2015 3rd International Conference on Future Internet of Things and Cloud

BLEmesh: A Wireless Mesh Network Protocol for Bluetooth Low Energy Devices

Hyun-soo Kim, JungYub Lee, Ju Wook Jang
Department of Electronic Engineering
Sogang University
Seoul, South Korea
e-mail: {vbiqve, yub0309, jjang}@sogang.ac.kr

***Abstract*—Bluetooth Low Energy is a highly adopted and anticipated communication solution for the Internet of Things. However, the restriction of data communication topology to point-to-point, limited range of communications, and the lack of IP support make Bluetooth Low Energy less attractive for Internet of Things applications. In Dec. 2014, the Bluetooth SIG standardized the Internet Protocol Support Profile to support the exchanging of IPv6 packets between devices over Bluetooth Low Energy. Still, a mesh networking protocol for multi-hop support is needed for it to overcome the limitations due to short range and restricted topology. This paper introduces BLEmesh, a wireless mesh network protocol which utilizes the broadcasting capability of wireless transmissions. We identify the available data payload using Bluetooth Low Energy Generic Access Profile – Non-connectable Advertisement Data for different number of nodes and packets to send in a batch. Then we compare BLEmesh with conventional routing method and flood routing method. Our preliminary evaluation results show that the number of transmissions by BLEmesh is significantly smaller compared to its competitors for some selected network configurations, reducing the aggregated energy consumption within the mesh network.**

Keywords-wireless mesh; opportunistic routing; Bluetooth Low Energy; Internet of Things; energy efficiency

I. INTRODUCTION

The Internet of Things (IoT) is a rapidly developing area in modern technology. Extending the scope from traditional network devices (PCs) to all kind of *things* (non-PC devices) requires a different communication solution. As an example, for wireless sensors opperated by coin cell batteries to send small sizes of data to the internet, it is critical that it satisfies certain level of energy efficiency. Without it, the sensors will fail after a small period of time. Bluetooth Low Energy (BLE) [1], a.k.a. Bluetooth Smart, is one of the latest developments in this area, which supports low power communications guaranteeing sensors to operate even up to two years, in spite of relatively short range of communications.

Another major communication solution for IoT devices such as Wireless Sensor Networks (WSNs) is 802.15.4, standardized by the IEEE. 802.15.4 offers PHY and MAC layers for low-cost, low-speed, low-power Wireless Personal Area Networks (WPANs). 802.15.4 supports star and peer-to-peer topologies, which can be extended to mesh network [2]. Also, the IETF has standardized mechanisms for encapsulation and header compression that allow IPv6 packets to be sent and received over 802.15.4 [3].

Although BLE is an already highly adopted and anticipated communication protocol within the IoT ecosystem, due to the rapid spread of smartphones [4]. It has some drawbacks such as restriction of data communication topology to point-to-point, i.e. cannot communicate to multiple nodes simultaneously, limited range of communications to 20-50m, and the lack of IP support for true IoT applications. 802.15.4 is clearly the winner from this perspective, since it does not have limitations in network topology and limited range of communications by supporting mesh topology from lower layers, and has integrated IP support through 6LoWPAN [3]. In conclusion, for BLE to be fully suitable for IoT applications, specifications for exchanging IP packets and multi-hop routing through mesh networking are a necessary.

In Dec. 2014, the Bluetooth SIG adopted the Internet Protocol Support Profile (IPSP), which supports the exchanging of IPv6 packets between devices over BLE [5]. In conjunction to the Bluetooth SIG, the IETF is almost at the standardization of transmission of IPv6 packets over BLE, using the similar methods from 6LoWPAN [6]. More recently, the Bluetooth SIG announced the formation of Smart Mesh WG, clearly showing its plans to adopt mesh networking to BLE [7].

