

# Demo Abstract: Seamless Sensor Network IP Connectivity

Geoff Mulligan<sup>\*</sup>, Colin O’Flynn<sup>†</sup>, Julien Abeille<sup>\*</sup>, Mathilde Durvy<sup>\*</sup>, Patrick Wetterwald<sup>\*</sup>, Blake Leverett<sup>\*</sup>, Eric Gnoske<sup>\*</sup>, Michael Vidales<sup>\*</sup>, Nicolas Tsiftes<sup>‡</sup>, Niclas Finne<sup>‡</sup>, Adam Dunkels<sup>‡</sup>  
geoff@mulligan.com, coflynn@newae.com, {jabeille,mdurvy,pwetterw}@cisco.com,  
{blake.leverett,eric.gnoske,michael.vidales}@atmel.com, {nvt,nfi,adam}@sics.se  
• Proto6 LLC, <sup>†</sup> NewAE, <sup>\*</sup> Cisco Systems, <sup>\*</sup> Atmel Corporation, <sup>‡</sup> Swedish Institute of Computer Science

**Abstract**—Most existing sensor network systems require custom gateways and tools to interface to the Internet. We present a seamless IP-based sensor network connectivity mechanism that allows a sensor network to be instantly connected to a PC and additionally to the Internet. We use an IP stack on the sensor network and a USB stick on the PC that acts as a wireless network card. The USB stick carries its own network drivers for both Linux and Microsoft Windows. By inserting the stick in a PC, the PC is instantly connected to the sensor network. This allows the PC to use standard tools, such as ping and web browsers to access the sensor network nodes and allows the nodes to send packets both to the PC and the Internet.

## I. INTRODUCTION

Even though many wireless sensor networks use IEEE 802.15.4, none provide seamless and direct IP connectivity of the sensor network to IP-based systems and the Internet. Such sensor networks require translation gateways to map their proprietary protocols to IP. This adds costs, complexity, requires additional training and knowledge and breaks many of the end to end models used in the Internet for reliability, confidentiality and authentication.

By using an IPv6-compliant protocol stack, the sensor network can be easily integrated into IPv6 networks and leverage existing tools, protocols, knowledge and networking infrastructure [4], [5], [9]. The IP-enabled sensor network can be managed with existing or readily available tools. Knowledge of IP network management can be applied to the sensor network. Additionally, development and programming of the sensor network does not require learning new network protocols or paradigms. Recent work has shown that by using a power-saving MAC protocol in the sensor network, IP-based sensor networks are as power-efficient as sensor networks based on proprietary or specialized mechanisms [7].

We run uIPv6, the world’s smallest fully compliant IPv6 stack [6], together with the SICSLOWPAN implementation of IPv6 transmission over 802.15.4 networks [8], [9] in the Contiki operating system [3].

We show how a set of IP-based sensors can be networked together and connected easily to a standard PC and via the PC to the Internet using only standard IP based protocols [2] and therefore without the need for special translation software. We also show how the nodes, the PC and Internet devices can communicate using standard software and tools, such as ping, telnet and HTTP. Contiki has supported IP-based sensor

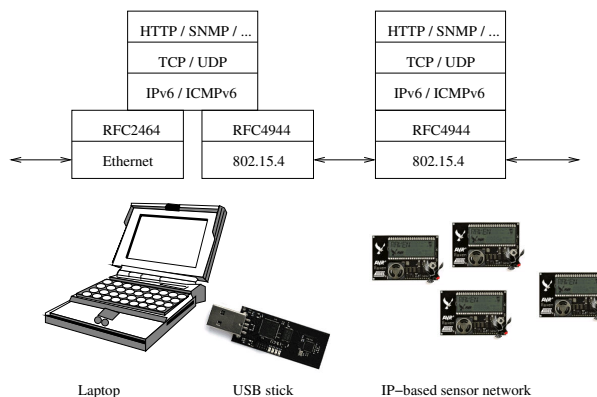


Fig. 1. The IP-based sensor network is seamlessly integrated with external networks through a USB stick inserted in a laptop or router. The sensor nodes and laptop/router runs an IP stack.

networks for several years [5] but until now, their integration with general-purpose IP-based networks have either required additional software on the router or non-standard access mechanisms such as Serial Line IP (SLIP).

## II. SYSTEM DESIGN

Our IP-based sensor network consist of a set of Atmel Raven boards, one PC laptop acting as an IP router, and one USB stick that bridges IP packets from the PC laptop to the sensor network. The USB stick provides a generic network interface to the PC laptop. The USB stick is supported in Linux, Windows XP, and Windows Vista. Linux does not require any driver file; Windows requires a simple INF file that tells Windows to use its built-in drivers.

The Raven boards include two Atmel microcontrollers, an ATmega 1281 with 128K Flash and 16K RAM to handle all the communications stack, an ATmega 3290 with 32K Flash and 16K RAM to control the onboard LCD and an Atmel AT86RF230 IEEE 802.15.4 compliant radio. In addition, the end nodes have temperature, light and audio sensors. The USB stick includes a single microcontroller, an Atmel AT90USB1287 with 128K of FLASH and 8K of RAM and an Atmel AT86RF230 radio. This microcontroller handles both the USB interface and the communications stack.

