Assessing the Feasibility of Combined BLE and Wi-Fi Communication for High Data Sensing Applications

Navya Sneha Chatharajupalli, Randolf Rotta, Reinhardt Karnapke, Jörg Nolte
Chair of Distributed/Operating Systems,
Brandenburgische Technische Universität Cottbus-Senftenberg
chathara@b-tu.de, rottaran@b-tu.de, karnapke@b-tu.de, joerg.nolte@b-tu.de

Abstract

Advanced sensors generate more data than can be transmitted using classic battery-powered wireless sensor networks. While using Wi-Fi would provide plenty of throughput, the receivers are energy hungry and, thus, the radio time needs to be reduced. Previous research suggests to combine Wi-Fi with low-power technologies such as BLE for an energy efficient coordination of the Wi-Fi radios. Existing approaches focus on individual point-to-point routes whereas certain applications require concurrent transmissions. We evaluated the combination of BLE and Wi-Fi on the ESP32-S3 multi-radio microcontroller through detailed energy and throughput measurements. The results show that established BLE connections allow to activate Wi-Fi quickly on-demand; the energy efficiency of the data transmission is significantly improved over previous works but the BLE connection setup is prohibitively slow. While the combination of BLE and Wi-Fi provides high throughput with good energy efficiency, careful design of the BLE-based signalling protocol is necessary to also achieve low latency.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—Wireless Communication

General Terms

Measurement,Performance

Keywords

BLE, Wi-Fi, Cross-Technology Communication

1 Introduction

When deploying sensor networks and IoT installations, wireless technologies such as IEEE 802.15.4, ZigBee, and Bluetooth Low Energy (BLE) are used often. These are optimized towards low energy consumption for small payloads, for example applications with small sample size and low sampling rate. However, applications such as predictive maintenance based on vibration sensors generate much bigger data sets. For example, the STMicroelectronics IIS3DWB accelerometer samples 3 axes with 16 bit resolution at 26.6 kHz, generating 160 kB/s of raw data while just consuming 1.1 mA. Transmitting such data via low-throughput radio links is inefficient because the radio and processor can no longer go into power-saving sleep.

IEEE 802.11 Wi-Fi would be able to provide the needed throughput and is supported by a few microcontroller platforms. The energy efficiency of battery-powered Wi-Fi devices is based on eliminating unnecessary idle listening times. One approach proposed in the research literature is the combination with low-power technologies such as Bluetooth Low Energy. The low power radio is then used as a wake-up radio for the bulk data transmissions via Wi-Fi. It can also be used for route discovery and repair, and small application payloads [3].

Previous works propose cross-technology communication radios, which are not commercially available, or used microprocessor-based hardware that lacked low-power modes. We implemented building blocks on the Espressif ESP32-S3 micro-controller in order to assess its feasibility. These SoCs provide 2.4GHz IEEE 802.11 b/g/n Wi-Fi and BLE 5.0 radios with a shared antenna. By attaching a 32.768 kHz crystal, low power modem sleep for BLE connections can be activated.

We present detailed throughput and energy measurements and analyze the overhead introduced by the combination of BLE and Wi-Fi. The evaluation shows that this combination is well suited for applications where 10 kB to 2 MB per second need to be transmitted.

2 Related Work

Studies that combine BLE and Wi-Fi can be grouped into different motivations: BLE to configure Wi-Fi connections ([5]); Cross-Technology Communication ([4],[6], [7]) to reuse Wi-Fi hardware and also communicate with BLE devices; and using BLE as wake-up radio for Wi-Fi.

The authors of Wi-BLE [3] combine BLE and Wi-Fi to improve the energy efficiency and throughput for multi-hop transmissions. They use BLE advertisements to implement a route discovery based on Ad-hoc On-demand Distance Vector routing and IPv6 addressing. A chain of BLE connections is established along the route and then used to start the Wi-

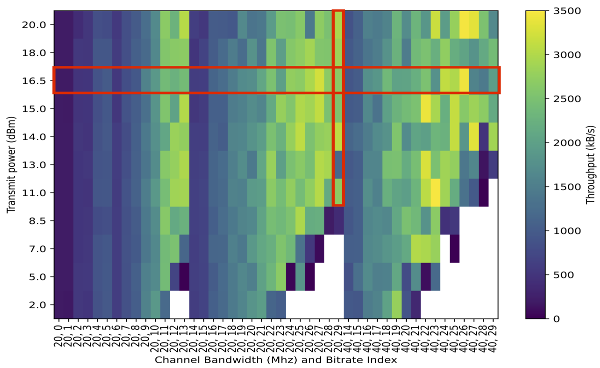

Figure 1. WIFITX+BLE throughput with a concurrent BLE connection at 50m distance and 73 kB payload.