This paper presents BLEmesh, a wireless mesh network protocol which utilizes the broadcasting capability of wireless transmissions. We identify the available data payload using BLE Generic Access Profile (GAP) – Non-connectable Advertisement Data, for different number of nodes participating in BLEmesh and the number of packets to send in a batch. Our preliminary evaluation results for the comparison between the proposed model, conventional routing method, and flood routing method show that the number of transmissions is significantly smaller compared to its competitors for some selected network configurations.

In section II, we discuss related works that implement wireless mesh protocols to BLE, such as, CSRmesh and Smart Mesh WG. Also, we introduce opportunistic routing, which is the foundation for BLEmesh. Section III highlights the key differences between conventional routing, flood routing, and BLEmesh. Section IV specifies the header fields of BLEmesh in detail. A preliminary evaluation through simulation is given in Section V. And lastly, Section VI gives a conclusion and suggests future works.

978-1-4673-8103-1/15 $31.00 © 2015 IEEE
DOI 10.1109/FiCloud.2015.21

II. RELATED WORKS

A. Flood Routing

In Feb. 2014, Cambridge Silicon Radio (CSR) released CSRmesh, a flood mesh protocol that uses BLE's non-connectable advertisements to send data to individual devices, groups, sub-groups or all devices. CSRmesh benefits from the simple mechanisms of flood mesh, scalable up to 64,000 devices or groups per network. In addition, due to the fact that flood mesh does not need to maintain a routing table, CSRmesh's setup time is close to zero. However, to support mesh topology in BLE, CSRmesh chose a multilayered approach that builds on top of BLE, which reduces the size of data payload per individual broadcasts [8] [9].

Unfortunately CSRmesh is not an open protocol, aside from the key concepts of CSRmesh such as mesh network protocol by using flood routing method, direct comparison between the proposed BLEmesh and CSRmesh would be inappropriate. In this paper, we assume that a flood routing protocol for unicast packet transmission with data payload 8 bytes is given for evaluation and comparison purposes.

B. Smart Mesh Working Group

In Feb. 2015, the Bluetooth SIG officially announced the formation of the Bluetooth Smart Mesh WG. Which will build the architecture for standardized mesh networking capability for Bluetooth Smart technology. Since the Smart Mesh WG is a group that just has started, no such specifications or drafts have yet been published by the WG. The WG expects to have the specification ready for prototype testing later in 2015, and the SIG will look to officially adopt profiles in 2016 [10].

C. Opportunistic Routing

Opportunistic routing methods, such as ExOR [11] and SOAR [12], exploits the characteristics of the wireless transport medium to gain overall performance over conventional single-path routing. Instead of computing a unicast path through the intermediate nodes before the transmission, it broadcasts the packets and selects the next forwarding nodes among those who successfully received the packet in an opportunistic manner. ExOR wins over conventional single-path routing significantly in terms of throughput. Whereas SOAR advanced ExOR by supporting multiple simultaneous flows, which is an important factor for the purpose of using it in real networks.

The main design challenge of an opportunistic routing is the development of an agreement protocol between the nodes to choose the appropriate forwarder to broadcast the packet to the higher priority, i.e. closer to the destination, nodes. This is done though the creation of forwarders list from the source node, aggregation of packets into batches, and operating batch-wise by using the batch map.

Several literatures such as [13] [14] [15] have implemented opportunistic routing in WSNs, characterized by low duty-cycle, unreliable links. However, there were no attempts to realize the opportunistic routing protocol stack on top of BLE.

III. BASIC IDEA

Let us consider the given abstract network topology depicted in Fig. 1. Also, for simplicity, we assume that the size of the data payload is the same for each method. Since the transmission probabilities between the intermediate nodes 1, 2, 3 and the destination node are 1.0, any packet that reaches nodes 1, 2, 3 from the source will be transmitted to the destination. Now, let us focus on the total number of transmissions or broadcasts needed to send a unicast packet from the source to the destination. The comparisons between the proposed method, BLEmesh, flood routing method, and conventional routing method are as follows.

