DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

University of Padua – Department of Information Engineering

Bachelor's Degree in Computer Engineering

Final Report

<<Overview of Simulation Platforms for Quantum Communications Research >>

Supervisor: Prof. Giordani Marco

Student's name: Alliegro Michela
Student ID: 2066623

Padua, 21/7/2026

Why are Quantum Communications needed:

Quantum computing applies the principles of quantum mechanics to information processing. Over the past decades, research has demonstrated that quantum computers can solve specific problems more efficiently than classical computers.

The first complete model of a quantum machine was designed by David Deutsch in 1985, ever since many significant publications have demonstrated possible advantages of employing these machines for specific computations:

- Quantum speedup (Deutsch, 1985)
- Integer factorization (Shor, 1995)
- Quantum database search (Grover, 1996)

Beyond computation, quantum technologies enable applications such as secure communication through protocols such as Quantum Key Distribution (QKD) and support distributed quantum computing, where multiple quantum processors communicate over quantum networks.
These developments create the need for communication networks capable of transmitting quantum information, laying the foundation for a future Quantum Internet.

Constraints of physical quantum systems:

Current quantum technologies face several practical limitations that restrict the deployment of large-scale quantum networks:

- High hardware costs make experimental research expensive;
- Decoherence and noise degrade quantum states during computation and transmission;
- Communication distance is limited by phase loss and decoherence;
- The no-cloning theorem prevents quantum information from being copied for error recovery;
- Scalability remains a major challenge as networks grow in size and become more complex.

These limitations make simulation platforms essential for designing, testing, and evaluating quantum communication networks before real-world implementation.

PURPOSE

Objectives:

This thesis aims to:

- Review the main quantum communication simulation platforms.
- Analyze their architectures, functionalities, and implementation approaches.
- Evaluate their strengths, limitations, and possible applications.
- Compare the platforms to identify their main differences and suitable use cases.
- Discuss the future evolution of quantum communication simulators and the challenges that remain.

Basic concepts:

Before comparing the simulation platforms, the thesis introduces the key concepts of quantum information:

- Qubit: The basic unit of quantum information, characterized by its quantum state.
- Superposition: The ability of qubits to exist in several states simultaneously until measured.
- Entanglement: A phenomenon in which two qubits share correlated quantum states, even when separated by large distances.
- Decoherence: The loss of a qubit's quantum state due to interactions with its environment.
- No-cloning theorem: A principle that states that an unknown quantum state cannot be copied exactly.
- Entanglement swapping: A protocol that creates entanglement between distant qubits through intermediate nodes called repeaters.

These concepts define the behaviour that quantum communication simulators try to accurately reproduce.

WORK'S DESCRIPTION

NetSquid:

NetSquid is a discrete-event quantum network simulator developed by QuTech. It focuses on realistic simulation of the physical and link layers of quantum networks.

Main features:

- Realistic models of quantum memories, channels, and processors.
- Simulates decoherence, noise, and communication delays.
- Modular architecture for custom network design.
- Provides performance statistics such as fidelity and latency for simulations.

Strengths:

- High physical realism and customization.
- Accurate event-driven simulations.
- Suitable for large-scale network analysis.

Limitations:

- Closed-source software.
- Requires registration.
- Difficult learning process.

WORK'S DESCRIPTION

NetQASM:

NetQASM is a hardware-independent open-source instruction set architecture (ISA) designed for quantum network applications programming. It provides a common interface between quantum applications and quantum network processing units (QNPU).

Main features:

- Standardized low-level instruction set.
- Low-weight programming for limited hardware capabilities.
- Hardware-independent design.
- Python SDK available for easier development.

Strengths:

- Portable across different hardware and simulators.
- Lightweight and efficient programming.
- Supports remote entanglement generation and other fundamental use cases of quantum applications.

Limitations:

- Not a simulator by itself.
- Requires an external backend for execution (either a simulator or real quantum hardware).

SquiASM:

SquidASM is an open-source application development framework built on top of NetSquid. It simplifies the creation of distributed quantum applications using a high-level Python interface.

Main features:

- Uses NetSquid as simulation engine.
- Integrates with NetQASM.
- Supports classical and quantum communication.
- Designed for application development.

Strengths:

- Easier programming than NetSquid.
- High-level abstractions.

Limitations:

- Depends on NetSquid, which needs to be independently downloaded.
- Not accurate physical-layer modelling.

SimulaQron:

SimulaQron is an open-source development platform for distributed quantum application developement. It simulates quantum processors con local classical machines and then connect them through a SimulaQron server simulating a quantum network.

