第４部

特集4 Quantum Internet

Rodney Van Meter

Abstract

The AQUA (Advancing Quantum Architecture) working group extended the breadth, depth, importance, and reach of its research in 2022. We now divide our activities into four major areas: Quantum Community, Quantum Education, Quantum Computing, and Quantum Internet.

We have significantly improved our simulation capabilities, published one journal paper, and prepared several more for submission, and contributed to the research and development community through leadership of the Quantum Internet Research Group (QIRG) and the Quantum Internet Task Force (QITF). We are particularly proud of the new web browser-based version of our simulator, built using WebAssembly (Wasm).

第１章　Quantum Internet Architecture

The content of this section is adapted from our preprint paper.

The coming Quantum Internet will provide new encryption services, enhance the sensitivity of sensor networks, and couple distant quantum computers to enhance secure computation, share quantum data and increase the size of problems that can be attacked [100, 93, 67, 64]. Hardware components are in rapid development [20]. Numerous architecture and protocol factors have also been investigated, but not yet brought together into a coherent architecture [19, 98, 78, 38, 96, 66]. And yet, our decades of experience with the classical Internet clearly show that architecture and hardware must develop in tandem, and that of the two architecture matures more slowly. Thus, it is imperative to begin laying the foundation for an architecture, driving development of hardware and learning from proposed applications as we go.

It is important to recognize that there will be an internetwork, a network of networks [92]. Without a doubt there will be more than one network architecture; but to build a true Quantum Internet there will ultimately have to be only a single internetwork architecture.

1.1　Quantum Communication is Different

We can summarize quantum communication as follows: *nonlocality is the goal, teleportation is the heart, decoherence is the reality, and the speed of light is still the constraint*.

Quantum entanglement arises from quantum nonlocality, a phenomenon in which distant systems obeying quantum mechanics share a state, allowing them to demonstrate correlations as if they are in direct, seemingly instantaneous communication. Entangled states can be either bipartite or multipartite.

Teleportation is currently the heart of quantum networking [26], as it is the primary method of transferring quantum information encoded in physical quantum states. In quantum teleportation, the state of a quantum variable is destroyed in one place and reconstructed in another. Teleportation from network node *A* to node *B* consumes a special entangled state spanning *A* and *B*, known as a *Bell pair*; hence, the task of a quantum network is to continually produce enough end-to-end entanglement to satisfy applications. Moreover, a form of teleportation known as *entanglement swapping* is used to stretch link-level entanglement into end-to-end entanglement. Other types of quantum networking, e.g., involving superposition but not teleportation, appear to be limited to single-hop configurations and are thus not considered further here.

Unfortunately, quantum data is exceedingly fragile. Photons get lost, so generally speaking we must use acknowledged link layers (though there are exceptions), dramatically affecting throughput. Errors in quantum states caused by noise, imperfect control of memories, etc. are collectively called *decoherence*. One measure of decoherence suffered is *fidelity*, an estimate of the closeness between the actuallyachieved and desired quantum states.

Finally, although entanglement shows nonlocal correlations, it cannot be used to communicate faster than the speed of light. Essentially, all quantum communications require supporting classical communication, which is naturally limited to c. Measurement outcomes on entangled qubits are (anti) correlated and at a first glance may appear to violate special relativity. However, the measurement collapse is random and cannot be controlled, making faster-than-light communication impossible.

All quantum communication relies on a classical communication infrastructure to enable control and coordination. This classical infrastructure is a distinct communication system that operates at the application layer, similar to how some routing protocols run as an application to manage router forwarding tables. This classical network need not share paths or topology with the quantum network it manages, but necessarily interconnects every controllable quantum network component, whether quantum (e.g., teleportation repeater) or classical (e.g., optical switch).

To read this paper, readers need only the notions above, along with the general idea that we are working with *qubits*, quantum binary digits that can be entangled with each other and follow a few simple rules [41]. Qubits can be encoded into photons (using a variety of encoding methods) or stored in stationary memories (implementable in many different physical systems). For a brief summary of quantum information concepts and both popular and technical references, see Appendix A.

1.2 Quantum Communication is Desirable

The unusual characteristics just described would be little more than a curiosity (or a physics experiment) without compelling reasons to integrate quantum communications into our existing IT ecosystem to provide new or better services. We can divide applications into three main, overlapping areas: cryptographic services, sensor networks, and distributed quantum computation [99, 93, 37, 84].

The best-known quantum cryptographic service is *quantum key distribution* (QKD), in which quantum characteristics are used to assess the probability of the presence of an eavesdropper as a stream of shared, random bits is created [＊1]. These random, shared, believed-to-besecret [46, 82, 103] bits can be used in key cryptographic protocols [17, 47, 73]. However, this is not the only cryptographic service that is possible; secret sharing [36, 56, 62, 69], secure election protocols [91], and byzantine agreement protocols [90, 24] are all known.

The second category, sensors, encompasses a range of uses. Arguably, QKD itself is a sensor application, as it physically detects the presence or absence of an eavesdropper. Other uses include enhanced interferometry for telescopes [63, 52] and higher-precision clock synchronization [65, 58], both of which can be viewed as using entanglement as a form of reference frame for time and space [86, 59, 23, 70, 79]. Challenges include determining whether the required precision for classical control of the quantum elements exceeds the gains from the use of entanglement in practice, and the extremely high data rates (entanglement generation rates) required.

The final area is distributed quantum computation [99, 37,

[＊1] Roughly speaking, QKD can be done using single photons [25, 80, 103] or E2E entangled states [27, 45]. Singlephoton demonstration networks have existed since the early 2000s [47], but without the ability to store and manipulate states mid-path, they are single-purpose networks and do not provide E2E security; instead, they depend on classical relays with only hop-by-hop security. Here, we focus on more general, entanglement-based systems.

31, 85], where individual quantum processors are networked together, communicating and sharing their resources to carry out quantum information processing tasks in a coordinated way. Extension of the paradigm of delegated quantum computation leads to applications such as blind quantum computation [30, 49], where a client is able to delegate her computation to a quantum server without revealing information about its input, the computation itself or its output.

1.3 Quantum Repeaters

Quantum repeaters are very different from classical signal repeaters; quantum states cannot be amplified or simply regenerated[*2], and as a general rule cannot be faithfully copied. Instead, the work of the network is to perform a distributed computation that builds the end-to-end entanglement that applications consume. Repeaters and routers serve as waypoints in that E2E problem, and perform four main tasks:

1. Creating base entanglement: Typically using single photons (though there are exceptions to this rule [39]), neighboring repeaters entangle stationary memory qubits. The most common outcome of this process is a *Bell pair*. A number of different link architectures can be used to achieve this task [61].

