An Optimized OPC UA Transport Profile to Bringing Bluetooth Low Energy Device into IP Networks

Ganesh Man Shrestha, Jahanzaib Imtiaz
inIT-Institut Industrial IT
Ostwestfalen-Lippe University of Applied Sciences
Liebigstraße 87, 32657 Lemgo, Germany
{ganesh.shrestha, jahanzaib.imtiaz}@hs-owl.de

Jürgen Jasperneite
Fraunhofer IOSB-INA Application Center
Industrial Automation
Liebigstraße 87, 32657 Lemgo, Germany
juergen.jasperneite@iosb-ina.fraunhofer.de

Abstract

Because energy efficiency is gaining more importance these days and Bluetooth Low Energy (BLE) could be used to make use of potential everyday objects into Internet of Things (IOT) - a software, platform and vendor independent common service interface that can be used in such low resource devices has high potential. OPC UA is an emerging middleware solution that addresses the above points but is bulky due to its abundant features. Further optimization is necessary to bring the OPC UA into such resource-limited devices. We have scaled down the OPC UA protocol stack footprint down to the chip level [16].

In this paper, we propose an optimization approach to minimize the OPC UA network footprint.

1. Introduction

BLE is a new distinct feature adapted from the classic Bluetooth (BT) standard and optimized for lowest possible power consumption. The BLE was integrated with BT 4.0 specification in 2010 [1]. A lot of new applications can be achieved by bringing the BLE devices in the IP network. Due to reduced complexity and power consumption, BLE can be realized in sports equipment, wearable devices, health care, automotive, home and industrial automation. Some typical use cases can be : 1) A car or drivers watch[1] communicating with a passing car to know about the availability of the parking space or the traffic situation, 2) sensors in the wheels or oil tank informing the driver about the air pressure or the oil level, 3) a patient's health condition being remotely monitored by a doctor in the hospital, 4) sensors in the factory floor communicating with watch or smart phone of the operator to indicate the occurred error or the status notification.

A typical scenario exhibiting the use of BLE devices in the IOT is shown in Figure 1 [2]. As per the paper [2], the BLE devices become part of the IP network each with a unique IP address. The gateway device is used as the router to provide internet connectivity to the BLE devices. BLE devices are proposed as the enablers of IOT and are gaining popularity in wide range of applications. 6LoWPAN working group of the IETF has proposed a draft of the standardization activity to bring BLE device into IP network [3]. A prototype solution based on the standardization of 6LoWPAN has been published in the beginning of the year [2]. We propose an enhancement to this solution by providing an OPC UA based common service interface. The OPC UA is one of the widely refereed platform-independent industrial middleware standards. Due to its abundant features, implementing OPC UA in low resource devices constitute two challenges - 1) large protocol stack footprint and 2) large network footprint. In our previous work, we have addressed the first challenge [16]. In this paper, we propose our approach to minimize the OPC UA network footprint that it fits in a single BLE frame.

Due to this reduced network footprint, the proposed optimization may also be used e.g. in a CAN bus which has a maximum payload size of only 8 bytes. The CAN bus is still a prominent transport technology in automotive automation and Oil industry [5], [6].

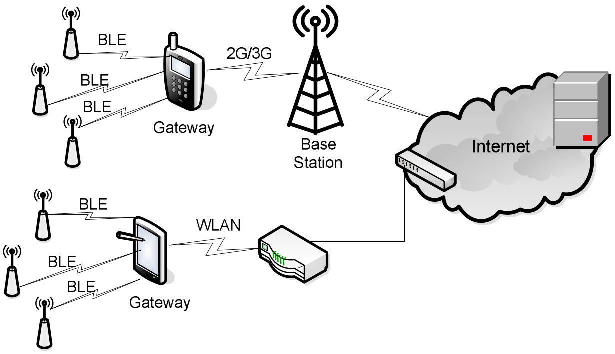

Figure 1: Typical scenario of BLE devices in IOT [2]

The rest of the paper is outlined as follows: section II presents the background and the recent works. The proposed architecture in section III derives an optimized OPC UA header from its current header and presents two use cases to bring BLE devices into IOT. Section IV presents our implementation approach and section V finally concludes the paper.

