GsOQDC: A GUI-Driven Interactive Framework for End-to-End Simulation of Optical Quantum Data Centers

Seyed Navid Elyasi[(1)], Sima Bahrani[(2)], Rui Wang[(2)], Dimitra Simeonidou[(2)], Paolo Monti[(1)] Rui Lin[(1)]

[(1)] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
[(2)] Smart Internet Lab, School of Electrical, Electronic, and Mechanical Engineering, University of Bristol, Bristol elyasi@chalmers.se

Abstract *We present GsOQDC, an open-source graphical framework integrating optical-network design, distributed quantum-circuit compilation, scheduling, and DES-based remote-gate simulation, enabling end-to-end cross-layer evaluation of entanglement-resource dynamics and system-level performance in Optical Quantum Data Centers ©2026 The Author(s)*

Review

Quantum Data Centers (QDCs), in which multiple Quantum Processing Units (QPUs) are interconnected through optical networks, are emerging as a promising architecture for scaling quantum computing beyond the limits of monolithic devices [1–3]. Recent experimental demonstrations of distributed quantum computation (DQC) across optically linked quantum modules [2, 4, 5] have renewed interest in DQC, where quantum circuits are partitioned across multiple QPUs and coordinated through entanglement distribution over an optical channel[2, 6, 7].

However, the software ecosystem for QDC research remains fragmented. Quantum software frameworks support circuit design and compilation[8, 9], distributed compilers address circuit partitioning[10], and network simulators [11]model communication infrastructures independently. As a result, researchers lack an integrated environment for evaluating the end-to-end behavior of distributed quantum applications and the interactions between quantum-computing and optical-network resources.

We present **GsOQDC (*GUI-Driven Simulator for Optical Quantum Data Centers*)**, an open-source graphical framework for the design, simulation, and evaluation of Optical Quantum Data Centers. As shown in Fig. 1, GsOQDC unifies the end-to-end DQC workflow, Circuit, Network, Compile, Schedule, and Run, within a single interactive environment. The platform integrates quantum-circuit modelling, optically switched QPU interconnection, network-aware distributed compilation, gate-level circuit scheduling, and a discrete-event simulation (DES) engine, enabling end-to-end cross-layer performance evaluation. Through an intuitive graphical interface, users can interactively construct QDC architectures, generate distributed workloads, visualize resource utilization, and investigate system-level trade-offs across the application, system, and network layers. The demonstration showcases an end-to-end DQC workflow, enabling rapid design-space exploration, repro-