A. Proposed method, BLEmesh

BLEmesh benefits from opportunistic routing. That being said, if any of the intermediate nodes receives the packet sent by the source, it will suffice. The probability of at least one node receiving the packet is 0.79, thus, approximately 1.27 broadcasts will be needed for the packet to reach any of the intermediate nodes. Adding the additional broadcast performed by the intermediate nodes to send the packet to the destination node, 2.27 broadcasts in total are needed.

B. Flood Routing

Flood routing method rebroadcasts every received packet. Because of this simple mechanism, a packet is possible to be delivered to the destination node even in the absence of a routing functionality. However, when two neighboring intermediate nodes receive the same packet, they will both rebroadcast the identical packet which leads to inefficient use of bandwidth.

In the case of flood routing, the total number of broadcast to send a data from the source to the destination can be calculated by adding the number of estimated broadcasts carried out by each node. Namely, the source node performs 1.27 broadcasts, and the three intermediate nodes perform 1.2 broadcasts adding up to total of 2.47 broadcasts. Compared with the number of broadcasts needed for BLEmesh, flood routing performs more broadcasts than BLEmesh. From this fact, it can be derived that BLEmesh will consume less power, due to lesser amount of total transmission performed within the network.

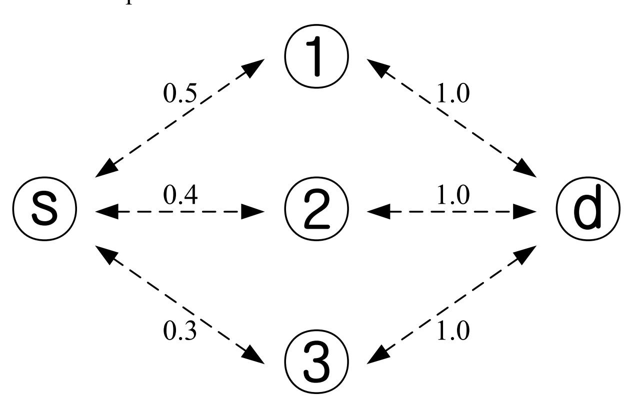

Figure 1. Sample Network Topology.

C. Conventional Routing

Conventional routing method uses a single-path routing protocol by choosing the intermediate nodes to forward the data between the source and the destination nodes. In this given network, node 1 is selected as the forwarding node, due to the highest probability of receiving the packet from the source. Probablitically, 2 transmissions are required between the source and node 1, adding up to total of 3 transmissions for the destination to receive the packet.

In conclusion, the proposed method has the loweset number of transmissions followed by flood routing method and conventional routing method, which implies that opportunistic routing is a more viable solution for implementing mesh network to BLE.

IV. Protocol Specification

Fig. 2. outlines the BLEmesh's packet header format using BLE's Generic Access Profile (GAP) – Non-connectable Advertisement Data, where the core header fields of BLEmesh resides.

The *Data Length* and *GAP AD (Advertised Data) Type* field is specified by the Bluetooth SIG to indicate the length of the advertised data and its type. The advertised data type values are assigned by the Bluetooth SIG and used in GAP for inquiry response, EIR data type values, manufacturer-specific data, advertising data, BLE UUIDs and appearance characteristics, and class of device. For instance, a proprietary manufacture-specific advertisement packet that utilizes all available payload should have the Data Length value of 0x1F (hexadecimal value of 31), and AD Type value of 0xFF (manufacturer-specific data). We assume that BLEmesh is given an unassigned AD type value such as 0xC8, thus making it possible for a node to understand its encapsulated data format, i.e. BLEmesh.

The *BLEmesh Header Length* field indicates the size of the BLEmesh header. Since the total size of the BLEmesh header, excluding the *Data Length* and *GAP AD Type* field of 2 bytes, will never exceed 29 bytes (232 bits), this can be represented by a field size of 1 byte (8 bits). The *Batch ID* field indicates which batch the received packet belongs to and it occupies another 1 byte.