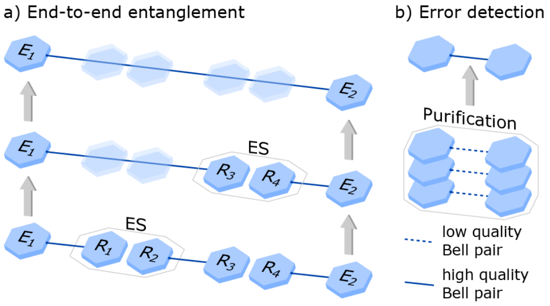


Figure 1: Quantum repeaters build end-to-end distributed entanglement for use by applications at end nodes. In the basic form shown in (a), that process is a distributed computation, depending on *entanglement swapping* (ES) to lengthen entanglement to span multiple hops and a form of error detection, shown in (b), known as *purification,* where multiple low quality Bell pairs can be winnowed down to a single pair of higher quality through a testing protocol that consumes some pairs.

2. Entanglement extension: Achieved via *entanglement swapping* [57] shown in Fig. 1(a), two entangled Bell pairs, $A \leftrightarrow B$ and $B \leftrightarrow C$ can be spliced to form a single $A \leftrightarrow C$ Bell pair. Classical communication is required.

3. Error management: Loss of photons is handled using acknowledged link layers, but state errors and operation (gate) errors must be addressed as well; *purification* is a form of error detection, shown in Fig. 1(b). With enough resources and high enough basic fidelity, quantum error correction can be used.

4. Network operations: Nodes must monitor their own links as well as participate in routing, multiplexing, network operational security, etc. in both networks and internetworks. Our use of this term includes what might be considered both the control and management planes of the quantum network, both of which operate over a classical network that interconnects quantum devices at the classical application layer. This is the focus of this paper.

The most commonly discussed architecture uses purification and entanglement swapping; unless otherwise stated, in this paper we are discussing these first generation, or 1G, networks. Purification requires bilateral confirmation of a qubit measurement result; on even parity, the entangled state is kept and proceeds, while on odd parity the state must be discarded. Entanglement swapping transfers entanglement from one node to another, which requires communicating with two nodes, one of which may be required to adjust its state using information known as a *Pauli frame correction*. Coordination of these operations in a robust but maximally asynchronous fashion is one of the primary tasks of the network protocol.

[*2] Quantum amplifiers [32, 34] are an existing quantum technology capable of boosting certain quantum signals, however quantum states where this is possible have limited use in the context of quantum communication [33].

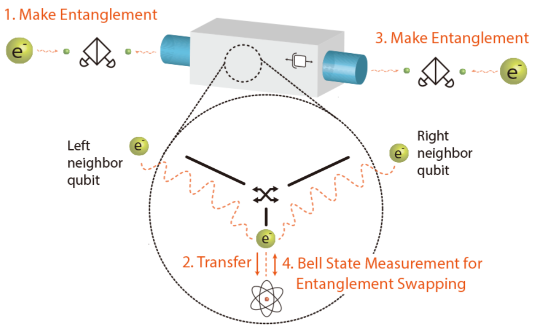


Figure 2: Present-day quantum repeaters [81] represent the absolute minimal form of hardware: a single transceiver qubit (e⁻), a single buffer memory qubit (atom symbol), a two-port optical switch in front, and the ability to initialize, store, manipulate and measure the qubits. This repeater can only attempt to build entanglement to either the left or the right in a given cycle; e.g., after succeeding in making entanglement to the left (Step 1), then the transceiver qubit's state is transferred to to the buffer qubit (Step 2), and entanglement to the right is attempted (Step 3). Once entanglement to the right is achieved, entanglement swapping is performed via a Bell state measurement (joint measurement) of the two qubits (Step 4). This is followed by classical communication with the neighbors (Step 5, not shown).

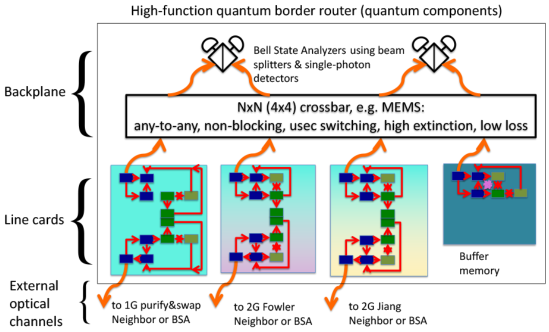

High-function quantum border router (quantum components)
Backplane
Bell State Analyzers using beam splitters & single-photon detectors
NxN (4x4) crossbar, e.g. MEMS:
any-to-any, non-blocking, usec switching, high extinction, low loss
Line cards
Buffer memory
External optical channels
to 1G purify&swap Neighbor or BSA
to 2G Fowler Neighbor or BSA
to 2G Jiang Neighbor or BSA

Figure 3: A full quantum router with hardware architecture similar to today's commercial Internet routers will have QNICs (line cards) coupled via a backplane consisting of optical ports, an optical switch, and Bell State Analyzer measurement devices. Using the BSAs, the qubits in the backplane buffers at the top of the line cards are entangled while the transceiver qubits in the lower portion attempt to create entanglement with neighboring nodes. Once both backplane and neighbor entangled states are made, entanglement swapping is used within each line card to splice the long-distance entanglement. A number of steps in hardware complexity (and cost) will exist between the minimal configuration of Fig. 2 and this one.

1.4 Architecture Decision Points

In developing a Quantum Internet architecture, our goals are similar to those of the classical Internet: we want a system that is robust in operation; easy to implement; and meets requirements such as scalability, security, manageability, and autonomy. Good definitions of interfaces will allow subsystems and hardware implementations to evolve independently and systems will continue to interoperate over time spans of (human) generations. Because we are designing an internetwork, our goal is to create a homogeneous service over heterogeneous subpaths, however, this must be balanced against the fact that early hardware generations will have substantial differences in capabilities.

A number of key design decisions must be made:

1. *The nature of the fundamental service*. Is it Bell pairs, measured-out classical bits, qubit teleportation or multipartite graph states?
2. *The nature of connections*. Is the network 1G, utilizing entanglement swapping and purification? Or is it 2G/3G [77], establishing connections using quantum error correction (QEC)? Alternatively, the connections can be all-photonic, without quantum memories [21, 55].
3. *APIs*. How do applications access the services provided by the network? What is a socket for quantum communication?
4. *Conveying requests*. The protocols for achieving the above services must be designed, including naming conventions for quantum resources.
5. *Stateful connections*. Connections will require both quantum and classical state at each repeater along a path, at least as long as that component is actively participating in building quantum states for the endpoints. What sort of handshake/signalling mechanism is used to establish a connection? Is this centralized or distributed?
6. *Node types*. The state of technology determines the types of nodes we can build; the above items determine the types of nodes required to build a quantum network.
7. *Routing*. How do we pick a path or route through the network?