---

[1] BLE incorporated wrist watches are already available [4]

978-1-4799-0864-6/13/$31.00 ©2013 IEEE

2. State of the art

The classic BT was originally designed for continuous, streaming data applications including voice while the BLE was aimed for transmitting small amount of data periodically. The BT 4.0 device can be a single mode device or a dual mode device. The single mode device only implements low energy feature of the specification and is not backward compatible. The dual mode device implements both BLE and classic BT features and is backward compatible. The comparison of some of the distinct features of classic BT and BLE is depicted in Table 1.

As seen in the table, some classic BT features are excluded or optimized and some features are added in BLE to reduce power consumption. Using the new advertising feature a BLE slave can also initiate communication and indicate that it has some data to transmit to other devices. This enables the BLE device to be in sleep mode and save power. The BLE devices are powered by a tiny coin cell battery which can be used for 5-10 years.

The theoretical results in [9] show the life time of BLE device to be up to 14.1 years, number of simultaneous slaves per master to be up to 5917 and minimum latency for sensor reading as 676 us. A solution for continuous data transmission, i.e. heart beat from ECG, using BLE is presented in [11] which use the pre-processing mechanism YOAPY presented in [12]. The solution is able to compress the data needed to transmit one heartbeat, i.e. approximately 250 to 300 bytes, to a mere 12 bytes that can be transmitted in a single packet.

The 6LoWPAN working group of IETF is working towards bringing low power devices, like sensors and actuators, into the IP network so that these devices would be accessible directly via Internet. The working group has already released a draft of the standardization activity for transmission of IPv6 packets over BLE [3].

In [13], a comparative study of BLE and Zigbee energy consumption is performed by measuring the energy consumption behaviour of real devices. It also models the energy consumption overhead to implement IPv6-based communication over BLE. The measured and the modelled results show that BLE is more energy efficient than Zigbee and also the IPv6 communication over BLE had acceptable energy consumption overhead.

The paper [2] claims of implementing the first prototype system to transmit IPv6 packets over BLE based on Bluez. The IPv6 header size is bigger than the payload size of a single BLE packet. So, in [2] IPv6 header is compressed using the compression scheme according to the standard RFC6282 [14].

Bringing resource limited devices into IP network is not sufficient to enable a true IOT concept. For this, we need a common service interface that not only can be fitted into such resources but also provide interoperability and means of an information modelling. OPC UA [15] is a middleware technology that is envisioned as an enabler of IOT but it is overwhelmed with a lot of features that makes them bulky. To overcome this problem, we have implemented a scaled down OPC UA server based on the "Nano Embedded Device Server profiles" of the OPC Foundation [15]. The implemented server is, to the best of our knowledge, one of the smallest OPC UA servers [16].

Table 1: Comparison of classic BT and Bluetooth Low Energy [1],[7],[8],[9],[10]

Feature	Classic BT	BLE
Design Aim	Designed for continuous , streaming data application	Designed for periodic transfer of small amount of data
Protocols	9 (RFCOMM, OBEX , HID, BNEP, AVCTP, AVDTP, SDP, HCRP, TCSBIN, MCAP)	1 (Attribute)
Communication	Isochronous (connection oriented)	Asynchronous (connection less)
Mode of operation	4 modes – Active, Hold, Sniff and Parked	2 modes – Active and Sleep
Topology	Scatternet	Star-Bus
No of active slaves	7	Unlimited*
Max Packet size	1021 bytes	27 bytes
Time to discover + Connect	Inquiry + Page scan 22.5 ms	Advertising 1.25 ms
Latency to send data (from a non-connected state)	~100ms	~6ms
Connection interval	625 µs , (625*3)µs and (625*5)µs	7.5 ms to 4s but in multiple of 1.25ms
Power consumption	Max ~25mA	Max -~15mA Sleep -~1µA
Power consumption ratio	1 as the reference	0.01 to 0.5 (use case dependent)

*Depends on implementation and available memory

Our approach in [16] was implemented in TPS-1 (ARM9@100MHz) with Real-time Ethernet interfaces but the challenge lies when it comes to devices with different physical transport technologies (e.g. BLE, CAN bus) because of their low payload size. Using OPC UA frames in its original structure will require fragmentation at the sender's side and reassembly at the receiver's side. This will lead to higher latency and more frequent communication which will result in high power consumption.

