Wireless Low-Latency Synchronization for Body-Worn Multi-Node Systems in Sports

Nico Krull[1], Lukas Schulthess[1*], Michele Magno[1], Luca Benini[12], Christoph Leitner[1*]
[1]*Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland*
[2]*Department of Electrical, Electronic, and Information Engineering, University of Bologna, Italy*

***Abstract*—Biomechanical data acquisition in sports demands sub-millisecond synchronization across distributed body-worn sensor nodes. This study evaluates and characterizes the Enhanced ShockBurst (ESB) protocol from Nordic Semiconductor under controlled laboratory conditions for wireless, low-latency command broadcasting, enabling fast event updates in multi-node systems. Through systematic profiling of protocol parameters, including cyclic-redundancy-check modes, bitrate, transmission modes, and payload handling, we achieve a mean Device-to-Device (D2D) latency of 504.99 ± 96.89 µs and a network-to-network core latency of 311.78 ± 96.90 µs using a one-byte payload with retransmission optimization. This significantly outperforms Bluetooth Low Energy (BLE), which is constrained by a 7.5 ms connection interval, by providing deterministic, sub-millisecond synchronization suitable for high-frequency (500 Hz to 1000 Hz) biosignals. These results position ESB as a viable solution for time-critical, multi-node wearable systems in sports, enabling precise event alignment and reliable high-speed data fusion for advanced athlete monitoring and feedback applications.**

***Index Terms*—Sport, Biomechanics, WBAN, Wireless, Wearable, Sensor**

I. Introduction

Human movements, particularly in sports, involve rapid and complex motion sequences [1]. Relying solely on single-source data (i.e., cameras) is often insufficient for assessing the quality of movements [2]. Consequently, multi-sensor-multi-node systems have become essential tools for analysis in sports applications [3]. These systems measure critical biomechanical variables, such as segment accelerations, forces, and muscle activation, in real-time, utilizing samples collected from unobtrusive, distributed sensor nodes on the athlete's body or embedded into sporting equipment. By gathering this information from sensors such as e.g. Inertial Measurment Units (IMUs), pressure insoles [4], or Electromyography (EMG) devices [5], a more comprehensive understanding of an athlete's movement and performance can be obtained. However, accurate data fusion across multiple nodes demands tight synchronization, particularly because biomechanical data acquisition often requires sampling rates ranging from several hundred Hz (e.g., IMU, pressure sensors) to over 1 kHz (e.g., EMG) [6]. Consequently, synchronization errors below 1 ms are key. This ensures that high-speed biomechanical events are accurately positioned in time across all sensor data streams, enabling precise temporal correlation and on-device processing or post-action analyses.

*Corresponding authors: L. Schulthess (lukas.schulthess@pbl.ee.ethz.ch), C. Leitner (e-mail: christoph.leitner@iis.ee.ethz.ch).

