SNAIL Gateway: Dual-mode Wireless Access Points for WiFi and IP-based Wireless Sensor Networks in the Internet of Things

Minkeun Ha, Seong Hoon Kim, Hyungseok Kim, Kiwoong Kwon, Nam Giang, and Daeyoung Kim
Department of Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, Korea
{minkeun.ha, shkim08, witbring, kiwoong, zang, kimd}@kaist.ac.kr

Abstract—**One of the important challenges in the Internet of Things (IoT) is how to acquire the physical context of things. IP-based wireless sensor networks (IP-WSNs) could be a promising approach to collecting the physical context of things and to integrating WSNs to the Internet. However, realizing IP-WSNs in IoT exposes two major challenges. One is how to embed the Internet Protocol (IP) in resource-constrained sensor nodes. The other is how to achieve real-world deployment of WSNs and its integration with the Internet on the fly and on the cheap. In this paper, we present the SNAIL (Sensor Networks for All-IP World) project and introduce a new type of IP-WSN gateway, which supports dual wireless access points for WiFi and IP-WSN, enabling deployment of SNAIL nodes in an easy and rapid manner as for the solution. To show the proof-of-concept, we implement a new SNAIL platform from tiny sensor nodes to a gateway.**

Keywords-Internet of Things; 6LoWPAN; Wireless Sensor networks; WiFi

I. INTRODUCTION

Thanks to the advance of miniaturization and falling costs of RFID, sensor networks, wireless communications and technologies, everyday objects are now empowered by identification, computation, and communication technologies. The Internet, which is a global network to interconnect human-operated computers through TCP/IP protocol suite, now attempts to embrace those everyday objects, bringing out the vision of the Internet of Things (IoT) [1]. Thus, the IoT encompasses not only human-operated computers but also everyday objects, which we call them things as a whole throughout this paper.

One of the important challenges in IoT is how to make things answer the questions about what circumstance they lie in. Enabling this needs to transform data generated by things to information about physical context of things. Needless to say, wireless sensor networks (WSNs) play a pivotal role of providing a means to acquiring the physical context of things. As a consequence, WSNs must be an essential integral part of the IoT. In this respect, many early WSNs have used proprietary networking solutions and RF technologies, and thus have been integrated with the Internet at the higher layer (e.g., application layer). However, applications in the IoT would prefer using standards-based networking and RF solutions in order to benefit from openness and interoperability.

The TCP/IP protocol suite would be a promising approach to integrating WSNs to the IoT. In particular, IPv6 is a good fit for WSNs since it provides nearly infinite address space enabling everyday objects to be identified, located and globally accessible within the Internet. Moreover, integration with the Internet infrastructure promotes flexible, reliable, and optimized communications. Thus, leveraging the existing Internet infrastructure could accelerate the rapid adoption and prevalence of everyday objects in the IoT. Indeed, to keep pace with these trends, IPSO [4], 6LoWPAN [2] and ZigBee/IP [5] adopt the IPv6 as an underlying networking solution to integrate WSNs with the Internet.

In practice, realizing IPv6 for WSNs in the IoT exposes two major challenges. One is how to embed the Internet protocol in resource-constrained sensor nodes. Another is how to achieve real-world deployment of WSNs and its integration with the Internet on the fly and on the cheap. As for the solution to the former, there have been many researches such as TinyOS [6], Contiki [7], MicroIP [8], SNAIL [3] which showed that embedding IP protocol in tiny sensor node is appropriate for WSNs. However, current approaches for the latter challenge are developed based on separate WSN gateways, which require additional efforts to connect the gateway to the Internet infrastructure through either wired or wireless medium. This may result in expensive partial reconstruction of existing infrastructure. Thus, a practical and cost effective system to build IPv6 infrastructure for everyday objects is still foggy.

