6LoWPAN stacks: a survey

Chen Yibo[1], Kun-Mean Hou[1], Haiying Zhou[2*], Hong-ling Shi[1], Xing Liu[1], Xunxing Diao[1], Hao Ding[1], Jian-Jin Li[1] and Christophe de Vaulx[1]

[1]LIMOS Laboratory UMR 6158 CNRS
Blaise Pascal University Clermont-Ferrand II
Clermont-Ferrand, France
chenyibo.clover@yahoo.fr, kun-mean.hou@isima.fr

[2]School of Computer Science & Technology
Harbin Institute of Technology
Harbin, China
haiyingzhou@hit.edu.cn

***Abstract*—Wireless Sensor Network (WSN) is one of the key technologies of 21st century, while it is a very active and challenging research area. It seems that in the next coming year, thanks to 6LoWPAN, these wireless micro-sensors will be embedded in everywhere, because 6LoWPAN enables P2P connection between wireless nodes over IPv6. Nowadays different implementations of 6LoWPAN stacks are available so it is interesting to evaluate their performance in term of memory footprint and compliant with the RFC4919 and RFC4944. In this paper, we present a survey on the state-of-art of the current implementation of 6LoWPAN stacks such as uIP/Contiki, SICSlowpan, 6lowpancli, B6LoWPAN, BLIP, NanoStack and Jennic's stack. The key features of all these 6LoWPAN stacks will be established. Finally, we discuss the evolution of the current implementations of 6LoWPAN stacks.**

Keywords- IEEE802.15.4; 6LoWPAN; IoT; Wireless Sensor Network

I. INTRODUCTION

IEEE802.15.4 (ZigBee) is the defacto wireless access medium standard for WSN due to its low energy consumption and cost, but the interoperability issue of WSN remaines a challenge. Nowadays the WSN interoperability problem seems to be overcome thanks to advanced made by the 6LoWPAN WG 'IPv6 over Low Power Personal Area Network working Group'. Thus in the near future all the wireless sensor nodes will be connected to Internet as any computer devices (e.g. PC or smart phone). This possibility provides new internet application, which calls Internet of Things 'IoT'. Notice that IEEE802.15.4 standard is adopted to be used as the 6LoWPAN wireless access medium. However the data payload of IEEE802.15.4 is 127 bytes [10] and the standard IPv6 packet header is 40 bytes. To minimize overhead it is necessary to compress the IPv6 packet header, stateless compression HC1 is applied and in best case the IPv6 header is compressed into 2 bytes [11]. Thanks to the IPv6 header compression the impact of send and receive messages over 6LoWPAN is acceptable comparing with the raw IEEE802.15.4 ones [30]. Currently several open 6LoWPAN stacks are available: 6lowpancli, b6lowpan, blip, uIP, SICSlowpPAN and NanoStack1.1. Notice that there are also proprietary 6LoWPAN stacks such as Jennic, Hitachi, and Arch Rock IP/6LoWPAN [16], NanoStack2.0 and m-stack. All of the previous 6LoWPAN stacks need small memory footprint and were implemented on different low power microcontroller platforms from 8 to 32-bit such as AVR [13], 8051 [15], MSP430 [14], STM32 [12] and JN5139.

*Corresponding author: Haiying Zhou, haiyingzhou@hit.edu.cn

The test and validation of 6LoWPAN stacks are carried out on a small number of wireless sensor nodes (less than 10). One of the TCP/IP protocol is the network scalability and interoperability, we think that the scalability of large scale of LoWPAN is still need to be proved because the wireless sensor nodes have limit resources and to increase WSN lifetime most of WSN adopted sleep&wakeup running mode. Another problem which should be mentioned is that the IEEE 8.14.5 standard has two communication modes: unicast and broadcast so it will not easy to implement multi-cast to meet the requirements of IPv6.

II. STATE-OF-ART

In this paper we will present only the main known 6LoWPAN stacks: 3 from academic research (Jacobs University, UC Berkeley and SICS) which based on TinyOS2.x and Contiki, and 2 commercial from Sensinode and Jennic companies. The survey of the different 6LoWPAN stacks were carried out by Yannis Mazzer et al. [4], Ricardo Silva et al. [6], Kevin Dominik Korte [9], Usman Sarwar et al. [7] and Guo [5] but due to the rapid evolution new 6LoWPAN stacks are available.

The basic concept of 6LoWPAN stack is illustrated in Fig. 1. LoWPAN is an adaptation layer. The operating system plays a key role on the development of 6LoWPAN in term of memory footprint and functionality.

