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

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

***Abstract*—Managing Wireless networks, particularly in industrial and factory environments, to meet the escalating demands of time critical applications has become more complex and warrants proactive management strategies. This work introduces an innovative approach to wireless network management enabled by a Digital Twin (DT) 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 DT representation of the network, enabling accurate prediction of wireless signal properties and network performance based on simulation models. The DT serves as an automation tool to analyze various scenarios, allowing for informed adjustments to user applications, devices and network configurations. The paper describes a real-life DT implementation of a wireless system in a real enterprise network scenario. Experimental results are provided demonstrating improved performance and user experience enabled by the proposed DT-based network management. The proposed methodology addresses the challenges of real-time network optimization and contributes to advance wireless network management based on device and network telemetry.**

***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 constantly evolving thus introducing new capabilities, such as Time-Sensitive Networking (TSN) and Ultra-Reliable Low Latency Communications (URLLC) features to ensure data delivery with bounded latency for time-critical systems and high quality user experiences in a variety of vertical sectors [1] [2] [3]. New wireless capabilities are also increasing complexity of network management and optimization/configuration tasks [4].

The advent of Digital Twins (DTs) 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 [5]. DTs, 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 [6]. Recently, DT concepts have also been adapted for wireless networks [7] [8] [9] as they offer new capabilities to capture complex device and network behavior as well as dynamic conditions. DTs based on real device/network telemetry can increase the accuracy of predictive analytics and simulations, thus leading to more efficient resource management and better quality of experience.

B. State of the Art: Digital Twins 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 [10]. 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). The ability to predict and preemptively address network performance bottlenecks caused by such dynamics becomes paramount, especially when applications are time-critical and distributed across a wireless network. Wireless networks are expected to become self-configuring and proactive-online-learning systems and the integration of DTs into network management presents a promising avenue for achieving these goals [7].

DTs, originally conceptualized in manufacturing and industrial settings [11], have found a natural fit in the realm of wireless network management. DTs are being considered as a key component of the sixth-generation (6G) wireless systems [12]. By creating a virtual counterpart of the physical network, it is possible to analyze various scenarios, predict performance outcomes, and proactively optimize network configuration. The basic concepts and envisioned architecture for DTs of wireless systems are described in [7] and [12]. The DT includes a physical interaction layer and a Twin object layer. The physical interaction layer deals with connectivity interfaces with end devices, network and computing infrastructure to collect telemetry and control/configure specific device/infrastructure capabilities. The Twin object is a virtual representations of a physical system or process and it can be built based on modeling, simulation or data-driven learning approaches. The DT also interacts with an application (or service) layer, which includes the actual business processes and applications of interest that define the requirements for the network.

DTs of wireless networks are gaining significant attention, but real deployments are still at very early stages [13]. Several comprehensive surveys and vision/architectural designs have been recently published highlighting potential directions and open challenges [7] [12]. However, practical deployment experiences and real-world testbed capabilities specific to wireless systems have been limited to simple visualization of system/network data [8] and wireless channel performance emulation [14]. DTs with Extended Reality (XR) interfaces have also appeared in the literature [15], [16], but an effective integration of those concepts in real-time wireless network management is still at a preliminary stage [17]. Enabling an XR interaction layer with the end user adds a new spatial dimension to the visualization of the wireless DT, providing better contextualization of the data generated and observed by the end user [18]. This information can be used to take action in managing the network, providing additional information to the wireless DT and further improving the modeling of Twin objects.

C. Contributions and Paper Organization

This work contributes to the evolving field of DT for wireless network management by presenting a comprehensive methodology for designing a DT of a wireless network and providing 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 devices, and a network twin object model that consists of a network simulation model calibrated by device/network telemetry. 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 and integrate data from the physical system into a DT through distribution of ”probe” nodes across client and network devices.
2) We describe a Twin Object of a Wi-Fi Network that combines simulation modeling and telemetry to predict QoS for client devices. The Twin object uses real device telemetry to enhance fidelity of the simulation models and achieve more reliable DT predictions.
3) We demonstrate an XR interface that allows users to provide feedback to tune the DT models in real-time.
4) We demonstrate the practical application of the proposed methodology with a Wi-Fi network DT that predicts QoS for mobile clients and recommends routes based on QoS predictions.