In this paper, as for the solution to the latter, we introduce a new type of 6LoWPAN gateways, which is a part of SNAIL (Sensor Networks for All Ip-World) [3] project. SNAIL is an IP-based WSN solution, which is fully compatible with IETF 6LoWPAN standards, characterized by additional advanced features including global mobility management, web enablement, global time synchronization, and security. Within the SNAIL platform, we have developed a new type of SNAIL gateway which enables both integration with the Internet and deployment of SNAIL nodes in an easy and rapid manner. That is, motivated by successful deployment of WiFi APs to provide infrastructure

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. R0A-2007-000-10038-0).

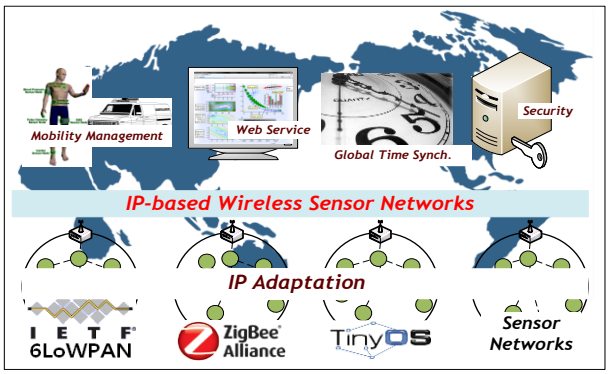


Figure 1. IP-based Wireless Sensor Networks for the IoT

for human-operated mobile computers like smart phones, we develop the SNAIL gateway integrated into the WiFi AP (e.g., OpenWRT) to operate as access points for WSNs.

The reason behind this is that as WiFi APs are distributed components as a part of Internet Infrastructure that we can easily see anywhere and anytime, we can reuse the existing WiFi APs by simply plugging a SNAIL PAN coordinator. Accordingly, by leveraging well-constructed existing WiFi infrastructure, we benefit from not requiring reconstruction of existing infrastructure and from ubiquity of WiFi APs, enabling low-cost and rapid deployment of SNAIL nodes and integration with Internet. To show the proof-of-concept, we implement the SNAIL platform from tiny sensor nodes to a gateway using OpenWRT.

The rest of this paper is organized as follows. In Section II, we highlight the requirements for IP-WSNs followed by our approaches based on the SNAIL platform. In Section III, we show the implementation of SNAIL platform focusing on the SNAIL gateway. Then, we end up with conclusion in Section V.

II. SNAIL: SENSOR NETWORKS FOR AN ALL-IP WORLD

The blueprint for the IoT supposes that trillions of things could be connected through the Internet over IP-WSN. Thus, the IoT is likely to encounter issues analogous to those the Internet has experienced in its history. In this section we show the requirements and key features of SNAIL: mobility management, web enablement, global time synchronization, and security, as depicted in Fig. 1. Then, we describe a new SNAIL gateway platform to integrate IP-WSN with the IoT; and we present important considerations to build IP-WSN.

A. IP-WSN Requirements to realize the Internet of Things

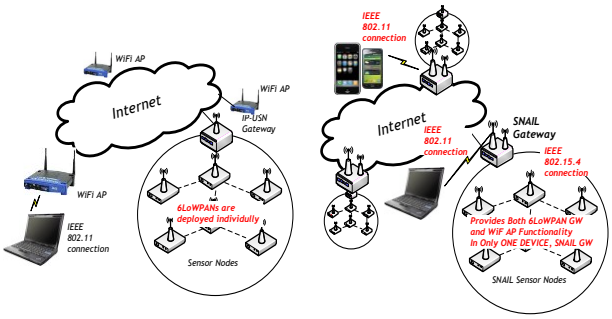
The research in the IoT field has tended to focus on how to build a fundamental architecture that enables the standard protocol, IP, to be used in a WSN space. The building process generally involves an implementation of IP adaptation to enable resource-limited things to be seamlessly connected to the Internet through lossy networks. The basic architecture incorporates protocols that are needed to accomplish full IP operations. The mobility protocol supports fast, seamless intra-personal area network (PAN) and inter-PAN handovers on a node and network basis as well as route optimization. The web enablement scheme gives end users a simple and user-friendly access method to things through web browsers. The global time synchronization protocol enables things to share global time in an efficient way. Lastly, the security protocol guarantees lightweight yet robust end-to-end security. Furthermore, it is important to consider how to achieve real-world deployment of WSNs and how to integrate this IP-WSN with already deployed IoT infrastructures on the fly and on the cheap.