In this paper, we propose an approach to optimize OPC UA information model in order to have a small network footprint appropriate for both BLE and CAN bus.

3. Proposed Architecture

Our proposed architecture enhances the typical scenario of BLE devices in IOT shown in Figure 1 by providing a common service interface between the BLE device and rest of the world. The use of common service interface enables the vendor and technology independent communication. We propose an approach to optimize OPC UA header to minimize the network footprint. While optimizing only the basic necessary information are kept and compressed with our proposed encoding scheme. The redundant and unnecessary information are excluded. This compressed and excluded information will be restored at the gateway to form the current OPC UA frame. The Table 2 shows the original and proposed optimized size of OPC UA fields.

Table 2: Original and proposed optimized size of OPC UA fields of ReadResponse

UA Fields	Actual Size (byte)	Proposed Optimized Size	Remarks
Message Type	3	3 bit	6 type of message and can be uniquely represented by 3 bit
Chunk Type	1	Excluded	Handled by Gateway
Message Size	4	5 bit	Message size is always less than 31 bytes
SecureChannelId	4	Excluded	Handled by Gateway
Security Token Id	4	Excluded	Handled by Gateway
Security Sequence Number	4	Excluded	Handled by Gateway
Security RequestId	4	Excluded	Handled by Gateway
Message : Encodeable object			
TypeId : ExpandedNodeId			
Encoding Byte	1	Excluded	Always Numeric
Namespace	1	1 byte	No change
Identifier	2	1 byte	Supports only up to 255 nodes and services
ReadResponse			
ResponseHeader : ResponseHeader			
Timestamp	8	Excluded	Will be added in the gateway. Difference of just 6ms
RequestHandle	4	2 bit	Reserved for multiple reads
ServiceResult	4	1 bit	Only says if the result is good or bad
ServiceDiagnostics : DiagnosticInfo			
EncodingMask	1	1 bit	Tells if the diagnostic info is available or not
StringTable[]	4	4 bit	Vendor specific diagnostic information
AdditionalHeader : ExtensionObject			
TypeId : ExpandedNodeId			
Encoding	1	Excluded	Not used. Reserved for future use.
Identifier	1	Excluded	Not used. Reserved for future use.
EncodingMask	1	Excluded	Not used. Reserved for future use.
Total	**52 bytes**	**4 bytes**	
ArraySize	4	Excluded	Array size will be fixed to 2 (NodeId and the actual value).
[0]:DateValue . [15]:DataValue	Depend on read data	Max. data size of 10 bytes	The data headers take 3 bytes leaving maximum payload size of 7 bytes. Please refer to Table 3 for details.
Array of DiagnosticInfos			
ArraySize	4	Excluded	Handled by Gateway

Since the optimization is focused for low resource devices, some limitations are introduced for simplicity. We are planning to implement "Nano Embedded Device Server profiles" of the OPC foundation. For this paper we focused on "ReadResponse" operation just as an example but we will consider all mandatory services specified in the Nano profile because we want to propose this optimization as next generation of OPC UA standard.

We present two use cases to enable common service interface between the low energy devices. First implements Optimized OPC UA Transport Profile (OOTP) along with a TCP/IP protocol stack while the second implements a layer 2 transport of OPC UA data.

3.1. Use Case 1: Integrating TCP/IP (IPv6) protocol stack and OOTP in BLE devices

For TCP/IP we focus in IPv6 because it is the next generation of IP protocols. Figure 2 shows the IPv6 header and OOTP included in the BLE payload. The IPv6 header is compressed from 48 bytes to 5 bytes [2].

Table 3: OPC UA DataValue array of ReadResponse

UA Fields	Actual Size (byte)	Proposed Optimized Size	Remarks
[0] : DataValue			
Encoding	1	1 bit	Tells if the value is good or bad
Value: variant			
Variant Type : <variant type>	1	Excluded.	Support only numeric NodeId
Value : NodeId			
NodeIdEncodingMask:<value>	1	Excluded.	Support only numeric NodeId
NodeId NamespaceId:<value>	2	7 bits	Support only 127 namespaces
String: <NodeId>	Depend on NodeId	1 byte	Support only numeric NodeId and maximum of 255 nodes
ServerTimestamp : <time>	8	Excluded	Will be added by gateway. 6ms overhead
[1] : DataValue			
Encoding	1	1 bit	Tells if the value is good or bad
Value: variant			
Variant Type : <variant type>	1	7 bit	Support 127 data types
<variant type> : <value>	Depend on variant type	7 bytes	Maximum payload of 7 bytes. If variant is string of higher length then fragmentation should be done.
SourceTimestamp : <time>	8	Excluded	Will be stored in the gateway
ServerTimestamp : <time>	8	Excluded	Will be added by gateway.6ms overhead

