A tool for Quantum Network design

IETF Quantum Internet Research Group interim
Rob Knegjens, on behalf of the NetSquid dev team
April 8th 2020
The NetSquid Project

Network Simulator for Quantum Information using Discrete events

- Developed at QuTech (TNO and TUDelft) since 2017
- In active use by
  - Groups of Stephanie Wehner and David Elkouss (QuTech QINC roadmap)
  - Partners in the Quantum Internet Alliance (EU Quantum flagship)

Public beta release: very soon!

https://netsquid.org
A Quantum Internet

Key resource: quantum entanglement

- Rate
- Fidelity

\[ \downarrow \text{loss} \]
\[ \downarrow \text{noise} \]

Non-ideal quantum channels and operations
A Quantum Internet

Key resource: quantum entanglement

- Rate
- Fidelity

\[ \downarrow \text{loss} \]
\[ \downarrow \text{noise} \]

Non-ideal quantum channels and operations

Mitigate:

Noise: entanglement purification

Loss: quantum repeaters

Designing a quantum internet involves solving complex timing dependencies
The NetSquid Simulator

- **Python 3 package**
  Optimised C/Cython code under the hood.

- **Specialised quantum computation library**
  “Qubit-centric” and optimized for repeated sampling.

- **Seamless choice of quantum state formalisms**
  Trade-offs: performance, scalability and versatility

- **netsquid package**
  - .protocols
  - .nodes
  - .qubits
  - .components
  - .pydynaa

- **Asynchronous framework**
  For programming quantum network protocols.

- **Modular component library**
  Physically model network hardware with composable base classes.

- **Discrete event simulation engine**
  Accurately track quantum decoherence across a network in time.
Use cases

- **Accurately model the effects of time** on the performance of scalable quantum networks
- Investigate the requirements and feasibility for the layers of a quantum internet stack:
  - *physical → control plane → user applications*
- Emulate future hardware for demonstrator setups

Examples

- Performance of a **quantum link layer protocol**
- Parameter optimization and benchmarking for a pan-European quantum internet (QIA)
- Parameter sensitivity for repeater chains*

* T. Coopmans et al, APS 2019
Getting started with NetSquid

First register at the NetSquid forum: https://forum.netsquid.org

Install via PyPI server: pip3 install --extra-index-url https://<username>:<password>@pypi.netsquid.org netsquid

User license (pending): free for non-commercial use
Snippets: user extensions to NetSquid

- **NetSquid Snippets** are Python packages that extend NetSquid
- Created, maintained and **shared** by users
- Some snippets already hosted on NetSquid PyPI server

New snippets can be generated using the template repository: 
[https://github.com/SoftwareQuTech/NetSquid-SnippetTemplate](https://github.com/SoftwareQuTech/NetSquid-SnippetTemplate)

More info at [https://netsquid.org](https://netsquid.org)

<table>
<thead>
<tr>
<th>Package name</th>
<th>Description</th>
<th>Maintainer*</th>
</tr>
</thead>
<tbody>
<tr>
<td>netsquid-physlayer</td>
<td>Physical layer modeling</td>
<td>Axel Dahlberg</td>
</tr>
<tr>
<td>netsquid-rv</td>
<td>NV centre modeling</td>
<td>Tim Coopmans</td>
</tr>
<tr>
<td>netsquid-ae</td>
<td>Atomic ensembles</td>
<td>David Maier</td>
</tr>
<tr>
<td>netsquid-netconf</td>
<td>Network configuration</td>
<td>Guus Avis</td>
</tr>
<tr>
<td>netsquid-magic</td>
<td>Magic EGPs</td>
<td>Tim Coopmans</td>
</tr>
<tr>
<td>netsquid-qmm</td>
<td>Quantum memory manager</td>
<td>Axel Dahlberg</td>
</tr>
<tr>
<td>netsquid-qpm</td>
<td>Quantum program manager</td>
<td>Wojciech Kozlowski</td>
</tr>
<tr>
<td>netsquid-simulationtools</td>
<td>Simulation tools for NetSquid</td>
<td>Guus Avis</td>
</tr>
</tbody>
</table>

* maintainers in some cases not the creator(s)
Backup slides
The NetSquid Package (I)

PyDynaa subpackage provides the **discrete event simulation engine**
*Based on the core of the DynAA simulator*

---

*de Oliveira, Papp, Djapic, Oosteveen, Proceedings of SASO (2013)*
The NetSquid Package (II)

- **Qubit** objects *dynamically* share quantum states
- Computation optimised for repeated sampling
- Seamless choice of quantum state formalisms for **small and large networks**

**Formalism trade-offs:** universality, memory efficiency, speed

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

<table>
<thead>
<tr>
<th>Ket vector</th>
<th>Density matrix</th>
<th>Stabilizer state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\Phi^+\rangle$</td>
<td>$\rho =</td>
</tr>
</tbody>
</table>

- $\sim 2^n$
- $\sim 2^{2n}$
- $\sim 2n^2$
Components are the physical entities of a network:

- **Base components**: channels, quantum processing devices, photon sources, ...
- **Composite components**: nodes, connections, …
- **Attachable physical models**: delay, loss, memory noise, gate noise
The NetSquid Package (IV)

Protocols dictate node behaviour (virtual entities)
- Numerically simulate via random sampling
- Massively parallelizable

Example: a distillation protocol

A ⊗ B

F = 0.75
F' = 0.79 > F
p_{wire} = 72%