Fi radios along the path. They evaluated their approach in three scenarios: Using BLE as wake-up radio together with Wi-Fi-based AODV routing; reusing the BLE routes as Wi-Fi routes; and optimizing the BLE routes with Wi-Fi-based AODV routing.

The main differences between [3] and our approach are the used hardware and the intended communication scenario. A Raspberry Pi micro-processor was used in [3], which does not provide low-power sleep modes. For a single-hop, the paper reports an energy utilization of 600 kBit/J (2.6µC/B at 5V) for a throughput of 4 MBit/s (512 kB/s), while our implementation achieved an energy utilization of 23 674 kBit/J (0.066µC/B) and throughput of 13.36 MBit/s (1.6 MB/s) for a similar 584 kB payload. Our better efficiency is also achieved by a more aggressive timing for the Wi-Fi radio.

While Wi-BLE focuses on point-to-point connections, we focus on collection tree scenarios. The authors used a 60s timeout for stopping Wi-Fi after the last transmission. Our experiments indicate that fast deactivation of the Wi-Fi radio is crucial for the overall energy efficiency.

3 BL-Fi Link Management Protocol

The protocol operates in 5 steps: opening a BLE connection, notifying the remote side to start Wi-Fi, transmission of data, notifying the remote side to stop Wi-Fi, and closing the BLE connection. The choice when to open and close BLE connections, the direction of Wi-Fi activation, and the direction of data transfers would depend on the routing protocol. Our design serves the experiments in Section 4.

On startup, the sender begins with BLE advertising, while the receiver side activates BLE scanning. When the receiver finds the sender's advertisement, it initiates a BLE connection. Success is signalled through an event on both sides. Then, the receiver subscribes for BLE notifications. The sender side waits for this subscription, which marks the completion of the connection setup.

The first step before transmissions is to send a BLE notification to start the other side's Wi-Fi. The sender waits for an acknowledgment and, then, turns on its own Wi-Fi. The data is sent in custom IEEE 802.11 data frames through a low-level API of the ESP32. The Wi-Fi MAC layer takes care of the Carrier Sense Multiple Access/Collision Avoidance,

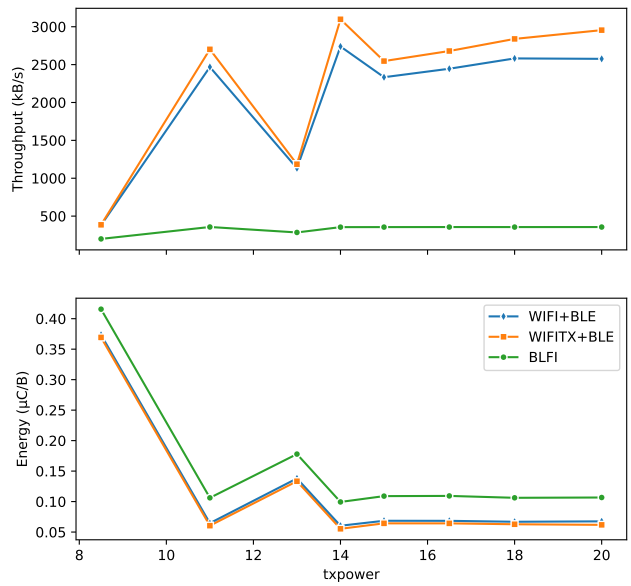

Figure 2. Varying transmit power at bit rate MCS7_SGI at 50m with 73 kB payload.

acknowledgments, and automatic re-transmission. Received frames are passed to a promiscuous mode event handler.

After completing all Wi-Fi transmissions, the sender turns off its Wi-Fi radio and sends a BLE notification to the receiver to do the same. The receiver turns off its Wi-Fi and acknowledges this via BLE. The analysis in Section 4.3 shows that this can be improved by using Wi-Fi messages to switch off the Wi-Fi radio.

4 Experiments

The first experiment setup focuses on the impact of the Wi-Fi bit rate, channel bandwidth (20/40 MHz), and transmit power (2–20 dBm) at a fixed payload size. The second setup focuses on the impact of the payload size.

