Communication-aware Robot Motion Planning via Online Estimation of Radio Maps

Daniel Gordon*, Mohammad Bariq Khan*†, Tommaso Zugno*,
Wu Yiqun*, Mate Boban*, Xueli An* and Falko Dressler†
*AWTL, Huawei Heisenberg Research Center (Munich), Germany
†School for Electrical Engineering and Computer Science, Technische Universität Berlin (Berlin), Germany
{daniel.gordon, mohammad.bariq.khan1, tommaso.zugno
wuyiqun, xueli.an, mate.boban}@huawei.com, dressler@ccs-labs.org

***Abstract*—Fast and reliable communication links are of vital importance for the successful operation of many robot systems. In situations where wired or local wireless communications are infeasible, such as robots operating in extremely remote conditions or in areas with limited communications infrastructure, mobile communication networks can provide a solution to share sensing data or provide motion plans over the network from a remote base station. However, typically robot motion planners do not consider the impact of network dynamics when performing robotic motion plans, and thus robots can easily enter areas of low signal quality, compromising overall performance. In this work, we address the co-simulation of networked robotic systems by directly integrating ray-tracing–based radio map estimation with robot motion planning algorithms. We present a framework for communication-aware motion planning that seamlessly incorporates wireless network dynamics into robot navigation. In particular, our approach converts radio map estimates into service-specific risk maps, which capture the likelihood of QoS violations along potential trajectories. By fusing uplink and downlink metrics into an end-to-end service-aware risk estimate, our method enables robots to proactively avoid regions of low communication quality. We evaluate the framework in simulation, showing that trajectories planned using QoS-to-risk maps yields $1.5 - 2\times$ improvement in communication performance compared to shortest-path baselines.**

***Index Terms*—Robot motion planning, communication-aware, radio maps, online estimation**

I. Introduction

Robots navigating in real-world environments, e.g., on city streets or on smart farms, can benefit from high quality communication links in order to transmit and receive sensory data for off-board processing. This is especially relevant due to the rise of computationally expensive image processing techniques, such as transformer-based vision-language models (VLMs) [1], which are more likely to be accessed remotely than hosted directly on robot hardware. Mobile communication networks can provide these links, however, due to variations in radio coverage, a robot navigating 'blindly' with respect to the underlying network conditions would be expected to intermittently enter areas of low coverage, thus compromising quality of service.

This interplay between robot and network dynamics has been studied in a number of previous works within the field of co-simulation of networking and physical systems, with approaches varying from combined simulation via virtual machines [2] to interfaces between concurrent robot/network simulations [3]. In a previous work, we investigated enabling robot motion planners to directly predict network coverage, thus influencing robotic motion plans [4] with minimal additional computational overhead. In this paper, we extend our work to the case of *online* communication-aware motion planning, with dynamic remapping of unknown or changing environments, and the potential for dynamic update of network infrastructure parameters, e.g., transmitter locations or operating frequencies. We demonstrate the framework through simulation scenarios which model robots with varying communication requirements navigating in a communication-constrained environment.

Specifically, we focus on communication-aware path planning using a QoS-to-risk map formulation. Here, radio maps are translated into service-specific risk maps that capture the likelihood of communication violations along potential trajectories. By fusing uplink and downlink metrics into a conservative, end-to-end risk estimate while preserving service awareness through category-specific QoS thresholds and weighting parameters, our approach enables robots to actively plan paths that minimize communication risk. Unlike our previous work, which focused on predicting coverage, this study demonstrates QoS-driven path optimization, quantifies communication performance via a trajectory-averaged QoS satisfaction ratio, and evaluates the effectiveness of risk-informed navigation across multiple service categories.

II. Related Work

Co-simulation of Networked Robotic Systems: Many current applications of robots involve wireless networked communication, for example, communication between mobile robots comprising a multi-agent system, communication between disaster recovery robots and a remote operator, or network offloading of computationally intensive tasks. In the coming years, it is widely expected that the uptake of robotic technologies will increase, leading to an increased demand on wireless communication systems, with robotics already a key topic area for current research & development towards 5G-Advanced and 6G mobile communications networks [5].