The OPC UA optimized header uses 7 bytes leaving 7 bytes for payload, which is sufficient to hold data produced by low energy device because of simplicity of application's information model. If the variant type is string whose payload size is higher than 7 bytes, then fragmentation should be performed. The BLE device will implement OOTP on top of IPv6 protocol stack. A gateway device, which could be any dual mode BT 4.0 device, will reconstruct the current OPC UA header from the optimized one. Similar to dual mode BT device, which supports both classic BT and BLE, our OPC UA gateway will support both the optimized OPC UA header and current OPC UA header.

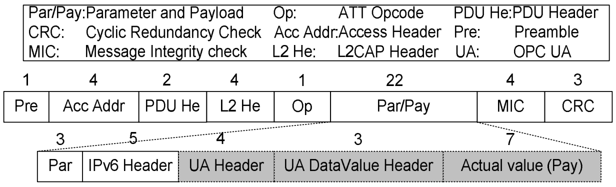

Figure 2: IPv6 header and OOTP fitted in BLE payload

3.2. Use Case 2: Integrating OOTP in BLE devices without TCP/IP protocol stack

Having IP for such a small device might not be promising approach for some applications. The applications can benefit from the increased payload size of 5 bytes resulting in total OPC UA payload size of 12 bytes when the IP stack is excluded. In this use case, BLE device will implement only the OOTP. The IP functionality can be easily provided at the gateway and the BLE device becomes part of IOT through the gateway.

Figure 3 shows the OOTP included in the BLE payload without the IPv6 header. This use case gives the opportunity of using the OOTP in CAN bus which has the maximum payload size of just 8 bytes. The current optimization leaves only 1 byte for CAN bus payload. But UA fields like reserved bit for request handle, service result quality, and vendor specific information of “ResponseHeader” as explained in Table 2 can be excluded to have the payload size of 2 bytes for CAN bus which could be enough for certain applications.

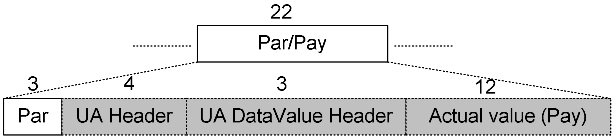

Figure 3: OOTP in BLE payload

Being a corporate member, we are working closely with OPC foundation and have presented our vision of enabling OPC UA based common service interface for low resource devices [17]. As a part of it, we presented our work related to bringing down the OPC UA into the chip level [16]. We are planning to propose OOTP to the OPC foundation for low resource devices.

4. Testbed Implementation

Our implementation setup, shown in Figure 4, uses cB-OLP425 BLE device from connectBlue and iPhone 4S from Apple. The OOTP with IPv6 protocol stack will be implemented in the cB-OLP425 as done in [2] and the OPC-UA gateway function is implemented in the iPhone.

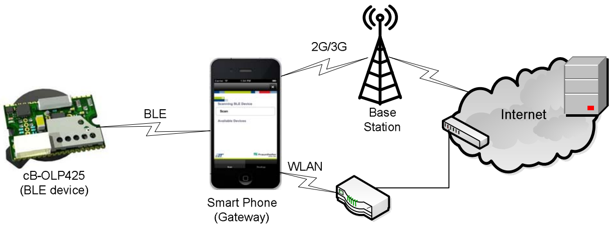

Figure 4: Implementation setup

Both use cases mentioned in section 3 uses the same OOTP. The only difference is the inclusion and exclusion of IP stack. In Use case 1, the BLE device has its own IP address and is directly connected to the Internet. The OPC UA gateway device just reconstructs the optimized header to the current one. In case 2, the BLE device does not have any IP. The gateway device in this case works both as an OPC UA gateway and interface to the Internet.

