

QIRG  
Internet-Draft  
Intended status: Informational  
Expires: 17 September 2026

R. Van Meter  
N. Benchasattabuse  
A. Taherkhani  
Keio University  
16 March 2026

A Quantum Network Architecture  
draft-van-meter-qirg-quantum-network-architecture-00

## Abstract

This quantum network architecture defines a set of planes providing different views of the network, supporting different responsibilities and modes of operation; a set of device, node and link types; some network topologies, deployment scenarios and their relationship to applications; and key design decisions as a result of corresponding requirements.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://moonshot-nagayama-pj.github.io/draft-van-meter-qirg-quantum-network-architecture/draft-van-meter-qirg-quantum-network-architecture.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-van-meter-qirg-quantum-network-architecture/>.

Source for this draft and an issue tracker can be found at <https://github.com/moonshot-nagayama-pj/draft-van-meter-qirg-quantum-network-architecture>.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 September 2026.

## Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 4  |
| 2. Requirements Language . . . . .                                 | 4  |
| 3. Goals and Non-Goals of this Document . . . . .                  | 4  |
| 3.1. Goals . . . . .   | 5  |
| 3.2. Non-Goals . . . . .   | 5  |
| 4. Relationship to Documents by Other Organizations . . . . .      | 5  |
| 5. Prerequisite Knowledge . . . . .                                | 6  |
| 6. Terminology . . . . .   | 7  |
| 7. Applications of Networks . . . . .                              | 10 |
| 7.1. Types of Applications . . . . .                               | 10 |
| 7.2. Entangled States Consumption Patterns . . . . .               | 10 |
| 7.2.1. The Entanglement Information Timeline . . . . .             | 11 |
| 7.2.2. Classification of Consumption Patterns . . . . .            | 11 |
| 7.2.3. Class Selection and Resource Implications . . . . .         | 12 |
| 7.2.4. Unentangled State Tolerant (B Class) Applications . . . . . | 13 |
| 7.2.5. Reactive Correction (C Class) Applications . . . . .        | 13 |
| 7.2.6. Deterministic (T Class) Applications . . . . .              | 14 |
| 8. Architectural Concepts . . . . .                                | 14 |
| 8.1. Quantum Devices . . . . .                                     | 14 |
| 8.2. Quantum Nodes . . . . .                                       | 15 |
| 8.3. Quantum Links . . . . .                                       | 15 |
| 8.4. Photonic Synchronization Domains . . . . .                    | 15 |
| 8.5. Direct and Indirect Multicomputer Architectures . . . . .     | 15 |
| 8.6. Detector-centric Architecture . . . . .                       | 15 |
| 8.7. Source-centric Architecture . . . . .                         | 16 |
| 9. A Sketch of the System Model . . . . .                          | 16 |
| 9.1. Multicomputer . . . . .                                       | 16 |
| 9.2. Data Center Network (QDCN) . . . . .                          | 20 |
| 9.3. Local-Area Network (QLAN) . . . . .                           | 20 |
| 9.4. Wide-Area Network (QWAN) . . . . .                            | 20 |
| 10. Quantum Optical Building Blocks . . . . .                      | 20 |
| 10.1. Qubits . . . . .   | 21 |
| 10.2. Photons, Wave Packets and Optical Modes . . . . .            | 21 |

|         |   |    |
|---------|---|----|
| 10.3.   | Photonic Qubits . . . . .                                   | 21 |
| 10.4.   | Memories . . . . .  | 22 |
| 10.5.   | Photon Sources . . . . .                                    | 22 |
| 10.5.1. | Unentangled Single Photons . . . . .                        | 22 |
| 10.5.2. | Entangled Photon Pairs . . . . .                            | 22 |
| 10.5.3. | Memory-Emitted Photons . . . . .                            | 23 |
| 10.6.   | Detectors . . . . .   | 23 |
| 11.     | Requirements . . . . .                                      | 23 |
| 11.1.   | General Requirements . . . . .                              | 23 |
| 11.1.1. | Functional Requirements . . . . .                           | 23 |
| 11.1.2. | Interface Requirements . . . . .                            | 24 |
| 11.1.3. | Physical Requirements . . . . .                             | 24 |
| 11.1.4. | Environmental Requirements . . . . .                        | 24 |
| 11.2.   | Network Management Requirements . . . . .                   | 24 |
| 11.2.1. | Fault Management . . . . .                                  | 24 |
| 11.2.2. | Configuration Management . . . . .                          | 24 |
| 12.     | Top Level Architecture . . . . .                            | 24 |
| 12.1.   | Deterministic Classical Control of Quantum States . . . . . | 24 |
| 13.     | Communication Service . . . . .                             | 25 |
| 14.     | Architectural Planes . . . . .                              | 26 |
| 14.1.   | Quantum . . . . .   | 26 |
| 14.2.   | Data . . . . .  | 26 |
| 14.3.   | Control . . . . .   | 27 |
| 14.4.   | Management . . . . .  | 27 |
| 15.     | Protocol Layers . . . . .                                   | 27 |
| 16.     | Nodes and Node Types . . . . .                              | 28 |
| 16.1.   | End Nodes . . . . .   | 28 |
| 16.2.   | Support Nodes . . . . .                                     | 30 |
| 16.3.   | Repeater Nodes . . . . .                                    | 31 |
| 16.4.   | Composite Nodes . . . . .                                   | 32 |
| 17.     | Links . . . . .   | 32 |
| 17.1.   | The Link Service . . . . .                                  | 32 |
| 17.2.   | Photonic Path Description . . . . .                         | 32 |
| 17.3.   | Point-to-point . . . . .                                    | 33 |
| 17.4.   | Switched . . . . .  | 33 |
| 17.5.   | Multidrop or Bus . . . . .                                  | 33 |
| 18.     | Connections . . . . .                                       | 34 |
| 19.     | Resource Management: Multiplexing and Routing . . . . .     | 34 |
| 20.     | Classical Communication . . . . .                           | 34 |
| 20.1.   | Quantum Plane . . . . .                                     | 34 |
| 20.2.   | Control Plane . . . . .                                     | 35 |
| 20.3.   | Data Plane . . . . .  | 35 |
| 20.4.   | Management plane: Node and link management . . . . .        | 35 |
| 20.5.   | Mechanisms . . . . .  | 35 |
| 21.     | Naming and Addressing . . . . .                             | 35 |
| 21.1.   | State naming and management in RuleSets . . . . .           | 36 |
| 22.     | Example Networks . . . . .                                  | 37 |
| 22.1.   | Fully Connected . . . . .                                   | 37 |

|       |  |    |
|-------|--|----|
| 22.2. | Q-Fly Multicomputer . . . . .                          | 37 |
| 22.3. | Optically Switched Fat Tree . . . . .                  | 38 |
| 22.4. | Repeater Fat Tree . . . . .                            | 38 |
| 22.5. | 2-D Grid Multicomputer . . . . .                       | 38 |
| 22.6. | Ring . . . . .   | 39 |
| 22.7. | QLAN . . . . .   | 39 |
| 23.   | APIs for Network Service ("Quantum Sockets") . . . . . | 39 |
| 24.   | Security Considerations . . . . .                      | 39 |
| 25.   | References . . . . .                                   | 39 |
| 25.1. | Normative References . . . . .                         | 39 |
| 25.2. | Informative References . . . . .                       | 40 |
|       | Contributors . . . . .                                 | 51 |
|       | Authors' Addresses . . . . .                           | 53 |

## 1. Introduction

This document introduces the key architectural decisions, classical and quantum communication systems, and main node types for several classes of quantum networks.

We define the verb `_to architect_` as: within a set of environmental constraints, using a set of building blocks, design a system that satisfies a need, elegantly and economically. We use the noun `_architecture_` as: the set of blocks or subsystems, their roles and their interfaces and their overall arrangement, that defines the system. This architecture defines the overall structure, and is connected to a specific implementation as an example.

For a description of the key concepts in quantum networks and additional references, see [RFC9340] and the book `_Quantum Communications_` [hajdusek-qcomm].

For more background and discussion of the design choices in this architecture, see the Ph.D. dissertation of Naphan Benchasattabuse [res-mgmt-het].

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Goals and Non-Goals of this Document

This section describes goals and non-goals for this document itself, rather than the technical goals and requirements for a network.

### 3.1. Goals

- \* To define the principal concepts in a quantum network, principally for quantum multicomputer interconnects but also data center and wide-area networks where possible.
- \* To enumerate some of the key architectural decisions for this architecture.
- \* To describe how device, link and node types are defined.
- \* To provide a guide to other documents.

### 3.2. Non-Goals

- \* Internetworking

## 4. Relationship to Documents by Other Organizations

Other organizations, including national laboratories and standards development organizations, are developing documents describing quantum networks and quantum computing technology. These are mostly pre-standardization documents, not yet on any formal standardization track. To the extent possible, this document conforms to their terminology. However, as this document describes a specific quantum network architecture, it does not attempt to conform to specific design decisions made in other contexts. See an August 2025 Science Policy Forum [aboy-governance] for additional discussion of some standardization efforts and their value.

Some of these are listed here for reference:

- \* ETSI
  - Industry Specification Group (ISG) on Quantum Key Distribution (QKD) (<https://www.etsi.org/committee/qkd>)
- \* IEEE Standards Association
  - IEEE Standards & Projects for Quantum Technologies (<https://standards.ieee.org/initiatives/quantum-standards-activities/>)
  - Standardization Roadmap on Quantum Applications (<https://ieee-sa.imeetcentral.com/p/eAAAAAASqm9AAAAAFU6etg>)
- \* ISO

- IEC/ISO JTC 3 Quantum technologies (<https://www.iso.org/committee/10138914.html>)

- \* ITU-T

- ITU-T Focus Group on Quantum Information Technology for Networks (FG-QIT4N) (<https://www.itu.int/en/ITU-T/focusgroups/qit4n/Pages/default.aspx>)
- Y.3800 series (<https://www.itu.int/itu-t/recommendations/index.aspx?ser=Y>) on quantum key distribution networks

- \* National Institute of Standards and Technology (NIST)

- Single-Photon Sources and Detectors Dictionary [nist-singles]

- \* Quantum Internet Research Group (QIRG) (<https://datatracker.ietf.org/group/qirg/about/>) (part of IRTF)

- [RFC9340]

- [RFC9583]

## 5. Prerequisite Knowledge

This document assumes basic knowledge of the underlying technology and goals of quantum communications. The following list of topics may help readers who are not yet familiar with the concepts.

- \* Linear algebra

- \* Quantum information basics

- Dirac ket notation
- von Neumann density matrix notation
- Superposition
- Entanglement
- Interference
- Unitary operation
- Measurement
- Decoherence

- No-cloning theorem
- Clifford group
- Basics of quantum error correction (QEC)
- \* Classical Internet-family networking
- \* Quantum networking
  - Teleportation
  - Purification
  - Entanglement swapping
  - Quantum key distribution: BB84, E91, BBM92
  - "Generations" of quantum repeaters
  - (Repeater graph states may be helpful, but are not used in the current architecture)

Because terms such as `_fidelity_` have varying definitions, they will be defined in this set of documents (where? Timing Regimes?).

