Journal of Engineering and Applied Sciences 13 (Special Issue 9): 6986-6991, 2018
ISSN: 1816-949X

Using IPv6 Over Bluetooth Low Energy on Low Costs Platforms

[1]Jan Stepan,[2]Richard Cimler and [1]Jan Matyska
[1]Faculty of Informatics and Management,
[2]Center for Basic and Applied Research (CZAV), Faculty of Informatics and Management,
University of Hradec Kralove, Hradec Kralove, Czech Republic

Abstract: It is common that almost every current smart device is able to communicate wirelessly. There are multiple technological solutions for wireless communication. They vary in transfer rates, maximum range or in transmission method. Bluetooth is currently dominant technology for data exchange in short distances. This study describes the latest branch of Bluetooth technology, Bluetooth Low Energy (BLE) and how to use it with Internet Protocol version 6 (IPv6) on low costs platforms. This approach offers simple and fast way how to implement wireless sensor network which communicates through a router with other services. Local and global link address distribution is described as well. An implementation example of wireless nodes using IPv6 over BLE integration into in smart-home project Hausy is presented.

Key words: Smart home, IPv6, Bluetooth, low energy, BLE, distances

INTRODUCTION

Internet of things is very popular topic these days. Idea of every possible thing connected to a cloud or a server service can make huge improvement in quality of life, energy savings or cost reduction with predictive maintenance. Example of predictive maintenance can be freezer in shop which can monitor inside temperature and door opening count. Based on measured data maintenance can be called before malfunction occurs based on statistical data from other freezers.

In order make this common feature, cheap and simple method how to connect devices to internet must exist. Some devices can be connected by Ethernet cable connection but that is not solution for most cases because many devices are mobile or wearable. Wireless WiFi connection seems like ideal approach (Krejcar and Cernohorsky, 2008; Behan and Krejcar, 2013) but it has large power requirements so long-term battery operation is not possible. Krejcar and Mahdal (2012) problem is addressed by new version of Bluetooth called Bluetooth low energy which is not backwards compatible. Main goal of this technology is to reduce power consumption during communication. Devices are able to work for many months while powered only by cheap coin cell batteries (Tei *et al*., 2015).

This study presents latest iteration of Bluetooth low energy number 4.2. It presents simple way how to encapsulate IPv6 packets into Bluetooth packers. Combination of edge routers and BLE presents solution how to have small devices with global IP address accessible anywhere from internet. Research on similar topic has been made by Wang *et al*. (2013). That research was focused only on communication between PCs with IPv6 using linux blueZ. Before introduction of BLE the most power saving communication technology was ZigBee. Study (Siekkinen *et al*., 2012) is focused on implementation of IPv6 over BLE on embedded devices. Solution which uses BLE devices to extend existing home automation system Hausy by wireless nodes is described in this study.

MATERIALS AND METHODS

Bluetooth low energy: All previous iterations of Bluetooth were backwards compatible and focused on audio/video streaming or peer to peer data transfer. In 2010 Bluetooth SIG presented Version 4.0. New technology called Bluetooth low energy (sometimes called Bluetooth smart and previously known as Wibree) (Kooker, 2008) has been introduced. This technology is not backwards compatible with standard Bluetooth (now called Bluetooth classic), although, it communicates on same 2.4 GHz baseband. Primary goal of BLE is to reduce power consumption and enable possibility of using coin cell battery to power devices with long operating time. While Bluetooth classic aims for audio, video or high speed data transfer devices, BLE aims for

Corresponding Author: Jan Stepan, Faculty of Informatics and Management, University of Hradec Kralove, Hradec Kralove, Czech Republic

