

Energy and performance costs evaluation for BLE mesh links

Henrique Carvalho Silva¹, Cíntia Borges Margi¹

¹Universidade de São Paulo (USP)
05.508-010 – São Paulo – SP – Brazil

henrique.c.s@usp.br, cintia@usp.br

Abstract. *Bluetooth Low Energy (BLE for short) is among the favorites to become a de facto standard in the context of the Internet of Things (IoT). However, its main challenge is the lack of standards for efficient mesh networking. Furthermore, the literature lacks works analyzing energy consumption trade-offs for BLE mesh networks. We address this issue by experimentally evaluating three minimal topologies for linking separate BLE star networks. We aim to determine a lower boundary in terms of energy and performance costs using the metrics of energy consumption, delivery rate, and goodput. We perform our experiments using a testbed comprised of TI CC1350 nodes running Contiki OS. Our results enable us to estimate similar costs for large scale networks.*

1. Introduction

For a long time, the Internet of Things (IoT) has been a speculation around the idea that everything around us would be connected and thus have its own representation in the World Wide Web [Want et al. 2015]. Bluetooth Low Energy (BLE), announced in 2010 as the low power successor of the classic Bluetooth technology, has been evolving to be now among the favorite alternatives to become the *de facto* standard for wireless personal area networks (WPAN) in the context of IoT [Raza et al. 2016].

BLE is nowadays present in millions of smartphones and is becoming more and more popular in many kinds of applications, from home automation, to healthcare and marketing [Chang 2014]. However, one of the main challenges still faced by this technology in this competition is its limited coverage. Such limitation is due to its low range and its design being originally restricted to star network topologies, thus making mesh networks a long anticipated feature [Hortelano et al. 2017, Raza et al. 2016].

The demand for BLE mesh networks has thus gained attention both from the industry and the academia, which have been providing numerous propositions for BLE mesh networks along the last few years [Darroudi and Gomez 2017]. The Bluetooth SIG, entity responsible for the BLE standard, has been working consistently to enable mesh in BLE networks. As a result of one of such efforts, BLE is no longer restricted to the star network topology since specification 4.1 [bsi 2013]. Furthermore, the SIG has also published a mesh standard [bsi 2017].

Besides the Bluetooth SIG, the IETF has also been trying to come up with its own mesh standard for BLE based on its IPv6 standards initially published for the IEEE 802.15.4 technology, more specifically 6LoWPAN [Montenegro et al. 2007] and RPL [Alexander et al. 2012]. 6LoWPAN has now adaptation layers for other technologies, including BLE [Shelby et al. 2015], and there is work towards leveraging RPL

to achieve mesh topologies [Gomez et al. 2017]. However, ad hoc wireless communication for low power devices is a challenge. Its plethora of trade-offs have been addressed before in many studies [Huang et al. 2013], being still an open issue nowadays, specially when energy consumption is a major concern such as in the case of BLE [Darroudi and Gomez 2017].

Regarding energy consumption trade-offs, the BLE stack has many parameters for fine tuning latency, delivery rate, and energy performance. The effect of these parameters on BLE communications has been studied both analytically [Gomez et al. 2011, Gomez et al. 2012, Liu et al. 2012] and experimentally [Liu et al. 2013, Mikhaylov et al. 2013, Mackensen et al. 2012, Dementyev et al. 2013, Spoerk et al. 2017] for BLE star networks, Therefore, there is a lack of studies on this matter concerning the case of BLE mesh networks.

Our main contribution is an experimental evaluation of three different topologies for linking different BLE star networks into a single BLE mesh network. The topologies we evaluate are the minimal building blocks for BLE mesh networks and we analyze their trade-offs regarding energy consumption, delivery rate, and goodput. These metrics are obtained through experiments on a testbed comprised of Texas Instruments' CC1350 LaunchPad [tiC 2016] nodes running our own modified version of BLEach [Spoerk et al. 2017], an open source IPv6 over BLE stack built for the popular Contiki OS [Dunkels et al. 2004]. The obtained results agree with the single hop results published by [Spoerk et al. 2017], and together with the multi hop results they allow us to trace a lower boundary in terms of energy cost and performance degradation when forming BLE mesh networks. We then take this boundary as a basis to extrapolate our results for large scale network scenarios.

This work is organized as follows: Section 2 offers a brief introduction to the BLE standard and to the standardization attempts for BLE mesh networks. Section 3 brings a summary of all relevant works related to BLE and BLE mesh evaluation. Section 4 describes the experiments we propose for our evaluation. Section 5 presents our results and discussion. Section 6 expands our results by building estimates to predict scenarios of larger scale. Finally, Section 7 concludes this work.

2. BLE and BLE mesh networks

The BLE standard [bsi 2013] was designed as a communication stack focused on low power, low rate, and low cost communications. It is divided in two main software components, a host and a controller. The controller, usually implemented as a dedicated black box System on a Chip (SoC), is comprised by the link and physical layers. The host, on the other hand, implements stack management as well as transport, security, and application layers.

BLE communication is organized in two different states: connected and non-connected. In each state, a device may assume a different role related to how it performs communication tasks. In non-connected state, devices may be advertisers or scanners/initiators. In this state, they communicate by broadcasting, thus allowing for device discovery and connection setup. In connected state, on the other hand, devices assume the roles of masters and slaves and can tap more sophisticated resources such as bidirectional communication and access to transport, security, and application layer features.