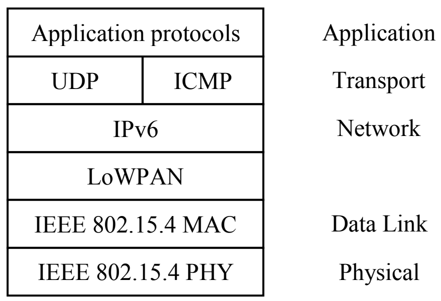

Figure 1. Basic structure of 6LoWPAN

A. TinyOS-2.x 6lowpancli, b6lowpan, BLIP

This subsection will mainly focus on the two 6LoWPAN implementations over TinyOS-2.x. In fact the first available and open implementation of 6LoWPAN was developed by Jacobs University Bremen, named 6lowpancli in 2007 [17, 18]. 6lowpancli supports header compression (accomplishing with the HC1 format), fragmentation and addressing, as well as IPv6 stateless configuration. Such stateless configuration requires the previous manual setting of the network address prefix. Regarding high-level protocols, 6lowpancli supports UDP, accomplishing with the HC_UDP format and ICMPv6 messages. Hence, it is possible to ping motes and use UDP sockets from any Internet node to a small sensor node.

To implement the bridge between the WSN and the conventional network, the same implementation provides a daemon to emulate an IPv6 tunnel over IEEE802.15.4, allowing the communication, through, for instance, the USB port of any laptop.

Therefore, 6lowpancli is completely static requiring manual configuration. It doesn't support any type of neighbor discovery mechanism, or mobility, like MIPv6. Besides, the support for mesh network is not provided and when a packet with different destination address is received, it is just dropped.

B6lowpan 'Berkeley 6LoWPAN', now renamed to blip 'the Berkeley Low power IP', was the second released implementation over TinyOS-2.x [6, 9, 28]. BLIP is currently the most advance 6LoWPAN stack in term of functionality [8, 28]. The code is not included in the TinyOS-2.x core, but can be found in the contributions from Berkeley UC. Implementing the basic features defined in RFC4944, namely header compression, fragmentation and addressing, b6lowpan also supports ICMPv6 and UDP packet. Besides that, the last version introduced the first prototype of a TCP stack.

BLIP implements Neighbor Discovery, in a light version, configuring a link local address on nodes' boot and a global address if a Router Advertisement is received.

As 6lowpancli, b6lowpan includes a daemon to create an IPv6 tunnel to connect conventional networks and WSNs. As part of the Sink Node application, the daemon can run on any Unix based machine. Since the last released version (20/3/2009) that ND is included in the daemon. Before that, radvd was required to run separately.

Unlike 6lowpancli and b6lowpan, blip supports mesh networks defined by RFC4944 and known as "mesh under". All these 6LoWPAN stacks run on the TelosB node (Figure 2) based on the MSP420 microcontroller [14] having 10 k bytes of RAM and 48 k bytes of Flash memory [9].

Figure 2. TelosB mote

B. Contiki uIP and SICSlowpan

Contiki is an open source, highly portable, hybrid operating system (event driven and multitasking) for memory-efficient networked embedded systems and wireless sensor networks. A typical Contiki configuration is 2 kbytes of RAM and 40 kbytes of ROM. Contiki provides IP communication, both for IPv4 and IPv6 and has been implemented in a variety of projects such as road tunnel fire monitoring, intrusion detection, water monitoring in the Baltic Sea, and in surveillance networks [1, 20, 21].

uIPv6 stack is the first 6LoWPAN stack implemented over Contiki and the current version is named SICSlowpan (Swedish Institute of Computer Science 6LoWPAN), which is IPv6 Ready Phase 1 certified and therefore has the right to use the IPv6 Ready silver logo [6, 29]. SICSlowpan is compliant with RFC4944 [24], implementing header compression, fragmentation and addressing. Besides that, it goes further and implements interoperability support based on draft-hui-6LoWPAN-interop-00 interoperability Test for 6LoWPAN and new header compression mechanisms defined in draft-hui-6LoWPAN-hc-01 compression format for IPv6 datagrams in 6LoWPAN Networks [11].

In fact, SICSlowpan does not implement mesh under and route over. SICSlowpan supports ICMPv6, implements Neighbor Discovery and supports UDP and TCP (Figure 3).