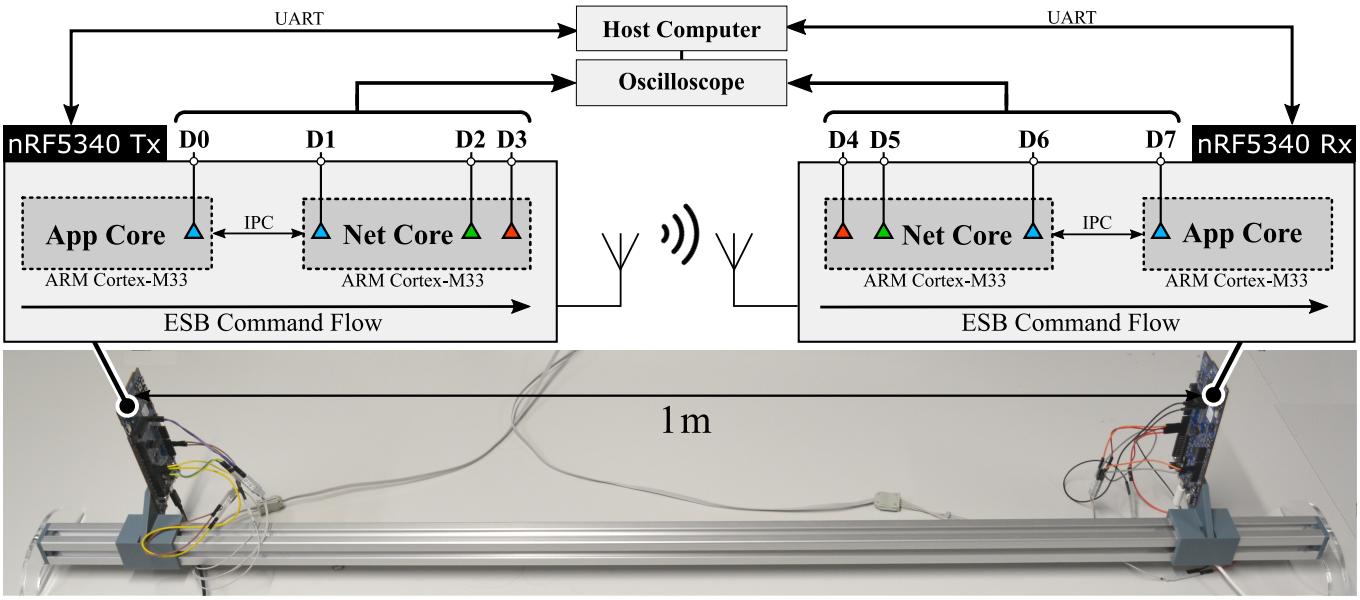
Bluetooth Low Energy (BLE) is a widely adopted communication standard in Wireless Body Area Networkss (WBANs) due to its wide availability and easy integration. With a low power consumption in the mW range, and a relatively high throughput of up to 2 Mbit s^{-1}, and the ability to support multi-node connectivity, it is well-suited for communication among wearables [7]. However, BLE's communication scheduling, based on coarse-grained connection intervals [8] and adaptive frequency hopping, inherently introduces latency and jitter [9]. Consequently, achieving deterministic sub-millisecond synchronisation with unmodified BLE remains challenging [10]. While connection intervals for packet transmission can be set statically, they are limited to a lower bound of 7.5 ms [11], which restricts fast, short-term event triggering. To address these limitations, previous works have explored analogue current profiling of transmitter (TX) and receiver (RX) as an indirect method for temporal alignment via crystal drift correction [6]. However, implementing this approach in practice requires additional hardware-assisted workarounds outside the BLE stack. In contrast, the Enhanced ShockBurst (ESB) protocol is designed for low-latency, low-power, short-range wireless communication, offering a promising alternative for fast data transmission on the 2.4 GHz Industrial, Scientific, and Medical (ISM) band, enabling sub-millisecond latency for time-critical applications [12].

The main contribution of this study is an accurate performance profiling of the ESB protocol in a controlled laboratory setting using two Nordic Semiconductor nRF5340 development boards. By probing signals at key points within the software stack of the dual-core architecture of both TX and RX nodes using General Purpose Input/Output (GPIO) signaling, we identify optimal configurations for achieving deterministic, low-latency synchronisation suitable for high-frequency biomechanical applications.

II. Methods

The ESB protocol [12] is a low-power, low-latency wireless communication protocol from Nordic Semiconductor offering bidirectional data transfer with packet queuing in TX and RX First-In, First-Out (FIFO) buffers, acknowledgements, and automatic retransmissions of lost packets.

For the characterization of ESB parameters, the payload size was set to 8 bytes of bit-encoded command data. Moreover, acknowledgement has been disabled to allow broadcast communication without waiting for responses. TX and RX devices



Transmitter		Receiver	
D0:	IPC transmit initiated (app core)	**D4:**	First command in ESB library after reception (net core)
D1:	IPC received (net core)	**D5:**	ESB event handler received payload (net core)
D2:	After ESB command, send payload (net core)	**D6:**	IPC transmit initiated (net core)
D3:	Last command in ESB library before transmission (net core)	**D7:**	IPC received (app core)

Fig. 1: Overview of the measurement setup and the GPIO trigger point locations.

are configured on the same ESB data pipe for uniform reception. The retransmission parameter was set to two, meaning that each packet is transmitted a total of three times, with a minimum delay of 435 µs between consecutive transmissions.