To appropriately control & plan for robots operating in such scenarios, it is important to consider both robotic (physical) dynamics as well as the dynamics of the communication

network. Typically, such approaches are termed *co-simulation* of networked robotic systems [6]. Of the existing works in this area, many are somewhat limited in scope by a focus towards a specific robotic embodiment (e.g., UAVs [7] or robot swarms [8]), or a specific combination of robot and network simulator.

ROS-NetSim [3] is a recent approach to co-simulation which aims to be more widely applicable, by defining a framework which interfaces separate network and robotic simulators, and is designed to be agnostic to the choice of each. The framework is built in ROS1, and uses TUN virtual network interfaces to capture messages from ROS1 nodes and pass them to a network simulator for processing. ROS-NetSim is flexible to the choice of network or robotics simulator, but is limited by being based on ROS1 which will be discontinued in 2025, and an inherent lack of scalability due to the need to introduce increasingly more TUN nodes in complex ROS1 systems (e.g., with many robots).

Ray Tracing for Radio Map Estimation: Simulation-based ray tracing is one approach to network propagation modeling, the process of estimating the field and signal strength characteristics of wireless channels given appropriate input parameters. Ray tracing models radio waves propagating through the environment as rays following geometric optics principles, and relies on numerical methods to compute all possible propagation paths from a source (transmitter) to a sink (receiver) [9].

GPU acceleration can significantly improve runtimes of ray-tracing algorithms compared to CPU-based approaches. In particular, Nvidia's Sionna and Instant-RM are two recent radio propagation modeling libraries with different feature sets that enable fast computation of radio maps [10], [11].

III. Methodology

A. Co-simulation Architecture

In our previous work on communication-aware motion planning [4], we presented a prototype implementation which utilized Nvidia Sionna to pre-compute radio maps for a robotic simulation environment. Although this work highlighted the potential of our approach to co-simulation of networked robotic systems, there were a number of limitations. Firstly, radio maps could only be computed for known environments, with environmental geometry and material properties required to be specified in advance. Furthermore, radio maps were static, and could not be dynamically updated in response to an unexpected change in the environment. Radio map computation could take up to 10s, and therefore intermittent re-planning based on radio map updates was slow, and not suited to online applications. Finally, the implementation was limited to a single transmitter, fixed at a point in space relative to the Sionna representation of the environment.

To address these limitations, the architecture of the technical implementation underlying this paper, termed ROS2-RM, has been designed to achieve the following advancements compared to our prior work, targeting four key features: (1) Support for robots navigating in unmapped areas; (2) Dynamically updating radio maps; (3) Real-time performance; (4) Support for multiple, mobile transmitters. The high-level architecture for our implementation is shown in Fig. 1. The key components of the framework are:

