the same time allowing for the use of various mesh networks and supporting fragmentation and reassembly where needed. This paper describes some of the underlying assumptions and decision points made during the development of 6LoWPAN and how the “stacked header” concept is applied so that in using the protocol you only have to “pay for” what you use. It concludes with open problems and challenges for further development and research.

## Categories and Subject Descriptors

C.2.2 [Network Protocols]: Protocol Description

## General Terms

Documentation, Design, Standardization

## Keywords

Internet, Internet Protocol, IPv6, 6LoWPAN, Sensors, Sensor Networks.

## 1. INTRODUCTION

When thinking about sensor networks the first thing that does not come to mind is the use of the Internet Protocol (IP). IP has always been considered the protocol for Local Area Networks or Wide Area Networks, PCs and Servers. It was not thought appropriate for sensor networks or Personal Area Networks, Premise Area Networks, Perimeter Area Networks (PANs) and the sensors themselves because of the perception that it was too heavy weight for these applications. Recently there has been a large community starting to rethink many of the misconceptions about IP and have embarked on the development and standardization of

6LoWPAN – with the question “Why invent a new protocol when we already have IP?” 6LoWPAN defines how to layer IP version 6 (IPv6) over low data rate, low power, small footprint radio networks (LoWPAN) as typified by the IEEE 802.15.4 radio.

Starting in 2001, during some of the original discussions around defining some “standard” protocols for IEEE 802.15.4, which was being finalized at that time, I proposed to use the Internet Protocol suite to various groups. Though Zigbee and others did not accept the proposal, I found other efforts such as “Internet 0” [2] from MIT’s Center for Bits and Atoms and tools such as Robust Header Compression from the ROHC Working Group in the IETF [1] and became convinced that IP not only should be applied to small RF connected sensors, detectors and devices, but in fact could be used. The continuing design and development was accomplished by a cadre of engineers, many from the IETF and the IETF 6LoWPAN Working Group and not solely by the author. The term “the working group” is used in this paper to represent these scientists and engineers, while “we” represents the authors of this paper. The name 6LoWPAN was born from the combining of IPv6 and Low Power Wireless Premise Area Networks, but more importantly so that the name would sort to the top of the working groups list.

## 2. Why IP based sensor networks

Utilizing IP in these networks and pushing it to the very edge of the network devices flattens the naming and addressing hierarchy and thereby simplifies the connectivity model. This obviates the need for complex gateways that, in the past, were necessary to translate between proprietary protocols and standard Internet Protocols and instead can be replaced with much simpler bridges and routers, both of which are well understood, well developed and widely available technologies. Additionally by using IP based protocols users can employ tools already developed for commissioning, configuring, managing and debugging these networks. It is not necessary to develop an entire new set of tools to deal with proprietary protocols and applications and to develop the skills necessary to use those tools. Since the entire network is IP based, programmers and administrators do not need to learn and develop expertise in a new programming paradigm. The network is programmed and functions the same as other Internet applications. When it is necessary to debug the network or application, engineers will already have the experience of debugging IP based network glitches.

Since the underlying protocols are the IP suite, the sensor application designers can make use of existing standards to speed up the design and development process. Rather than having to start from scratch to create a management protocol, it would be prudent to investigate using SNMP (Simple Network Management Protocol) RFC1067; rather than inventing a new mechanism to find services that are available on the local network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*EmNets2007*, June 25-26, 2007, Cork, Ireland.

Copyright 2007 ACM 1-58113-000-0/00/0004...\$5.00.

it might be better to use SLP (Service Location Protocol) as defined in RFC 2608. There are a plethora of other existing standard protocols that may be applicable to these sensor applications, such as UDP, TCP, ICMP, DNS, and TFTP. There are also many standardized higher level services such as Load Balancing, Caching, Firewalling, and Mobility that could be easily leveraged in this space. This concept might even be extended to building the mesh routing between the nodes on top of IP rather than below – this will be discussed later in the paper. IP also brings the advantage that a number of legacy sensor and control protocols have already been adapted to run over IP. Modbus and Modbus TCP, BACNet has BACNet/IP [1], Profibus has the TCP/IP adapter, and Foundation Field Bus has options to use Ethernet/IP or HSE - the basic difference is that one uses TCP and the other uses UDP to transport packets [10], respectively.

