NetSquid

A tool for Quantum Network design

IETF Quantum Internet Research Group interim

Rob Knegjens, on behalf of the NetSquid dev team

April 8th 2020

TNO innovation for life

TUDelft

QUTech

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

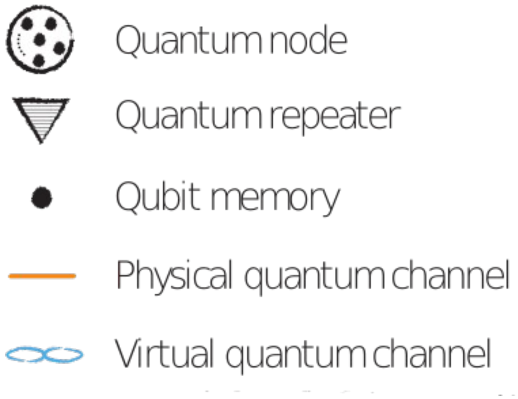
A Quantum Internet

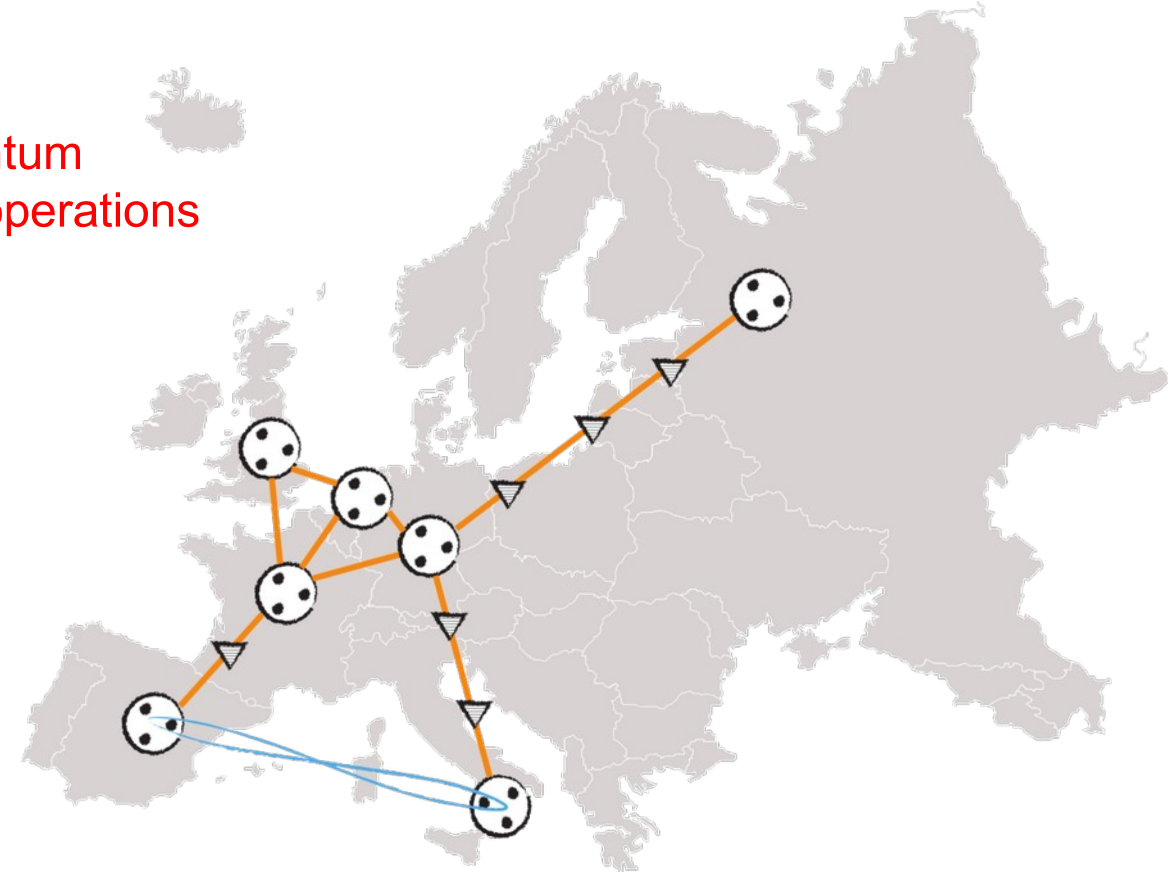
Key resource: quantum entanglement

- Rate
- Fidelity

- ⇓ **loss**
- ⇓ **noise**

Non-ideal quantum channels and operations





A Quantum Internet

Key resource: quantum entanglement

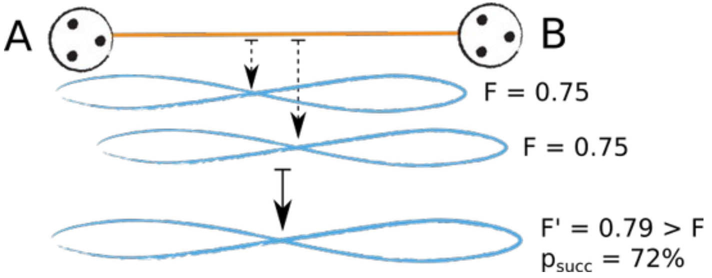
- Rate
- Fidelity

- ⇓ **loss**
- ⇓ **noise**

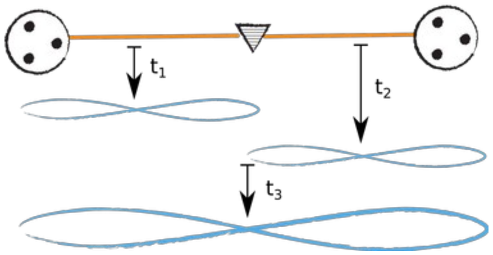
Non-ideal quantum channels and operations

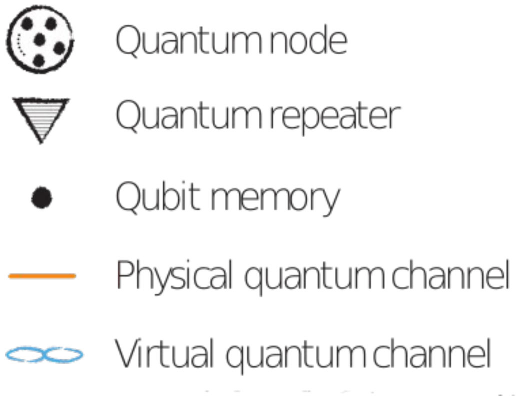