Readers needing additional background are referred to:

- \* [RFC9340]
- \* [RFC9583]
- \* Van Meter, `_Quantum Networking_` [van-meter-q-net-book]
- \* Hajdusek and Van Meter, `_Quantum Communications_` [hajdusek-qcomm]

## 6. Terminology

In this document, we use the abbreviations and other related technical terms listed in the following table:

| Term | Description   |
|------|---|
| BSA  | Bell state analyzer, generally optical and incorporating one or more beamsplitters and either two or four single-photon detectors |

|              |  |
|--------------|--|
| device       | manipulates photons in some fashion; a component of a node   |
| FASQ         | fault-tolerant application-scale quantum   |
| fidelity     | measures how close a quantum state is to the state we have tried to create. Varies between 0 and 1, with unit fidelity indicating the actual state is the same as the desired state. It expresses the probability that the state will behave exactly the same as our desired state. (adapted from RFC 9340)  |
| group switch | In the Q-Fly architecture, the set of devices that connect the end nodes to the pool of BSAs, and the group to other groups  |
| NISQ         | near-term intermediate-scale quantum   |
| node         | a self-contained subsystem with a clear boundary that is visible to other such nodes as a single entity on one or more planes  |
| optical mode | roughly, a place and time that a photon might be, which may be occupied by some, all or none of the amplitude of a single photon. More accurately, it is a field distribution obtained from solving Maxwell's equations given some boundary conditions. An optical mode is an eigensolution of the wave equation given the physical properties of the waveguide (dimensions, refractive index). It is a particular field distribution that can propagate through the waveguide. The state of a single photon can be expressed as a coherent superposition of such optical modes. For optics engineers, a 'mode' (or optical mode to be more precise) refers to a specific spatiotemporal distribution of electromagnetic field fluctuation. For quantum network engineers, especially those that would most likely be reading RFCs, two optical detectors are detecting different optical modes if one detector can be activated by light without activating the other. In that case, it is not wrong to say that the two detectors are detecting two independent optical modes. |

|                               |   |
|-------------------------------|---|
| path                          | the set of nodes and links that a connection passes through, including the end points   |
| plane<br>(architectural)      | a view of the entire network for a particular purpose, such as management or data transmission  |
| plane (optical)               | a defined plane in physical space through which a photon passes; also referred to as a surface  |
| QBER                          | quantum bit error rate  |
| QNIC                          | quantum network interface card; in practice, the physical interface to a link plus the set of quantum memories under the control of the network stack.                              |
| QPU                           | quantum processing unit   |
| RuleSet                       | a set of SDN-inspired, event-driven, short, real-time or near-real time, near-deterministic programs executed at nodes along a path to build application-requested entangled states |
| SNSPD                         | Superconducting Nanowire Single Photon Detector   |
| multiqubit<br>Pauli operators | tensor product of Pauli operators acting on two or more qubits  |
| switch point                  | a 2x2 junction that can be either X (cross) or = (straight)   |
| switch device                 | a single integrated, fiber- or free space-connected (physical) component, comprising one or more switch points  |
| teledata                      | application execution via teleporting data from node to node, then executing gates locally. May be done remotely, mediated by Bell pairs.   |
| telegate                      | application execution via remote gates (as defined by Eisert et al.). May be done remotely, mediated by Bell pairs.   |

|           |                                   |         |
|-----------|-----------------------------------|---------|
| time bin  | compare to window and time slot   |         |
| +-----+   | +-----+                           | +-----+ |
| time slot | compare to window and time bin    |         |
| +-----+   | +-----+                           | +-----+ |
| window    | compare to time bin and time slot |         |
| +-----+   | +-----+                           | +-----+ |

Table 1

## 7. Applications of Networks

The requirements for a network are determined by the application workload. This section orients architectural decisions by briefly introducing applications and the communication patterns they exhibit, which is a key factor determining the suitability of particular architectures.

### 7.1. Types of Applications

Applications fall into two large categories: inherently distributed applications, or subdivision of monolithic applications for distributed execution, as in supercomputing applications running on multicomputer architectures.

For a discussion of some inherently distributed applications of quantum networks, see [RFC9583]. This network architecture supports the applications listed in that RFC, though not all networks will support all applications.

### 7.2. Entangled States Consumption Patterns

(Adapted and extended from unpublished text in [van-meter-opt-timing].)

Distinct from the classification of quantum repeater generations by Muralidharan et al. [muralidharan-generations], one can categorize distributed quantum systems by how applications interface with the network; specifically, the timing at which network interface qubits are freed after attempting entangled state generation.

In the early days of quantum information research, Bennett et al. recognized [bennett-mixed] that the component qubits of an entangled state may be held at different times in different locations, termed `_time-separated Bell pairs_`. Based on this principle, we can describe the timeline of information availability between two nodes. This model assumes entangled states are requested dynamically during execution, rather than pre-caching entangled states [schoute-shortcuts] for immediate consumption.

### 7.2.1. The Entanglement Information Timeline

The lifecycle of an entanglement request, from initiation to full state knowledge, follows three distinct stages:

1. **\*Attempt:\*** The entangled states are requested. For memory-based links, this corresponds to the quantum memory emitting a photon and transmitting it through the link.
2. **\*Heralded:\*** The entangled states are physically established, but the specific states are unknown. The node has received confirmations that photons arrived at the BSA and the BSM succeeded, but the Pauli frame information is not yet available.
3. **\*Correct:\*** The classical message regarding the Pauli frame arrives. The node now knows the exact entangled state created and can apply corrections (or software frame updates) to align with the expected state.

### 7.2.2. Classification of Consumption Patterns

Based on the timeline above, we classify Bell pair consumption into three classes [van-meter-opt-timing]. These classes are defined by whether the application must **\*block execution\*** while waiting for information at the `_Heralded_` or `_Correct_` stages. Note that the term "blocking" here refers to the blocking versus non-blocking execution models, similar to kernel-level I/O blocking or the event-driven programming paradigm, and is distinct from the concept of blocking in network switches.

| Class                                  | Wait for<br>Heralding? | Wait<br>for<br>Pauli<br>Frame? | Application Behavior  |
|--|------------------------|--------------------------------|---|
| *Unentangled<br>State Tolerant<br>(B)* | No                     | No                             | *Non-Blocking:* The application consumes the qubit immediately, handling failures or corrections in post-processing.                      |
| *Reactive<br>Correction<br>(C)*        | Yes*                   | No                             | *Partially Blocking:* The application waits only for confirmation of existence (heralding), then proceeds by tracking errors in software. |
| *Deterministic<br>(T)*                 | Yes                    | Yes                            | *Strictly Blocking:* The application blocks until the state is fully verified and corrected.  |

Table 2

\_\*Note for Class C: If the link is fully deterministic (guaranteeing success), the application does not need to pause for heralding and become non-blocking.\_

### 7.2.3. Class Selection and Resource Implications

It is important to note that these classes are determined by the *\*nodes\**, not the network. The application nodes assess their own capabilities (number of buffer memories, gate and qubit error rate) and the network's performance (fidelity, rate, success probability) to select the appropriate operating class.

The selection of a class significantly impacts network resource utilization. For example, if a node possesses sufficient buffer memory, it may elect to operate in \*Class B\* from the perspective of the network interface. By moving the entangled state from the communication qubit to storage immediately (or measuring it immediately), the node frees up the network interface to service other requests. This reduces the workload on the network and increases the repetition rate, even if the application logic itself eventually requires the data for a Class T operation.

#### 7.2.4. Unentangled State Tolerant (B Class) Applications

The defining characteristic of B Class applications is the ability to consume the (potential) entangled states immediately without waiting (fully non-blocking). The application effectively takes over the burden of validation from the network.

**\*Execution Flow:** The application does not stop. It measures or stores the qubit immediately. If the entanglement attempt failed (information received later), the data is discarded or treated as an erasure error.

**\*Examples:**

- \* **\*Classical Correlation:** Applications like Quantum Key Distribution (QKD) protocols (e.g., E91) or link fidelity estimation. The application filters out failed attempts during classical post-processing.
- \* **\*Fault-Tolerant Operations:** Certain remote quantum error correction schemes, such as remote lattice surgery [horsman-lattice-surgery], [ramette-remote], [leone-remote], [sinclair-ft-interconnect] of the surface code. If the probability of creating a link is high enough, unsuccessful attempts can be treated as depolarizing errors, which the logical code can tolerate without stalling the pipeline.

#### 7.2.5. Reactive Correction (C Class) Applications

In C Class applications, the system requires confirmation that a link exists, but does not wait for the state details. The application proceeds by assuming a specific Bell state and managing deviations via software tracking.

**\*Execution Flow:** The application pauses briefly to ensure the Bell pair is `_Heralded_`. Once confirmed, it executes immediately. It does not wait for the `_Correct_` stage (Pauli frame); instead, it uses a "Pauli Frame Tracker" to propagate the necessary corrections through

the circuit virtually. If the network link is deterministic, the specific wait for heralding is removed, as the node assumes success by default.

*\*Examples:* Clifford circuit execution, distributed Pauli-based computation with time-optimal scheme [litinski-gosc], and state teleportation.

#### 7.2.6. Deterministic (T Class) Applications

T Class applications impose the strictest timing constraints, requiring the entangled state to idle the longest before being consumed.

*\*Execution Flow:* The application *\*completely stalls\**. It must wait until the network provides both the confirmation of creation (*\_Heralded\_*) and the specific state information (*\_Correct\_*). Execution only resumes once the exact state is known or corrected.

*\*Examples:* Execution of circuits involving non-Clifford gates. While non-Clifford operations *\_can\_* theoretically be corrected post hoc (similar to Class C), doing so often requires consuming *\_additional\_* entangled states to fix the error. Since consuming extra resources is more costly than waiting, these operations default to Class T to ensure the state is correct before proceeding.

### 8. Architectural Concepts

This section introduces concepts in classical and quantum computer and network architecture that may be unfamiliar, or that have a specific role in this network architecture.

#### 8.1. Quantum Devices

A quantum device stores, carries, measures or computes on quantum variables.

A quantum device (often shortened to just "device" in this specification) is under software control; i.e. optical fibers or beam splitters are not classified as quantum devices.

Control of devices is usually done with respect to some physical characteristic of the device itself, rather than dealing with the abstract notion of qubits. A controllable wave plate, for example, may be adjusted in units of degrees of rotation of the plate. An example of a software package that provides such functionality is PnPQ.

## 8.2. Quantum Nodes

A node comprises one or more quantum devices, and serves as a single locus of control for network protocols. The classes of nodes are described later in this document.

## 8.3. Quantum Links

Links are described in Section 17.

## 8.4. Photonic Synchronization Domains

A photonic synchronization domain (PSD) is the range of devices and fibers over which photons must be controlled with high precision in order to effect e.g. photonic entanglement swapping [mori-psds]. The primary concern of a PSD is getting photons to arrive at beamsplitters "simultaneously", with sufficient overlap, as specified in [I-D.draft-hajdusek-qirg-timing-physics].

## 8.5. Direct and Indirect Multicomputer Architectures

In multicomputer architectures, a `_direct_` architecture features links that go directly from computational node to computational node. Hypercubes, meshes and toruses are typically direct architectures. An `_indirect_` architecture interposes one or more switches between computational nodes. Fat trees, Clos and Benes networks, and the various -fly topologies are generally indirect [dally-towles].