These and other “industrial standards” could be used directly and would not require a translation mechanism at the gateway in order to provide functionality and connectivity. The legacy applications already running over the installed IP networks could be simply ported to encompass these IP capable sensor and controller nodes. For example, it is simple to develop a demonstration of standard Modbus sensors and controllers interconnected using Modbus TCP over 6LoWPAN. And finally many architectures and protocols are already deployed and tested for dealing with the management and security needs of these networks.

### 3. 6LoWPAN – Not too big

6LoWPAN is a developing standard from the Internet Engineering Task Force (IETF) 6LoWPAN Working Group and it was designed from the start to be used in small / pico sensor networks. It is not the case that IP is too “expensive” to use; expensive, in this case, being a measure of code size, protocol complexity, required configuration infrastructure or header / protocol overhead. Implementations of 6LoWPAN easily fit into 32K flash memory parts (typically smaller than Zigbee or other protocols). See Table 1 for a comparison of RF Network Stacks – the reported size of the Zigbee stack varies significantly from document to document, with reports as low as 32K [5] and as high as 90K [8][13][4]. The information about the 6LoWPAN stack is from an implementation I have written for the Freescale 13192 radio and GT60 micro-controller.

The working group has also been able to eliminate the necessity of configuration servers (DHCP and NAT) by utilizing the Zero-Conf and Neighbor Discovery capabilities of IPv6. By basing the protocol on IPv6 the working group also was able to define a unique stateless header compression mechanism that allows the transmission of IPv6 packets in as few as 4 bytes. The fears around needing to transmit a 40 byte IPv6 header can be allayed while at the same time allowing the use of the immense IPv6 address space. [The IPv6 protocol uses 128 bit addresses as opposed to 32 bits in IPv4. With 128 bits IPv6 can provide  $56 \times 10^{27}$  addresses per person or  $667 \times 10^{21}$  per square meter on the Earth.]

The working group has written a problem statement document, “6LoWPAN: Overview, Assumptions, Problem Statement and Goals” [9], which outlines the area of focus for the working group and a format document, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks” [11], which defines the specific header encoding and header compression mechanisms used in adapting IPv6 into 802.15.4 frames. Both documents are soon to be published as Proposed Standard RFCs. It was not an original goal of the working group to choose or develop a mesh network / routing protocol but to instead define an encoding mechanism that was layer 2 and 3 routing protocol agnostic. The resulting design provides a unique concept in that you only “pay” for what you use – “pay” meaning header overhead and processing. Rather than defining a single monolithic header, as was done for IPv4 and Zigbee, the working group decided to use the same techniques that are used in IPv6 – stacked headers. In this way if a device is sending short packets directly to another node it does not “pay” for header fields for mesh networking or fragmentation and is only required to send the minimum encoding headers. There are currently four basic header types defined in the standard: Dispatch Header, Mesh Header, Fragmentation Header, and the HC1 Header (IPv6 Header Compression Header). For the simplest case the only necessary headers are a Dispatch Header, an HC1 header and the compressed IPv6 header – the total of which is a svelte 4 bytes. Only when you send large packets do you need to include a Fragmentation Header just as only when you are using a mesh network layer under 6LoWPAN do you need to include a Mesh Networking header. The cost of implementing 6LoWPAN is less than or at most equal to other similar protocols and the overhead for the most common packets are much less than other protocols.

**Table 1. Stack Comparison**

	<b>Zigbee</b> [4,6,11,3]	<b>Zensys</b> [14]	<b>6LoWPAN</b>
Code Size with mesh	32K to 64K+	32K	22K
Code Size w/o mesh	Not Possible	Not Possible	12K
RAM requirements	8K	<2K	4K
Header Overhead	8 to 16 bytes	Proprietary	2 to 11 bytes
Network Size	~65K	232	$2^{64}$
RF Radio support	802.15.4	Proprietary	802.15.4 ++
Transport Layer	None	Proprietary	UDP/TCP
Mesh Network Support	Zigbee	Zensys	Many
Internet Connectivity	Zigbee Gateway	Zensys Gateway	Bridge/Router

This reduced overhead translates into energy savings.