The Raven boards run uIPv6, our IPv6 Ready protocol

stack [6], [1], along with the SICSslowpan implementation of the 6LoWPAN adaption layer [8], [9], and a simplified, compliant IEEE 802.15.4 MAC on top of the Contiki operating system [3]. On top of uIPv6, we run a web server that provides access to the on-board sensors.

The USB stick runs the SICSslowpan header compression and decompression software and the necessary software for the USB interface and to present an Ethernet like interface to the connected PC. The USB stick acts like a stand-alone network card to the PC and a Zone Area Network controller (ZAN-controller) to the sensor nodes. The USB stick additionally presents a serial port to allow debug messages from the stack to be viewed.

#### A. Network Architecture

Figure 1 shows the network architecture of our demonstration. IP packets are routed by the PC laptop to and from the sensor network. The USB stick bridges the IP packets from the laptop and compresses and decompresses the IPv6 headers with 6LoWPAN header compression.

The sensor network uses a star and leaf topology. Within the Sensor Zone Area Network (S-ZAN), all sensor nodes communicate via a simple star. End-node to end-node communication is accomplished via forwarding by the ZAN controller. This initial demonstration does not provide multi-hop forwarding with the ZAN.

#### B. Address Assignment

The sensor nodes use IPv6 Stateless Address Autoconfiguration [11] to determine and set their global IP addresses. When the network is started, by the PC and USB stick, the PC requests an IPv6 prefix for the sensor network using DHCPv6-PD. Next, the PC starts listening for Neighbor Discovery (ND) [10] requests through the USB network interface. When the sensor nodes are powered up, the sensors first create a link-local IPv6 address by combining the standard IPv6 link-local prefix - fe80::0/64 - with its MAC address. The sensor node then performs Duplicate Address Detection (DAD) to ensure this address is not in use. Next, the node transmits an ND Router Solicitation message to autoconfigure its network address. The ZAN-controller responds with a Router Advertisement packet to the node.

#### C. Routing and Packet Forwarding

In general, packets in the sensor network can be sent from any node to any node in the sensor network, but in our demonstration, packets sent by the sensor node are sent through the default router, the ZAN-controller running on the USB stick. The ZAN-controller forwards any packets it receives that are not addressed to itself. If the Network Prefix is that of the local sensor network, the ZAN-controller forwards the packet to the destination endnode. In this way, sensor nodes can communicate with other sensor nodes within the local sensor network (the ZAN). If the Network Prefix is not on the local IPv6 network, the ZAN-controller forwards the packet via its USB interface to the PC. The PC will then intern

forward the packet as necessary and usually via its Default Route.

### III. EVALUATION

We demonstrate the usefulness of IP-based sensor networking by showing that regular IP-based network tools can be used to access, inspect, and manage the sensor network. Because the sensor nodes have IP addresses, a general-purpose web browser can be used to access the sensor nodes. Similarly, IP-based tools such as ping, traceroute, and telnet can be used to communicate with the network.

Because the USB stick acts as regular network interface on the PC, communication in the sensor network can be captured using a standard IP network capture tool such as Tcpcap or Wireshark. Similarly, the USB stick can be used on any nearby PC to capture the traffic without requiring physical access to the network router.

### IV. CONCLUSIONS

Traditional sensor networks have required proprietary tools and mechanisms for access and management. IP-based sensor networks simplify access to and management of sensor networks by using the open standard IP set of protocols. We leverage the base of existing protocols and technology to further extend the capabilities of sensor networks.

Our demonstration shows an IP-based sensor network that seamlessly connects a sensor network the Internet. We show that the severely constrained sensor nodes with limited Flash program space and even more limited RAM buffer space, it is possible to deploy these types of embedded IP systems allowing IP connectivity and network management and control without the need for complex translating gateways.

### REFERENCES

- [1] The IPv6 Ready Logo Program. <http://www.ipv6ready.org>.
- [2] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF, December 1998.
- [3] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Workshop on Embedded Networked Sensors*, Tampa, Florida, USA, November 2004.
- [4] A. Dunkels and J-P Vasseur. IP for Smart Objects, September 2008. The Internet Protocol for Smart Objects Alliance. White Paper 1.
- [5] A. Dunkels, T. Voigt, and J. Alonso. Making TCP/IP Viable for Wireless Sensor Networks. In *Proc. EWSN'04 work-in-progress session*, January 2004.
- [6] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels. Making Sensor Networks IPv6 Ready. In *Proceedings of the Sixth ACM Conference on Networked Embedded Sensor Systems (ACM SenSys 2008)*, Raleigh, North Carolina, USA, November 2008.
- [7] J. Hui and D. Culler. IP is Dead, Long Live IP for Wireless Sensor Networks. In *Proceedings of the 6th International Conference on Embedded Networked Sensor Systems (SenSys 2008)*, Raleigh, North Carolina, USA, November 2008.
- [8] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Internet proposed standard RFC 4944, September 2007.
- [9] G. Mulligan. The 6lowpan architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 78–82, New York, NY, USA, 2007. ACM.
- [10] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861, IETF, September 2007.
- [11] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862, IETF, September 2007.