WifiRoam: An open-source simulator to optimize Wi-Fi handover in industrial networks

Pietro Chiavassa*, Stefano Scanzio*, Gianluca Cena*, Francesc Wilhelmi†, Boris Bellalta†
*National Research Council of Italy (CNR–IEIIT), Turin, Italy.
†Universitat Pompeu Fabra, Engineering Department, Barcelona, Spain.
Emails: {pietrochiavassa, stefano.scanzio, gianluca.cena}@cnr.it, {francisco.wilhelmi, boris.bellalta}@upf.edu

***Abstract*—The increasing adoption of mobile industrial equipment, e.g., autonomous mobile robots (AMR) and autonomous guided vehicles (AGV), in next-generation production plants is driving a rising need for reliable industrial wireless communication. Due to its cost-effectiveness and recent shift towards reliability with IEEE 802.11bn, Wi-Fi is currently considered as one of the most promising technologies for filling that gap. However, one of the main challenges in industrial Wi-Fi networks is the management of the handover process of a station (STA) from one access point (AP) to another. This must be conducted in a way that ensures optimal connectivity for client STAs, while also preventing service disruptions related to the re-association process. This work presents WifiRoam, an open-source simulator for studying and optimizing Wi-Fi handover in industrial networks. The simulator allows defining a 2D environment in which a mobile STA moves along a polygonal path, simulating the trajectory of a mobile robot on a factory floor. An arbitrary number of APs can be placed inside the environment, which the STA can associate with. The simulator, written in Python, allows for easy implementation and testing of innovative handover strategies, thus laying the groundwork for future roaming algorithms. In this paper, a simulation environment is generated and imported into WifiRoam, by modeling spectrum conditions via network simulations conducted with the ns-3 library. Then, a novel network-digital-twin-based handover solution is presented and compared to traditional strategies (received signal strength indicator (RSSI) and distance-based), in order to showcase and evaluate the potential of the WifiRoam simulator.**

***Index Terms*—Wi-Fi, Handover, Roaming, Network Digital Twins, Industrial Wireless Communication, IEEE 802.11.**

I. Introduction

Research on the use of wireless networks in industrial scenarios began more than two decades ago. The ability to make devices communicate without the need for cables has the potential to increase flexibility substantially, both in the deployment phase and when the plant is operating. As a matter of fact, modern industrial settings increasingly rely on technologies aimed at supporting fast and inexpensive reconfiguration. A popular trend, both for warehouses and on the shop floor, is to integrate in the production process automated guided vehicles (AGVs) and autonomous mobile robots (AMRs). For example, by restructuring the internal logistics of its Detroit plant using mobile robots, a known car manufacturer has recently managed to speed up operations by about two orders of magnitude with respect to manual handling. Building on the concept of collaborative robotics introduced in the context of Industry 4.0, AMRs have the potential to become a true game changer, enabling unprecedented levels of production efficiency. Although this may resemble the *dark factory* vision conceived in the 1980s, the new approach provides dramatically greater adaptability to the ever-changing global market conditions, and also fits the human-centered paradigm foreseen by Industry 5.0.

To get the most out of a fleet of AMRs involved in a given industrial process, they must be able to interact in real-time with other AMRs and the factory backbone. The primary requirements of these systems concern dependability and responsiveness, and this also applies to the case where AMRs are moving around. Communication technologies derived from mobile cellular networks (e.g., private 5G) are quite reliable and may be able to cover the whole area of the plant. However, one of the main downsides is that they rely on licensed spectrum, leading to higher costs. On the other hand, recent IEEE 802.11 versions (Wi-Fi 7 [1] and, soon, 8) are relatively inexpensive, easier to connect to the Ethernet backbone, including setups based on time-sensitive networking (TSN), and show very high performance. The main drawback of Wi-Fi is its limited spatial coverage (some tens of meters at best), which can be overcome by suitably deploying a large number of access points (APs) across the plant, to provide moving devices with ubiquitous connectivity.

This leads, however, to an issue that concerns roaming: when a mobile station (STA) moves away from its current AP and enters a region served by another AP, a handover procedure must be carried out. The involved operations (e.g., re-association) may take some time (up to a few seconds), during which communication is severely impaired. Therefore, there is an increasing need for advanced roaming techniques able to minimize disruptions. Among the different approaches, a promising solution is to rely on a network digital twin (NDT), which informs the moving device when it needs to re-associate and suggests the best available option(s) for the next AP. Possibly, machine learning can be exploited to combine local information with the one obtained from the AP (and NDT), in such a way that optimal decisions can be reached.

In this paper, we present WifiRoam, a simulation tool designed to study, develop, and improve handover strategies in the context of industrial applications, in order to make the

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”).

process faster and to lower its impact on communication quality. By comparing a novel NDT-based approach to traditional handover strategies, we evaluate the extent of the improvement that such a solution can provide to a mobile industrial device connected through Wi-Fi.

The paper is structured as follows: Section II discusses state-of-the-art related works on Wi-Fi handover and network digital twins. Section III introduces the WifiRoam simulator, presenting all of its components and the implemented handover algorithms. Section IV describes the methodology and parameters used to build the simulation environment and test the different handover strategies. Experimental results are presented in Section V. Finally, Section VI discusses the current limitations of WifiRoam, while Section VII summarizes the main conclusions of the study.