Even though over 60% of the energy used by these devices is spent waiting for the radio and micro-controller to “warm up” [3][12] and for most sensor applications over 90% is wasted waking up and listening when there is nothing to be heard [12] it is still important to efficiently utilize the limited available energy or bandwidth. For 6LoWPAN, when implemented on a TI MSP430 micro-controller and CC2420 radio the energy overhead ranges from 2.8% for very small data payloads (10 bytes) to under 2% for data payloads near frame capacity [3]. See Figure 1 for a graph of both local and global transmission header energy costs. These values are calculated by adding up the total energy to transmit the packet, including the time to wake up the micro-controller and radio, compared to the added overhead of transmitting the 6LoWPAN header. For example, using the MSP430 and CC2420, radio startup and configuration, clear channel assessment and radio turn around require approximately 7.5ms and to transmit 10 bytes of data with a standard 6LoWPAN header requires just .64ms or less than 10% of the total energy cost to send the packet. The extra time for the 6LoWPAN header is just .32ms. [In Figure 1 the difference in overhead between local and global packets is the ability, as described later, to completely remove / compress the IPv6 128 bit addresses for local transmissions.]

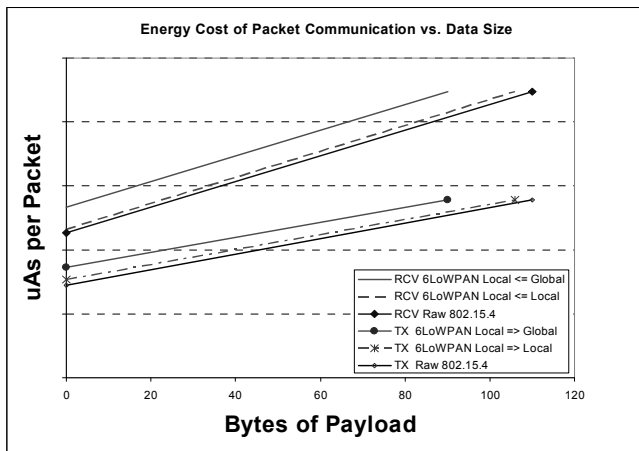


Figure 1. 6LoWPAN Energy Overhead. [3]

#### 4. 6LoWPAN – In Detail

Each 6lowpan header includes a type identifier and the most common headers are identified with a predefined prefix for each. The Dispatch Header (1 byte) is used to define the type of header to follow. The dispatch header is identified by the first two bits set to either 00 or 01. In order to provide a means for coexistence with non-6LoWPAN networks, the bit pattern 00 is reserved to identify these non-6LoWPAN frames. The remaining 6 bits indicate if the following field is an uncompressed IPv6 header or an HC1 header defining the IPv6 compressed header. Only 5 of the 64 dispatch header types have thus far been defined. The special value of all ones indicates that the header contains an additional byte to allow for 256 more header types.

The Mesh Header (4 bytes) is used to standardize how to encode the hop limit and the link layer source and destination of the packet. Since 802.15.4 allows for the use of 16 bit short addresses in addition to the standard 64 bit addresses, the Mesh Header includes two single bit fields to indicate if the originating or final address is a short or long address. The “hops left” field is a 4 bit field used to limit the number of intermediate hops between the source and destination. Though the 15 hop constraint of these 4 bits might be considered sufficient for today’s sensor networks, the working group felt it was important to be able to accommodate deeper networks. As a result the value of 0xF (all ones) was reserved to indicate that an extra byte is included allowing for network depths of up to 255 hops.

The Fragmentation Header (4 bytes for the first fragment and 5 bytes for subsequent fragments) supports the fragmentation and reassembly of payloads larger than the size of the 802.15.4 frame (102 bytes of payload) [7] and includes fields to specify the size of the original datagram as well as a sequence number for ordering the received packets. The datagram tag field is used to identify all of the fragments of a single original packet. Even though for most sensor networks the application would be aware of the underlying link layer frame size limitation, in order to be compatible with the IPv6 specification the protocol was required to be able to support a minimum MTU of 1280 bytes. Therefore it was necessary to include a fragmentation and reassembly mechanism. See Figure 2 for the layout of 6LoWPAN headers.

