

A Quantum Internet Architecture

Rodney Van Meter
Keio University

Ryosuke Satoh
Keio University

Naphan Benchasattabuse
Keio University

Takaaki Matsuo
WIDE Project

Michal Hajdušek
Keio University

Takahiko Satoh
Keio University

Shota Nagayama
Mercari, Inc.

Shigeya Suzuki
Keio University

Abstract

Entangled quantum communication is advancing rapidly, with laboratory and metropolitan testbeds under development, but to date there is no unifying Quantum Internet architecture. We propose a Quantum Internet architecture centered around the Quantum Recursive Network Architecture (QRNA), using RuleSet-based connections established using a two-pass connection setup. Scalability and internetworking (for both technological and administrative boundaries) are achieved using recursion in naming and connection control. In the near term, this architecture will support end-to-end, two-party entanglement on minimal hardware, and it will extend smoothly to multi-party entanglement and the use of quantum error correction on advanced hardware in the future. For a network internal gateway protocol, we recommend (but do not require) qDijkstra with seconds per Bell pair as link cost for routing; the external gateway protocol is designed to build recursively. The strength of our architecture is shown by assessing extensibility and demonstrating how robust protocol operation can be confirmed using the RuleSet paradigm.

1 Introduction

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 [50, 53, 81, 89]. Hardware components are in rapid development [5]. Numerous architecture and protocol factors have also been investigated, but not yet brought together into a coherent architecture [4, 22, 52, 65, 85, 86]. 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 [80]. 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: *non-locality 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 [11], 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 actually-achieved 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 [25]. 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 [21, 71, 81, 88].

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 created¹. These random, shared, believed-to-be-secret [29, 69, 91] bits can be used in key cryptographic protocols [2, 30, 60]. However, this is not the only cryptographic service that is possible; secret sharing [20, 41, 48, 55], secure election protocols [78], and byzantine agreement protocols [9, 77] 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

¹Roughly speaking, QKD can be done using single photons [10, 67, 91] or E2E entangled states [12, 28]. Single-photon demonstration networks have existed since the early 2000s [30], 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.

uses include enhanced interferometry for telescopes [35, 49] and higher-precision clock synchronization [44, 51], both of which can be viewed as using entanglement as a form of reference frame for time and space [8, 45, 56, 66, 73]. 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 [15, 21, 72, 88], 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 [14, 32], 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 regenerated², 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 [23]), 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 [47].
2. **Entanglement extension:** Achieved via *entanglement swapping* [43] 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

²Quantum amplifiers [16, 18] 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 [17].

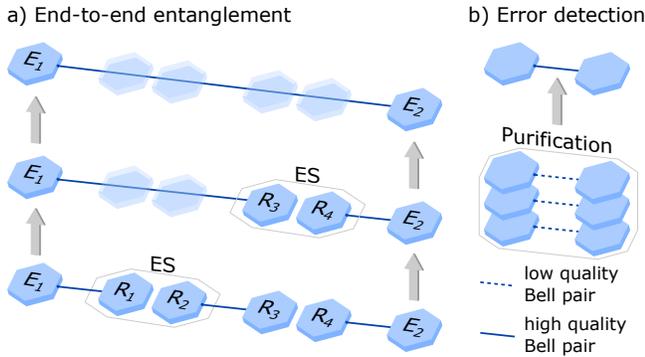


Figure 1: Quantum repeaters build end-to-end distributed entanglement for use by applications at end nodes. In the basic form shown in (a), that process is a distributed computation, depending on *entanglement swapping* (ES) to lengthen entanglement to span multiple hops and a form of error detection, 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.

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.

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 internet network, 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:

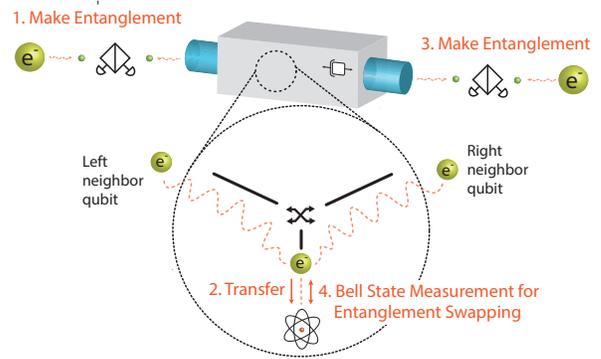


Figure 2: Present-day quantum repeaters [68] represent the absolute minimal form of hardware: a single transceiver qubit (e^-), a single buffer memory qubit (atom symbol \otimes), a two-port optical switch in front, and the ability to initialize, store, manipulate and measure the qubits. This repeater 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).

1. *The nature of the fundamental service.* Is it Bell pairs, measured-out classical bits, qubit teleportation or multipartite graph states? (Sec. 2)
2. *The nature of connections.* Is the network 1G, utilizing entanglement swapping and purification? Or is it 2G/3G [63], establishing connections using quantum error correction (QEC)? Alternatively, the connections can be all-photonc, without quantum memories [6, 40]. (Sec. 2.1)
3. *APIs.* How do applications access the services provided by the network? What is a socket for quantum communication? (Sec. 2.2)
4. *Conveying requests.* The protocols for achieving the above services must be designed, including naming conventions for quantum resources. (Sec. 3.)
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

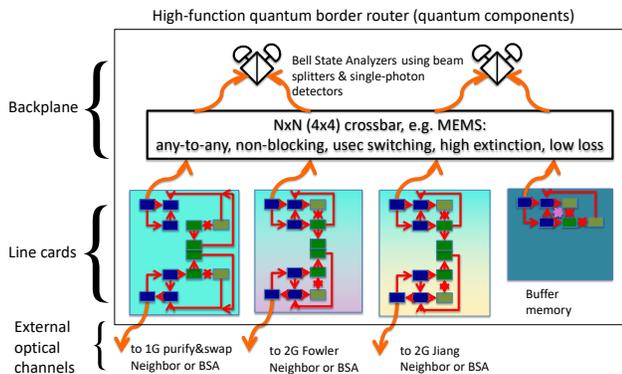


Figure 3: A full quantum router with hardware architecture similar to today’s commercial Internet routers will have QNICs (line cards) coupled via a backplane consisting of optical ports, an optical switch, and Bell State Analyzer measurement devices. Using the BSAs, the qubits in the backplane buffers 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.

connection? Is this centralized or distributed? (Secs. 3.5, 4.3 and 5)

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. (Sec. 4.1)
7. *Routing.* How do we pick a path or route through the network? (Sec. 4.2)
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. (Secs. 4.3, 4.4)
9. *Security.* Quantum networks allow numerous new attack vectors which have to be considered [75]. 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. (Sec. 4.5)
10. *Making an internetwork.* How should the networks come together to create an internetwork and what is the nature of their interactions? (Sec. 5)

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 [53].

