Demo: OpenCSI: A Self-Calibrating, Cross-Hardware WiFi Sensing Mesh

Karim Khamaisi, Simon Sigg, Bruno Rodrigues
Sensing Group SG, Institute of Computer Science in Vorarlberg ICV, University of St. Gallen HSG
Hintere Achmühlerstraße 1, 6850 Dornbirn, Austria
E-mail: {karim.khamaisi, bruno.rodrigues}@unisg.ch, simon.sigg@student.unisg.ch

***Abstract*—WiFi Channel State Information (CSI) enables privacy-preserving, camera-free occupancy detection on commodity hardware as cheap as a 3 USD ESP32 microcontroller. A model trained on one deployment, however, rarely survives a move to a different room or a different chip generation, because standard normalization embeds chip- and room-specific artifacts into the sensing features. This demonstration deploys *OpenCSI*, a self-calibration layer that exposes each mesh link as a dimensionless Z-score learned online from a per-link quiet-period baseline. Attendees interact with a live four-node ESP32 mesh and a real-time dashboard that shows the mesh topology, each link's calibration maturity as its baseline forms, and the occupancy decision as a visitor steps into the sensing volume. A classifier trained beforehand on an ESP32-C6 mesh then continues to detect occupancy on a physically different ESP32-S3 mesh in the same room, without retraining and without target-domain data. Following the full paper, the demonstration showcases the OpenCSI calibration layer itself rather than a specific sensing task.**

***Index Terms*—WiFi CSI, calibration, temporal normalization, occupancy sensing, zero-shot transfer, ESP32**

I. Introduction

Indoor occupancy and presence detection are the foundations of energy-efficient buildings, which draw a large share of global energy and where demand-controlled ventilation alone can substantially reduce consumption once accurate occupancy is available [3], [1]. The common sensors are unsatisfying: cameras break privacy, passive-infrared sensors miss seated occupants, and wearables require compliance [1]. WiFi Channel State Information (CSI), the complex response per subcarrier an OFDM receiver already extracts from every frame, instead senses presence and motion using radios already installed for communication, with no cameras or wearables [8], [10], and low-cost ESP32 microcontrollers expose CSI directly from commodity 802.11 frames [4], so a dense sensing mesh is inexpensive to build.

The barrier to deployment is not accuracy, but portability, and its root is calibration. A CSI model trained in one room rarely transfers to another, to a different node batch, or even across a single power cycle of the same node, a loss of generalization that remains a recognized, unsolved obstacle across the field [13], [2]. Standard pipelines normalize each recording against its own statistics, so per-boot phase, gain-control state, and chip-specific gain leak into the features, and a model cannot separate hardware from channel. The cost compounds with scale, because a mesh of N nodes exposes $N(N-1)$ directed links that each carry their own calibration state, so per-deployment recalibration is precisely what has kept WiFi sensing from scaling beyond single-room demonstrations. Making that calibration automatic and observable while it runs is the problem this demonstration puts on display.

OpenCSI removes that coupling by exposing each link as a single dimensionless signal. While the room is empty, every link learns its own quiet-period mean and *temporal* standard deviation through a Welford accumulator [9], and each subsequent frame becomes a Z-score $z = |\bar{A} - \mu|/\sigma$. Because chip gain and room geometry scale numerator and denominator alike, the ratio largely cancels them, so a deviation carries a comparable meaning across the chips, rooms, and reboots tested. The full paper establishes this quantitatively and was accepted at LCN. The present demonstration makes it visible and interactive rather than re-arguing the measurements [7].

The demonstration showcases two properties of OpenCSI itself rather than a specific sensing application (Fig. 1): the per-link temporal normalization that lets a model transfer across hardware without retraining, and the calibration-maturity signal that makes drift observable while the system runs. Unlike WiFi-sensing demonstrations that show a downstream task such as gesture or activity recognition on fixed hardware, this booth exposes the calibration layer beneath them, and to the authors' knowledge is the first to make cross-hardware CSI transfer observable through a live physical swap rather than reported metrics.

