Evaluating ESP-NOW for transmitting sensor data between wearable devices

Michael Brilka and
Kristof Van Laerhoven

University of Siegen, 57076 Siegen, Germany

Abstract. This paper investigates the use of ESP-NOW for frequent event updates for wearable sensor systems. The proposed setup consists of multiple nodes that transmit sensor events to a central collector board, which stores the received data locally after adding its own data. To evaluate the communication performance, we first analyse how system performance changes with an increasing number of active nodes. We do two different versions: One with high overhead, by storing the received data directly towards the SD card, and one with low overhead, by forwarding the received data over a UART connection. We then investigate local data aggregation on the transmitting nodes as a strategy to improve communication efficiency and extend the number of supported nodes. Finally, power consumption is measured to assess the suitability of the system for wearable operation. The results show that reliable communication is possible. When the receive-function is not overly complex. Power measurements show that ESP-NOW communication uses 0.295W. Overall, ESP-NOW is a promising solution for decentralized sensing applications with moderate real-time requirements.

Keywords: Wireless Sensor Networks · Real-Time Communication · ESP-NOW.

1 Introduction

Wearable motion-sensing systems have become increasingly important in human-computer interaction, robotics, and virtual reality applications. Among these systems, sensor gloves are particularly useful because they can capture detailed hand and finger movements in real time and translate them into digital information. To achieve accurate motion tracking, many gloves rely on inertial measurement units (IMUs), which combine accelerometers, gyroscopes, and sometimes magnetometers to estimate orientation and movement. However, the usefulness of such systems depends not only on sensing accuracy but also on the reliability of data transmission between the controller and the finger-nodes.

Wired communication is commonly employed as a reliable, low latency method for signal transmission due to its stability and low susceptibility to interference. However, it is limited by reduced flexibility and increased system complexity when the number of connections grows, which is limited during the design phase.

In wearable systems, excessive cabling can restrict movement, reduce comfort, and complicate integration. Moreover, when transmission conditions exceed the limits of the physical link, signal degradation may occur, potentially leading to data corruption or loss.

These limitations are particularly relevant in IMU-based gloves ,therefore reducing the number of physical connections is desirable. This motivates the investigation of wireless alternatives for real-time sensor data transmission.

This paper examines the use of ESP-NOW as a wireless communication protocol for an IMU glove system. The proposed system aims to provide an efficient alternative to wired connections while maintaining sufficient reliability for wearable motion sensing. The objective of this work is to assess whether ESP-NOW can provide dependable data transfer for real-time motion sensing tasks. The communication is analysed with respect to its implementation feasibility and communication characteristics in the context of wearable sensor systems.

2 Related Work

Wireless transmission of sensor data is widely used in ubiquitous and embedded systems. While numerous wireless communication protocols exist, ESP-NOW has emerged as an attractive option due to its low overhead and straightforward integration into ESP32-based systems.

Pasic et al. provide an overview of the fundamental capabilities of ESP-NOW v1. Their work demonstrates reliable communication over distances of up to 190 m in open-field conditions. However, the study focuses primarily on communication range and reliability, and does not provide detailed information regarding achievable update rates for the transmitted temperature and humidity data [9].

Zuev et al. employ ESP-NOW to transmit status information and IMU data from multiple sensor nodes to a base station. To avoid transmission conflicts, each node is assigned a dedicated time slot, guaranteeing collision-free communication. Their results demonstrate the feasibility of coordinated multi-node communication using ESP-NOW [10].

Magzym et al. introduce a synchronized version of ESP-NOW designed to reduce energy consumption in wireless sensor networks. In their approach, multiple end devices transmit data according to a shared synchronization schedule. The proposed method achieves an energy reduction of approximately 97%. However, the reported update frequency is only 0.1 Hz, making the system unsuitable for applications requiring high-frequency data transmission [8].

Although previous studies demonstrate the suitability of ESP-NOW for long-range communication, multi-node networks, and energy-efficient sensor systems, there is limited experimental evaluation of its performance in high-frequency data transmission scenarios. In particular, the achievable update rates, latency, and reliability of ESP-NOW for transmission intervals below 50 ms remain insufficiently investigated.

3 Wireless Communication Selection

The ESP32 μM have build in wireless module. These supports BLE, Wi-Fi, and ESP-NOW. With this we already get a decent selection of capable communication standards to choose from. We compare selected metrics in Table 1.