D. Organization of the Paper

The remainder of this paper is organized as follows: Section II provides a general overview of DTs, typical ingredients, architecture, and applicability in the context of wireless network management and optimization. Section III details our implementation methodology, including data collection, calibration of the simulation model, and predictive analysis. Subsequent sections describe 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 wireless network management.

II. Overview of DTs of Wireless Networks

This section describes the basic components of DTs of wireless systems and some of the main design challenges.

A. Digital Twins of Wireless Networks

A DT 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 [12] describe a vision where DT-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 [7], DTs 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 DT 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 Indicator (RSSI), 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 DT implementation and they may be developed based on

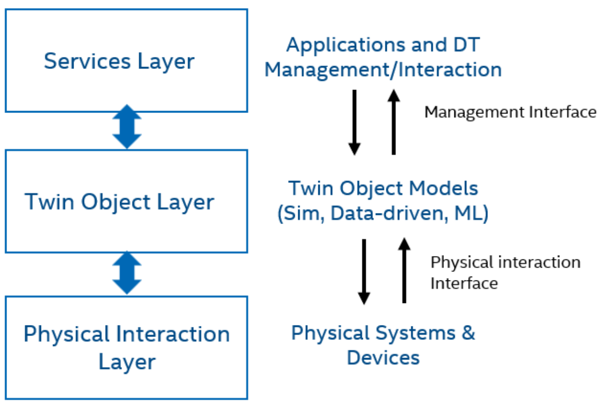

Fig. 1. Digital Twin Components.

mathematical, physical 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. They 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 DT capabilities. This layer may be used by autonomous devices to make use of the DT's predictive analytics and scenario optimization capabilities. For instance, a mobile device may access a wireless network DT as it starts a roaming procedure to request best candidate neighboring access points (APs) to connect to, in order to minimize data delivery latency. The service layer also provides an interface for network managers to configure and update DT 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 DT visions [12] [7] to practical wireless networks and their applications involve many open research questions. This work focuses on network twin object design, prototyping and accuracy including capabilities to capture 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, data-driven modeling, or a combination of these modeling approaches [7]. Mathematical models are widely used in the evaluation of wireless communication systems, but they are typically based on generic assumptions that may not always be valid for a specific real-life scenario. Experimental data has also been widely used to model wireless signal propagation [19]. 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 training data requirements may represent a challenge. In order to achieve real performance gains by using the DT 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 is 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 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 next sections describe a methodology for designing a DT of wireless network that includes real-time telemetry and an XR interface to update a twin object in real-time to maintain high fidelity.

III. Wireless DT Implementation Methodology

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

A. Physical System Interaction through Mobile Probes

The physical system is the system or environment that the DT tries to model and optimize. The physical system is expected to provide interfaces to collect telemetry of various system state parameters. The physical system is also expected to expose configuration interfaces and control knobs for controlling the state of the system. In our implementation the physical system consists of a Wi-Fi network deployed at a real warehouse/enterprise environment shown in Fig. 3. 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. 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 AP.

The interaction interface with the physical system (e.g. mobile robot and other devices on the wireless network) is

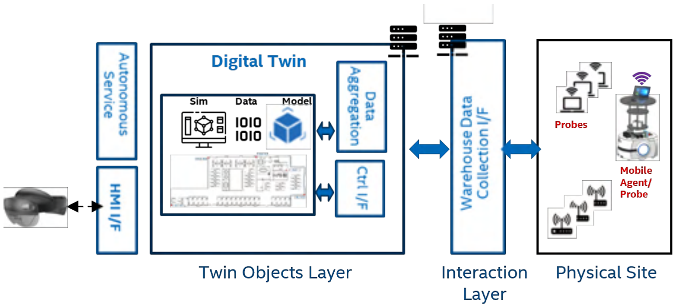

Fig. 2. Architecture of the DT System Implementation.