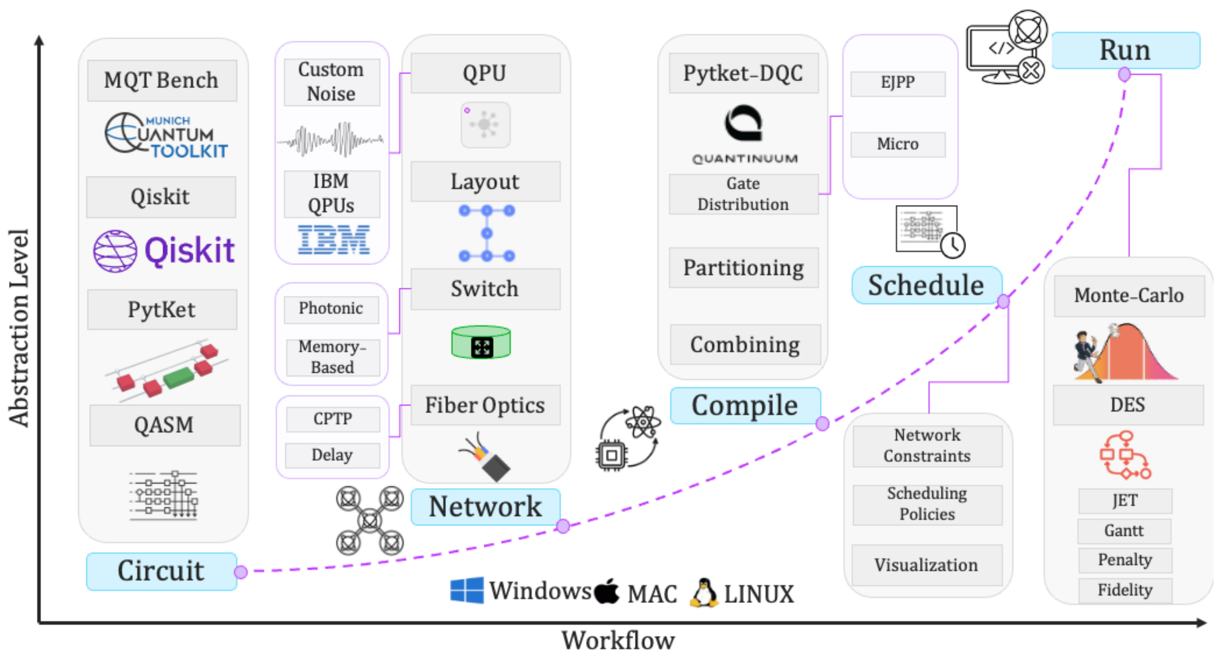

Fig. 1: GsOQDC workflow and software architecture. The platform integrates the end-to-end DQC workflow, enabling users to progress from quantum-circuit design and optical-network configuration to distributed compilation, scheduling, execution, and performance evaluation within a single graphical environment.

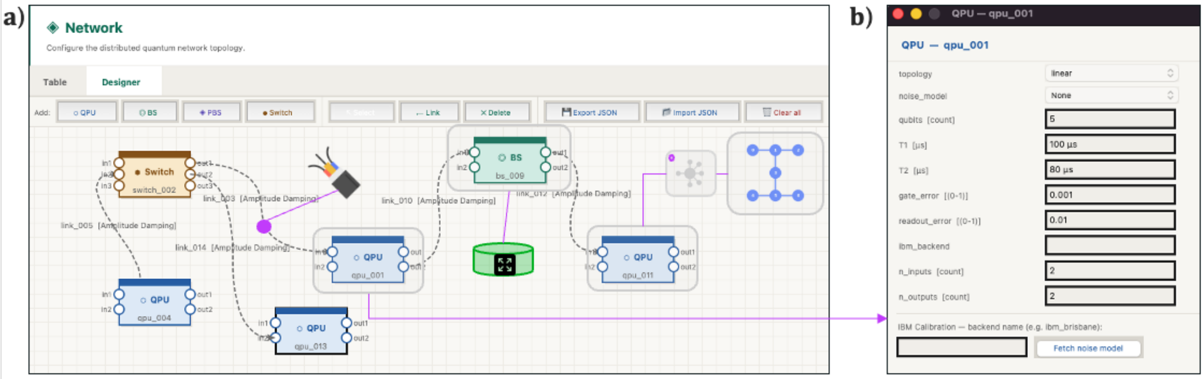

Fig. 2: GsOQDC Network Designer. (a) Interactive topology editor for constructing Optical Quantum Data Center architectures using QPUs, optical components, and communication links. (b) Component configuration panel, accessible via right-click, allowing users to modify hardware and noise parameters, including qubit properties, gate errors, coherence times, readout errors, and device-specific calibration data. Together, these capabilities enable realistic, cross-layer modeling of distributed quantum-computing infrastructures.

ducible experimentation, and quantitative analysis of how optical-network characteristics influence application-level performance in future Optical QDCs.

Novelty

GsOQDC provides an integrated environment for the design and evaluation of distributed quantum applications over Optical Quantum Data Centers (OQDC). As illustrated in Fig.1, the platform unifies the key components of a distributed quantum computing stack, spanning circuit specification, compilation, partitioning, scheduling, and network-level simulation, allowing researchers to move seamlessly from application specification to system-level analysis.