For the application, the required data rate is modest. Assuming an interval of 50 Hz and a payload of 11 bytes (4 B timestamp, 6 B sensor data, 1 B checksum), we require a data rate of 26.4 kbps (6 nodes). This requirement can be fulfilled by all standards. As raw bandwidth is not the determine factor we need to lock at the communication architecture, scalability, and implementation effort.

While currently we only need to connect 6 nodes, future extensions may include additional modules.In this context, scalability is aconsideration. BLE supports a limited number of concurrent connections, which restricts expansions. These limitations are reflected again in Wi-Fi. ESP-NOW provides sufficient support for the current system and additional headroom for expansion.

Next we consider the connection model. BLE and Wi-Fi rely on connection oriented communication, which enables the detection of a disconnection. This behaviour can be advantageous for monitoring the link status, but it introduces additional connection-management complexity. ESP-NOW operates in a connectionless manner and does not maintain a persistent session state.

The reconnection process differs significantly across the three standards. In BLE, a disconnected peripheral must typically be re-established through the central device, which may require re-scanning and re-pairing.This can interrupt ongoing data collection and increase recovery time. Wi-Fi generally allows a device to rejoin the network independently, although network association and session recovery still remain necessary. In ESP-NOW a node can resume transmitting packets once it is operational again, without requiring registration.

The three standards differ in implementation effort. BLE requires the definition of services, characteristics, and GATT logic, in addition to connection and error handling. Wi-Fi necessitates network configuration, including station setup and communication management. ESP-NOW requires only initialization of the Wi-Fi interface and the ESP-NOW protocol, together with the peer MAC address, which results in a comparatively low implementation needs.

Taken together, ESP-NOW provides an advantage for use in decentralized wearable devices, when operating inside the limits, as seen in Table 1.

Feature	BLE [4] [3]	Wi-Fi [7][6]	ESP-NOW [5]
Bandwidth	0.7 - 1.4 Mbps	∼20-30 Mbps	∼1 Mbps
No. of nodes	9	10	20
Connection model	Connection oriented	Connection oriented	Connectionless
Overhead	GATT , pairing	TCP/ IP	Low
Notice disconnect	Supervision timeout	Link/session timeouts	None
Reconnection behaviour	Re-establishment	Reconnect	Immediate resend
Implementation difficulty	Moderate to high	Moderate	Low

Table 1. Comparison of key characteristics of BLE, Wi-Fi, and ESP-NOW.

4 ESP-NOW

ESP-NOW is a communication standard by Espressif, and published in August 2021 [2] [1]. ESP-NOW uses action frames defined in IEEE 802.11 (Fig. 1 Top), which operate in the data-link layer of the OSI model [5][1]. Optional encryption using the CCMP method from IEEE 802.11 is supported, but reduces the number of possible devices. This enables ESP-NOW to run alongside WiFi and BLE[5].

The ESP-NOW header is adjusted from IEEE 802.11 frames. In the Frame Control Field, Type is Management Frame and the Subtype to vendor specific. The bits for FromDS and ToDS are 0. Address 1 is the destination mac address, address 2 is senders mac address and address 3 is set to broadcast. The Sequence Control- and Duration-fields are not altered. Inside the Vendor specific content ESP-NOW frames (see Fig. 1 Bottom) are inserted[5].

In these frames many values are fixed. Element ID (221), Organizations Identifier (0x18fe34) and Version (set to ESP-NOW version). The length field is dependent on the body field, in which the actual data is being send. ESP-NOW V1 is capable of sending up to 250 bytes while V2 slightly modifies the frame (using the bit 4 of the reserved section to indicate more data) to send data over multiple frames, to achieve up to 1470 bytes in one ESP-NOW sending, receive operation. By using these specific action frames ESP-NOW does not need to be connected to an access point and can therefore be considered connectionless[5] .

5 System Overview

The system consists of two parts: a set of finger pcbs and a collector pcb. The finger pcbs are equipped with IMUs connected to an ESP μM, which periodically polls the sensors and transmits the acquired data via ESP-NOW. The collector pcb contains a similar IMU configuration and an SD card for local data storage. It receives the transmitted data and records it for subsequent analysis. Together, these two subsystems provide a compact wireless solution for capturing and storing motion data from the glove.