The *Packet Number* field indicates the offset of the received packet based on the batch, the *Batch Size* field indicates the total number of packets in a batch, the *Fragment Number* field indicates the offset of the received packet based on the fragment, and the *Fragment Size* field indicates the total number of packets in a batch. Because of the fact that intermediate nodes will not likely receive all of the packets in a batch, the fragment related fields are a necessary. In addition, without the *Fragment Number* and the *Fragment Size* fields, an intermediate node will wait much longer assuming that the higher priority nodes haven't yet finished broadcasting their packets. This results in lower throughput, higher latency, etc.

The sizes of *Packet Number*, *Batch Size*, *Fragment Number*, and *Fragment Size* fields are identical. Whilst the reason for the equality in sizes between [*Packet Number*, *Batch Size*] and [*Fragment Size*, *Fragment Number*] are

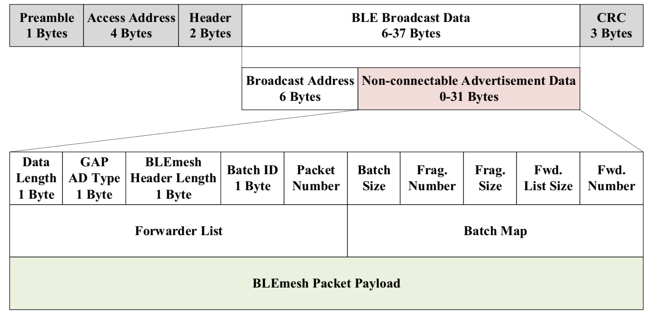

Figure 2. BLE packet structure and BLEmesh packet header format.

trivial, some might argue that the *Fragment Size* field should be smaller compared with the *Batch Size* field. However, noticing that the source node of the BLEmesh network must also contain the fragment related fields in its header, it is fairly straightforward that the two must be equal in their sizes. Given n number of packets in a batch, the size of the four fields can be computed by, $[log_2n]$ bits. Therefore, the aggregated size of the four fields is $4\cdot[log_2n]$ bits.

The *Forwarder List Size* field specifies the number of nodes participating in BLEmesh, and the *Forwarder Number* field indicates the sender's offset within the forwarder list. The size of the *Forwarder List Size* and the *Forwarder Number* fields are identical. Given k number of participating nodes, the size of the both fields can be computed by, $[log_2k]$ bits.

The *Forwarder List* is a list of participating nodes sorted by priority. Namely, the highest priority is given to the destination node and is on the top of the list, whilst the source node has the lowest priority and is the last entity on the list. The size of the *Forwarder List* is derived from the number of participating nodes. Given k from above, the size of the *Forwarder list* is $k\cdot[log_2k]$. The *Forwarder list* is formulated by the source node, where it periodically floods and collects the transmission probability of every other links between two nodes.

The last portion of the header is the *Batch Map*, which indicates for each packet in a batch the highest priority node to have received a copy of that packet. When a node receives a packet from another node, it inspects the *Batch Map* field sent within the packet and decides whether to broadcast that packet or not. For any packet entry on the *Batch Map*, if the receiving node's priority is higher than the listed node, the receiving node enlists itself as the highest node on the *Batch Map* and broadcasts the corresponding packet later on. On the other hand, if the receiving node's priority is lower than the listed node, or the recipient's former *Batch Map* prior to receiving a new one had a much higher priority node on the list, the recipient will not broadcast the corresponding packet and rather discard it. The size of the *Batch Map* field is related to the number of packets in a batch, and the number of participating nodes. Given n and k, the size of the *Batch Map* is $k\cdot[log_2n]$.

The *BLEmesh Packet Payload* follows the BLEmesh header described thoroughly from above. Clearly, the size of the actual data payload for BLEmesh is not fixed, and it depends heavily on the number of packets in a batch n, and the number of participating nodes k. It's also worth mentioning that the "BLEmesh Payload Length" field is unnecessary as it can be inferred directly from the *Data Length* and the *BLEmesh Header Length* fields.