The throughput and energy efficiency is measured with respect to different protocol layers: *BLE* refers to data transmission through BLE notifications. *WIFI* refers to the time from starting to stopping the radio, including the transmissions. *WIFITX* measures just from enqueuing the first message to receiving the last acknowledgement. *WIFITX+BLE* and *WIFI+BLE* include the overhead of maintaining a concurrent BLE connection with 50ms connection interval. *BLFI* is measured from sending the BLE notification to receiving the acknowledgement that Wi-Fi has been switched off again. Comparisons between these will provide insights into the impact of the introduced overheads.

Note that the WIFI and WIFITX benchmarks have the receiver active all the time. Only the BL-Fi implementation has the ability to switch on the receiver's Wi-Fi on demand.

4.1 BL-Fi with Fixed Payload Size

First we conducted the BL-Fi experiment for a fixed payload, for this two nodes were placed 50m apart on a lawn in 1m height. We measured the throughput and energy

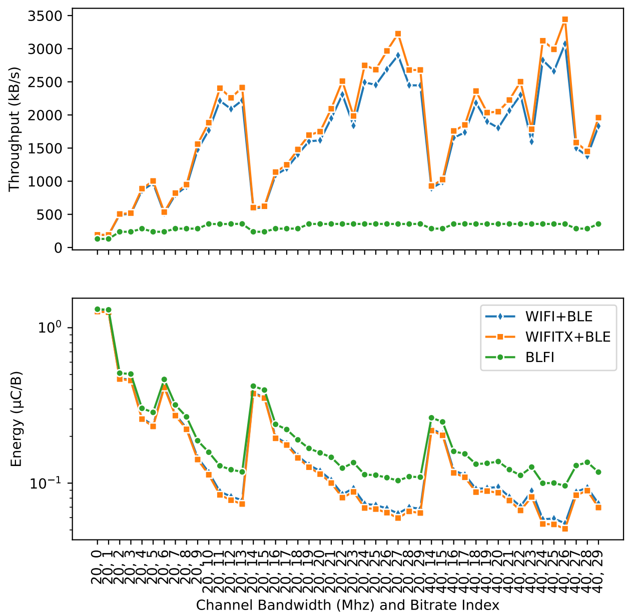

Figure 3. Varying bit rate at 16.5 dBm power at 50m with 73 kB payload.

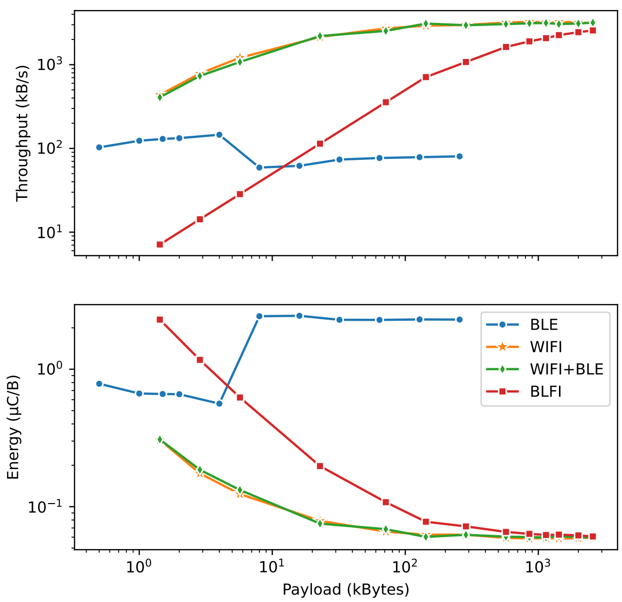

Figure 4. Variable payload sizes at a fixed bit rate MCS7_SGI at 20 MHz and 16.5 dBm transmit power.

consumption for all combinations of bit rates and transmit power levels for 50 Wi-Fi frames with 1462 bytes of payload (73.1 kB in total).

The Wi-Fi transmit throughput with a concurrent BLE connection is shown in Figure 1. All bit rates and channel bandwidth combinations were usable above 14 dBm transmit power and higher power did not improve the throughput.

Figure 2 shows varying transmit power for bit rate MCS7_SGI (72.2 MBit/s) with 20 MHz channel bandwidth. This corresponds to the vertical cut in Figure 1. On first sight, the BL-Fi throughput appears to be much worse than the Wi-Fi transmissions but this is caused by the necessary BLE latency needed to start and stop the Wi-Fi radio. The low throughput indicates that the payload is too small or the BLE connection interval is too large. In contrast, the energy efficiency is near to the Wi-Fi transmissions. This is caused by the much lower consumption during the BLE notification phases while the Wi-Fi radios are off.