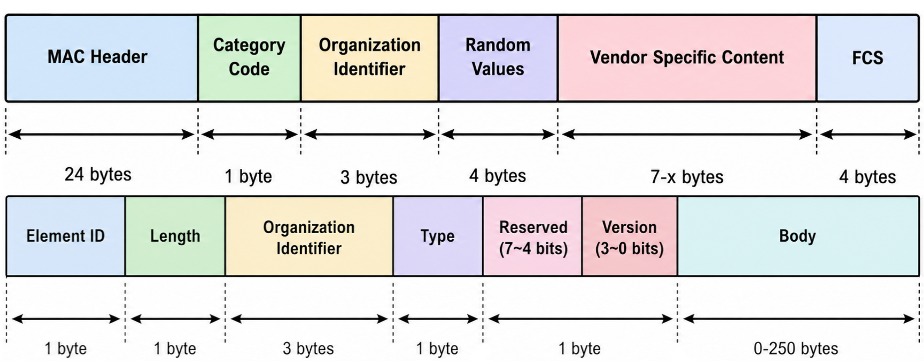


Fig. 1. (Top) IEEE 802.11 frame format; (Bottom) ESP-NOW V1 frame format

5.1 Hardware Components and Schematic

The platform is organized into two units: the finger pcbs and the collector pcb. Each unit integrates a BHI360 IMU with a BMM350 magnetometer and an ESP32-C3 μM. The collector pcb includes additionally a BME688 sensor and a SD card module. The BHI360 is selected due to its compact integration, onboard sensing capabilities, and suitability for wearable motion acquisition. For magnetic field measurements, a BMM350 is added. The BHI360 is connected to the ESP32-C3 via I2C.Communication between the finger pcbs and the collector pcb is implemented via ESP-NOW using a compact Wi-Fi antenna.The ESP32-C3 is selected for its small form factor.Its wireless connectivity, computational capability, and low-power operation speak in favour. To further simplify the overall system, a single shared battery is used for the current prototype. This approach reduces hardware complexity, lowering the physical bulk of the glove and reduces the likelihood of user errors.The use of a single power source allows the implementation of one charging and voltage-conversion circuit, simplifying the electrical design and system maintenance.

To streamline mass production, we place SMD components exclusively on the top layer while incorporating configuration pads on the bottom to define node roles and IDs via software logic. This layout eliminates the need for external connectors.Additionally, the design prioritized a minimal component count and footprint, resulting in optimized dimensions of 17.1 mm Œ 15.4 mm for the finger PCBs and 57.8 mm Œ 26.4 mm for the collector PCB. Full schematics are provided in Appendix A.1 and A.2.

5.2 Data Flow and Communication Pipeline

Each finger node acquires IMU measurements at a fixed interval and transmits the data to the collector node using ESP-NOW.The collector pcb receives the data and also records its own IMU measurements locally.To support temporal analysis, each IMU event is assigned the local timestamp of the respective node at the time of acquisition. This approach preserves timing information at the source and avoids the need for centralized clock synchronization. Before logging begins, the collector pcb waits until each node has transmitted at least one valid sensor event. Only after this initial synchronization condition is satisfied does the collector start writing incoming data to a output mediumOverall, the communication pipeline is designed to remain simple and efficient.

6 Evaluation

To assess the performance of the system, the evaluation is organized into four aspects: scalability, efficiency, performance, and power consumption. Scalability is examined to determine how the system behaves as the number of connected nodes increases. Data aggregation is analysed to evaluate communication characteristics. Performance to determine the suitability for distributed data collection. Power consumption is investigated to characterize the energy requirements

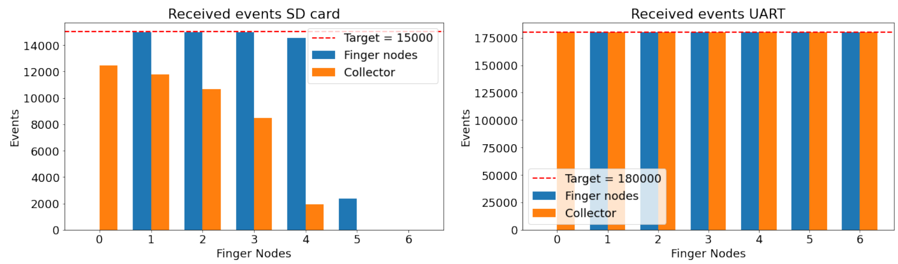


Fig. 2. (Left) Saved events when saving data directly towards SD card after reception. (Right) Forwarded events when directly sending data over UART after reception.

of the systemFor all experiments in this section, the actual sensor values are omitted and replaced by a timestamp (4 bytes) and an incremental message ID (3 bytes). This abstraction allows the communication pipeline to be evaluated independently of the sensing modality.