II. Related Works

In recent years, extensive research has focused on digital twins for wireless networks [2], particularly for 5G networks [3]. More recently, this trend has extended to Wi-Fi networks as well [4], [5]. Concerning Wi-Fi, these solutions can find their roots in the IEEE 802.11k/r/v amendments [6]. IEEE 802.11k, approved in 2008, already introduced protocol functionalities to address the need for monitoring wireless links and network status. Interestingly enough, starting with IEEE 802.11r and later with IEEE 802.11v, this was strictly tied to the problem of improving handover performance.

State-of-the-art work on Wi-Fi roaming focuses on two main problems: the first is the implementation of seamless handover solutions to prevent connectivity disruptions during the re-association process, while the second is the definition of strategies to decide when to re-associate and to which AP. Network digital twins are typically employed to tackle the latter problem, while also trying to focus on other aspects of the Wi-Fi protocol.

In [7], a Wireless Time Sensitive Networking (WTSN) strategy for industrial AMRs was used to achieve zero-delay roaming. The solution was tested both via simulation and on a real testbed deployed inside a warehouse. A robot was equipped with two Wi-Fi cards and associated to multi-radio APs with two separate connections, using IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) techniques to eliminate duplicate packets. In that case, seamless roaming was achieved by re-associating one link at a time to the new AP. In [8], a similar solution was presented that exploits MLO to implement seamless roaming in IEEE 802.11bn networks. A “make-before-break” approach was followed, consisting in using one of multiple links to exchange the roaming handshake messages and associate with the new AP. Then, all remaining links are also switched to the new AP with minimal delay.

In [9], an architecture for a Wi-Fi NDT is introduced that splits handover management into four independent operations: feature acquisition, channel model, roaming/handover, and MLO/MLD re-association. Each of these operations is the subject of specific research activities. For instance, [10] proposes an in-band network telemetry protocol to transfer information about the quality of the communication networks, [11] uses machine learning to predict the future quality of a Wi-Fi channel, [12] proposes and experiments in practice a handover mechanism, and [13] analyzes the effect of using MLO. A quite interesting kind of NDT for Wi-Fi, which will be introduced in the next IEEE 802.11bn amendment (Wi-Fi 8), is multi-access point coordination (MAPC) and, in particular, the coordinated spatial reuse (C-SR), which permits the concurrent transmission on the same channel by acting on transmission power [14].

Finally, in [15] and [16], a multi-parameter machine-learning-based handover decision algorithm is introduced. It employs a long short-term memory (LSTM) artificial neural network (ANN) to predict future key performance indicators for a mobile STA, moving along either a linear or curvilinear trajectory, when connected to two different APs. The predicted indicators are fed to either a decision rule or an ANN classifier, which determines whether to trigger the re-association process.

This paper uses the WifiRoam simulator to compare three handover mechanisms for non-redundant links: a) a simple approach based on the measured RSSI, which requires no specific knowledge from the STA; b) an approach where decisions are based on the distance between devices, which requires the STA to know its position in real-time, as well as the position of APs; and c) an advanced approach based on an NDT where the STA selects the AP depending on a specific metric (mean number of transmission attempts per packet), as estimated by the NDT given the position of the STA.

III. Simulator

WifiRoam[1] is an object-oriented, event-based Python simulator that relies on the `SimPy` library. Its purpose is to study, test, and improve Wi-Fi handover algorithms for industrial networks. To do so, the simulator allows defining a 2-dimensional (2D) space of arbitrary dimension containing any number of Wi-Fi APs. A mobile STA, referred to as STA under test (SUT), moves in the simulation space following a randomly generated polygonal trajectory; this behavior is useful, for instance, to represent the movement of an autonomous mobile robot on a factory floor. By incorporating the spectrum conditions of the recreated environment, which can be analytically defined, synthetically generated via network simulators (e.g., ns-3, OmNet++, etc.), or measured in a real-world setting, the simulator keeps track of the performance of the SUT along a given trajectory. Then, by simulating the association of the SUT with different APs along its path, it allows evaluating the effectiveness of a given handover strategy (e.g., strongest signal first) and computing relevant network metrics.

In what follows, we describe the three main components of WifiRoam in detail: the *environment*, the *trajectory simulator*, and the *handover algorithm*.

[1]https://github.com/ptrchv/WifiRoam

A. Environment

The environment component is used to represent the spectrum conditions observed by the wireless network at every point of the 2D simulation space. Its software interface is described by the base abstract class `WifiEnvironment`, which can be implemented by derived classes to provide custom functionalities and behaviors. An environment is characterized by the X and Y dimensions of the rectangular simulation space, the number of APs, and their positions. Given a point in space and time, it allows modeling packet transmissions from the SUT towards one of the APs, and receiving beacon information, i.e., received signal strength indicator (RSSI) and signal-to-noise ratio (SNR), from all the APs in range of the SUT. In addition, it allows for the retrieval of statistics (e.g., mean, 99th percentile, 99.9th percentile, etc.) of relevant network metrics (e.g., latency, number of retransmissions, packet loss ratio, etc.) encountered by the SUT during packet transmissions at every point in space. Currently, two implementations of the base class `WifiEnvironment` are provided: `SimpleWifiEnv` and `MapWifiEnv`. `SimpleWifiEnv` was developed for initial testing, and describes packet transmissions, beacon information, and network metrics as an analytical function of the SUT's position. Instead, `MapWifiEnv` uses packet captures of data transmissions and received beacons to describe the simulation environment. These captured samples can be collected either via a network simulator or a mobile measuring equipment that records network conditions on a real factory floor. Results presented in this work are obtained with `MapWifiEnv` using synthetic samples generated via the ns-3 network simulator.