After proposing answers to these questions in the next several sections, we provide some evidence for the correctness of our choices (Sec. 6) before concluding (Sec. 7).

2 Quantum Network Services

2.1 Semantics: Bell Pairs and On-Path Distributed Computation

Entanglement is the resource that will fuel quantum applications such as QKD, teleportation, quantum sensing, or delegated quantum computation. Continuous, reliable and efficient replenishment of this resource is one of the primary tasks of a quantum network. However, entangled states come in many shapes and sizes [36, 42].

Bell pairs are the most basic bipartite states and form the fundamental building blocks of entangled quantum networks. They can be generated by a network link equipped with stationary quantum memories at each end that are entangled via flying photons. Due to the Bell pairs’ importance to virtually all quantum communication protocols it is generally agreed that they will be part of the fundamental network service.

End-to-end multipartite states such as GHZ, W and graph states [39] are resources for a variety of multiparty protocols, and therefore are likely to be extremely valuable. Bell pairs alone would be sufficient; multipartite states can be built using them, but because the efficiency will matter, it is an open question whether multipartite states are part of the fundamental service or should be created and managed entirely by applications running at end nodes.

For this reason, we focus on distribution of Bell pairs in this manuscript. This distribution can be achieved in a number of ways and depending on the nature of the connection networks are classified into three generations [63]. 1G quantum networks build E2E entangled pairs using physical Bell pairs, spliced and error tested using entanglement swapping and purification. Such connections are the most basic way of establishing E2E entanglement and therefore the first implementations of quantum networks are expected to be 1G. 2G and 3G networks are designed to reach and maintain higher fidelities (especially useful for distributed computation) by utilizing quantum error correction, placing very demanding requirements on the hardware. These error management schemes require work at each repeater, but with the goal of E2E error management [64].

All of the above generations of quantum networks rely on quantum memories to store the qubits while the networks entangle them using flying photons. It is prudent to mention that all-photon quantum repeaters have been proposed that do not require quantum memories [6, 40].

In Sec. 1.3, we noted that this distribution of Bell pairs is, in fact, a limited form of distributed computation all along the path. The semantics of the network service must be defined to take this into account.

Finally, it is important to note that *time is part of the service* [53]. For sensor applications in particular, very high precision timestamps of some events are necessary information, and must be provided to applications.

2.2 Application Access: Quantum Sockets

Once entangled qubits are ready, an application consumes them. As mentioned earlier, there are three types of currently envisioned applications. It is possible to categorize the three into two types: those that use qubits in larger quantum applications and those that measure the qubits to produce classical information right away. Applications that consume qubits directly will measure the result immediately after the execution of the application; thus, eventually, both cases have a classical result.

We are designing the Quantum Socket API (API from now on) in an analogy with the classical socket API widely used in the classical Internet. The API is very similar to the classical socket API. Like the classical socket API, the API has functionalities such as: creating a socket, connecting to the socket endpoint, reading, writing, setting options, and destroying the socket. The API is node-type agnostic; i.e., it can handle three end-node types (MEAS, COMP, SNSR described in Sec. 4.1) corresponding to three different application classes.

Operations on nodes of the type that return classical information (MEAS, SNSR) as described earlier are synchronous since the result will eventually become classical reads (read system calls, etc.). For these applications, the stochastic arrival time of completed Bell pairs is not a problem. In contrast, COMP nodes involve substantial coordination with other work at a quantum computer. How to build distributed quantum programs that deal robustly with stochastic entanglement delivery, e.g. via asynchronous callbacks, is an open problem.

Both applications or controlling programs can configure specific parameters for each physical interface, such as the RuleSet specific to the QNIC, via the ioctl-like interface. In other words, classical components communicate with physical quantum components via the socket API.

3 Expressing Connection Semantics: RuleSets

Having just established that the core service of a quantum internet is building E2E entanglement, now we need an internetwork protocol capable of communicating the actions necessary to span different connection architectures (1G, 2G). Here, we describe a mechanism efficient enough for use as the basis of a network protocol, and rich and abstract enough for use as an internetwork protocol (Sec. 5).

1G networks need a mechanism for conveying requests such as, “Bob, once you get a Bell pair with Alice and a Bell pair with Charlie, execute entanglement swapping, then send the Pauli frame correction to Charlie and a notice-of-entanglement-transfer to Alice,” and “If you have two Bell pairs with Alice, both with fidelity less than 0.9, then purify.” 2G networks will work on *logical* qubits encoded using quantum error correction, making for complex operations for entanglement swapping and error correction while presenting high-fidelity logical qubits to applications.

Our approach is to define Rules that have a *condition clause* and an *action clause*, very analogous to the OpenFlow extensions of classical software defined networking (SDN) [59]. For a connection, each node is given a RuleSet that should comprehensively define what to do as local events occur (entanglement success, timeout, etc.) and as messages arrive [57, 58]. This RuleSet-based operation is the heart of our work, and allows for explicit reasoning about how to achieve the maximum asynchrony and autonomy in the network (rather than waiting for explicit instructions at every operation or attempting to make everything proceed in lockstep). Our version of RuleSets includes local state, which cannot be expressed in SDN OpenFlow.

There is one RuleSet for each connection. Once a resource (e.g., link level Bell pair) is assigned to a RuleSet, that assignment does not change. How that assignment is done is the responsibility of the multiplexing scheme (Sec. 4.3).

RuleSets and any qubits at the nodes that are currently assigned to a particular connection are *connection state* that must be held at each repeater/router. The scalability of this needs to be assessed, and it affects AAA (Sec. 4.5), but we currently see no approach to quantum networking that allows mid-path routers and repeaters to be fully stateless.

We have adapted the RuleSet approach from [57, 58] and incorporated an additional construct called a *Stage*. A RuleSet can be thought of as a program that oversees the processing of the states. Entangled states are allocated to a given Stage. In each Stage, there can be multiple Rules. Each Stage can also have its own variables which are shared by Rules in the same Stage. After one of the Rules in the Stage fires, the entangled state is either promoted to the next Stage or declared defunct and the physical resources are returned to the pool available for reuse. This ensures that the flow of qubits is unidirectional and terminates with being either delivered to an application or service, or discarded (either consumed as part of the protocol operation or determined to likely be in error).