The distinction is somewhat artificial in that direct architectures sometimes incorporate a small switch inside the node, in which case the matching term depends on where you draw the boundary of the node, and because computational nodes can be configured to act only as routers within the network, modeling an indirect architecture using direct hardware.

## 8.6. Detector-centric Architecture

Several detectors may be packaged as a single subsystem for purposes such as cooling, power, control and time stamping. An architecture built around a shared pool of detectors is a `_detector-centric_` architecture. This structure facilitates entanglement distribution in a quantum system interconnect, data center network or some forms of local area network.

A detector-centric architecture is an indirect architecture if the pool of detectors is behind a switch or network of switches.

### 8.7. Source-centric Architecture

Many of the detector-centric network topologies can be inverted, such that the pool of detectors is replaced by a pool of entangled photon pair sources. The network then is used to distribute entanglement, with the photons measured, absorbed into memories, interfered with locally generated photons, or otherwise utilized at or near the end nodes. Such a network was proposed by Drost *et al.* [drost].

## 9. A Sketch of the System Model

As noted in the 2022 roadmap for quantum interconnects [awschalom-roadmap], entangled quantum network technology can be deployed in a variety of scenarios with different requirements and assumptions. A full description of each of these is delegated to other documents, but a brief description here will help to orient discussions of design points in order to justify certain decisions.

### 9.1. Multicomputer

The first deployment of production-level, distant quantum entanglement is likely to be in a *\_quantum multicomputer\_*, based on the same principles as classical distributed-memory supercomputers from the Caltech Cosmic Cube ([https://en.wikipedia.org/wiki/Caltech\\_Cosmic\\_Cube](https://en.wikipedia.org/wiki/Caltech_Cosmic_Cube)) to Fugaku ([https://en.wikipedia.org/wiki/Fugaku\\_\(supercomputer\)](https://en.wikipedia.org/wiki/Fugaku_(supercomputer))) [rdv-thesis]. Multicomputer deployments will likely involve computational nodes, optical switches, Bell state analyzers, and possibly entangled photon pair sources (all defined below). Quantum repeaters with memory are less likely to be deployed in multicomputers, though one such architecture [choi-fat-tree] has been proposed. Because the current technology roadmaps favor this type of deployment, where design choices are in conflict or unclear, multicomputer designs are given priority over wide-area networks in this set of specifications.

The execution model is expected to be much like the classical supercomputing Message Passing Interface (MPI) ([https://en.wikipedia.org/wiki/Message\\_Passing\\_Interface](https://en.wikipedia.org/wiki/Message_Passing_Interface)).

General hardware environment:

- \* A multicomputer consists of a set of quantum nodes that are connected via quantum optical channels. The system may be either a direct (point-to-point, end node-to-end node) or an indirect (switched) design.
- \* Every quantum node has a set of quantum devices and a classical controller.

- \* End nodes are generally computational nodes, but measurement-only, sensor, and specialized memory storage nodes may be included.
- \* End nodes are capable of running application programs as well as executing the minimum operations to build end-to-end entangled states.
- \* End nodes may be built using multi-level hardware interconnects; when gates between physically distant qubits are mediated by first creating shared entangled states, the creation of those entangled states is the responsibility of the network subsystem.
- \* Multicomputer interconnects are closed systems, with no need for cryptographic security mechanisms.

Many aspects of compilation and job execution are beyond the scope of this set of specifications, but some points will affect the network node definitions and interfaces, so it is important to present them here to establish a basis for design decisions:

- \* The general purpose of a multicomputer system is to execute application programs that exceed the capabilities of a single quantum node. The subdivision of the application into smaller quantum programs and the assignment of those sub-programs to nodes in the network is beyond the scope of this specification, and may be either automatically done by the compiler or manually done by the programmer.
- \* Programs are centrally compiled and distributed to nodes. (Note that, technically, this is different from classical MPI, where a `_command_` is sent to worker nodes, but the mapping of that command to an executable program and versioning and distribution of programs are outside the scope of MPI. It is, however, a convenient assumption and common practice to ensure the same program is executed at all worker nodes.)
- \* Execution is coordinated by a job controller: every node executes the same program, but with different parameters (e.g., which portion of the problem to work on).
- \* Applications consist of both classical computation and quantum computation; within a node, the classical portion of the program delegates certain computational tasks to the quantum processor (sometimes called a QPU), similar to a classical coprocessor (<https://en.wikipedia.org/wiki/Coprocessor>) such as a GPU ([https://en.wikipedia.org/wiki/Graphics\\_processing\\_unit](https://en.wikipedia.org/wiki/Graphics_processing_unit)).

- \* Classical data related to quantum operations (principally measurement results and event notifications that trigger further actions) is sent peer-to-peer, not back to the centralized controller, during execution.
- \* Within the runtime system, the interface between the (portion of the) application running at each node's classical controller is analogous to the interface between an application and the MPI messaging system or a socket ([https://en.wikipedia.org/wiki/Network\\_socket](https://en.wikipedia.org/wiki/Network_socket)) in an ordinary Internet application. This quantum socket is an active area of research and is not defined here.
- \* The creation of sequences of end-to-end entangled states, roughly equivalent to TCP, is the responsibility of RuleSets, inspired by software-defined networking (SDN) ([https://en.wikipedia.org/wiki/Software-defined\\_networking](https://en.wikipedia.org/wiki/Software-defined_networking)). A RuleSet can also be viewed as something like a Berkeley Packet Filter (BPF): it's a small program that handled actions that the application could do, but the application can't achieve the low, reliable latency to do it.
- \* In principle, all nodes are running the same program, distributed to all nodes. Since the compiled application circuits are often parameter- or input-dependent as well, separate nodes may have separate instances of the application. This may result in a small additional burden on the execution management system.
- \* In principle, the application and the communication system are separately compiled and managed. However, in practice the RuleSet may be compiled as part of the application by using a library of network functions. (As with classical parallel program runtime systems, the boundary between the application program, supplied libraries, and the kernel itself (if any) is implementation-dependent.)
- \* Compiling the network communication into the application program eliminates the need for separate program and RuleSet distribution protocols. However, the event messages that are part of the architecturally defined RuleSet operation are sent and received as usual, such that the behavior of the node is the same regardless of such implementation choices.
- \* Compilation and execution may achieve application goals via teledata, telegate or measurement of multiqubit Pauli operators transparently; the network is unaware of this distinction. Management of application-level variables and their movement from node to node, if any, is the responsibility of the compiler and is beyond the scope of this specification.

- \* The resources in a multicomputer may be partitioned to run multiple jobs [sane-jobs], but this is beyond the scope of the current specifications.

The realities of quantum hardware result in a few important differences from classical multicomputers:

- \* Multicomputer interconnects may be either optically switched or composed entirely of point-to-point links. In switched networks, depending on optical hardware as well as photonic qubit representation, reconfiguring the switch or switches may be an extremely high-latency operation.
- \* Application execution at end nodes proceeds in phases tied to specific communication patterns. Even with a fixed sequence of application-level operations, the execution time of a phase can be variable because entanglement generation is probabilistic. Local gates can also be probabilistic under some circumstances, but for many node hardware types, fast local gate execution means the impact of variable gate execution time will be small.
- \* In switched systems, reconfiguration of switches is centrally coordinated between phases. Thus, while classical data moves directly node-to-node during autonomous phase execution, advancing from phase to phase must be done at the direction of the job controller.
- \* Because execution is centrally coordinated, the network system is not required to provide multiplexing. (This is a substantial difference from data center networks, QLANs and QWANs.)
- \* The compiler may include multihop communication within a phase using hop-by-hop teleportation without reconfiguring any switches; this is beyond the scope of this architecture.
- \* Execution of the quantum portion of the node program generally involves hard real-time actions, both unconditional and conditioned on prior quantum measurement results. This generally requires compilation of the quantum program to very low-level actions to be executed by FPGAs or ASICs.
- \* Systems may be partially or completely emulated, decoupling development of different subsystems. e.g., an EPPS plus a MEAS together can emulate a COMP node that emits single photons.
- \* Systems may be noisy, intermediate-scale quantum (NISQ); near-term, small-scale fault tolerant; or fault-tolerant, application-scale quantum (FASQ).

- \* Quantum error correction is above the level of these specifications, but may involve distributed lattice surgery [ramette-remote], [leone-remote], or [sinclair-ft-interconnect].

## 9.2. Data Center Network (QDCN)

The hardware for quantum data center networks will be almost identical to multicomputers; the primary differences are in the workload, scheduling, programming model and assumptions of trust. Distributed control and protocols for multiplexing and connection setup may be necessary.

## 9.3. Local-Area Network (QLAN)

A QLAN will be deployed within a building or across a campus. It may connect quantum computers (including but not necessarily multicomputers) and sensors. A sensor, for example, may be connected to a computer with substantial memory for e.g. shadow tomography, learning from few measurements and related protocols.

A QLAN will have a less regular topology than a multicomputer or QDCN. Distance, latency, fidelity, and success probability will all vary on a per-link basis.

Distributed control and protocols for multiplexing and connection setup are necessary.

## 9.4. Wide-Area Network (QWAN)

Wide-area networks may involve client-server or peer-to-peer communication. One particular scenario of interest is a measurement-only (MEAS) client end node connecting to a centralized, supercomputer-scale quantum computer (perhaps, but not necessarily, a multicomputer) for the purposes of executing `_blind quantum computation_` [fitzsimons-blind], [morimae-blind].

QWAN client-server communication very likely will suffer from the "incast" problem of excessive traffic concentrating near certain nodes. Management of this problem is beyond the scope of this document.

## 10. Quantum Optical Building Blocks

This section informally describes the physical building blocks and concepts used in the physical layer of a quantum network.

### 10.1. Qubits

In this network architecture, we use only qubits, which may have two states identified as 0 and 1. Quantum information systems using qutrits, qudits, qunats or continuous variable (c.v.) quantum states are beyond the scope of the current set of specifications.

Qubits (also defined in RFC 9340) must conform to a sufficient subset of the DiVincenzo criteria [divincenzo-criteria].

### 10.2. Photons, Wave Packets and Optical Modes

Optical mode (link-layer view).

An optical mode is a well-defined slot of a physical link, specified by path, time window, frequency, polarization, or similar parameters, such that the receiver can be configured to monitor that slot and determine whether it is occupied by at least one photon or is empty.

The mode exists regardless of whether a photon is present; a photon is an excitation of the mode, not the mode itself.

In quantum networking, link capacity and state must be described in terms of modes (slots), not photons; photons merely occupy modes, while empty modes correspond to vacuum states that are still physically and operationally meaningful.

Technical note: In idealized models, distinct modes correspond to orthogonal field solutions, ensuring perfect distinguishability.

Short example: A quantum optical link may define one mode per time window. During each window, the receiver monitors the mode. A detection event indicates that the mode was occupied by at least one photon; the absence of a detection indicates that the same mode was empty. Both outcomes correspond to distinct physical states of the link.

### 10.3. Photonic Qubits

Photons are used in networks to carry qubits. There are several possible on-the-wire photonic qubit representations, e.g.