8. *Multiplexing discipline for resources*. Options for multiplexing the use of quantum resources may include circuit switching, time division muxing, statistical muxing or buffer space muxing. Naturally, stateful connections and many of the muxing candidates require authentication, authorization and accounting.
9. *Security*. Quantum networks allow numerous new attack vectors which have to be considered [88]. These attacks sometimes coincide with the defining property of the service provided by the quantum network, e.g., as in QKD; in other cases, such as for distributed computation, they represent challenges to be overcome.
10. *Making an internetwork*. How should the networks come together to create an internetwork and what is the nature of their interactions?

The above list is by no means exhaustive but covers the critical points. For a more complete list, the interested reader can turn to the QIRG Internet Draft [67].

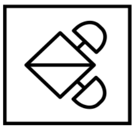
1.5 Quantum Network Components

The previous section discussed individual connections in the abstract; here we show how to operate in complex topologies with complex traffic patterns and actors whose interests aren't always perfectly aligned.

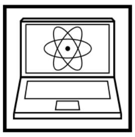
Quantum networks are distinct from their classical counterparts because they cannot exist in isolation; quantum networks incorporate and rely on classical networks to interconnect their components to enable classical control. So despite the name, a quantum network is really a hybrid of a quantum and a classical network.

Just as today's classical Internet consists of Ethernet switches, IP routers of varying capabilities, home routers, WLAN access points, and terminals of various types, nodes comprising the Quantum Internet will come in a variety of flavors. All of the node types below can be implemented in numerous technologies (NV diamond, ion traps, superconducting, quantum dot) [68], using a variety of optical qubit representations (polarization, time bin, spatial path, energy/wavelength, etc.). We divide these into three categories: *end nodes*, *repeater nodes*, and *support nodes*. In addition to definitions of the node types, we propose icons for use in network diagrams.

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



MEAS A node that can only measure received photons (in at least two different ways) and does not store qubits is actually surprisingly useful. A pair of such nodes can conduct quantum key distribution, or a single node of this type can serve as a terminal connecting to a full COMP node in order to execute one form of secure blind quantum computation [76]. However, its error management capabilities are very limited.



COMP Computational end node capable of measuring quantum states as well as storing them in a quantum memory. This greatly enhances the nodes functionality and leads to advanced applications such as blind quantum computation [30, 54]. This node may vary in its processing abilities. Simple clients may be only able to generate, store and manipulate single-qubit states while advanced quantum servers may be able to create large multi-qubit entangled states and hence be capable of universal fault-tolerant quantum computation.

SNSR A sensor node uses the entangled states in a cyber-physical operation, e.g. as a reference frame for interferometry or clock synchronization. For these nodes in particular, recall *that time is part of the service*.

Quantum repeaters are responsible for distribution and management of entanglement across the quantum network. We have three kinds of repeater nodes:

REP1 A 1G repeater. Always has two interfaces; a recent experiment (Fig. 2 and [81]) allows only one to be active at a time, but the generalized form allows both to be active simultaneously. Its primary task is to perform entanglement swapping and error management in the form of purification on physical qubits.

REP2 A 2G repeater. Has the same primary task of entanglement swapping as REP1 but operates at the level of encoded logical qubits composed of multiple physical qubits. Error management is achieved via error correction, signified by the check mark in the REP2 icon. REP2 must be equipped with hardware capable of handling a large number of physical qubits, which necessitates more advanced computational capabilities.

RTR A router. As in Fig. 3, a router likely consists of multiple line cards and a backplane, but for network architectural purposes, the important fact is that a router runs a full suite of protocols governing network operations. Typically, an RTR will have three or more network interfaces, and is capable of governing a network border, where it may be called upon to speak both 1G and 2G protocols and to rewrite RuleSets, behaving as a Responder for connection requests (outbound or transit).

Finally, support nodes are tasked with aiding end and repeater nodes in entanglement distribution. There are five kinds of support nodes:

EPPS An entangled photon pair source, implemented using e.g. symmetric parametric down conversion (SPDC). An EPPS simply produces pairs of entangled photons, which must be captured or measured at link end points. An EPPS can be used in terrestrial links [61] or on a satellite, with the photons captured by telescopes on the ground [104].

BSA Bell State Analyzer, which projects two photons into one of the Bell states; usually used to swap memory-photon and photonmemory entanglement to memorymemory entanglement. The theoretical efficiency limit with linear optics implementation is 50%. The hardware complexity of the BSA depends on the particular qubit encoding.

RGSS Repeater Graph State Source generates entangled multipartite photonic states used in memoryless repeater networks. It sends one half of the generated repeater graph state to its neighboring nodes where the photons are measured.

ABSA Advanced Bell State Analyzer. The basic BSA always performs the same operation, but alloptical repeaters based on repeater graph states require two-photon and single-photon measurements. The measurement basis (type of measurement) is selected dynamically based on prior measurement outcomes as well as the logical encoding and structure of the underlying repeater graph state. This makes the hardware, software and protocol implementations much more complex than a BSA.

OSW Optical switches (nanomechanical or otherwise) can be incorporated into the above node types, but they can also stand alone in the network, switching photons from link to link without measuring them.

This list is by no means exhaustive but covers the main components of a quantum network. The division into end, repeater and support nodes is not mutually exclusive, as there may be some overlap in functionality. For example, the ABSA may be viewed as a type of repeater node as well, as it realizes the task of entanglement swapping. The ABSA requires sophisticated RuleSets and is visible in the connect planning process; the simpler BSA, on the other hand, is tasked only with notifying two nodes about the success of entanglement creation, and need not be visible to nodes farther away in the path.

1.6 Routing

Routing is the process of determining the path of communication between a given set of end nodes. In quantum networks, there are two distinct routes used: one that consists of quantum nodes, and a separate set of classical routes between the control mechanisms of each of those quantum devices.

Picking a route can be achieved with qDijkstra (quantum Dijkstra's algorithm) [98]. The link cost in this case is defined as "seconds per Bell pair of some index fidelity *F*". Fidelity is not an easy metric to obtain in practice, and requires constant link monitoring. An expensive but accurate measure is via *tomography* of the link; lower-cost means of characterizing quantum states is an active area of research [44].

By including fidelity in the link metric, route calculation automatically takes into account the tradeoff between links with high data rate but poor fidelity versus those with low data rate and high fidelity. This approach has yielded good agreement between calculated path cost and throughput obtained via simulation of various paths with heterogeneous links [98].

One of the big open questions that we are investigating is how to combine paths with multiplexing and resource reservation (and starvation), which we take up next.

