Wireless Network Digital Twin Calibrated by Real Time Telemetry and XR Feedback Interface

Susruth Sudhakaran †,Javier, Perez-Ramirez †, Dave Cavalcanti†,
Cazan Cosmine †, Olson, Nicholas†, Rosales, Rafael †, Valerio Frescolla
*Intel Labs, Intel Corporation
†Intel Labs, Intel Corporation Hillsboro, OR, USA
Emails: {susruth.sudhakaran,javier.perez-ramirez, dave.cavalcanti, cosmin.cazan,nicholas.olson, rafael.rosales,valerio.frescolla }@intel.com

***Abstract*—Managing Wireless networks, particularly in industrial and factory environments, to meet the escalating demands of time critical applications have become more and complex and warrants proactive management strategies. This paper introduces an innovative approach to wireless network management and optimization enabled by a Digital Twin designed and continuously enhanced by real-time device telemetry and user inputs through an Extended Reality interface. By collecting real telemetry data from network devices, our methodology defines and calibrates a digital twin representation of the network, enabling accurate prediction of signal properties and network performance based on simulation models. The model serves as an automation tool to analyze various scenarios, allowing for informed adjustments to use applications, devices and network configurations. The paper describes a real-life implementation of the Digital Twin of a wireless system in real enterprise network scenario, demonstrating improved performance and user experience. The methodology not only addresses the challenges of network optimization but also contributes to the advances of real-time wireless network management based on Digital Twins.**

***Index Terms*—Wireless, Digital Twin, IEEE 802.11, WLAN, Wi-Fi Simulations**

I. INTRODUCTION

A. Background and Motivation

As wireless applications become more demanding, the challenges associated with managing and optimizing wireless networks have also become more pronounced. Wireless connectivity and mobile standards are evolving and introducing new capabilities, such as Time-Sensitive Networking (TSN) and Ultra-Low Latency High Reliability (URLLC) features to ensure data delivery with bounded latency for time-critical systems and high quality user experiences [1] [2]. New wireless capabilities are also increasing complexity of network management and optimization/configuration tasks.

The advent of Digital Twin and AI capabilities have the potential to enable new network management automation approaches where networks can use telemetry and AI to optimize performance in real-time. Digital Twins, based on virtual models of physical systems, telemetry and optimization algorithms, have been adopted across several industries as a platform for decision making and optimization of engineering processes and systems [3]. Recently, Digital Twin concepts have also been adapted for wireless networks [4] [5] as it offers new capabilities to capture complex device and network behavior as well as dynamic conditions increasing the accuracy of predictive analytics and simulations leading to more efficient resource management and better quality of experience.

B. The Role of Digital Twin Technology in Wireless Network Management

Traditionally, deployment of industrial wireless networks involves significant design, planing and configuration efforts to achieve optimized experiences and performance. In practice, typical wireless network management is still reactive, mainly responding to issues that are reported by users or devices. Emerging and future industrial control and automation systems are expected to rely even more on networks to leverage advanced Edge/Cloud computing resources. As such, managing wireless resources has become a core challenge to enable stable operation of Edge/Cloud-based control and automation [6]. In complex industrial environments, network performance can be impacted by dynamic and stochastic factors, such as varying channel conditions, interference, changes in the environment (e.g. movement of users/infrastructure/machines, etc). The ability to predict and preemptively address network performance bottlenecks caused by such dynamics becomes paramount, especially when applications are time-critical and running across a wireless network. Wireless networks are expected to become self-configuring and proactive-online-learning systems and the integration of Digital Twins into network management presents a promising avenue for achieving these goals [4].

Digital Twins, originally conceptualized in manufacturing and industrial settings [7], have found a natural fit in the realm of wireless network management and are being considered as a key component of the sixth-generation (6G) wireless systems [8]. By creating a virtual counterpart of the physical network, it is possible to analyze various scenarios, predict performance outcomes, and proactively optimize the network configuration. The basic concepts and envisioned architecture of digital twins for wireless systems are described in [4] and [8]. The Digital Twin includes a physical interaction layer a a Twin object layer. The physical interaction layer deals with connectivity interfaces with end devices, network and computing infrastructure components to collect telemetry and control/configure specific device/infrastructure capabilities. The Twin objects are virtual representations of a physical system or process and can be built based on modeling, simulation or data-driven learning approaches. The Digital Twin also interacts with an applications (or service) layer, which includes the actual business processes and applications of interest that define the requirements for the network.