`MapWifiEnv` permits the definition of a simulation space of arbitrary dimensions containing a variable number of APs. Each of the APs is assumed to work on a distinct non-interfering Wi-Fi channel c, making them completely independent of each other (this assumption is often verified quite well in real Wi-Fi infrastructures). Therefore, every AP can be identified by the related channel c and separately characterized by defining its map. A map is a 2D square grid centered on the AP that covers its entire communication range. The resolution of the grid, i.e., the number and dimensions of its cells, can be arbitrarily defined. Each cell is characterized by a dataset $\mathcal{D}^c_{<x,y>} = \{p_i\}$ containing the outcomes of Wi-Fi data packet transmissions from a stationary STA positioned in the center of the cell and associated to the AP ($< x, y >$ are the quantized relative spatial coordinates with respect to the AP and c is the Wi-Fi channel on which the AP operates). Each packet transmission is represented by a tuple in the format $p_i =$<*acked, latency, num_tries, rssi, noise*>, where: *acked* is a boolean value indicating whether the transmission was successful; *latency*, for successful transmissions, is the time elapsing between the generation of the packet at the application layer and the reception of its acknowledgement (ACK frame as per IEEE 802.11); *num_tries* is the number of transmission attempts required to deliver the packet (at least one); finally, *rssi* and *noise* describe the RSSI and noise measurements of the last beacon received from the AP before the start of the data packet transmission. Datasets can be used to compute statistics of network metrics and characterize every cell of the AP map.

The overall simulation environment is created by associating a map with each of the considered APs. Knowing the absolute positions of the SUT and the AP to which it is currently associated, a data packet transmission can be simulated very quickly by computing the relative position of the SUT with respect to the AP and using this information to address the relevant cell in the AP's map. Then, by following a Monte Carlo approach, the transmission outcome is obtained by drawing a random sample from the dataset associated with the cell.

B. Trajectory simulator

The trajectory simulator component is used to simulate SUT movement in the environment, generate packet transmission events, and update the state of the roaming algorithm. The SUT follows a randomly generated polygonal path with constant speed. Each new segment of the trajectory (angle and length) is known to the SUT only when it reaches the end of the previous one. This simulates, e.g., the movement of an AMR that is assigned its next task only upon completion of the previous one.

The traffic pattern exchanged over the air is characterized by periodic data packet transmissions from the SUT to the AP, to resemble the exchange of control messages in a typical industrial application. This is implemented by adding periodic events in the event loop of the simulator. When an event is triggered, the event handler updates the SUT position and queries the roaming component to know which is the current *association state* of the SUT. Three association states are defined: CONNECTED, ROAMING, and DISCONNECTED. In the CONNECTED state, the SUT is associated with one of the APs. In this case, a packet transmission event is (randomly) sampled from the environment component, and the outcome is directly logged in the output file of the simulation. Instead, the ROAMING state is used to represent the handover procedure of the SUT, while the DISCONNECTED state means that the SUT is not associated to any of the APs. In the last two cases, a failed packet transmission attempt is logged in the output file.

The log file contains, on each line, a tuple representing the outcome of a single transmission of a data packet performed while the SUT is moving around. The tuple is structured with the format <*time*, *position*, *segment*, *ap*, *count*, *state*, *acked*, *rssi*, *latency*, *num_tries*>. Attributes *time* and *position* represent, respectively, the simulation time and the SUT position when the packet is transmitted; *segment* is a progressive number that identifies the segment of the trajectory in which the SUT is located; *ap* uniquely identifies the AP to which it is connected; *count* counts the number of AP (re-)associations the SUT performed from the beginning of the experiment; and finally, *state* is the association state of the SUT during data transmission. If the SUT is in the CONNECTED state, all the remaining fields are copied from the sample returned by the environment component. Otherwise, *acked* is set to false (to denote packet loss) and the rest of the fields are left unspecified.

The trajectory simulator component also adds periodic events to update the state of the roaming algorithm. Upon their occurrence, the event handler updates the position of the SUT and queries the environment to obtain the RSSI and noise value in that location for every AP.

In `MapWifiEnv`, due to the ns-3 default settings that were used in the simulations, these values are constant and only depend on the relative distance between the SUT and the APs. This information, together with the current SUT position, is fed to the roaming component via an interface method common to all roaming algorithms. Both the update frequency and the information that is actually used by the roaming component to update its state depend on the particular algorithm chosen.