A. Parameter Configuration

The following ESB protocol parameters were investigated for their impact on latency performance:

CRC mode: Cyclic Redundancy Check (CRC) can be configured to 16 bits, 8 bits, or disabled entirely, affecting error detection reliability and effective payload size.

Protocol mode: can be configured as either fixed or dynamic payload length.

Bitrate mode: can be configured to 1 Mbit s^{-1} or 2 Mbit s^{-1}, either using standard settings or BLE-specific modes with adjusted radio parameters.

TX mode: offering automatic, manual, and manual start modes, influencing how packets are dequeued from the TX FIFO.

TX power levels: adjustable from −70 dBm and 10 dBm.

Besides optimizing the ESB protocol parameters, the payload construction was also refined to further reduce latency. In the standard setup, the Application Core (APP-core) generates the data and transfers it to the Network Core (NET-core) via Inter-Process Communication (IPC) before it is written to the TX FIFO. In the mode optimised for time-critical execution, a minimal payload size of 1-byte is pre-constructed directly on the NET-core, allowing it to remain ready for immediate transmission upon command. This approach eliminates the need for additional IPC transfers and runtime payload formatting, thereby reducing transmission latency. For a fair comparison between the standard and optimized payload construction modes, the payload size was fixed at 1-byte for these investigations.

B. Measurement Setup and Characterization

The measurement setup consisted of four key components: (1) a digital oscilloscope (MSOX3024T, Keysight Technologies Inc.), two nRF5340DK development boards (Nordic Semiconductor) as TX (2) and RX (3), and a host computer (4). Both TX and RX boards were mounted on a custom linear rail, spaced 1 m apart with upward-facing, opposing antennas to ensure consistent and comparable measurements (see Figure 1). A host computer orchestrated the test setup via a Python script. The computer handled communication with the TX and RX boards by sending configuration commands via Universal Asynchronous Receiver/Transmitter (UART) and retrieving measurements from the oscilloscope for post-analysis using Standard Commands for Programmable Instruments (SCPI). The nRF5340 Microcontrollers (MCU) features an APP-core and a NET-core, with the ESB application running entirely on the NET-core and requiring data transfer between cores via IPC. To capture precise event timing, GPIO pins were toggled at key stages of the Tx-Rx process and recorded by the oscilloscope (referred to as Digital input channel of the oscilloscope (D), see caption of Figure 1). Meanwhile, the RX board also returned the collected payload to the host computer for further analysis.

C. Measurement Protocol and Analyses

Each parameter in Section II-A was independently evaluated for its impact on transmit latency, with all other parameters held constant. All possible configuration options were tested,

TABLE I: NUMBER OF PACKETS TRANSMITTED AND RECEIVED UNDER THE THREE CRC CONFIGURATIONS. AN 8-BYTE PAYLOAD WAS SENT IN FIVE ROUNDS OF 150 TRANSMISSIONS PER CONFIGURATION.

	16-bit CRC	8-bit CRC	CRC off
Sent	750	750	750
Received	736	740	735
Unique	736	740	734
Valid	736	740	719

and their order was randomly shuffled before each test round to minimize systematic bias. Depending on the number of configurations, three to five rounds were conducted, each recording 150 TX attempts to ensure statistical reliability. The oscilloscope was configured to trigger on the rising edge of D3, with a time base of 600 µs/Div for a 6 ms capture window and 60000 sampled points per transmission, providing a temporal resolution of ±0.1 µs. This measurement setup was kept consistent across all tests to ensure high precision in latency evaluation. For the experimental data analysis, the mean, median, and Standard Deviation (SD) were calculated.