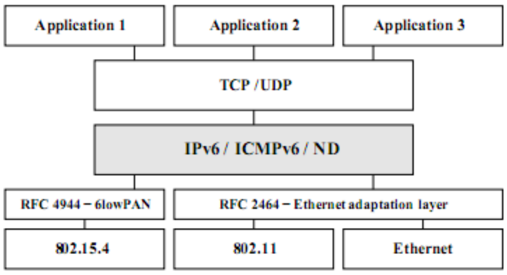

Figure 3. SICSlowpan structure [31]

Atmel's RAVEN board equipped with an Atmegal 1284P MCU having 128 KB of flash and 16 KB of SRAM is used as test-bed (showed in the figure below) to evaluate the implementation of uIP and SICSlowpan.

Notice that the SICSlowpan is integrated to the Cooja which is the Contiki network simulator [22].

C. FreeRTOS NanoStack

NanoStack is developed by Sensinode over FreeRTOS conforms to RFC 4944 specifications [24].

Two versions of NanoStack are available 1.x (open source) and 2.x (proprietary). In this paper we address the NanoStack 1.1 [25], which was implemented on the Sensinode hardware platform (Figure 4). Sensinode's platform (NanoSensor) is one of the uncommon platform based on the CISC 8-bit 8051.

Nanostack 1.1 provides an easy to use socket interface for the programmer and the drivers to manage TI's CC2420 CC2430 radios interfaces. Moreover thanks to the FreeRTOS supports asynchronous timer service and real-time multi-task scheduling policy. However, NanoStack needs more memory resource. Notice that the NanoStack TM2.0 supports MSP430 microcontroller and CC2430/2530 with 32 to 64KB of ROM and 4 to 8 KB of RAM [15].

Figure 4. NanosensorTM N710 [15]

D. *Jennic system*

Jennic's 6LoWPAN system employs a blend of wired and wireless technologies to achieve flexible communication with a powerful range of applications. A straightforward architecture of standard 6LoWPAN system including boarder-router, coordinator and host devices is implemented in the Jennic JN5139-EK036 6LoWPAN Evaluation Kit.

Jennic's 6LoWPAN protocol stack is complied with IETF standards and provides a wireless connectivity solution based on the IEEE802.15.4 standard at 2.4GHz. This 6LoWPAN stack is designed to work on the Jennic's JN5139 wireless modules and microcontroller which is consist of a 32-bit RISC processor with 192KB of ROM and 96KB of RAM [26] and is suitable for IEEE802.15.4 and JenNet [26, 27].

Moreover, it provides simple point-to-point and star connectivity scenarios based on IEEE802.15.4 standard, users also are able to choose alternatively to run 6LoWPAN over Jennic's JenNet networking stack which provides a self-healing cluster tree solution that has been already proven and trustful. A complete package of necessary components including C APIs, stack software is also offered to easily and efficiently develop wireless IP network applications. The highly abstracted C based Jenie API reduces the complexity of WSN design and let the developers focus on increased product functionality and pursue the market. The features of this protocol stack show that it can support typical clusters of 100 nodes and the automatic route formation and repair [26, 27]. Figure 6 illustrates the Jennic 6LoWPAN stack.

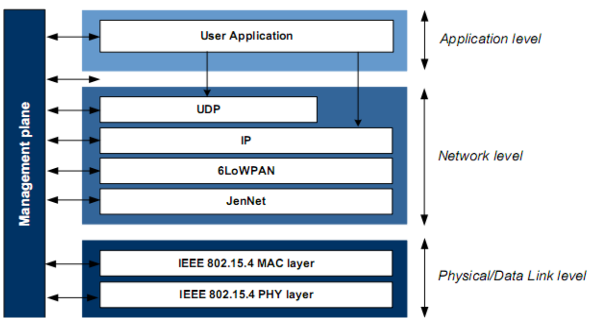

Figure 5. Jennic 6LoWPAN Protocol Stack

Due to the encapsulation and proprietary of these libraries, some key techniques become blurry and this brings annoying problems to the real production or research world. However, the JN5139 microcontroller is adopted widely for experiments platform except the application demos provided by Jennic, such as the ad-hoc sensing and servicing industrial machinery [32] and the landmark firefighter navigation system [33].

E. *Advantages and drawbacks*

The table 1 summarizes the main key features of the 6LoWPAN stacks mentioned above.