Mitigate:

Noise: entanglement purification



Loss: quantum repeaters



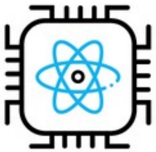


Designing a quantum internet involves **solving complex timing dependencies**

The NetSquid Simulator

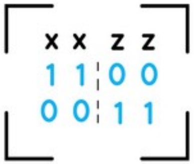
Python 3 package

Optimised C/Cython code under the hood.



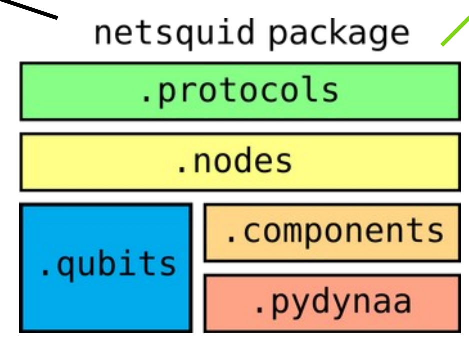
Specialised quantum computation library

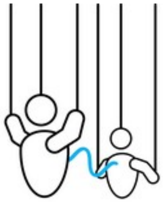
“Qubit-centric” and optimized for repeated sampling.



Seamless choice of quantum state formalisms

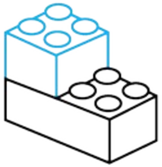
Trade-offs: performance, scalability and versatility





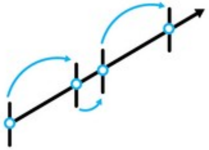
Asynchronous framework

For programming quantum network protocols.