01	000001	IPv6 Uncompressed		
01	000010	IPv6 HC1 Compressed Encoding		
01	111111	Additional Dispatch byte		
Dispatch Header				
10	O	F	Hops Left	Orig. Address, Final Address
Mesh Header				
11	000	Datagram Size		Datagram Tag
First Fragment Header				
11	100	Datagram Size		Datagram Tag
Datagram Offset				
Subsequent Fragment Header				

Figure 2. 6LoWPAN Header Layout

Utilizing these separate headers it is possible to “stack” them in combination for the specific needs of each packet and network. The following figure (Figure 3) shows three examples the stacked header: 1) the simple case of a small compressed IPv6 packet being transmitted in a point to point or star network – only the dispatch header and HC1 header are included in the 802.15.4 payload comprising two bytes; 2) the first fragment of a large packet again being transmitted in a point to point or star network – where only the fragmentation, dispatch and HC1 header are necessary, made up of 5 bytes; and 3) a small packet being transmitted over a mesh connected network – only the mesh, dispatch and HC1 headers are used requiring 6 bytes.

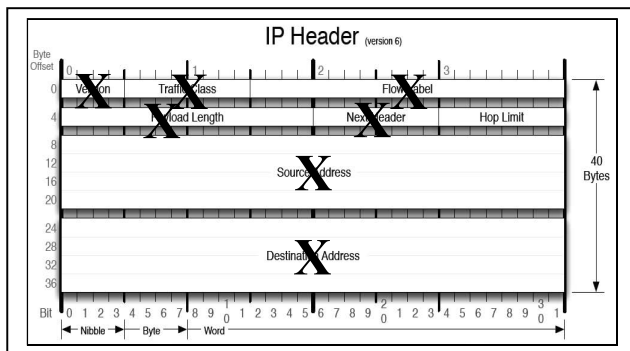
Frame Hdr	Dispatch Hdr	HC1 Hdr	IPv6 compressed Hdr	UDP Hdr	Application Data
Point to Point Small Packet					
Frame Hdr	Frag Hdr	Dispatch Hdr	IPv6 compressed Hdr	UDP Hdr	Application Data
Fragmented Packet Large Packet					
Frame Hdr	Mesh Hdr	Dispatch Hdr	IPv6 compressed Hdr	UDP Hdr	Application Data
Mesh Transmitted Packet					

**Figure 3. Examples of Stacked Headers**

This HC1 header allows for the stateless header compression of the IPv6 header. To accomplish this goal the protocol uses a combination of the following facts:

- the low order 64 bits of an IPv6 address (the link local address) can be the device's MAC address
- the 802.15.4 frame carries these MAC addresses
- a number of the fields in the IPv6 header are static.

Combining all of these features allows the protocol to compress the standard 40 byte IPv6 header down to just 2 bytes (including the HC1 Header byte) for most intra-PAN packets. All of the rest of the fields can be reconstituted without any state information at any of the receiving or intermediate nodes. This is different from standard stateful header compression used in IPv4 where a number of packets must be exchanged between the source and destination before enough state is built up to allow for the compression and should either end lose synchronization (by missing or dropping a packet) the process must restart. Additionally by assigning the link local address to the device's MAC address 6lowpan can use Stateless Address Autoconfiguration (Zero-conf) and eliminates the need to infrastructure servers like DHCP servers. See Figure 4 for a representation of an IPv6 header and fields elided via HC1 compression.



**Figure 4. IPv6 Elided Headers**

The stacked header approach has one other distinct benefit. It allows the protocol to be extended for new header types. To define a new mesh network layer all that is necessary is to request an allocation for a new Dispatch Type from IANA (Internet

Assigned Numbers Authority) and the new header can be used within the 6LoWPAN protocol. This capability can be used by any 802.15.4 network (including Zigbee) to solve a problem not addressed by IEEE. During the design of the 802.15.4 standard, IEEE failed to include a Protocol Type field that would disambiguate the various network protocols riding on top of 15.4 frames. It is possible for two different protocols to collide in their bit formats in the 15.4 payload. Even as the 15.4 standard was being updated from 802.15.4-2003 to 802.15.4-2006 the IEEE failed to fix this problem.