From the fixed BLEmesh fields sizes [*Data Length*, *GAP AD Type*, *BLEmesh Header Length*, *Batch ID*] and the fields where sizes vary by n and k. The actual size of BLEmesh data payload can be computed as follows.

$$216 - \left(4\cdot\lceil log_2 n\rceil + 2\cdot\lceil log_2 k\rceil + k\cdot\lceil log_2 k\rceil + k\cdot\lceil log_2 n\rceil \right) \qquad (1)$$

The result is the size of the data payload in [bits] where 216 was derived from subtracting the sum of fixed BLEmesh fields, 4 bytes, from the BLE advertised data payload 31 bytes. Fig. 3. illustrates the size of the data payload for any given number of packets in a batch, and participating nodes. For example, if there are 10 participating nodes and a total of 178 bytes of data to be sent, the batch will consist of 22 packets with BLEmesh data payload of 8 bytes each. Similarly, if there are 20 participating nodes and a total of 56 bytes of data to be sent, the batch will consist of 7 packets with BLEmesh data payload of 8 bytes each.

V. Evaluation

This section presents preliminary evaluation results through simulations which show that BLEmesh increases overall energy efficiency of a network when compared with conventional routing method and flood routing method. The network topology used for the evaluation is shown in Fig. 4. (a) The network consists of 5 nodes forming a chain from the source node (N0) to the destination node (N4), where the link delivery probability for one hop, two hops, and three hops between the nodes are defined as 90%, 50%, and 10%.

In all three routing methods used for this evaluation, the total amount of unicast data sent from N0 to N4 is 96 bytes. It should be noted that, due to the difference in their size of data payload, the total number of fragmented *packets* sent from N0 to N4 differs for each of the methods. To elaborate, conventional routing method can deliver up to 29 bytes per packet, assuming that *Data Length* and *GAP AD Type* headers are excluded from the original BLE non-connectable advertisement. On the other hand, flood routing method is assumed to have 8 bytes of payload as aforementioned, and finally, BLEmesh carries 171 bits of data per packet in this network setup. Therefore, as presented in Table I., 4 packets are sent by conventional routing, 12 packets by flood routing, and 5 packets by BLEmesh.

TABLE I. Comparison of Number of Packets for Each Methods

Total amount of unicast data sent from N0 to N4 = 96bytes			
	Conventional routing	**Flood routing**	**Proposed method**
Payload size	29 bytes	8 bytes	171 bits (≈ 21 bytes)
Number of packets	4 packets	12 packets	5 packets

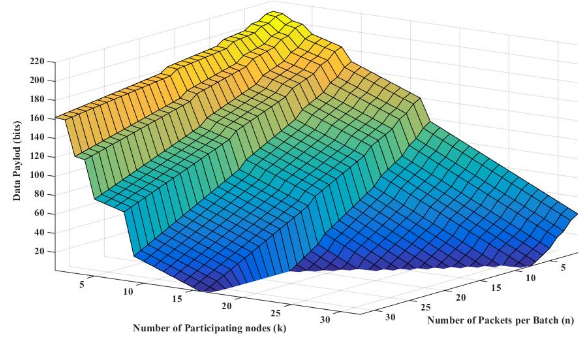

Figure 3. BLEmesh data payload (bits) for any given number of packets per batch (n) and the number of participating nodes (k).

First, in conventional routing, the nodes to participate in the forwarding of the data are selected. Here, the routing path has been selected by N0-N2-N4 through ETX calculations. Thus, N1 and N3 do not participate in the routing, even though they both have the possibility of receiving the packets sent in between the participating nodes. The source node (N0) first transmits the 4 packets (P1-P4) to N2. Each packet has 50% chance of being received by N2, and similarly, each acknowledgment to N0 for each received packet from N2 also has 50% chance. Mathematically, a packet should be sent 4 times for the sender to receive the acknowledgment and move on to the subsequent packet.