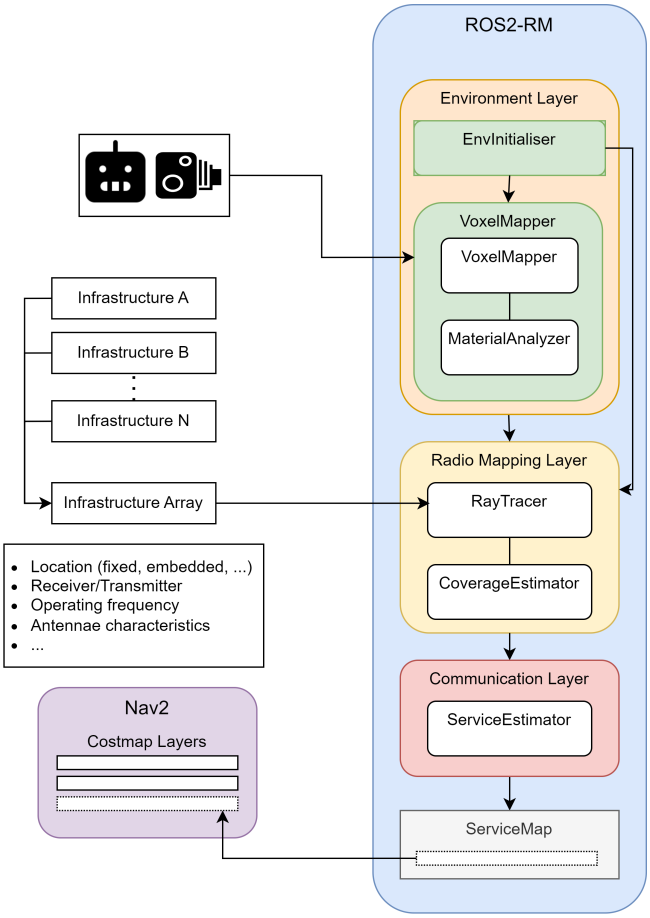


Fig. 1. High-level architecture of ROS2-RM for interfacing robot motion planning with real-time radio map estimation.

1) The *environment layer*, which maintains a representation of the environment compatible with the underlying ray tracing simulator. The environment can be static, or updated dynamically based on robot sensor data.
2) The *radio mapping layer*, which integrates the ROS2/Gazebo-based robotics simulator with the ray tracing simulator to produce radio map estimates at the current time t. Network infrastructure parameters (e.g., transmitter locations or frequencies) can be specified on startup and modified during runtime to update the corresponding radio map estimate. Transmitter locations can be attached to frames published on a ROS2 topic, allowing for mobile transmitters.
3) The *communication layer*, which translates from raw radio maps to high-level communication metrics according to the communication requirements of individual robots, and maps these metrics on to a costmap for motion planning.

In the following, we provide a detailed description of each component.

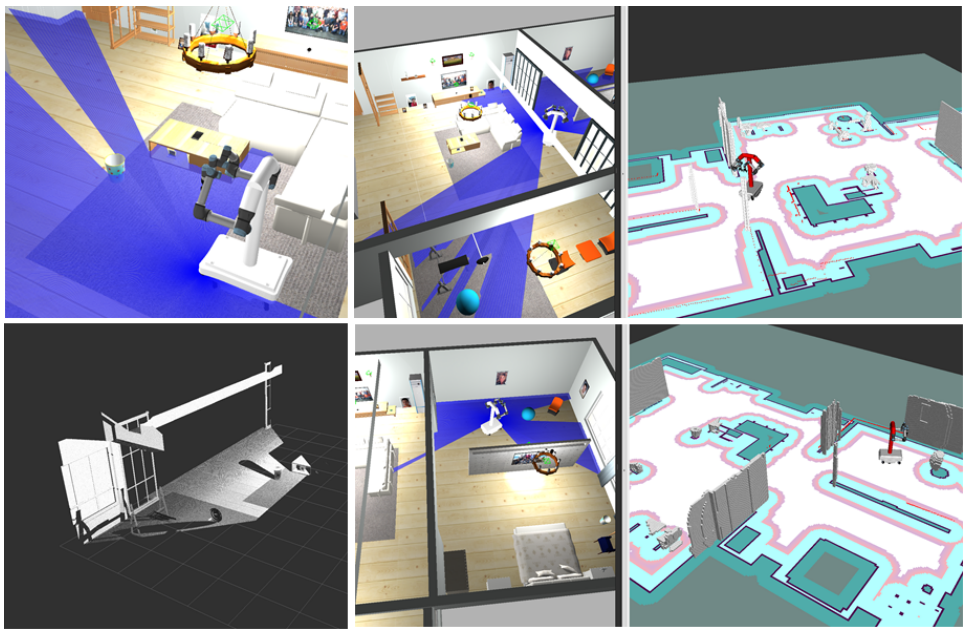


Fig. 2. Snapshots from the simulation test environment during environment reconstruction with Nav2's STVL, showing simulated depth sensor data, the robot navigating in the simulated environment, and the subsequent formation of voxels.

B. Environment Layer

Robots are expected to operate in dynamic scenarios, with changing environments and the presence of moving obstacles. In such conditions, network propagation characteristics are likely to be highly variable, and a static radio map is unlikely to be accurate. To provide support for estimating radio maps of dynamic environments, our framework includes the capability to generate 3D maps of environments from robot sensor data, via a two-point process of:

1) Collecting and maintaining a voxel-filtered grid of point cloud data from the environment.
2) Generating meshes of the environment from the point cloud information.