B. Key Features of SNAIL Protocol

The SNAIL protocols are designed to comply with RFC4944 [11] which defines a standard IP adaptation method for IEEE 802.15.4 based wireless sensor networks. The SNAIL adaptation layer includes all the necessary packet formats and operations, and is fully compliant with standards pertaining to header compression, addressing, fragmentation and reassembly, and so on. Furthermore, to complete the IP adaptation, it employs important protocols that are not specified in the standard such as a bootstrapping protocol to obtain an IPv6 address in a start-up process, neighbor discovery to maintain the relation of the associated neighbor nodes, and routing protocol. For the routing protocol, we use a hierarchical routing protocol because of its scalability, efficiency of address-based routing scheme, and no memory overhead by obviating the need for maintaining a routing table.

SNAIL employs a novel mobility management protocol called MARIO [9], which stands for mobility management protocol to support intra-PAN and inter-PAN handover with route optimization for 6LoWPAN. The design of MARIO is based on the basic scheme of mobile IPv6 (MIPv6). However, because of the difference between the Internet environment and 6LoWPAN, mobility management protocol in 6LoWAPN should include a fast and seamless handover scheme in a lightweight fashion and a header compression method. To reduce handover delay, MARIO is designed with a make-before-break method that infrastructure nodes preconfigure mobile node's handover to its candidate next parent. In addition, MARIO provides a header compression method: 32 bytes binding update into 13 bytes and 12 bytes binding acknowledge into 3 bytes.

A web access approach endeavors to integrate things networks into the Internet from the viewpoint of user friendliness and platform independence. SNAIL enables web access to the sensor node which hosts a tiny web server and supports web browsing with presentation metadata such as web templates, multimedia, and application codes to provide rich user interface through the client's web browser. However, since sensor nodes have limited resource and low data rate characteristics, it is difficult to support standard web protocols and rich user interfaces. Thus, we distribute the web data traffic such as presentation metadata and sensing data to a presentation server and sensor nodes respectively. In addition, the application code served by presentation servers can provide a method to process sensing data, to control actuator nodes, and to integrate sensing data with third-party web applications which serve OpenAPI such as Google Maps API, Twitter API, and Facebook API. In order to support these functions lightweightly, SNAIL defines a header compression method for HTTP and TCP



(a) Traditional IP-WSN Platforms (b) SNAIL Platform

Figure 2. SNAIL GW: a new type of IP-WSN Gateway

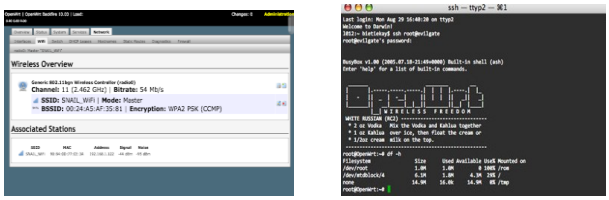
protocols which transforms string-based HTTP headers to binary-based header and reduces redundant fields and unused field in 6LoWPAN.