III. DISCUSSION

With the current status, we think that the most important contributions of the current 6LoWPAN stacks are come from UC Berkeley (b6lowpan and blip based on TinyOS-2.x) and Swedish Institute of Computer Science (uIP and SICSlowpan based on Contiki). For all these implementation operating system plays a key role in term of memory footprint and functionality. Notice that all the 6LoWPAN stacks are monolithic stack a new modular 6LoWPAN stack is under development at INRIA Rhône Alpes Lyon France [4].

Nowadays millions of computers are interconnected through TCP/IP networks. Thanks to the advanced of 6LoWPAN, IoT is seemed available now but the ability of large scale wireless sensor nodes to interconnect through TCP/IPv6 is still need to be tested and evaluated. Notice that the current 6LoWPAN stacks test-bed platforms are limited to some nodes (<10).

Moreover it's not easy to implement multi-cast over IEEE802.15.4 thus an extension of IEEE802.15.4 may be proposed to solve this problem.

IV. CONCLUSION

Although the IP support in WSNs is nowadays a reality, the large scale application in real world scenarios remains cannot be trusted and therefore inexistent. The IP protocol, specifically IPv6, allows the use of well-known and well-tested high-level protocols. Those protocols could be considered as energetically undesirable, in part due to the enormous packets size and the uncompleted documents. However, in this survey we have introduced the existent 6LoWPAN implementations and tried to analyze these systems from the aspects of the architecture of software stack and hardware platform.

Finally, in spite of some drawbacks the IoT technology is nowadays quite mature and we believe that in the next coming year most of everyday living objects may be connected to internet. This new technology will change the way of metering, monitoring, control and diagnosing. Thus the economic and social impact of IoT currently will be extremely important. How to evaluate the impact of IoT is an open question.

ACKNOWLEDGMENT

This work is supported by the 'Conseil Général du Puy-de-Dôme'(France) and International S&T Cooperation Program of China (N°.2010DFA14400), the Harbin S&T Technology Development Funds (N°. 2009RFLXG009) and the Doctoral Fund of Ministry of Education of China (N°. 200802131024), National S&T Pillar Program of China(N°.2011BAH04B03).

TABLE I. KEY FEATURES OF THE DIFFERENT AVAILABLE 6LOWPAN

Item	6lowpancli	B6LoWPAN	Blip	uIP	SICSlowpan	Nanostack 1.1	Jennic
Hw	TlosB (MSP-16 bit, TI)	TlosB (MSP-16 bit, TI)	TlosB (MSP-16 bit, TI)	Raven (AVR 8-bit, ATMEL) & TlosB (MSP-16 bit, TI)	Raven Raven (AVR 8-bit, ATMEL)	Sensinode (8051 8-bit and MSP-16-bit TI)	Jennic (RISC 32-bit, proprietary)
OS	TinyOS-2.x	TinyOS-2.x	TinyOS-2.x	Contiki	Contiki	FreeRTOS	proprietary
ROM(B)	48k	24,0382	48k	11,4882	11,4882	32-64K	192K
RAM(B)	10k	3,5982	10k	1,7482	1,7482	4-8K	96K
TCP	no	no	prototype		yes		no
ICMPv6	no	no	Yes	yes	yes	yes	yes
Multicast	no	no	no	no	no	no	no
ND	no	no	yes	yes	yes	yes	yes
Mesh under	no	no	yes	no	No	yes	yes
Route over	no	no	yes	no	no	yes	not clear
License	3BSD	3BSD	3BSD	3BSD	3BSD	GPL	proprietary
maintain	no	yes	yes	yes	yes	yes	yes

REFERENCES

[1] A. Dunkels, SICSlowpan - Internet-connectivity for Low-power Radio Systems, report.

[2] Z. Shelby and C. Bormann, 6LoWPAN: The Wireless Embedded Internet, J. Wiley & Sons, Nov 2009, 254pg. ISBN: 978-0-470-74799-5

[3] Fredrik ¨Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, Thiemo Voigt, Demo Abstract: Cross-level Simulation in COOJA, Swedish Institute of Computer Science Email: {fros,adam,joakime,nfi,thiemo}@sics.se

[4] Yannis Mazzer, Bernard Tourancheau, Comparisons of 6LoWPAN Implementations on Wireless Sensor Networks, 2009 Third International Conference on Sensor Technologies and Applications.