The link layer is what makes BLE highly energy efficient. It organizes connected state communication in connection events. Connection events are periodic, with each instance being set in a different pseudo-random frequency channel, thus leveraging Time Division Multiple Access and Frequency Division Multiple Access techniques. These techniques reduce idle listening and overhearing, major responsables for wastes in low power communication [Huang et al. 2013]. They also improve resilience against interference in the crowded 2.4 GHz ISM band.

The period for such connection events to take place is named connection interval, an important parameter for the BLE link layer. The ability to skip these connection events (i.e. not participate in them to save energy) is also an important link layer parameter and is named slave latency. Since the payload of the link layer frame is only 23 bytes long, large chunks of data must be fragmented in order to be transmitted. Because of this, a connection event is a continuous exchange of packets between a master and one of its slaves. This exchange keeps going until: (1) both involved devices have no more data to transmit, or (2) the master decides to stop it so it can talk to another slave, or (3) a transmission error is detected.

Since specification 4.1, a device may simultaneously assume the roles of master and slave. This feature was introduced by the Bluetooth SIG specifically in order to enable mesh network topologies, since it allows a device in a star topology to discover or to be discovered by another device outside it. By this means, it is then possible to link two star networks together, thus forming a mesh network as depicted in Figure 1.

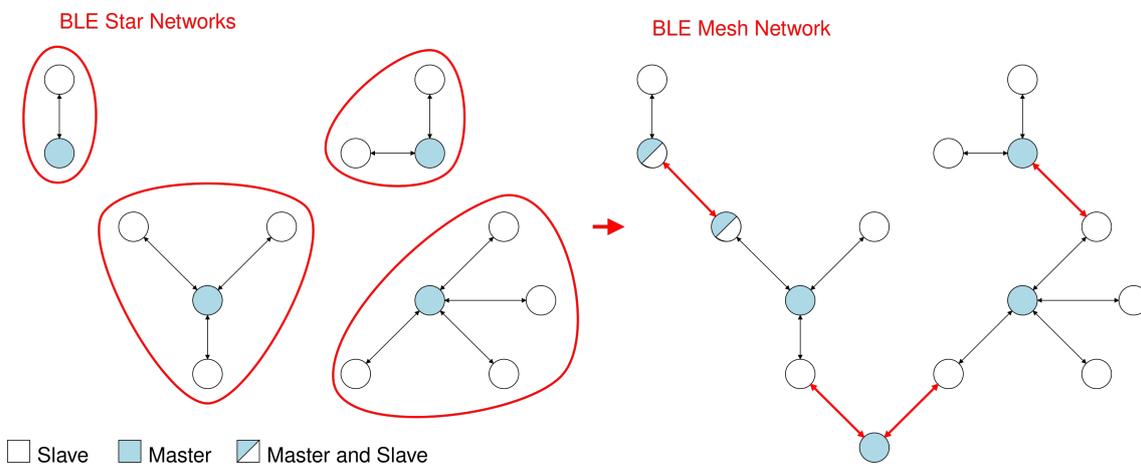


Figure 1. BLE mesh network composition from four star networks. On the top left, we have a master becoming also a slave of a node which is already a slave in another network. On the bottom, we place a new node as master of two nodes which are already slaves of other masters. On the top right, we make a node become slave of two different masters simultaneously.

As mentioned above, there are many attempts to make BLE mesh a reality. Darroudi and Gomez presented a comprehensive survey [Darroudi and Gomez 2017] on existing strategies to do so, addressing both standard and non-standard efforts, either attempted by the industry or by the academy. Its authors state their opinion that the attempts more likely to be successful are the standards maintained by the IETF and Bluetooth SIG. Both these standards build the mesh network on top of connected BLE devices in opposi-

tion of many existing attempts of building it solely on top of non-connected state.

The IETF standard is aimed to enable compatibility between the BLE link layer and the traditional IPv6 stack that operates under the whole Internet. The standard is published in RFC 7668 [Shelby et al. 2015] that leverages on the 6LoWPAN mechanisms proposed for IEEE 802.15.4 networks in RFCs 4944, 6282 and 6775 to provide IPv6 addressing, packet fragmentation, header compression and neighbour discovery for BLE star networks. It further allows the employment of the RPL routing protocol (RFC 6550) to enable mesh topologies in BLE networks by making use of features introduced for connected mode in BLE specification 4.1. However, this is still a work in progress [Gomez et al. 2017].

The mesh standard by Bluetooth SIG [bsi 2017] proposes a mesh network based on controlled flooding that may be built on top of devices complying to BLE specifications 4.0 to 5.0. It constrains applications to standard scenarios called models and allows unicast, broadcast, and multicast communication, both using devices in connected and in non-connected states. To this end, it builds a whole network stack on top of the BLE stack described in the core specification. We can see in Figure 2 how this stack compares to the IETF stack, specially in height and number of layers.