The IoT requires globally synchronized time to ensure the consistency and precision of information. We designed a multi-hop time synchronization protocol called the 6LoWPAN Network Time Protocol (6LNTP) which enables global time synchronization to 6LoWPAN. 6LNTP has two phases: gateway synchronization phase and LoWPAN synchronization phase. In gateway synchronization phase, a gateway synchronizes global time based on a standard protocol, SNTP. Then, in LoWPAN synchronization phase, nodes synchronize their time with gateway. This latter phase operates by exchanging two types of messages: a time-sync message, which includes a reference time, and a follow-up message, which includes a time error that is accumulated delays of processing time, back-off time, and transmission time in each hop. The latter phase is initiated by leaf nodes and the intermediate nodes are synchronized on the basis of the leaf nodes' time requests. This is the reason that if each sensor node initiates time synchronization to a gateway, the network overhead increases exponentially with the number of sensor nodes. Thus, 6LNTP benefits from the significantly reduced signaling overhead of the intermediate nodes.

The global accessibility of things raises additional security issues; that is, things are directly exposed to potential threats from powerful sources with abundant resources on the Internet. There are two key tasks for building a secure things network: scrutinizing possible attacks on a things network which is transparently open to the Internet; and designing an efficient, strong, and interoperable security framework. The key concept of our SNAIL security protocol (SSNAIL) [10] is to design of an elliptic curve cryptography (ECC)-based lightweight security protocol for things that are exposed to the Internet on an end-to-end basis without losing any security robustness. Our implementation confirmed the success of the operation of the ECC with the 160-bit curve secp160kl: it provides the same level of security with smaller keys as an RSA with a 1024-bit key. The performance advantages are a fast computation, a reduced memory requirement, and a low transmission overhead.

C. *SNAIL Gateway for 6LoWPAN and WiFi networks*

One of the challenges in realizing SNAIL to be a part of the IoT is how to achieve real-world deployment on the fly and on the cheap. Since current approaches using separate IP-WSN gateways require additional efforts to connect the gateway to the Internet infrastructure through either wired or wireless medium. This may result in expensive partial reconstruction of existing infrastructure. Thus, it makes deployment of IP-WSNs difficult in real-world.



(a) Web-based Administration Interface (b) SSH Connection to SNAIL GW

Figure 3. Administration Interface for SNAIL GW

Fortunately, we have witnessed successful deployment of WiFi APs anywhere and anytime as a part of the Internet Infrastructure. Therefore, if we can reuse the existing WiFi APs, we may benefit from not requiring reconstruction of existing infrastructure and from ubiquity of WiFi APs, enabling low-cost and rapid deployment of SNAIL nodes and integration with Internet.

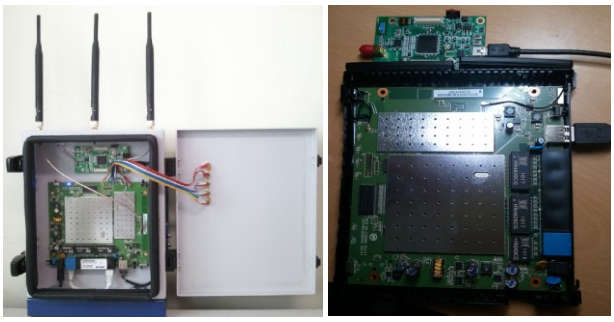
Motivated by the successful deployment of WiFi APs, we developed a new type of 6LoWPAN gateway, SNAIL GW, which supports both IEEE 802.11 b/g/n based WiFi access point and 6LoWPAN gateway, as depicted in Fig. 2. The SNAIL GW is implemented on the OpenWRT [12] which is a GNU/Linux based firmware program for embedded devices such as residential gateways and routers. SNAIL GW provides easy setup and easy deployment. To enable 6LoWPAN in our SNAIL GW platform, following set up process is enough. Connect SNAIL PAN coordinator to SNAIL GW through a USB interface, install the gateway software which packed in ipkg package, and run the software. Additionally, as shown in Fig. 3, network administrator can configure network properties through a web-based interface. Furthermore, in case of cost effectiveness, we reduce the deployment cost for the Internet infrastructure because we can have two functionalities for WiFi and 6LoWPAN by deploying only one SNAIL GW. Moreover, since our SNAIL GW is implemented on the GNU/Linux based OpenWRT project, we can also easily enable other network features such as UPnP/DLNA, NAS service, and so on. In the next section, the implementation will be explained in detail.