1.7 Multiplexing and Resource Reservation

Circuit switching, time-division multiplexing, statistical multiplexing (like Internet best-effort forwarding) and buffer space multiplexing are all possible approaches. In buffer space multiplexing, each qubit at each router or repeater node is assigned to one of the specific connections passing through the node, akin to network slicing [22]. Aparicio studied aggregate throughput and fairness for these approaches, and found that statistical multiplexing works pretty well [18, 19]. Statmux allows separate regions of the network to work productively at the same time while sharing the bottleneck link, surpassing circuit switching in terms of aggregate throughput. However, those simulations were for small-scale networks. We believe this topic needs to be studied in much more detail to assess robustness in the face of complex, varying traffic patterns. In particular, we fear that something akin to congestion collapse is possible, or that short-distance connections can starve long-distance connections.

Multiplexing has to coordinate with routing and with AAA, below. Naturally, we want to avoid a fully blocking multiplexing protocol if possible. Any multiplexing scheme that results in extended occupation of resources requires us to determine how those resources are to be allocated, and such a policy will involve identity and likely some form of payment or at minimum debit against some system credit.

1.8 Authentication, Authorization and Accounting

As just noted, it seems likely that performance well below demand will force early implementations to adopt fixed allocation of resources to individual connections. This, in turn, implies that authentication, authorization and accounting (AAA) will become important elements of the architecture [48].

Economics may come to define who has access to the early networks, unless an AAA architecture that explicitly focuses on fairness or some metric other than direct bids for access is put into place.

1.9 Security

Quantum mechanics promises unprecedented levels of confidentiality between communicating parties, which is why quantum key distribution has attracted attention of the theoretical physics and computer science community. However, the focus on QKD also painted a skewed and incomplete picture of security in quantum networks as a whole. This has been slowly changing lately and it has been recognized that while in principle quantum mechanics offers new methods of detecting malicious players in a network, it also enables new vectors of attack [88].

All of the protocols discussed above need authentication and tamper resistance; whether privacy is also required or useful is an open question. Given the previous Internet (and, to a lesser extent, telephone network) experiences with lack of security in routing, accounting, etc., and the likely high cost of quantum connections, it is imperative to have a solid framework in place very early in the Quantum Internet, ideally well before a truly operational network is implemented. This ties into the multiplexing and AAA decisions as outlined above.

第２章　Quantum Internet Simulation Package

In technical developments, our most important progress is coding work on the Quantum Internet Simulation Package (QuISP) [*3].

- Key design and implementation for graph states in repeater networks are under way:

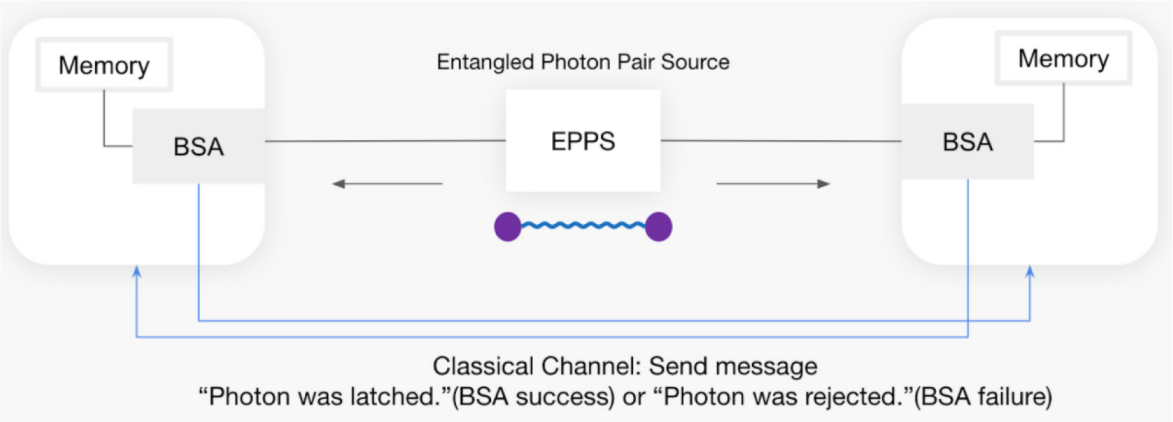

Memory
BSA
Entangled Photon Pair Source
EPPS
Memory
BSA
Classical Channel: Send message
"Photon was latched."(BSA success) or "Photon was rejected."(BSA failure)

Figure 4: Memory-source-memory (MSM) link architecture. This link architecture results in a lower entanglement success probability but orders of magnitude higher attempt rate due to less need to hold a memory and wait for ACK/NAK over the link latency.