C. Handover algorithm

The roaming component is characterized by the software interface exposed by the `RoamingAlgorithm` abstract base class. This class also provides the basic logic for disconnections, by checking whether the SUT has exited the communication range of the AP to which it is currently associated. The full implementations of roaming algorithms are provided by derived classes, as specified below:

1) `RSSIRoaming`: This algorithm uses a configurable RSSI threshold to trigger the handover procedure. It maintains a data structure containing the RSSI measurements for each of the APs. This data structure is updated periodically, simulating a continuous scanning process. In the following, we assume that the SUT is equipped with a Wi-Fi 7 card supporting MLO. This enables the SUT to dedicate one interface to real-time communication with the associated AP, while the other interface (secondary, possibly with reduced capabilities) is devoted to scanning the remaining channels, thus preventing service disruptions on the primary link. While not strictly necessary, this arrangement improves predictability yet keeping costs acceptable. In the case that a former Wi-Fi 5/6 technology is used, it is not unrealistic to assume that a passive scan performed only on the APs that are known to be deployed near the current SUT position can be carried out quite quickly (sub-second).

At the end of every scanning cycle, the RSSI value measured for the associated AP is checked to determine whether it falls below the configured threshold. If this occurs for three consecutive scanning cycles, the AP with the highest RSSI value (if different from the current one) is selected as the new target, and handover is started.

2) `DistanceRoaming`: This algorithm assumes that the SUT knows its location, which is refreshed periodically, and the number and position of all the nearby APs deployed in the area of interest. As of today, assuming that an STA can determine its current spatial position is no longer unrealistic. This can be obtained in different ways: querying the navigation system used for managing the movement of the AMR in the plant, using commercially-available GPS/UWB/5G devices and services, or, in the near future, directly exploiting Wi-Fi-based location mechanisms. Upon location updates, the SUT checks whether it is associated with the closest AP, and, if not, it starts the handover process.

3) `OptimizedRoaming`: This roaming algorithm mimics the operation of a simple NDT-based solution. It assumes that the SUT, in addition to knowing its location, is also provided with a model of the environment, which can be used to retrieve statistics of relevant metrics concerning communication quality at any possible location. For a real scenario, e.g., for a factory floor, a mobile measuring equipment (small rover) that records spectrum conditions can be used to initially build this model. In the simulator, this is implemented by allowing the roaming and environment components to interact. Consequently, the roaming algorithm is enabled to query statistics (e.g., mean values, 99th and 99.9th percentile, etc.) of all the available network metrics (e.g., number of retransmissions, packet loss ratio, etc.) for every possible position of the SUT, when associated to any of the APs.

The algorithm requires selecting a network metric relevant to communication quality, e.g., the average number of retransmissions per packet, to be optimized along the trajectory followed by the SUT. This is achieved by optimizing handover decisions. Each time the SUT reaches a new trajectory segment, the algorithm selects N equally spaced points on that segment. The spacing between the points is a configurable parameter of the algorithm. For each of these points and for every AP, the environment component is queried to retrieve the value of the network metric to be optimized. This information is used to divide the trajectory segment into multiple sub-segments, so that in each sub-segment there is only a single AP that optimizes the selected network metric. The points that mark the beginning of each sub-segment are saved as *switch points*. When a switch point is reached by the SUT, the handover process is triggered towards the new optimal AP.

To prevent the SUT from switching between APs too frequently, which would cause many service disruptions due to the handovers, it is possible to configure a minimum length for the sub-segments. This is done by selecting the minimum time that the SUT should take to traverse the sub-segment, which, given the speed of the SUT, determines their minimum length. If a segment is deemed to be too short, it is merged with either the next or the previous sub-segment, depending on which one of the two APs provides the best metrics across it.

IV. Methodology

This section describes the methodology for generating, via the ns-3 simulator, the AP maps used by `MapWifiEnv`. In addition, it presents the algorithm parameters used in the experiments.

A. Environment simulation

The ns-3 simulation setup is similar to the one described in [17], [18], but extended to a 2D space. Two AP maps were

TABLE I
POSITION OF INTERFERING STAS IN THE AP MAPS

Device	Map_1 position	Map_2 position
AP	(50 m, 50 m)	(50 m, 50 m)
INT_1	(30 m, 30 m)	(30 m, 20 m)
INT_2	(30 m, 70 m)	(30 m, 80 m)

created by simulating two different network configurations. Such maps were later combined via `MapWifiEnv` to define the spectrum conditions of the multi-AP environment.

1) Network configurations: The simulation space of both network configurations has a size of $100\,\text{m} \times 100\,\text{m}$, with a single AP positioned in its center. The SUT is statically positioned on a $1\,\text{m} \times 1\,\text{m}$ grid covering the entire simulation space. An independent ns-3 simulation is run for each SUT position, and a dataset of packet transmission attempts is collected to characterize every $1\,\text{m} \times 1\,\text{m}$ cell. Due to the large number of simulations required to build all entire AP maps, the simulation time was limited to one hour.

In addition to the SUT, two additional interfering STAs (INT) are introduced in both configurations. They are connected to the central AP and inject interfering traffic. The two AP maps are differentiated only by the position of the two interferes. By introducing a Cartesian coordinate system for the $100\,\text{m} \times 100\,\text{m}$ simulation space, the positions of the AP and interfering STAs in the two maps are shown in Table I.