A key capability of GsOQDC is its network-centric design environment (Fig. 2), which elevates the optical infrastructure from a hidden communication abstraction to a configurable design object. Existing quantum-network simulators such as NetSquid [12] and QuNetSim [13] provide detailed protocol-level modeling of quantum communication systems but do not support distributed quantum-circuit compilation, scheduling, or system-level execution analysis.

Conversely, DQC frameworks and partitioning tools focus on circuit decomposition and resource allocation while typically assuming simplified or abstract communication models [10, 14, 15]. GsOQDC bridges this gap by integrating QPU placement, optical switching architectures, quantum memories, optical links, distributed compilation, scheduling, and system-level evaluation within a single graphical environment. This enables direct investigation of how network architecture and physical-layer impairments impact system-level metrics such as fidelity, communication overhead, entanglement-resource utilization, and Job Execution Time (JET).

The platform further integrates distributed compilation and scheduling into a unified workflow. Compiled circuits are partitioned across multiple QPUs using configurable partitioning strategies, while interactive scheduling views as Gannt representation(Fig. 3) expose the temporal relationship between local operations, remote-gate execution, and communication events. By visualizing resource contention and communication overheads, GsOQDC provides insight into performance bottlenecks that are typically hidden within standalone compilation frameworks. In particular, the platform enables the design and evaluation of scheduling strategies for remote gates, a critical consideration given that remote gate execution times are significantly longer than local gates, directly affecting qubit decoherence and overall circuit fidelity.

The final stage employs a custom DES and Monte Carlo engine to model the execution of remote quantum operations across distributed QPUs. The framework captures the end-to-end lifecycle of remote-gate execution, including entanglement generation, distribution, storage, consumption, and communication-induced delays. This enables direct evaluation of entanglement fidelity, JET, entanglement-resource utilization, and penalty delays caused by entanglement unavailability.

To the best of our knowledge, GsOQDC is the first graphical platform that enables end-to-end design-space exploration of distributed quantum applications while jointly considering optical-network behavior, entanglement-resource dynamics, and system-level performance.Unlike existing research tools that typically require command-line interaction and programming expertise, GsOQDC provides a user-friendly graphical environment that lowers the barrier to entry for DQC research. As shown in Fig.1, the platform is designed as a standalone cross-platform application and can be readily deployed on conventional Windows, macOS, and Linux systems, facilitating accessible and reproducible experimentation across the quantum networking and optical communications communities.

ECOC Relevance

Optical networking is emerging as a key enabler of scalable DQC, where optical interconnects distribute entanglement between physically separated QPUs. In such architectures, network-level decisions, including topology design, optical switching, entanglement generation policies, com-