II. The OpenCSI System

OpenCSI sits between the radio hardware and the sensing logic, and emits one normalized signal per link through three phases: acquire, calibrate, score. A hardware abstraction layer parses the chip-specific packet formats (an ESP32-S3 delivers 52 subcarriers over 802.11n HT20, an ESP32-C6 delivers 242 over 802.11ax HE20) into a canonical packet, so everything downstream is hardware-agnostic. Sanitization then inverts the per-packet gain, removes a linear phase trend, and discards outlier frames.

The calibrated amplitude Z-score for link $i \to j$ is

$$z_{ij}(t) = \frac{\left|\bar{A}_{ij}(t) - \mu_{ij}\right|}{\sigma_{ij}}, \tag{1}$$

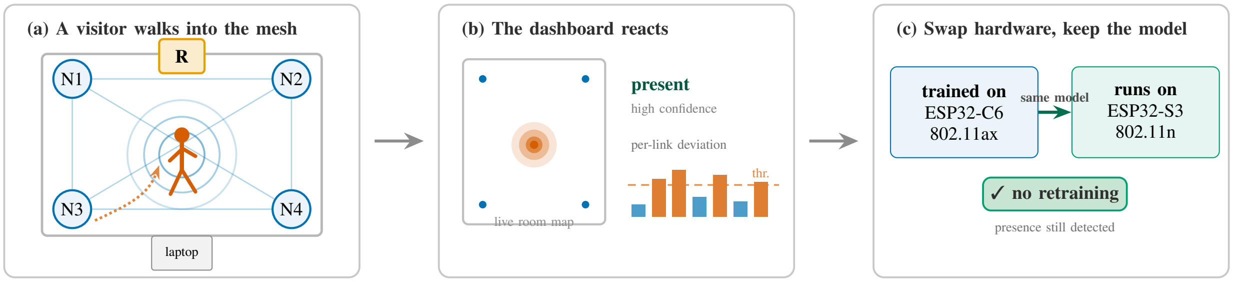

Fig. 1: What attendees experience at the booth. **(a)** A visitor walks into a live four-node ESP32 mesh (a WiFi router and a laptop backend complete the setup), perturbing the links around them. **(b)** The real-time dashboard reacts: the per-link deviations against each link's calibrated quiet-period baseline rise above threshold, and the room is marked *present*. **(c)** The mesh hardware is then physically swapped from one ESP32 generation to another, and the same model, unchanged, still detects the visitor.

where $\bar{A}_{ij}(t)$ is the mean amplitude across subcarriers and (μ_{ij}, σ_{ij}) are the mean of the quiet-period of the link and the temporal standard deviation. A quiet-period gate combines temporal variance with spectral entropy, so periodic interference such as HVAC does not corrupt the baseline, and each admitted frame updates a Welford accumulator [9] that maintains (μ_{ij}, σ_{ij}) without storing raw samples. Because chip gain and room geometry enter numerator and denominator multiplicatively, they largely cancel. Reducing each link to this single scalar also makes the feature independent of the subcarrier count, so across the ESP32 chips tested, the Z-score behaves consistently despite differences in band, PHY format, and subcarrier count.

Each link also carries a reliability score $r_{ij} \in [0, 1]$ that decays with the age of its baseline and saturates once enough quiet samples accumulate. The frame-level *calibration maturity*, their reliability-weighted mean, plateaus after about five minutes of quiet observation. The demo dashboard visualizes this signal as the baseline forms, and a sudden drop flags a stale link, so the mesh abstains rather than fails silently.

In the backing study [7], a mesh deployed across three rooms and three ESP32 variants (S3, C3, C6) detects occupancy with a lightweight random-forest classifier over the calibrated Z-scores. Because the Z-scores are largely hardware-invariant, a classifier trained on one chip generation keeps working when the mesh is rebuilt with another, with no retraining and no labeled target-domain data, where standard per-session normalization does not transfer. Unlike model-side approaches that need target-domain data (domain adaptation and meta-learning [6], [12], foundation models [5]) or algebraic cancellation that needs a co-located antenna pair (CSI-ratio [11]), OpenCSI calibrates each link statistically from its own quiet window, which is what lets a model transfer across hardware. This demonstration reproduces that behavior live. OpenCSI is released as open source under Apache 2.0 with the full evaluation dataset.[1]

III. The Demonstration

The demonstration runs on live CSI from the booth, and the dashboard in Fig. 2 is the interface attendees observe as they move through the mesh.