- \* Polarization
- \* Time bin
- \* Which path

\* Energy level (wavelength)

These representations and more about the concept of an `_optical mode_` are discussed in the Standard Photons document, and are provided here only for informational purposes.

#### 10.4. Memories

For our purposes, we do not need to worry about the physical implementation of memories, only that they may hold quantum data (qubits) that may be under the management of the network software and protocols. They may be entangled with photons or with other memory qubits.

Generally, the creation or entanglement of a state in memory is imperfect, and the memory has a finite lifetime.

The entanglement of a memory qubit with a photon is a technology-dependent process. To create fiber-compatible and optical equipment-compatible photons, wavelength conversion via transduction may be necessary. In 2025, transduction is a low-probability process, and hurts fidelity as well.

#### 10.5. Photon Sources

Photons may be emitted by `_sources_` of many types [nist-singles] . Single photons may come from attenuated lasers, or be emitted by a variety of quantum devices, such as quantum dots, or by individual atoms.

Photons may be unentangled, entangled with other photons, or entangled with quantum memories.

##### 10.5.1. Unentangled Single Photons

Unentangled photons exhibit quantum properties. They can carry information in any of the characteristics listed above, and may be put into a superposition of multiple basis states for e.g. quantum key distribution purposes. In this document, unentangled individual photons are not used.

##### 10.5.2. Entangled Photon Pairs

Pairs of photons entangled with each other can be made via a variety of physical processes. Devices that make such pairs can be components of nodes such as the Entangled Photon Pair Source (EPPS), described in a separate document.

### 10.5.3. Memory-Emitted Photons

Photons emitted by quantum memories, such as single atoms, may remain entangled to the memory, if the memory was in a superposition of basis states.

## 10.6. Detectors

Detectors may be either `_single-photon detectors_`, which click when `_one or more_` photons hit the detector, or `_number resolving detectors_`, which can distinguish between one, two, or more photons hitting the detector within the same time window [nist-singles]. In this document, detectors may be assumed to be single-photon detectors.

## 11. Requirements

This section documents the requirements for all networks adhering to this architecture.

### 11.1. General Requirements

#### 11.1.1. Functional Requirements

- \* Operates on qubits. (Qutrits, qudits, qunats and continuous-variable systems are out of scope of this architecture, except where physical or link layers present such physical variables as qubits.)
- \* Is independent of physical implementation of memories, photonic data representations, etc. (Multipartite states created by the network are not a requirement of the network.)
- \* Supports pairwise Bell pair creation between nodes with one or more of the B, C or T timing classes above.
- \* The architecture must support deployments ranging from multicomputer to wide area networks.
- \* The architecture must support multiple photonic synchronization domains, as either point-to-point or optically switched paths. The architecture must support some form of buffering between PSDs.
- \* The architecture must support entanglement swapping. (Note that single PSD deployments may not need entanglement swapping.)

- \* The architecture must support evolution of single-photon, unentangled, single-purpose quantum key distribution networks to fully entangled, multipurpose networks.

#### 11.1.2. Interface Requirements

- \* Supports one or more applications, such as the ones in [RFC9583], with APIs consistent with the B, C, or T classes.
- \* The network must enable applications to match quantum states at each end of the Bell pair by name.

#### 11.1.3. Physical Requirements

Physical requirements such as distance, wavelength, vibration, power, etc. will be case-dependent.

#### 11.1.4. Environmental Requirements

Physical requirements such as distance, wavelength, vibration, power, etc. will be case-dependent.

### 11.2. Network Management Requirements

#### 11.2.1. Fault Management

- \* Supports isolation of hardware and software faults.
- \* Supports monitoring and reporting of fidelity.

#### 11.2.2. Configuration Management

- \* The architecture must support the use of both manual and automated network configuration tools.

## 12. Top Level Architecture

### 12.1. Deterministic Classical Control of Quantum States

This network architecture is entirely classically controlled. Its task is to generate shared quantum states for applications residing at separate nodes. While many quantum events are inherently probabilistic, and loss of photons is also inherently probabilistic, this architecture does not use multipartite quantum states for e.g. routing of two-party requests. (An extension of this network architecture may support generation of multi-partite state for applications at a later date.)

### 13. Communication Service

(Substantial portions of this section are adapted from Naphan Benchasattabuse's Ph.D. thesis, which in turn is adapted from earlier papers by Van Meter et al. [van-meter-qi-arch] and others.)

The design of a quantum network must begin with a clear definition of its fundamental services --- what quantum states or capabilities the network is expected to provide to end users. These decisions determine the complexity of the protocols at the network layer and the applications that run above it. A minimalist design treats `_Bell pairs_` as the primary network-level service. Bell pairs serve as the smallest unit of entanglement and the foundation for nearly all quantum communication protocols. Restricting the service to Bell pair distribution simplifies the network's responsibilities. However, this approach shifts complexity to the applications, which must construct multipartite or fault-tolerant states themselves and manage the coordination overhead that entails.

At the other end of the spectrum, networks may offer richer services such as multipartite entangled states [ghz], [dur-w-state], [hein-multiparty], [hein-graph-entanglement] or fault-tolerant state teleportation. While applications can, in theory, synthesize these states from Bell pairs, direct network-level support may offer efficiency gains and reduce sensitivity to noise by internalizing complex procedures like direct graph state generations or supporting the delivery of error-correcting code encoded logical qubits.

In our architecture, we adopt Bell pair distribution as the core network service, as it allows for a well-scoped, foundational architectural framework.

Importantly, the semantics of Bell pair distribution are not merely those of passive delivery. Even in this basic model, distributed quantum computation occurs along the path via entanglement swapping, possibly combined with purification at intermediate repeater nodes. A proper service definition must account for this processing, as it directly affects fidelity, latency, and trust assumptions in the network.

In addition to quantum state delivery, timing information is often a critical part of the service. Applications in distributed quantum sensing and clock synchronization [degen-sensing], [proctor-quantum-sensing], [proctor-multiparm], [giovannetti-metro], [gottesman-telescope], [ilo-okeke-clock] require precise knowledge of when entanglement was established or when measurement events occurred. Hence, high-precision timestamps may need to be bundled into the service interface offered by the network.

## 14. Architectural Planes

All nodes in the network will have one or more of the following classes of interfaces, also referred to as `_planes_`.

- \* `*Quantum:` The quantum signals and in-channel, hardware-dependent, real-time classical signals for timing and synchronization of photon wave packets. A node with a quantum plane incorporates one or more quantum devices. The quantum devices may be local or remote.
- \* `*Data:` Classical data plane for exchange of messages about in-progress quantum communication sessions. Generally, the data plane consists of `_event notifications_` and `_measurement results_` related to RuleSet-driven connections or testing sessions. A data plane must be accompanied by a control plane.
- \* `*Control:` Classical control plane for establishing and managing connections and testing sessions. Routing and multiplexing messages are included in the control plane.
- \* `*Management:` Network management for configuring the devices and monitoring operation. Managing services provided by nodes, security, addressing, etc., and monitoring health of links, collecting statistics on traffic through the node, any alerts such as security, etc.

### 14.1. Quantum

The quantum signals and in-channel, hardware-dependent, real-time classical signals for timing and synchronization of photon wave packets. A node with a quantum plane incorporates one or more quantum devices. The quantum devices may be local or remote.

The quantum plane functionality executes the functions described as "Interferometric Stabilization" and "Wave Packet Overlap" and subject to the constraints in "Detector Timing Windows" in the Timing Regimes document.

### 14.2. Data

When operating in qubit mode, the Data Plane consists of classical messages that convey events for RuleSet operation and the communication ports and software that generate and consume such messages.

When operating in device mode, the Data Plane consists of RPCs for controlling individual devices.

Data plane functions may share data with management plane functions as part of the link management process, e.g. using disti-mation. If this is done, security and privacy concerns must be addressed.

The control plane functionality executes the functions described as "Pre-configured Event-driven Tasks" in the Timing Regimes document.

#### 14.3. Control

The classical control plane is responsible for establishing and managing connections and testing sessions. Routing and multiplexing messages are included in the control plane.

Control plane functions may share data with management plane functions, e.g. sharing parameter adjustment values and timings. If this is done, security and privacy concerns must be addressed.

The control plane functionality executes the functions described as "Measurement basis selection", "Optical switch control" and some tasks in "Host-side Application-level Tasks" in the Timing Regimes document.

#### 14.4. Management

While connection-specific changes to configuration, such as switching and necessary, immediate changes to e.g. polarization and optical delay may appear as control plane functions, slow-rate monitoring and adjustment of parameters such as timing or polarization due to drift in temperature, voltage or other parameters is the responsibility of the management plane. The management plane may receive useful data on the health and fidelity of links as a result of data plane and control plane operations.

The management plane functionality executes the functions described as "Background Tasks" in the Timing Regimes document.

#### 15. Protocol Layers

This document describes a network architecture. Network designs are often described in terms of a layered protocol stack such as the 7-layer OSI model ([https://en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)), although the Internet can be more accurately described as a three-, four-, or five-layer model ([https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite#Layering\\_evolution\\_and\\_representations\\_in\\_the\\_literature](https://en.wikipedia.org/wiki/Internet_protocol_suite#Layering_evolution_and_representations_in_the_literature)), depending in part on whether a distinction is made between the physical and link layers and in part on how the application layer is subdivided.

In a layered network architecture, each layer processes a prepended header to complete its task. As messages move down the stack (from application toward actual transmission), they may be subdivided into smaller units, and additional layer-specific headers are prepended. On receipt, headers are removed as the message moves up the stack, and boundaries between messages may be altered.

A complete network architecture consists of much more than the layers processing individual packets; many of the critical supporting protocols around naming, routing, security, network management, etc. utilize messages carried using the same protocol stack designed for application data.

In a quantum network, this layering is less clear.

(More to be added here.)

## 16. Nodes and Node Types

(Substantial portions of this section are adapted from Naphan Benchasattabuse's Ph.D. thesis, which in turn is adapted from earlier papers by Van Meter et al. and others.)

The architecture of a quantum network is defined by its constituent nodes and their specialized functions. For clarity in describing our system, we group these nodes according to their primary contributions to network operation. In our architecture, we classify nodes into three main types: end nodes, for application interaction; repeater and router nodes, for extending entanglement and path management; and support nodes, for auxiliary operational tasks.

The qNode specification provides additional details on the common roles and responsibilities of all quantum network nodes, and serves as the equivalent of the Internet hosts requirements RFCs [RFC1122], [RFC1123]. Each node type is further defined in a detailed specification in a separate document.

### 16.1. End Nodes

End nodes represent hosts that wish to execute a quantum application such as quantum key distribution, secret sharing and blind quantum computation [broadbent-bfk-protocol], [fitzsimons-blind]. The technological maturity required of an end node heavily depends on the desired application. There are four major kinds of end nodes:

\*A measurement (MEAS) node\* is the most basic type of quantum end node, designed primarily for protocols that do not require quantum state storage. Its core capability is to receive individual photons

and perform measurements on them in at least two different bases. Lacking quantum memory, MEAS nodes are well-suited for applications like quantum key distribution (QKD) or as simple terminals in certain forms of secure delegated computation protocols [morimae-blind],[fitzsimons-blind], typically interacting with the network in a synchronous manner where measurement results directly yield classical data.