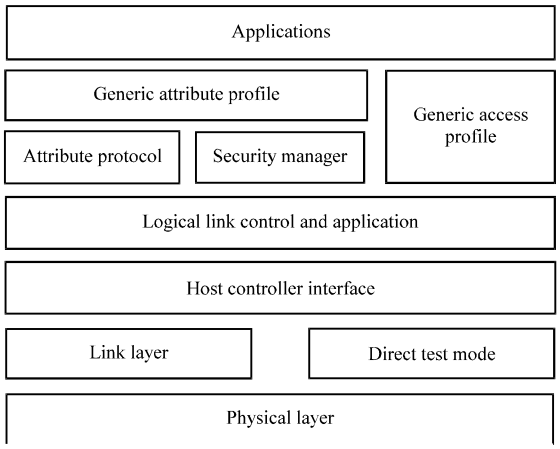

Fig. 1: Architecture of Bluetooth low energy

smart accessories as wearable devices or intelligent home devices. There are transmitters capable of communicating through both Bluetooth classic and BLE. This functionality is called dual-mode. Transmitters which support only BLE are called single-mode transmitters.

Main difference of BLE from Bluetooth classic is significantly reduced data rate to only 1 Mbit/sec with application throughput approximately 250 kbit/sec. This transfer rate is more than sufficient for target usage but not sufficient for multimedia application. That is why Bluetooth classic is still under development and continues to be supported.

General attribute profiles: The most significant change is architecture of BLE which is shown on Fig. 1. Every application now should use Generic Attribute profile (GATT) which specifies roles, characteristics, services and descriptors of device. Role can be either client or server. Client device initiates GATT commands and requests and gets responses. Client can be for example smartphone or computer. Server in the opposite receives GATT command and requests from client. Server can be for example heart rate sensor, temperature sensor or fitness band. Characteristic is data value which is transferred between client and server. For example characteristic in temperature sensor is temperature itself. Service describes collection of related characteristics and access direction of these characteristics. This means client can read characteristic, write characteristic or both of those from server.

Multiple services can be defined on one device. For example above mentioned temperature sensor will have temperature service with defined temperature data and sampling interval data. Another service can be battery service which will tell client current state of battery. Descriptors are optional and provide additional information about characteristics such as units in temperature sensor. Services, characteristics and descriptors are collectively called as attributes. All attributes are identified by Universally Unique Identifiers (UUIDs). Each BLE implementer is allowed to pick random or pseudo-random UUID for own proprietary use with exception of reserved range of UUIDs by Bluetooth special interest group. UUIDs are 16 bytes long numbers but can be represented for better efficiency as 2 or 4 bytes. Unlisted part of UUID is then considered to be zero. Bluetooth low energy device can be peripheral, central or both. This role has no relationship with client/server. Common case is when periphery acts as server and central acts as client. Opposite situation is possible but could not be supported in all available BLE transmitters. Difference between peripheral and central device differs in who connects to whom. Peripheral device broadcasts information about itself and central device scans for it. Every BLE device has own unique address very similar to MAC address used in WiFi or Ethernet devices. Ranges of addresses are provided to hardware manufactures by Bluetooth SIG. Address is 6 bytes long written as single hexadecimal bytes separated by colon. For example 12:34:56:78:9A:BC. This means that 248 unique devices can be made. When BLE peripheral broadcasts its name and GATT service UUID, LE Advertisement packet is used. This packet contains physical address, all services and characteristics, manufacturer id, manufacturer specific data and connectable information. If device is only broadcasting (acts like beacon) its parameter "Connectable" is set to false. If device is connectable, secure pairing and connection is done by BLE hardware. All communication in connected mode is encrypted with 128 AES cipher. Bluetooth SIG does not specify maximum number of multiple connections between central and peripherals but it depends on hardware manufacturer.

Hardware: BLE radios come in different forms and offers different level of abstraction. All major mobile operating systems offers API to work with GATT services and hardware itself is hidden from developers. BLE is also supported on PCs or laptops with Windows, Linux or OS X. These operating systems have abstract API to work with GATT services. Hardware radios are most often built in laptops or are available as external USB dongles.

Several microcontrollers from various manufacturers are available for peripheral development. Microcontrollers are often referred as Systems on Chip (SoC) solutions because no other active components are required for full functionality. Many of them are user programmable and have enough memory for BLE protocol implementation and user application. Overview of some available SoC on the market is shown in Table 1.