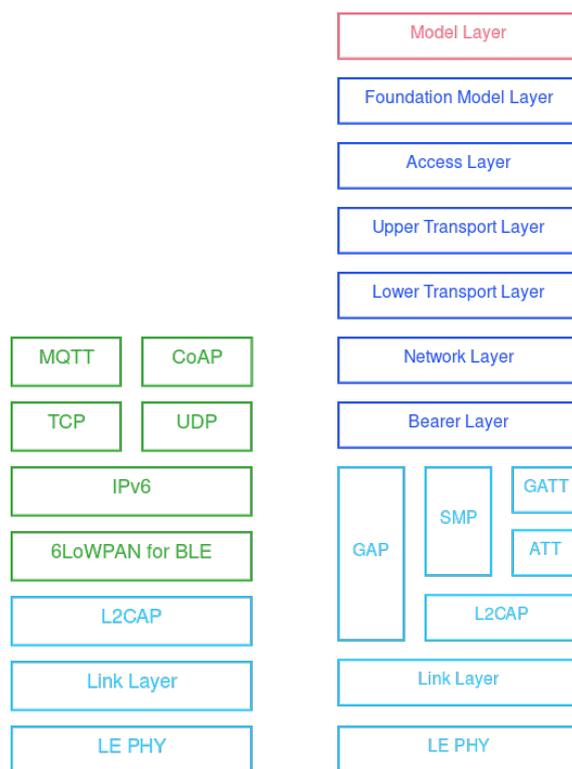


Figure 2. BLE mesh enabled stacks according to the IETF (left) and Bluetooth SIG (right) standards.

3. Related work

At the moment of writing, the majority of existing works base their studies on star topologies. Gomes et al. offer an analytical model [Gomez et al. 2011] of BLE throughput on connected state as a function of bit error rate (BER) and connection interval. In the

works of [Siekkinen et al. 2012] and [Gomez et al. 2012], analytical models are provided to evaluate performance regarding energy consumption and throughput based on the results of measurements with real connected BLE devices.

Evaluation of BLE device discovery has been done in [Liu et al. 2012] and [Liu et al. 2013] both analytically and experimentally regarding latency and energy consumption. The authors derive expressions for power consumption on advertising and scanning activities, then validate their results through simulations and experiments on real testbeds.

The experiments on [Mackensen et al. 2012] are performed on a miniature car and evaluate the performance on obtaining inertial measurements both through connected and non-connected states. The scenarios evaluated by [Dementyev et al. 2013] and [Mikhaylov et al. 2013] show how BLE can be fast and energy efficient when compared with different communication standards for WPANs, such as IEEE 802.15.4, ANT and SimpliciTI.

The authors of [Spoerk et al. 2017] model and experimentally characterize the latency and energy consumption in a BLE star network built with IETF's 6LoWPAN and IPv6 stack. They also show the impact of link layer parameters mentioned in section 2 (connection interval and slave latency) on latency and energy consumption, demonstrating how dynamic tuning of these parameters can greatly impact on the performance of BLE networks.

Finally, [Sirur et al. 2015] and [Lee et al. 2016] are the only ones we found that explore the challenges on employing the IETF standards to build BLE mesh networks using the RPL routing protocol. None of them seems to be concerned with the trade-offs of performance and energy consumption in restricted resource nodes, since the experiments in [Sirur et al. 2015] are executed with mobile phones and, in [Lee et al. 2016], using the Raspberry Pi platform.

To the best of our knowledge, there is still no work related to the recent standard published by Bluetooth SIG proposing BLE mesh networks.

4. Experimental method

The main goal of our experiments is to measure energy consumption, delivery rate, and goodput considering different network topologies fit to serve as chain links in a BLE mesh network. To this end, we consider all possible topologies consisting of three nodes that may enable us to put together two separate BLE star networks. We also consider two basic topologies comprised of two nodes, a master and a slave, to serve as basis for our comparisons. A description of all these topologies is provided below and can be visualized in Figure 3.

- **Topology 1:** node which is slave of two different master nodes, thus pertaining to two different star networks centered around each of the masters;
- **Topology 2:** node which is master to two slaves, each pertaining to two different star networks centered around other masters;
- **Topology 3:** node which is master to a slave in its own star network but is also a slave of another master, thus belonging to another star network centered around this second master.

- **Topology 4:** simple topology where we have one master and one slave linked to each other, with the slave as the central server node;
- **Topology 5:** simple topology where we have one master and one slave linked to each other, with the master as the central server node;

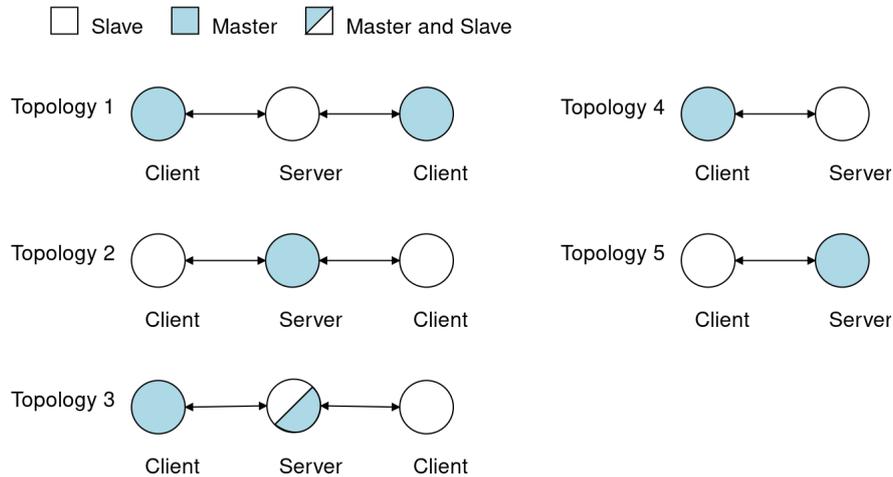


Figure 3. Evaluated topologies. In each one, our test application measures good-put and delivery rate by having a central node acting as a server and one to two marginal nodes acting as its clients.