*\*A sensor (SNSR) node\** is a specialized end node designed to utilize entangled states, often shared with distant parties, for high-precision measurements of physical quantities, for clock synchronization tasks, or for distributed sensing tasks [ge-linear],[proctor-quantum-sensing],[degen-sensing],[giovannetti-metro],[proctor-multiparm]. These nodes typically feature limited quantum memory to hold working qubits (e.g., one half of an entangled pair) and possess specific quantum processing capabilities tailored for sensing protocols. Such capabilities include performing joint measurements, like Bell State Measurements (BSMs), between their stored qubits and photons that have interacted with the environment [huang-imaging-stars],[gottesman-telescope]. While an SNSR node's internal processing is specialized, certain sensing applications may also necessitate high-rate entanglement generation from the network to achieve desired performance. For SNSR nodes, precise timing information is almost invariably a critical component of the service they provide or require, and their operation typically culminates in outputting classical data that corresponds to the sensed phenomenon.

*\*A store (STOR) node\** is a specialized end node whose primary function is to serve as a high-fidelity quantum data repository. Its core capabilities are the long-term storage of quantum states---often prepared and teleported from other locations---and the ability to teleport these states out on demand. While a STOR node does not require a universal gate set, it must support certain gates (e.g., Clifford gates) for active quantum error correction to preserve the stored quantum data. This includes using quantum error-correcting codes to protect against decoherence, along with mitigation strategies for correlated errors from events such as cosmic ray strikes [martinis-correlated],[sane-phonons],[xu-dist-qec],[vepsaelaeinen-ionizing],[wu-mitigating],[li-cosmic]. In a network context, STOR nodes may function as data servers, enabling asynchronous applications where valuable states are prepared and stored for later retrieval.

*\*A computational (COMP) node\** represents a full-fledged quantum processing endpoint within the network. Equipped with quantum memory and additional algorithmic qubits, it can store, manipulate, and perform complex computations on quantum states received from the network or generated locally. COMP nodes support a wide range of

advanced quantum network applications, including distributed quantum algorithms, more general forms of blind quantum computation, and potentially fault-tolerant quantum computing, often requiring asynchronous interfaces to coordinate their local quantum workloads with network operations [ambainis-multiparty-coin], [taherkhani-byz], [mayers-unconditional], [christandl-anon], [broadbent-bfk-protocol], [fitzsimons-blind], [mahadev-homomorphic], [dulek-homomorphic], [shapourian-qdc-infra], [sutcliffe-dist-qec], [yoder-tour-de-gross], [kim-ft-million].

## 16.2. Support Nodes

\*An entangled photon pair source (EPPS)\* is a device dedicated to generating pairs of entangled photons, commonly through processes like Spontaneous Parametric Down-Conversion (SPDC). These entangled photons are then typically distributed over quantum channels to be captured or measured at link endpoints, forming the initial resource for entanglement-based protocols. EPPS nodes can be deployed in various scenarios, including terrestrial fiber links or free-space satellite-to-ground communication [fittipaldi-sat], [haldar-sat-dist], [khatri-spooky], [yin-1200km].

\*A Bell state analyzer (BSA)\* is a crucial component for performing projective measurements on two incoming photons, ideally projecting their combined state into one of the four Bell states. BSAs are fundamental for realizing photonic entanglement swapping [zukowski-entanglement-swapping], a primary process that creates link-level entanglement, used particularly to convert memory-photon entanglement into memory-memory entanglement between distant quantum memories. The efficiency and complexity of a BSA depend on the optical implementation and the specific photonic qubit encoding used.

\*A Repeater Graph State Source (RGSS)\* is a specialized source that generates multipartite entangled photonic states, specifically tailored for all-photonic (memory-less) quantum repeater architectures [azuma-rgs], [hilaire-rgs-optimizing-gen-time], [buterakos-graph-generation], [hilaire-logical-bsm]. An RGSS typically distributes segments of the generated repeater graph state to adjacent network nodes, where subsequent measurements on these photonic qubits are performed to establish long-distance entanglement without relying on quantum memories.

\*An advanced Bell state analyzer (ABSA)\* represents a more sophisticated version of a BSA, particularly required in advanced all-photonic repeater protocols based on repeater graph states [azuma-rgs],

[hilaire-rgs-optimizing-gen-time],[buterakos-graph-generation],[hilaire-logical-bsm]. Unlike basic BSAs, an ABSA must be capable of performing measurements on single or multiple photons in dynamically selectable bases. The choice of measurement basis often depends on the outcomes of prior measurements within the network and the specific structure of the repeater graph state being utilized, implying more complex hardware and real-time classical control logic.

*\*An optical switch (OSW)\** is a device that can passively route photons from input optical fibers or paths to different output paths without performing measurements on them [koyama-24]. OSWs, which can be based on technologies like nanomechanical systems or nanophotonic circuits, can be integrated as components within other node types (e.g., routers or complex end nodes) or can function as standalone elements in the network to dynamically reconfigure optical pathways.

### 16.3. Repeater Nodes

*\*A first-generation repeater (REP1)\** is a network node with two quantum interfaces, whose main task is to extend entanglement. Its primary operations include generating link-level Bell pairs with its neighbors, performing entanglement swapping to connect these segments, and managing errors through heralded entanglement purification on physical qubits along the connection path.

*\*A second-generation repeater (REP2)\** also focuses on entanglement swapping to bridge distances but generally requires a larger quantum memory capacity and higher fidelity local quantum operations than a REP1. It utilizes quantum error correction (QEC) to manage operational errors in conjunction with heralded link-level entanglement generation. Instead of, or in addition to, purifying link-level physical Bell pairs, a REP2 node operates on logical qubits, where quantum information is encoded across multiple physical qubits to protect it from errors. This approach inherently demands more sophisticated hardware and advanced computational capabilities for encoding, decoding, and error correction routines during swapping.

*\*A quantum router (RTR)\** is a more complex and versatile node, possessing all capabilities of a quantum repeater and typically featuring three or more quantum network interfaces, enabling it to make sophisticated path selection decisions in complex topologies. Architecturally, an RTR may consist of multiple line cards and a quantum backplane, allowing it to run a full suite of network operation protocols. Beyond basic repeating functions like entanglement swapping, an RTR can govern network borders potentially interfacing between different repeater generations or technologies, participate in generating multipartite entangled states if the

network provides such a service [meignant-dgs] [bugalho-dist-multipartite] [fischer-dgs] [fan-dgs-dist], and may act as a Responder in connection setups by rewriting or generating new RuleSets for different network domains.

#### 16.4. Composite Nodes

A node may also aggregate the functions of more than one node, in a form known as a `_composite node_` or `_composite logical node_`. A common form of composite node is a switching BSA. When multiple devices of the same type are controlled by a single controller, they may be presented either as a node with multiple devices or as a composite node where each device is in turn represented as a node. Presenting as a single node is preferred.

### 17. Links

#### 17.1. The Link Service

A link provides Bell pairs across a single PSD. Each Bell pair is named via an identifier. This service may be either real time or batched.

#### 17.2. Photonic Path Description

The optical path over which photons flow from source to detector in the process of creating a Bell pair can be described using terminology that names the node types in the path; the direction of flow of photons can be inferred. This path description can be applied to point-to-point or switched links within a single PSD.

Device type single-letter abbreviations and their corresponding node type:

- \* M: memory (COMP or STOR)
- \* S: source (of entangled photons) (EPPS)
- \* I: interference (i.e., Bell state measurement) (BSA)
- \* D: detector (i.e. a measurement node) (MEAS)
- \* X: switch (OSW)

Examples of path descriptions that may commonly appear:

- \* DSD: detector-source-detector

- \* MIM: memory-interference-memory
- \* MSM: memory-source-memory
- \* MM: memory-memory
- \* DSISD: detector-source-inteference-source-detector

At any point in a path except the ends, a photon may pass through one or more switches.

In a switched architecture, for example, photons may pass through paths such as:

- \* DXSXD
- \* MXIXM
- \* MXXIXM

(Question: Does this notation also need to represent frequency conversion?)

### 17.3. Point-to-point

Point-to-point links may be either fiber-based or free space. A link encompassing the path of one or more photons may be partially fiber and partially free space.

### 17.4. Switched

A system built around a pool of detectors, particularly organized as BSAs, utilizing switched MIM links can also be characterized as a \_detector-centric architecture\_.

For pseudocode for switching (routing) certain types of devices, see Koyama et al. [koyama-24].

### 17.5. Multidrop or Bus

A multidrop link, or a bus, is a shared physical channel to which more than two stations may be attached.

## 18. Connections

No task involving quantum communication ever involves a single qubit or single entangled state. The connection provides the framework for managing the creation of an order set of entangled states to be consumed by applications. A connection is `_stateful_` at the end nodes. Nodes involved in the creation of end-to-end entanglement for those end nodes will be `_connection aware_`, meaning that they can identify resources and messages and carry out communication tasks necessary for a specific connection, but may not have substantial amounts of state that is dynamically updated on a per-action basis; any actions for nodes in this class must be idempotent or known to occur only once. Some or all nodes may be `_fully stateful_`, tracking the disposition of specific, named quantum states.

Connections may be created using either a fully-distributed protocol [I-D.draft-van-meter-qirg-quantum-connection-setup] or a centralized mechanism. In either case, `qNodes` involved in the connection receive `RuleSets` that are created by a single controller to coordinate local operations to build the end-to-end entangled states requested by an application.

Connections are unaware of the shared use of resources and of other connections. Multiplexing is the responsibility of a separate subsystem, though connection setup should be done with awareness of the availability of unavailability of resources at involved nodes.

## 19. Resource Management: Multiplexing and Routing

Both link usage time slots and memory can be shared among multiple connections and therefore must be actively managed via a multiplexing system. This task is particularly challenging in switched networks.

## 20. Classical Communication

A quantum network depends on classical communication; indeed, almost all of the behavior is governed by, initiated by, or managed and reported via classical messages and signals. These messages and signals have several key roles, described in the following subsections. See the Timing Regimes document [I-D.draft-hajdusek-qirg-timing-physics] for an outline of the physics driving these requirements.

### 20.1. Quantum Plane

Yes, the quantum plane includes some classical signals.

## 20.2. Control Plane

- \* Qubit mode connections and communication
  - connection-level control of switching of photons
  - real-time event notification for RuleSets
- \* Device mode control

## 20.3. Data Plane

- \* Qubit mode connections and communication
  - connection-level control of switching of photons
  - real-time event notification for RuleSets
- \* Device mode control

## 20.4. Management plane: Node and link management

- \* reporting of parameters fixed by device physics
- \* reporting and setting of parameters selectable by node configuration
- \* reporting and adjustment of slowly-changing parameters (such as polarization drift)
- \* typical network node management tasks such as software updates

## 20.5. Mechanisms

- \* RPC for some tasks
  - especially device-mode control
  - qRPC wrapper for classical RPC mechanisms
- \* message broker or event broker for other tasks
  - especially qubit-mode RuleSet notifications

## 21. Naming and Addressing

### 21.1. State naming and management in RuleSets

(Incorporates naming material from Naphan's thesis. Probably needs updating to account for testbed realities.)