3.1 Naming States (Qubits)

Managing the qubits and agreeing on the consumption of entangled states among shared nodes are two of the most critical tasks for RuleSets. In the IP architecture, network addresses are associated with a network interface; here, we assume the same. Thus, a physical qubit can be uniquely identified

by its network address and the index of the qubit within the QNIC, using the tuple $\langle \text{QNICAddress}, \text{QubitIndex} \rangle$. QNIC firmware applies quantum operations based on that index, and the tuple is unique within the scope of the network address.

However, rather than this physical address, we are usually interested in the state (e.g., half of a Bell pair) that is held in the qubit, which is dynamic and has a finite lifetime; the distinction is philosophically similar to a register versus a temporary variable. Therefore, when nodes that share entangled resources want to communicate changes to other parties, they use another (external) name which is only known by the shared parties. This name needs to be unique. Initially, the name is determined by one of the nodes involved in the creation of the link-level Bell pair (e.g., the BSA node described in the next section). The name might be, for example, the tuple $\langle \text{NodeAddress}, \text{Timestamp} \rangle$, where the timestamp is of high enough precision that at most one Bell pair may have been created. The mapping of that external name to internal qubit address is maintained independently and privately by each node.

When entanglement swapping is completed, a new name for the Bell pair is created by the node performing the swapping. That name is communicated to the two end points as part of the notification of the entanglement being transferred to new partners.

3.2 Messages

Tab. 1 lists the primary messages included in our protocol. (Naturally, each message transmission is initiated by an Action clause, and its reception matches a Condition clause; in the interest of space these are not included in Tabs. 2 and 3.) Purification involves testing the parity of two qubits at each end and exchanging the results using a measurement outcome (MEAS) message. Each end compares the parity it calculates to the parity it receives, and either discards both Bell pairs (on mismatch) or raises the software’s estimated fidelity of one and discards the other (on match). Entanglement swapping requires that both ends be notified of the transfer of entanglement to new partners, and one end must also receive a Pauli frame update.

3.3 Condition Clauses

Tab. 2 shows the Condition Clauses that can be defined in Rules. The Condition Clauses can be thought of as defining the trigger for moving from one state to another in a state machine, while the Action Clause for the Rule defines the side effects.

Sometimes, a Condition Clause needs to match only one entangled state, for example when matching a Bell pair and deciding to deliver it to an application (in which case it passes out of our ken). More often, it needs to match two: either with

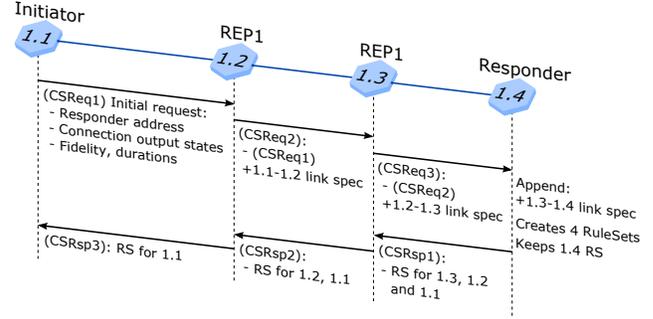


Figure 4: Two-pass connection setup (CS) within a single quantum network. RuleSets are created by the Responder, offering a distributed innovation point.

the same end points, for purification, or with different end points, for entanglement swapping.

3.4 Action Clauses

Tab. 3 shows the Action Clauses that can be defined in Rules. The Action Clauses can be thought of as defining sequences of local quantum operations and messages to be sent. The actions are chosen from a restricted set of options and do not include loop primitives; despite the existence of QCIRC which applies a quantum circuit, this is not a Turing complete computation platform. Conditional execution is done by creating separate rules with distinct Condition clauses. These restrictions make it easier to reason about distributed protocol actions in terms of termination, robustness, deadlock, security, and other issues.

As noted above, message generation is not included in this table but is a natural consequence of QCIRC, MEAS and some of the local software actions.

3.5 Two-Pass Connection Setup

Our approach to connection setup uses two passes, as proposed by Van Meter and Matsuo [85]. On the outbound leg (starting at the *Initiator*), information about links and available resources is collected. The connection request eventually reaches the *Responder*, which takes that information and builds RuleSets for every node along the path. Those RuleSets are distributed in a return pass, then the operation for the connection begins.

Setup within a single network is illustrated in Fig. 4. As in the classical Internet, we expect that the majority of connections will be initiated by a client node reaching out to a server. This architecture places the server in charge of RuleSet creation, allowing service providers a single point of innovation; if they create better RuleSets than their competitors, then connections will be faster or more robust, providing a competitive advantage.

Table 1: **Protocol Messages**

Name	Descriptive Name	Arguments	Comments
Remote Events (Message Transmission)			
FREE	Release a state	Partner addr., resource IDs	Release a state back to the free pool. Used after purification.
UPDATE	State change notification	Partner addr., resource IDs, Pauli frame correction	Used to indicate a Pauli frame correction to a state. Most commonly used with TRANSFER to complete entanglement swapping.
MEAS	Measurement outcome	Partner addr., resource IDs, result	Exchange purification results. Each partner sends this message, and a separate rule will recognize whether purification results agree and proceed appropriately. Numerous types are possible.
TRANSFER	Entanglement transfer notification	Partner addr., resource IDs	Distribute the result of a swapping circuit. Generalizes to a notice of entanglement transfer from one location or partner to another. Carries a new resource ID to used for the resulting state.

Table 2: **Condition Clauses**

Name	Descriptive Name	Arguments	Comments
Local Software Events			
CMP	Check whether a variable is equal, less than, or greater than some values	variable ID, comparison operator, value	Used to track number of operations done (e.g. purification count, measurement count, or number of notification message received)
TIMER	Timer expiration	Timer ID	Must be used with caution when dealing with distributed states; race conditions can occur.
Quantum State Events (Local Hardware Notifications, Message Reception)			
RES	Enough Resources	Partner address (or wildcard) and fidelity	Matches Bell pairs. Used commonly for purification and entanglement swapping. Used to check fidelity of Bell pairs, this also serves as the primary “meets application requirements” clause for delivering to apps at EndNodes.

4 Networking

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.

4.1 Quantum Network Components

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) [54], 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*.