2) ns-3 configuration: Two different types of network applications are running on the SUT and the interfering STAs, both sending UDP messages to a server installed on the AP. The traffic generated by the SUT is characterized by periodic messages transmitted every 0.5 seconds and containing a payload of only 22 bytes, allowing it to fit inside a single IEEE 802.11 PSDU. This type of traffic represents a typical industrial application exchanging control messages. Instead, the traffic pattern generated by the interfering STAs is designed to model bandwidth-intensive general-purpose operations, such as file transfers and video streaming. For this reason, it consists of bursty packet transmissions. The number of packets in each burst is described by an exponential distribution with an average of 100 packets and bounded to a maximum of 500 packets. Packet payload is 1472 bytes and packet spacing inside each burst is set to $500\,\mu\text{s}$. Similarly, the spacing between adjacent bursts is also modeled with an exponential distribution having an average of $250\,\mu\text{s}$ and an upper bound of $10\,\text{s}$.

The chosen Wi-Fi standard is IEEE 802.11a. This decision was made to reduce complexity by disabling all throughput-enhancing features, which are not beneficial given the type of traffic that is generated by the SUT (short periodic messages). The analysis of mechanisms introduced in Wi-Fi 6 and 7, such as trigger frames and MLO, which are currently not widely supported, is left as future work. Finally, RTS/CTS is not enabled, since the duration of the related control frames is comparable to the data frames generated by the SUT (hence, no benefits would be provided by their adoption). Simulated devices operate in the $5\,\text{GHz}$ frequency band, sharing the same $20\,\text{MHz}$ channel. Finally, all devices are configured to use Minstrel as rate adaptation algorithm.

The PHY layer is simulated via the ns-3 class `SpectrumWifiPhy`. The selected channel model is `SpectrumChannel`, which is configured with the `ConstantSpeedPropagationDelayModel` to describe the signal propagation speed and the `LogDistancePropagationLossModel` to describe the path loss. Default parameters were used for all these models, resulting in a maximum communication range, common to all devices, of $D_{\max} = 51.45\,\text{m}$.

3) Multi-AP environment: Figures 1 and 2 show the spatial distributions of, respectively, the RSSI and the average number of per-packet transmission attempts in Map_1. The red dot in the figures represents the APs. It is interesting to notice that, while the RSSI follows a predictable trend, the average number of transmission attempts shows some discontinuities. These are due to the presence of the interfering STAs, shown as orange dots in Figure 2, which can cause both direct interference and hidden node effects. The latter have a higher impact, and happen when both the SUT and an interferer are in the range of the AP but not in range of each other. Since the SUT cannot see the interfering traffic generated by the other STA, channel sensing is ineffective at preventing collisions for transmissions towards the AP. This explains why the heat map shown in Figure 2 is spatially divided into three distinct areas, represented by different color intensities. When the SUT is in the darker area, it is in range of both interferers, and, therefore, channel sensing prevents most collisions by detecting all the interfering traffic. The slightly lighter areas on the right and left sides, instead, mark the positions for which the SUT is in range of only one of the two interferers and therefore subjected to the hidden node effect. Finally, when the SUT is in the lightest area at the bottom, it is affected by two hidden nodes (one for each interferer), leading to the worst connectivity metrics. These phenomena are also present for Map_2, but the hidden-node areas have different shapes due to the different position of the interfering STAs.

The multi-AP simulation environment is created via `MapWifiEnv` by defining the number and location of the APs, and assigning to each of them the two AP maps. The environment used in the experiments presented in this paper features five APs. Their positions and assigned maps are shown in Table II. Figure 3 shows the spatial distribution of the average number of per-packet transmission attempts for the entire environment, also indicating the color of the AP that provides the best performance (mapping between colors and APs is provided in Table II). The edges of the different color zones represent the optimal handover locations for optimizing the depicted metric, i.e., the average number of per-packet transmission attempts. As it can be seen, the combination of the different AP maps, where hidden node effects are present, generates a non-trivial multi-AP simulation environment.

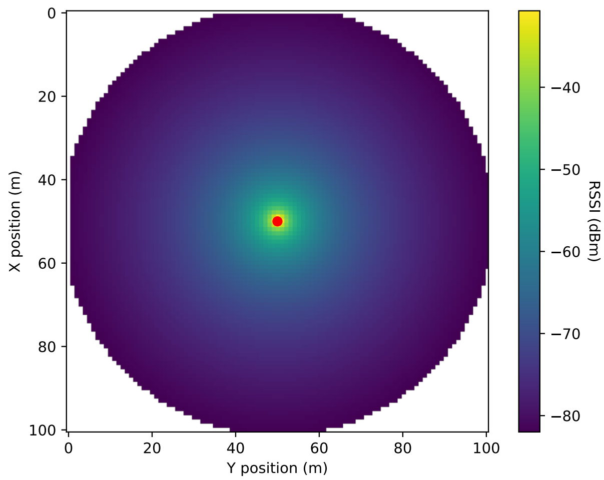

Fig. 1. Spatial distribution of RSSI (Map_1)