Fig. 4. (b) illustrates the first several transmissions of conventional routing simulation. For sending the first packet (P1), N2 received the packet in the first attempt and its acknowledgment vice versa. However, in the case of P2, the acknowledgment from N2 failed to reach N0 in the first attempt, resulting N0 to resend P2 to N2. At termination, a total of 26 transmissions were made by N0, N2, and N4.

Next, using the flood routing method is shown in Fig. 4. (c). Since individual packets are independently broadcasted, summing up the number of broadcasts for a packet based on probability, and then multiplying it by 12 packets can give us the total number of broadcasts mathematically.

In our simulation of flood routing method presented in Fig. 4. (c), P1 was successfully delivered to N1 and N2 but not N3. In flood routing, each node that has received the broadcasted packet rebroadcasts it to the network without concerns for other nodes that might have received the identical packet, i.e. *flooding*. Therefore a rebroadcast of P1 was made by N1 and N2, the former was received by N0 and N1 whilst the latter was received by N1 and N3 but not N4. Having been informed that its broadcast for P1 has been received by other nodes in the network, i.e. N0 from N1 and N1 from N2 respectively, N0 and N1 will not rebroadcast P1 again during this phase. Except for the case where the source (N0) does not hear an acknowledgment from the destination (N4) until the defined *timeout* period.

The arrival of all 12 packets to the destination and their corresponding acknowledgments to the source were finished after 96 broadcasts in the network.

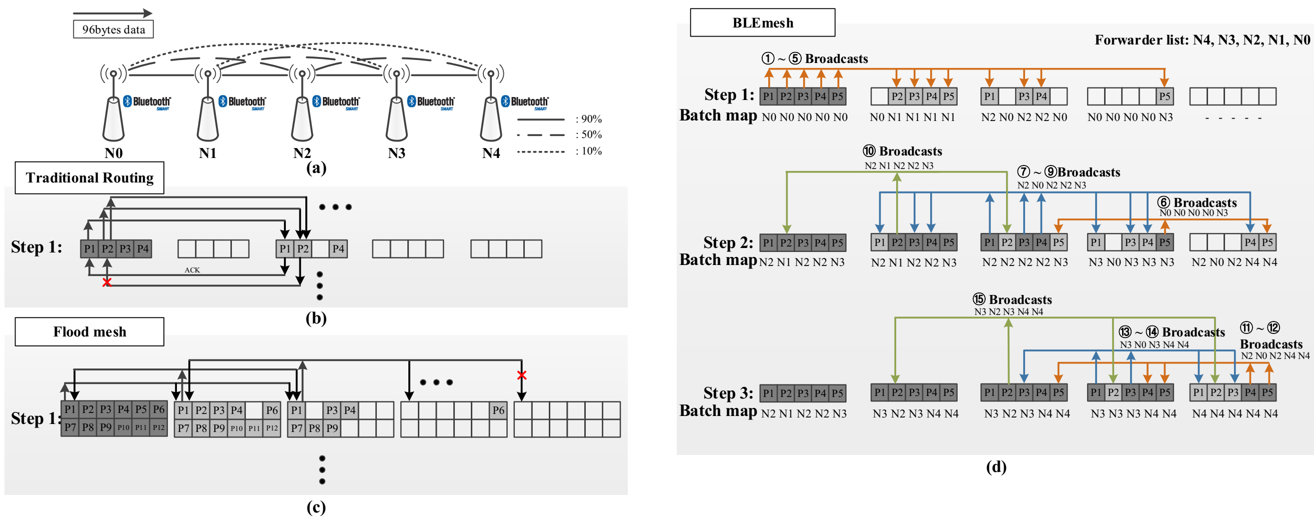

Figure 4. BLEmesh evaluation and comparison with other methods: (a) Network topology; (b) First few transmissions for Conventional routing; (c) First few broadcasts for Flood routing; (d) Step-by-step visualization for BLEmesh broadcasts.