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:

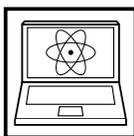
Table 3: **Action Clauses**

Name	Descriptive Name	Arguments	Comments
Local Software Actions (Classical)			
SETTIMER	Set timer	Timer ID	Use with caution; distributed race conditions can occur.
PROMOTE	Promotion of qubits	Qubit IDs, Rule ID, Stage	Used to transfer the control/ownership of Qubits from current Rule (Stage) to Another Rule (Stage)
FREE	Free qubits	Qubit IDs	Release qubits to the pool of unallocated resources.
SET	change value of a Rule/RuleSet variable	variable identifier	Can be used to track how many measurements have occurred for tomography.
Local Hardware Actions (Quantum)			
MEAS	Measure qubits	Qubit IDs, meas. basis	Measure one or more qubits in specified basis or a randomly chosen one.
QCIRC	Apply quantum circuit	Qubit identifiers, Qcircuit	Apply a general unitary quantum operation on one or more qubits, without measuring. Bell state measurement, purification, and entanglement swapping execute QCIRC first, then MEAS. Encoding into logical qubits also uses.

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 [62]. 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 [14, 38]. 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.

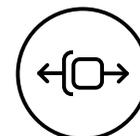


SNSR A sensor node uses the entangled states in a cyber-physical operation, e.g. as a reference frame for interferometry or clock synchronization. For these nodes in particular, recall that *time is part of the service*.



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

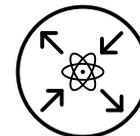
REP1 A 1G repeater. Always has two interfaces; a recent experiment (Fig. 2 and [68]) allows only one to be active at a time, but the generalized form allows both to be active simultaneously. Its primary task is to perform entanglement swapping and error management in the form of purification on physical 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.

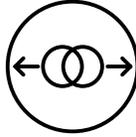


RTR A router. As in Fig. 3, a router likely consists of multiple line cards and a backplane, but for network architectural purposes, the important fact is that a router runs a full suite of protocols governing network operations. Typically, an RTR will have three or more network interfaces, and is capable 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:

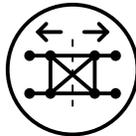
EPPS An entangled photon pair source, implemented using e.g. symmetric parametric down conversion (SPDC). An EPPS simply produces pairs of entangled photons, which must be captured or measured at link end points. An EPPS can be used in terrestrial links [47] or on a satellite, with the photons captured by telescopes on the ground [92].



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



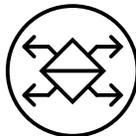
RGSS Repeater Graph State Source generates entangled multipartite photonic states used in memoryless repeater networks. It sends one half of the generated repeater graph state to its neighboring nodes where the photons are measured.



ABSA Advanced Bell State Analyzer. The basic BSA always performs the same operation, but all-optical 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.

4.2 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 quan-

tum 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) [86]. 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 [27]. 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 [86].

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.

4.3 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 [7]. Aparicio studied aggregate throughput and fairness for these approaches, and found that statistical multiplexing works pretty well [3,4]. 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.

4.4 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 [31].

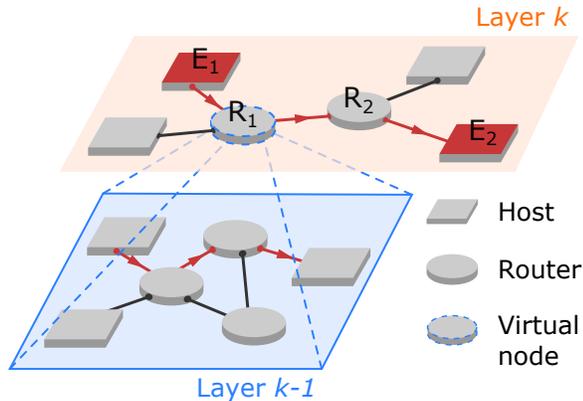


Figure 5: QRNA uses a fully recursive architecture that can virtualize a network as a node. Note that QRNA can work down to the link layer, or networks can be internally different [52] as long as they participate at the network border.

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.

4.5 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 [75].

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.

5 Internetworking and Scalability: Recursion

An idealization of today’s Internet is that it is a two-level system. External gateway protocols such BGP are used for routing between networks while internal gateway protocols such as OSPF and IS-IS are responsible for routing within

the networks. The reality, however, is not so elegant. Tunneling, switched Ethernets requiring spanning tree protocol underneath even though they are nominally “link-layer”, and recent emphasis on virtualization of networks and services [7] has shaped the Internet into a multi-tier system with ad hoc interactions at each level. Given the opportunity to create the system from scratch, and knowing the evolution path that the Internet has taken, we would probably design the Internet in a unified way that naturally takes into account interactions across multiple layers.

One such unified approach, known as the Recursive Network Architecture, was proposed by Touch *et al.* [79]. RNA presents an attractive blueprint for the design of the Quantum Internet, which Van Meter *et al.* named the Quantum Recursive Network Architecture (QRNA) [87]. This approach is intended to provide scalability to global proportions, including connecting physically and logically heterogeneous networks and providing autonomy, security and privacy.

Recursion naturally affects naming (Sec. 3.1) and routing (Sec. 4.2). Recursion describes the hierarchy of names; the relationship among names can be described as a directed acyclic graph. This approach provides scalability in naming and routing, and enhances autonomy, security and privacy.

Traditionally, connections may be of two types; boundary-to-boundary for transit and boundary-to-end node for termination. In QRNA, both of these connections are treated as the same thing but at different levels of the network. In Fig. 5, a host node E_1 wishes to establish a end-to-end connection with another host node E_2 at Layer k . From the perspective of Layer k the path to E_2 is straightforward and leads through routers R_1 and R_2 . When the connection request reaches the first router it is embedded and passed to Layer $k - 1$ by the border router. The border router is then responsible for requesting an end-to-end connection across Layer $k - 1$ to an appropriate border router that then passes the original connection request up to Layer k . The recursive nature of the architecture allows the connection requests to be embedded into as many levels as is required.

5.1 Connection Setup: Two-Pass with Rewrite

Recursion must work with the two-pass connection setup described in Sec. 3.5. We accomplish this via RuleSet rewriting where crossing recursion layers, such as at network boundaries. Setup in an internetwork is shown in Fig. 6 (compare to Fig. 4). In order to maintain network autonomy and privacy and improve scalability, the border router rewrites the existing set of link information into a single hop, much like a single hop in BGP routing hides network internal topological information for the same purposes. The border router acts as a Responder to the original Initiator, and its estimate of the performance of the path from the Initiator to its location is used to derive the performance characteristics it reports when describing the virtual link at a higher layer of recursion.