## 5. Route Over vs. Mesh Under

One of the active areas of interest in 6lowpan is mesh networking for sensor networks. Some of the main reasons for this are the limited RF footprint of the low power radio – a mesh seamlessly extends the device's reach and the need for simple deployment and configuration. Requiring the installer to perform an RF site survey and use complex tools to setup and located the sensor or controller nodes would be untenable. A mesh simplifies this commissioning by providing multiple paths between nodes, so all that should be necessary is for the node to be able to communicate with just one other node, any node, in the mesh. Additionally a mesh provides improved reliability though path diversity. If one link fails because of signal fade or multi-path interference another path can be used instead. Rather than using antenna diversity to mitigate multi-path a mesh network provides this same robustness via path diversity. An extremely interesting alternative to the current research in mesh routing is presented when utilizing 6LoWPAN. Rather than building the mesh network under the IP layer (*mesh under* or layer 2 mesh), it is possible to build the mesh network above IP using standard or new routing protocols (*route over* or layer 3 routing).

When originally designing 6LoWPAN, the working group assumed a *mesh under* philosophy as a simplifying assumption. *Mesh under* provided a broadcast network abstraction to the IP layer – the underlying network could be viewed as a slow, high latency, high jitter, lossy, (lousy) Ethernet, but IP was designed to run in this type of environment. The IP layer could assume that all nodes within its subnet were directly reachable - send down a packet to the lower link layer and it will make its best effort to deliver it – the IP model did not need to change. There are some significant downsides to this approach though. No longer can you use various IP network diagnostic tools, such as traceroute / tracepath and some SMTP based diagnostics. Since every node in the mesh is one hop away, from the viewpoint of IP, traceroute is not able to show the path through the mesh and IP based source routing is no longer applicable. Since the underlying mesh is not updating or utilizing an IP routing table it is not possible to use standard tools to query the routing tables. Therefore new traceroute type tools and routing table lookup tools must be designed, developed, deployed in order to diagnose network failures.

An alternative is to use either currently available routing protocols or new routing protocols designed to be more efficient for low power and limited bandwidth networks. These protocols would manage the IP routing tables to let IP “route” packets between the various nodes. In a very real sense the entire Internet is an extremely large mesh network. IP routing works by separating the routing engine and forwarding engine into separate distinct functions. The routing function is responsible for maintaining the

routing tables and the forwarding function (IP) examines the routing table for a “path” to forward the packet. Sometimes the routing table is as simple as a single “default route” entry. This would be completely sufficient for an 802.15.4 Reduced Function Device (RFD) [7] or edge node with a single parent. More recent versions of IP allow for multiple default routes, which would provide the path diversity previously mentioned. One of the benefits of this design is that tools like traceroute and strict source or loose source routing would continue to function as valuable tools for mesh network diagnostics. Two issues need to be addressed though. One is the ability for the routing protocol to be able to query the radio device for various parameters and characteristics, such as battery life, RAM availability, link performance and other information in order to be able to properly advertise and utilize route information. Another is, understanding how one device per subnet might impact the IP model. Since IP routes between subnets (at least as currently defined) this routing design would require each node to be in its own subnet. Even though IPv6 offers nearly an infinite number of subnets, investigations into the ramifications of this choice are necessary. One interesting feature is that it is not an “either or” choice, but actually both could be used. The sensor network could be comprised of many small *mesh under* networks interconnected with a *route over* hierarchy.

## 6. 6LoWPAN Today and Tomorrow

As previously mentioned the 6LoWPAN working group has generated the Proposed Standard for the format document, as described in this paper. In order for a Proposed Standard to mature into a Draft Standard, there must be multiple independent implementations of the standard – today there are at least 6 different implementations of 6LoWPAN being developed, demonstrated and productized on multiple different 802.15.4 radio platforms. Additionally there are several prototype and pilot deployments of 6LoWPAN networks around the world. The United States Department of Defense is developing several prototype 6LoWPAN networks to demonstrate IPv6 mobility, security and self configuring, self healing mesh sensor networks. Other projects currently under development are the South Korea's IP Ubiquitous Sensor Network (IP USN) project and the associated IP USN Forum, and a number of sensor products in Europe and in the US.