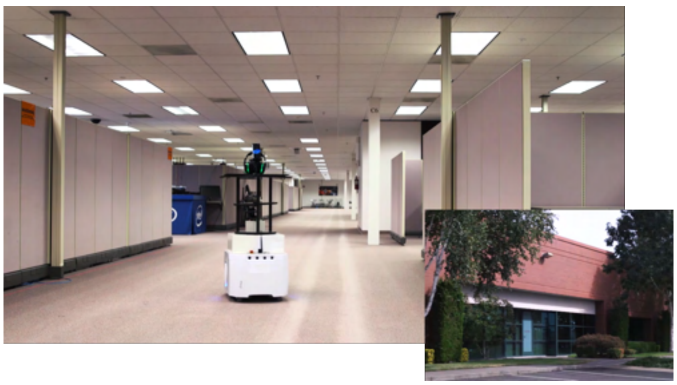

Fig. 3. Intel over-the-air wireless validation site (CASPER). RF controlled office environment (21,000 ft^2).

implemented by a 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 RSSI, transmit data rate, connectivity/link status, and 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 in the DT implementation in this paper. The probes report the results to a central server at periodic intervals, which are configurable. The reporting can also be configured to capture specific locations in the environment and to report results based on events. Furthermore, the probes can also generate synthetic traffic in order to collect the data associated with the quality of the links. In addition to collecting telemetry from the client devices, the probes can also be deployed at the network infrastructure, i.e., APs, to collect operational parameters such as AP transmit power, channel utilization. The mobile probe data is used to build the network twin object model and continuously enable model updates leading to higher fidelity.

B. XR based Human-Machine-Interface

The Human-Machine-Interface (HMI) defines a mechanism for users to interact with the Wireless DT using XR devices. XR offers an efficient way to simultaneously visualize telemetry of the network generated by multiple probes. Figure 4

Fig. 4. XR human-machine-interface. *Probe* application in laptops share their location and RSSI with the digital twin. Information is integrated in XR interface for user visualization. Tile color under laptops represent RSSI values with green denoting high and red low. Association between mobile devices and APs is displayed by green line.

shows an example of the XR interface designed for a real enterprise building, referred in this paper as *CASPER site*, where we can observe the real time telemetry, such as RSSI collected by each of the laptops with *probe* application, by simply looking at the color of the tiles below them. Users can also visualize associations between client devices and APs. Additional features include the capability to summon a terminal or command window to inspect telemetry collected by each probe, and a real time video feed of CASPER site.

A critical aspect of the XR interface is the capability to annotate the environment. Some of the characteristics captured in the original 3D model of the site might change over time. Thus, annotation allows the user to add or change properties of the physical environment for fine tuning the DT over time.

C. CASPER Network Twin Object

We implemented a twin object of the Wi-Fi network, referred here as CASPER network twin object, which models the network performance experienced by client devices in the environment. We have selected a mobile robot application as example to demonstrate the capability of the DT in optimizing network performance. In our example, as the mobile robot moves within the environment, the DT recommends routes that are more likely to meet the QoS requirements of the application traffic.

The main goal in designing the CASPER network twin object is to model the impact of the Wi-Fi network on the performance of applications running on clients under specific scenarios and network conditions. We used a Wi-Fi network simulation model that takes as initial input environment configuration parameters including the site dimensions, devices locations as well as operational parameters of the APs (e.g. operating channels, transmit power, MAC layer configurations). We use measurement data collected from the mobile probes to calibrate the simulation model and improve the fidelity of the twin object. The twin object is then used to estimate performance that can be achieved by the mobile client across various hypothetical routes in the environment.

The CASPER network twin object consists of a grid based wireless performance model, 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 consists of a sequence of cells that a mobile client or a robot, for example, would traverse to move from one point to another while it is accomplishing a given task. In this work, the RSSI is used as metric to measure the QoS, given that RSSI will directly impact the achievable data rate over the wireless link. The twin object estimates the RSSI of every AP in each cell in the route for all routes. The RSSI estimate of a particular AP in a cell is given by

$$\widehat{RSSI}_{(i,j)} = P_i - 10N_j log(d_{i,j}) - \epsilon_j \quad (1)$$

where $\widehat{RSSI}_{(i,j)}$ is the estimated $RSSI$ corresponding to the i-th AP and j-th cell, with $i = 1,..,M$, M the total number of APs, $j = 1,..,C$ and C the total number of cells. P_i is the transmit power of the i-th AP, N_j is the adjusted environmental coefficient associated to the j-th cell, $d_{(i,j)}$ is the distance of the j-cell from the i-th AP, and ϵ_j is the observed error between previously estimated and observed values for the j-th cell. N_j and ϵ_j is initialized to 2, assuming free space propagation, and zero respectively and as we get measurements for a cell these are then adjusted. Every time a new measurement is available in a cell, N_j for that cell is adjusted in an iterative fashion using formulas listed below.