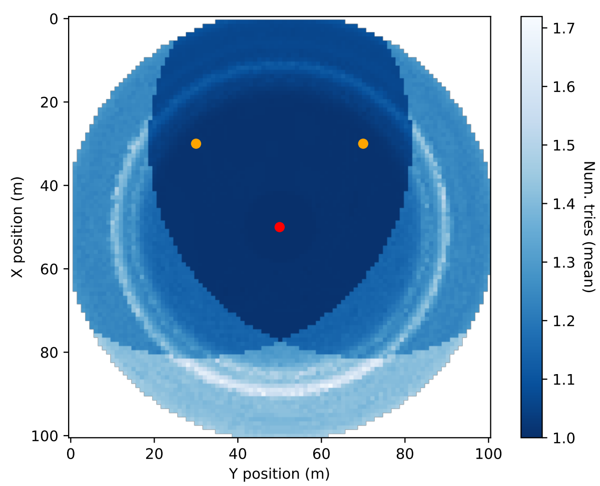

Fig. 2. Spatial distribution of average per-packet tx attempts (Map_1)

B. Handover algorithm simulation

The parameters used for the handover algorithm simulation are reported in Table III. A polygonal trajectory composed of 1500 randomly selected segments is generated and used for all the experiments, to ensure fair comparison. The SUT moves along the trajectory with a speed of $0.5\,\mathrm{m/s}$, generating periodic packets (sent to the AP) with a period of $100\,\mathrm{ms}$. Given the length of the random trajectory, of about $72.5\,\mathrm{km}$, and the speed of the SUT, the duration of the simulated packet exchange is around $40.3\,\mathrm{h}$. The time required to complete the handover procedure is set to $200\,\mathrm{ms}$. During this time, the association state of the SUT is set to ROAMING, and all the transmitted packets are lost. Packet transmission starts two seconds from the beginning of the simulation, allowing the SUT to complete its first association to one of the APs.

Table IV contains the parameters used to test the handover algorithms. `RSSIRoaming` is tested using two different RSSI thresholds to trigger the handover process (middle column in

TABLE II
POSITIONS OF THE APS IN THE SIMULATION ENVIRONMENT, ASSIGNED MAPS, AND COLOR CODES

Device	Position	Assigned map	Color
AP_1	$(0\,\mathrm{m},\ 0\,\mathrm{m})$	Map_1	black
AP_2	$(60\,\mathrm{m},\ 0\,\mathrm{m})$	Map_1	purple
AP_3	$(0\,\mathrm{m},\ 120\,\mathrm{m})$	Map_2	blue
AP_4	$(60\,\mathrm{m},\ 120\,\mathrm{m})$	Map_2	green
AP_5	$(30\,\mathrm{m},\ 60\,\mathrm{m})$	Map_2	orange

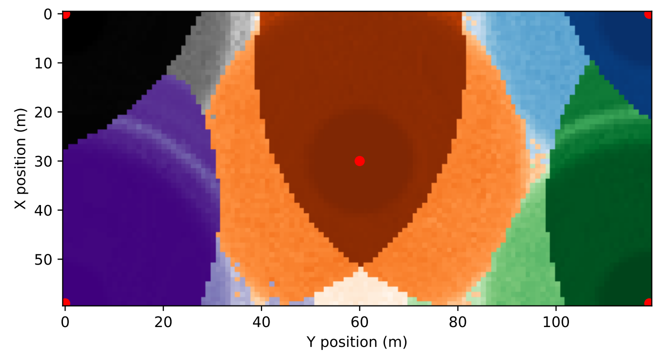

Fig. 3. Spatial distribution of average per-packet tx attempts in the multi-AP environment

the table). By limiting the scanning to five channels (one for each AP), the state of the algorithm can be updated every second. Assuming to use a MLO device with one link dedicated to perform passive scanning, or by knowing when the next packet will be sent (due to the periodic traffic pattern), the scanning process for beacons does not disrupt communication between the SUT and the current AP. For what concerns `DistanceRoaming` and `OptimizedRoaming`, the update period has been lowered to $100\,\mathrm{ms}$, since passive scanning is not required. This is due to the fact that: a) the RSSI information is not used, and, b) the SUT, by knowing its position, it is already aware of neighboring APs. The network metric used in the configuration of `OptimizedRoaming` is the minimization of the average number of per-packet transmission attempts. `OptimizedRoaming` is tested with two minimum sub-segment lengths, expressed as the time required by the SUT to traverse them (rightmost column in Table IV). The shorter segment length, corresponding to $200\,\mathrm{ms}$, gives just enough time to the SUT to complete the handover process before having to start the next one.

V. RESULTS

Table V shows the results of the evaluation campaign of the different roaming algorithms. The metrics reported in the table are evaluated from the outcomes of packet transmissions carried out by the SUT when it moves along its path.

TABLE III
HANDOVER SIMULATION PARAMETERS

Trajectory segments	Trajectory length (km)	Sim. time time (h)	Packet period (ms)	SUT speed (m/s)	Handover time (ms)	TX start time (s)
1500	72.5	40.3	100	0.5	200	2

TABLE IV
ALGORITHM SIMULATION PARAMETERS

Algorithm	Update period (μs)	RSSI threshold (dBm)	Min. switch interval (μs)
RSSI (-80)	1	-80	_
RSSI (-75)	1	-75	_
Distance	0.1	_	_
Optim. (0.2)	0.1	_	0.2
Optim. (10)	0.1	_	10

The *handover packets* metric (rightmost column) counts the number of packets that are lost during handover operations, giving a compact indication of their overhead and frequency. This is not the same as the *packet loss ratio* (PLR), which counts all packets that went lost, including those dropped after the maximum number of retransmissions has been reached. Latency metrics and the mean number of transmission attempts per packet are evaluated only for successful transmissions. Finally, the mean RSSI value encountered along the trajectory is also indicated.