Digital Twins for wireless networks are gaining significant attention, but its real deployment is still at very early stages. While several comprehensive surveys and vision/architectural designs have been recently published highlighting potential directions and open challenges [4] [8], practical deployments experiences and real-world testbed capabilities specific to wireless systems have been limited to simple visualization of system/network data [5] and wireless channel performance emulation [9]. Integration of digital twins with Extended Reality (XR) interfaces have been explored in the literature [10], [11], but not in the context of wireless networks. Adding an XR interaction layer with the end user would provide additional information to the wireless digital twin, further improving the modeling of digital twin objects.

C. Contributions and Paper Organization

This paper contributes to the evolving field of Digital Twin for wireless network management by presenting a comprehensive methodology for implementing a Digital Twin of a wireless network and a real-life implementation of this methodology applied to a Wi-Fi network deployed in an real-world enterprise environment. Our proposed methodology includes physical interaction layer interfaces that collect telemetry data from real network devices, and a network twin object model that combines a network simulation model that is calibrated based on the telemetry data. The network twin object is then used to predict network behavior that impacts certain performance metrics. The ultimate goal is to leverage these predictions to drive informed changes in the configuration of the network or individual devices, thereby improving overall network performance and user experience. The contributions of this paper are as follows:

1) We describe a methodology to collect data for integrating into a digital twin model through distribution of ”probe” nodes.
2) We provide a method for the integration of real data collected into a simulation model so as to ensure and improve the accuracy and relevance of the simulation model, enhancing the reliability of predictions and the effectiveness of subsequent network optimizations.
3) We demonstrate an XR interface that allows human to provide feedback to tune the models.
4) We demonstrate the practical application of our approach in a real-world WiFi network deployment.

D. Organization of the Paper

The remainder of this paper is organized as follows: Section II provides a general overview of Digital Twins, typical ingredients and architecture and its applicability in the context of wireless network optimization. Section III details our implementation methodology, including data collection, calibration of the simulation model, and predictive analysis. Subsequent sections describes application of this method to optimize a simple use case , followed by a discussion of challenges, lessons learned, and future directions. The paper concludes with a summary of findings and their implications for the field of WiFi network management.

II. Overview of Digital Twins of Wireless Networks

This section describes the basic components of Digital Twins of Wireless Systems and some of the main design challenges.

A. Digital Twins of Wireless Networks

Digital Twins of a wireless network can be used to optimize communication resources while addressing diverse, and sometimes, conflicting Quality of Service (QoS) requirements for users and applications. The authors in [8] describe a vision where Digital Twin-enabled 6G networks are self-sustaining, and proactive-online-learning based wireless systems that can meet highly dynamic, and extreme latency, reliability and throughput requirements. According to the taxonomy in [4], Digital twins of wireless systems take inputs from real world devices to create virtual representations of a wireless network, using tools from optimization theory, game theory, and machine learning, to make predictions and/or control decisions. As illustrated in Fig. 1, a Digital Twin of a wireless network consists of the following building blocks:

1) Physical System Interaction Layer: This layer provides interfaces to collect telemetry from the physical system and to configure the physical system. The physical system includes user devices, network devices, and any relevant infrastructure component of the wireless system that is being considered. The interfaces may provide access to various device state parameter. The interfaces may also include access to device/network configuration parameters that determine the behavior of the system. For instance, device and link state parameters may include Received Signal Strength, achievable data rates, latency statistics, etc. Configuration parameters may include transmit power, operational bandwidth, medium access control (MAC) configuration parameters, traffic shaping configurations, etc.
2) Twin Object Layer: this layer includes the twin objects representing the state and/or behavior of the physical system of interest. The twin objects are the core of the Digital Twin implementation and they may be developed based on mathematical, physics-based or simulation models that capture the relevant behavior of the physical system that is being modeled. Multiple twin objects may be developed for different aspects (or sub-systems) of a physical system. The twin objects are used to perform predictions based on scenarios of interest and

provide as outputs information that can be converted into configuration parameters of the physical system.

