Wireless Connectivity Testing with Mobile Robots: A Bluetooth Low Energy Case Study

Fernanda Neves
Federal University of Pernambuco
Recife, Brazil
fmn@cin.ufpe.br

Beatriz Barros
Federal University of Pernambuco
Recife, Brazil
bob@cin.ufpe.br

Adrien Durand-Petiteville
Federal University of Pernambuco
Recife, Brazil
adrien@cin.ufpe.br

Breno Miranda
Federal University of Pernambuco
Recife, Brazil
bafm@cin.ufpe.br

Abstract

The growing adoption of wireless technologies in mobile applications demands reliable and reproducible methods for evaluating connectivity behavior. This work presents a robot-assisted approach for evaluating Bluetooth Low Energy (BLE) signal quality, in which a mobile robot carries one smartphone while a second device remains fixed at a known position in a controlled indoor environment. During each run, wireless signal metrics are collected alongside navigation data from the robot system, allowing signal degradation to be characterized as a function of distance until the connection drops. The proposed approach contributes to software engineering by demonstrating how mobile robots can support reproducible, instrumented testing of wireless connectivity in mobile and cyber-physical systems. Results indicate that the setup consistently captures signal degradation and connection loss events, supporting its use as a practical tool for evaluating wireless connectivity behavior across different routes, devices, and technologies.

Keywords

Mobile Robotics, Wireless Connectivity Testing, Bluetooth Low Energy, Test Automation

1 Introduction

Many modern software systems rely on wireless connectivity to operate correctly, including mobile applications, IoT devices, and cyber-physical systems [1]. This makes connectivity reliability a critical non-functional property: failures can degrade user experience, compromise safety in cyber-physical systems, or render an application entirely non-functional in real-world deployment.

Testing such systems requires evaluating their behavior under varying physical conditions, typically by having a human tester carry a device along a predefined path, a process that is challenging, time-consuming, and difficult to control [12]. Factors such as walking speed and body shadowing (the attenuation caused by the human body between devices) have been shown to affect Received Signal Strength Indicator (RSSI) measurements [5]. As a result, such tests are difficult to reproduce, a fundamental requirement for reliable empirical evaluation in software engineering [9].

Although human-carried tests are commonly used in practice, little empirical evidence exists comparing their reproducibility against robot-assisted executions under the same testing conditions. This paper investigates whether an autonomous mobile robot can serve as a controlled physical execution platform for Bluetooth Low Energy (BLE) connectivity testing. By coupling robot navigation with synchronized signal collection, we explore whether robotic execution produces more reproducible measurements than human-carried approaches. To this end, we conducted a case study in which a mobile robot carrying a BLE-enabled device traversed a predefined indoor trajectory while continuously collecting RSSI, Physical Layer (PHY) transitions, and disconnection events. The main contributions of this work are:

- Robot-assisted setup for connectivity testing under controlled and repeatable physical conditions;
- Synchronized data collection protocol combining robot navigation data with wireless signal metrics;
- Preliminary evidence that the proposed setup reduces execution variability by controlling route, speed, and device movement across runs, supported by a direct comparison with human-carried data collection.

The rest of this paper is organized as follows. Section 2 provides an overview of the related work. Section 3 describes the proposed approach. Section 4 presents the Bluetooth Low Energy case study. Section 5 reports the obtained results and corresponding observations. Section 6 discusses the main findings and implications of the study. Finally, Section 7 concludes the paper and outlines directions for future work.

2 Related Work

2.1 Wireless Connectivity as a Software Quality Concern

Evaluating wireless connectivity under realistic conditions requires the physical displacement of devices, a dimension that static laboratory setups fail to capture. Pascacio et al. [12] constructed an indoor BLE dataset by deploying multiple static mobile devices to capture RSSI fluctuations caused by human movement along predefined paths, explicitly acknowledging that the absence of systematic collection infrastructure makes such datasets tedious to produce and difficult to replicate. Their work illustrates both the demand for controlled mobility data and the limitations of manual data collection approaches under dynamic human interference. While their approach captures environmental dynamics, achieving true device-to-device mobility repeatably remains an open challenge.

