SensingWall: Ultra-low Cost WiFi Wireless Sensing

Anatolij Zubow, Muhammad Elhwawshy, Sascha Rösler, Lorenz Pusch, Adam Wolisz and Falko Dressler
School of Electrical Engineering and Computer Science, TU Berlin, Germany
{zubow, roesler, lorenz.pusch, wolisz, dressler}@ccs-labs.org

***Abstract*—Future wireless sensing applications heavily depend on the analysis of wide-band channel state information (CSI) acquired from extremely dense device deployments. However, a gap exists in practical evaluations within existing literature. To bridge this gap, we introduce SensingWall, a highly adaptable, cost-effective solution leveraging IEEE 802.11 and the readily available ESP32 platform. SensingWall simplifies the process for application designers by abstracting away intricate technical details, offering visualization tools, and enabling direct access to a centralized database housing real-time CSI data. Our demonstration includes a functional prototype showcasing two key features. Firstly, real-time visualization of CSI time-series data from both uplink and downlink. Secondly, a demonstration of custom CSI retrieval through SQL-like query operations.**

I. Introduction

Wireless sensing using channel state information (CSI) involves the extraction of useful information from the wireless channel itself, like changes in signal characteristics due to alterations in the environment. CSI provides insights into the wireless medium which goes beyond the traditional communication purposes and can be used for various applications, including localization, gesture recognition, activity detection, human sensing, and material identification [1], [2]. The ability to exploit CSI for sensing purposes relies on advanced signal processing techniques analyzing changes in signal strength, phase, and other channel parameters to derive meaningful information about the environment. Here, machine learning algorithms often play an important role in extracting patterns from the data.

However, there are several challenges associated with wireless sensing [1]. First, analyzing CSI requires advanced signal processing techniques, and interpreting this data accurately can be complex requiring expert knowledge. Second, some applications, like gesture recognition or activity detection, require real-time processing of CSI data from a large number of distributed devices, which can be demanding in wireless environments. Third, wireless devices often have limited computational power and energy resources, impacting the complexity of sensing algorithms they can run. Forth, wireless channels are prone to noise, interference, and multipath effects, making it necessary to test algorithms under real conditions.

We believe that there is a need for a scalable, ultra-low cost and easy-to-use wireless sensing platform. The user of such a platform should be able to focus completely on algorithmic design and application and not on low-level operations like CSI collection, cleansing, transfer, storage and retrieval. Furthermore, it should be easy to implement collaborative approaches where sensing accuracy is enhanced by combining the CSI data from multiple devices.

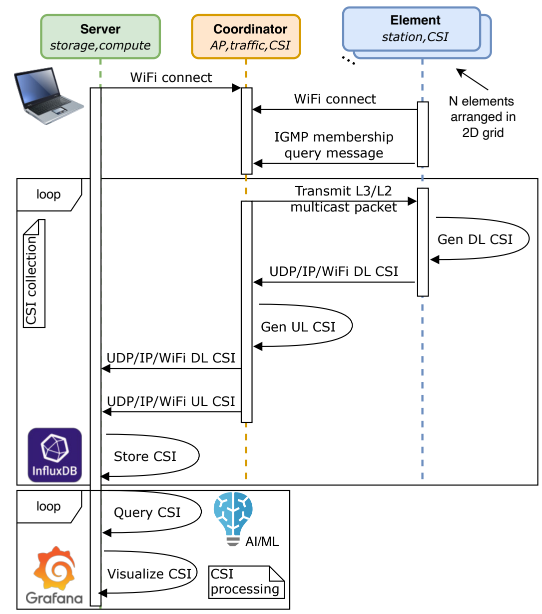

Figure 1. Interaction between SensingWall components

II. SensingWall Concept

We present our sensing platform termed `SensingWall`, which was designed with the following objectives: i) ultra-low cost, i.e., usage of commodity & 3D printed parts, ii) suitable for dense deployments, i.e., possibility to have 10-100 of sensing devices within a small space, iii) open hardware and software, iv) full wide-band CSI, i.e., amplitude and phase per OFDM subcarrier, v) easy to use for non-wireless experts, i.e., CSI as time-series data, vi) easy to extend, and vii) CSI data query and visualization in real-time. The `SensingWall` consists of the following three components:

- *Server*: a powerful computer, e.g., a modern laptop, which is used for storage of CSI data in a database optimized for time-series data (InfluxDB) with tools for visualization (Grafana) and processing (Python, machine learning frameworks),
- *Coordinator*: a single ESP32 SoC node[1] operating in Access Point (AP) mode for bridging between the *Server* node and the N *Element* nodes (note: ESP32 operating in softAP mode supports a maximum of $N = 16$ stations,

[1] https://www.espressif.com/en/products/modules/esp32

for larger N an ordinary AP can be used, however, the uplink (UL) CSI data will not be available),

- *Element*: N ESP32 nodes arranged in a 2D grid (vertical/horizontal spacing of 6 cm) operating in WiFi station mode and connected to the *Coordinator* node (Figure 2).