3) Service and Management Layer: the service layer includes interfaces for interacting with, managing and accessing the Digital Twin capabilities. This layer may be used by autonomous devices to make use of the Digital Twin predictive analytics and scenario optimization capabilities. For instance, a mobile device may access a wireless network Digital Twin as it starts a roaming procedure to query about best candidate neighboring access points to connect to in order to minimize data delivery latency. The service layer also provides an interface for network managers to configure and update the Digital Twin models. As discussed in the following sections, keeping the twin object models up to date is a challenge, especially in dynamic environments.

Implementing and applying existing Digital Twin visions [8] [4] to practical wireless systems and their applications involve many open research questions. This work focuses on network twin object design, prototyping and accuracy challenges including aspects related to dynamic environment changes. The following sections discuss the challenges related to accuracy of network twin objects and issues caused by dynamic environment changes.

B. Twin Object Fidelity

Twin objects of wireless systems can be designed by mathematical, experimental or data-driven modeling, or a combination of these modeling approaches [4]. Mathematical models are widely in evaluation of wireless communication systems, but they are typically based on generic assumptions that may not always valid for a specific real-life scenario. Experimental data has also been widely used to model wireless signal propagation [12]. While widely used in the design of wireless systems and standards, such generic channel models try to represent typical scenarios and may not accurately reflect specific properties of a complex environment such as an industrial plant or manufacturing environment. Data-driven models are emerging as a promising alternative to improve accuracy of twin objects, although it also requires data for training, which may represent a challenge. In order to achieve real performance gains by using the digital twins of a wireless system, it is necessary that the twin object represents the wireless system with enough fidelity. Combination of multiple design methods, such as mathematical models augmented by experimental or data-driven insights, is expected to provide the best results.

C. System and Environment Dynamics

The fidelity of the twin object also highly dependent on the ability of the twin object to capture system and environment dynamics that may impact the system behavior. For instance, in a factory environment where machines, metal objects and people are continuously moving, wireless channel conditions and therefore link capabilities (e.g. capacity/throughput, error

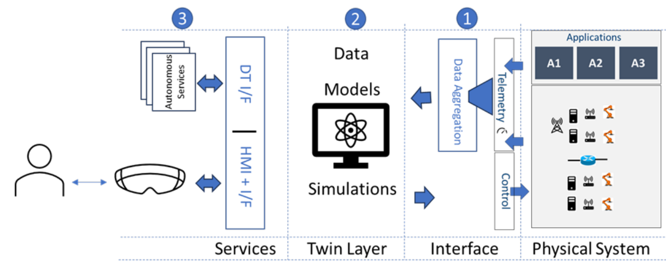

Fig. 1. Template building blocks architecture of the Digital Twin

rate, latency, etc.) may dynamically change. Although mathematical and experimental models of wireless channels do account for certain dynamics, modeling specific events and updating the models in real time is not done in practice. The digital twin capabilities enables the real-time feedback into the twin object of the wireless system. The next sections describe a design methodology for a digital twin of wireless network that includes real-time telemetry and an Extended Reality interface to update a twin object in real-time to maintain high fidelity.

III. Wireless Digital Twin Implementation Methodology

Fig. 2 shows the high level layout of the Wireless Digital System components and interfaces used in the real-life implementation developed as part of this work. The rest of this section describes the various blocks of the system and interfaces implementation choices used in the paper.

A. Physical System

The physical system is the system or environment that the Digital Twin is trying to model and optimize. This is expected to provide interface to collect observations of various state parameters of the system. The physical system is also expected to have configuration interfaces and control knobs that can be activated for controlling the state of the system. The Physical System in our implementation consists of a representative warehouse and enterprise environment, part of real building, where connectivity is provided by a Wi-Fi network. We use a mobile robot as a representative use case where the robot is configured to navigate through the environment while maintaining connectivity with the Wi-Fi network. The mobile robot executes emulated tasks that generate traffic flows that require a certain level of QoS from the Wi-Fi network. As in any real wireless network, the QoS is tightly coupled to the received signal strength and a minimum transmission rates achievable in the link between the robot and its serving Access Point (AP).