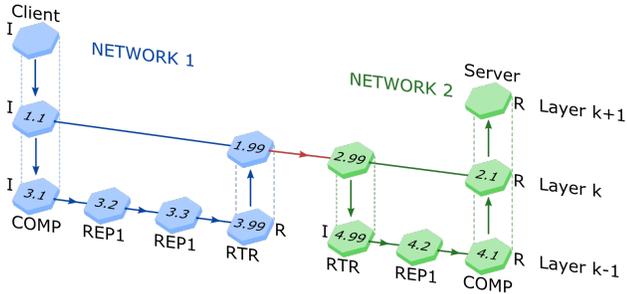


Figure 6: Two-pass connection setup in an internetwork. The arrows indicate how the initial connection request gets passed up/down the Layers and between the networks. Each Layer has its own Initiator (I) and Responder (R), due to the recursive encapsulation of the connection request Layer $k - 1$ has two pairs I-R.

In the Quantum Internet (at least through the first two generations), a connection is a form of distributed computation, with all nodes on the path participating in purification and entanglement swapping. Connections will have to be established in advance along the path, and will be *stateful*.

During connection setup, at every layer, the node is given a Responder (destination) address and can determine the nexthop based on local selection policy. To advance the setup process, the node sends the request to the neighbor (if we have reached the physical link level) or recurses. At layer k , we recurse to layer $k - 1$ by translating our k address and the k layer nexthop to layer $k - 1$ addresses, then passing to layer $k - 1$ with the latter as the new Responder address. Each network constructs provisional RuleSets upon the connection request reaching the corresponding Layer $k - 1$ Responders (3.99 and 4.1 in Fig. 6). These RuleSets are distributed backwards along the network path (not shown in Fig. 6). Upon acceptance of the connection request by the Server, a reply is sent backwards along the same path confirming the RuleSets.

Performing recursion at administrative boundaries has several benefits: a) it limits the amount of information each node has to have on hand about the entire internetwork, enhancing E2E scalability and network autonomy; b) it allows Responders to innovate (within the bounds of the RuleSet architecture); c) it allows us to reason about connections effectively; d) it serves as a convenient point for 1G-2G inter-operation as new technologies are deployed [65]; and e) it facilitates interoperation with different network architectures [52].

5.2 Routing, Multiplexing and AAA

The core routing problem of selecting a path in Internet-scale systems is solved, as noted above, using two-level or three-level systems, with the Internet’s top level being the global BGP. However, issues of policy, economics and especially

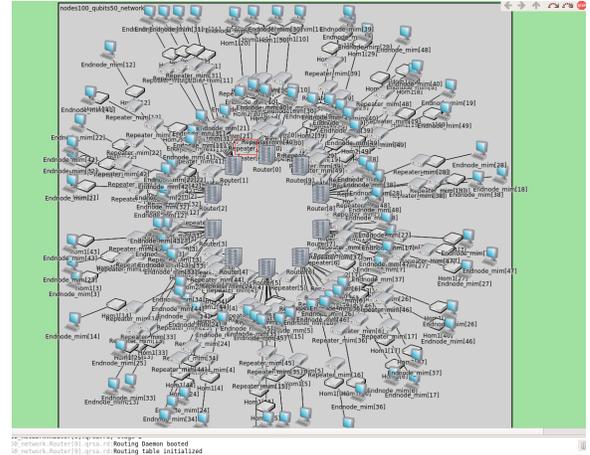


Figure 7: Our open source simulator, QuISP, focuses on protocol and scaling issues in order to further network and internetwork architecture research. Here, 100 COMP, 110 REP1, and 10 RTR nodes, collectively having 44,000 memory qubits, are connected via 220 links. 100 of the links use BSA nodes; the rest are direct connections.

of security still exist at the top level [26]. In QRNA, this approach is generalized and extended using full recursion; a routing protocol is required at each layer.

At the lowest layer, we follow the qDijkstra link cost metric of seconds per Bell pair at a particular fidelity. Using QRNA’s recursion, at the next layer up, the intra-network path will appear as a link. This link will, in turn, have a reportable performance metric of Bell pair creation rate. However, as the intra-network RuleSet can be tuned with different numbers of rounds of purification, that rate can be traded off for higher fidelity. Prior work has shown that performing more purification closer to the link level results in higher end-to-end throughput [84], so we expect the policy to be set such that each network presents a slower but higher quality link.

A bigger problem is multiplexing, which as noted requires AAA. Any network will have many connections originating, terminating or transiting. The Internet community unfortunately provides less guidance here; inter-domain QoS mechanisms have been under development since the 1990s but are not widely deployed. Thus, we consider this to be one of the most important open research issues.

6 Evidence

In an ideal world, the long-term proof of an architecture would be widespread adoption. In reality, of course, plenty of splendid architectures (processors, OSes, communication systems) have fallen by the wayside for reasons unrelated to technology. Moreover, such a retrospective view does not help us assess a prospective architecture. Here, we discuss how the RuleSet approach leads to robust protocols, and how we are validating

our architecture via simulation, documenting the protocols, and working toward real-world implementation.

6.1 Reasoning with RuleSets

A key purpose of the RuleSet architecture is to make it possible to reason rigorously about distributed behavior. At any point in time, we can enumerate the set of possible events at all nodes and ask if execution of specific Action Clauses will result in unwanted operation, such as *leapfrogging*. In leapfrogging, in a chain of nodes A-B-C-D, if B and C are each tasked with performing entanglement swapping, uncoordinated selection of resource states can result in A-C and B-D entanglement, rather than the desired A-D entanglement.

In another example, if A-B entanglement has been achieved and B is waiting on B-C entanglement to perform swapping, a *race condition* can occur in which A decides to discard the Bell pair (due to memory decoherence incurred during a long wait) just as B receives notification of B-C entanglement and performs the swapping operation. The message from B to A informing A of the swapping event arrives too late, and A has already reinitialized its qubit for reuse. This is especially problematic if C chooses, upon receipt of notification from B, to use the ersatz A-C pair to teleport C's important data to A. Using RuleSet logic, we can detect this potential race condition and define Rules such that A will not discard the Bell pair until after B does, by giving B a discard timer that is more than the one-way messaging latency with A.