1) `RSSIRoaming`: it can be seen that using a threshold of $-80\,\mathrm{dBm}$ allows this algorithm to reduce the number of handovers compared to the other solutions, as the SUT tends to remain associated with the current AP even when network conditions have significantly degraded. This is reflected by the slightly lower RSSI value measured along the trajectory. However, even if the lowest PLR is achieved, all other metrics are worse than the competing solutions. Increasing the RSSI threshold to $-75\,\mathrm{dBm}$ generates the opposite effect: all metrics are improved, except for the PLR and the number of packets lost during handovers.

2) `DistanceRoaming`: this algorithm obtains very similar results to `RSSIRoaming` with the threshold set to $-75\,\mathrm{dBm}$. This is due to the way RSSI is modeled by ns-3 when the default parameters are used, i.e., it only depends on the distance from the AP. In these conditions, `RSSIRoaming` becomes a variation of the `DistanceRoaming` algorithm, for which the RSSI threshold coincides with the RSSI observed at the point (characterized by its distance from the AP) where re-association is triggered. However, this generally does not hold for more realistic simulated spaces and real-world environments, where the RSSI is affected by the presence of obstacles and reflective surfaces.

3) `OptimzedRoaming`: when configured with the shortest switch interval of $200\,\mathrm{ms}$, this algorithm manages to

TABLE V
PERFORMANCE OF HANDOVER ALGORITHMS

Algorithm	PLR (%)	Latency mean (μs)	Latency 99th (μs	Latency 99.9th (μs)	Attempts mean #	RSSI mean (dBm)	Hand. packets #
RSSI (-80)	0.18	300.5	2326.8	23342.3	1.1372	-70.1	1706
RSSI (-75)	0.23	268.2	1949.7	18698.5	1.1094	-69.4	2626
Distance	0.23	268.0	1946.7	18671.4	1.1093	-69.4	2588
Optim. (0.2)	0.32	258.4	1875.2	16319.1	1.0978	-69.7	4132
Optim. (10)	0.24	259.0	1883.3	16503.8	1.0987	-69.7	2900

optimize all latency metrics and the average number of transmission attempts across all segments. However, this is done at the expense of a bigger overhead due to the large number of performed handovers, which worsens the PLR. By increasing the minimum switch interval to $10\,\mathrm{s}$, it is possible to drastically reduce the PLR to a value that is comparable with the other solutions, while improving all the other metrics.

VI. LIMITATIONS

WifiRoam is in a preliminary development stage and currently presents several limitations. The first limitation is that packet queuing is not implemented, and therefore, packet transmissions are treated independently of one another. This aspect does not affect the presented results, since the traffic pattern that was modeled with ns-3 (that resembles industrial control applications) does not lead to queuing effects. In the case queuing is present, it is possible to include in the datasets the related contribution to latency, and analyze it as a separate metric. However, this approach would still not take into account the dependencies between consecutive packet transmissions, as happens when, for example, interfering traffic leads to multiple retransmissions and further delays the queued packets. Addressing this problem would require non-negligible modifications to the trajectory component, in order to refine how packet transmissions are modeled.

A second limitation is the fact that, at present, it is possible to simulate a single mobile SUT. The presence of multiple moving STAs would lead to constantly varying interfering effects, depending on their relative positions. This can be partially addressed by modifying the environment component to take into account these effects by altering the statistical distribution of the dataset samples.

Finally, WifiRoam should also be extended to simulate incoming data exchanges for the SUT, non-linear trajectories (that now can be approximated), and different traffic patterns.

VII. CONCLUSION

This work presents WifiRoam, a simulator designed to simplify the study, development, and testing of handover algorithms for industrial Wi-Fi networks. It is not meant to replace general-purpose Wi-Fi simulators, such as ns-3 and OmNet++, but to be used as a complementary tool. Once an environment is defined to describe spectrum conditions (this can be achieved via traditional simulation software programs or by collecting real-world data), handover algorithms can be quickly and repeatedly tested without running full-fledged protocol simulations. Its simplicity, smaller codebase, and the fact that it is written in Python, a high-level programming language designed for code readability, permit to quickly modify, improve, and correct algorithm implementations and test new ideas without requiring excessive development efforts. When a solution is found that is deemed interesting and well-performing, it can (and should) be further validated using traditional full-fledged tools. Finally, the design of WifiRoam allows seamless integration into network digital twins, since the environment model can be shaped and continuously updated using data collected in real-time from the network.

Experiments were conducted in which different handover algorithms, i.e., `RSSIRoaming`, `DistanceRoaming`, and `OptimizedRoaming`, were implemented, tested, and compared using multiple configurations of their parameters. Among the others, `OptimizedRoaming` is an example of a simple handover strategy that relies on an NDT, which WifiRoam allowed to easily implement and study.

Future work will focus on overcoming the current limitations of the simulator by adding support for packet queuing, simulation of incoming traffic, multiple moving STAs, non-linear trajectories, and more complex traffic patterns. Effort will be spent in designing more sophisticated NDT-based handover algorithms to be implemented and tested using the WifiRoam simulator.