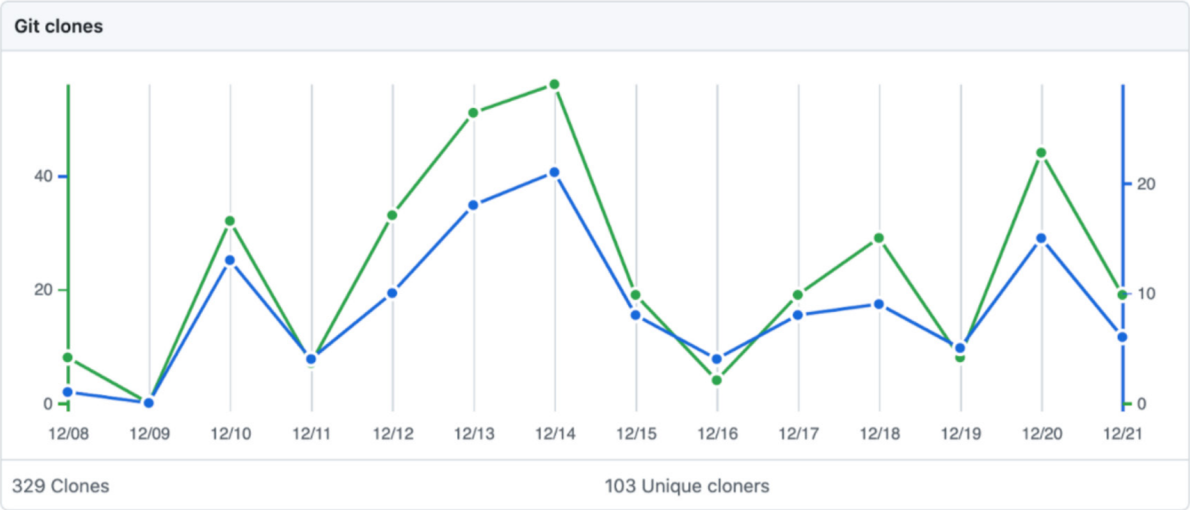
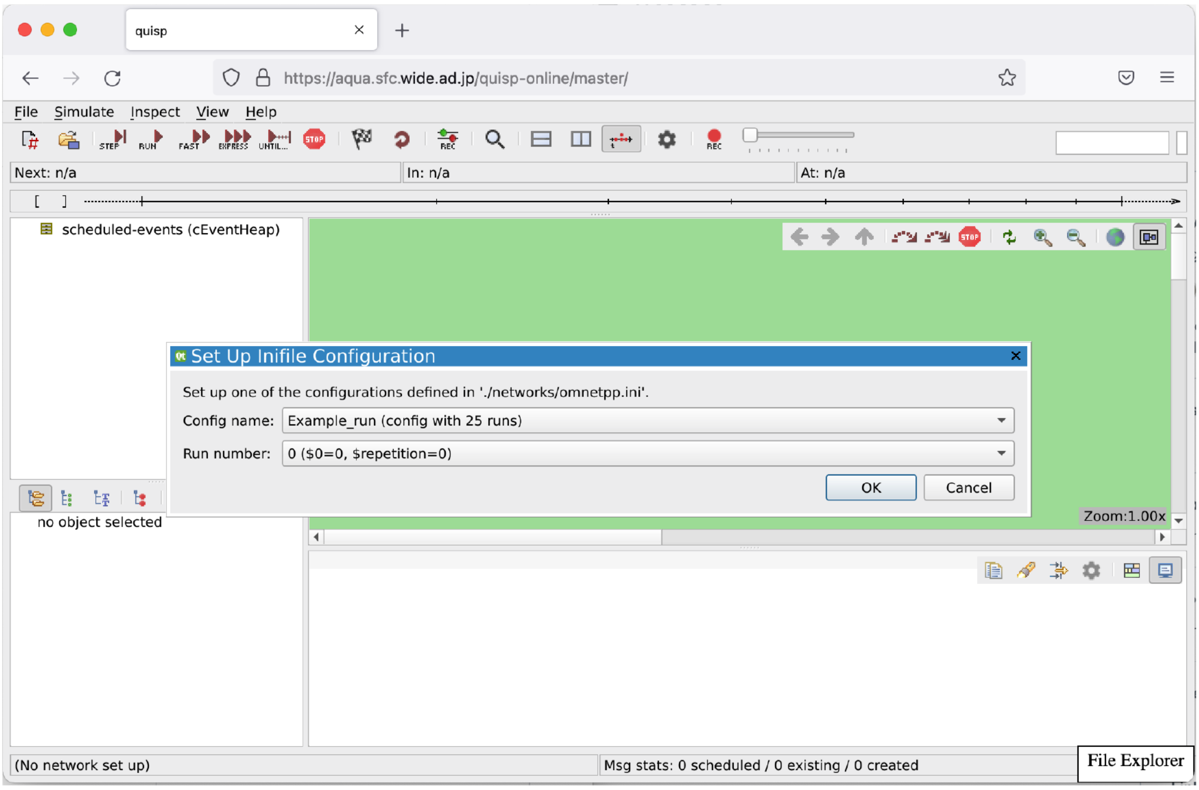


Figure 5: Activity on the QuISP Github repository in mid-December 2021, measured as number of clones per day. Some of this activity is due to automated software testing activated by check-ins to the master branch. The rest is due to potential users cloning the repository and creating their local copies.

[*3] https://github.com/sfc-aqua/quisp.



quisp
https://aqua.sfc.wide.ad.jp/quisp-online/master/
File Simulate Inspect View Help
STEP RUN FAST EXPRESS UNTIL... STOP REC
Next: n/a
In: n/a
At: n/a
scheduled-events (cEventHeap)
no object selected
Zoom:1.00x
File Explorer

Set Up Inifile Configuration

Set up one of the configurations defined in './networks/omnetpp.ini'.

Config name: Example_run (config with 25 runs)

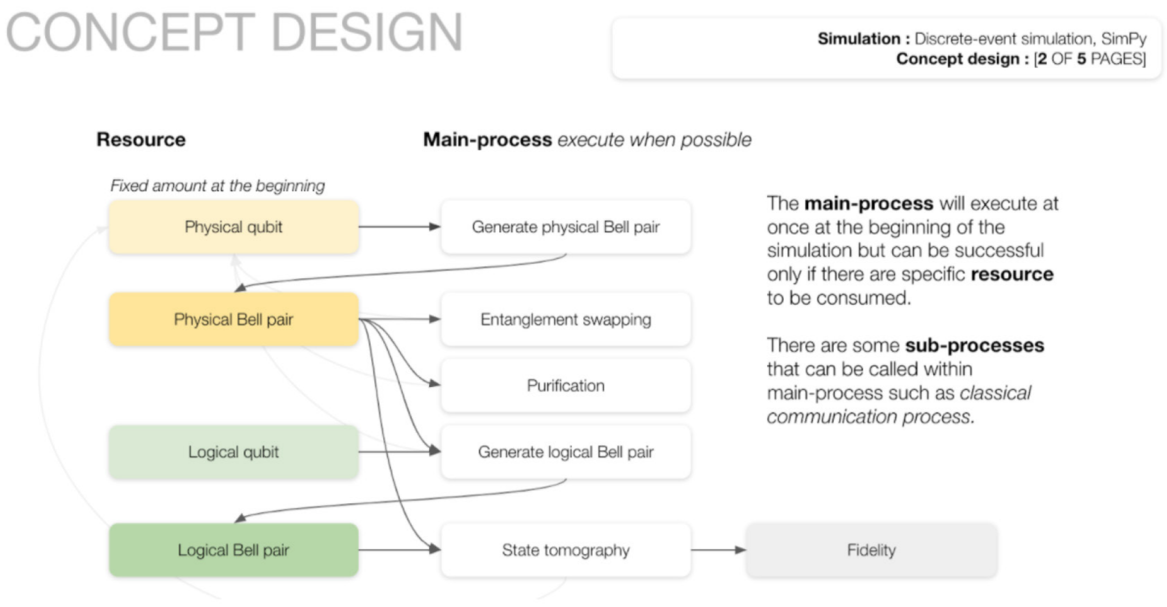
Run number: 0 ($0=0, $repetition=0)

OK Cancel

(No network set up)

Msg stats: 0 scheduled / 0 existing / 0 created

Figure 6: QuISP, complete with the full OMNeT++ GUI, operating in the web browser via Wasm.



CONCEPT DESIGN

Simulation : Discrete-event simulation, SimPy
Concept design : [**2** OF **5** PAGES]

The **main-process** will execute at once at the beginning of the simulation but can be successful only if there are specific **resource** to be consumed.

There are some **sub-processes** that can be called within main-process such as *classical communication process.*

Figure 7: 2G quantum networks will apply quantum error correction to build end-to-end, errorcorrected entangled states. Figure by Poramet Pathumsoot, Mahidol University.

- Simulation of multi-party graph state creation [72] is nearing completion (joint project with Sorbonne University, France). Performed using our open-source Quantum Internet Simulation Package (QuISP).
- Implementation of simulation of RGS (repeater graph state) is under way [55].

- Code for simulation of memory-sourcememory (MSM) (or midpoint source) links is nearing completion and is published as a separate code branch on Github; remaining tasks are testing, integration and documentation. See Fig. 4.