6.2 Simulation, Specification and Implementation

To validate our designs, we are implementing a highly scalable simulator called QuISP (Quantum Internet Simulation Package) [74]. 1G networks, entanglement swapping and purification governed by RuleSets, and connection setup are complete (but continue to evolve); rudimentary routing and circuit switched multiplexing are all functional and pass included tests, but remain in active development. All-optical paths are in active development. RuleSets are currently being designed for 2G and multi-party states. The full QRNA protocol set is in design, and the simulator's performance has been measured to scale adequately for hundreds of nodes on a laptop, enough to demonstrate complex, multi-level, recursive internetworking.

Any network system, especially one intended to be open, must be supported by specifications for protocols and behavior. The difficulty of writing such documents can be viewed as one piece of evidence about the elegance and simplicity of an architecture. Our simulator work began with a set of design documents, and we have specifications for some of the core protocols now in development.

Moreover, we are working closely with the Quantum Internet Task Force (QITF), a quantum Internet testbed initiative

that expects to build not only a single network but to actually focus on scalability in network and internetwork architecture. We expect some aspects of the architecture presented here to be adopted directly, while others doubtless will undergo significant evolution as a result of the collaboration.

7 Conclusion

Ultimately, our proposed quantum internetwork architecture builds on three critical points: a recursive architecture for internetworking and scalability, RuleSet-based connection operation providing the right vocabulary across disparate hardware, and a two-pass connection setup routine (outbound info collection, inbound RuleSet distribution). This structure will allow for continuing evolution of the internetwork, providing a platform for distributed, independent advances in physical technology and in protocols.

Our work is maturing rapidly, with design, specification, and simulation well advanced and real-world implementation in the serious planning stages. In particular, with different groups now involved in detailed discussions, the RuleSet design will be challenged to work in heterogeneous environments, which we expect to further validate the general approach even as it is likely that details will change. Although there is solid work on routing and multiplexing, designing a system that will be robust at scale and that will serve us well for decades is perhaps the area of most concern.

With this structure in place, we feel that architecture and protocols are on pace to mature to usable levels alongside hardware, though as noted in the introduction experience shows that architecture matures more slowly. However, we expect to take full advantage of knowledge gained over the last half-century of data networking research and development. This should carry us through evolutionary stages to a full Quantum Internet supporting cryptographic, sensor, and distributed computation applications.

Acknowledgments and Availability

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-19-1-4038.

The authors thank Joe Touch for clarification of past contributions.

Our open source simulator ³ and in-preparation RFC-like specifications are or will be made available on the web.

³<https://github.com/sfc-aqua/quisp#quisp>

References

- [1] Scott Aaronson. *Quantum computing since Democritus*. Cambridge University Press, 2013.
- [2] R. Alléaume, C. Branciard, J. Bouda, T. Debuisschert, M. Dianati, N. Gisin, M. Godfrey, P. Grangier, T. Länger, N. Lütkenhaus, C. Monyk, P. Painchault, M. Peev, A. Poppe, T. Pornin, J. Rarity, R. Renner, G. Ribordy, M. Riguidel, L. Salvail, A. Shields, H. Weinfurter, and A. Zeilinger. Using quantum key distribution for cryptographic purposes: A survey. *Theoretical Computer Science*, 560, Part 1:62 – 81, 2014. Theoretical Aspects of Quantum Cryptography, celebrating 30 years of BB84.
- [3] Luciano Aparicio. Design and evaluation of communication protocols for quantum repeater networks. Master’s thesis, University of Tokyo, 2011.
- [4] Luciano Aparicio and Rodney Van Meter. Multiplexing schemes for quantum repeater networks. In *Proc. SPIE*, volume 8163, page 816308, August 2011.
- [5] David Awschalom, Karl K. Berggren, Hannes Bernien, Sunil Bhave, Lincoln D. Carr, Paul Davids, Sophia E. Economou, Dirk Englund, Andrei Faraon, Martin Fejer, Saikat Guha, Martin V. Gustafsson, Evelyn Hu, Liang Jiang, Jungsang Kim, Boris Korzh, Prem Kumar, Paul G. Kwiat, Marko Lončar, Mikhail D. Lukin, David A.B. Miller, Christopher Monroe, Sae Woo Nam, Prineha Narang, Jason S. Orcutt, Michael G. Raymer, Amir H. Safavi-Naeini, Maria Spiropulu, Kartik Srinivasan, Shuo Sun, Jelena Vučković, Edo Waks, Ronald Walsworth, Andrew M. Weiner, and Zheshen Zhang. Development of quantum interconnects (quics) for next-generation information technologies. *PRX Quantum*, 2:017002, Feb 2021.
- [6] Koji Azuma, Kiyoshi Tamaki, and Hoi-Kwong Lo. All-photonic quantum repeaters. *Nature Communications*, 6(1):1–7, 2015.
- [7] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020.
- [8] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Reference frames, superselection rules, and quantum information. *Rev. Mod. Phys.*, 79:555–609, Apr 2007.
- [9] M. Ben-Or and A. Hassidim. Fast quantum Byzantine agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 481–485. ACM, 2005.
- [10] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proc. IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179. IEEE, December 1984.
- [11] C. H. Bennett, G. Brassard, C. Crépeau, R. Josza, A. Peres, and W. Wootters. Teleporting an unknown quantum state via dual classical and EPR channels. *Physical Review Letters*, 70:1895–1899, 1993.
- [12] Charles H. Bennett, Gilles Brassard, and N. David Mermin. Quantum cryptography without Bell’s theorem. *Phys. Rev. Lett.*, 68:557–559, Feb 1992.
- [13] H.-J. Briegel, W. Dür, J.I. Cirac, and P. Zoller. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters*, 81:5932–5935, 1998.
- [14] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526, 2009.
- [15] Harry Buhrman and Hein Röhrig. *Mathematical Foundations of Computer Science 2003*, chapter Distributed Quantum Computing, pages 1–20. Springer-Verlag, 2003.
- [16] Carlton M. Caves, Joshua Combes, Zhang Jiang, and Shashank Pandey. Quantum limits on phase-preserving linear amplifiers. *Phys. Rev. A*, 86:063802, Dec 2012.
- [17] A. Chia, M. Hajdušek, R. Nair, R. Fazio, L. C. Kwek, and V. Vedral. Phase-preserving linear amplifiers not simulable by the parametric amplifier. *Phys. Rev. Lett.*, 125:163603, Oct 2020.
- [18] Andy Chia, Michal Hajdušek, Rosario Fazio, Leong-Chuan Kwek, and Vlatko Vedral. Phase diffusion and the small-noise approximation in linear amplifiers: Limitations and beyond. *Quantum*, 3:200, November 2019.
- [19] Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180, 2017.
- [20] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proc. Symposium on Theory of Computing*. ACM, 2002.
- [21] Daniele Cuomo, Marcello Caleffi, and Angela Sara Cacciapuoti. Towards a distributed quantum computing ecosystem. *IET Quantum Communication*, 1(1):3–8, 2020.