We observe that, by applying any combination of the first three patterns described above, we may link different BLE star networks in order to compose a BLE mesh network. This is exemplified in Figure 1. We may therefore say that these topologies provide the minimal links for composing BLE mesh networks and that, by evaluating the cost and performance degradation with such links, we are evaluating the lower boundary on cost and performance degradation to link two different BLE star networks in one single BLE mesh network.

In order to measure such boundary, we modified BLEach [Spoerk et al. 2017], an open source IPv6 over BLE stack built on top of the Contiki OS, by making it capable of assuming different roles of master and slave in simultaneous fashion, such as proposed by the BLE specification 4.1. This allowed us to build a custom application for the TI CC1350 LaunchPad platform to test the topologies described above. As can be seen in Figure 3, we basically have one server node at the center, which is connected to one or two client nodes which are able to asynchronously send data to each other in a periodic fashion through UDP sockets. All nodes are placed less than 10 cm apart from each other, since it is not our intention to measure the effects of fading and interference. This setup runs in a typical office environment, with several WiFi networks and other Bluetooth devices running in the 2.4 GHz ISM band alongside it.

In our application, each node sends application layer payloads that are 100 bytes long with a time interval of one second between them. This is a stressed scenario if we consider WPANs, which are often employed in low power and low rate applications such as environmental sensing and home automation. Regarding link layer control parameters, our application sets BLE connections with a slave latency of 0 and a connection interval of 500 ms, so that we will not experience any congestion given our configured transmission

rate. We also enable all 37 available channels to be used in BLE's adaptive frequency hopping algorithm. Furthermore, our application can be configured in four different ways regarding network traffic flow, thus leading to four different load scenarios for each one of our possible topologies, as described below.

- **Bidirectional communication:** clients and server are independently sending data to each other with periodicity of 1 second.
- **Unidirectional communication from server:** server is independently sending data to each of its clients with periodicity of 1 second.
- **Unidirectional communication from clients:** clients are independently sending data to their server with periodicity of 1 second.
- **No communication:** BLE connection is maintained between server and clients, but no application data is exchanged so we can measure the cost of connection maintenance.

To evaluate delivery rate, we monitor the serial output of each node so that we may count how many packets left each source node and how many arrived at their destination nodes. Our estimates take into account the exchange of the first 500 packets exchanged after connection setup within each evaluated network scenario. We do this by making our application count received packets from each source through sequence numbers put inside the packets it sends.

To measure energy consumption we make use of a setup comprised by an MDO4014B mixed domain oscilloscope, along with a Zhaoxin KXN-305D DC power supply and one of ee-equipment's ee203 real-time current monitors. This setup can be seen in Figure 4. We took care to isolate power supply to the CC1350 in our CC1350 LaunchPad board, so that we would measure only the energy consumed by the SoC. For our evaluation we only consider the energy consumed by the node acting as server in our topologies. This is the node under the heaviest burden, since it is responsible for linking two different BLE star networks and thus relaying all the traffic between them.

For each of the studied cases, we first wait 20 seconds for the network to stabilize after all nodes have connected. Then we monitor current consumption through the oscilloscope for 20 seconds with a frequency of 1 Msamples/s, thus resulting in 20 Msamples per measured period. We then take the average of current consumption for each measured period of 20 seconds and thus get the charge in Ampere hour (Ah) consumed in each scenario for each topology.

Finally, in our experiments, we consider goodput, as defined in [Floyd 2008], in two different ways by measuring two forms of delay for transmitting our application data: (1) the latency for all activities related to the communication of one application payload between two nodes; and (2) the time on air spent for transmitting and receiving it. To measure both these time intervals, we also make use of the current monitoring setup we described above.

We consider latency (1) to be the time interval spent by the node in a BLE connection event, which can be taken by the time a node spends in transmission and reception activities (i.e. consuming more current than the average idle current) in a connection event. By making use of radio wave amplitude correlation in our equipment, we can detect the moments when nodes are exchanging packets and thus identify a connection event.