Modular component library

Physically model network hardware with composable base classes.

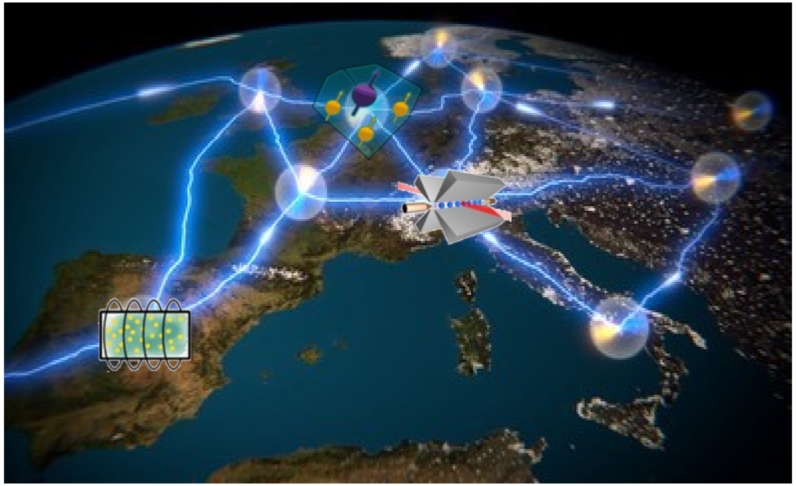


Discrete event simulation engine

Accurately track quantum decoherence across a network in time.

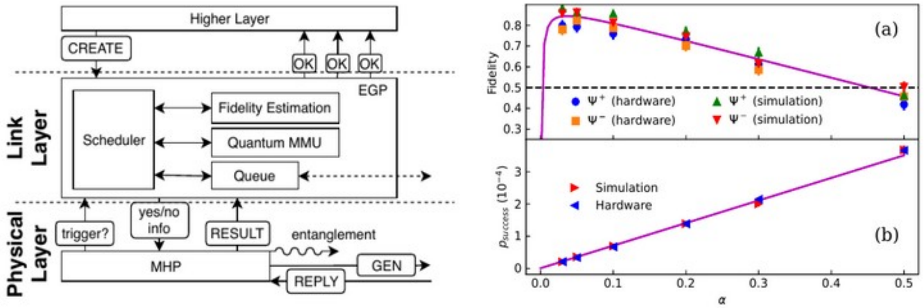
Use cases

- *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*



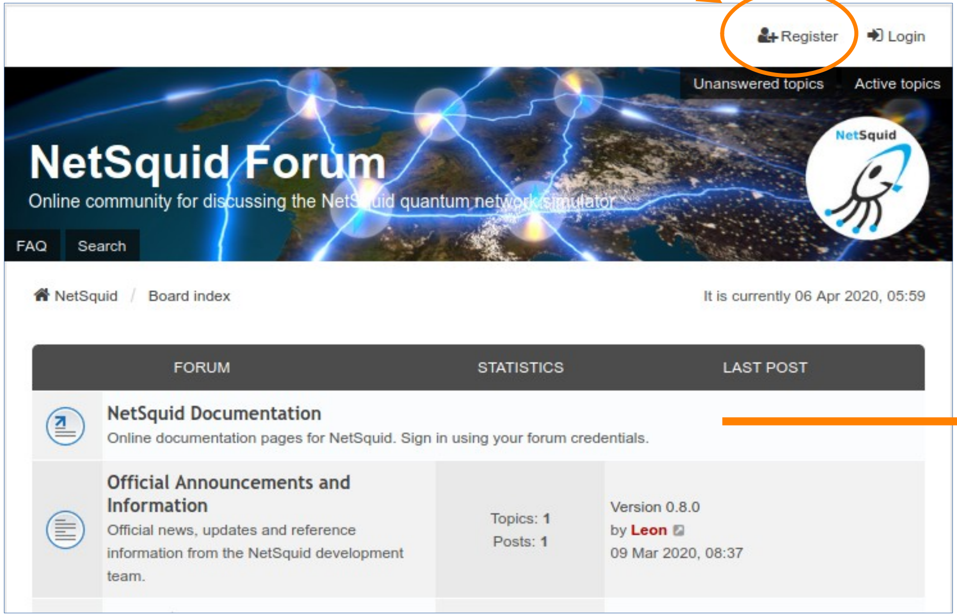
A Link Layer Protocol for Quantum Networks

Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, Stephanie Wehner

*T. Coopmans et al, APS 2019

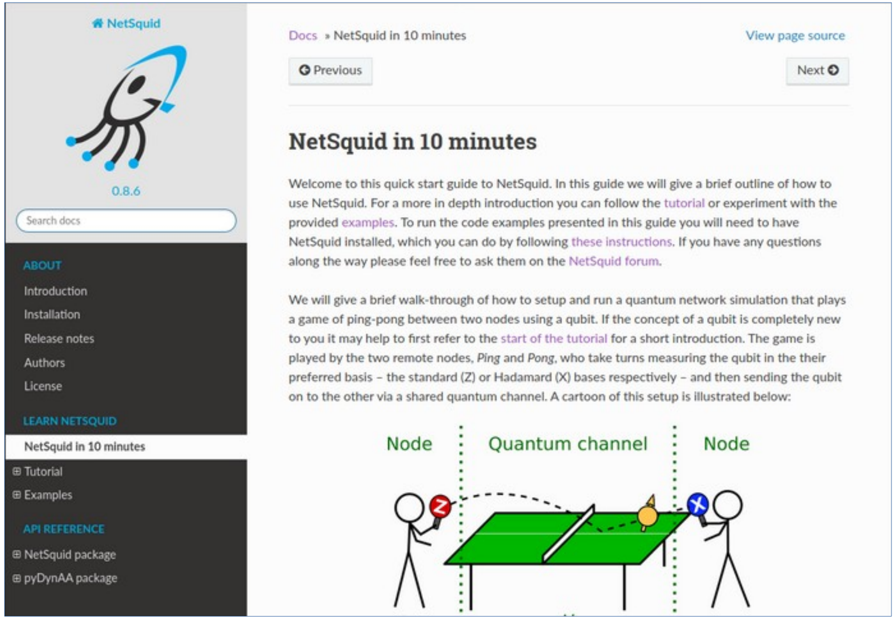
Getting started with NetSquid

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



uses forum credentials

Online documentation



Install via PyPI server:

```
pip3 install --extra-index-url https://<username>:<password>@pypi.netsquid.org netsquid
```

Your forum credentials

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

Examples:

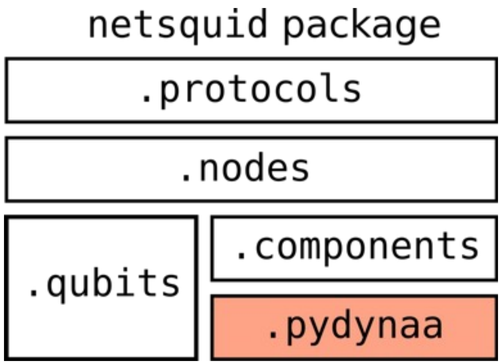
Package name	Description	Maintainer*
netsquid-physlayer	Physical layer modeling	Axel Dahlberg
netsquid-nv	NV centre modeling	Tim Coopmans
netsquid-ae	Atomic ensembles	David Maier
netsquid-netconf	Network configuration	Guus Avis
netsquid-magic	Magic EGPs	Tim Coopmans
netsquid-qmm	Quantum memory manager	Axel Dahlberg
netsquid-qpm	Quantum program manager	Wojciech Kozlowski
netsquid-simulationtools	Simulation tools for NetSquid	Guus Avis