- [22] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arjan Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication, SIGCOMM '19*, page 159–173, New York, NY, USA, 2019. Association for Computing Machinery.
- [23] Simon J Devitt, Andrew D Greentree, Ashley M Stephens, and Rodney Van Meter. High-speed quantum networking by ship. *Scientific Reports*, 6:36163, 2016.
- [24] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013.
- [25] D.P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000.
- [26] Jerome Durand, Ivan Pepelnjak, and Gert Doering. BGP Operations and Security. RFC 7454, February 2015.
- [27] Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. *Nature Reviews Physics*, 2(7):382–390, 2020.
- [28] A.K. Ekert. Quantum cryptography based on Bell’s theorem. *Physical Review Letters*, 67(6):661–663, 1991.
- [29] Artur Ekert and Renato Renner. The ultimate physical limits of privacy. *Nature*, 507(7493):443–447, 2014.
- [30] Chip Elliott, David Pearson, and Gregory Troxel. Quantum cryptography in practice. In *Proc. SIGCOMM 2003*. ACM, ACM, August 2003.
- [31] Victor Fajardo, Jari Arkko, John A. Loughney, and Glen Zorn. Diameter Base Protocol. RFC 6733, October 2012.
- [32] Joseph F Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):1–11, 2017.
- [33] Austin G. Fowler, David S. Wang, Charles D. Hill, Thaddeus D. Ladd, Rodney Van Meter, and Lloyd C. L. Hollenberg. Surface code quantum communication. *Phys. Rev. Lett.*, 104(18):180503, May 2010.
- [34] Future Learn. Understanding quantum computers. <https://www.futurelearn.com/courses/intro-to-quantum-computing>.
- [35] Daniel Gottesman, Thomas Jennewein, and Sarah Croke. Longer-baseline telescopes using quantum repeaters. *Phys. Rev. Lett.*, 109:070503, Aug 2012.
- [36] Michal Hajdušek and Vlatko Vedral. Entanglement in pure and thermal cluster states. *New Journal of Physics*, 12(5):053015, 2010.
- [37] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203, 2017.
- [38] Masahito Hayashi and Michal Hajdušek. Self-guaranteed measurement-based quantum computation. *Phys. Rev. A*, 97:052308, May 2018.
- [39] M. Hein, J. Eisert, and H. J. Briegel. Multiparty entanglement in graph states. *Phys. Rev. A*, 69:062311, Jun 2004.
- [40] Paul Hilaire, Edwin Barnes, and Sophia E. Economou. Resource requirements for efficient quantum communication using all-photonic graph states generated from a few matter qubits. *Quantum*, 5:397, February 2021.
- [41] M. Hillery, V. Bužek, and A. Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829–1834, 1999.
- [42] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865–942, Jun 2009.
- [43] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert. “Event-ready-detectors” Bell experiment via entanglement swapping. *Phys. Rev. Lett.*, 71:4287–4290, Dec 1993.
- [44] Ebubechukwu O Ilo-Okeke, Louis Tessler, Jonathan P Dowling, and Tim Byrnes. Remote quantum clock synchronization without synchronized clocks. *npj Quantum Information*, 4(1):1–5, 2018.
- [45] Tanvirul Islam, Loïck Magnin, Brandon Sorg, and Stephanie Wehner. Spatial reference frame agreement in quantum networks. *New Journal of Physics*, 16(6):063040, 2014.
- [46] Liang Jiang, J. M. Taylor, Kae Nemoto, W. J. Munro, Rodney Van Meter, and M. D. Lukin. Quantum repeater with encoding. *Phys. Rev. A*, 79(3):032325, Mar 2009.
- [47] Cody Jones, Danny Kim, Matthew T Rakher, Paul G Kwiat, and Thaddeus D Ladd. Design and analysis of communication protocols for quantum repeater networks. *New Journal of Physics*, 18(8):083015, 2016.

- [48] Anders Karlsson, Masato Koashi, and Nobuyuki Imoto. Quantum entanglement for secret sharing and secret splitting. *Phys. Rev. A*, 59:162–168, Jan 1999.
- [49] E. T. Khabiboulline, J. Borregaard, K. De Greve, and M. D. Lukin. Quantum-assisted telescope arrays. *Phys. Rev. A*, 100:022316, Aug 2019.
- [50] H. J. Kimble. The quantum Internet. *Nature*, 453:1023–1030, June 2008.
- [51] P. Kómár, E.M. Kessler, M. Bishof, L. Jiang, A. S. Sorensen, and M. D. Lukin. A quantum network of clocks. *Nature Physics*, June 2014.
- [52] Wojciech Kozłowski, Axel Dahlberg, and Stephanie Wehner. Designing a quantum network protocol. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '20*, page 1–16, New York, NY, USA, 2020. Association for Computing Machinery.
- [53] Wojciech Kozłowski, Stephanie Wehner, Rodney Van Meter, Bruno Rijsman, Angela Sara Cacciapuoti, Marcello Caleffi, and Shota Nagayama. Architectural principles for a quantum internet. Internet-Draft draft-irtf-qirg-principles-06, IETF Secretariat, February 2021.
- [54] T.D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J.L. O’Brien. Quantum computers. *Nature*, 464:45–53, March 2010.
- [55] D. Markham and B.C. Sanders. Graph states for quantum secret sharing. *Physical Review A*, 78(4):42309, 2008.
- [56] S. Massar and S. Popescu. Optimal extraction of information from finite quantum ensembles. *Phys. Rev. Lett.*, 74:1259–1263, Feb 1995.
- [57] Takaaki Matsuo. Simulation of a dynamic, ruleset-based quantum network. *arXiv preprint arXiv:1908.10758*, 2019.
- [58] Takaaki Matsuo, Clément Durand, and Rodney Van Meter. Quantum link bootstrapping using a ruleset-based communication protocol. *Phys. Rev. A*, 100:052320, Nov 2019.
- [59] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74, 2008.
- [60] Alan Mink, Sheila Frankel, and Ray Perlner. Quantum key distribution (QKD) and commodity security protocols: Introduction and integration. *International Journal of Network Security & Its Applications (IJNSA)*, 1(2), July 2009.
- [61] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2:15023, 2016.
- [62] Tomoyuki Morimae and Keisuke Fujii. Blind quantum computation protocol in which alice only makes measurements. *Phys. Rev. A*, 87:050301, May 2013.
- [63] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. Optimal architectures for long distance quantum communication. *Scientific Reports*, 6:20463, 2016.
- [64] Shota Nagayama. Towards end-to-end error management for a quantum internet. Released simultaneously to arXiv., December 2021.
- [65] Shota Nagayama, Byung-Soo Choi, Simon Devitt, Shigeya Suzuki, and Rodney Van Meter. Interoperability in encoded quantum repeater networks. *Phys. Rev. A*, 93:042338, Apr 2016.
- [66] Asher Peres and Petra F. Scudo. Transmission of a cartesian frame by a quantum system. *Phys. Rev. Lett.*, 87:167901, Sep 2001.
- [67] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden. Advances in quantum cryptography. *Adv. Opt. Photon.*, 12(4):1012–1236, Dec 2020.
- [68] Matteo Pompili, Sophie LN Hermans, Simon Baier, Hans KC Beukers, Peter C Humphreys, Raymond N Schouten, Raymond FL Vermeulen, Marijn J Tiggelman, Laura dos Santos Martins, Bas Dirkse, et al. Realization of a multinode quantum network of remote solid-state qubits. *Science*, 372(6539):259–264, 2021.
- [69] Christopher Portmann and Renato Renner. Security in Quantum Cryptography, 2021. arXiv:2102.00021v1.
- [70] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.
- [71] Quantum Protocol Zoo. Protocol library. https://wiki.veriqloud.fr/index.php?title=Protocol_Library.
- [72] R. Raz. Exponential separation of quantum and classical communication complexity. *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 358–367, 1999.