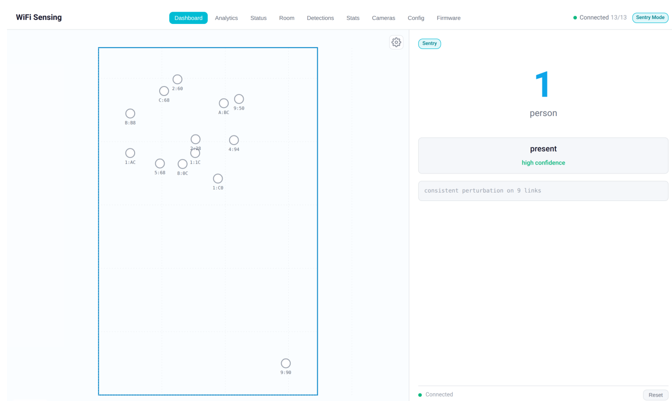

Fig. 2: The live *WiFi Sensing* dashboard that runs at the booth, driven by OpenCSI's per-link calibration (shown with illustrative data; the booth mesh uses four nodes). The room map (left) shows the connected mesh, and the panel (right) reports presence together with the per-link evidence behind it, here a consistent deviation on nine links against their calibrated quiet-period baselines.

A. Setup

This demonstration uses a self-contained, portable subset of the evaluation testbed: four ESP32 nodes at the corners of a roughly $2\,\mathrm{m} \times 2\,\mathrm{m}$ area, one commodity WiFi router, and a laptop running the OpenCSI backend and dashboard (Fig. 1a). The four nodes expose $4 \times 3 = 12$ directed links, enough for a clear topology view while staying transportable. Nodes stream raw CSI to the laptop over UDP through the router, and a second set of nodes of a *different* ESP32 generation sits ready for the swap. The dashboard is a browser interface that shows, in real time, the connected mesh on a room map, the per-link deviations against each link's calibrated quiet-period baseline, and the resulting presence decision.

B. What Attendees Will Experience

The mesh is calibrated in the empty booth before the session and its per-link baselines are checkpointed to disk, so the booth is responsive from the start and can be reset in seconds if the floor disturbs it (Sec. IV). Visitors move through four short stages, each needing no explanation beyond the dashboard:

[1] https://github.com/sensing-group/OpenCSI

1) **Watching calibration form:** with the room empty the operator runs *Calibrate Baseline*, and the dashboard learns each link's quiet-period reference.
2) **Stepping into the mesh:** a visitor enters and the dashboard reports *present* within about a second as the per-link deviations rise, returning to empty when they leave.
3) **The zero-shot hardware swap** (Fig. 1c)**:** the mesh is physically rebuilt from ESP32-C6 to ESP32-S3, and the same unchanged model keeps detecting occupancy.
4) **Breaking and recovering calibration:** a node is unplugged or a chair moved, and the dashboard flags the stale links, which recover after about $30\,\mathrm{s}$.

The first two stages make the calibration layer visible. Running the baseline calibration with the room empty and watching the per-link references settle shows what the system does before it can sense. When a visitor steps in, the dashboard shows the per-link deviations behind the decision, so presence can be related to the calibrated signal rather than treated as a black box.

The hardware swap is the centerpiece, and two details keep it honest. The swapped-in S3 nodes carry their own quiet-period baseline, checkpointed from an earlier empty-room calibration at those positions, an unsupervised baseline rather than labelled target data, so only the classifier is held fixed. The S3 hardware speaks 802.11n HT20 rather than the 802.11ax HE20 the model was trained on and reports 52 subcarriers instead of 246, yet detection continues, and the dashboard shows the training and deployment hardware (model: C6, mesh: S3).

In the final stage, unplugging a node or moving furniture drops the affected links' maturity, so the mesh reports its own degradation rather than emit silent errors, and only the disturbed links re-mature, so detection returns in about $30\,\mathrm{s}$ rather than the few minutes a cold start needs.