* *maintainers in some cases not the creator(s)*

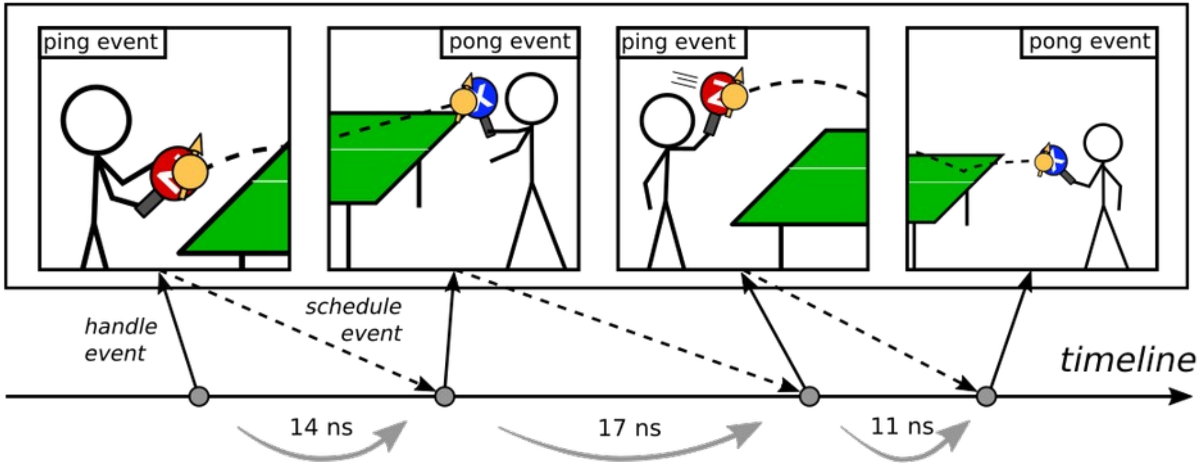
More info at https://netsquid.org

Backup slides

The NetSquid Package (I)



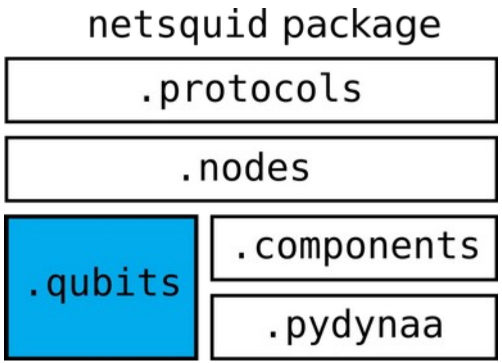
PyDynaa subpackage provides the **discrete event simulation engine**
*Based on the core of the DynAA simulator**



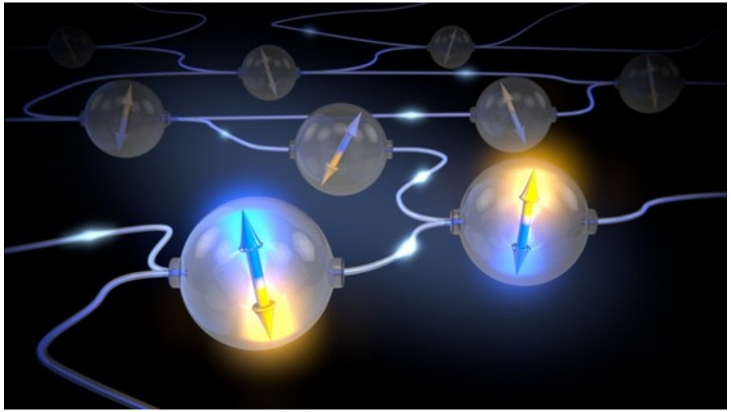
Time progresses by stepping from event to event.