III. DESIGN AND IMPLEMENTATION

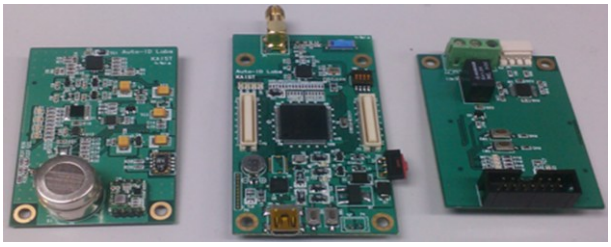
We implemented all software and hardware components for the SNAIL as a solution for IP-WSN. This section will describe in detail the implementation of each elements of SNAIL.

A. *SNAIL Gateway*

The gateway provides a connection between the sensor nodes and the Internet, and it plays a role to compress incoming packets from the Internet to the inside of sensor networks and to decompress outgoing packets from the inside of sensor networks to the Internet, so that HTTP/TCP



(a) SNAIL Gateway Hardware



Sensor board Processor board Actuator board

(b) SNAIL Sensor Node Hardware

Figure 4. SNAIL Hardware Platform

packets can fit into resource-constraint sensor networks. The gateway also provides the WiFi AP functionality. The SNAIL GW hardware platform is described in Fig. 4(a).

Previous SNAIL GW described in [3] is solely dedicated to the Ethernet connection. Unlike this, we implement a new type of SNAIL GW on the off-the-shelf product, Buffalo WZR-HP-G300NH, a Linux-based device which has a 400 MHz MIPS CPU, 64MB ram, 32MB flash rom, IEEE 802.11 b/g/n WLAN, four Ethernet ports, one WAN port, and one USB interface. Its firmware is then replaced with backfire version of OpenWRT [12], a Linux distribution for embedded devices. Besides all the basic functions of a typical WiFi AP such as firewall, NAT, DHCP server, port forwarding, IPv4-IPv6 dual stack, and 6-to-4 tunneling, OpenWRT comes with many remarkable and advanced features. It also includes a software package manager and the software repository which provides roughly more than 2,000 packages. In addition, thanks to LuCI [13], OpenWRT features a web interface for administration that all the configurations and setup can be done effortlessly. Since it can be easily configured as a WiFi AP, a wireless repeater, a wireless bridge or a wireless client, SNAIL network can be accessed from the Internet through both wired and wireless connection.

As shown in Fig. 5, in which we omit some layers to simplify the figure, we implemented two network stacks: one for the Internet connection (Internet-stack) and the other for the WSN connection (SNAIL-stack). These two stacks are interconnected through a universal TUN/TAP (TUN as in network TUNnel; TAP as in network TAP) driver, which passes incoming IP packets from the Internet to SNAILGW software and vice versa. SNAIL-stack includes all of the IP

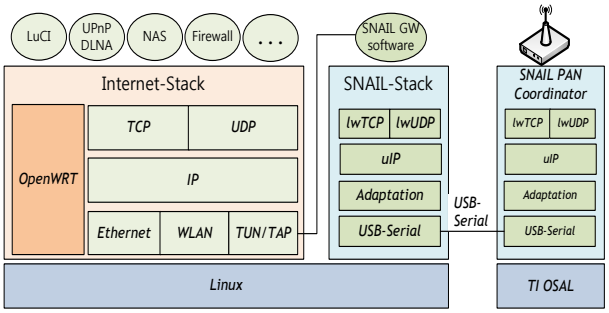


Figure 5. SNAIL Gateway Block Diagram

adaptations such as bootstrapping, header compression, and so on. For the purpose of connecting to sensors network, we utilize the USB-serial communication with SNAIL PAN coordinator and modified its routing table to forward packets from the Internet to WSN and vice versa. The SNAIL GW software is developed as a Linux daemon. The detail software stack of SNAIL GW is described in Fig. 6(a).