$$N_j = \frac{1}{M} \sum_{i=1,..,M} \overline{RSSI}_{(i,j)}/10log(d_{i,j}) \quad (2)$$

where $\overline{RSSI}_{(i,j)}$ is the measured $RSSI$ between the i-th AP and j-th cell. The adjusted N values are then averaged to get $N_{\text{adj}}(cell)$ for a cell. The $error(cell)$ is calculated using

$$\epsilon_j = \frac{1}{M} \sum_{i=1,..,M} \left(\overline{RSSI}_{(i,j)} - \widehat{RSSI}_{(i,j)} \right) \quad (3)$$

The model then computes an expected QoS score estimate for each cell, which is given by

$$\widehat{QoS}_{(i,j)} = \frac{\widehat{RSSI}_{(i,j)} - min(\widehat{RSSI}_{(i,j)})}{max(\widehat{RSSI}_{(i,j)}) - min(\widehat{RSSI}_{(i,j)})} \quad (4)$$

where $max(\widehat{RSSI}_{(i,j)})$ and $min(\widehat{RSSI}_{(i,j)})$ denote the historical maximum and minimum estimated RSSI respectively.

Let $\{r_1^l, .., r_K^l\} \subset \{1, .., C\}$ be a subset of cell indices describing the l-th route in the DT, then we define the l-th route QoS score as

$$\widehat{QoS}^l = \frac{1}{KM} \sum_{j=1,..,K} \sum_{i=1,..,M} \widehat{RSSI}_{(i,r_j^l)} \quad (5)$$

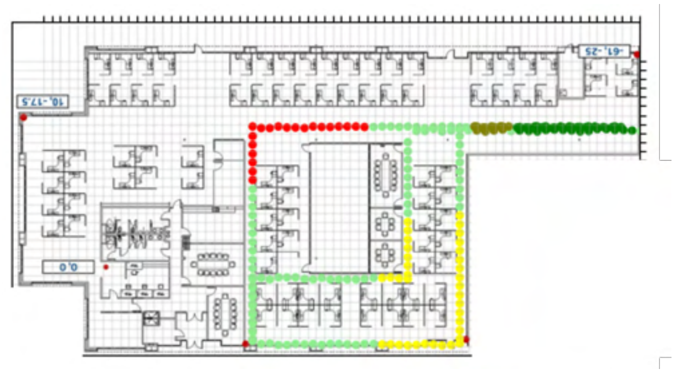

Fig. 5. Example of expected QoS Estimates for all routes calculated by the model.

To estimate the RSSI in cells where real measurement is not available, the model uses the nearest cell where measurement is available as reference to estimate the path loss and the RSSI. Let $j_n \in \{1, .., C\}$ be the index associated to the nearest cell mentioned above, then instead of (1) we consider

$$\widehat{RSSI}_{(i,j)} = \widehat{RSSI}_{(i,j_n)} - 10N_{j_n} log \frac{d_{(i,j)}}{d_{(i,j_n)}} \quad (6)$$

Fig. 5 shows an example of the output of CASPER network twin object for a sample set of routes. This shows an example of the QoS estimate available along the routes specified in 6. Each point is color coded with red representing lowest possible QoS score and dark green representing the highest possible QoS score. The model starts assuming free space path loss to calculate the RSSI in each cell and it is updated over time by incorporating telemetry from the mobile probes as they become available. As the real telemetry comes in, the twin object first uses regression to estimates the path loss exponent at the cell (2), thereby getting a measure of deviation from free space simulation model. Secondly, a QoS score is estimated for all cells in all configured routes as described in equation 5. The QoS score estimation for a cell also takes into account the channel utilization value reported by the APs as a cost that is also applied to the QoS score in (4). This simulation and data-driven calibration 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 twin object 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.

Other network twin object models can be built using different simulation modes and telemetry data. The goal in this work is to provide a representative methodology and examples that can be extended to build other types of Wireless DTs.

Fig. 6. Physical Site Layout and possible routes for mobile agent.