In practice, Nav2's Spatio-temporal Voxel Layer [12] is used to maintain a voxel grid of the environment, while Open3D [13] is used to subsequently produce meshes from voxel grids. The process of producing a voxelised representation of the environment is presented for reference in Fig. 2. Here, a custom mobile manipulator robot MELISAC [4], consisting of a MIR600 base and two UR5e robot arms attached to a frame, navigates in a simulated living room environment [14].

Currently, material characteristics from dynamically updated environments must be specified in advance (or manually during operation), but estimation of material characteristics from sensor data is an immediate source of further work, and a potential application of radio-based sensing modalities such as THz sensing, which may be suited to the identification of material characteristics. In scenarios which are expected to not have a dynamically changing environment, it remains possible to generate a static radio map from an existing 3D model of the space.

C. Radio Mapping Layer

The primary purpose of the radio mapping layer is to provide an interface between the robotics simulator and ray tracing tool via ROS2. In our previous work, we used Nvidia Sionna [10] as our ray tracing tool. To improve computation speed, our framework integrates Instant-RM [11], a package for real-time estimation of radio maps given information on the structure of the environment and network parameters, which utilises alternate models of material characteristics, as well as modifications to the underlying renderer in order to compute radio maps at a faster rate.

In addition to improvements in computation speed, the radio mapping layer allows for transmitter positions to be specified relative to frames published via tf2, the ROS2 transformation library. This allows transmitters to be attached directly to robots to simulate mobile base stations.

D. Communication Layer

We derive spatially varying Quality-of-Service (QoS) maps for downlink (DL) and uplink (UL) communication using a theoretical model that converts ray-traced radio signal strength (RSS) maps into per-location estimates of throughput, delay, and packet error rate (PER).

Let $\mathbf{R}_{\mathrm{DL}} \in \mathbb{R}^{H \times W}$ denote the downlink RSS map in dBm, where each grid cell corresponds to a spatial location. The map captures large-scale path loss and site-specific propagation effects and is resized to match the robot navigation grid. To model large-scale fading variability, Gaussian log-normal shadowing may be added:

$$\tilde{\mathbf{R}}_{\mathrm{DL}}(i,j) = \mathbf{R}_{\mathrm{DL}}(i,j) + \Delta_{\mathrm{shadow}}(i,j)$$

where $\Delta_{\text{shadow}}(i,j) \sim \mathcal{N}(0, \sigma^2_{\text{shadow}})$ are Gaussian random variables modeling log-normal shadowing, with standard deviation $\sigma_{\text{shadow}} = 2$ dB. The downlink SINR is computed as

$$\gamma_{\text{DL}} = \frac{P_{\text{sig}}}{P_{\text{noise}} + P_{\text{int}}}, \quad P_{\text{sig}} = 10^{\tilde{R}_{\text{DL}}/10},$$

with thermal noise

$$P^{\text{dBm}}_{\text{noise}} = -174 + 10\log_{10}(B) + NF, \quad B = N_{\text{RB}} \cdot 12 \cdot \Delta f,$$

where NF is the receiver noise figure, N_{RB} the number of resource blocks, and Δf the subcarrier spacing. Interference power P_{int} is optional.

For the uplink, pathloss reciprocity is assumed. With $P^{\text{tx}}_{\text{DL}}$ and $P^{\text{tx}}_{\text{UL}}$ the gNB and UE transmit powers, respectively, the pathloss is

$$L = P^{\text{tx}}_{\text{DL}} - \tilde{R}_{\text{DL}}, \quad \mathbf{R}_{\text{UL}} = P^{\text{tx}}_{\text{UL}} + G_{\text{gNB}} - L,$$

and the UL SINR γ_{UL} is computed analogously.