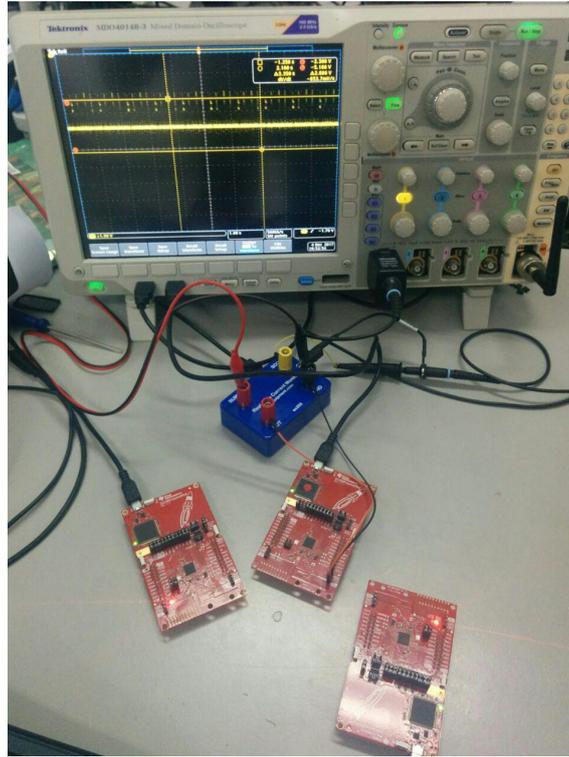


Figure 4. Energy measurement setup.

To do this we set our MDO4014B to capture frequencies in the 2.4 GHz ISM band and point its antenna to the nodes we are measuring. It is possible to see in Figure 5 how the moments where we have transmissions correlate to the instants we observe current consumption peaks related to transmission and reception in our monitored node. We can also measure the time on air (2) for the transmission of 100 bytes of application data through these captures.

5. Results and discussion

The energy profile in Figure 5 was taken for one of the connection events happening as the node in our testbed exchanged data. It shows current consumption (upper curve) and radio wave amplitude (lower curve) for our measured node during a BLE connection event. For each scenario and each topology (except for those where no application data was exchanged between nodes), we have captured ten similar profiles. We then measured communication latency (1) and time on air (2), as described at the end of section 4, and finally averaged them in order to obtain goodput and its standard deviation.

The results of these procedures showed a goodput of $25.9 \pm 1.0KB/s$ for definition (1) considering node latency on communication activities. As for definition (2), considering time on air, we obtained a goodput of $32.0 \pm 1.5KB/s$. The averages were taken considering all topologies and scenarios for which goodput was measured. Thus, as evidenced by the low standard deviation values, we can notice that there was not much variation between topologies 1 to 5 regarding goodput in all considered traffic scenarios.

The profiles obtained by measuring current consumption and radio wave amplitude in a connection event also enable us to characterize the efficiency of the BLE stack.

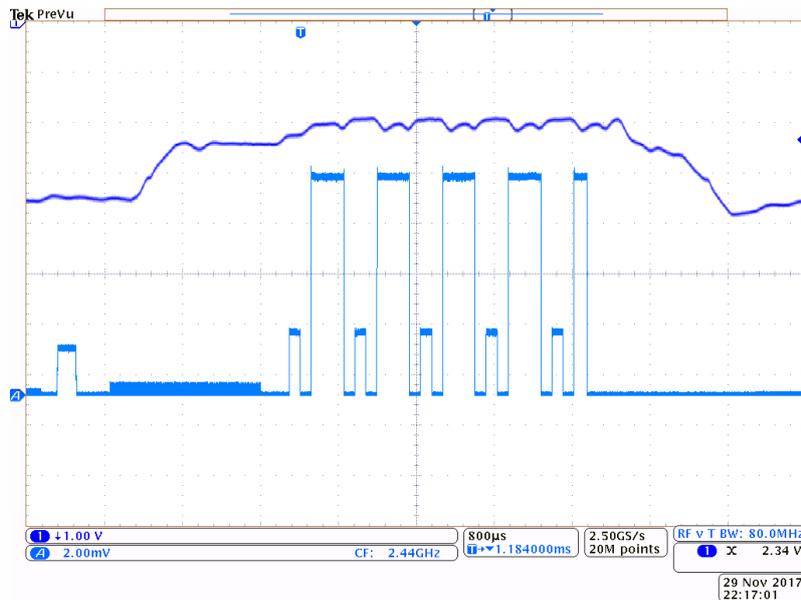


Figure 5. Energy profile of a connection event. In purple, energy consumed in transceiver activity (logarithm of electric current in Amperes). In blue, transmitted signal amplitude in mV.

We can do this by looking at Figure 5 and measuring the relative time when signal amplitude is significantly higher than our noise floor. The stack spends around 81% of a connection event performing actual data transmissions and receptions. This efficiency can vary with the amount of data transferred in each connection event. In our case, since our transmitted payload is larger than BLE link layer frames, we experience fragmentation, which contributes to increase this efficiency.

The results regarding delivery rate for each scenario and each topology can be seen in Figure 6. Although delivery rate was not affected in unidirectional communication, it was reduced to as low as 82.8% and 77.8% for bidirectional traffic, but only when using topologies 1 and 2. We believe that the cause for this performance loss regarding delivery rate is related to the scheduling of link layer connection events.

Increasing the number of concurrent links maintained by a node will also increase the chance for overlapping connection events pertaining to different interlocutor devices. When one of such overlaps occur, it will cause slaves to abruptly end connection events as if they had stopped answering their masters (in case of topologies 1 and 3). They will also cause masters to interrupt an ongoing connection event and delay data to a next one (in case of topologies 2 and 3), further increasing the chance of connection event overlapping. For masters, additional concurrent links will be even more problematic if the stack does not properly schedule different connections to start with a reasonable interval between each other, in order to prevent the overlapping of connection events.

