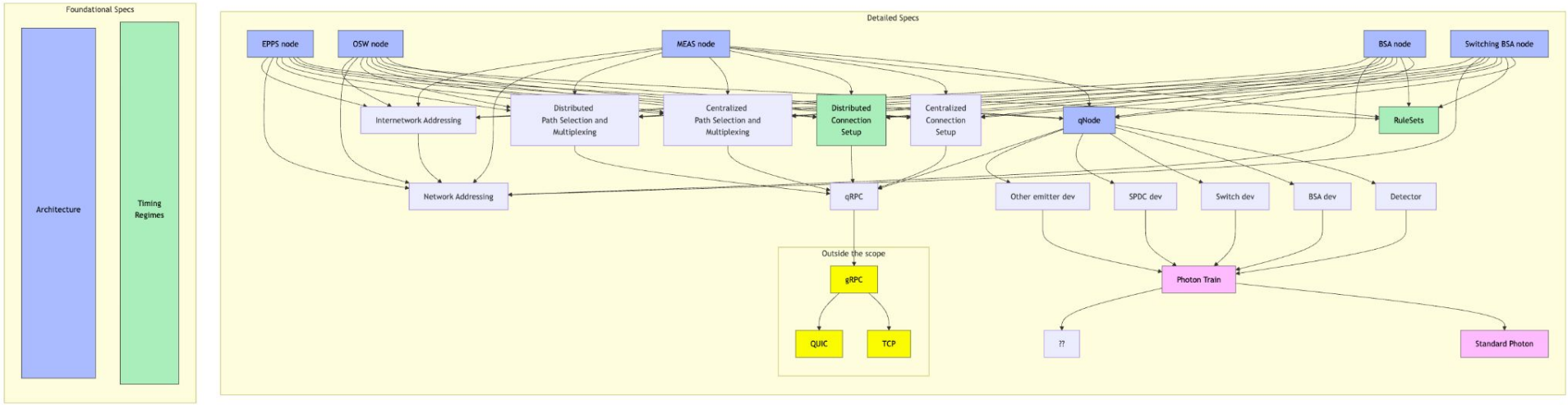
QUANTUM @IETF Hackathon

IETF 124
1–2 November 2025
Montreal, Canada

Hackathon Plan

- Gearing up for a big set of network specifications, some of which we hope will become RFCs (see next page)
  - Initial target is data center scale/multicomputer interconnects
  - Wide-area networks definitely in the mix and show up in parts of the work
- Learning more about SRv6, Dragonfly networks and other things that impact our network design
- Learning about PQC and other things that quantum affects

Overall Scheme of Specifications



(just an impression, don’t try to read this eye chart)
Green: available now
Blue: coming soon
Yellow: out of scope/existing protocols
Pink: where should we publish???

Rough Equivalents

- Host Requirements
- Network Architecture
- Physical Layer (would not be IRTF)

Currently Publicly Available Docs

Prerequisite: read RFC 9340, RFC 9583

n.b.: all of the below are preliminary; most of the material is our approach rather than community consensus:

- Timing Regimes
  - lays groundwork to identify how layers will be divided
- Connection Setup and Teardown
  - stateful, two-pass distributed method
- RuleSet
  - SDN-inspired, event-driven actions

What got done

- Progress on our set of specifications
  - Quantum Network Architecture
  - qNode (quantum node)
  - EPPS (Entangled Photon Pair Source node type)

What we learned

- Learning about RabbitMQ message broker
- Learning about Spanish network design
- Sharing our network design

Wrap Up

Team members:
Rod Van Meter
Monet Tokuyama Friedrich

(kibitzing on design, simulation & specs w/
Diego Lopez
Blanca Lopez)

Come to QIRG Monday (tomorrow) morning!

Three software packages available from our group (see following pages)

QuISP: Quantum Internet Simulation Package

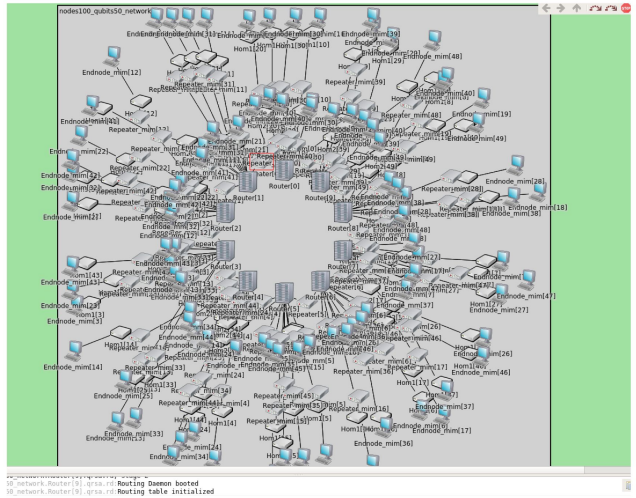
Open-source simulator for quantum network built on top of OMNeT++.