Bit error rate (BER) for modulation order M and coding rate r_c is approximated by [15]

$$\text{BER} \approx \frac{4}{\log_2 M} Q\left(\sqrt{\frac{3\, r_c\, \text{SINR}_{\text{linear}}}{M-1}}\right),$$

with packet error rate

$$\text{PER} = 1 - (1 - \text{BER})^{L_{\text{pkt}}},$$

for packet length L_{pkt}. The achievable physical-layer data rate (C_{eff}) and goodput (G) in bits per second are

$$C_{\text{eff}} = \eta B \log_2(1+\gamma), \quad G = C_{\text{eff}}(1 - \text{PER}),$$

where $\eta \in (0,1)$ accounts for coding, pilot, and MAC-layer overhead.

End-to-end packet delay is modeled as the sum of scheduling, retransmission, and queueing delays:

$$\mathbb{E}[N] = \frac{1}{1-\text{PER}}, \quad D = T_{\text{sched}} + \mathbb{E}[N] \cdot T_{\text{RTT}} + D_{\text{queue}},$$

where T_{RTT} is the HARQ round-trip time, and D_{queue} increases linearly with PER to reflect congestion from retransmissions: $D_{\text{queue}} = D_{\text{base}}(1 + 2 \cdot \text{PER})$.

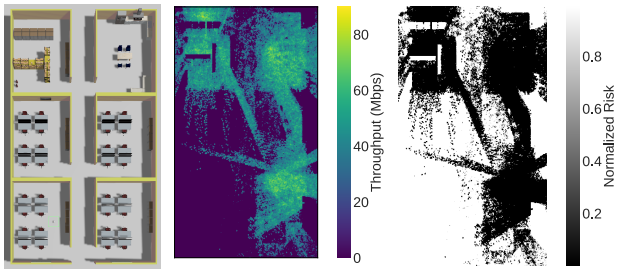
E. QoS-to-Risk Map Formulation

To uniformly capture the impact of heterogeneous QoS requirements, we define a normalized risk map that combines throughput, latency, and packet error rate (PER) into a single scalar metric.

Let $\mathbf{q}(\mathbf{x}) = [T(\mathbf{x}), D(\mathbf{x}), P(\mathbf{x})]$ denote the QoS vector at location $\mathbf{x}$, where T is the achievable goodput, D is the end-to-end delay, and P is the packet error rate. The overall communication risk is defined as

$$R(\mathbf{x}) = w_T\, r_T(T(\mathbf{x})) + w_D\, r_D(D(\mathbf{x})) + w_P\, r_P(P(\mathbf{x})), \quad (1)$$

where $w_T + w_D + w_P = 1$, and $r_T(\cdot)$, $r_D(\cdot)$, and $r_P(\cdot)$ are normalized risk functions mapping each QoS metric to the interval $[0,1]$.



(a) (b) (c)

Fig. 3. (a) Simulation setup in Gazebo; (b) Sample estimated uplink throughput map; (c) Service-aware normalized risk map.

Specifically, we adopt

$$r_T(T) = 1 - \frac{1}{1 + \exp\left(-k_T(T - T_{\text{th}})\right)}, \quad (2)$$

$$r_D(D) = \frac{1}{1 + \exp\left(-k_D(D - D_{\text{th}})\right)}, \quad (3)$$

$$r_P(P) = \frac{1}{1 + \exp\left(-k_P(log_{10}P - log_{10}P_{\text{th}})\right)}, \quad (4)$$

where T_{th}, D_{th}, and P_{th} denote service-specific QoS thresholds, and k_T, k_D, and k_P control the steepness of the corresponding risk transitions.

The parameters $\{w_T, w_D, w_P\}$ and thresholds are selected according to the QoS category of the service under consideration, as summarized in Table I and II.

IV. Simulations

The communication-aware motion planning framework was evaluated in a simulated office environment using the Gazebo simulator. This environment is a modified version of the world first shown in [4], with updated material characteristics to match the material surface models used by Instant-RM. The materials contained within the model correspond to concrete (for the floor, ceiling, and walls), metal (for the shelving and storage units) and wood (for the chairs, desks, and boxes). The office is 17m in width, 30m in length, and 2m in height. A top-down view of the environment is shown for reference in Figure 3a.

To generate radio maps, three transmitters were placed in the scene at fixed locations corresponding to the top left, top right, and bottom-right rooms. All transmitters were oriented towards the floor, and shared the same operating frequency of 2.5 GHz.