Because the employed stack does not implement this kind of scheduling, we believe this was the cause for topology 1 to outperform topology 2 in the bidirectional scenarios represented in Figure 6. In topology 3, on the other hand, we have concurrent links maintained by different roles in the same node. We believe that this may help with link layer scheduling since the master role may end connection events so it can answer its own

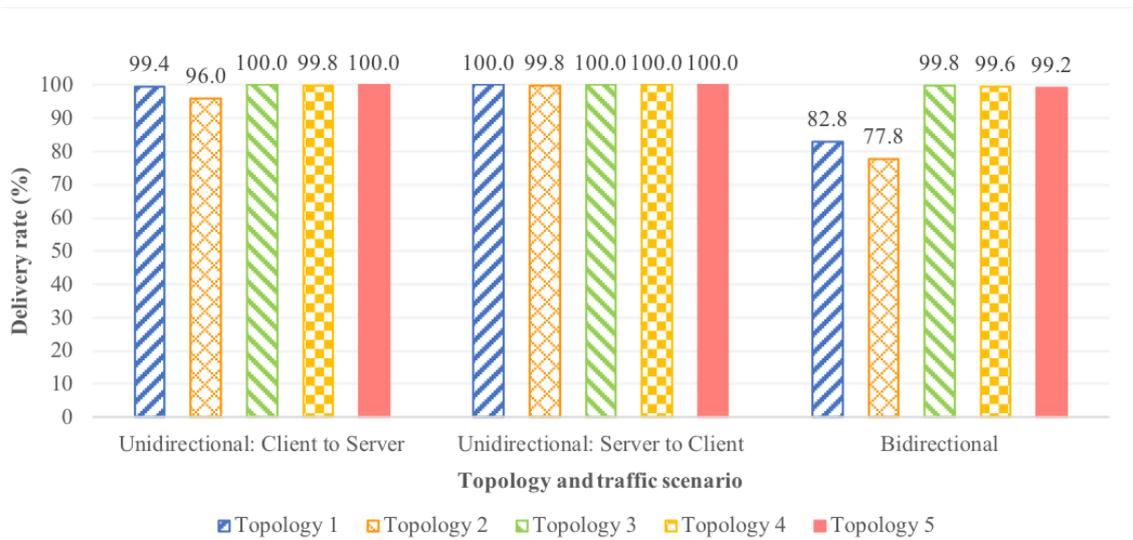


Figure 6. Delivery rate for each topology and traffic scenario.

master as a slave. For this reason, the slave role will be given priority, thus allowing the node to reach some scheduling balance that will prevent it from suffering with connection event overlaps.

Finally, Figure 7 shows the electric charge consumed in each of the studied topologies, considering each experimented network traffic scenario. Besides the values displayed on the chart, we also measured an average consumption of 173 uAh for a node kept permanently in sleep mode. These results are compatible with what is displayed by the chart in Figure 5 on the work of [Spoerk et al. 2017] for an equivalent of topologies 4 and 5, which here are serving as the basis for our comparison. Furthermore, Figure 7 reveals that connection maintenance cost is nearly the same for all considered topologies. This is expected since this specific scenario involves nodes exchanging only two packets of minimal lengths between each other at each connection event, just to keep their connection alive.

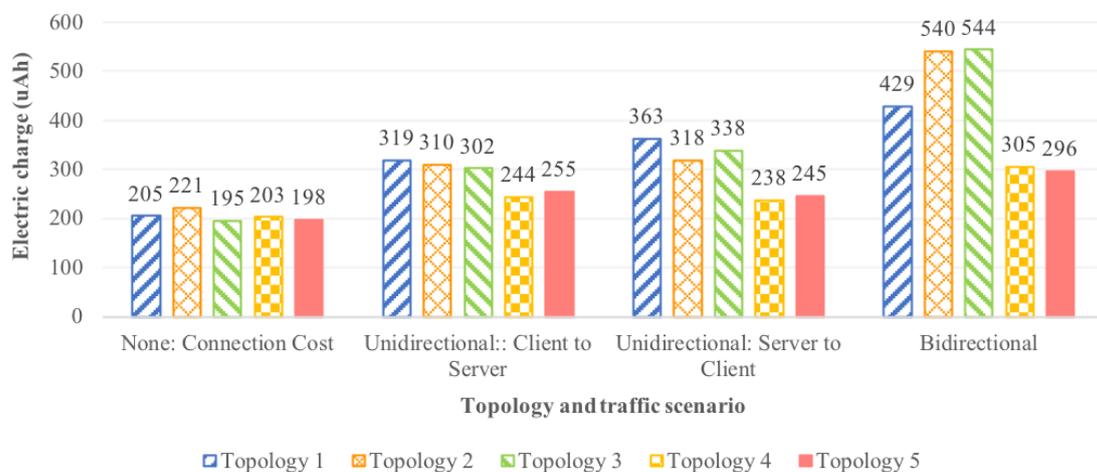


Figure 7. Energy cost for each topology and traffic scenario.

It is also possible to notice in Figure 7 how the first unidirectional scenario (Client to Server) is very similar to the second unidirectional scenario (Server to Client). In both scenarios, we can also observe topologies 4 and 5 to have the lowest energy consumption. This was expected because, in topologies 4 and 5, the measured node had only one link to maintain. As for topologies 1, 2, and 3, they have similar energy costs in these scenarios since they maintain the same number of simultaneous links.