Finally, an instance of BLEmesh simulation is fully presented in Fig. 4. (d). In Step 1, a batch which consists of 5 packets (P1-P5) was broadcasted by N0. And at least one of the packets was successfully received by N1, N2, and N3. Upon receiving a BLEmesh packet, which contains the *Forwarder List* and the *Batch Map*, a node becomes fully aware of the entire batch, its priority, acknowledgments from higher priority nodes, etc.

In Step 2, N3 broadcasts its fragment of the batch, i.e. P5, to the network with the modified batch map [N0 N0 N0 N0 N3]. The broadcast was received by N2 and N4. The former, N2, modifies its batch map from [N2 N0 N2 N2 N0] to [N2 N0 N2 N2 N3], and broadcasts P1, P3, and P4, which were received by (N1, N3), (N1, N3), and (N1, N3, N4) respectively. The latter, N4, has currently received P5 from N3 and P4 from N2, changes its batch map and awaits for its broadcast sequence depicted in Step 3. The last broadcast during Step 2 is made by N1, who has received at least 1 packet from N2, alters its batch map to [N2 N1 N2 N2 N3] and then broadcasts P2, the only packet in the batch not to have been acknowledged by higher priority nodes. The broadcast of P2 was received by N0 and N2. As a result, N0 does not broadcast any packets, since every packets in the batch have been acknowledged by higher priority nodes.

Last, in Step 3, the destination node (N4) broadcasts its batch map (not the packets) which is [N2 N0 N2 N4 N4]. Next, N3 broadcasts P1 and P3 with batch map [N3 N0 N3 N4 N4], and so forth. At the 15th broadcast of our presented simulation, the destination has successfully received all 5 packets of the batch and modified the batch map to [N4 N4 N4 N4 N4]. The following broadcast (16th) by N4, not illustrated on Fig. 4. (d), was received by N2 and N3, terminating the BLEmesh.

In comparison, the total numbers of transmissions or broadcasts made by each of the methods are 12, 96, and 16 as drawn on Fig. 5.

As in [11], the opportunistic routing method ExOR, BLEmesh clearly wins over conventional routing method, even though it sent more number of packets then its competitor, 4 for conventional routing, 5 for BLEmesh. For flood routing method, the assumed data payload of 8 bytes per packet became its bottleneck resulting in 96 broadcasts. A modified simulation for flood routing method, altering the payload size to match the number of packets as in BLEmesh, showed a total of 33 broadcasts. Both conventional routing and flood routing have noticeable number of transmissions compared with BLEmesh.

In BLE, based on TI's CC2540 SOC, the energy consumption for TX phase is noticeably higher when compared with RX phase or other phases [16]. The more the number of transmissions (broadcasts) needed for data delivery, the more energy will be consumed within the network. Therefore BLEmesh has clear advantage in energy efficiency for mesh networking on BLE.

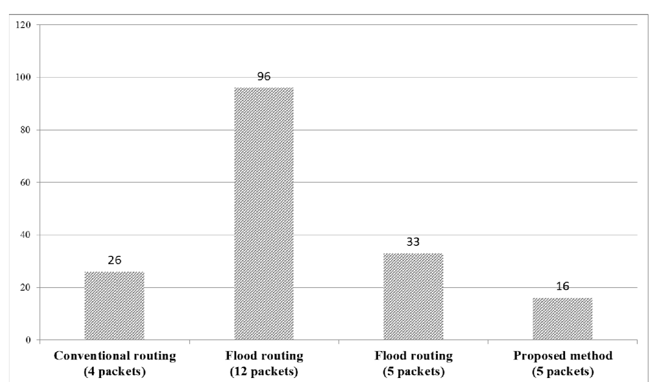

Figure 5. Number of transmissions (broadcasts) for each method.

VI. Conclusion and Future Works

This paper presents BLEmesh, an opportunistic routing based wireless mesh network protocol for BLE. We identified the available data payload using BLE Generic Access Profile (GAP) – Non-connectable Advertisement Data, for different number of nodes participating in BLEmesh and the number of packets to send in a batch.