SNAIL PAN coordinator is a representative of a PAN, which is responsible for starting the formation of a 6LoWPAN network. It is implemented on the processor board, which has a 16bit TI MSP430F5438 MCU and TI CC2520 radio transceiver, as depicted in Fig. 4(b). The software stack of PAN coordinator is same with SNAIL sensor nodes except USB-serial based network interface for communication between SNAIL GW and SNAIL PAN coordinator. We can enable IEEE 802.15.4 communication by simply plugging PAN coordinator into USB port of SNAIL GW.

B. *SNAIL Sensor Nodes*

In IP-WSN, the sensor nodes play a role to gather environmental information corresponding to equipped sensors. It is implemented on the 16bit TI MSP430F5438 based SNAIL sensor node hardware platform as a low-power general-purpose sensor device, which equipped TI CC2520 radio transceiver and various sensors such as light, temperature, humidity, 2-axis gyroscope. Thus, the sensor node can apply diverse power consumption mode and provide IEEE 802.15.4 compliant DSSS base band modem with 250 kbps data rate. The SNAIL sensor node hardware platform consists of three stackable boards: sensor board, processor board, and actuator board, as shown in Fig. 4(b). Due to our stackable board design, we can reorganize the hardware corresponding to its applications. The actuator board has a relay circuit so that it can control other devices. Using the detachable boards, we can apply our sensor nodes to wide range of applications.

The SNAIL sensor node software runs on top of TI OSAL (Operating System Abstraction Layer), which works an event-driven architecture. The SNAIL sensor node software stack follows layered architecture, mainly due to its flexibility for efficient development of an IPv6-based

