User Defined Interactions between Devices on a 6LoWPAN Network for Home Automation

Thomas Gonnot and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— This paper presents a new concept of interaction among the devices within a 6LoWPAN network. More specifically, this concept defines a set of device classes with specific triggers and actions that can be linked to achieve certain tasks. For example, in home automation, an alarm clock can send a trigger command to a coffee machine and a radio when the alarm is activated. An important aspect of 6LoWPAN home automation network is the power management and security. The 6LoWPAN connects the devices within a network wirelessly at low power while maximizing the diversity and flexibility of the interconnections and communications. The 6LoWPAN neighbor detection allows for a "plug and play" utilization of the devices in the communication range of the network. Integration of 6LoWPAN requires minimal infrastructure modification, and consequently can become the preferred method for home automation. The interactions between the devices can be configured remotely by the user connected to a local network with the access control from other smart devices such as smartphones, tablets, computers or perhaps a server. In this paper, we present the architecture, design flow and implementation of a 6LoWPAN home automation platform to demonstrate the feasibility of the user defined interaction between the devices.**

I. Introduction

In the last decade, the number of connected devices increased at an unanticipated rate. This grow is partially due to the exploding market of smartphone and other smart devices. This emerging "Internet of Things" encourages more and more connected devices to be produced for diverse applications, including home automation [1]. One of the first interests of home automation is energy savings. It allows, for example, careful monitoring of the power consumption of devices across the house, or fine tuning of the thermostats; one can now control the temperature directly from his smartphone or computer from anywhere in the world through the Internet.

Another aspect is security. Houses are usually required to be equipped with smoke and CO detectors, and they can also implement some basic motion monitoring system combined with an alarm. Therefore, connecting all of these devices to the Internet can allow a better security by contacting the appropriate services or authorities.

Another important application of home automation is convenience and comfort. The house can be equipped, for example, with "smart" lighting and blind, and adapt the lighting to the actions of the users, or the time of the day.

The common problem of home automation at present-day is that there is no standardized protocol. The devices use the most common wireless standards such as Wi-Fi, Bluetooth or NFC, and rely on proprietary application to operate. Recently, a new protocol 6LoWPAN is considered for home automation. It is an implementation of the IPv6 protocol, compressed to maximize the efficiency of the IEEE 802.15.4 standard [2], used for low power communication, such as the ZigBee protocol.

In this paper, we describe a standard for the communication of devices in the context of home automation relying on 6LoWPAN. The objective is twofold: first provide a standardized way for the devices to communicate depending on their category, and then allow direct interaction of the devices between each other using a central node.

In this document, Part II introduces the 6LoWPAN protocol, while Part III describes the classification of the devices inside the home automation network. Part VI defines the central node used for the devices interactions and finally Part V describes the security features and the connection procedure for a new device in the network.

II. 6LoWPAN Implementation

The 6LoWPAN is a protocol based on the IPv6 protocol. It is designed to be used over the IEEE 802.15.4 standard for low power wireless communication. The issue with this standard is that the frames are limited to 127 bytes, including the MAC header of 23 bytes and an optional AES encryption header of 21 bytes. With a conventional IPv6 protocol, the remaining payload is reduced to 33 bytes for UDP and 21 bytes for TCP.

The idea of 6LoWPAN is to keep the IPv6 protocol, but compressing it to leave more space to the payload [3]. This way, transmitting a given amount of data takes less frames and consequently less power. The 6LoWPAN also removes the implementation of TCP that requires multiple back and forth transmissions to ensure the reliability of the transmission. Instead, the device is limited to UDP protocol, and the application layer is in charge of handling potential errors, giving the possibility to delay the transmission when it is more convenient. As a result, the payload can vary from 65 to 75 bytes of data. Figure 1 shows the difference between the

978-1-4799-3312-9/14/$31.00 ©2014 IEEE

normal IPv6 headers compared to the 6LoWPAN compressed header on an IEEE 802.15.4 frame.