6.1 Number of nodes

Here we investigate how the system performance changes as the number of connected nodes increases. The experiment starts with no attached finger nodes, meaning that only the collector node writes data. The number of active nodes is then increased step wise.In addition to writing the received data directly to flash storage, we also performed a second experiment in which the data is transmitted via UART for output. With this we want to simulate a heavy processing task, and an easy processing task. We first did a short 5 minute run, which equals to 15000 captured events per node, to validate proper integration test. Afterwards if it is successfully we conducted a full 1 hour run, which equals to 300000 captured events per node. The corresponding results are shown in Fig. 2.

If we write the data constantly towards the flash (Fig. 2 Left), we can see a limitation. Beyond this point, a noticeable increase in missing transmissions occurs. If the number of nodes is increased further, the system reaches a point where no additional messages can be reliably revived as the writing task blocks the receiving process. On the other hand if we lower the overhead by forwarding the data (Fig. 2 Right), ESP-NOW is capable of transmitting approximately 300 messages of 7 bytes each per second, without significant packet loss.

6.2 Data aggregation

While the previous experiment showed that ESP-NOW is capable of sending hundreds of messages per second, it is also inefficient at that. With a small 7 byte message, we are sending 43 bytes of overhead. This brings the byte-efficiency to $(7/(43 + 7)) * 100 = 14\%$. This inefficiency can be mitigated through local data aggregation. By buffering sensor events before transmission, the system can send more data in a single packet. This should reduce the number of transmission initiations and therefore improve communication efficiency.

In our test no difference in the saved data can be seen, except that multiple entries are coming from the same finger node at once. It should make the communication more efficient. By using up to 250 bytes in the body, we increase the communication efficiency to $((35 * 7)/(43 + (35 * 7))) * 100 = 85.06\%$. While this is expected, we also increase the penalty when we miss a communication event.

6.3 Timing Performance

To ensure the system communication is useful for on device ML, two key metrics can be used: message delay and clock drift. Message delay provides insights into the responsiveness and latency of network communications, which is critical for real-time applications and user experience. Meanwhile, clock drift highlights discrepancies between a systems internal clock and an external time source.

To analyse message delay we wrote a small program to measure the round trip time (RTT) between just two devices, and output the result with UART. We let the program run for 1 hour. The result of this experiment is that in 99.94% of the time the RTT is in the range between 2 and 3ms. In 7.3 ppm of cases the RTT is above 10ms. The jitter is low with just 0.5ms.

Measuring the clock drift in our data we see that the clocks of the finger nodes drifted about 100 ms during a runtime of 1.5 hours. This equals a frequency error of 18.5ppm. Which is outside the stated rate of the quartz (10ppm). We suspect that the waste heat, of the ESP32-C3 distributes to the quartz. Load capacitance likely increased the discrepancy as well.

6.4 Power consumption

While performing the experiments, power consumption is simultaneously measured. To isolate the energy cost of ESP-NOW from the sensor subsystem, the DC/DC converter is removed. Power consumption is measured by inserting a 0.5Ω resistor on the input side and recording the voltage drop across it, using a Picoscope 5244B, every ms. The results can be seen in Fig. 3 and Fig. 4.

We observed that each node adds 0.085A * 5V = 0.425W power consumption. This means that we need a 800 mAh LiPo battery to power the glove for one hour. We also want to understand the sources of energy consumption, we therefore need to identify the contribution of each component. With the μM running an empty program, the power consumption is approximately 0.10 W. Running the RTOS increases the power consumption by 0.03 W. Enabling the Wi-Fi module and connecting it to a node, increases the power consumption to 0.28 W. Therefore, the Wi-Fi module account for an additional 0.15 W. Comparing this value to the total power consumption of 0.425 W, the remaining 0.145 W can be attributed to data transmission and other node-related activities.

For the data aggregation we did not see improvement in power consumption (see Fig. 4). It could enable grater power savings by turning off the Wi-Fi module, but this comes with startup- and reconnection-times ($\approx$ 75 ms).

7 Discussion

ESP-NOW can be used to collect real-time sensor data from multiple nodes, but its advantages must be considered together with its limitations. A practical limitation is the amount of processing that can be done on the collector node. The collector node benefits from using a more powerful ESP32 μM, since data reception and data processing needs to be separated to avoid timing/ processing conflicts.