1) Mobile Probes: The interaction interface with the physical system (e.g. mobile robots and other devices on the wireless network) is implemented by mobile ”probe” application that provides telemetry from the Wi-Fi stations (STA). The telemetry is collected in real-time at multiple points along the robot’s path and it includes Received Signal Strength, Transmit Data Rates, Received Data Rates, connectivity/link status, current associated AP. Telemetry is also collected from other devices on the network (e.g. multiple user compute platforms). The probe also collects other metrics related to its compute platform performance, however they have not been used the digital twin implementation in this paper. This is accomplished by placing ”probes” or sensors at strategic points in the physical environment. These sensors measure various parameters w.r.t. the system we are trying to mimic using the Digital Twin. In this case we measure the received signal quality of all the Access Points and associated channels at various points in the deployment environment by scanning the environment and reporting the results at periodic intervals. The probes can also be active probes in the sense that they can be configured to actively send out active traffic probes to assess quality of transmission. The sensors themselves are placed at points that are important from the usage of the network standpoint. The sensors need not be actual physical hardware, they can be a software service running gathering configured relevant metrics from various locations and systems within the environment. he metrics collected mainly falls into two categories - operational parameters from the infrastructures and perceived QoS metrics from the infrastructure as well as the various probes and nodes in the environment. The collected metrics includes perceived metrics representing QoS at various locations in the environment through the various sensors, which include metrics like Received Signal Strength,receive and transmit rates at the various sensors, connectivity states and location related information. The system also collects operational parameters from the various Access Points and this includes transmit power, channel utilization as perceived by the Access Point, connectivity states as perceived by the access points etc.. All this metrics and operational configurations

This data provides a reference to the Digital Simulation model to adjust its parameters to correct for errors in its estimation.

B. External Interfaces

1) HMI Interface: This interface defines mechanism for interaction with the Digital Twin models and data including AR/VR visualization and overlay. A human operator can also annotate or manually fine tune the models with information that is not available to the models.

2) Digital Twin Services Interface: Autonomous agents and services can utilize these interfaces to exercise the digital twin models and query the model to simulate and estimate behavior of the system for hypothetical scenarios. The results and insights from this analysis can then be used by the agents to optimize the system proactively and intelligently in response.

In this paper we put together a Digital Twin implementation for optimizing the Wireless Network performance of a representative mobile application in a representative warehouse environment. Fig. 2 shows show a high-level system components view of the Digital Twin system implementation that was put together following the general building blocks template discussed in the previous section. The Digital Twin strives to model the quality of the network performance available to

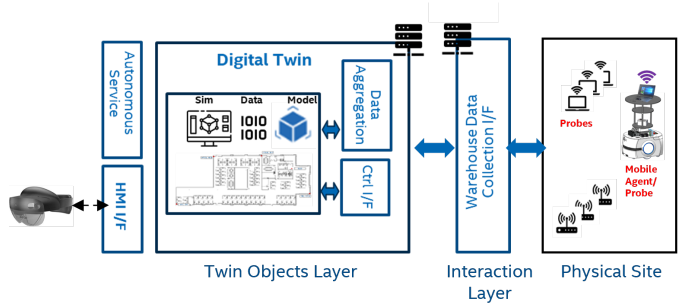

Fig. 2. Digital Twin System Implementation

a mobile client in the environment and it tries to optimize the performance of the representative mobile application by recommending routes within the environment that would meet the QoS requirements of the application.

C. Twin Object Layer

In this layer the physical system or an aspect of the physical system is modelled so as to understand the behaviour over time and use this understanding to proactively implement optimizations or adjustments to improve performance of aspects of the system. Here we are trying to model how the Wi-Fi environment affects the performance of specific mobile application and then use this model to understand the behaviour under specific scenarios and then proactively detect and configure performance degradation. To model the Wi-Fi environment and its performance, we initialize a Wi-Fi simulation model with environment configuration mainly the site dimensions, Wireless Access Points, and their locations as well as the various configuration and operational parameters of the wireless deployment including channels, transmit power, Wi-Fi features supported etc. The measurement data w.r.t performance metrics collected periodically from the environment is then compared with the results from simulation to calibrate the simulation model. The calibrated simulation model is then used to estimate performance that can be achieved by the mobile client across various hypothetical routes in the environment and based on this data, an optimal route is recommended.