- Flexible traffic generation model for testing behavior under different conditions nearing completion.

- Significant improvements to the QuISP simulator infrastructure:

  - A web browser-based (web assembly, or Wasm) version of the latest simulator is now available [*4]. See Fig. 6.
  - Dramatic improvements in code quality and maintainability.
  - Increase in automated software testing.
  - Improved availability and robustness on various platforms, including improved installation.
  - Reorganization of user and programmer documentation, including a shift to wiki-based documentation.

- Community uptake for QuISP.

In addition to QuISP-based simulation, a non-QuISP simulation of 2G (quantum error correction-based) repeaters is nearing completion. This is a joint project with Mahidol University, Thailand. See Fig. 7.

The following subsections are adapted from our preprint paper [87].

2.1 Quantum Network Architectures

There is currently no consensus on the best overarching network architecture for quantum networking, though the key principles are coming into view [67, 97] and some protocol elements have been proposed [38, 66, 71]. Supporting further research in this area is the primary purpose of QuISP. However, it is becoming clear that some basic hardware and software components will most likely be shared between future candidate architectures. We give a brief outline of these components in this subsection.

Hardware architecture: There exist a number of potential candidate physical systems that are suitable for encoding qubits, broadly divided into two categories. *Stationary qubits* or *matter qubits* are envisioned to store and process information at the nodes of a quantum network, acting as the hosts. Candidate physical systems include nitrogen-vacancy centers in diamond [81], trapped ions [43], atomic ensembles [105] and superconducting qubits [74].

Inter-node quantum communication is achieved by using *flying qubits* encoded onto single photons travelling through optical fibers. We refer to these fibers as *quantum links*. Photons are ideal information carriers as they do not interact strongly with their environment and they travel at very high speeds.

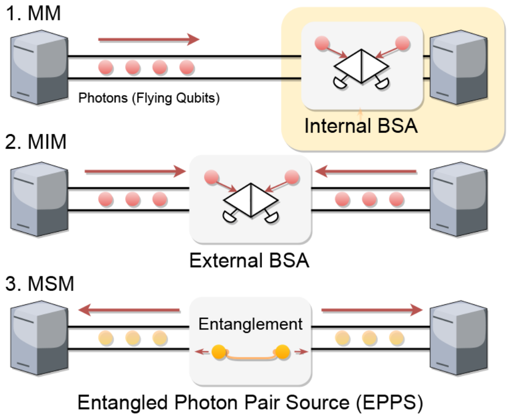


Figure 8: Three quantum link architectures. MM and MIM differ mainly by the position of the BSA while MSM replaces the BSA in the middle with an EPPS.

[*4] https://aqua.sfc.wide.ad.jp/quisp-online/master/.

Using flying qubits, it is possible to distribute entangled Bell pairs between two distant nodes of a quantum network. This can be achieved using one of the three existing quantum link architectures [61] depicted in Fig. 8. *Memory-Memory* (MM) link connects two quantum nodes directly where either node is equipped with a *Bell-state analyzer* (BSA), an optical device that performs a Bell-state measurement on two incoming photons. *Memory-Interfere-Memory* (MIM) link places the BSA in the middle of the quantum link. *Memory-Source-Memory* (MSM) link replaces the the BSA in the middle of the quantum link with a source of entangled photonic pair states (EPPS). Despite the architectures appearing fairly similar, they differ significantly in their performance as well as the technological maturity required to implement them.

Since quantum communication operates at the single-photon level, attenuation becomes a major source of error. Unlike classical bits, qubits cannot be copied and resent owing to the *nocloning theorem* [101], a fundamental property of quantum mechanics forbidding to deterministically copy arbitrary states of quantum systems. Amplification at the level of single photons becomes ineffective as well [34, 33]. This limits the practical length of quantum links to mere tens of kilometers.

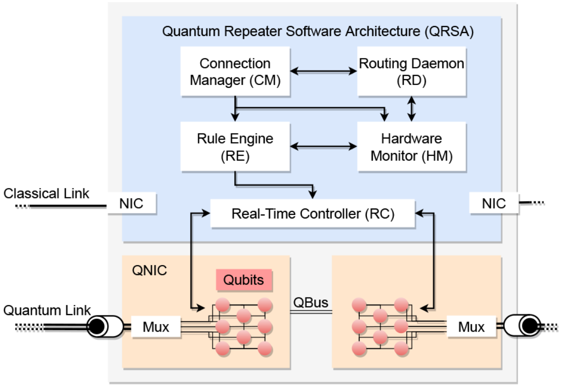


Figure 9: Our target quantum repeater architecture. The top blue section denotes QRSA composed of five distinct software components discussed in the main text. The arrows out of each component represent directions of messages. The bottom orange sections are QNICs which contains multiple stationary qubits to hold quantum information, and manipulate these qubits to extend entanglement via entanglement swapping.

In order to get around this problem, a new type of node was introduce known as a *quantum repeater* [29, 42]. One of the jobs of a quantum repeater is to share Bell pairs with its neighboring nodes and implement entanglement swapping in order to create a Bell pair between these nodes. In this way, the no-cloning theorem can be sidestepped and photon loss mitigated, resulting in the possibility of establishing end-to-end Bell pairs between arbitrarily separated quantum hosts.

A particularly important component of the quantum repeater is the *Quantum Network Interface Card* (QNIC). The QNIC is the quantum analogue of a classical NIC with one major difference being that a QNIC is able to apply quantum operations to the store quantum information, making it a quantum computer with limited capabilities. In particular, the QNIC is capable of applying single-and two-qubit gates as well as single-and two-qubit measurements.

Software architecture Classical software running on a quantum repeater will play a crucial role when designing efficient repeater-based quantum networks. Our proposed software architecture, *Quantum Repeater Software Architecture* (QRSA), as shown in Figure 9, consists of five software components.

- *Connection Manager* (CM): CM manages the connection from the Initiator to the Responder. Once a connection setup request is initiated at an Initiator, it is passed to a Responder through a specific path. At this point, intermediate nodes provide the required information, such as QNIC interface information. The most important task for the Connection Manager is to generate *RuleSets*, which we discuss in Section 2.2.

- *Hardware Monitor* (HM): HM is responsible for monitoring quantum links between the neighboring network nodes. In quantum networking, the quality of links is critical to the final quality of the end-to-end Bell pair. The HM collects information about fidelity and generation rate that is used by RD and CM.

- *Rule Engine* (RE): The main responsibility of the RE is executing RuleSets issued by the CM. To achieve this, the RE constantly monitors the quantum resources available and manages these resources. The results of executed actions are reported back to the RE, and are also distributed to partner nodes where appropriate. RE updates the state of qubits based on incoming messages from itself and other nodes.