Figure 3 shows varying bit rate at transmit power 16.5 dBm. This corresponds to the horizontal cut in Figure 1. The highest throughput and best energy efficiency were attained with 40 MHz channel width. However, the advantage over the best results at 20 MHz is not big, notably far below the 2x improvement that should have been achieved with twice the bandwidth. We observed that the ESP32 uses Wi-Fi RTS/CTS messages, which adds a constant latency that is much longer than the 40MHz transmission. Thus, mesh networks might refrain from 40 MHz channels and forego the RTS/CTS mechanism because it is ineffective anyway [2].

4.2 Variable Payload Size: BLE, Wi-Fi, BL-Fi

The second experiment focuses on the impact of the payload size. For the Wi-Fi transmissions, the payload size ranges from 1.46 KB to 2.631 MB, which corresponds to 1 to 1800 Wi-Fi frames. For the BLE transmission, the payload ranges from 512 bytes to 262.144 kB with 512 byte per BLE GATT write+notify operation. The BLE transmit power was set to 9 dBm.

Figure 4 compares the throughput and energy consumption. BL-Fi surpasses the throughput of BLE-based transmissions at 10 kB payloads and the energy efficiency already at 4 kB. This indicates the minimum amount of data that needs to be aggregated for BL-Fi alike protocols.

The difference between BL-Fi and Wi-Fi throughput vanishes with increasing payload size. This has to be expected because the BLE notification overhead is constant and independent of the payload size. More importantly, the results show, this is very promising for the realization of multi-hop forwarding with aggregation for multiple sensors. The throughput for the 2.6 MB payload was 2.5 MB/s. Thus, the nodes would be able to receive, aggregate and forward around 1 MB of data per second.

The BLE throughput ranges from 70 to 80 KB/s, which is around 650 kBit/s and in-line to what is achievable with 1 MBit/s BLE connections [1]. The energy efficiency for BLE-based transmissions worse than Wi-Fi, which confirms that it should be used only for signalling.

4.3 Analysis of the Power Profiler Traces

The Nordic Power Profiler Kit II was used to measure the energy consumption. Figure 5 shows an example that includes the BLE notifications and Wi-Fi data transmissions. The logic line 2 shows the BL-Fi transmission phase. Two BLE connection intervals are needed to notify the receiver side and receive the acknowledgement. Then, logic line 3 shows the Wi-Fi transmission phase. This includes starting and stopping the Wi-Fi radio. Finally, two BLE connection intervals are needed to stop the Wi-Fi on the receiver side.

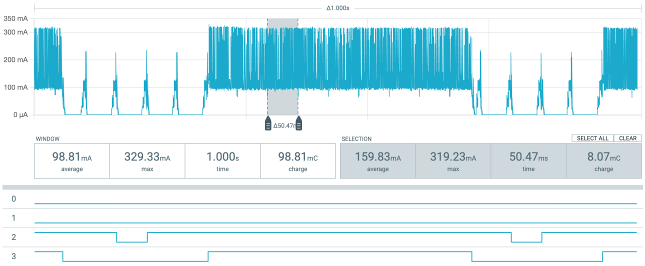

Figure 5. Current trace that includes starting and stopping the Wi-Fi radio via BLE and transmitting data via Wi-Fi.

The trace shows the effectiveness of the ESP32's light sleep between the BLE connection intervals. However, the time needed to activate the BLE radio, send and receive a BLE messages and going back to sleep is quite long with up to 7 ms. The BLE radio becomes active around 4 ms before the incoming BLE packet. This could be a guard time to compensate for clock drift.

When looking closer into the Wi-Fi transmissions, the BLE connection interval appears to have a guard time of just around 0.7 ms. Switching to BLE seems to be faster when the Wi-Fi radio is already active. It was also visible, that the BLE connection event simply interrupts the ongoing Wi-Fi transmission and continues with the remaining transmission. Of course, such messages cannot be received. Thus, the impact of a concurrent BLE connection on the Wi-Fi throughput can be estimated: Each connection interval causes the re-transmission of one Wi-Fi frame.