- [73] Terry Rudolph and Lov Grover. Quantum communication complexity of establishing a shared reference frame. *Phys. Rev. Lett.*, 91:217905, Nov 2003.
- [74] Ryosuke Satoh, Michal Hajdušek, Naphan Benchasatatabuse, Shota Nagayama, Kentaro Teramoto, Takaaki Matsuo, Sara Ayman Metwalli, Takahiko Satoh, Shigeya Suzuki, and Rodney Van Meter. QuISP: a quantum internet simulation package. Released simultaneously to arXiv., December 2021.
- [75] Takahiko Satoh, Shota Nagayama, Shigeya Suzuki, Takaaki Matsuo, Michal Hajdušek, and Rodney Van Meter. Attacking the quantum internet. *IEEE Transactions on Quantum Engineering*, 2:1–17, 2021.
- [76] Robert S. Sutor. *Dancing with Qubits*. Packt Publishing, 2019.
- [77] Mohammad Amin Taherkhani, Keivan Navi, and Rodney Van Meter. Resource-aware system architecture model for implementation of quantum aided byzantine agreement on quantum repeater networks. *Quantum Science and Technology*, 3(1):014011, 2018.
- [78] Seiichiro Tani, Hirotada Kobayashi, and Keiji Matsumoto. Exact quantum algorithms for the leader election problem. *ACM Trans. Comput. Theory*, 4(1):1:1–1:24, March 2012.
- [79] Joe Touch, Yu-Shun Wang, and Venkata Pingali. A recursive network architecture. *ISI, Tech. Rep.*, 626, 2006.
- [80] Rodney Van Meter. Quantum networking and inter-networking. *IEEE Network*, 26(4):59–64, July/August 2012.
- [81] Rodney Van Meter. *Quantum Networking*. Wiley-ISTE, April 2014.
- [82] Rodney Van Meter. A #quantumcomputerarchitecture tweetstorm. September 2019.
- [83] Rodney Van Meter and C. Horsman. A blueprint for building a quantum computer. *Communications of the ACM*, 53(10):84–93, October 2013.
- [84] Rodney Van Meter, Thaddeus D. Ladd, W. J. Munro, and Kae Nemoto. System design for a long-line quantum repeater. *IEEE/ACM Transactions on Networking*, 17(3):1002–1013, June 2009.
- [85] Rodney Van Meter and Takaaki Matsuo. Connection setup in a quantum network. Internet-Draft draft-van-meter-qirg-quantum-connection-setup-01, IETF Secretariat, September 2019.
- [86] Rodney Van Meter, Takahiko Satoh, Thaddeus D. Ladd, William J. Munro, and Kae Nemoto. Path selection for quantum repeater networks. *Networking Science*, 3(1):82–95, 2013.
- [87] Rodney Van Meter, Joe Touch, and C. Horsman. Recursive quantum repeater networks. *arXiv preprint arXiv:1105.1238*, 2011.
- [88] Chonggang Wang, Akbar Rahman, Ruidong Li, and Melchior Aelmans. Applications and use cases for the quantum internet. Internet-Draft draft-irtf-qirg-quantum-internet-use-cases-06, IETF Secretariat, May 2021.
- [89] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 2018.
- [90] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, October 1982.
- [91] Feihu Xu, Xiongfeng Ma, Qiang Zhang, Hoi-Kwong Lo, and Jian-Wei Pan. Secure quantum key distribution with realistic devices. *Rev. Mod. Phys.*, 92:025002, May 2020.
- [92] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, Guang-Bing Li, Qi-Ming Lu, Yun-Hong Gong, Yu Xu, Shuang-Lin Li, Feng-Zhi Li, Ya-Yun Yin, Zi-Qing Jiang, Ming Li, Jian-Jun Jia, Ge Ren, Dong He, Yi-Lin Zhou, Xiao-Xiang Zhang, Na Wang, Xiang Chang, Zhen-Cai Zhu, Nai-Le Liu, Yu-Ao Chen, Chao-Yang Lu, Rong Shu, Cheng-Zhi Peng, Jian-Yu Wang, and Jian-Wei Pan. Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356(6343):1140–1144, 2017.

A Quantum Concepts

There are many good introductions to quantum computing, on the web [34] and in print [76], 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 [1]. 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 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 [90]. 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 [13]. 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 [80], [46] and [33].)

Teleportation. Teleportation destroys the state of a qubit at the sender and recreates that state at the destination, teleporting information rather than matter [11]. 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 long-distance 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 research-oriented, in-depth survey of quantum computing systems, we recommend the following short list of papers: [19, 24, 25, 37, 54, 61, 70, 82, 83]. For communication, we recommend: [5, 13, 50, 53, 88, 89].