Ready to use – create your network topology file and simulation file parameters
(e.g., number of memories, noise channels, simple traffic modeling)

Customization needs C++ and recompilation.
(e.g., changing scheduling policy)

Available via docker (full functionality) or wasm browser (limited functionality).

QoL for beginners in progress…



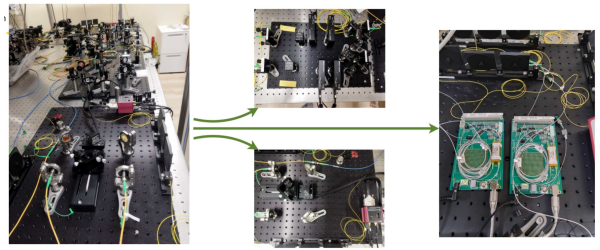
https://github.com/sfc-aqua/quisp

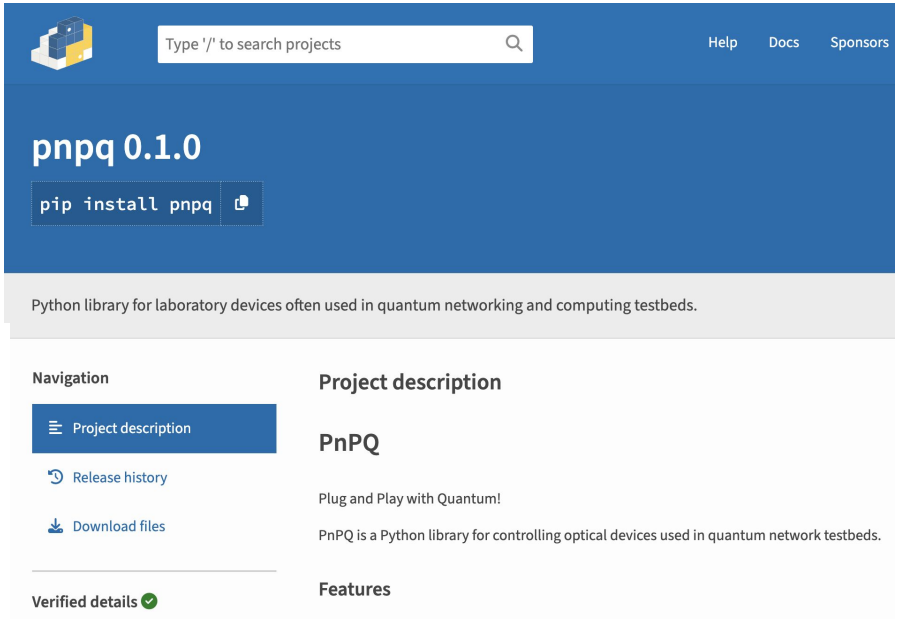
PnPQ: Controlling Devices in the Lab

https://github.com/moonshot-nagayama-pj/PnPQ

Open-source Python library for controlling common testbed equipment like waveplates, switches, polarization controllers, and optical delay lines

(we run on a set of RasPis)



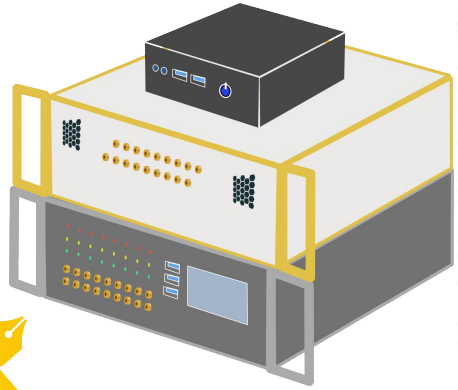


TDC Toolkit: Towards Measurement & BSA Nodes

https://github.com/moonshot-nagayama-pj/tdc_toolkit

Open-source Rust library for controlling high-precision time taggers used in photon coincidence counting

Will allow network nodes to implement real-time processing of time-tag data



crates.io | Type 'S' or '/' to search | Browse All Crates | Log in with GitHub

tdc_toolkit v0.3.0

Rust CLI and library, as well as Python bindings, for working with time-to-digital converters (TDCs) such as the PicoQuant MultiHarp 160.

#physics #quantum #tdc #multiharp

Readme | 1 Version | Dependencies | Dependents

tdc_toolkit

Rust command-line interface and library, as well as Python bindings, for working with time-to-digital converters (TDCs) such as the PicoQuant MultiHarp 160.

Supported devices

The PicoQuant MultiHarp 150/160 series of devices are the only ones supported at this time. Stub implementations of this device are also available for testing and development.

PicoQuant MultiHarp

Metadata

pkg:cargo/tdc_toolkit@0....
1 day ago
v1.89.0
BSD-3-Clause
63.1 KiB

Install

```
cargo install
tdc_toolkit
```

Running the above command will globally install the `tdc_toolkit` binary.