References

[1] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2024 (Revision of IEEE Std 802.11-2020)*, pp. 1–5956, 2025. [Online]. Available: https://doi.org/10.1109/IEEESTD.2025.10979691

[2] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong, "Digital Twin of Wireless Systems: Overview, Taxonomy, Challenges, and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2230–2254, 2022. [Online]. Available: https://doi.org/10.1109/COMST.2022.3198273

[3] N. P. Kuruvatti, M. A. Habibi, S. Partani, B. Han, A. Fellan, and H. D. Schotten, "Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 112 158–112 186, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3215493

[4] X. Wang et al., "AIML use cases for IEEE 802.11," *AIML TIG Technical Report*, November 2025, accessed: 2026-01-30. [Online]. Available: https://mentor.ieee.org/802.11/dcn/24/11-24-1650-05-aiml-aiml-sc-1st-technical-report-draft.docx

[5] J. Pulido, S. Fortes, and R. Barco, "Digital Twin-Based in Next-Generation Wi-Fi Networks: Survey and Future Challenges," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, 2025, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WFCS63373.2025.11077636

[6] M. I. Sanchez and A. Boukerche, "On IEEE 802.11K/R/V amendments: Do they have a real impact?" *IEEE Wireless Communications*, vol. 23, no. 1, pp. 48–55, 2016. [Online]. Available: https://doi.org/10.1109/MWC.2016.7422405

[7] S. Sudhakaran, I. Ali, M. Eisen, J. Perez-Ramirez, C. Cazan, V. Frascolla, and D. Cavalcanti, "Zero-Delay Roaming for Mobile Robots Enabled by Wireless TSN Redundancy," in *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*, 2023, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WFCS57264.2023.10144124

[8] B. Mecklenburg, A. H. Ansari, B. Richerzhagen, M. Bahr, and G. Carle, "Seamless Roaming based on Distributed Multi-Link Operation over IEEE 802.11bn," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, 2025, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WFCS63373.2025.11077616

[9] S. Scanzio, M. Rosani, G. Formis, D. Cavalcanti, V. Frascolla, G. Marchetto, and G. Cena, "Multi-Link Operation and Wireless Digital Twin to Support Enhanced Roaming in Next-Gen Wi-Fi," in *2024 IEEE 20th International Conference on Factory Communication Systems (WFCS)*, 2024, pp. 1–4. [Online]. Available: https://doi.org/10.1109/WFCS60972.2024.10540931

[10] R. V. Bhat, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "In-Band Network Telemetry-Based Congestion Control Algorithm for Industrial Wireless Networks," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ETFA61755.2024.10711064

[11] G. Formis, G. Cena, L. Wisniewski, and S. Scanzio, "Accurate and Efficient Prediction of Wi-Fi Link Quality Based on Machine Learning," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2025. [Online]. Available: https://doi.org/10.1109/TII.2025.3609224

[12] S. Sudhakaran, J. Perez-Ramirez, D. Cavalcanti, C. Cazan, N. Olson, R. Rosales, and V. Frascolla, "Wireless Network Digital Twin Calibrated by Real Time Telemetry and XR Feedback Interface," in *2024 IEEE 20th International Conference on Factory Communication Systems (WFCS)*, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WFCS60972.2024.10540718

[13] M. Carrascosa-Zamacois, L. Galati-Giordano, F. Wilhelmi, G. Fontanesi, A. Jonsson, G. Geraci, and B. Bellalta, "XR Streaming Performance With Wi-Fi 7 Multi-Link Operation," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 6477–6490, 2025. [Online]. Available: https://doi.org/10.1109/OJCOMS.2025.3594950

[14] M. Wojnar, W. Cieżobka, A. Tomaszewski, P. Chołda, K. Rusek, K. Kosek-Szott, J. Haxhibeqiri, J. Hoebeke, B. Bellalta, A. Zubow, F. Dressler, and S. Szott, "Coordinated Spatial Reuse Scheduling With Machine Learning in IEEE 802.11 MAPC Networks," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 11, pp. 3666–3682, 2025. [Online]. Available: https://doi.org/10.1109/JSAC.2025.3584555

[15] E. Ferrari, D. Cavalcanti, V. Frascolla, and R. Rosales, "Multi-Parameter Machine Learning Enhanced Wi-Fi Handover in Dynamic Trajectory Simulations," in *2025 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2025, pp. 1–7. [Online]. Available: https://doi.org/10.1109/CSCN67557.2025.11230743

[16] E. Ferrari, D. A. Cavalcanti, V. Frascolla, and R. Rosales, "Ai-Enhanced Seamless Handover in Wi-Fi Networks With Multi-Link Management," in *2025 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2025, pp. 1–7. [Online]. Available: https://doi.org/10.23919/SoftCOM66362.2025.11197451

[17] P. Chiavassa, S. Scanzio, and G. Cena, "On the Suitability of Wi-Fi for Interconnecting Moving Equipment in Industrial Environments," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, 2025, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WFCS63373.2025.11077620

[18] ——, "Wi-Fi Rate Adaptation for Moving Equipment in Industrial Environments," in *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2025, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ETFA65518.2025.11205662