- *Real-Time Controller* (RC): RC is in charge of initializing physical qubits and coordinating their photon emissions for the purpose of entanglement distribution. The RC selects which qubits are scheduled to emit photons and at what time. After the qubits no longer take part in entanglement distribution, the RC reinitializes them. RC is device drivers and lower-level software with a hard real time component, interfacing directly to the hardware.

- *Routing Daemon* (RD): RD's responsibility is to create and maintain the routing table for the quantum interfaces. It exchanges information with RDs in neighboring nodes in accordance with a standardized routing protocol. It conveys the information about route and QNIC identifiers required to reach other end node (destination) to other components of the QRSA.

These components communicate as needed to convey when to start operations and the current status of devices.

An end node has almost the same functionality as a repeater, but it also has an Application component responsible for performing end-to-end applications.

2.2 RuleSet Protocol

In order to distribute end-to-end entanglement, both end nodes and quantum repeaters must know what actions to perform, when to execute them, and what other nodes are taking part in the process if the actions need to be coordinated. For example, the repeater must know the nodes that it shares Bell pairs with when executing entanglement swapping since the results of the procedure must be shared with those nodes.

To this end, the *RuleSet* protocol was proposed in [71], which QuISP supports. The goal of this protocol is decentralized, autonomous but coordinated actions of the quantum repeaters with minimal classical inter-node communication. Figure 10 is an example of RuleSet structure. The RuleSet is a collection of *Rules* such purification and entanglement swapping. These

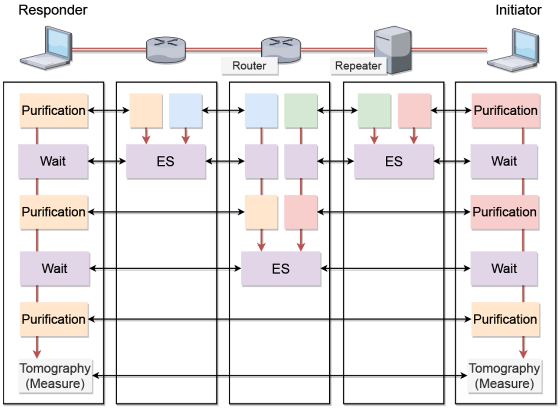


Figure 10: Example of RuleSets. Each node in the path has one RuleSet for the connection. *Rules* are executed from top to bottom while communicating to the proper operation partners. Horizontal arrows represent the partners that are coordinating actions and vertical arrows represent the order of execution. ES is entanglement swapping.

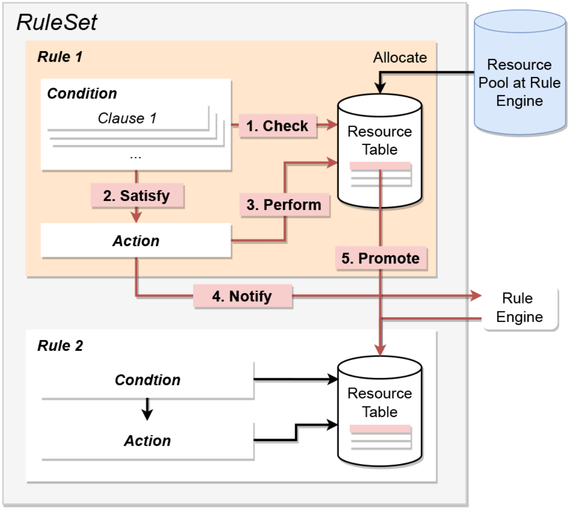


Figure 11: RuleSet execution. 1. Condition clauses checked one by one. 2. If all condition clauses are met, go to step 3, otherwise goes back to step 1 and wait for next allocation of resources. 3. Rules start performing actions. 4. The action notifies the result to the RuleEngine. 5. RuleEngine promotes the resource from one Rule to the next.

RuleSets are built in the connection setup phase discussed in Section 2.3, and executed in a specified order. After being acted upon, the Bell pairs belonging to a particular Rule are passed to the next in sequence.

Figure 11 describes the details of RuleSet and Rule. Every Rule has a *Condition* and corresponding *Action*. The Actions are executed upon satisfaction of local conditions, usually relating to the number and quality of available quantum resources (Bell pairs).

In quantum networking, shared Bell pairs must be managed by each node in a coordinated fashion and appropriately structured Rule-Sets provide this required consistency in terms of quantum operations.

Condition Clauses are composed of single or multiple conditions to be met before the Action is executed. For example, if node A requires two entangled states with node B to perform one action, A must track the number of shared entangled states with B. In such a situation, the Condition used is the *Enough Resource Clause*, which is satisfied when the number of total entangled pairs shared with the proper partner is larger than a threshold (In this case, the threshold is two). Other than Enough Resource Clause, there are several clauses supported in this simulator.

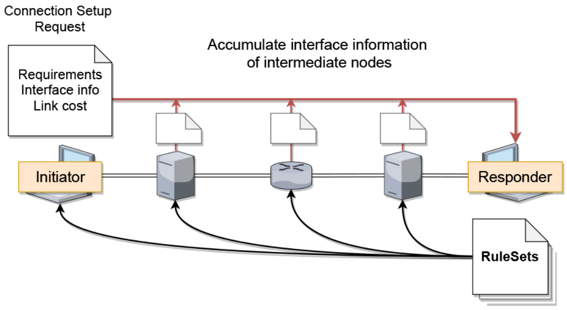
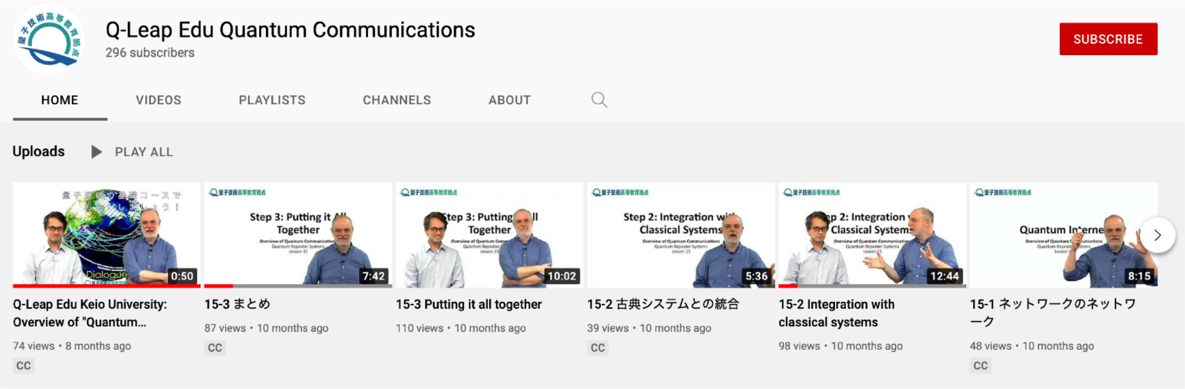


Figure 12: Connection setup process to establish agreements between initiator and responder

An Action Clause is a set of operations including resource assignment changes, qubit manipulation, and classical message transfer. Once Condition Clauses are met, the corresponding action is immediately executed. For example, *Swap* refers to the resource table that belonging to the Rule and recognizes the corresponding qubits. Then, this action chooses one state entangled to its left and one entangled to its right and applies Bell state measurement. Informing the partners of the result of the Bell state measurement is one responsibility of the action.