Table 1: Comparison of available system on chip solution

Manufacturer	Nordic semiconductor	ST	Texas instruments	Cypress semiconductor	Dialog semiconductor	CSR	Microchip
Model	nRF52832	BlueNRG	CC2541	PsoC 4 BLE	DA14680	CSR1013	RN4020
Architecture	32 bit, ARM cortex M4F	32 bit, ARM cortex M0	8 bit, 8051	32 bit, ARM cortex M0	32 bit, ARM cortex M0	16 bit RISC	32 bit RISC
RAM (kB)	64	N/A	8	16 or 32	128	64	N/A
Flash (kB)	512	N/A	128 or 256	128 or 256	Ext., 16 OTP	64	N/A
PIN count	48	32	40	56	60	34	24
Peripherals	SPI, 12C, UART PWM, RTC, PDM	SPI for control	12C, UART, ADC	SPI, 12C UART, PWM	SPI, 12C, UART, PWM, ADC, DAC	SPI, 12C, UART, PWM, ADC, DAC	UART
Operation Voltage (V)	1.7-3.6	1.7-3.6	2-3.6	1.7-5.5	1.7-5	1.6-4.4	1.8-3.6
Standalone	YES	NO	YES	YES	YES	YES	NO
BLE compatibility	4.2	4.1	4.0	4.2	4.2	4.1	4.1

Only the most performance powerful chips were selected to demonstrate that any possible sensor can be used. This table is not complete overview of all BLE SoC manufacturers. Every manufacturer mentioned in this table offers more than one chip. Some SoC are standalone which means that no other micro-controller is needed for operation. Others require secondary micro-controller which controls primary SoC through interface. Advantage of this approach is possibility to use existing micro-controller that developer is used to work with. Disadvantage is increased power consumption because two micro-controllers are operating at once and also firmware updates are more difficult. Level of Bluetooth support is very various. Some SoC only offers hardware for transmission of BLE packets but implementation is left on developers. This can increase amount of time required for development.

Use case scenarios: Typical use case scenario with BLE is located periphery near user or worn by user such as wearable device. This periphery communicates through BLE with users smartphone, tablet or computer. Application is running on this device and reads or writes data from or to periphery through operating systems API. Translation layer must be implemented if application uses some kind of cloud service or server connection. Role of this layer is to pass data from BLE application interface to service. For example REST interface with JSON messages. This scenario when periphery belongs to single user, may be called as user local scenario.

Other scenario is when periphery is present inside building and is operated by multiple users. Best example of this case is smart lightning. Some users want to use buttons, some want to use mobile phone application and building manager wants to use server application. If light bulb is controlled by BLE it needs to maintain persistent connection with dedicated computer and this computer waits for commands from server application via WiFi, Ethernet or fiber. Every user can manipulate with light bulbs using connection to server application but it is quite expensive to have server only to translates server API into GATT commands. Better solution is to connect peripherals to a router directly. This approach can be used in different types of architectures and is now possible with the latest specification of BLE 4.2. It defines procedures for native implementation of IPv6 protocol. Peripherals therefore can use any protocol based on IPv6 and can be accessed from internet by global address.

Implementing IPv6 on BLE: Latest version of Bluetooth low energy number 4.2 fully adopts draft specified by Internet Engineering Taskforce (Nieminen *et al*., 2015). This document describes how IPv6 is transported over Bluetooth LE connections using IPv6 over low power wireless personal area networks (6LoWPAN) techniques. Internet protocol support service with UUID 0x1820. Periphery capable of IPv6 must broadcast this service in advertisement packet. This service has no defined characteristic. Instead of it, binary data with IPv6 packets are transmitted in connected mode. Older version of Internet protocol called IPv4 is not supported. IPv6 is in good adoption state all around world and because public IPv4 addresses are almost run out, support of only new standard is a good approach. Peripheral devices capable of IPv6 communication should be referred as nodes because they are end points in network.

Edge routers: Nodes connects to central device which is referred as edge router. Goal of edge router is to route packets between BLE to WiFi, Ethernet or fiber internet connection to enable accessibility of nodes anywhere from internet.