We also analysed the BLE connection setup. Around 6 advertisement intervals were needed before the other device connected. The connection setup took 36 connection intervals in order to query the BLE server for service, characteristic, and descriptor handles and subscribe for notifications. Thus, BL-Fi works best when keeping BLE connections alive over long time or when starting with a very short connection interval and adjusting it after the initialization.

4.4 Estimating the Energy Consumption

Lets assume n incoming and 1 outgoing connections. The node receives n data streams from neighbors and forwards this data together with the own sensor data to one upstream neighbor. The BLE connection interval T_{CI} defines the number of connection events, i.e. beacon transmissions, per second. Each connection event during light-sleep consumes around C_{conn} of charge multiplied by the number of connections $n+1$. The charge consumed during light-sleep is the average sleep current I_{sleep} multiplied by the runtime t. Then, the latent consumption for maintaining the network is

$$C_{BLE}(t) = \frac{C_{conn}}{T_{CI}}(n+1) \cdot t + I_{sleep} \cdot t \qquad (1)$$

We estimate the charge needed for maintaining 8 BLE connections with 50ms connection interval for 24 hours as 8337 C. With a battery capacity of 2600 mAh = 9360 C, the lifetime would be just a day. The reason is the very low BLE connection interval of 50ms. To actually sleep, an interval of at least 500 ms is needed for 8 open connections.

The consumption of the BLE notifications in the BL-Fi protocol is already covered in the BLE maintenance above. Let C_W be the consumption for starting and stopping the Wi-Fi radio. For all data streams we assume a data rate D of 200 KB/s. Once per second, each neighbor requests a Wi-Fi transmission and $(n+1)200$ KB are sent to the upstream neighbor. We assume, that these transmissions do not overlap. In practise they would overlap, which hopefully reduces the overhead. The time required to transmit D bytes at throughput thr is multiplied with the average current I_{Tx} of the chosen Wi-Fi configuration. Then, the charge consumed for t seconds of sensor data is

$$C_{BLFI}(t) = (n+1)C_W \cdot t + I_{Tx}\frac{(2n+1)D}{thr} \cdot t \qquad (2)$$

We estimate the charge needed for the data transmissions to be 180 mC. A 2600mAh battery would allow to transmit and forward sensor data for 14 hours, ignoring the BLE maintenance. Improvements are possible by exploiting the 8 to 16 MB memory of the ESP32-S3 to aggregate data.

5 Conclusions

This paper presented a combined BLE and Wi-Fi link management protocol *BL-Fi* for low-power data transmission in battery powered sensor networks. Using BLE for signaling and Wi-Fi for high throughput transmissions reduces the overall energy consumption. We have shown the feasibility on the ESP32-S3 processor family. Although the energy efficiency of the integrated BLE radio is not optimal, it is sufficient if large connection intervals are used.

6 Acknowledgments

This research was partly funded by the Bundesministerium für Digitales und Verkehr under project number 19FS2024D (FlexiDug) and by the German Federal Ministry of Education and Research grant 16ES1128K as part of the iCampus 2, Cottbus.

7 References

[1] P. Bulić, G. Kojek, and A. Biasizzo. Data transmission efficiency in bluetooth low energy versions. *Sensors*, 19(17), 2019.

[2] G. R. Hiertz, S. Max, Y. Zang, T. Junge, and D. Denteneer. Ieee 802.11s mac fundamentals. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–8, 2007.

[3] M. Lee, T. Lee, H.-S. Kim, and S. Bahk. Wi-BLE: On cooperative operation of Wi-Fi and Bluetooth low energy under IPv6. *Journal of Communications and Networks*, 24(4):489–502, 2022.

[4] Z. Li and Y. Chen. BlueFi: Physical-layer Cross-Technology Communication from Bluetooth to WiFi. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 399–409, 2020.

[5] W. Park, D. Ryoo, C. Joo, and S. Bahk. BLESS: BLE-aided Swift Wi-Fi Scanning in Multi-protocol IoT Networks. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[6] Y. Shu, J. Wang, L. Kong, J. Yu, G. Yang, Y. Cai, Z. Wang, and M. K. Khan. WiBWi: Encoding-based Bidirectional Physical-Layer Cross-Technology Communication between BLE and Wi-Fi. In *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 356–363, 2021.

[7] Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, and J. Ma. Zifi: Exploiting cross-technology interference signatures for wireless lan discovery. *IEEE Transactions on Mobile Computing*, 13(11):2675–2688, 2014.