Denis et al. [5] showed that body shadowing introduces significant RSSI variability and packet loss, prompting statistical and machine learning approaches to detect and mitigate this effect. Beyond signal attenuation, human-carried approaches are inherently inconsistent: walking speed varies naturally across and within test runs, making it difficult to reproduce the same mobility conditions. Together, these factors make manually conducted wireless connectivity tests difficult to reproduce and compare across runs.

2.2 Mobile Robots as Testing Infrastructure

The use of robots as testing infrastructure has been explored in software engineering to overcome the limitations of manual and intrusive test execution. Maciel et al. [8] conducted a systematic mapping study on robotic testing of mobile devices, finding that robots enable more realistic interactions, improve repeatability, and reduce human-induced variability. However, the surveyed studies predominantly employ stationary robotic arms for GUI interaction and performance testing, leaving the physical displacement of the device and its effect on wireless connectivity largely unaddressed.

More recently, Araújo and Miranda [2] proposed a reconfigurable testbed based on mobile robots for evaluating Wi-Fi mesh network performance under realistic mobility conditions, demonstrating that robotic motion can induce and capture connectivity degradation, latency spikes, and handover events in a controlled and reproducible manner. Their work validates the use of mobile robots as a physical testing infrastructure but focuses exclusively on Wi-Fi and does not address BLE connectivity or integrate autonomous navigation with synchronized signal and position data collection.

Together, these works demonstrate the value of mobile robots as testing infrastructure, but none has applied this approach to BLE connectivity with synchronized signal and position data.

3 Proposed Approach

Wireless connectivity testing often depends on observing how a connection behaves as devices move through a physical environment. When such a movement is performed manually, variables such as route, speed, distance, and execution timing are difficult to control, which limits repeatability and makes it harder to relate connection degradation or loss to physical conditions. A mobile robot combined with a navigation strategy helps address this limitation by transforming movement into a controllable part of the testing procedure. The robot can execute predefined routes, maintain configurable speeds, and provide position and velocity logs that contextualize wireless measurements.

Based on this analysis, this paper proposes the use of a mobile robot as a controllable physical execution platform for wireless connectivity testing, combining a fixed and a robot-carried device with synchronized navigation and signal logging, as illustrated in Figure 1. As detailed in Section 3.1, the infrastructure executes the same route under configurable conditions such as speed, path, and stopping criteria, supporting reproducible and instrumented evaluation of wireless connectivity behavior.

The expected outcome is not only the observation of a connection loss event, but the contextual data needed to interpret it, *e.g.*, where the robot was when the connection degraded, its velocity, the route

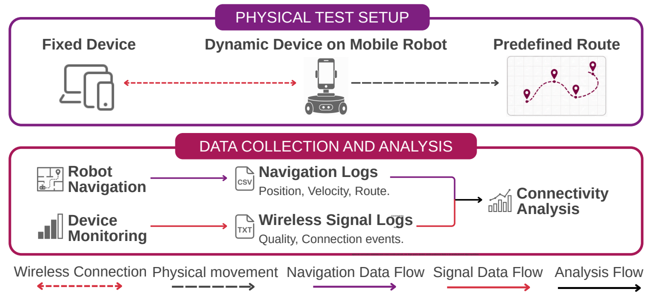

Figure 1: Overview of the proposed robot-assisted testing infrastructure, including the physical setup and the data collection flow.

segment being executed, and whether the loss occurred consistently across repeated runs.

3.1 Test Infrastructure

Table 1: Proposed testing infrastructure components.

Component	Role
Mobile robot	Carries the dynamic device along a predefined route.
Fixed device	Remains at a known position and maintains the wireless connection.
Dynamic device	Moves with the robot during the run.
Navigation stack	Controls localization, path planning, and movement execution.
Monitoring tools	Collect wireless signal data and connection status.
Logging mechanisms	Store navigation data, wireless logs, and execution evidence.