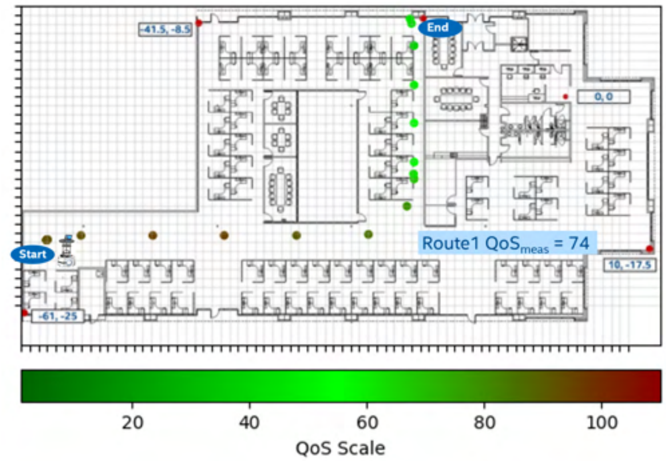

Fig. 7. Route 1 QoS Scores from Measured metrics.

IV. Performance Evaluation

In this section we describe an example use case and experiments using the DT of the Wi-Fi network at the CASPER site. The use case is a mobile robot connected to the Wi-Fi network moving from one starting point to a target location, emulating a mobile robot accomplishing a task. To complete its task successfully, the robot has to move from the starting point to the target location while maintaining a connection with an Edge compute server to continuously transfer data emulating a camera-generated video flow. As the robot moves it has a choice of routes that it can take. The Wi-Fi network consists of four APs deployed at fixed locations, as shown in Figure 6. Although there is only one mobile robot in the experiments, load can be generated in the network by using other static client devices placed at various points. The static clients as well as the mobile robot all run the mobile probes (described in Section III.A) that collect telemetry and relay it back to a back-end server to be aggregated along with telemetry coming from the APs in the network. The aggregated telemetry is sent to a server at located at a remote location where the DT is hosted.

The DT derives estimates of the QoS that can be achieved by the mobile robot for all the choices of routes that are available to the robot and at any given point recommends a route to take that has the expectation of best achievable QoS as described by the equations in section 3.B. In order to evaluate the DT fidelity, we compare the estimate of the performance that the model has predicted for a specific route with the actual measured performance achieved as the robot navigates that route. Since the DT is continuously calibrating and improving its accuracy, we also measure how many iterations the model takes to achieve estimates on par with the actual measurements.

V. Results and Analysis

The following steps details the DT initialization procedure.

Step 1: Execute Robot Task for a given route.
Step 2: Collect telemetry and update measur.
Step 3: Compare measured QoS scores with the DT generated QoS score for the same route.
Step 4: If Error from Step 3 is greater than a threshold, adjust model parameters as detailed in section 3.B and go to Step 1.
Step 5: If Error from Ste 3 is smaller or equal to a threshold, stop and repeat procedure for the next route.

To evaluate the DT initialization, we run multiple iterations of the initialization steps for one sample route. As we feed the telemetry into the DT, we also extract the predictions from the DT and compute the error between predicted and measured cell QoS score. Fig 9 shows that for all cells in route 1, after three iterations, the error between measured and predicted QoS scores is significantly reduced.

After the DT is initialized, we set experiments where the robot moves along select routes and we compare the QoS scores measured with the scores generated by the DT. After the robot moves along route 1, we collect the telemetry from all probe nodes and compute the QoS score for each point along the route. We then feed the measured values to the DT for calibrating the model. The DT adjusts its parameters using the measured values. Fig. 7 shows measured QoS scores using the telemetry. The color map shows the scale of the QoS scores for the cells in the route. The overall QoS score for the route was measured to be 74. Fig. 8 shows the DT generated QoS scores for the same route. The DT estimates a route QoS score of 71, which is very close to the measured score. The DT also estimates scores for all cells for which measurements were not available. We can see that the model, after calibration, has been able to estimate the expected QoS scores that are fairly close to the actual measured values.