Our preliminary evaluation results showed that the number of broadcasts in the case of proposed BLEmesh is 54.5% smaller compared with flood routing (modified simulation using 5 packets), and 42.3% smaller compared with conventional routing. Therefore, BLEmesh effectively reduces the aggregated energy consumption within the selected network topology.

One major characteristics of BLE the authors have not taken account for are the three advertisement channels used in BLE. The simulations were done under the assumption of one advertising channel in existence. BLEmesh could merit from the utilization of all three channels, resulting in higher throughput and lower latency. Furthermore, the support for multiple simultaneous flows in opportunistic routing could also enrich BLEmesh.

Acknowledgment

This Work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. B0126-15-1051, A Study on Hyper Connected Self-Organizing Network Infrastructure Technologies for IoT Service)

References

[1] Bluetooth Special Interest Group, “Bluetooth Core Specification Version 4.2,” December, 2014.

[2] IEEE Std 802.15.4-2011, “IEEE standard for local and metropolitan area networks – part 15.4: Low-rate wireless personal are networks (LR-WPANs),” 2011, standards.ieee.org/getieee802/download/802.15.4-2000.pdf.

[3] G. Montenegro, N. Kushalnager, J. Hui, and D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” RFC4944, IETF, Sept. 2007.

[4] A. West, “Smartphone, the key for Bluetooth® low energy technology,” Available at: http://www.bluetooth.com/Pages/Smartphones.aspx.

[5] Bluetooth Special Interest Group, “Bluetooth Internet Protocol Support Profile Specification Version 1.0.0,” December, 2014.

[6] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, “Transmission of IPv6 Packets over BLUETOOTH® Low Energy,” draft-ietf-6lo-btle-11 (Standards Track), IETF, April, 2015.

[7] KH. Chang, “Bluetooth: A Viable Solution for IoT?[Industry Perspectives],” IEEE Wireless Communications, vol. 21, no. 6, Dec. 2014, pp. 6-7.

[8] R. Walker, “CSRmesh FAQs,” Available at: http://www.csr.com/blog/2015/04/csrmesh-faqs-2/.

[9] CSR plc, “CSRmesh wiki,” Available at: http://wiki.csr.com/wiki/CSRmesh.

[10] Bluetooth Special Interest Group, “Bluetooth® Technology adding Mesh Networking to Spur New Wave of Innovation,” Press Release, Feb. 2015, Available at: http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=224.

[11] S. Biswas and R. Morris, “ExOR: Opportunistic Multi-Hop Routing for Wireless Networks,” ACM SIGCOMM Computer Communications Review, vol. 35, no. 4, 2005, pp. 133-144.

[12] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, “SOAR: Simple Opportunistic Adaptive Routing Protocol for Wireless Mesh Networks,” IEEE Transactions on Mobile Computing, vol. 8, no. 12, Dec. 2009, pp. 1622-1635.

[13] S. Guo, L. He, Y. Gu, B, Jiang, and T. He, “Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links,” IEEE Transactions on Computer, vol. 63, no. 11, Nov. 2014, pp. 2787-2802.

[14] E. Ghadimi, O. Landsiedel, P. Soldati, S. Duquennoy, and M. Johansson, “Opportunistic Routing in Low Duty-Cycle Wireless Sensor Networks,” ACM Transactions on Sensor Networks, vol. 10, no. 4, June. 2014.

[15] S-S. Wang and Z-P. Chen, “LCM: A Link-Aware Clustering Mechanism for Energy-Efficient Routing in Wireless Sensor Networks,” IEEE Sensors Journal, vol. 13, no. 2, Feb. 2013, pp. 728-736.

[16] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, “How Low is Bluetooth Low Energy? Compartive Mesasurements with Zigbee/802.15.4,” 2012 IEEE Wireless Communications and Networking Converence Workshops (WCNCW), April. 2012, pp. 232-237.