Our proposed testing infrastructure is composed of hardware and software components (see Table 1) that jointly support controlled device movement, signal monitoring, and data collection. The hardware includes a mobile robotic platform and two wireless-enabled devices. The software includes the robot navigation stack, signal monitoring tools, and logging mechanisms.

The mobile robotic platform is responsible for carrying one of the devices along a predefined route. In the current implementation, we use a TurtleBot 4 [11], which supports autonomous navigation. One device remains fixed at a known position, while the other is placed on top of the robot and moves along the route.

The TurtleBot 4 is operated using Robot Operating System 2 (ROS2) [6], with the Nav2 library [7] responsible for localization, path planning, and navigation control. The environment is previously mapped, and the robot navigates using coordinates defined in the map reference frame. The localization system provides the estimated robot position during the run, while odometry provides velocity information. These data allow the infrastructure to record the movement of the dynamic device throughout the route.

Wireless signal monitoring is performed using mobile-device tools. In the current setup, wireless metrics are collected using nRF

Connect [10], while screen recording and video can be used to provide visual evidence of the connection behavior during the run.

The navigation script records information about time, position, distance, and velocity. These data are exported to a structured log file, such as a comma-separated values file. Wireless signal logs and screen recordings are stored separately and can also be synchronized with the robot logs using timestamps or run identifiers.

A key aspect of the infrastructure is its configurability. Navigation parameters, such as route points and robot speed, can be adjusted according to the testing goal. This configurability enables the same infrastructure to support different routes, speeds, devices, and wireless technologies.

3.2 Workflow

The proposed workflow combines environment preparation, robot navigation, device setup, data collection, and execution monitoring into a repeatable sequence.

First, the physical environment is mapped and loaded into the robot navigation system. Then, a route is defined using reference points, such as an initial point, optional intermediate points, and a final point. The selection of these reference points should consider the physical layout of the environment, the navigable path available and the expected behavior of the wireless technology under evaluation. These points determine the path followed by the dynamic device during the run, allow the same movement pattern to be repeated across different executions and ensure that the connectivity event of interest is captured.

Next, the devices are positioned. The fixed device is placed at a known location, while the dynamic device is placed on top of the mobile robot. Before navigation starts, the wireless connection between both devices is established and verified. Once the connection is active, signal monitoring and screen recording are started.

After the device setup is complete, the robot navigation script is executed. The script sets the robot's initial pose in the map and sends navigation goals according to the predefined route. During movement, the script continuously records navigation data, while the mobile-device tools collect wireless signal information. This creates two complementary data streams: one describing the robot's movement and another describing the connectivity behavior.

The execution continues until one of two conditions occurs: the wireless connection is lost, or the robot reaches the final point. At the end of each run, all logs and recordings are saved with a common run identifier. This organization allows the data sources to be analyzed together after execution.

This workflow provides the basis for a reusable testing procedure, adaptable to different routes, speeds, device types, or wireless technology. In the next section, we instantiate this approach in a case study focused on Bluetooth Low Energy connectivity degradation and connection loss between two smartphones.

4 Case Study: Bluetooth Low Energy Connectivity

To evaluate the proposed infrastructure, we conducted a case study focused on BLE connectivity. The study investigates whether the robot-assisted execution can provide controlled and repeatable conditions for wireless connectivity testing while collecting synchronized navigation and wireless signal data. Two device configurations were evaluated: a homogeneous configuration using two identical smartphones and a heterogeneous configuration using smartphones from different manufacturers.

This study focuses on characterizing BLE RSSI behavior as a function of physical distance under realistic, uncontrolled indoor conditions. Environmental factors that could also influence signal strength, such as co-channel Wi-Fi traffic, other wireless radios, and pedestrian presence, were not isolated, disabled, or otherwise controlled during data collection, as quantifying their individual impact falls outside the scope of this work.