IV. Requirements and Logistics

The demonstration is self-contained: it comprises both ESP32 node sets (S3 and C6) on small stands, the router, the laptop backend, a power strip, and all cabling. The venue needs to provide only a table of about $1.5\,\mathrm{m}$, one power outlet, and roughly $2\,\mathrm{m}\times 2\,\mathrm{m}$ of adjacent floor space for visitors to step into the mesh, with cabling routed to the perimeter to keep the walk-in area clear. A single marked step-in spot lets the presenter admit one visitor at a time so the empty-to-occupied transition stays legible, and the hardware swap is performed on request a few times per session rather than per visitor. The mesh runs on its own router, so it needs neither conference WiFi nor internet, and the least-congested $2.4\,\mathrm{GHz}$ channel is selected on site.

Two properties make the booth robust to a busy conference floor. Because the quiet-period baseline is learned in place, it absorbs the local interference floor rather than assuming a clean channel. Since that baseline is checkpointed before doors open, it can be reloaded in a few seconds if constant foot traffic disturbs it, so the booth does not depend on finding a quiet window mid-session. A short recorded walkthrough serves as a fallback, and one presenter continuously runs the interaction loop alongside a poster. Setup takes about fifteen minutes.

V. Conclusion

OpenCSI exposes each mesh link as a dimensionless Z-score against its own quiet-period baseline, so the sensing layer above need not model chip- or room-specific physics. This demonstration lets attendees observe that abstraction in operation: a mesh calibrating itself, an occupancy decision responding to their presence, and a trained model continuing to work after the sensing hardware is swapped between ESP32 generations. The software and data are open source for reproduction on commodity hardware.

Acknowledgment

AI tools assisted with language and style only. The conceptualization, design, and interpretation are the authors'.

References

[1] J. Ahmad, H. Larijani, R. Emmanuel, M. Mannion, and A. Javed, "Occupancy detection in non-residential buildings: A survey and novel privacy preserved occupancy monitoring solution," *Applied Computing and Informatics*, Vol. 17, No. 2, pp. 279–295, 2021.

[2] Z. Chen, L. Zhang, C. Jiang, J. Cao, and W. Cui, "Cross-domain WiFi sensing with channel state information: A survey," *ACM Computing Surveys*, Vol. 56, No. 1, pp. 1–35, 2023.

[3] B. Dong, V. Prakash, F. Feng, and Z. O'Neill, "A review of smart building sensing system for better indoor environment control," *Energy and Buildings*, Vol. 199, pp. 29–46, 2019.

[4] Espressif Systems, "ESP-CSI: ESP32 Wi-Fi channel state information," https://github.com/espressif/esp-csi, 2023.

[5] C. Jiang, Y. Yan, Y. Wang, C. T. Chou, and W. Hu, "Scale what counts, mask what matters: Evaluating foundation models for zero-shot cross-domain Wi-Fi sensing," *arXiv preprint arXiv:2511.18792*, 2025.

[6] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proc. 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2018, pp. 289–304.

[7] K. Khamaisi, S. Sigg, and B. Rodrigues, "OpenCSI: Self-calibration layer for heterogeneous mesh wireless sensor networks," in *Proc. IEEE 51st Conf. on Local Computer Networks (LCN)*, 2026, software and dataset: https://github.com/sensing-group/OpenCSI.

[8] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys*, Vol. 52, No. 3, pp. 1–36, 2019.

[9] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, Vol. 4, No. 3, pp. 419–420, 1962.

[10] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, Vol. 55, No. 10, pp. 98–104, 2017.

[11] D. Zhang, Y. Zeng, D. Wu, and J. Xiong, "Boosting WiFi sensing performance via CSI ratio," *IEEE Pervasive Computing*, Vol. 20, No. 1, pp. 62–70, 2021.

[12] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "CrossSense: Towards cross-site and large-scale WiFi sensing," in *Proc. 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2018, pp. 305–320.

[13] G. Zhu, Y. Hu, W. Gao, W.-H. Wang, B. Wang, and K. J. R. Liu, "CSI-Bench: A large-scale in-the-wild dataset for multi-task WiFi sensing," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025, arXiv:2505.21866.