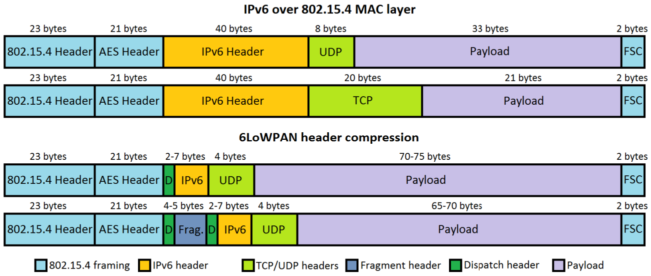

Figure 1. IPv6 and 6LoWPAN framing over 802.15.4 standard

The 6LoWPAN implements functionalities of network discovery introduced by IPv6 as neighbor discovery. In contrast to a wired network's topology that doesn't change by itself, a 6LoWPAN network can change topology depending on the device movements, power status or even device activity. The devices in an IEEE 802.15.4 network are divided into three devices categories:

- The Reduced-Function Devices (RFD) implements all the functions to discover and communicate through the networks, but are limited to simple communications.
- The Full-Function Devices (FFD) implements all the function of the network including the routing capabilities.
- The PAN coordinator is a Full-Function Device that is the central node of the network. It is often referred to as border router when it has access to another network, such as Internet for example.

Figure 2 shows an example of mesh network. The RFD devices too far from the PAN coordinator relay their packets through the FFD devices.

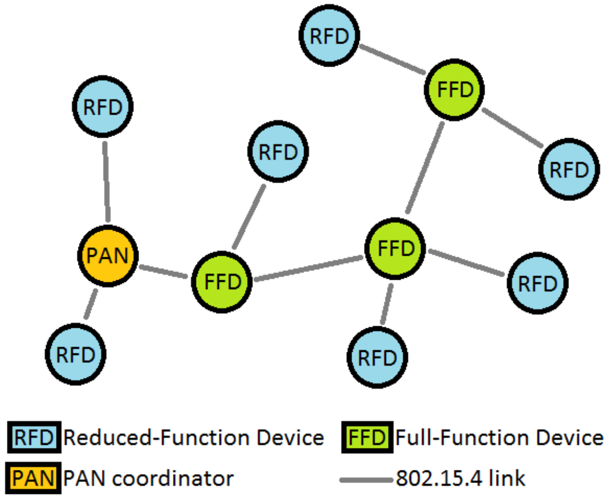

Figure 2. Example of mesh network with 6LoWPAN

III. Home Automation Devices Protocol

Even if the network part is fully determined by the 6LoWPAN protocol, nothing defines what happens on the application layer. The goal of this paper is to standardize the communications depending on a specific device type. For example, the type of data transmitted by a smoke detector will not be the same as the data transmitted by a thermostat. Also some devices provide information, some others accept commands, or there can be devices doing both.

The organization of the data in the packets has to match the type of device that is sending or receiving the data. Therefore, a new protocol can be implemented, referred here as Home Automation Devices Protocol (HADP). This protocol defines a data structure adapted for the various device types and provides a support for possible loss of data in the network. Furthermore, the protocol defines the type of information each device can provide to describe itself as for its name, its functions, and the type of data it communicates. Upon connection, a new device is interrogated by the PAN coordinator to provide its identification and description. The devices can also be organized in sub-functions in order to allow splitting apart composite devices. It can also allow several devices to be connected together to the same radio transducer to save power or space.

The packet headers are organized for a total of 3 bytes as:

- A packet identifier, on 8 bits, incremented for each new packet, randomly initialized for the first frame.
- A packet type field, on 4 bits, defining the type of data in the frame: descriptor, data, command, acknowledgment or error types.
- A sub-device identifier, on 4 bits, identifying individual devices sharing the same radio transducer.
- A packet size field, on 8 bits, defining the size of the frame, up to 256 bytes.

For each type of frame, there is a different structure. In the case of the descriptor, the first byte defines the type of information, and the second byte defines the size of the data. Several descriptors can follow each other's on the same frame. The descriptor types are listed in TABLE I.