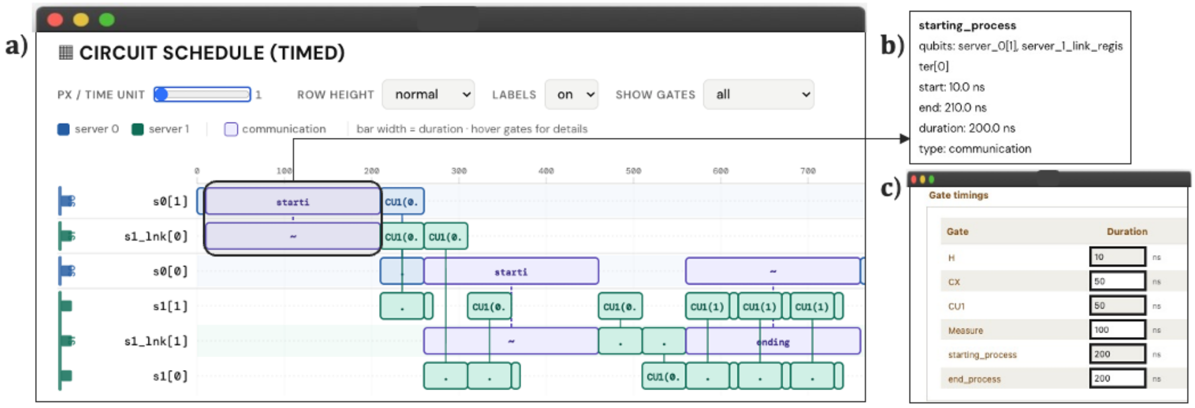

Fig. 3: Distributed-workload scheduling in GsOQDC. (a) Interactive Gantt-chart visualization showing local operations, remote-gate execution, and communication events across multiple QPUs. (b) Contextual information panel displayed when hovering over a scheduled operation, providing timing, duration, resource, and process details. (c) User-configurable gate and process timing parameters.

munication latency, and optical impairments, directly influence application-level performance. Understanding these interactions is therefore essential for the design of future Optical Quantum Data Centers and quantum-enabled network infrastructures.

GsOQDC addresses this challenge by providing an end-to-end framework for the design, scheduling, and evaluation of DQC applications over configurable optical-network architectures. By enabling cross-layer analysis across the application, compilation, scheduling, and network stack, the platform allows to quantify the impact of optical network characteristics on metrics such as entanglement fidelity, JET, inter-QPU communication overhead, and entanglement-resource utilization. The demonstration highlights a novel and timely application of optical communication technologies, aligning with ECOC's interests in advanced optical networks, data-center interconnects, and emerging quantum networking systems.

Demo Implementation

The demonstration will be conducted using the GsOQDC graphical platform running on a standard workstation and will showcase the end to end lifecycle of a DQC executed over a QDC. Attendees will be guided through the workflow illustrated in Fig. 1, beginning with quantum-circuit specification and progressing through optical-network design, distributed compilation, scheduling, and system-level performance evaluation.

Using the interactive Network Designer (Fig. 2), attendees can construct QDC topologies by placing QPUs, optical switching elements, entanglement-generation resources, quantum memories, and optical links. Hardware and noise parameters can be configured through component-specific dialogs, enabling rapid exploration of different architectural and physical-layer assumptions. The configured infrastructure is then used to compile quantum circuits across multiple QPUs using distributed partitioning algorithms, exposing communication requirements, remote-gate dependencies, and entanglement-resource demands.

The resulting distributed workload is transformed into an executable schedule and visualized through interactive Gantt charts (Fig. 3). Attendees can inspect individual operations, communication events, and resource allocations while observing how network constraints and scheduling policies influence execution timelines. This stage provides direct visibility into communication bottlenecks, remote-operation overheads, and resource contention that arise in distributed quantum applications.

Finally, GsOQDC employs a custom DES and Monte Carlo framework to model the execution of remote quantum operations over the configured optical infrastructure. The simulation captures entanglement generation, distribution, storage, and consumption processes together with communication delays and hardware imperfections. Performance metrics, including entanglement fidelity, JET, communication overhead, entanglement-resource utilization, and waiting-time penalties caused by unavailable entangled resources, are reported in real time. By allowing attendees to modify network, hardware, and scheduling parameters and immediately observe their impact on application-level performance, the demonstration highlights the value of cross-layer design-space exploration for future OQDCs.

Conclusions

GsOQDC introduces a unified graphical framework for the design, scheduling, and evaluation of DQC over QDCs. By integrating optical-network modeling, distributed compilation, scheduling, and system-level DES-based simulation within a single environment, the platform enables end-to-end cross-layer analysis of how optical-network characteristics influence entanglement fidelity, communication overhead, entanglement-resource utilization, and JET. Through its user-friendly and cross-platform design, GsOQDC lowers the barrier to entry for DQC research while providing a practical platform for exploring the architectural and networking foundations of future OQDC.

Acknowledgements

This work is supported by the Swedish Research Council (VR) and the UK EPSRC IQN Hub (EP/Z533208/1).