All ESP32 nodes, i.e., *Coordinator* and the N *Element* nodes, do not require any wired communication backhaul as all data (CSI) is exchanged over the wireless interface. The USB port is only used for power supply. As the cost of the `SensingWall` (server excluded) is only around 10 \$ per element, even large configurations can be set up inexpensively. Furthermore, collaborative approaches, i.e., multiple `SensingWall` installations, can be implemented cheaply.

III. SensingWall Details

The interaction between the three components is depicted in Figure 1. In order to support a large number of *Element* nodes we use Layer-3/2 multicasting in the downlink (DL) where the CSI towards all *Element* nodes can be obtained efficiently from a single WiFi frame transmission. The UL CSI data is estimated from the UL frames which contain the DL CSI data and are sent by the *Element* nodes. On the *Server* side we use Python multiprocessing which enables scalable CSI data reception and storage into the database. For our prototype we used an Intel Xeon from 2018 as *Server* node with a 3×3 configuration for the `SensingWall`, i.e., $N = 9$ *Element* nodes, allowing us to store the stream of CSI data at maximum sampling rate of 100 Hz, i.e., CSI is determined every 10 ms, in real-time. Here the required storage in the database is ≈1 GByte/hour.

The entirety of stored CSI data is accessible for queries through InfluxQL, an SQL-like language. Real-time visualization can be achieved using the Grafana tool or accessed directly via Python. The software installation is fully automated, i.e., InfluxDB, Grafana and the ESP toolchain run in a docker container. The holder for the *Element* nodes is 3D printed using ABS plastic (Figure 2). It is easy to extend beyond the current 4×4 configuration as our Blender model is freely available. Currently we use ESP32s3 nodes operating in 2.4 GHz ISM spectrum and supporting a bandwidth of 40 MHz. For CSI tracing we extended the tool from Hernandez et al. [3]. As future work we plan to move to ESP32s5 platform which would support the much larger 5 GHz spectrum and larger channel bandwidth. Our software and the hardware description are freely available for download.[2]

IV. Demonstration

During the demo we will visualize the CSI data from DL and UL in real-time using the Grafana tool (Figure 3). Specifically, we will change the environment, e.g., rotating the wall and/or moving the *Coordinator* node, to show in real-time the impact on the CSI of different *Element* nodes. During the demo we also plan to showcase the use of InfluxQL queries for retrieval of CSI data from the database. As an example the following query returns from the last 100 s the amplitude of subcarrier 41 of the specified DL link averaged over a window of 1 s:

[2]https://github.com/zubow/sensingwall

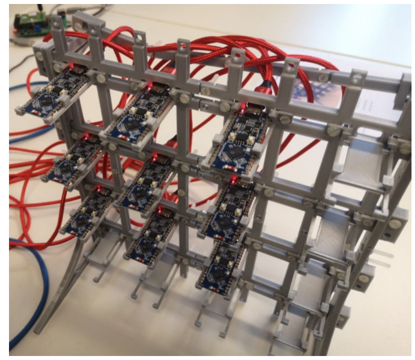

Figure 2. `SensingWall` prototype in 3×3 configuration (*Element* nodes)

Figure 3. Screenshot of `SensingWall` dashboard in Grafana

```
from(bucket: "csi")
  |> range(start: -100s)
  |> filter(fn: (r) => r["_measurement"] == "csi"
  and r["_field"] == "ampl"
  and r.source == "48:27:e2:3b:33:2d"
  and r.destination == "34:85:18:b9:1b:8c"
  and r["subcarrier"] == "41")
  |> map(fn: (r) => ({_value: r._value, _time: r._time}))
  |> aggregateWindow(every:1s, fn:mean, createEmpty:false)
  |> yield(name: "downlink0")
```

V. Conclusion

Through this demo, we have introduced an open, low-cost platform for wireless sensing utilizing commonly available WiFi hardware. This platform facilitates real-time access to CSI time-series data via a user-friendly database interface, empowering developers to focus on improving their sensing algorithms without the burden of intricate hardware access.

Acknowledgments

This work was supported by the Federal Ministry of Education and Research (BMBF, Germany) within the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K as well as by the German Research Foundation (DFG) within the project ML4WiFi under grant DR 639/28-1.

References

[1] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–36, 2019.

[2] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang, "Wireless Sensing for Human Activity: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1629–1645, Jul. 2020.

[3] S. M. Hernandez and E. Bulut, "WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 46–76, Jan. 2023.