The QoS risk map was estimated using the communication model described in Section III-D with $N_{\text{RB}} = 50$ resource blocks and a subcarrier spacing of $\Delta f = 15$ kHz. The gNB downlink transmit power is set to 44 dBm, while the UE uplink transmit power is 23 dBm, with a receiver noise figure of 6 dB. A practical link efficiency factor of $\eta = 0.6$ is applied with M=4 (QPSK) modulation, coding rate $r_c = 0.5$ and a packet size of 1500 bytes. Hybrid ARQ is modeled with a round-trip time of 8 ms, scheduling delay of 2 ms and $D_{\text{base}} = 5$ ms. A sample of the the resulting throughput and risk map is depicted in Subfig. 3c and 3b.

TABLE I
QOS-DRIVEN SERVICE CATEGORIES AND DOMINANT RISK DIMENSIONS.

Category	Latency	Throughput	Reliability
UC1: URLLC-like control	High	Low	Very High
UC2: Tele-operation	High	High	Medium
UC3: Periodic control	Medium	Low	Very High
UC4: HD map streaming	Low	Very High	Medium

TABLE II
USE-CASE–SPECIFIC QOS TARGETS AND WEIGHTING PARAMETERS.

Use case	Latency (ms)			Throughput (Mbps)			PER		
	g_D	w_D	k_D	g_T	w_T	k_T	g_P	w_P	k_P
UC1	20	0.45	1	5	0.1	0.2	0.001	0.45	25
UC2	80	0.4	1	25	0.4	0.4	0.01	0.2	10
UC3	200	0.2	0.03	5	0.1	0.5	0.001	0.7	25
UC4	300	0.2	0.05	30	0.6	0.35	0.05	0.2	5

QoS-Driven Service Categorization: As summarized in Table I, we classify the considered use-cases into four *QoS-driven categories* to enable a unified, service-aware risk modeling framework for network-aware routing and control. Category A (UC1) comprises ultra-low-latency and ultra-reliable services, where packet error rate and latency dominate the risk. Category B (UC2) includes latency and throughput-sensitive services, requiring a balance between timely delivery and high data rates. Category C (UC3) captures reliability-dominated periodic control services with relaxed latency but stringent consistency requirements. Finally, Category D (UC 4) covers throughput-dominated streaming and map provisioning services, where sustained data rate is the primary concern and short-term latency or reliability degradations can be tolerated.

V. RESULTS AND DISCUSSION

Using the QoS-to-risk mapping in Section III-E, we generate risk maps for each service category. For simplicity, a weighted linear combination of the UL–DL risk maps is used, providing a conservative end-to-end risk estimate while maintaining service awareness via category-specific QoS thresholds and weights, as summarized in Table II.

A. Communication Performance

To evaluate communication-aware navigation, we quantify communication performance via a *QoS satisfaction ratio* metric, defined as the trajectory-averaged satisfaction score. For each trajectory sample, throughput, delay, and packet error rate (PER) are mapped to normalized satisfaction values in $[0, 1]$ with respect to task-specific QoS requirements. These per-metric scores are then combined using use-case-dependent weights (w_d, w_T, w_p), capturing the relative importance of each QoS dimension. To preserve sensitivity to strict service guarantees, the aggregated soft score is penalized whenever hard QoS constraints (g_D, g_T, g_p) are violated, yielding a hybrid metric that balances robustness and interpretability. The resulting QoS satisfaction ratio reflects how well a trajectory supports the intended service over its entire execution. For each use-case,