We have developed an iPhone App that can establish BLE connection with the cB-OLP425. We are currently working on the implementation of use case 2 as it is our core work. We eventually have to port the full OPC UA stack into the iOS to enable effective gateway functionality. There already exists a Java OPC UA stack for Android [18]. The inclusion of IP stack is a solved issue [2]. We will consider the IP stack after the completion of OOTP integration on top of BLE stack. We are planning to validate our implementation in a modular smart factory. We think that OPC UA has a potential to enable self-configurable industrial systems where inclusion of new sensors should trigger new services.

5. Conclusion

The work presented in this paper has demonstrated that the OPC UA network footprint can be scaled down to fit into different devices and transport technologies with small frame size. The optimized OOTP needs 7 bytes for the header information. In case of BLE, building OOTP on top of IP stack leaves 7 bytes for payload and only having OOTP in the BLE stack leaves 12 bytes for the payload. The proposed OOTP can be further optimized to use in CAN bus leaving 2 bytes for payload which might be sufficient for certain applications. We are currently working on the OOTP implementation and are planning to suggest it as a next generation of OPC UA transport profile to the OPC foundation for low resource devices.

Acknowledgment

This work was partly funded by EU FP7 STREP project IOT@Work under the grant number ICT-257367 as well as the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster “Intelligent Technical Systems OstWestfalenLippe” (it’s OWL).

References

[1] Bluetooth SIG. Bluetooth specification 4.0. June 2010.

[2] H. Wang, M. Xi, J. Liu and C. Chen, Transmitting IPv6 Packets over Bluetooth Low Energy based on Bluez. *ICACT*, Jan 2013.

[3] J. Nieminen et. al, Draft-ietf-6Lowpan-btle-08 Transmission of IPv6 Packets over Bluetooth Low Energy. June 2012.

[4] http://world.g-shock.com/us/en/ble/, visited 17.05.2013.

[5] J.H. Park, M.H. Kim, S. Lee and K.C. Lee, Implementation of a CAN System with Dual Communication Channel to Enhance the Network Capacity. *IEEE IECON 2010*, pp. 3135-3140, 2010.

[6] Z. Qi-zhi, H. Yu-yao, L Lin and S. Xiao-hui, The Design of PROFIBUS-DP and CAN Bus Communications Unit in Drilling Rig Control System. *IEEE ICCET*, vol. 3, pp.492-494, 2010.

[7] http://www.imd.uni-rostock.de/ma/gol/lectures/embedded/Literatur/Low Energy Training.pdf. visited 17.05.2013.

[8] http://www.bluegiga.com/files/bluegiga/LEPublic/LowEnergyHowItWorks.pdf. May 2011, visited 17.05.2013.

[9] C. Gomez, J. Oller and J. Paradells, Overview and Evaluation of Bluetooth Low Energy-An Emerging Low-Power Wireless Technology. *Sensors 2012*, no.9, pp.11734-11753, 2012.

[10] R. Heydon, Bluetooth Technology – Basics and Brand. *Bluetooth World*, Shanghai, China 2013.

[11] A.J. Jara et. al, Evaluation of Bluetooth Low Energy capabilities for continuous data transmission from a wearable electrocardiogram. *IEEE IMIS*, pp.912-917, July 2012.

[12] A.J. Jara et. al, Evaluation of 6LoWPAN Capabilities for Secure Integration of Sensors for Continuous Vital Monitoring. *V International Symposium on Ubiquitous Computing and Ambient Intelligence* (*UCAmI'11)*, 2011.

[13] M. Siekkinen et. al, How Low Energy is Bluetooth Low Energy. *IEEE WCNCW 2012*, pp.232-237, April 2012.

[14] J. Hui, Ed. And P. Thubert, RFC6282 Compression format for IPv6 datagrams over IEEE 802.15.4-based networks. Sep. 2011.

[15] OPC Foundation. OPC Unified Architecture. http://www.opcfoundation.org/.

[16] J. Imtiaz and J. Jasperneite, Scalability of OPC-UA Down to the Chip Level Enables Internet of Things. *IEEE INDIN 2013*, Bochum, July 2013.

[17] J. Jasperneite and J. Imtiaz, OPC UA as an Enabler for Internet of Things. OPC Day Europe, 2013.

[18] Prosys OPC UA Android Client, http://www.prosysopc.com/opc-ua-android-client.php/, visited 02.07.2013.