References

[1] S. N. Elyasi, P. Monti, J. Li, and R. Lin, "A framework for quantum data center emulation using digital quantum computers", *arXiv preprint*, 2025. arXiv: 2509.04029. [Online]. Available: https://arxiv.org/abs/2509.04029.

[2] D. Main et al., "Distributed quantum computing across an optical network link", *Nature*, vol. 638, pp. 383–388, 2025. DOI: 10.1038/s41586-024-08404-x. [Online]. Available: https://doi.org/10.1038/s41586-024-08404-x.

[3] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: A survey", *Computer Networks*, vol. 254, p. 110 672, 2024. DOI: 10.1016/j.comnet.2024.110672. [Online]. Available: https://doi.org/10.1016/j.comnet.2024.110672.

[4] Y. Zhou et al., "A 1-km photonic link connecting superconducting circuits in two dilution refrigerators", *Nature Photonics*, 2026. DOI: 10.1038/s41566-026-01866-7.

[5] V. Krutyanskiy et al., "Entanglement of trapped-ion qubits separated by 230 meters", *Physical Review Letters*, vol. 130, no. 5, p. 050 803, 2023. DOI: 10.1103/PhysRevLett.130.050803. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.130.050803.

[6] M. Mirhosseini, A. Sipahigil, M. Kalaee, and O. Painter, "Quantum transduction of optical photons from a superconducting qubit", *Nature*, vol. 588, pp. 599–603, 2020. DOI: 10.1038/s41586-020-3038-6. [Online]. Available: https://doi.org/10.1038/s41586-020-3038-6.

[7] C. Monroe et al., "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects", *Physical Review A*, vol. 89, p. 022 317, 2014. DOI: 10.1103/PhysRevA.89.022317. [Online]. Available: https://doi.org/10.1103/PhysRevA.89.022317.

[8] S. Sivarajah, S. Dilkes, A. Cowtan, W. Sherburn, A. Misra, and R. Duncan, "t|ket>: A retargetable compiler for NISQ devices", *Quantum Science and Technology*, vol. 6, no. 1, p. 014 003, 2020. DOI: 10.1088/2058-9565/ab8e92. [Online]. Available: https://doi.org/10.1088/2058-9565/ab8e92.

[9] A. Javadi-Abhari et al., *Quantum computing with Qiskit*, 2024. arXiv: 2405.08810. [Online]. Available: https://arxiv.org/abs/2405.08810.

[10] P. Andres-Martinez, D. Mills, T. Forrer, and L. Henaut, *CQCL/pytket-dqc: Distributed quantum computing compiler based on pytket*, 2024. [Online]. Available: https://github.com/CQCL/pytket-dqc.

[11] Cisco Systems, *Cisco packet tracer*, 2024. [Online]. Available: https://www.netacad.com/courses/packet-tracer.

[12] T. Coopmans et al., "NetSquid, a NETwork Simulator for QUantum Information using Discrete events", *Communications Physics*, vol. 4, p. 164, 2021. DOI: 10.1038/s42005-021-00647-8. arXiv: 2010.12535.

[13] S. DiAdamo, J. Nötzel, B. Zanger, and M. M. Bese, "QuNetSim: A software framework for quantum networks", *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–12, 2021. DOI: 10.1109/TQE.2021.3092395. arXiv: 2003.06397.

[14] S. Zhang, Y. Liu, B. Mark, W. Jiang, Z. Yang, and L. Yang, "Simulating circuit layout for distributed quantum computing", *arXiv preprint*, 2024. arXiv: 2512.21403. [Online]. Available: https://arxiv.org/abs/2512.21403.

[15] Anonymous, "SimDisQ: An end-to-end distributed quantum circuit simulator", *arXiv preprint*, 2025. arXiv: 2511.19791. [Online]. Available: https://arxiv.org/abs/2511.19791.