4.1 Setting

Experiments were conducted in an indoor university hallway using a TurtleBot 4 mobile robot. Before the experiments, the environment was mapped using the ROS 2 navigation stack, allowing the robot to autonomously follow a predefined route between an initial point, an intermediate point, and a final point, as illustrated in Figure 2.

Following the criteria described in Section 3, the route was defined by three points. The initial point represents the starting position of the robot and the approximate location of the fixed device. The midpoint was defined at the turn connecting the two corridor segments, approximately 64.08 m from the initial point. This choice allows the route to be divided into two segments of comparable navigational structure while marking a physical reference for the analysis. The final point represents the end of the navigation, located approximately 31.17 m from the midpoint. This structure supports a controlled movement pattern while allowing the evaluation to cover a longer distance than a single-point target.

The fixed device was a smartphone running Android 16 with a custom OS layer. The mobile device used in the homogeneous configuration had the same operating system. For the heterogeneous configuration, the mobile device was replaced with a smartphone running Android 13 with a custom OS layer.

Figure 2 shows the corresponding navigation visualization generated in RViz during the route's two segments, displaying the occupancy grid map, the localized robot pose, and the path executed between the initial and midpoint (top) and between the midpoint and final point (bottom) references.

4.2 Robot-Assisted Procedure

Each experimental run followed the workflow presented in Section 3.2. Three independent runs were performed for each device configuration (homogeneous and heterogeneous), with navigation and wireless data synchronously collected until the BLE connection was lost or the robot reached the final point. All runs used the same navigation parameters, route, and stopping criteria.

4.3 Collected Data

Following the synchronized data collection process, each run produced navigation and wireless signal logs referenced by timestamps. These logs were merged to generate the dataset analyzed in this study. The collected data include: RSSI measurements, BLE PHY

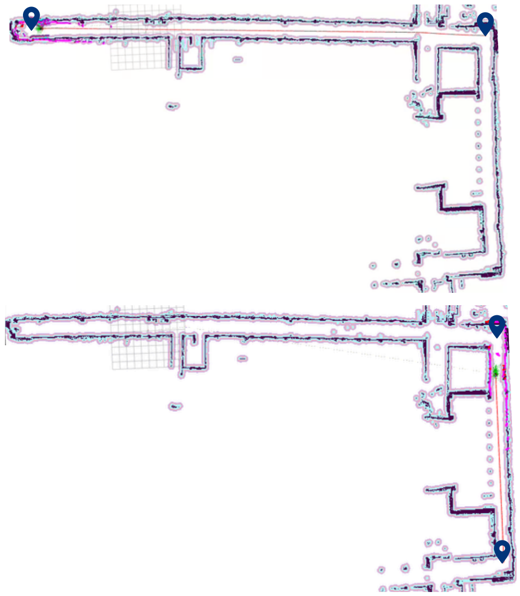

Figure 2: RViz navigation visualization of the route segments, showing the occupancy grid map, robot localization, and executed path between reference points.

transitions, Disconnection events, Robot trajectory, Velocity, Distance traveled until disconnection and Execution time. These metrics were used to characterize signal degradation and evaluate the repeatability of robot-assisted executions.

4.4 Baseline Human-Carried Procedure

To provide a baseline for comparison, the same experimental protocol was executed by a human participant carrying the dynamic smartphone along the predefined route. The baseline represents the conventional approach adopted in wireless connectivity testing, in which a tester transports the device through the environment.

To ensure a fair comparison, all experimental conditions were kept equivalent, including the environment, route, smartphone placement relative to the carrier, BLE connection configuration, signal collection tools, stopping criteria and number of runs. The only intentionally modified variable was the execution platform, replacing the mobile robot with a human participant.

Unlike the robot-assisted execution, which provides accurate map-based localization, the human-carried baseline relied on GPS Logger application [3], running on the dynamic device alongside the wireless signal monitoring tool. Recording timestamped position, instantaneous speed, and cumulative distance traveled throughout each run, providing a human-carried counterpart to the robot navigation log. Since GPS positioning is inherently less accurate in indoor environments, cumulative traveled distance and average velocity were used as the primary metrics to characterize the execution, rather than the absolute spatial trajectory.