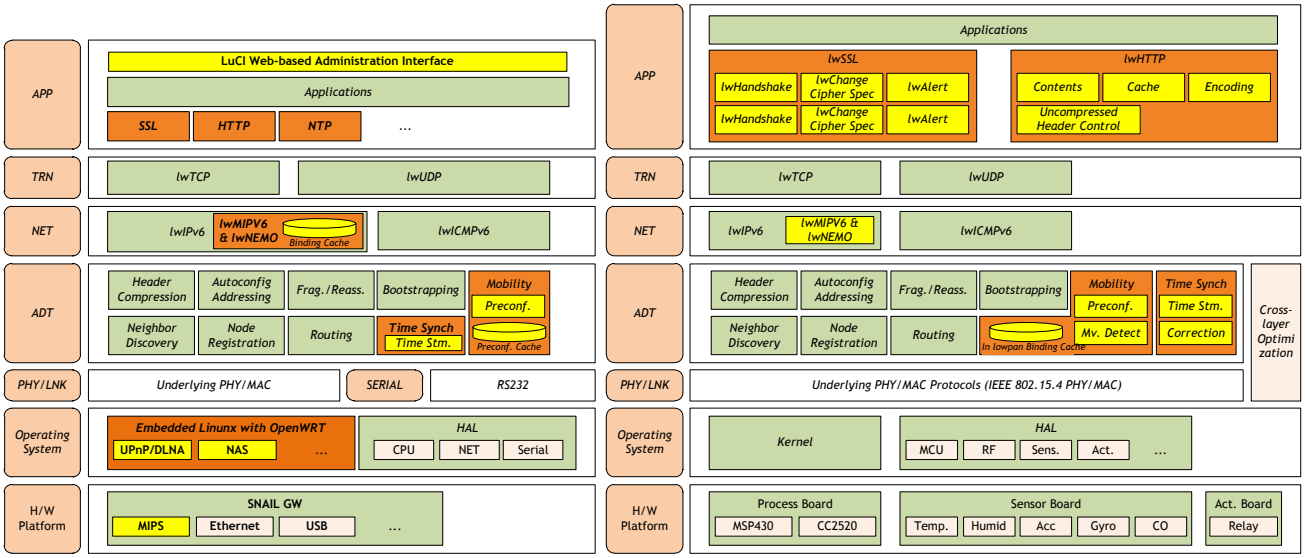


Figure 6. SNAIL Software Stack

architecture, as shown in Fig. 6(b). A cross-layer optimization layer vertically locates from a PHY/LNK to the adaptation layer in order to provide a collaboration interface with the upper layers. The adaptation layer includes several core functions of IP adaptation for operating IP operations over IEEE 802.15.4, such as header compression, bootstrapping, fragmentation and reassembly, neighbor discovery, node registration, and routing. Also, mobility management resides in both the adaptation and network layers so it can support mobile IPv6 transparently as well as the handover preconfiguration of MARIO. A lightweight TCP/IP stack is located on top of an adaptation layer, and the lightweight SSL and HTTP are located in an application layer over the transport layer for the security and web enablement. Thus, SNAIL can provide Internet services compatible with existing Internet services.

IV. CONCLUSIONS

Realizing IoT requires WSNs to be integrated with the Internet and to be deployed in real-world. To achieve this, we introduced SNAIL platform, which is an IP-based WSN solution for everyday objects, as an approach to integrating WSNs with the Internet. Aiming at placement of WSNs everywhere on the fly and on the cheap as current WiFi APs are deployed, we developed a SNAIL gateway that integrates WiFi AP and IP-WSNs using SNAIL. Thus, SNAIL platform made WSNs globally accessible from/to the Internet in a rapid and cost-effective manner. To show the proof-of-concept, we implement a new SNAIL platform from tiny sensor nodes to a gateway.

REFERENCES

[1] M. Chui., M. Loffler, and R. Roberts, "The Internet of Things," *McKinsey Quarterly*, no. 2, pp. 1-9, Mar. 2010.

[2] IETF, IPv6 over low-power WPAN (6LoWPAN), available : http://www.ietf.org/html.charters/6lowpan-charter.html

[3] Sungmin Hong, Daeyoung Kim, Minkeun Ha, Sungho Bae, Sangjun Park, Wooyoung Jung, and Jae-eon Kim, "SNAIL: An IP-based Wireless Sensor Network Approach Toward the Internet of Things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34-42, Dec. 2010.

[4] IPSO Alliance, website : http://ipso-alliance.org/

[5] ZigBee Alliance, websit : http:// www.zigbee.org/

[6] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A.Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An Operating System for Sensor Networks," *Ambient Intelligence, Springer*, pp. 115–148, Dec. 2005.

[7] A. Dunkels, B. Groenvall, and T. Voigt. "Contiki - a lightweight and flexible operating system for tiny networked sensors," *In Proceedings of the First IEEE Workshop on Embedded Networked Sensors (EmNets)*, Tampa. Florida. USA, Nov. 2004.

[8] A. Dunkels, "Full TCP/IP for 8-bit architectures," *In Proceedings of the First International Conference on Mobile Systems, Applications, and Services (Mobisys) 2003*, San Francisco, USA, May 2003.

[9] Minkeun Ha, Daeyoung Kim, Seong Hoon Kim, and Sungmin Hong, "Inter-MARIO: A Fast and Seamless Mobility Protocol to support Inter-PAN Handover in 6LoWPAN," *In Proceedings of IEEE Global Communications Conference(GLOBECOM) 2010*, Miami, USA, Dec. 2010.

[10] Wooyoung Jung, Sungmin Hong, Minkeun Ha, Young-Joo Kim, and Daeyoung Kim, "SSL-based Lightweight Security of IP-based Wireless Sensor Networks," *In Proceedings of the 2009 IEEE International Workshop on Quantitative Evaluation of large-scale Systems and Technologies(QuEST)*, Bradford, UK, May 2009.

[11] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," IETF RFC4944, Sep. 2007.

[12] OpenWRT, website : https://openwrt.org/

[13] LuCI, website : http://luci.subsignal.org/