The simulation model employed is a simple path loss based WiFi link simulation model. The model takes as input the deployment site dimensions, the location of Access Points, location of all probes and agents in the environment and wireless configuration of the agents and AP’s respectively. Based on the input the, simulation model builds a grid based wireless performance model of the site, where the entire site is divided into cells of a configurable dimension. The model also takes as input routes of interest within the site. A route is made up of a sequence of cells that a mobile agent, for example, would traverse to move from one point to another while it is accomplishing a representative task. The simulation model estimates the received signal strength of every access point in each cell in the route for all routes. Based on this estimate the model also computes an estimate of the expected QoS score that can be experienced by mobile agent in these cells for all the configured routes. Fig. 6 shows an example of the output of this estimate for a sample set of routes in the environment. The model starts out using path loss model assuming free space loss to calculate the received signal strength in the cells of interest using the AP transmit power.

$$RSSI_{(est,AP,cell)} = TxPwr_{AP} - 10Nlog(d_{cell}) - error_{cell} \tag{1}$$

As the real measurement data comes in, the model, first, uses regression to estimates the path loss exponent at the cell, thereby getting a measure of deviation from free space.

$$N = (TxPwr_{AP} - RSSI_{Meas})/10log(d_{cell}) \tag{2}$$

Secondly, to estimate the path loss in cells where real measurement is not available, the model uses the nearest cell where measurement is available as reference points to estimated the path loss and thr received signal strength.

$$PL_{(est,AP,cell)} = PL_{nearest-ref-cell} - 10Nlog(d_{cell}/d_{nearest-ref-cell})$$

Based on this signal strength estimate and correlation of the signal strength with the receive and transmit rate at each cell, a QoS score estimate for all mobile agents is estimated for all cells in all configured routes. The QoS score estimation also takes into account the channel utilization value reported by the APs along the path, which gives an indication of congestion in the AP. Each cell also maintains an error value indicating deviation from the measured value when a measurement for that cell is available. This error is then applied to the estimation process. This simulation and estimation is continuously carried out whenever a new measurement data is available. As new measurement data becomes available for a particular cell, its estimates are adjusted and past estimates are aggregated. This way the model estimation error is minimized at every iteration step and at any point in time, the model maintains an estimate of expected QoS along all the configured routes for all stations in the system. This information can be used to proactively identify a degradation in expected QoS and take necessary action.

IV. Use Case/application/ Case Study

In this section we describe the representative environment and use case to evaluate the Digital Twin system. We selected a wireless testing environment where we configured a representative uses. The use case is that of a mobile agent connected to the Wi-Fi network moving from one point to a goal location inside the environment emulating a robotic agent accomplishing a task. There is also traffic configured to run between the mobile agent and load server emulating camera traffic for example. In order for the agent to complete its task successfully, it has to move from one point in the environment to the goal location while at same time maintaining a specific level of traffic performance. The agent continuously does this and it has a choice of routes it can take. A Wi-Fi deployment