III. RESULTS AND DISCUSSION

To determine the Optimal Latency Configuration (OLCfg) concerning latency, various ESB parameters were evaluated and compared.

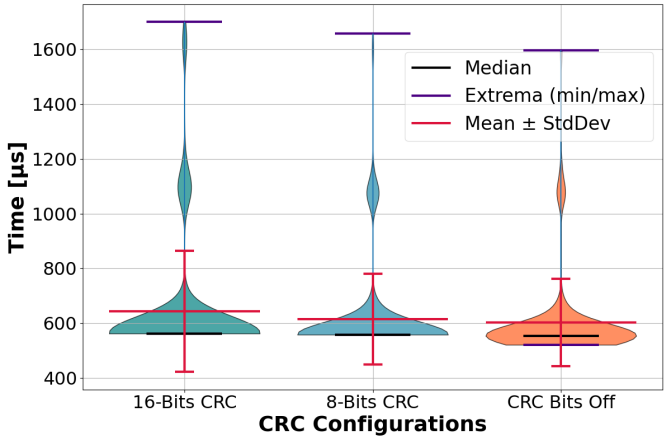


Fig. 2: Distribution of the time difference between the digital inputs D0 and D7 for the three CRC configurations 8 bit,16 bit, disabled (8-byte payload, 750 transmissions per configuration).

A. ESB Parameter Analyses

1) Cyclic Redundancy Check Analysis: The CRC mode directly influences payload length and checksum computation time, thereby affecting overall transmission latency. Three CRC configurations are supported: 16 bit, 8 bit, and CRC disabled entierly. Figure 2 shows the total transmission time, measured as the time difference between D0 and D7. It defines the duration from issuing the transmission command on the APP-core of the TX to receiving the reception notification on the APP-core at the RX. All measurements were conducted with two retransmissions configured, generating three bulges in the violin plot. The first bulge, occurring at around 600 µs, corresponds to packets from the first transmission, where the upper two originate from retransmissions one and two. Table I shows that the overall number of transmitted and received packages remains relatively stable across all modes, with variations below 0.7% likely attributable to environmental factors. The table also provides a fine-grained analysis of duplicates and corrupted packages relative to the total number of transmitted packages. Disabling CRC yields the lowest latency, followed by 8 bit CRC, and then 16 bit CRC. This aligns with our expectations, as fewer bits are sent and less time is spent on checksum computation and verification. However, disabling CRC introduces data integrity issues. Without CRC, 735 packets were received, including one duplicated package and 15 corrupted packets, resulting in a total success rate of $\frac{735-15-1}{750} \times 100 = 95.87\%$. For our target of wireless time synchronization between nodes, the primary requirement is a successful payload reception, as packets are used for low-latency signaling rather than data transmission. Consequently, occasional payload corruption is acceptable in this application and therefore CRC is disabled for the OLCfg.

TABLE II: SUMMARY OF LATENCY MEASUREMENTS ACROSS ALL EVALUATED ESB CONFIGURATIONS.

Configuration	Mean ± SD [μs]	Median [μs]
CRC Mode		
16-bits CRC	642.87 ± 221.35	562.39
8-bits CRC	614.18 ± 166.12	558.39
CRC off	602.43 ± 160.14	554.30
Protocol Mode		
dynamic	590.82 ± 111.39	562.39
static	603.64 ± 146.58	563.10
Bitrate Mode		
2 Mbit/s (BLE)	594.45 ± 156.24	559.32
2 Mbit/s	607.45 ± 153.66	562.39
TX Modes		
automatic	570.21 ± 143.55	532.49
manual	574.19 ± 149.56	534.23
manual start	562.95 ± 128.19	534.23
TX Power Levels		
10 dBm	619.68 ± 202.84	548.67
5 dBm	626.90 ± 204.34	549.79
0 dBm	633.30 ± 212.35	552.25
-12 dBm	598.16 ± 164.95	548.87
-30 dBm	626.84 ± 212.04	550.66
-70 dBm	621.74 ± 203.19	550.00
Payload Construction		
standard mode†	692.96 ± 186.54	625.16
optimized mode‡	683.43 ± 177.47	617.89

All measurements are based on the timing difference between D0 and D7. For each config., 3–5 rounds were performed, yielding 450–750 data points.
† Payload constructed on APP-core and transferred via IPC to NET-core.
‡ Pre-constructed payload on NET-core ready for immediate transmission.