There are still a number of areas to be investigated and explored. The 6LoWPAN Working Group is continuing to investigate the areas of Neighbor Discovery (determining the IPv6 network prefix, local routers, and other network configuration parameters), Secure Neighbor Discovery, Service Discovery (to automatically locate other sensors and controllers and available higher layer services), Security (applying IPsec to these small nodes is problematic), and Life Cycle Management (how are the nodes bootstrapped, how is the network commissioned, and maintained, how are updates applied to the embedded codes), as well as the development of new 6LoWPAN header types. Along with these, the area of efficient *route over* protocols designed for these low power networks is of keen interest and a new working group (Routing for Sensor Networks – RSN) may be formed. Equally important is the question of how to mix the *mesh under* and *route over* mechanisms.

6LoWPAN does not solve all of the problems and issues related to sensor networks and low power RF applications. It does provide a small, stable, well understood, and open beginning upon which to build prototypes, pilots, test-bed networks and devices which can be used to explore these areas of research.

## 7. ACKNOWLEDGMENTS

The design and development of 6LoWPAN is due to the work of many people in the Internet Engineering Task Force and the 6LoWPAN Working Group. In particular, the authors of the two drafts Gabriel Montenegro, Nandakishore Kushalnagar, Christian Schumacher, Jonathan Hui and David Culler.

## 8. REFERENCES

- [1] BACnet, ANSI/ASHRAE 135-1995 Annex J, 1995
- [2] Bormann C., Burmeister C., Degermark M., Fukushima H., Hannu H., Jonsson L-E., Hakenberg R., Koren T., Le K., Liu Z., Martensson A., Miyazaki A., Svanbro K., Wiebke T., Yoshimura T., Zheng H. RObust Header Compression (ROHC) RFC3095 IETF, July 2001
- [3] Culler, D., Hui, J. 6LoWPAN Tutorial. Tiny OS Technology Exchange 2007
- [4] Elixmann, M. The Internet of Things  
[ftp://ftp.cordis.europa.eu/pub/ist/docs/ka4/au\\_conf670306\\_elixmann\\_en.pdf](ftp://ftp.cordis.europa.eu/pub/ist/docs/ka4/au_conf670306_elixmann_en.pdf)
- [5] Galeev, M. Home Networking with Zigbee  
<http://www.embedded.com/showArticle.jhtml?articleID=18902431>
- [6] Gershenfeld, N., Kirkorian, R., Cohen, D. The Internet of Things. Scientific American, October 2004.
- [7] IEEE Computer Society IEEE 802.15.4 – Wireless Medium Access Control and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)
- [8] Jones, S. TI Zigbee Z-Stack on the CC2430 and MSP430  
<http://www.motherboardpoint.com/t161906-ti-zigbee-zstack-on-the-cc2430-and-msp430.html>
- [9] Kushalnagar, N., Montenegro, G., Schumacher, C. 6LoWPAN: Overview, Assumptions, Problem Statement and Goals
- [10] Mackay, S. Foundation Fieldbus High Speed Ethernet (HSE) and TCP/IP  
[http://www.iceweb.com.au/ffeuca/papers/JAPerth/03\\_FF\\_H1\\_and\\_Ethernet\\_TCP\\_IP.pdf](http://www.iceweb.com.au/ffeuca/papers/JAPerth/03_FF_H1_and_Ethernet_TCP_IP.pdf)
- [11] Montenegro, G., Kushalnagar, N., Hui, J., Culler, D. Transmission of IPv6 Packets over IEEE 802.15.4 Networks
- [12] Pister K., Conant, R. Dust Networks – Bringing the information revolution to the Physical World  
[http://www.dust-inc.com/news/web\\_seminars/06-12-06%20Seminar\\_Presentation.pdf](http://www.dust-inc.com/news/web_seminars/06-12-06%20Seminar_Presentation.pdf)
- [13] Tunheim, S. Implementing an IEEE 802.15.4 and Zigbee Compliant RF-Solution  
[http://www.chipcon.com/files/Chipcon\\_Presentation%20IIC-China%20ESC-China%20April%202005.pdf](http://www.chipcon.com/files/Chipcon_Presentation%20IIC-China%20ESC-China%20April%202005.pdf)
- [14] Zensys <http://www.zen-sys.com/index.php?page=227>