Fig. 3. QoS Route Estimates calculated by the Model

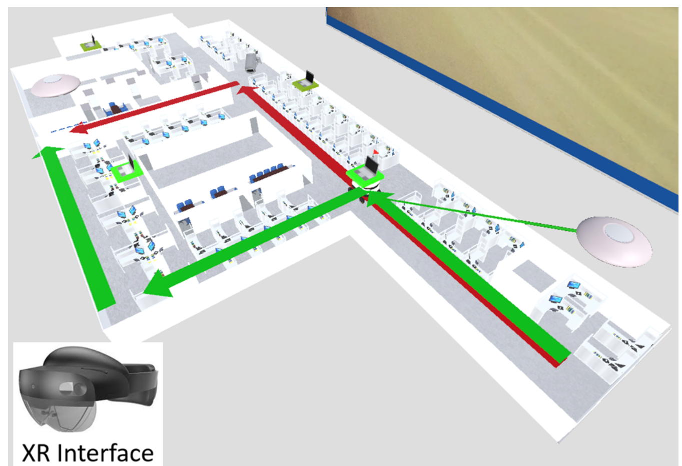

Fig. 4. QoS Route Estimates calculated by the Model

Fig. 5. QoS Route Estimates calculated by the Model

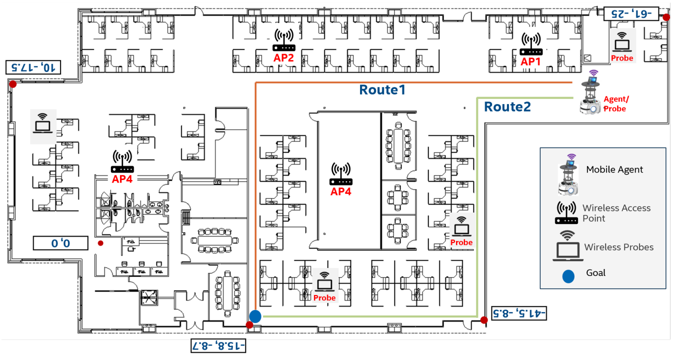

Fig. 6. Casper Site Layout and routes

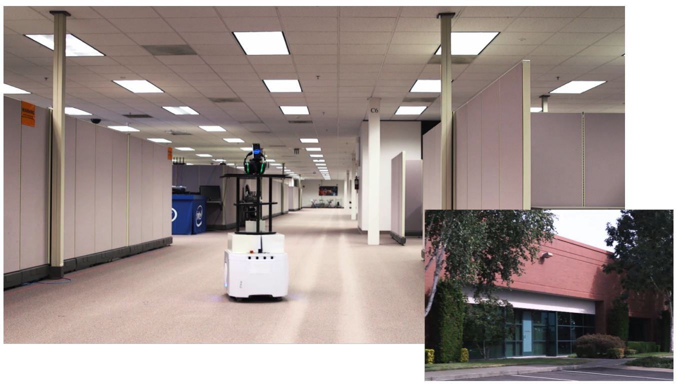

Fig. 7. Casper Site

consisting of 4 Access Points provide connectivity to the mobile agent. Although there is only one mobile agent, load can be generated in the network to simulate other agents in the environment via sensor nodes placed at various points in the environment. The sensor nodes as well as the agent node also has sensing software that collects information on the signal and performance characteristics of the wireless environment and relays it back to a back-end server where this is aggregated along with other telemetry coming from all the Access Points in the environment. A snapshot of this aggregated measurement data is send out to a Digital Twin server located at a remote location. The Digital Twin derives an estimates of the Quality of Service that can be achieved by the mobile agent for all the choice of routes that is available to the agent and at any given point recommends a route to take that has the expectation of best achievable QoS. In order to evaluate the Digital Twin Model, we compare the estimate of the performance that the model has predicted for a specific route with the actual measured performance achieved as the agent navigates that route. Since the Digital Twin is continuously calibrating and improving the accuracy of its model, we also measure how many iterations the model takes to achieve estimates on par with the actual measurements.

A. Evaluation Procedure

The model calibration is evaluated both in terms of how many iterations of calibration is needed to get an accurate

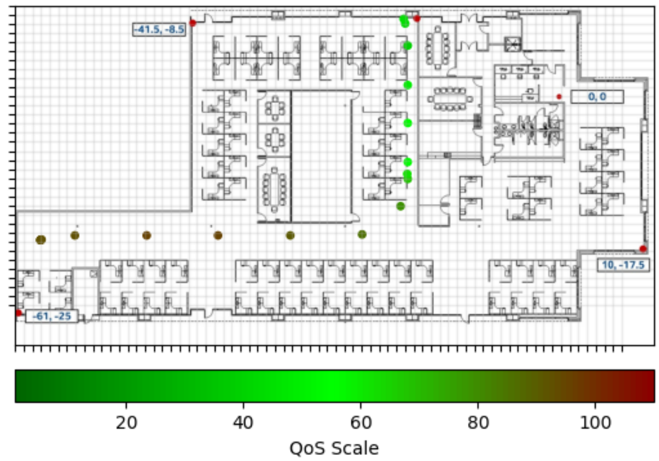

Fig. 8. Route 1 QoS Scores from Measured metrics

enough calibration the matches the real results as well as the accuracy of the model estimation in terms of deviation from the real measurements. We let the mobile agent execute its task by explicitly making it take the various routes and randomly collecting measurements along the routes. These measurements are then fed into the model which simulates the same scenario but compares it with the real results and iteratively adjusts its parameters until the error between the real results and the measured values are minimal. We repeat this experiment for all the routes. At the end of the process we would have a calibrated model which we can then use for basing decisions on.

After the model is calibrated, we run the scenario while randomly creating congestion in one of the Access Points along one of the routes. During this step the model is used to decide which route is best to take adn we evaluate if the model can accurately predict the route where there is no congestion.