For evaluating the decision making and recommendation capability of the DT, we add congestion along one of the routes as the mobile robot starts its task. We assess whether the DT is able to predict the degradation along the route and provide an alternative route. The prediction and alternative routes can be visualized by a user interacting over the XR interface or by an autonomous service through a management interface. We then repeat the experiment over both the routes with congestion

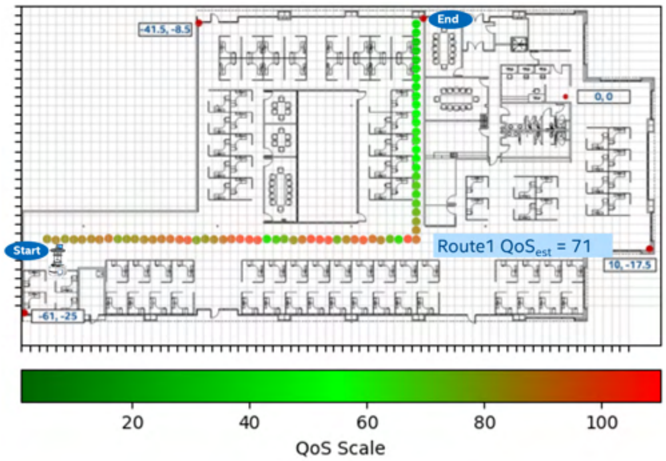

Fig. 8. Route 1 QoS Scores predicted and extrapolated by model.

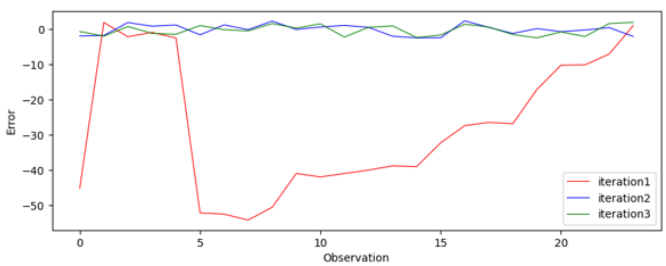

Fig. 9. Error between the measured data and estimated results as the Digital Model is being calibrated. In this case after the third iteration of the calibration, the error is minimum.

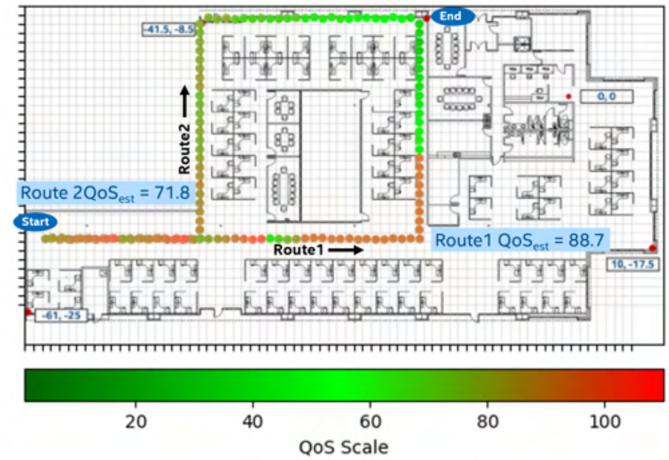

Fig. 10. The expected QoS over the two routes as predicted by the Digital Twin. Here the route 2 is slightly better than the route 1 owing to congestion along that route.

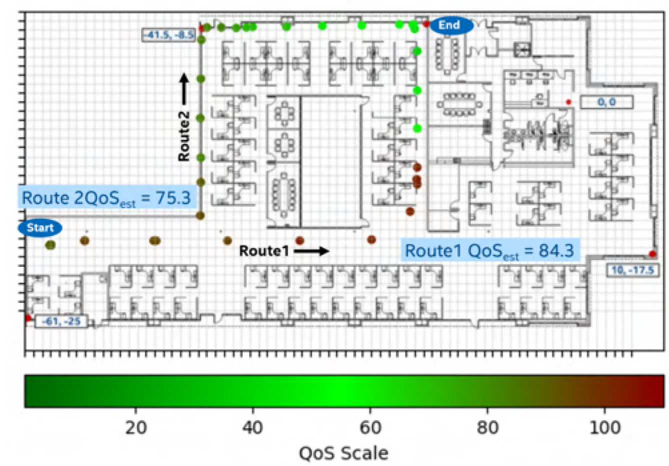

Fig. 11. Actual measured performance of the mobile agent as it traverses both the routes when there is congestion along one of the routes. Performance over route 2 is slightly better than performance over route 1.

enabled and compare the results.

Fig 10 and Fig 11 show the DT estimates and measured value, respectively, of the route QoS score for both routes with congestion along route 1. We can see that the model was able to pick up the increase in channel utilization from the AP telemetry and it was able to predict resulting degradation along the route due to the congested AP.