Managing and agreeing upon the specific qubits for joint operations like entanglement purification or swapping among collaborating nodes are critical responsibilities handled by RuleSets. While it is possible to draw inspiration from the IP architecture by employing network-wide unique naming for qubits, RuleSets have local execution contexts, making global naming for individual qubits unnecessary.

Instead, RuleSets use local logical names for qubits. At the lowest level, physical qubits within a QNIC can be addressed by the local node using a tuple like `<QNICAddress,QubitIndex>`. However, this level of detail is not, and should not be, visible to the RuleSet logic.

Within a RuleSet instance on a given node, the RuleSet mechanism identifies a quantum resource primarily by its `*tag*` --- a label that categorizes the resource into a specific pool corresponding to its current role or stage in the protocol. To ensure an unambiguous and absolute ordering of multiple resources that may share the same tag (e.g., several Bell pairs awaiting purification), each resource, upon being allocated to a tag, is automatically assigned a `\emph{sequence number}` by the local RuleSet engine. The combination `<tag,seq no.>` thus serves as a distinct and locally unique key within that RuleSet instance for referencing a resource and associating it with relevant metadata, such as its tracked fidelity. Actions specified within Rules typically reference these resources based on their tag and their relative order within that tag, for instance, by operating on the resource with the smallest sequence number (i.e., the oldest available in that pool). This structured internal naming is vital for deterministic local operations and for preventing local operational mismatches that could lead to problems like the leapfrogging issue if not handled carefully at a higher protocol design level.

The RuleSet execution mechanism itself, therefore, only requires and manages this local `<tag,seq no.>` system for resource identification within a single node's RuleSet instance; it does not impose any inherent restrictions or requirements on how shared entangled states are identified between nodes. It is recommended that a strategy be used wherein the RuleSet creator (e.g., the Responder during connection setup) devises RuleSets such that the `<tag,seq no.>` is kept consistent across the RuleSets of all nodes involved in a particular joint operation on that resource. The responsibility thus lies with the RuleSet author to ensure that RuleSets are designed---

potentially using this consistent naming strategy---to correctly manage shared resources and achieve the desired end-to-end protocol behavior. This minimal local naming scheme provided by the RuleSet engine, combined with judicious design of the RuleSets themselves, is sufficient for implementing a wide range of complex quantum network protocols while allowing creativity with minimal restrictions on RuleSet creator parts.

## 22. Example Networks

While a full taxonomy of networks is neither desirable nor possible here, we present a few network examples using point-to-point links or switched architectures. In this section, the topology is briefly described, followed by analysis of the path characteristics of the shortest path and network diameter.

### 22.1. Fully Connected

A number of the early quantum multicomputer proposals assumed a single, large optical switch.

Shortest paths:

### 22.2. Q-Fly Multicomputer

An `_indirect_` interconnect. A multi-group, BSA-centric architecture. All nodes are part of the same PSD. The Q-Fly architecture is described in Sakuma et al. [sakuma-q-fly].

For DPFD topologies:

- \* Shortest paths
  - intra-group: MIXM
  - inter-group (network diameter): MXXIM
- \* Longer paths:
  - non-shortest path: MXXIXM or longer

For DPHD topologies:

- \* Shortest paths
  - intra-group: MXIXM
  - inter-group (network diameter): MXXIXM

- \* Longer paths:
  - non-shortest path: MXXXIXM or longer

For SPHD topologies:

- \* Shortest paths
  - intra-group: MXIXM
  - inter-group (network diameter): MXXIXM
- \* Longer paths:
  - non-shortest path: MXXXIXM or longer

### 22.3. Optically Switched Fat Tree

An `_indirect_` interconnect. Several parameters are needed to describe the full topology of a fat tree, which can also be called a `_k_`-ary `_n_`-tree:

- \* `_k_` is the switch radix
- \* `_n_` is the tree depth

This simplest description assumes homogeneous hardware, where all switches have the same number of ports and all links are the same bandwidth. Leiserson's original fat tree proposed single links of increasing bandwidth at higher levels of the tree, giving the network its name; this approach provides no redundancy or path diversity, and achieving higher transfer rates is impractical in some technologies, including quantum. Consequently, most fat tree deployments use multiple links to several switches at higher levels of the tree, in a configuration that is also known as a `_folded Clos_` network.

### 22.4. Repeater Fat Tree

The repeater fat tree is described in [choi-fat-tree].

### 22.5. 2-D Grid Multicomputer

A `_direct_` interconnect. A 2-D grid of nodes, where nodes with memory and certain computational capabilities (canonically COMP nodes) have up to four interfaces connecting to neighboring nodes. Each node must act as a memory buffer and repeater to enable communication between non-neighboring nodes.

## 22.6. Ring

A `_direct_` interconnect. All nodes in a ring have exactly two neighbors. Each node must act as a memory buffer and repeater to enable communication between non-neighboring nodes.

A ring is described in (something from Simon's group).

## 22.7. QLAN

A quantum local area network will have:

- \* irregular topology, possibly of heterogeneous link types
- \* distributed multiplexing
- \* distributed routing

## 23. APIs for Network Service ("Quantum Sockets")

The API used by classical software to interface with the quantum depends on which class of timing dependency pattern (B, C, or T) is to be supported.

## 24. Security Considerations

Quantum multicomputer systems are assumed to be constructed as isolated, centrally controlled systems with no need for confidentiality, integrity, and availability (the "CIA triad") assurance via cryptographic methods.

Security considerations for other network types are an open topic of study and as such are not yet ready for specification and standardization.

## 25. References

### 25.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

## 25.2. Informative References

## [aboy-governance]

Aboy, M., Gasser, U., Cohen, I., and M. Kop, "Quantum technology governance: A standards-first approach", American Association for the Advancement of Science (AAAS), Science vol. 389, no. 6760, pp. 575-578, DOI 10.1126/science.adw0018, August 2025, <<https://doi.org/10.1126/science.adw0018>>.

## [ambainis-multiparty-coin]

Ambainis, A., Buhrman, H., Dodis, Y., and H. Rohrig, "Multiparty quantum coin flipping", IEEE, Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004. pp. 250-259, DOI 10.1109/ccc.2004.1313848, November 2004, <<https://doi.org/10.1109/ccc.2004.1313848>>.

## [awschalom-roadmap]

, Awschalom, D., Bernien, H., Brown, R., Clerk, A., Chitambar, E., Dibos, A., Dionne, J., Eriksson, M., Fefferman, B., Fuchs, G., Gambetta, J., Goldschmidt, E., Guha, S., Heremans, F., Irwin, K., Jayich, A., Jiang, L., Karsch, J., Kasevich, M., Kolkowitz, S., Kwiat, P., Ladd, T., Lowell, J., Maslov, D., Mason, N., Matsuura, A., McDermott, R., van Meter, R., Miller, A., Orcutt, J., Saffman, M., Schleier-Smith, M., Singh, M., Smith, P., Suchara, M., Toudeh-Fallah, F., Turlington, M., Woods, B., and T. Zhong, "A Roadmap for Quantum Interconnects", Office of Scientific and Technical Information (OSTI), DOI 10.2172/1900586, July 2022, <<https://doi.org/10.2172/1900586>>.

## [azuma-rgs]

Azuma, K., Tamaki, K., and H. Lo, "All-photonic quantum repeaters", Springer Science and Business Media LLC, Nature Communications vol. 6, no. 1, DOI 10.1038/ncomms7787, April 2015, <<https://doi.org/10.1038/ncomms7787>>.

## [bennett-mixed]

Bennett, C., DiVincenzo, D., Smolin, J., and W. Wootters, "Mixed-state entanglement and quantum error correction", American Physical Society (APS), Physical Review A vol. 54, no. 5, pp. 3824-3851, DOI 10.1103/physreva.54.3824, November 1996, <<https://doi.org/10.1103/physreva.54.3824>>.

[broadbent-bfk-protocol]

Broadbent, A., Fitzsimons, J., and E. Kashefi, "Universal Blind Quantum Computation", IEEE, 2009 50th Annual IEEE Symposium on Foundations of Computer Science pp. 517-526, DOI 10.1109/focs.2009.36, October 2009, <<https://doi.org/10.1109/focs.2009.36>>.

[bugalho-dist-multipartite]

Bugalho, L., Coutinho, B., Monteiro, F., and Y. Omar, "Distributing Multipartite Entanglement over Noisy Quantum Networks", Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, Quantum vol. 7, pp. 920, DOI 10.22331/q-2023-02-09-920, February 2023, <<https://doi.org/10.22331/q-2023-02-09-920>>.

[buterakos-graph-generation]

Buterakos, D., Barnes, E., and S. Economou, "Deterministic Generation of All-Photonic Quantum Repeaters from Solid-State Emitters", American Physical Society (APS), Physical Review X vol. 7, no. 4, DOI 10.1103/physrevx.7.041023, October 2017, <<https://doi.org/10.1103/physrevx.7.041023>>.

[choi-fat-tree]

Choi, H., Davis, M., Iesta, ., and D. Englund, "Scalable Quantum Networks: Congestion-Free Hierarchical Entanglement Routing with Error Correction", arXiv, DOI 10.48550/ARXIV.2306.09216, 2023, <<https://doi.org/10.48550/ARXIV.2306.09216>>.

[christandl-anon]

Christandl, M. and S. Wehner, "Quantum Anonymous Transmissions", Springer Berlin Heidelberg, Lecture Notes in Computer Science pp. 217-235, DOI 10.1007/11593447\_12, ISBN ["9783540306849", "9783540322672"], 2005, <[https://doi.org/10.1007/11593447\\_12](https://doi.org/10.1007/11593447_12)>.

[dally-towles]

Dally, W. J. and B. P. Towles, "Principles and Practices of Interconnection Networks", ISBN 978-0-08-049780-8, 2004.

[degen-sensing]

Degen, C., Reinhard, F., and P. Cappellaro, "Quantum sensing", American Physical Society (APS), Reviews of Modern Physics vol. 89, no. 3, DOI 10.1103/revmodphys.89.035002, July 2017, <<https://doi.org/10.1103/revmodphys.89.035002>>.

## [divincenzo-criteria]

DiVincenzo, D. and , "The Physical Implementation of Quantum Computation", arXiv, arXiv, DOI 10.48550/ARXIV.QUANT-PH/0002077, 2000, <<https://doi.org/10.48550/ARXIV.QUANT-PH/0002077>>.

## [drost]

Drost, R., Moore, T., and M. Brodsky, "Switching Networks for Pairwise-Entanglement Distribution", Optica Publishing Group, Journal of Optical Communications and Networking vol. 8, no. 5, pp. 331, DOI 10.1364/jocn.8.000331, April 2016, <<https://doi.org/10.1364/jocn.8.000331>>.

## [dulek-homomorphic]

Dulek, Y., Schaffner, C., and F. Speelman, "", Theory of Computing Exchange, Theory of Computing vol. 14, no. 1, pp. 1-45, DOI 10.4086/toc.2018.v014a007, 2018, <<https://doi.org/10.4086/toc.2018.v014a007>>.

## [dur-w-state]

Dr, W., Vidal, G., and J. Cirac, "Three qubits can be entangled in two inequivalent ways", American Physical Society (APS), Physical Review A vol. 62, no. 6, DOI 10.1103/physreva.62.062314, November 2000, <<https://doi.org/10.1103/physreva.62.062314>>.