V. Results and Analysis

After letting the agent execute task over route 1 we collect the measured metrics from all the sensors and we evaluate a QoS score for each point along the route. We then feed this measured values to the Digital Twin model for calibrating the model. The model simulates agent running task over route 1 while at the same time comparing and adjusting its parameters using the fed measured values. Then we compare the QoS score computed by the model to the another set of measurements. Fig. 8 shows scores calculated using the measured metrics and Fig. 9 shows the scores computed by the model. The model will also estimate scores for all the points for which measurements were not available bur are in the path and between the measured points. We can see that the model, after calibration, has been able to estimate expected quality of service scores that are fairly close to the actual measured values.

As we feed the measured values into the model we also extract the predictions from the model and the error with the

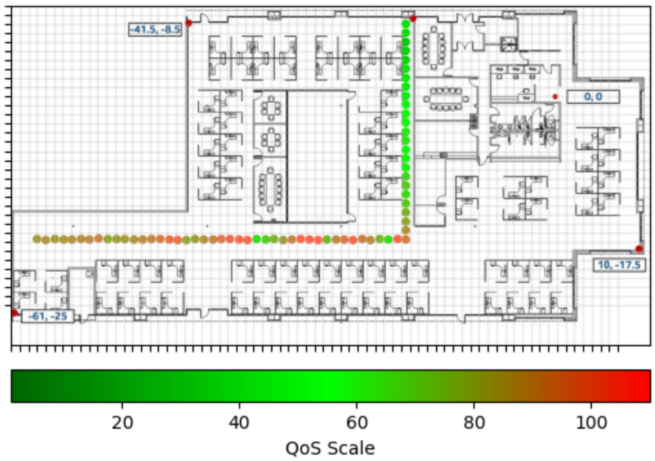

Fig. 9. Route 1 QoS Scores predicted and extrapolated by model

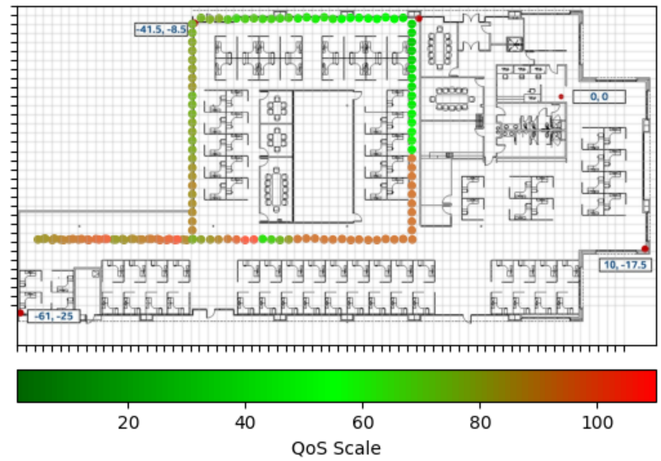

Fig. 11. Route Performance prediction when there is Congestion

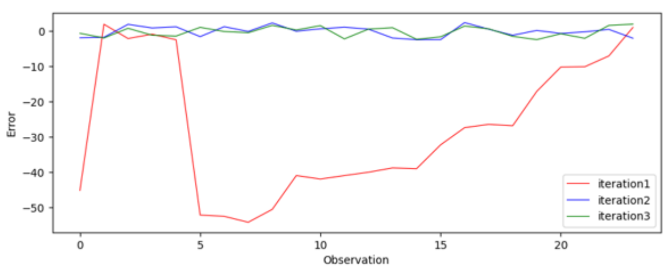

Fig. 10. Route 1 model calibration iterations

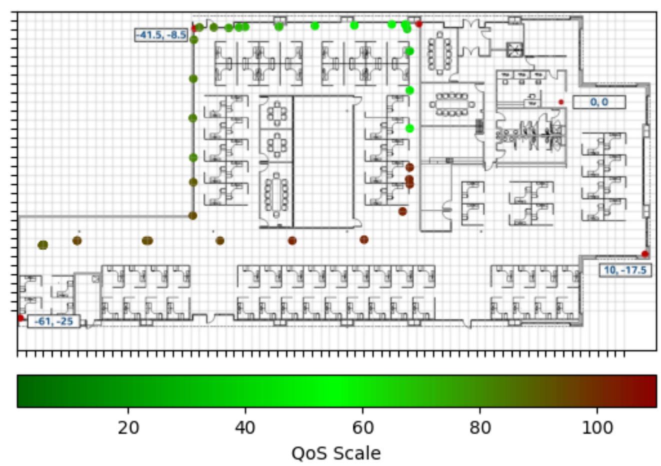

Fig. 12. Actual measured performance during congestion

real values and compute number of iterations required to arrive at an acceptable minimum error value. Fig 10 shows that for route 1 the model took about three iterations to bring the error value to minimum.