These edge routers are not yet available on the market. But there is simple and affordable way how to create them by using embedded computers. Latest version of Raspberry Pi has now integrated WiFi, Ethernet and BLE radio. Raspberry uses ARM architecture processor and has 1GB of RAM memory. Multiple Linux distribution

can be used for edge router implementation. Usage of IPv6 in Linux requires kernel version at least 4.0 or above. These kernels are now available in any recent Linux distribution and contains Bluetooth 6lowpan module and BlueZ, official Linux Bluetooth low energy stack implementation. This module implements IPv6 broadcasting and can be controlled by writing into specific file. Unfortunately automatic node connection is not supported yet and must be done manually.

When module is loaded (it could be during boot or manually by modprobe command) file/sys/kernel/debug/bluetooth/6lowpan control is created. Physical address of nodes can be obtained with hcitool-lescan command. Command returns list of BLE devices in range with their address and name if is broadcasted. To ensure that device is IPv6 capable, list of GATT services can be obtain by using gatttool command which is part of BlueZ Software. Connection between router and node is made by writing connect PHY 1 into 6lowpan control file. PHY needs to be replaced with physical address of device. This procedure can be automated by writing script which performs these tasks periodically. New network device named bt0 is created after first node is connected and acts same as any other Linux interface.

Obtaining IP address: After BLE connection is established between router and node, sending IPv6 packets can begin. Every packet must contain source and destination IP address. If router wants to send packet to node it is not possible because does not know IP address of node. This problem is Solved by Stateless Auto-Configuration (SLAAC) and Router Advertisement.

First step is for node to create its own local address. Three transformation must be made to create link-local public address (Carpenter and Jiang, 2014). First transformation is to extend 48 bit Bluetooth physical address to 64 bit which is used as MAC address in IPv6. To do this, bytes 0xFF and 0xFE must be add in the middle of address. If device has address 12:34:56:78:9A:BC, the result of first step is 12:34:56:FF:FE:78:9A:BC. Second transformation is to set seventh bit of first byte to logical one. After this set is Bluetooth address transformed into IPv6 IID. Last transformation is to add local-link prefix FE80::/10 to IID. Result address in this example will be FE80:0000:0000:0000:1234:56FF:FE78:9ABC and can be shorten to FE80::1234:56FF:FE78:9ABC.

Second step is to setup Router Advertisement Daemon on Linux with RADVD software. Router advertisement is standard part of ICMPv6 protocol and can be used for obtaining global public IP address. RADVD can be set to provide global IPv6 address prefix for specific interface. In this case it is bt0 network interface. Also, routing IP addresses between internet interface and bt0 interface must be set.

When node gets to connected state, it either waits for periodic router advertisement message which contains global IP address prefix or sends or calls for this message immediately by sending multicast Router solicitation message packet. Node generates IP address same as in link-local scenario only with different prefix. Router then sends neighbor discovery message and waits for neighbor solicitation message with node new global IP address. Router checks this address for possible duplicate and if address is unique, new routing rule is add to system. After this node can use its global IP address which can be accessible anywhere from internet.

RESULTS AND DISCUSSION

Implementation in Hausy project: Only theoretical aspects of using Ipv6 has been described in this study, so far. This technology is already implemented in currently developed smart home project. Home automation system (shortened as Hausy) has three layer architecture. Lowest layer contains wired nodes which are using 8 bit family microcontrollers from microchip. Different sensors, actuators or controllers can be connected to micro-controllers. On the other hand, only one type of sensor can be connected to node. Count of connected peripheries is saved in micro-controller as channel count. Peripheries communicate using industry standard RS485 bus with middle layer called subsystem. Diagram of this architecture is shown on Fig. 2 and described detailed by Horaek *et al*. (2015).

Subsystem is using popular Raspberry Pi 3 ARM minicomputer. There are two applications, controller written in C++ language a local business logic written in Rust. Controller is responsible for detecting new nodes and reading or writing (depends on node type) states of nodes. Local business application is just a transparent gateway between controller and upper layers in default state. Top layer is either local server or cloud hosted service. This layer contains rules for home automation for every present node and subsystem. For example turn on air condition when temperature in kitchen is greater than 30°C: If (subsystem1.node2.channel1 ¿ 30) subsystem 3.node5.channel 2 = 1.