## [fan-dgs-dist]

Fan, X., Zhan, C., Gupta, H., and C. Ramakrishnan, "Optimized Distribution of Entanglement Graph States in Quantum Networks", Institute of Electrical and Electronics Engineers (IEEE), IEEE Transactions on Quantum Engineering vol. 6, pp. 1-17, DOI 10.1109/tqe.2025.3552006, 2025, <<https://doi.org/10.1109/tqe.2025.3552006>>.

## [fischer-dgs]

Fischer, A. and D. Towsley, "Distributing Graph States Across Quantum Networks", IEEE, 2021 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 324-333, DOI 10.1109/qce52317.2021.00049, October 2021, <<https://doi.org/10.1109/qce52317.2021.00049>>.

`[fittipaldi-sat]`

Fittipaldi, P., Teramoto, K., Benchasattabuse, N., Hajduek, M., Van Meter, R., and F. Grosshans, "Entanglement Swapping in Orbit: a Satellite Quantum Link Case Study", IEEE, 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 1924-1930, DOI 10.1109/qce60285.2024.00222, September 2024, <<https://doi.org/10.1109/qce60285.2024.00222>>.

`[fitzsimons-blind]`

Fitzsimons, J., "Private quantum computation: an introduction to blind quantum computing and related protocols", Springer Science and Business Media LLC, npj Quantum Information vol. 3, no. 1, DOI 10.1038/s41534-017-0025-3, June 2017, <<https://doi.org/10.1038/s41534-017-0025-3>>.

`[ge-linear]`

Ge, W., Jacobs, K., Eldredge, Z., Gorshkov, A., and M. Foss-Feig, "Distributed Quantum Metrology with Linear Networks and Separable Inputs", American Physical Society (APS), Physical Review Letters vol. 121, no. 4, DOI 10.1103/physrevlett.121.043604, July 2018, <<https://doi.org/10.1103/physrevlett.121.043604>>.

`[ghz]`

Greenberger, D., Horne, M., and A. Zeilinger, "Going Beyond Bell's Theorem", arXiv, arXiv, DOI 10.48550/ARXIV.0712.0921, 2007, <<https://doi.org/10.48550/ARXIV.0712.0921>>.

`[giovannetti-metro]`

Giovannetti, V., Lloyd, S., and L. Maccone, "Advances in quantum metrology", Springer Science and Business Media LLC, Nature Photonics vol. 5, no. 4, pp. 222-229, DOI 10.1038/nphoton.2011.35, March 2011, <<https://doi.org/10.1038/nphoton.2011.35>>.

`[gottesman-telescope]`

Gottesman, D., Jennewein, T., and S. Croke, "Longer-Baseline Telescopes Using Quantum Repeaters", American Physical Society (APS), Physical Review Letters vol. 109, no. 7, DOI 10.1103/physrevlett.109.070503, August 2012, <<https://doi.org/10.1103/physrevlett.109.070503>>.

`[hajdusek-qcomm]`

Hajduek, M. and R. Van Meter, "Quantum Communications", arXiv, DOI 10.48550/ARXIV.2311.02367, 2023, <<https://doi.org/10.48550/ARXIV.2311.02367>>.

`[haldar-sat-dist]`

Haldar, S., Agullo, I., Brady, A., Lamas-Linares, A., Proctor, W., and J. Troupe, "Towards global time distribution via satellite-based sources of entangled photons", American Physical Society (APS), Physical Review A vol. 107, no. 2, DOI 10.1103/physreva.107.022615, February 2023, <<https://doi.org/10.1103/physreva.107.022615>>.

`[hein-graph-entanglement]`

, , , , and , "Entanglement in graph states and its applications", IOS Press, Proceedings of the International School of Physics "Enrico Fermi" vol. 162, pp. 115218, DOI 10.3254/978-1-61499-018-5-115, 2006, <<https://doi.org/10.3254/978-1-61499-018-5-115>>.

`[hein-multiparty]`

Hein, M., Eisert, J., and H. Briegel, "Multiparty entanglement in graph states", American Physical Society (APS), Physical Review A vol. 69, no. 6, DOI 10.1103/physreva.69.062311, June 2004, <<https://doi.org/10.1103/physreva.69.062311>>.

`[hilaire-logical-bsm]`

Hilaire, P., Barnes, E., Economou, S., and F. Grosshans, "Error-correcting entanglement swapping using a practical logical photon encoding", American Physical Society (APS), Physical Review A vol. 104, no. 5, DOI 10.1103/physreva.104.052623, November 2021, <<https://doi.org/10.1103/physreva.104.052623>>.

`[hilaire-rgs-optimizing-gen-time]`

Hilaire, P., Barnes, E., and S. Economou, "Resource requirements for efficient quantum communication using all-photonic graph states generated from a few matter qubits", Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, Quantum vol. 5, pp. 397, DOI 10.22331/q-2021-02-15-397, February 2021, <<https://doi.org/10.22331/q-2021-02-15-397>>.

`[horsman-lattice-surgery]`

Horsman, D., Fowler, A., Devitt, S., and R. Meter, "Surface code quantum computing by lattice surgery", IOP Publishing, New Journal of Physics vol. 14, no. 12, pp. 123011, DOI 10.1088/1367-2630/14/12/123011, December 2012, <<https://doi.org/10.1088/1367-2630/14/12/123011>>.

[huang-imaging-stars]

Huang, Z., Brennen, G., and Y. Ouyang, "Imaging Stars with Quantum Error Correction", American Physical Society (APS), Physical Review Letters vol. 129, no. 21, DOI 10.1103/physrevlett.129.210502, November 2022, <<https://doi.org/10.1103/physrevlett.129.210502>>.

[I-D.draft-hajdusek-qirg-timing-physics]

Hajdusek, M. and R. Van Meter, "Timing Regimes in Quantum Networks and their Physical Underpinnings", Work in Progress, Internet-Draft, draft-hajdusek-qirg-timing-physics-00, 2 March 2026, <<https://datatracker.ietf.org/doc/html/draft-hajdusek-qirg-timing-physics-00>>.

[I-D.draft-van-meter-qirg-quantum-connection-setup]

Van Meter, R. and T. Matsuo, "Connection Setup in a Quantum Network", Work in Progress, Internet-Draft, draft-van-meter-qirg-quantum-connection-setup-01, 11 September 2019, <<https://datatracker.ietf.org/doc/html/draft-van-meter-qirg-quantum-connection-setup-01>>.

[ilo-okeke-clock]

Ilo-Okeke, E., Tessler, L., Dowling, J., and T. Byrnes, "Remote quantum clock synchronization without synchronized clocks", Springer Science and Business Media LLC, npj Quantum Information vol. 4, no. 1, DOI 10.1038/s41534-018-0090-2, August 2018, <<https://doi.org/10.1038/s41534-018-0090-2>>.

[khatri-spooky]

Khatri, S., Brady, A., Desporte, R., Bart, M., and J. Dowling, "Spooky action at a global distance: analysis of space-based entanglement distribution for the quantum internet", Springer Science and Business Media LLC, npj Quantum Information vol. 7, no. 1, DOI 10.1038/s41534-020-00327-5, January 2021, <<https://doi.org/10.1038/s41534-020-00327-5>>.

[kim-ft-million]

Kim, J., Min, D., Cho, J., Jeong, H., Byun, I., Choi, J., Hong, J., and J. Kim, "A Fault-Tolerant Million Qubit-Scale Distributed Quantum Computer", ACM, Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 pp. 1-19, DOI 10.1145/3620665.3640388, April 2024, <<https://doi.org/10.1145/3620665.3640388>>.

[koyama-24]

Koyama, M., Yun, C., Taherkhani, A., Benchasattabuse, N., Sane, B., Hajduek, M., Nagayama, S., and R. Meter, "Optimal Switching Networks for Paired-Egress Bell State Analyzer Pools", IEEE, 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 1897-1907, DOI 10.1109/qce60285.2024.00219, September 2024, <<https://doi.org/10.1109/qce60285.2024.00219>>.

[leone-remote]

Leone, H., Le, T., Srikara, S., and S. Devitt, "Resource overheads and attainable rates for trapped-ion lattice surgery", arXiv, DOI 10.48550/ARXIV.2406.18764, 2024, <<https://doi.org/10.48550/ARXIV.2406.18764>>.

[li-cosmic]

Li, X., Wang, J., Jiang, Y., Xue, G., Cai, X., Zhou, J., Gong, M., Liu, Z., Zheng, S., Ma, D., Chen, M., Sun, W., Yang, S., Yan, F., Jin, Y., Zhao, S., Ding, X., and H. Yu, "Cosmic-ray-induced correlated errors in superconducting qubit array", Springer Science and Business Media LLC, Nature Communications vol. 16, no. 1, DOI 10.1038/s41467-025-59778-z, May 2025, <<https://doi.org/10.1038/s41467-025-59778-z>>.

[litinski-gosc]

Litinski, D., "A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery", Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, Quantum vol. 3, pp. 128, DOI 10.22331/q-2019-03-05-128, March 2019, <<https://doi.org/10.22331/q-2019-03-05-128>>.

[mahadev-homomorphic]

Mahadev, U., "Classical Homomorphic Encryption for Quantum Circuits", Society for Industrial & Applied Mathematics (SIAM), SIAM Journal on Computing vol. 52, no. 6, pp. FOCS18-189-FOCS18-215, DOI 10.1137/18m1231055, December 2020, <<https://doi.org/10.1137/18m1231055>>.

[martinis-correlated]

Martinis, J., "Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays", Springer Science and Business Media LLC, npj Quantum Information vol. 7, no. 1, DOI 10.1038/s41534-021-00431-0, June 2021, <<https://doi.org/10.1038/s41534-021-00431-0>>.

## [mayers-unconditional]

Mayers, D., "Unconditional security in Quantum Cryptography", arXiv, arXiv, DOI 10.48550/ARXIV.QUANT-PH/9802025, 1998,  
<<https://doi.org/10.48550/ARXIV.QUANT-PH/9802025>>.

## [meignant-dgs]

Meignant, C., Markham, D., and F. Grosshans, "Distributing graph states over arbitrary quantum networks", American Physical Society (APS), Physical Review A vol. 100, no. 5, DOI 10.1103/physreva.100.052333, November 2019,  
<<https://doi.org/10.1103/physreva.100.052333>>.

## [mori-psds]

Mori, Y., Sasaki, T., Ikuta, R., Teramoto, K., Ohno, H., Hajduek, M., Van Meter, R., and S. Nagayama, "Scalable Timing Coordination of Bell State Analyzers in Quantum Networks", IEEE, 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 1890-1896, DOI 10.1109/qce60285.2024.00218, September 2024,  
<<https://doi.org/10.1109/qce60285.2024.00218>>.

## [morimae-blind]

Morimae, T. and K. Fujii, "Blind quantum computation protocol in which Alice only makes measurements", American Physical Society (APS), Physical Review A vol. 87, no. 5, DOI 10.1103/physreva.87.050301, May 2013,  
<<https://doi.org/10.1103/physreva.87.050301>>.