For evaluating the decision making and recommendation capability of the Digital Twin model, we add congestion along one of the routes as the mobile agent starts its task and assess whether the model is able to predict the degradation along the route and provide alternative route, which can be chosen by a user interacting over the HMI or by an autonomous service managing the use case. We then repeat the experiment over both the routes with congestion enabled and compare the results.

Fig 11 shows the estimation produced by the Digital Twin model of expected performance along both the routes with congestion along route 1. Here we can see that the model was able to pick up the increase in channel utilization from the operational telemetry of the Access Point and was able to predict degradation around the congested AP.

Fig 12 shows the estimation produced by the Digital Twin model of expected performance along both the routes with congestion along route 1. Here we can see that the model was able to pick up the increase in channel utilization from the operational telemetry of the Access Point and was able to predict degradation around the congested AP.

VI. Conclusion

We have shown how a Digital Twin model of a Wireless Network environment and associated use case, calibrated using real telemetry coming from the physical system can be used in real time to proactively and autonomously detect performance degradation and suggest configuration changes to improve and optimize the system performance. We have show how ther Digital Twin model can be evaluated by comparing real performance with estimates of performance generated by the model as well as measure calibration time required to improve the model's accuracy. We have also shown how a HMI interface can be used to interact with the Digital Twin model and adjust the model based on out of band information that is available to an operator occasionally. Although we have demonstrated the concept by integrating a simple simulation model and calibrating it with real data in a representative test environment, the concept can be scaled and used in real world scenarios like an enterprise IT network or an industrial wireless network to proactively manage the performance of applications over the network. Next steps to this work may include expanding upon this concept and incorporating coordination of multiple application traffic across all the nodes in a wireless environment by first modelling the application behaviour, calibrating the model based on observation and then use the model to coordinate the traffic flows for improved quality of service.

ACKNOWLEDGMENTS

This work has been partially funded by the European Commission Horizon Europe SNS JU PREDICT-6G (GA 101095890) Project.

REFERENCES

[1] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending Accurate Time Distribution and Timeliness Capabilities Over the Air to Enable Future Wireless Industrial Automation Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1132–1152, 2019.

[2] J. Perez-Ramirez, O. Seijo, and I. Val, "Time-critical IoT applications enabled by Wi-Fi 6 and beyond," *IEEE Internet of Things Magazine*, vol. 5, no. 3, pp. 44–49, 2022.

[3] M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi, "A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future," *IEEE Access*, vol. 11, pp. 17471–17484, 2023.

[4] 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.

[5] J. Zhao, X. Xiong, and Y. Chen, "Design and implementation of a cloud-network resource management system based on digital twin," in *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pp. 1–5, 2023.

[6] M. Eisen, M. M. Rashid, K. Gatsis, D. Cavalcanti, N. Himayat, and A. Ribeiro, "Control aware radio resource allocation in low latency wireless control systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7878–7890, 2019.

[7] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–6, 2016.

[8] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6g: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[9] Z. Lai, D. Yuan, H. Chen, Y. Zhang, and W. Bao, "Wirelessdt: A digital twin platform for real- time evaluation of wireless software applications," in *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 146–150, 2023.

[10] S. Jeršov and A. Tepljakov, "Digital twins in extended reality for control system applications," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 274–279, 2020.

[11] D. I. Gonzalez-Aguirre, J. Perez-Ramirez, J. Felix-Rendon, J. F. Leon, J. Bourgault, J. Z. Esquivel, and L. Nachman, "Robot-based uniform-coverage and high-resolution lidar mapping for physically-grounded metaverse applications," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 40–45, 2023.

[12] A. F. Molisch, *Channel Models*, pp. 125–143. 2011.