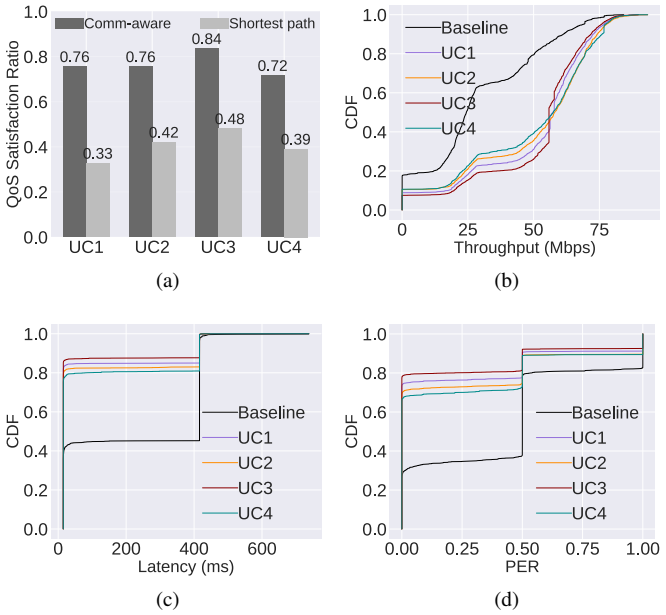


Fig. 4. Communication performance: (a) Communication-aware navigation significantly improves the QoS satisfaction ratio; (b)-(d) Empirical CDFs of the QoS metrics measured along the executed trajectories.

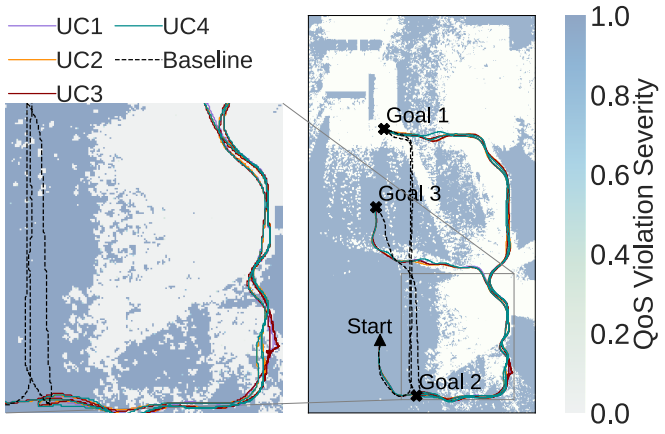


Fig. 5. Communication-aware navigation actively avoids severe QoS violation regions, in contrast to the shortest-path baseline.

including the baseline, the robot is assigned the same sequence of goal waypoints, starting from the bottom-left room.

As shown in Figure 4a, communication-aware navigation consistently improves the QoS satisfaction ratio across all four categories, with gains ranging from $1.5\times$ to $2\times$. These improvements are also evident in the empirical CDFs of the QoS metrics measured along the executed trajectories. In particular, the throughput-dominant UC4 exhibits a pronounced right shift in the throughput CDF (Figure 4b), with a steep increase around 55 Mbps. Conversely, the reliability-dominant UC1 and UC3 show strong left shifts toward near-zero packet error rates in the PER CDF (Figure 4d), reflecting improved link reliability. In the delay CDF (Figure 4c), all use-cases strongly outperform the baselines, although the relative gains are more moderate.

To provide spatial insight into where and why QoS degra-

TABLE III
WEIGHT ABLATION STUDY FOR UC1 WITH QUALITATIVE BEHAVIOR.

Weighting Scheme	Mean Risk	High-Risk Area	Risk Assessment
Uniform	0.454	53.6%	Underestimates
Service-Aware	0.528	53.9%	Balanced
PER Only	0.577	57.8%	Conservative

TABLE IV
SENSITIVITY OF THE RISK MAP TO LATENCY THRESHOLD VARIATIONS.

Threshold Scaling	Mean Risk	High-Risk Area	Risk Change
0.8× (Strict)	0.585	55.9%	Increased
1.0× (Nominal)	0.528	53.9%	–
1.2× (Relaxed)	0.524	53.4%	Decreased

dations occur, we also construct a QoS violation map, which aggregates normalized violations of hard QoS thresholds across throughput, delay, and PER. At each spatial location, the violation value is zero when all requirements are satisfied and increases proportionally with the severity of any violation. Overlaying executed trajectories on this map, as illustrated in Figure 5, shows that communication-aware navigation actively avoids regions with severe QoS violations, whereas the shortest-path baseline traverses communication-infeasible areas.