If network connection fails or server suffers fatal hardware fault, local business application in subsystem has latest rules for locally connected nodes and takes control of them until problem is fixed.

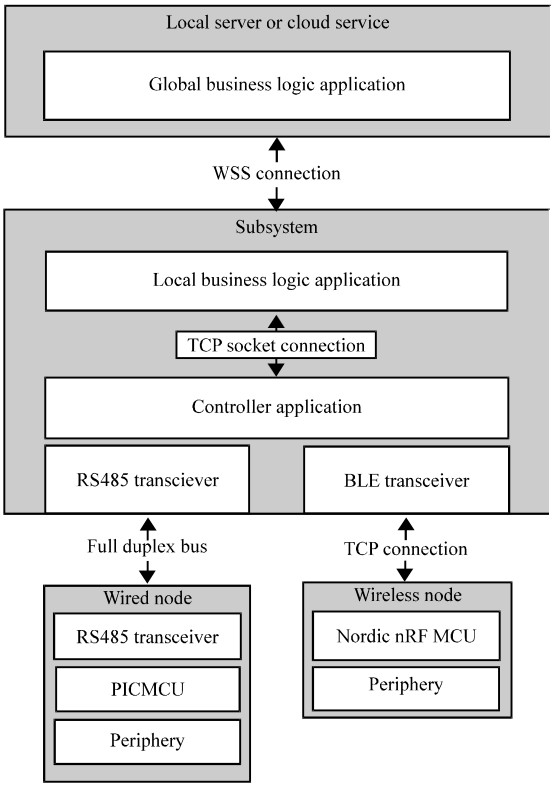

Fig. 2: Architecture of IPv6 enabled network

Wired nodes are good for new home installations and for transmission critical data. Sometimes it is not possible to install cables into existing environment. For this reason Wireless node must be implemented. Easiest way how to start developing with IPv6 on BLE is by using nRF52832 SoC from Nordic Semiconductor. It does not mean that nRF52832 is only suitable solution for using IPv6 over BLE. It has been chosen for this project because at this time it offers the most production ready and finished software development kit. Implementation of proposed solution can be achieved with any SoC that offers direct GATT manipulation, enough memory and capabilities to receive or transmit IPv6 packets.

BLE stack implementation by Nordic is proprietary and is available as precompiled SoftDevice hex file for their SoC. This hex file starts from first byte of FLASH memory. User application can be written in ANSI C language and compiled with open source GCC for ARM compiler and linked after this hex file. User application can access Bluetooth functions in SoftDevice by well documented API. Stack takes approximately 96 kB of FLASH memory and 16 kB of RAM memory leaving more than sufficient 416 kB of FLASH and 48 kB of RAM to user application. Only BLE communication and IPv6 packet verification is implemented in SoftDevice and higher protocols are implemented in user memory region.

Nordic IoT SDK comes with their own implementation of some popular protocols based on IPv6. Plain TCP socket is used for node implementation in Hausy. Node is broadcasting IPv6 service. When scanning for new nodes is initiated on server or cloud, controller application is using BlueZ API to find them. Link-local address is calculated by previously mentioned rules in controller application. No router advertisement is used in this approach because nodes are part of internal system and do not require global IPv6 address. Node starts to listen as TCP server on 9000 port when IPv6 connection is established by 6lowpan kernel module. Controller application then connects as client and creates socket connection through file descriptor. TCP connection is shutdown when node is out of reach from subsystem. Controller application then periodically scans for node and automatically reconnects when is in range again.

Same communication protocol is used for wireless sensor and wired sensors. This protocol is described in detail in (Stepan *et al.*). Only difference is that wired nodes uses request reply scenario because two nodes cannot send data at same time. This would cause power shortage on bus. Controller application is sending requests for specific node to acquire latest periphery state. In wireless nodes where every node connection is handled in separate thread, nodes can send new state only when change on periphery occurs. This allows to increase battery life because node can enter power saving mode and be waken only by external periphery. After implementation of this technology communication from node to cloud can be encrypted (Holik *et al.*, 2015).