<u>Main features:</u>

- Simulates distributed quantum nodes.
- Supports Python and CQC programming.
- Models entanglement and teleportation.

<u>Strengths:</u>

- Adapted specifically for software development.
- Easy testing of distributed applications.

<u>Limitations:</u>

- No realistic timing simulation.
- Not suitable for hardware performance evaluation.

QKDNetSim+:

QKDNetSim+ extends the NS-3 network simulator and the QKDNetSim module, with support for QKD protocols and their integration with classical communication networks.

Main features:

- Simulates QKD protocols.
- Integrates quantum and classical networking.
- Generates realistic cryptographic keys.
- Encrypts and authenticates transmitted packets.

Strengths:

- Excellent for QKD research.
- Strong integration with classical Internet protocols.
- Open-source and highly configurable.

Limitations:

- Simplified quantum hardware model.
- Limited scalability for complex quantum networks.

WORK'S DESCRIPTION

SeQUeNCe:

SeQUeNCe is a full-stack quantum network simulator that models both quantum hardware and communication protocols within a modular architecture.

Main features:

- Models hardware, resource management, routing, and applications.
- Realistic event scheduling in the simulation kernel.
- Supports entanglement generation, swapping, and purification.
- Scalable modular design.

Strengths:

- Most complete simulation platform.
- High physical realism.
- Full-stack architecture.
- Open-source.

Limitations:

- Computationally expensive.
- More complex configuration than other platforms.
- The stack architecture is defined by the organization of its modules.

CONCLUSIONS

User Experience:

Although the analyzed simulators have different objectives, they provide a similar user experience. Most of them are command-line applications developed in Python or C++, requiring users to create simulations through code rather than graphical interfaces.

Most platforms are officially supported on Linux and macOS, while Windows users generally rely on the Windows Subsystem for Linux (WSL).

NetSquid presents additional usability limitations because it requires user registration and is distributed under a research license. Furthermore, its protocol implementations cannot be inspected or modified because the source code is not publicly available.

Overall, the usability of these platforms depends strongly on the user's programming experience and the objectives of the research.

CONCLUSIONS

Comparison tables:

Simulator	*Layer*	*Language*	*Strengths*	*Limitations*	*Quantum mechanisms*	*Simulated channel*	*Stack layer*	*Classical integration*
NetSquid	Physical, link	Python	Accurate hardware models, accurate discrete-time simulation	Limited higher-level networking	Entanglement, memories, noise	Physical, optical	Physical, Link	Full
SquidASM	Application	Python	Higher-level application development, NetQASM compatibility	Dependent from NetSquid for simulations	NetQASM protocols	Abstract	Application	Full
SimulaQron	Application	Python	Higher-level application development, CQC compatibility	Poor hardware realism	Entanglement, teleportation	Abstract	Application	Full
QKDNetSim+	QKD	C++/NS-3	Integration with classical networks	Limited scalability	QKD	Simplified quantum	Network	Native NS-3
NetQASM	Physical	NetQASM, Python	Lightweight quantum programming	Dependent on other software for simulations	ISA instructions	None	ISA	External
SeQUeNCe	Full stack	Python	Complete Quantum Networking	Computationally expensive	Entanglement, swapping, purification	Physical optical	Full stack	Full

CONCLUSIONS

Comparison tables:

Simulator	*Applications*	*Scalability*	*Abstraction vs realism*	*Reproducibility*	*Extensibility*	*Usability*	*Outputs/Metrics*	*Critical limitation*
NetSquid	Repeaters, routing	High	Very realistic	Low	Medium	Medium	Fidelity, latency	Closed source
SquidASM	Distributed applications	Medium	High abstraction	High	High	High	Correctness	Needs NetSquid
SimulaQron	Distributed software	Medium	High abstraction	High	Medium	High	Correctness	No timing realism
QKDNetSim+	QKD	High	Medium abstraction	High	High	Medium	QKD results, latency	Simplified quantum model
NetQASM	Quantum internet	Backend dependent	Very abstract	High	High	Medium	Instruction results	Needs backend
SeQUeNCe	Quantum internet	Very high	Very realistic	High	Very high	Medium	Fidelity, throughput	Complex configuration

CONCLUSIONS

Final Observations:

This thesis compared six simulation platforms developed for different aspects of quantum communication research.
The analysis shows that no single simulator is capable of accurately modelling every layer of a quantum network. Instead, each platform specializes in a particular area, such as hardware simulation, application development, or QKD.
Consequently, the choice of simulator depends on the specific research objective.

As quantum technologies continue to evolve, future simulators should combine accurate physical modelling, application-level support, and improved scalability to enable realistic simulation of the future Quantum Internet.