5 Results and Observations

A total of six test runs were conducted along the same predefined trajectory. To demonstrate the setup capability across different hardware suppliers, three runs were performed using devices from the same manufacturer (homogeneous configuration), while the remaining three used devices from different manufacturers (heterogeneous configuration). Furthermore, to ensure reproducibility, the target robot velocity was set to a maximum of 0.26 m/s across all trials, establishing a consistent baseline for comparison. Finally, the navigation and wireless logs were successfully merged using timestamps, producing a combined dataset that relates each signal measurement to the corresponding robot position and velocity.

5.1 Feasibility of the Setup

In all runs, the robotic platform successfully followed the defined route while simultaneously collecting position, velocity, and wireless connectivity data, with navigation and wireless signal data effectively synchronized throughout each run.

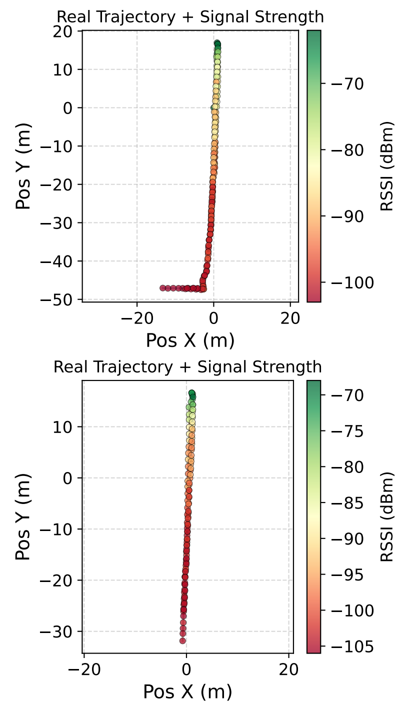

Figure 3: Trajectories of the robot with points colored by RSSI value, overlapping the three repeated runs, in homogeneous (top) and heterogeneous (bottom) configurations.

Figure 3 shows the trajectory of the robot across the three repeated runs, with points colored by RSSI values. The robot departed from the initial point and navigated until the connection was lost. The trajectories overlap closely across all runs, demonstrating that the robot consistently reproduced the planned path. The two configurations, however, exhibited distinct connectivity behaviors. In the homogeneous configuration, the BLE connection remained active after the robot passed the midpoint and continued toward the final point before disconnection. Conversely, in the heterogeneous configuration, the connection was consistently lost before the midpoint was reached, suggesting a reduced communication range when different smartphone models were used.

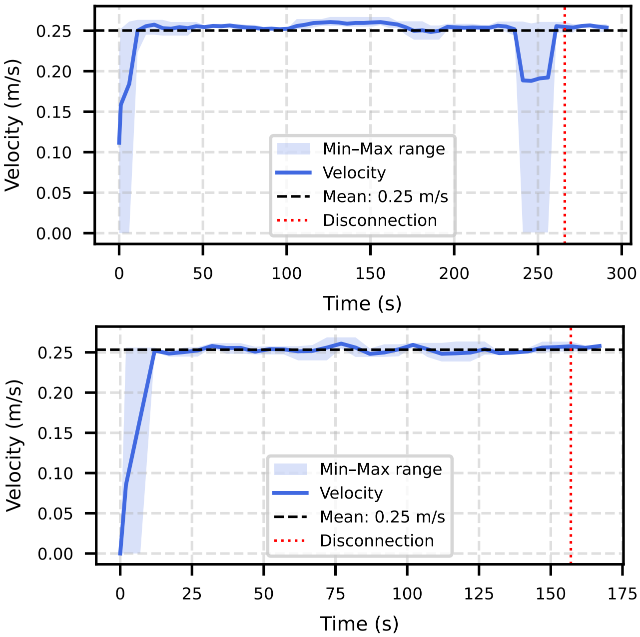

Figure 4: Velocity over time in homogeneous (top) and heterogeneous (bottom) configurations during run 1.