B. Ablation and Sensitivity Analysis

To evaluate the impact of service-aware QoS modeling and assess the robustness of the proposed risk formulation, we conduct a minimal ablation and sensitivity study focusing on the most influential parameters.

First, we perform a weight ablation study by comparing the proposed service-aware weighting for UC1 against two baselines: (i) uniform weighting across throughput, latency, and PER, and (ii) a single-metric configuration that considers PER only, emphasizing reliability exclusively. Table III reports the resulting mean risk and the fraction of the map classified as high risk. The service-aware configuration yields a higher mean risk than the uniform baseline, reflecting its conservative treatment of stringent URLLC requirements, while maintaining a comparable high-risk spatial extent. In contrast, the PER-only configuration results in the highest mean risk and the largest high-risk area, acting as a worst-case upper bound. This behavior confirms that packet error rate is the dominant failure mode for URLLC services, while also demonstrating that incorporating latency and throughput prevents overly pessimistic risk inflation. Overall, the service-aware weighting strikes a balance between conservativeness and spatial risk assessment.

Second, we analyze the sensitivity of the risk map to the latency threshold by varying it by $\pm 20\%$ around its nominal value again for UC1 URLLC-control service. As shown in Table IV, stricter latency requirements lead to higher overall risk, while relaxed thresholds reduce risk accordingly. The observed changes are smooth and monotonic, with no abrupt structural shifts in the risk map. This indicates that the proposed formulation is robust to moderate parameter variations and does not rely on fine-grained tuning.

Overall, these results demonstrate that service-aware weighting is essential to capture application-specific risk characteristics, and that the proposed risk model exhibits stable behavior under realistic uncertainty in QoS operating points.

VI. CONCLUSION

In this paper, we presented a framework for communication-aware path planning that integrates ray-tracing–based radio map estimation with robot motion planning. The framework enables real-time, dynamically updating radio map estimation which is leveraged for QoS-to-risk map formulation to guide robots along trajectories that minimize the likelihood of service violations. Simulation results demonstrate significant improvements in QoS satisfaction compared to shortest-path baselines, highlighting the practical effectiveness of risk-informed navigation in communication-constrained environments.

REFERENCES

[1] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *arXiv preprint arXiv:2210.05714*, 2022.

[2] J. Reitz and J. Roßmann, "A simulation framework for multi-robot cooperation over wireless networks," in *4th IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0/5.0 (ARCI 2024)*, 2024, pp. 37–42.

[3] M. Calvo-Fullana, D. Mox, A. Pyattaev, J. Fink, V. Kumar, and A. Ribeiro, "Ros-netsim: A framework for the integration of robotic and network simulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1120–1127, 2021.

[4] D. F. Gordon, Y. Wu, and X. An, "Communication-aware Motion Control for Mobile Robot Applications," in *4th IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0/5.0 (ARCI 2024)*, 2024, pp. 327–331.

[5] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[6] W. Li, X. Zhang, and H. Li, "Co-simulation platforms for co-design of networked control systems: An overview," *Control Engineering Practice*, vol. 23, pp. 44–56, 2014.

[7] S. Acharya, A. Bharadwaj, Y. Simmhan, A. Gopalan, P. Parag, and H. Tyagi, "Cornet: A co-simulation middleware for robot networks," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, IEEE, 2020, pp. 245–251.

[8] M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "RoboNetSim: An integrated framework for multi-robot and network simulation," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 483–496, 2013.

[9] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[10] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An open-source library for next-generation physical layer research," *arXiv preprint arXiv:2203.11854*, 2022.

[11] F. A. Aoudia, J. Hoydis, M. Nimier-David, S. Cammerer, and A. Keller, *Instant Radio Maps*, https://github.com/NVlabs/instant-rm, 2024.

[12] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, 2020.

[13] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[14] S. Gundry, A. Wong, and A. A. A., *AWS RoboMaker, Small House World ROS package*, https:// github [dot] com/aws-robotics/aws-robomaker-small-house-world, 2025.

[15] J. Musovic, A. Lipovac, and V. Lipovac, "BER Aided Energy and Spectral Efficiency Estimation in a Heterogeneous Network," *MDPI Computation*, vol. 10, no. 9, 2022.