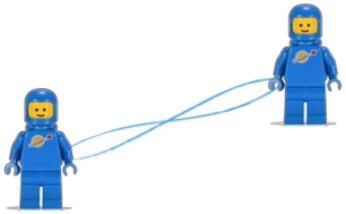
*de Oliveira, Papp, Djapic, Oosteveen, Proceedings of SASO (2013)

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

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

Ket vector

$$|\Phi^+\rangle$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

~ 2^n

Density matrix

$$\rho = |\Phi^+\rangle\langle\Phi^+|$$

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

~ 2^{2n}

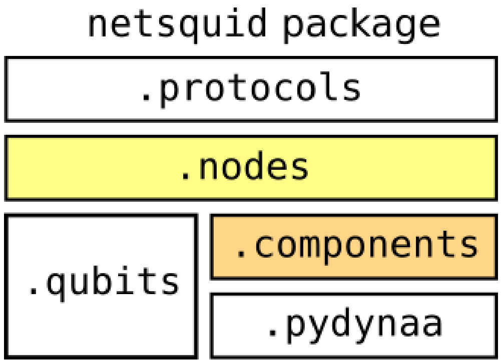
Stabilizer state

$$\{X \otimes X,\ Z \otimes Z\}$$

$$\begin{array}{c} \begin{array}{cccc} X & X & Z & Z \end{array} \\ \left(\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right) \end{array}$$

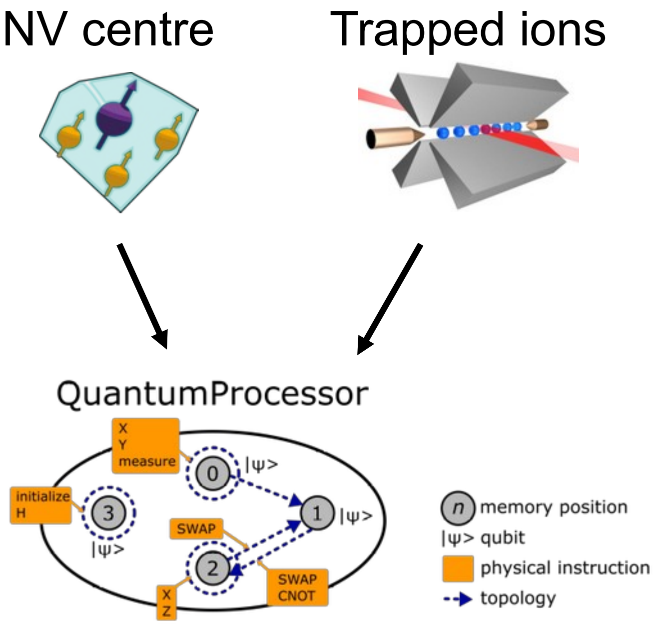
~ $2n^2$

The NetSquid Package (III)



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)

netsquid package

.protocols

.nodes

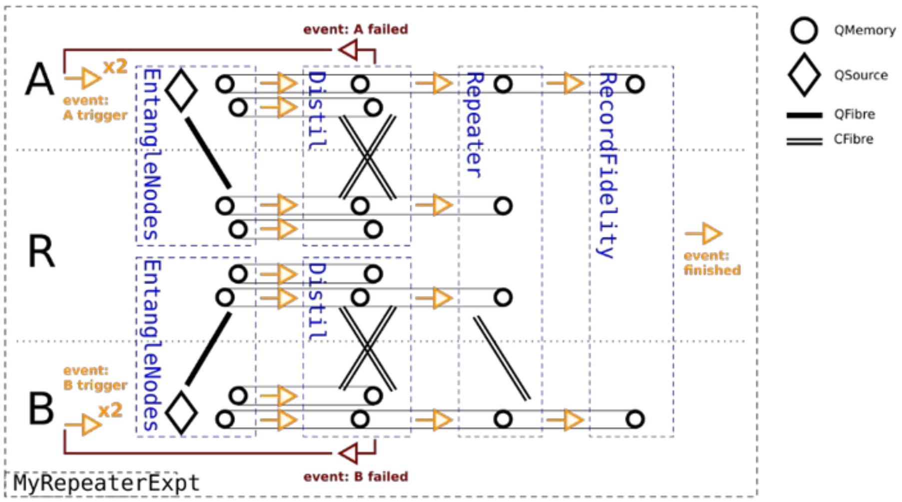
.qubits

.components

.pydynaa

Protocols dictate node behaviour (virtual entities)

- Numerically simulate via random sampling
- Massively parallelizable



Example: a distillation protocol