Another important metric correctly captured during execution was robot velocity. As shown in Figure 4, the robot maintained a velocity close to its programmed speed of 0.25 m/s throughout the experiments. In the homogeneous configuration, a short reduction in velocity occurred when the robot changed direction after the midpoint to proceed toward the final point. Nevertheless, this maneuver had little impact on the overall execution, with the average velocity remaining close to the target value.

Table 2 summarizes the main metrics extracted from the merged navigation and wireless logs for each of the six runs, including mean RSSI, mean velocity, and disconnection distance (the distance traveled from the start point until the wireless connection loss event was recorded). Within each configuration, the runs exhibited low variance in mean velocity and disconnection distance, indicating that the robotic platform was effective in maintaining controlled and repeatable physical conditions across independent executions.

Table 2: Summary of robot runs for homogeneous and heterogeneous device configurations.

Config.	Run	RSSI (dBm)	Vel. (m/s)	Dist. (m)	Time (s)
Homog.	1	-89.83	0.247	68.4	294.8
Homog.	2	-90.51	0.244	67.3	275.4
Homog.	3	-91.97	0.247	66.9	268.0
Heterog.	1	-95.72	0.253	39.4	168.6
Heterog.	2	-97.30	0.253	47.3	189.8
Heterog.	3	-97.11	0.253	41.6	168.9

Taken together, the near-constant velocity, the closely overlapping trajectories, and the low run-to-run variation in Table 2 indicate that the platform maintains consistent conditions from run to run, supporting its use as a reproducible testing infrastructure.

5.2 Observed Connectivity Behavior

The behavior of BLE connectivity metrics was also examined throughout the runs. As shown in Table 2, mean RSSI remained close across the three runs within each configuration (homogeneous: -89.83 to -91.97 dBm; heterogeneous: -95.72 to -97.30 dBm), with a spread of less than 2.5 dB in either case. This narrow spread indicates that repeated exposure to the same conditions did not introduce significant RSSI variability, in contrast to the substantial attenuation and variability that body shadowing introduces in human-carried measurements [5]. The homogeneous configuration showed a mean disconnection distance of 67.5 m, while the heterogeneous configuration yielded a lower mean of 42.8 m, alongside slightly higher RSSI variation. Although preliminary, these results illustrate that the proposed infrastructure captures a distinct but internally consistent connectivity behavior across different device configurations.

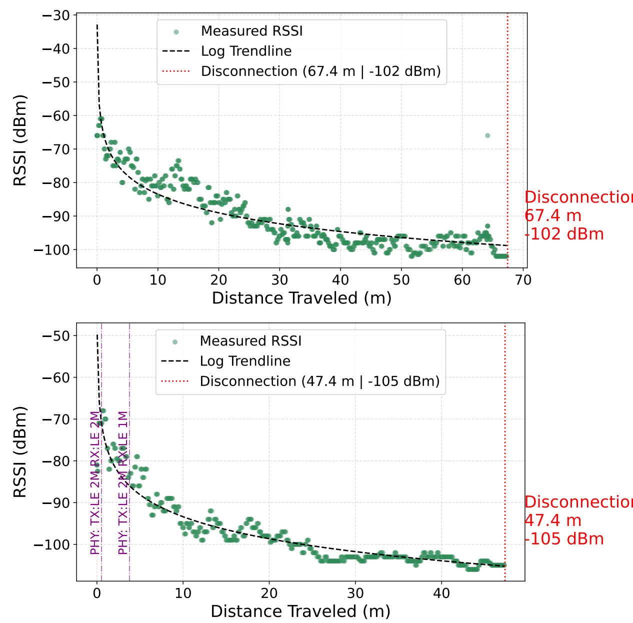

Figure 5: RSSI degradation, PHY transitions, and disconnection event over distance traveled during run 2, in homogeneous (top) and heterogeneous (bottom) configurations.