[5] Weijun Guo, Performance Analysis of IP over IEEE 802.15.4Radio using 6LoWPAN, http://www1.cse.wustl.edu/~jain/cse567-08/ftp/7lowpan.pdf

[6] Ricardo Silva, Jorge Sá Silva and Fernando Boavida, Evaluating 6lowPAN implementations in WSNs, Department of Informatics Engineering University of Coimbra Pólo II - Pinhal de Marrocos, 3030-290 Coimbra, PORTUGAL, rnsilva@dei.uc.pt, sasilva@dei.uc.pt, boavida@dei.uc.pt

[7] Usman Sarwar, Gopinath Sinniah Rao, Zeldi Suryady, and Reza Khoshdelniat, A Comparative Study on Available IPv6 Platforms for Wireless Sensor Network, World Academy of Science, Engineering and Technology 62 2010.

[8] Nithin K N, BLIP Implementation, Department of ECE, Indian Institute of Science, Bangalore November 17, 2010, http://www.ece.iisc.ernet.in/6panview/wpcontent/.../blip_implementation.pdf

[9] Kevin Dominik Korte, Evaluation of 6lowpan Implementations, Master thesis, Computer Science Jacobs University Bremen Campus Ring 128759 Bremen Germany.

[10] "IEEE 802.15.4-2006 standard," http://standards.ieee.org/getieee802/802.15.html, April 2010.

[11] draft-hui-6lowpan-hc-01 Compression format for IPv6datagrams in 6lowpan Networks.

[12] ST, STM32F103x6, STM32F103x8, STM32F103xB: Performance line, ARM-based 32-bit MCU with Flash, USB, CAN, seven 16-bit timers, two ADCs and nine communication interfaces, STM32 datasheet.

[13] ATMEL, AVR raven, http://www.atmel.com, April 2010.

[14] http://focus.ti.com/en/download/mcu/Sensinode_CC430.pdf

[15] Sensinode, NanoSensor™ N710 data sheet, www.sensinode.com.

[16] Arch Rock, IP/6LoWPAN Overview: An IPv6 Network Stack for Wireless Sensor Networks, http://www.cs.berkeley.edu/~jwhui/6lowpan/Arch_Rock_Whitepaper_IP6LoWPAN_Overview.pdf.

[17] Matus Harvan, Jurgen Schonwalder. A 6LoWPAN Implementation for TinyOS 2.0. Jacobs University Bremen, Germany.

[18] Matus Harvan, Connecting Wireless Sensor Networks to the Internet - a 6lowpan Implementation for TinyOS 2.0, Jacobs University Bremen, Bremen, Germany, May 25th, 2007.

[19] draft-hui-6lowpan-interop-00 Interoperability Test for 6LoWPAN.

[20] Adam Dunkels. Contiki wiki. http://www.sics.se/contiki/wiki/.

[21] Adam Dunkels. uIP TCP/IP stack user guide. http://www.sics.se/~adam/uip/.

[22] Fredrik O¨ sterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, Thiemo Voigt, Cross-Level Sensor Network Simulation with COOJA, Swedish Institute of Computer Science, http://soda.swedish-ict.se/305/1/osterlind06crosslevel.pdf.

[23] N. Kushalnagar, G. Montenegro and C. Schumacher. IPv6 over Low-power wireless personal area networks (6LoWPAN): overview, assumptions, problem statement, and goals. RFC 4919.

[24] N. Kushalnagar, G. Montenegro, J. Hui and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944.

[25] Sergio Lembo, Jari Kuusisto, Jukka Manner, IN-DEPTH BREAKDOWN OF A 6LOWPAN STACK FOR SENSOR NETWORKS, International Journal of Computer Networks & Communications (IJCNC) Vol.2, No.6, November 2010 DOI: 10.5121/ijcnc.2010.2614 204.

[26] Jennic JN-UG-3053-JN5139-EK036-User-Guide-1v1.

[27] Jennic JN-UG-3054-6LoWPAN-APIs-1v0.

[28] http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip

[29] http://www.sics.se/projects/SICSlowpan.

[30] David E. Culler, Wireless Embedded Systems and Networking Foundations of IP-based Ubiquitous Sensor Networks 6LoWPAN, University of California, Berkeley Arch Rock Corp, July 11, 2007.

[31] Mathilde Durvy et al., Poster Abstract: Making Sensor Networks IPv6 Ready, http://www.sics.se/~adam/durvy08making.pdf.

[32] Nicolaie L. Fantana, Senior Member, IEEE, Till Ridel. A pragmatic Architecture for Ad-hoc Sensing and Servicing of Industrial Machinery

[33] Markus Scholz, Till Ridedel and Christian Decker. A flexible architecture for a robust indoor navigation support device for firefighters