Another drawback is the comparatively high energy cost at short transmission distances. However, if a limited battery lifetime is acceptable, this limitation can be ignored. Unfortunately aggregating data beforehand did not reduce power consumption.

Overall, when reliable timing is critical, when many additional nodes need to be integrated and/or when power efficiency is highly important, a wired connection might be the better option. In these cases, the determinism is better suited. By contrast, for bigger decentralized wearable systems, where moderate real-time performance is acceptable and the nodes are intended to be further away, ESP-NOW could be a promising solution.

8 Conclusion

We investigated the use of ESP-NOW for decentralized, real-time sensor data collection in a wearable multi-node system. The results demonstrate that ESP-NOW enables communication between distributed finger nodes and a central collector node at an update rate of 50Hz, making it a promising solution for wireless data exchange. However, the overall performance is limited by the additional processing tasks performed by the collector node. Reliable transmission therefore depends strongly on how quickly the receiving task can process incoming data. As shown by our results, forwarding the data achieves better performance than saving it to a SD card.

We further investigated the origin of the 0.425W energy requirement and have seen that most of it comes from the enabling and usage of the network interface. Furthermore, we evaluated network performance and showed that the Round Trip Time lies between 2 to 3 ms with a jitter of 0.5 ms.

Overall, ESP-NOW proved to be a promising solution for decentralized wearable sensing systems with moderate timing requirements and limited software complexity. For applications that require strict determinism, continuous real-time communication, or a larger number of synchronized data sources, a wired connection may still be the more robust alternative.

The hardware and software is publicly available under the following link `https://github.com/Michael-Brilka/ESP-Glove`

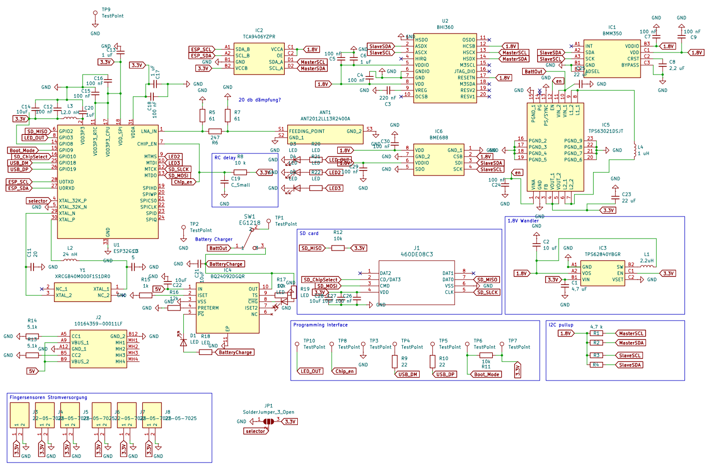
Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Project-ID 262513311 SFB 1187.

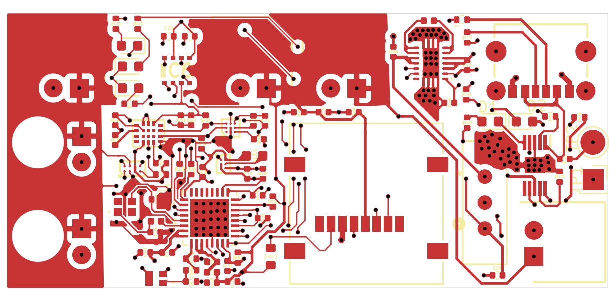
References

1. Esp-now a wireless communication protocol for quick responses and low-power control (-), https://www.espressif.com/en/solutions/low-power-solutions/esp-now
2. esp-now (2021), https://github.com/espressif/esp-now
3. Bluetooth LE & bluetooth (2026), https://docs.espressif.com/projects/esp-faq/en/latest/software-framework/bt/ble.html
4. Bluetooth low energy (2026), https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/api-guides/ble/index.html
5. ESP-NOW (2026), https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html
6. ESP-WIFI-MESH programming guide (2026), https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp-wifi-mesh.html
7. Wi-fi driver (2026), https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi-driver/index.html
8. Magzym, Y., Eduard, A., Urazayev, D., Fafoutis, X., Zorbas, D.: Synchronized ESP-NOW for improved energy efficiency. In: 2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). pp. 57–62. IEEE (2023)
9. Pasic, R., Kuzmanov, I., Atanasovski, K.: ESP-NOW communication protocol with ESP32. Izzivi prihodnost **6** (03 2021). https://doi.org/10.37886/ip.2021.019
10. Zuev, A., Karaman, D., Olshevskiy, A., Jarmolovičius, M.: Algorithmic design and software for a microcontroller-based wearable biomedical sensor monitoring system via esp-now protocol. Advanced Information Systems **9**(4), 51–56 (2025)