Figure 5 provides a detailed view of each configuration, showing that RSSI degrades as distance between the fixed and robot-mounted devices increases, consistent with expected Bluetooth propagation behavior [13]. Driven by this degrading signal, a PHY transition was observed during the trajectory, indicating an adaptive response of the BLE stack to maintain link stability, as standardized by the Bluetooth Core Specification [4]. Furthermore, a disconnection event was recorded, enabling precise identification of both the threshold RSSI value and the exact distance of disruption.

5.3 Comparison with Human-Carried Tests

Table 3: Summary of human-carried runs for homogeneous and heterogeneous device configurations.

Config.	Run	RSSI (dBm)	Vel. (m/s)	Dist. (m)	Time (s)
Homog.	1	-92.54	1.17	66.70	61.50
Homog.	2	-94.54	1.27	70.30	56.60
Homog.	3	-94.92	1.21	68.00	59.00
Heterog.	1	-87.04	1.01	50.00	50.10
Heterog.	2	-86.66	0.90	55.00	64.60
Heterog.	3	-89.20	0.97	46.40	50.10

Table 3 summarizes the metrics collected during the human-carried baseline runs described in Section 4.4. For these executions, average velocity was computed from the traveled distance and total execution time. In contrast to the robot-assisted runs, whose trajectory, velocity, and localization were continuously provided by the robot navigation system, the human-carried executions relied on manual movement, making it more difficult to reproduce identical physical conditions across repeated runs.

The results show that the human participant walked at a considerably higher average speed (approximately 0.9–1.3 m/s) than the robot's programmed speed of 0.25 m/s. Although the same experimental protocol was followed, the disconnection distances exhibit greater variation between repeated runs, particularly in the heterogeneous configuration. In comparison, the robot-assisted executions maintained highly consistent trajectories, controlled speed, and synchronized signal acquisition, producing nearly identical disconnection points across repetitions.

These observations reinforce the main contribution of this work. By combining controlled navigation with synchronized wireless signal collection, the robotic platform provides a more reproducible and precise testing infrastructure than the human-carried procedure. This enables connectivity measurements to be collected under repeatable conditions, reducing execution-induced variability and improving the reliability of experimental comparisons.

6 Discussion

The results presented in Section 5 confirm the proposed setup as a controlled, reproducible testing platform for wireless connectivity. This section discusses the broader implications of these findings for the use of mobile robots as infrastructure for reproducible testing of mobile and cyber-physical systems.

6.1 Implications for Software Engineering

The proposed setup transforms physical movement into a controlled test variable: near-constant velocity, a predefined trajectory, and synchronized data collection allow the same scenario to be consistently replicated. This makes connectivity a measurable non-functional requirement, similarly to performance or availability. For instance, the disconnection distances in Table 2 could inform a concrete acceptance criterion such as "the application shall preserve a stable BLE connection with the paired device for at least 60 meters of continuous movement", automatically and repeatably verifiable through the proposed infrastructure, analogous to how throughput or response-time thresholds are verified in performance testing.

The comparison with the human-carried baseline reinforces this argument empirically. While the robot maintained velocity within a narrow range across all six runs and disconnection distance with limited spread within each configuration, the human-carried runs varied substantially in both metrics, despite following an otherwise identical protocol. This contrast illustrates concretely the reproducibility gap that motivated this work: the same acceptance criterion discussed above would be difficult to evaluate consistently using a human tester, since the conditions under which the test is executed cannot be reliably controlled. The robotic platform, in contrast, allows the same scenario to be repeated with comparable velocity and trajectory across independent executions, making connectivity thresholds testable in a manner analogous to deterministic test oracles in conventional software testing.

6.2 Threats and Limitations

This work has limitations. Experiments were conducted in a single indoor environment with no physical obstacles, and each configuration was evaluated with only three runs, restricting generalizability. The human-carried baseline relied on a single participant and on GPS-based positioning, which is less accurate indoors than the robot's map-based localization and may have introduced noise into velocity and distance metrics. Only two device manufacturers were considered, and wireless interference from the surrounding environment was not controlled or measured.