2) Protocol Mode: Table II shows a latency difference of 8.85 µs between protocol modes. While small, this difference is relevant for time-sensitive applications. Therefore, the dynamic mode is selected for the OLCfg.

3) Bitrate Mode: Our analysis focuses on the two 2 Mbit s^{-1} configurations, as this bitrate is significantly faster than 1 Mbit s^{-1} and more relevant for latency optimization. Table II shows that both configurations perform similarly, with the BLE-optimized mode achieving a slightly lower mean and median latency. This suggests that using 2 Mbit s^{-1} with BLE parameters is preferable for reducing overall latency.

TABLE III: SELECTED CONFIGURATION PARAMETERS FOR OPTIMAL LATENCY PERFORMANCE.

Parameter	Configuration
CRC Mode	disabled
Protocol Mode	dynamic
Bitrate Mode	2 Mbit s^{-1}
TX Mode	manual
TX Output Power	0 dBm
Packet Construction Mode†	optimized mode

† Optimized mode minimizes header overhead compared to the default packet structure.

TABLE IV: LATENCY MEASUREMENTS FOR TIME DIFFERENCES ACROSS MULTIPLE INPUT CHANNELS.

Time Intervals	Mean ± SD [µs]	Median [µs]
D0–D7	504.99 ± 96.89	486.30
D2–D5	311.78 ± 96.90	293.07
D3–D4	204.27 ± 96.91	185.86

4) TX Mode: Table II shows that the manual start TX mode achieves the lowest mean latency and SD, making it the preferred choice for minimizing overall latency.

5) TX Power Levels: Table II shows no clear trend in latency across TX power levels, likely due to the short 1 m test distance. To clarify the impact of this parameter, further evaluations over longer ranges are needed. To estimate the upper bound in latency, a TX power of 0 dBm was used as it produced the biggest median timing discrepancy.

B. Payload Construction

While not an ESB parameter, optimized payload generation can significantly reduce latency (see Section II-A). For a fair comparison, both the standard and optimized modes used a one-byte payload. Table II shows that the optimized implementation achieves slightly lower latency than the standard mode.

C. Optimal Latency Configuration (OLCfg)

After evaluating all parameters, the configurations yielding the shortest transmission latencies were identified (see Table III). In addition, a one-byte payload was selected for testing the OLCfg, both due to the use of the optimized payload construction mode and the reduced transmission delay associated with minimal payloads. Table II shows that the OLCfg achieves significantly lower latencies than any individual parameter variation tested.

The transmission latency from the TX APP-core to the RX APP-core (D0 - D7) is 504.99 µs, while the latency from the TX NET-core to the RX NET-core (D2 - D5) is lower at 311.78 µs. The NET-core-to-NET-core path is already fully optimized, leaving limited further improvement potential. In contrast, the higher APP-core to APP-core latency indicates that additional performance gains may be achievable through optimization of inter-core IPC communication.

IV. CONCLUSION

This paper presents an accurate latency evaluation of the ESB protocol for low-latency command broadcasting in multi-node systems, achieving a consistent transmission latency, measuring 311.78 µs for NET-core-to-NET-core and 504.99 µs for APP-core-to-APP-core transfers. In contrast, BLE shows higher and less predictable latency, as it depends on the connection interval (7.5 ms minimum) between exchanges. Transmission latency can range from nearly zero to an entire interval, without application-layer control of timing. However, with the ESB protocol, the RX must remain active and listening, trading deterministic latency for a significant decrease in energy efficiency. Our results demonstrate that ESB delivers deterministic, low-latency performance, making it well-suited for time-sensitive and performance-critical applications. Hardware constraints of the NRF5340 prevented the use of the fast ramp-up feature, which could further reduce the latency.