2.3 Connection Setup

The connection setup is the step requires to gather all the information to create RuleSets to be executed for nodes that will be participating in the end-to-end Bell pair generation. The connection setup process used in QuISP is adapted from protocol outlined by Van Meter and Matsuo [96]. Figure 12 shows the procedure of the connection setup. It involves a two pass process, gathering the link information along the path starting from the node that tries to establish the connection (Initiator) and planning the RuleSet to be distributed among



Q-Leap Edu Quantum Communications
296 subscribers

Uploads

Q-Leap Edu Keio University: Overview of "Quantum...
74 views • 8 months ago

15-3 まとめ
87 views • 10 months ago

15-3 Putting it all together
110 views • 10 months ago

15-2 古典システムとの統合
39 views • 10 months ago

15-2 Integration with classical systems
98 views • 10 months ago

15-1 ネットワークのネットワーク
48 views • 10 months ago

Figure 13: The Q-Leap Edu Quantum Communications YouTube channel. Videos are available in both English and in Japanese.

the nodes along the path at the other half of the connection (Responder).

The first part, at the Initiator node, it receives the requirement for the connection from application level, like the quality of the connection (fidelity of the end-to-end Bell pair) and the number of Bell pairs. In this outbound pass, every node along the path will include their link characteristics into the message, reserve the QNIC, and relay this connection setup message to the next hop. If the QNIC cannot be reserved because it is already in use for another connection, the node will reject the request and the connection setup reject message will be sent to all the previous nodes along the path.

When the connection setup message arrives at the Responder node, the Responder's job is to plan out how each node should execute their share of work in order for the end-to-end Bell pair creation to succeed. After planning out and creating RuleSets for all nodes, the Responder sends *Connection Setup Response* with the Rule-Sets back to all nodes along the path.

第３章　Quantum Internet Task Force

The Quantum Internet Task Force (QITF)[＊5] is continuing its technical work on system architecture (including protocol architecture and design), and is advancing toward an experimental metropolitan area testbed. We continue to recruit members for both financial support and technical expertise.

A Quantum Concepts

There are many good introductions to quantum computing, on the web [51] and in print [89], but for convenience the following is a brief summary of the key aspects of quantum communication and computation that impact network and system architecture.

The primary difference between quantum mechanics and classical probability is that quantum mechanics uses *probability amplitudes*, rather than straight probabilities [16]. Probability amplitudes can be complex numbers; if the amplitude of a given state is α, then the probability of finding that state is $|\alpha|^2$. Most of the concepts below derive fairly directly from this fact and the general wave nature of quantum systems.

Quantum information is most often discussed in terms of *qubits*. A qubit, like a classical bit, is something with two possible values that we can label zero and one. Unlike a classical bit, a qubit can occupy both values simultaneously, known as *superposition*.

To understand quantum computation, we need seven basic concepts:

Superposition. A qubit can represent multiple values in different proportions at the same time, e.g., two-thirds of a "one" and one-third of a "zero". This superposition determines the relative probability of finding each value when we measure the state.

Entanglement (and Bell pairs). Groups of qubits can exhibit strong correlation between the qubits that cannot be explained by independent probabilities for individual qubits. Instead, the group must be considered as a whole, with interdependent probabilities. This phenomenon is known as quantum entanglement. A special entangled state known as a *Bell pair* or *EPR pair*, consisting of two quantum bits, figures prominently in quantum communication. Each qubit in the pair has a 50% probability of having a value of 1 and a 50% probability of having a value of 0 when we measure it. Although we cannot predict which will be found, when we measure one member of the pair, the value of the other is immediately determined. This happens independent of the distance between the two members of the Bell pair.

Interference. Quantum algorithms use some building blocks

[＊5] https://qitf.org/.

derived from classical concepts, such as adder designs, but the overall thrust of a quantum algorithm is very different from that of a classical algorithm. Rather than attempting to solve a problem and checking for the answer, a quantum algorithm's goal is to create *interference* between the elements of a superposition quantum state. Constructive interference reinforces desirable states, increasing the probability of finding a desirable outcome on measurement, while destructive interference reduces the probability.

Unitary, or reversible, gates. Manipulating those probability amplitudes, including creating entanglement and making the interference patterns, involves the use of logical operations known as *gates*. These gates are similar to Boolean logic, but must be reversible, which in mathematical terms means they are represented by a *unitary* transformation matrix.

Measurement. As described above, when we measure a qubit, we get only a single classical bit of information (the "one" or "zero"), and the superposition *collapses*. The probability of finding a zero or a one depends on the probability amplitudes.

Decoherence. Unfortunately, any physical operation (including simply storing a qubit) gradually degrades the state. Decoherence is the single most important technological fact driving quantum computer and quantum network implementations. We can counter this by using a form of error correction or detection.

No cloning. As mentioned above, a key restriction of quantum systems is that we cannot make *independent* copies of an unknown state [102]. This makes error correction difficult. A few additional concepts will augment understanding quantum networks.

Fidelity. The quality of a quantum state is described by its *fidelity*, which is, roughly, the probability that we correctly understand the state – if we ran the same experiment many times and measured the results, how close to our desired statistics would we be? This is one simple measure of the amount of decoherence.

Purification. The form of error detection historically favored in quantum repeater networks is *purification*, which uses minimal resources [28]. It sacrifices some quantum states to test the fidelity of others. There are various purification mechanisms, with different purification algorithms and different methods for determining which states are sacrificed, each with particular tradeoffs.

Quantum error correction (QEC). QEC may be based on classical codes or purely quantum concepts. The primary difficulties are extraction of errors without damaging quantum state, avoiding error propagation, and the increased resources required. (See references contained in [92], [60] and [50].)

Teleportation. Teleportation destroys the state of a qubit at the sender and recreates that state at the destination, teleporting information rather than matter [26]. The process uses a Bell pair's long-distance correlation, followed by transmission of a pair of classical bits. Teleportation consumes a Bell pair.

Entanglement swapping. Splicing two longdistance Bell pairs together to make one longer Bell pair is known as entanglement swapping.

With these basic concepts, we can begin to construct networks.

For those interested in a more researchoriented, in-depth survey of quantum computing systems, we recommend the following short list of papers: [41, 35, 40, 53, 68, 75, 83, 95, 94]. For communication, we recommend: [100, 67, 99, 20, 28, 64].