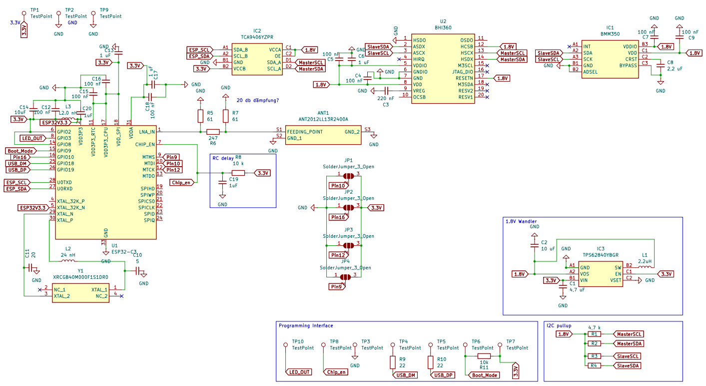
A Schematic and PCB-Top for the glove

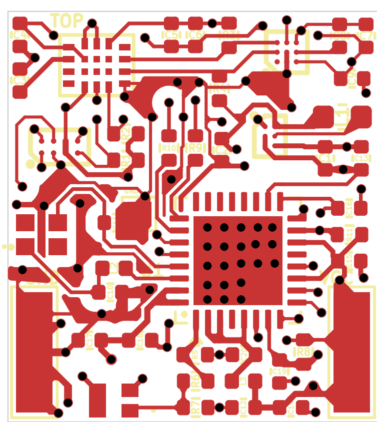
A.1 Collector





A.2 Finger nodes





B Power consumption graph

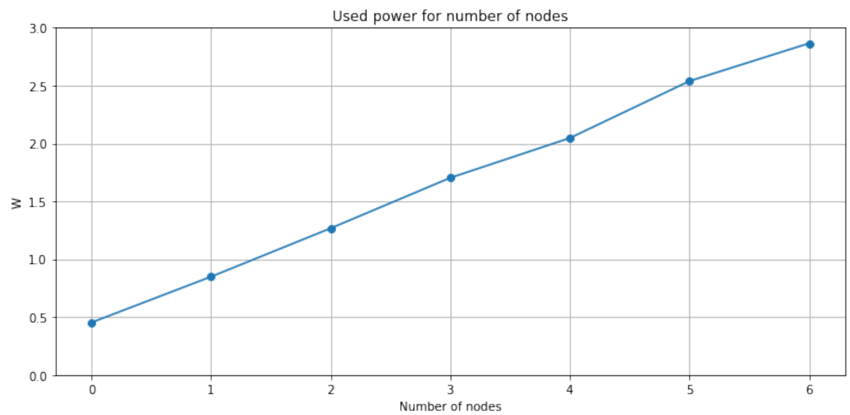


Fig. 3. Power consumption with an increasing number of finger nodes

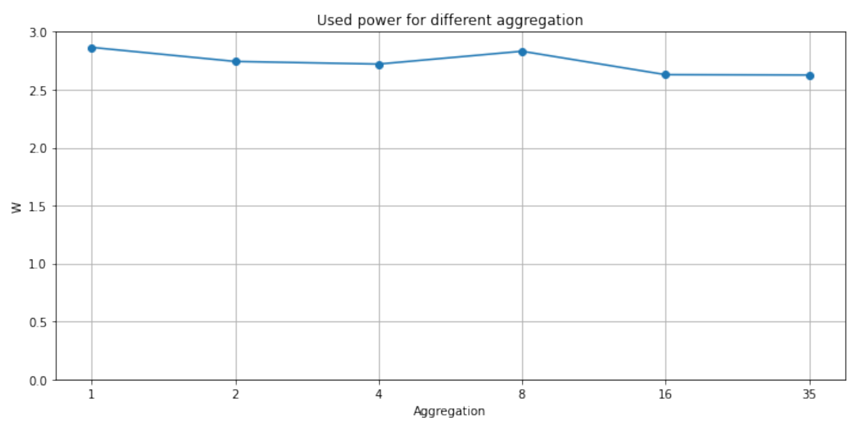


Fig. 4. Power consumption of aggregated data.