CONCLUSION

This study presented usage of Bluetooth low energy technology for communication over IPv6. This solution is shown on the implementation example in the smart home project Hausy. Using IPv6 which is embedded into generic attribute profile is simple and cheap solution for internet of things implementation. Any device is able to connect to server or cloud service with global IPv6 address assigned by edge router. Edge router is used as a gateway between Bluetooth connection and common internet connection like WiFi or Ethernet. Edge routers are not yet available on market but it is possible to create them by using all-in-one ARM computers like Raspberry Pi. All connection is provided by IP protocols, thus, secure SSL or TLS encryption can be used. Devices

connected through the internet can share data which can be statistically analyzed and used for different purposes such as predictive maintenance.

ACKNOWLEDGEMENT

The support of the Specific research project at FIM UHK and Czech Science Foundation GA_CR #15-11724S is gratefully acknowledged.

REFERENCES

Behan, M. and O. Krejcar, 2013. Modern smart device-based concept of sensoric networks. Eurasip J. Wireless Commun. Networking, 2013: 1-13.

Carpenter, B. and S. Jiang, 2014. Significance of IPv6 interface identifiers. Internet Eng. Task Force, 2014: 1-10.

Holik, F., J. Horalek, S. Neradova, S. Zitta and O. Marik, 2015. The deployment of security information and event management in cloud infrastructure. Proceeding of the 25th International Conference on Radioelektronika (RADIOELEKTRONIKA), April 21-22, 2015, IEEE, Pardubice, Czech Republic, ISBN:978-1-4799-8119-9, pp: 399-404.

Horaek, J., J. Matyska, J. Stepan, M. Vancl and R. Cimler *et al*., 2015. Lower Layers of a Cloud Driven Smart Home System. In: New Trends in Intelligent Information and Database Systems, Barbucha, D., N.T. Nguyen and J. Batubara (Eds.). Springer, Berlin, Germany, ISBN:978-3-319-16210-2, pp: 219-228.

Kooker, J., 2008. Bluetooth, zigbee and wibree: A comparison of WPAN technologies. Can. Securities Exch., 20: 1-4.

Krejcar, O. and J. Cernohorsky, 2008. Database Prebuffering as a Way to Create a Mobile Control and Information System with Better Response Time. In: Computational Science, Bubak, M., G.D.V. Albada, J. Dongarra and P.M.A. Sloot (Eds.). Springer, Berlin, Germany, ISBN: 978-3-540-69383-3, pp: 489-498.

Krejcar, O. and M. Mahdal, 2012. Optimized solar energy power supply for remote wireless sensors based on IEEE 802.15. 4 standard. Int. J. Photoenergy, 2012: 1-9.

Nieminen, J., T. Savolainen, M. Isomaki, B. Patil and Z. Shelby *et al*., 2015. Ipv6 over bluetooth (r) low energy. Internet Eng. Task Force, 2015: 1-21.

Siekkinen, M., M. Hiienkari, J.K. Nurminen and J. Nieminen, 2012. How low energy is bluetooth low energy? Comparative measurements with zigbee/802.15. 4. Proceeding of the Conference on Workshops Wireless Communications and Networking (WCNCW), April 1-1, 2012, IEEE, Helsinki, Finland, ISBN: 978-1-4673-0681-2, pp: 232-237.

Tei, R., H. Yamazawa and T. Shimizu, 2015. BLE power consumption estimation and its applications to smart manufacturing. Proceeding of the 54th Annual Conference on the Society of Instrument and Control Engineers of Japan (SICE), July 28-30, 2015, IEEE, Tokyo, Japan, ISBN: 978-4-9077-6448-7, pp: 148-153.

Wang, H., M. Xi, J. Liu and C. Chen, 2013. Transmitting IPv6 packets over bluetooth low energy based on bluez. Proceeding of the 15th International Conference on Advanced Communication Technology (ICACT), January 27-30, 2013, IEEE, Beijing, China, ISBN: 978-89-968650-1-8, pp: 72-77.