Finally, we can also point out that the bidirectional scenario is, as expected, the most expensive one, since we double the amount of data exchanged between the nodes in relation to the unidirectional scenarios. Even so, we can see that topology 1 has a lesser cost than topologies 2 and 3. This is very plausible, since topology 1 places a slave as the central node and BLE slaves were designed to consume slightly less than masters as we can see comparing topologies 4 and 5, and was also demonstrated in [Spoerk et al. 2017]. Although this difference between topology 1 and topologies 2 and 3 was measured as only around 100 uAh, this could mean up to 790h of battery life for this device on a common 1500 mAh battery.

6. Study for mesh networks scalability

In this section we try to extrapolate based on the data gathered in our experiments in order to predict scalability in BLE mesh networks. Our first goal is to estimate the costs for adding to a node what we consider to be a mesh link: a link capable of joining together two separate star networks as in Figure 1. We derive these costs from the results obtained for the topologies we measured, which we consider as the building blocks for BLE mesh networks. This way we can try to forecast the costs implied for increasing the size of a BLE mesh network, both in terms of energy cost and delivery rate losses.

Since the energy cost for both unidirectional scenarios was so similar, we can take an average of both scenarios for each of topologies 1, 2, and 3 and compare them with topologies 4 and 5. Topologies 4 and 5 are meant to be our basis for comparison, and we assume its costs include both the cost of a single traditional link plus the base cost for a node. Based on this assumption we can subtract the cost for topology 4 (in case of topologies 1 and 3) or for topology 5 (in case of topologies 2 and 3) from the costs of other topologies in order to estimate the energy of the mesh link evaluated in each of topologies 1, 2, and 3. Specially for topology 3, which can be built both by adding links to topologies 4 and 5, we consider an average taking both costs of these topologies.

Following this method, we have the energy costs for: (a) making a slave node become a slave of yet another master; (b) making a master node become a master for yet another slave; (c) making a slave node become also a master for another slave. The results for this estimates are presented in Table 1. It would be expected that each additional link would render the average increase of equivalent amounts of consumed energy to a given node. We can see the costs are very similar for unidirectional scenarios but not for the bidirectional one. In bidirectional scenarios the cheaper cost for making a node slave to yet another master becomes evident.

We can also estimate the delivery rate along a path comprised of our evaluated links. Since delivery rate was very close to 100% to all topologies for unidirectional traffic, we consider only the stressed scenario of bidirectional traffic. Figure 8 shows our estimates for the delivery rate of messages sent along paths of up to 32 hops in this

Table 1. Additional average energy cost for each evaluated mesh link.

Traffic scenario	Link (a)	Link (b)	Link (c)
Unidirectional	100 uAh	64 uAh	74 uAh
Bidirectional	118 uAh	245 uAh	244 uAh

scenario. In our estimates, we assume each transmission to be an independent event and thus we elevate the rates we presented on Figure 6 to a power equal to the considered number of hops. The chart makes evident the negative effect of topologies 1 and 2 for scaling BLE mesh networks under heavy traffic scenarios.

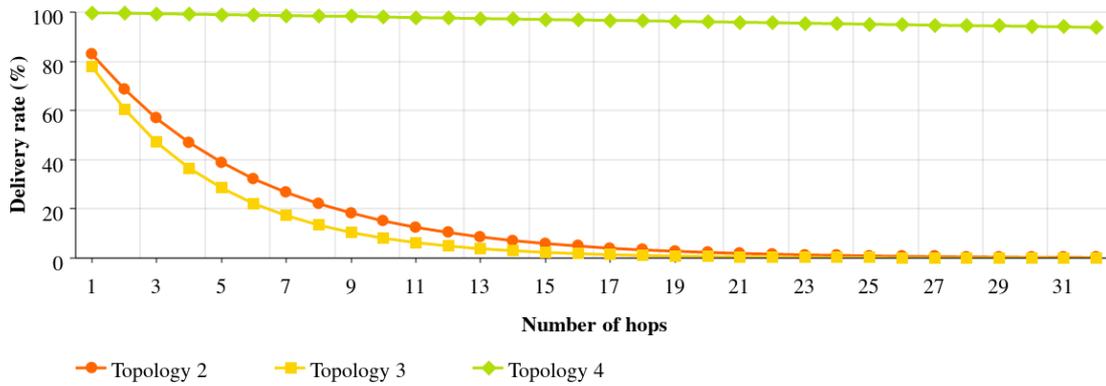


Figure 8. Delivery rate across a path of up to 32 hops.

We consider that the estimates in this section provide means for one to model a cost for each node placed in a BLE mesh topology. This modelling can be done in terms of energy expenditure, by taking into account the number of links and the traffic maintained by a node. The presented estimates can also aid on planning delivery rate degradation along the paths across a BLE mesh network. This planning should include employing the proper kinds of links in order to achieve range extension as expected along a route, considering the lifetime of the involved nodes and the devised traffic through them.

7. Conclusions

We have evaluated different minimal topologies for linking two BLE star networks into one single BLE mesh network. We also estimated an average cost for each kind of link represented by these topologies. We did that under four different load scenarios and thus measured the lower boundary in terms of energy cost and performance degradation using the metrics of energy consumption, delivery rate, and goodput for each one of these proposed scenarios. Through our results, we could see that, although goodput is not directly affected by the tested topologies, delivery rate and energy consumption may be.