7 Conclusion and Future Work

This work introduced a robot-assisted approach for wireless connectivity testing, using a mobile robot as a controlled movement platform. The preliminary results indicate that the setup captures signal degradation and connection loss while maintaining substantially more consistent conditions across runs than a human-carried baseline, supporting its use as extensible infrastructure for evaluating connectivity-dependent software systems.

Future work should focus on four directions: extending the setup to other wireless technologies, such as Wi-Fi; conducting runs in environments with physical obstacles and varying interference conditions; further automating the setup; and expanding the human-carried comparison with more participants and trials for a statistically robust analysis.

Artifact Availability

To support reproducibility, an artifact package is available at https://figshare.com/s/b83b8c3c59c02d096990. The package includes the navigation script, route definition, sample navigation logs, wireless signal logs, and documentation describing the evaluation protocol.

Acknowledgements

This work was supported by the research and innovation cooperation project between Softex (funded by the Brazilian Ministry of Science, Technology, and Innovation through Law 8.248/91 within the scope of the National Priority Program) and CIn-UFPE, developed under the CRIAR project.

References

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2347–2376. doi:10.1109/COMST.2015.2444095

[2] Eduardo Araújo and Breno Miranda. 2025. Mobile Robotics for Throughput Testing in a Reconfigurable Test Environment. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, Porto Alegre, RS, Brasil, 859–864. doi:10.5753/sbes.2025.11613

[3] BasicAirData. 2023. GPS Logger for Android. https://github.com/BasicAirData/GPSLogger. Accessed: 2026-07-20.

[4] Bluetooth Special Interest Group (SIG) 2023. *Bluetooth Core Specification Version 5.4*. Bluetooth Special Interest Group (SIG). https://www.bluetooth.com/specifications/specs/core-specification-5-4/

[5] Benoît Denis, Nicolas Amiot, Bernard Uguen, Arturo Guizar, Claire Goursaud, Anis Ouni, and Claude Chaudet. 2014. Qualitative analysis of RSSI behavior in cooperative wireless body area networks for mobility detection and navigation applications. In *2014 21st IEEE international conference on electronics, circuits and systems (ICECS)*. IEEE, IEEE, Piscataway, NJ, USA, 834–837.

[6] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics* 7, 66 (2022), eabm6074. doi:10.1126/scirobotics.abm6074

[7] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. 2020. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, IEEE, Piscataway, NJ, USA, 2718–2725.

[8] Lucas Maciel, Alice Oliveira, Riei Rodrigues, Williams Santiago, Andresa Silva, Gustavo Carvalho, and Breno Miranda. 2022. A systematic mapping study on robotic testing of mobile devices. In *2022 48th Euromicro conference on software engineering and advanced applications (SEAA)*. IEEE, IEEE, Piscataway, NJ, USA, 475–482.

[9] Francisco G De Oliveira Neto, Richard Torkar, and Patricia DL Machado. 2015. An initiative to improve reproducibility and empirical evaluation of software testing techniques. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. IEEE, IEEE, Piscataway, NJ, USA, 575–578.

[10] Nordic Semiconductor. 2024. nRF Connect for Mobile. https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-mobile. Accessed: 2026-07-20.

[11] Open Robotics and Clearpath Robotics. 2025. TurtleBot 4. https://www.turtlebot.com. Accessed: 2026-05-22.

[12] Pavel Pascacio, Joaquín Torres-Sospedra, Antonio R Jiménez, and Sven Casteleyn. 2022. Mobile device-based Bluetooth Low Energy Database for range estimation in indoor environments. *Scientific Data* 9, 1 (2022), 281.

[13] Adel Thaljaoui, Thierry Val, Nejah Nasri, and Damien Brulin. 2015. BLE localization using RSSI measurements and iRingLA. In *2015 IEEE international conference on industrial technology (ICIT)*. IEEE, IEEE, Piscataway, NJ, USA, 2178–2183.