## [muralidharan-generations]

Muralidharan, S., Li, L., Kim, J., Ltkenhaus, N., Lukin, M., and L. Jiang, "Optimal architectures for long distance quantum communication", Springer Science and Business Media LLC, Scientific Reports vol. 6, no. 1, DOI 10.1038/srep20463, February 2016,  
<<https://doi.org/10.1038/srep20463>>.

## [nist-singles]

Bienfang, J., Gerrits, T., Kuo, P., Migdall, A., Polyakov, S., and O. Slattery, "Single-photon Sources and Detectors Dictionary", National Institute of Standards and Technology, DOI 10.6028/nist.ir.8486r1, August 2025,  
<<https://doi.org/10.6028/nist.ir.8486r1>>.

## [proctor-multiparm]

Proctor, T., Knott, P., and J. Dunningham, "Multiparameter Estimation in Networked Quantum Sensors", American Physical Society (APS), Physical Review Letters vol. 120, no. 8, DOI 10.1103/physrevlett.120.080501, February 2018, <<https://doi.org/10.1103/physrevlett.120.080501>>.

## [proctor-quantum-sensing]

Proctor, T., Knott, P., and J. Dunningham, "Networked quantum sensing", arXiv, DOI 10.48550/ARXIV.1702.04271, 2017, <<https://doi.org/10.48550/ARXIV.1702.04271>>.

## [ramette-remote]

Ramette, J., Sinclair, J., Breuckmann, N., and V. Vuleti, "Fault-tolerant connection of error-corrected qubits with noisy links", Springer Science and Business Media LLC, npj Quantum Information vol. 10, no. 1, DOI 10.1038/s41534-024-00855-4, June 2024, <<https://doi.org/10.1038/s41534-024-00855-4>>.

## [rdv-thesis]

Van Meter III, R. D., "Architecture of a Quantum Multicomputer Optimized for Shor's Factoring Algorithm", Ph.D. Dissertation, Keio University, 2006, <<https://arxiv.org/abs/quant-ph/0607065>>.

## [res-mgmt-het]

Benchasattabuse, N., "Resource Management in Heterogeneous Quantum Repeater Networks", Ph.D. Dissertation, Keio University, 2025, <<https://aqua.sfc.wide.ad.jp/publications/whit3z-thesis-local-compiled.pdf>>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/rfc/rfc1122>>.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <<https://www.rfc-editor.org/rfc/rfc1123>>.

[RFC9340] Kozlowski, W., Wehner, S., Van Meter, R., Rijsman, B., Cacciapuoti, A. S., Caleffi, M., and S. Nagayama, "Architectural Principles for a Quantum Internet", RFC 9340, DOI 10.17487/RFC9340, March 2023, <<https://www.rfc-editor.org/rfc/rfc9340>>.

[RFC9583] Wang, C., Rahman, A., Li, R., Aelmans, M., and K. Chakraborty, "Application Scenarios for the Quantum Internet", RFC 9583, DOI 10.17487/RFC9583, June 2024, <<https://www.rfc-editor.org/rfc/rfc9583>>.

[sakuma-q-fly]

Sakuma, D., Tsuno, T., Shimizu, H., Kurosawa, Y., Friedrich, M., Teramoto, K., Taherkhani, A., Todd, A., Ueno, Y., Hajduek, M., Ikuta, R., Van Meter, R., Sasaki, T., and S. Nagayama, "Q-Fly: An Optical Interconnect for Modular Quantum Computers", arXiv, DOI 10.48550/ARXIV.2412.09299, 2024, <<https://doi.org/10.48550/ARXIV.2412.09299>>.

[sane-jobs]

Sane, B., Hajduek, M., and R. Van Meter, "Optimizing Resource Allocation in a Distributed Quantum Computing Cloud: A Game-Theoretic Approach", arXiv, DOI 10.48550/ARXIV.2504.18298, 2025, <<https://doi.org/10.48550/ARXIV.2504.18298>>.

[sane-phonons]

Sane, B., Meter, R., and M. Hajdusek, "Fight or Flight: Cosmic Ray-Induced Phonons and the Quantum Surface Code<sup>\*</sup>", IEEE, 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 1378-1388, DOI 10.1109/qce57702.2023.00156, September 2023, <<https://doi.org/10.1109/qce57702.2023.00156>>.

[schoute-shortcuts]

Schoute, E., Mancinska, L., Islam, T., Kerenidis, I., and S. Wehner, "Shortcuts to quantum network routing", arXiv, DOI 10.48550/ARXIV.1610.05238, 2016, <<https://doi.org/10.48550/ARXIV.1610.05238>>.

[shapourian-qdc-infra]

Shapourian, H., Kaur, E., Sewell, T., Zhao, J., Kilzer, M., Kompella, R., and R. Nejabati, "Quantum Data Center Infrastructures: A Scalable Architectural Design Perspective", arXiv, DOI 10.48550/ARXIV.2501.05598, 2025, <<https://doi.org/10.48550/ARXIV.2501.05598>>.

[sinclair-ft-interconnect]

Sinclair, J., Ramette, J., Grinkemeyer, B., Bluvstein, D., Lukin, M., and V. Vuleti, "Fault-tolerant optical interconnects for neutral-atom arrays", arXiv, DOI 10.48550/ARXIV.2408.08955, 2024, <<https://doi.org/10.48550/ARXIV.2408.08955>>.

## [sutcliffe-dist-qec]

Sutcliffe, E., Jonnadula, B., Le Gall, C., Moylett, A., and C. Westoby, "Distributed Quantum Error Correction Based on Hyperbolic Floquet Codes", IEEE, 2025 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 649-657, DOI 10.1109/qce65121.2025.00076, August 2025, <<https://doi.org/10.1109/qce65121.2025.00076>>.

## [taherkhani-byz]

Taherkhani, M., Navi, K., and R. Meter, "Resource-aware system architecture model for implementation of quantum aided Byzantine agreement on quantum repeater networks", IOP Publishing, Quantum Science and Technology vol. 3, no. 1, pp. 014011, DOI 10.1088/2058-9565/aa9bb1, December 2017, <<https://doi.org/10.1088/2058-9565/aa9bb1>>.

## [van-meter-opt-timing]

Van Meter, R., Satoh, T., Nagayama, S., Matsuo, T., and S. Suzuki, "Optimizing Timing of High-Success-Probability Quantum Repeaters", arXiv, DOI 10.48550/ARXIV.1701.04586, 2017, <<https://doi.org/10.48550/ARXIV.1701.04586>>.

## [van-meter-q-net-book]

Van Meter, R., "Quantum Networking", Wiley, DOI 10.1002/9781118648919, ISBN ["9781848215375", "9781118648919"], April 2014, <<https://doi.org/10.1002/9781118648919>>.

## [van-meter-qi-arch]

Van Meter, R., Satoh, R., Benchasattabuse, N., Teramoto, K., Matsuo, T., Hajdusek, M., Satoh, T., Nagayama, S., and S. Suzuki, "A Quantum Internet Architecture", IEEE, 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) pp. 341-352, DOI 10.1109/qce53715.2022.00055, September 2022, <<https://doi.org/10.1109/qce53715.2022.00055>>.

## [vepsaelaeinen-ionizing]

Vepsilinen, A., Karamlou, A., Orrell, J., Dogra, A., Loer, B., Vasconcelos, F., Kim, D., Melville, A., Niedzielski, B., Yoder, J., Gustavsson, S., Formaggio, J., VanDevender, B., and W. Oliver, "Impact of ionizing radiation on superconducting qubit coherence", Springer Science and Business Media LLC, Nature vol. 584, no. 7822, pp. 551-556, DOI 10.1038/s41586-020-2619-8, August 2020, <<https://doi.org/10.1038/s41586-020-2619-8>>.

## [wu-mitigating]

Wu, X., Joshi, Y., Yan, H., Andersson, G., Anferov, A., Conner, C., Karimi, B., King, A., Li, S., Malc, H., Miller, J., Mishra, H., Qiao, H., Ryu, M., Xing, S., Shi, J., and A. Cleland, "Mitigating cosmic-ray-like correlated events with a modular quantum processor", American Physical Society (APS), Physical Review Applied vol. 24, no. 4, DOI 10.1103/4ctq-r6w6, October 2025, <<https://doi.org/10.1103/4ctq-r6w6>>.

## [xu-dist-qec]

Xu, Q., Seif, A., Yan, H., Mannucci, N., Sane, B., Van Meter, R., Cleland, A., and L. Jiang, "Distributed Quantum Error Correction for Chip-Level Catastrophic Errors", American Physical Society (APS), Physical Review Letters vol. 129, no. 24, DOI 10.1103/physrevlett.129.240502, December 2022, <<https://doi.org/10.1103/physrevlett.129.240502>>.

## [yin-1200km]

Yin, J., Cao, Y., Li, Y., Liao, S., Zhang, L., Ren, J., Cai, W., Liu, W., Li, B., Dai, H., Li, G., Lu, Q., Gong, Y., Xu, Y., Li, S., Li, F., Yin, Y., Jiang, Z., Li, M., Jia, J., Ren, G., He, D., Zhou, Y., Zhang, X., Wang, N., Chang, X., Zhu, Z., Liu, N., Chen, Y., Lu, C., Shu, R., Peng, C., Wang, J., and J. Pan, "Satellite-based entanglement distribution over 1200 kilometers", American Association for the Advancement of Science (AAAS), Science vol. 356, no. 6343, pp. 1140-1144, DOI 10.1126/science.aan3211, June 2017, <<https://doi.org/10.1126/science.aan3211>>.

## [yoder-tour-de-gross]

Yoder, T., Schoute, E., Rall, P., Pritchett, E., Gambetta, J., Cross, A., Carroll, M., and M. Beverland, "Tour de gross: A modular quantum computer based on bivariate bicycle codes", arXiv, DOI 10.48550/ARXIV.2506.03094, 2025, <<https://doi.org/10.48550/ARXIV.2506.03094>>.

## [zukowski-entanglement-swapping]

Kaszlikowski, D., Kwek, L., Chen, J., ukowski, M., and C. Oh, "Clauser-Horne inequality for three-state systems", American Physical Society (APS), Physical Review A vol. 65, no. 3, DOI 10.1103/physreva.65.032118, February 2002, <<https://doi.org/10.1103/physreva.65.032118>>.

## Contributors

Michal Hajdusek  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan

Michal was involved in document development and technical discussions from the beginning.

Andrew Todd  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan

Andrew leads the software team implementing much of the network, and provides feedback on system structure.

Monet Tokuyama Friedrich  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan

Contributed presentations and clarity; reviewed this document; authoring related specifications.

Shota Nagayama  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan

Contributed presentations and clarity; reviewed this document and related documents; technical and managerial leadership.

Akihito Soeda  
National Institute for Informatics  
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo  
101-8430  
Japan

Technical and managerial discussions.

Authors' Addresses

Rodney Van Meter  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan  
Email: [rdv@sfc.wide.ad.jp](mailto:rdv@sfc.wide.ad.jp)

Naphan Benchasattabuse  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan  
Email: [whit3z@sfc.wide.ad.jp](mailto:whit3z@sfc.wide.ad.jp)

Amin Taherkhani  
Keio University  
5322 Endo, Fujisawa, Kanagawa  
252-0882  
Japan  
Email: [amin@sfc.wide.ad.jp](mailto:amin@sfc.wide.ad.jp)