TABLE I. Different descriptors available

Field Name	Type	Description
Device Name	String	Name of the device (Required)
Manufacturer code	2 Bytes	Code of the manufacturer of the device
Product code	2 Bytes	Code of the product
Device serial	2 Bytes	Device serial number
Command types	N bytes	Number of available command channels and type of each one of them
Data types	N bytes	Number of data channels available and type of data for each one of them

For each command channel, the device provides the type of control, or action. These are still to be defined, but can include, for example, power outlets, lights, coffee maker or blinds. This is used by the PAN coordinator to control the behavior of the device. When doing so, the frame starts with the first byte corresponding to the control channel number.

The same principle is used for the data channels, or triggers. When the device transmits data to the PAN coordinator, it sets the first byte after the frame header to the data channel number. The data rate can be eventually configured using one data channel.

For critical data or command, the transmitter will require an acknowledgement from the receiver. If the acknowledgment is not received, the transmitter can resend the data with the same frame identifier. On the same principle, in case a packet is missing or corrupted, the receiver can send an error frame, containing the packet identifier, packet type and sub-device identifier to ask for a retransmission.

IV. Central Node and Devices Interactions

The point of home automation is to allow devices to interact all together. In case of the HAD protocol, the router is used as a central node that will collect all the information from all the devices. Consequently, the central node allows a multitude of different interactions. As mentioned before, the devices are required to provide specific information that describes them. These information are collected and memorized by the central node to allow the user to create links. However, the use of 6LoWPAN also allows the central node to be on another network, due to the IPv6 encapsulation of the data. In this case, the PAN coordinator is used as a router to the other network or the Internet and the central node can be located anywhere [4]. Figure 3 shows the central node as part of a local network, whereas Figure 4 shows the central node as a deported server accessed through the Internet.

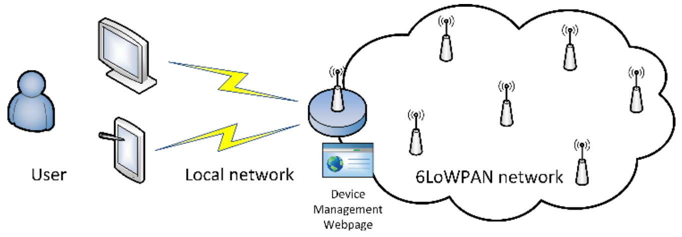

Figure 3. Home automation network with local central node

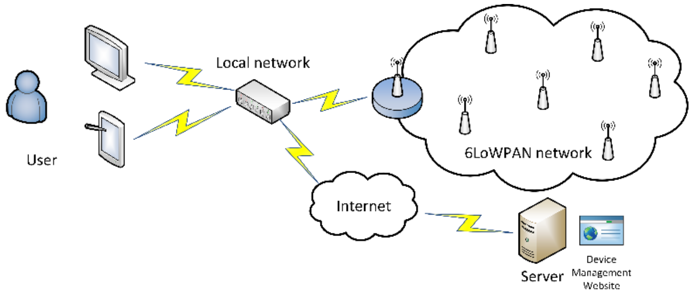

Figure 4. Home automation network with deported central node

Regardless of local or deported, the user can access the configuration of the network directly from the central node, using its computer to connect to it using a specific program or a web browser. The interface then allows the user to see the current devices connected to the network as well as the devices that were previously connected. For each device, a list of actions is proposed and the user can decide to connect the functionalities of two devices together. A simple example would be a network containing a coffee maker and an alarm clock. It would become possible for the user to program the coffee maker to start whenever the alarm rings. On the same pattern, if a smoke detector is triggered in one room, the power outlets can be disconnected automatically.

V. Security and Device Connection

The home automation involves collecting data that can be highly private, and that should remain confidential. Also, the system can in term be in control of several critical devices. Therefore, the network should be protected from any attempt of intrusion. The IEEE 802.15.4 standard defines an optional 21 bytes header for AES encryption of the frames. This allows the communication to be protected using a key that is common to the network.

In order to connect a new device on the network, it must have the encryption key of the network already in memory. The problem is that most of the devices for home automation offers only little interfacing with the user, and entering the key manually can become fastidious. One solution is developing a procedure to securely transmit the key to the device on its first connection.