The DT could be used to recommend an alternate route - route 2 for the mobile robot to take. Note that the DT could find multiple suitable routes. In this case, the user through the XR interface annotated areas of the floor not allowed for robot navigation, leaving the two routes discussed in this Section as the only options Fig. 12 shows the XR visualization of the possible routes found by the DT.

VI. Conclusion

We have shown how a DT model of a Wireless Network environment and an associated use case, calibrated using real telemetry coming from the physical system, can be put together and used in real time to proactively and autonomously detect performance degradation and suggest configuration changes to improve and optimize the system performance. We have also shown how the DT 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 accuracy of the model. We have also shown how a human-machine interface can be used to interact with the DT 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 and autonomously 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 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 QoS.

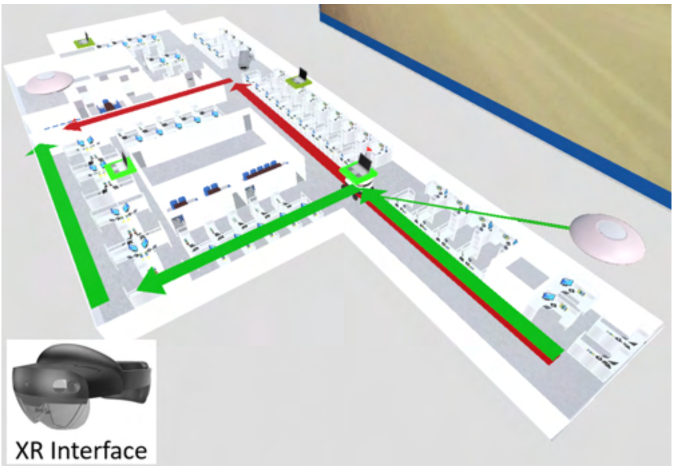

Fig. 12. Visualization of calculated routes in XR interface. Routes are color coded to indicate which one is best. User can use XR interface to select route or allow the DT to automatically select it.

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] C. Xu, Y. Wang, Z. Zhou, B. Gu, V. Frascolla, and S. Mumtaz, "A Low-Latency and Massive-Connectivity Vehicular Fog Computing Framework for 5G," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2018.

[4] P. G. Giardina, P. Szilágyi, C. F. Chiasserini, J. L. Carcel, L. Velasco, S. Spadaro, F. Agraz, S. Robitzsch, R. Rosales, V. Frascolla, R. Doostnejad, A. Calvillo, and G. Bernini, "A hierarchical AI-based control plane solution for multi-technology deterministic networks," in *Proceedings of the Twenty-Fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '23, (New York, NY, USA), p. 316–321, Association for Computing Machinery, 2023.

[5] E. Kartsakli, J. Perez-Romero, O. Sallent, N. Bartzoudis, V. Frascolla, S. K. Mohalik, T. Metsch, A. Antonopoulos, F. Tuna, Y. Deng, X. Tao, M. A. Serrano, and E. Quiñones, "AI-Powered Edge Computing Evolution for Beyond 5G Communication Networks," in *2023 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, pp. 478–483, 2023.

[6] 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.

[7] 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.

[8] 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.

[9] H. Liao, Z. Zhou, Z. Jia, Y. Shu, M. Tariq, J. Rodriguez, and V. Frascolla, "Ultra-Low AoI Digital Twin-Assisted Resource Allocation for Multi-Mode Power IoT in Distribution Grid Energy Management," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3122–3132, 2023.

[10] 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.

[11] 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.

[12] 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.

[13] Z. Min, Z. Liu, Z. Kang, S. Zhou, S. Gokhale, S. Shekhar, C. Mahmoudi, and A. Gokhale, "A Novel 5G Digital Twin Approach for Traffic Prediction and Elastic Network Slice Management," in *2024 16th International Conference on COMmunication Systems NETworkS (COMSNETS)*, pp. 497–505, 2024.

[14] 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.

[15] 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.

[16] 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.

[17] H. M. Kamdjou, D. Baudry, V. Havard, and S. Ouchani, "Resource-constrained extended reality operated with digital twin in industrial internet of things," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 928–950, 2024.

[18] L. Adriana Cárdenas-Robledo, Óscar Hernández-Uribe, C. Reta, and J. Antonio Cantoral-Ceballos, "Extended reality applications in industry 4.0. – A systematic literature review," *Telematics and Informatics*, vol. 73, p. 101863, 2022.

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