Delivery rate can suffer from the addition of nodes and the increase of traffic due to how BLE connection events are scheduled. We can see that topologies 1 and 2 cause the greatest impact on delivery rate. By our estimates, we could infer that this impact would be even more severe for intense traffic scenarios in large scale networks. Energy costs, on the other hand, are influenced by the amount of network load. This happens

mainly because the increase of traffic means more transmissions and receptions, specially if we consider the central node which was the target of our measurements.

In terms of energy cost, topology 1 seems to be more effective in scenarios where network traffic is more intense, though we have to bear in mind that it also performed poorly in terms of delivery rate, so there is a trade-off to be considered. As future work, we plan to repeat these measurements for other BLE chipsets and stacks, in order to confirm if these results are valid even considering different vendor optimizations. We hope this will give us more evidence in order to build a mechanism that would take benefit from the observed trade-off to improve performance of BLE mesh networks by dynamically changing network topology based on traffic demand.

References

- (2013). *Specification of the Bluetooth System*. Bluetooth Special Interest Group, Kirkland, WA, USA.
- (2016). *CC1350 SimpleLink Ultra-Low-Power Dual-Band Wireless MCU*. Texas Instruments, Dallas, Texas. SWRS183A.
- (2017). *Mesh Profile*. Bluetooth Special Interest Group, Kirkland, WA, USA.
- Alexander, R., Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R., and Winter, T. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550.
- Chang, K. H. (2014). Bluetooth: a viable solution for iot? [industry perspectives]. *IEEE Wireless Communications*, 21(6):6–7.
- Darroudi, S. M. and Gomez, C. (2017). Bluetooth low energy mesh networks: A survey. *Sensors*, 17(7).
- Dementyev, A., Hodges, S., Taylor, S., and Smith, J. (2013). Power consumption analysis of bluetooth low energy, zigbee, and ant sensor nodes in a cyclic sleep scenario. In *Proceedings of IEEE International Wireless Symposium (IWS)*. IEEE.
- Dunkels, A., Gronvall, B., and Voigt, T. (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462.
- Floyd, S. (2008). Metrics for the Evaluation of Congestion Control Mechanisms. RFC 5166.
- Gomez, C., Darroudi, S. M., and Savolainen, T. (2017). IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP. Internet-Draft draft-ietf-6lo-blemesh-02, Internet Engineering Task Force. Work in Progress.
- Gomez, C., Demirkol, I., and Paradells, J. (2011). Modeling the maximum throughput of bluetooth low energy in an error-prone link. *IEEE Communications Letters*, 15(11):1187–1189.
- Gomez, C., Oller, J., and Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734.

- Hortelano, D., Olivares, T., Ruiz, M. C., Garrido-Hidalgo, C., and López, V. (2017). From sensor networks to internet of things. bluetooth low energy, a standard for this evolution. 17:372.
- Huang, P., Xiao, L., Soltani, S., Mutka, M. W., and Xi, N. (2013). The evolution of mac protocols in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 15(1):101–120.
- Lee, T., Lee, M. S., Kim, H. S., and Bahk, S. (2016). A synergistic architecture for rpl over ble. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9.
- Liu, J., Chen, C., and Ma, Y. (2012). Modeling neighbor discovery in bluetooth low energy networks. *IEEE Communications Letters*, 16(9):1439–1441.
- Liu, J., Chen, C., Ma, Y., and Xu, Y. (2013). Energy analysis of device discovery for bluetooth low energy. In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pages 1–5.
- Mackensen, E., Lai, M., and Wendt, T. M. (2012). Performance analysis of an bluetooth low energy sensor system. In *2012 IEEE 1st International Symposium on Wireless Systems (IDAACS-SWS)*, pages 62–66.
- Mikhaylov, K., Plevritakis, N., and Tervonen, J. (2013). Performance analysis and comparison of bluetooth low energy with ieee 802.15.4 and simplici. *Journal of Sensor and Actuator Networks*, 2(3):589–613.
- Montenegro, G., Hui, J., Culler, D., and Kushalnagar, N. (2007). Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944.
- Raza, S., Misra, P., He, Z., and Voigt, T. (2016). Building the internet of things with bluetooth smart.
- Shelby, Z., Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., and Gomez, C. (2015). IPv6 over BLUETOOTH(R) Low Energy. RFC 7668.
- Siekinen, M., Hiienkari, M., Nurminen, J. K., and Nieminen, J. (2012). How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pages 232–237, Paris, France. IEEE.
- Sirur, S., Juturu, P., Gupta, H. P., Serikar, P. R., Reddy, Y. K., Barak, S., and Kim, B. (2015). A mesh network for mobile devices using bluetooth low energy. In *2015 IEEE SENSORS*, pages 1–4.
- Spoerk, M., Boano, C. A., Zimmerling, M., and Romer, K. (2017). BLEach: Exploiting the Full Potential of IPv6 over BLE in Constrained Embedded IoT Devices. In *Proceedings of the 15th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*.
- Want, R., Schilit, B. N., and Jenson, S. (2015). Enabling the internet of things. *Computer*, 48(1):28–35.