One possible solution for a “plug and play” connection is to use a Near Field Communication (NFC) tag. The device itself can come with a NFC tag embedded, or included with a user manual, containing the initial key programmed in the device. Using a smartphone, or the PAN coordinator if equipped accordingly, the user read the NFC tag to start an automated secured negotiation. The PAN coordinator then sends the encrypted encryption key, and resumes its normal operations. The device uses the new encryption key to connect to the network and communicate its capabilities according to the protocol described in Part III. Figure 5 illustrates the first time connection procedure.

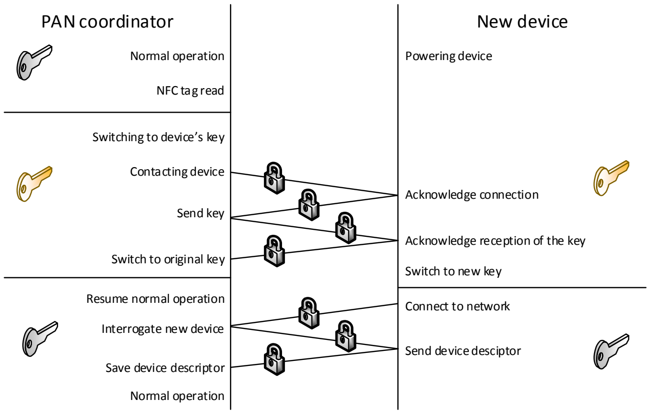

Figure 5. New device connection procedure with key exchange

VI. Current Work

After defining the protocol to be used, it has to be tested on a real system. The system used to test the HADP protocol is composed of devices from STMicroelectronics, more specifically the STM32W series. It features a Cortex-M3 combined with a set of peripherals such as GPIOs, ADCs, DAC, timers, RTC clock, USART, SPI, I2C and most importantly, a 2.4GHz IEEE 802.15.4 MAC radio. In this study, we are using a development kit providing a USB serial to IEEE 802.15.4 adapter, and a remote unit with a battery holder, several push buttons and an accelerometer. Figure 6 shows a picture of the development kit.

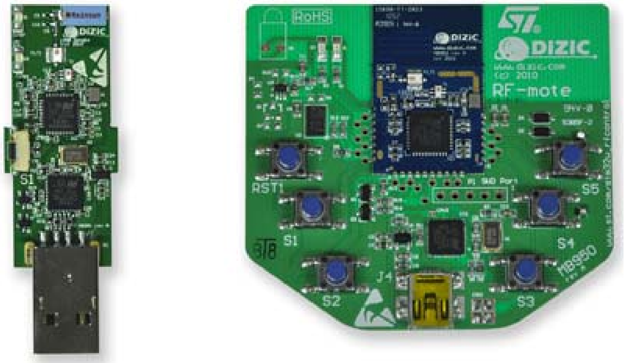

Figure 6. STM32W-RFCKIT development kit

The Contiki open source operating system is used in our study. It contains an IP stack providing full support for the 6LoWPAN implementation, and is supporting the STM32W chips. It is also lightweight and a widely used OS for the Internet of Things.

The testing is performed using a set of four devices. One is connected to a computer and acts as a PAN coordinator. Two nodes are on batteries and transmitting simulated data to the PAN coordinator. One last device is used only as an IEEE 802.15.4 protocol analyzer and register the traffic between the devices. The network is tested in various conditions, depending on the type of data transmitted, or the network topology.

VII. Conclusion

This paper presents a concept of protocol for interactions among the devices within a 6LoWPAN network. The protocol is designed to standardize the communications in the context of home automation, but can be extended to wider networks for larger areas and more diverse set of devices. It also defines a central node to allow a user to control the devices interaction simply from its computer or smartphone, in local, or from the Internet. Finally, it can implement a “plug and play” connectivity for new devices for a simpler setup by the user.

References

[1] B. Proffitt, "How Big The Internet Of Things Could Become," 30 September 2013. [Online]. Available: http://readwrite.com/2013/09/30/how-big-the-Internet-of-things-could-become.

[2] "IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003),* pp. 1-320, 206.

[3] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," *RFC 6282,* 2011.

[4] J. Hui, D. Culler and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture," *Internet Protocol for Smart Objects (IPSO) Alliance - White paper #3,* 2009.