Future work will investigate fast-ramp-up using a capable NRF52 MCU, while also considering effects from different acquisition setups, such as body shadowing and external device interference.

ACKNOWLEDGMENT

This work was funded by the Swiss National Science Foundation (SNSF) under the projects “BodyLink” (Grant 220867), “SNOW” (Grant 209675), and the Ambizione Grant 233457.

REFERENCES

[1] S. Pastel, K. Petri, C.-H. Chen, A. Cáceres, M. Stirnatis, C. Nübel, and L. Schlotter, “Training in virtual reality enables learning of a complex sports movement,” *Virtual Reality*, vol. 27, p. 3, 07 2022, doi: 10.1007/s10055-022-00679-7.

[2] T. Swain, M. Mcnarry, A. Runacres, and K. Mackintosh, “The role of multi-sensor measurement to detect movement quality - a systematic review,” *Sports Medicine*, 09 2023, doi: 10.1007/s40279-023-01905-1.

[3] L. Yang, O. Amin, and B. Shihada, “Intelligent wearable systems: Opportunities and challenges in health and sports,” *ACM Comput. Surv.*, vol. 56, no. 7, Apr. 2024, doi: 10.1145/3648469.

[4] L. Schulthess, T. M. Ingolfsson, M. Nölke, M. Magno, L. Benini, and C. Leitner, “Skilog: A smart sensor system for performance analysis and biofeedback in ski jumping,” 2023-10, Conference Paper, doi: https://arxiv.org/abs/2309.14569.

[5] E. Wu, T. Matsumoto, C.-C. Liao, R. Liu, H. Katsuyama, Y. Inaba, N. Hakamada, Y. Yamamoto, Y. Ishige, and H. Koike, “Skitech: An alpine skiing and snowboarding dataset of 3d body pose, sole pressure, and electromyography,” in *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, ser. MMSports ’23, New York, NY, USA, 2023, p. 3–8, doi: 10.1145/3606038.3616151.

[6] D. Weber and N. Pfeiffer, “Methods for microsecond accuracy synchronization of wireless body area networks for biosignal acquisition using bluetooth low energy,” *Measurement*, vol. 253, p. 117635, 2025, doi: https://doi.org/10.1016/j.measurement.2025.117635.

[7] C. Y. Kim, J. Lee, E. Y. Jeong, Y. Jang, H. Kim, B. Choi, D. Han, Y. Oh, and J.-W. Jeong, “Wireless technologies for wearable electronics: a review,” *Advanced Electronic Materials*, p. 2400884, 2025.

[8] P. Bulic, A. Biasizzo, and M. Kljun, “Wake-up radio assisted bluetooth low energy for wireless sensor networks,” *Sensors*, vol. 17, no. 12, p. 2898, 2017, doi: 10.3390/s17122898.

[9] B. S. I. Group. (2025) How bluetooth® technology uses adaptive frequency hopping to overcome packet interference. Accessed: 2025-07-03. [Online]. Available: https://www.bluetooth.com.

[10] R. Ohara, S. Izumi, S. Imanaka, T. Yamamura, I. Toru, and H. Kawaguchi, “Microsync: Sub-micro second accuracy wireless time synchronization service,” *IEEE Access*, vol. 12, pp. 124 478–124 494, 2024, doi: 10.1109/ACCESS.2024.3446668.

[11] T. Instruments. (2025) Cc2540 and cc2541 bluetooth® low energy software developers guide. Accessed: 2025-07-03. [Online]. Available: https://www.ti.com/lit/ug/swru271i/swru271i.pdf.

[12] N. Semiconductor. (2025) Enhanced shockburst (esb). Accessed: 2025-06-01. [Online]. Available: https://docs.nordicsemi.com/bundle/ncs-latest/page/nrf/protocols/esb/index.html.