Quantum Networks: Architecture and Techniques

Huiping Lin

April 20, 2022

Abstract

In this paper, we will give a introduction on quantum networks, describing its architecture and usable techniques. We will follow the classical 5-layer network model, and give the quantum version of each layer, including related protocols and algorithms.

1 Introduction

With the development of quantum computing, information transitions between computing centers also become important. In classical computing world, we have computer networks, which link different computers and achieve communication between them. There have been plenty of theoretical and implementation research work on classical computer networks. Different network architecture model have been proposed, including the OSI model, and the TCP/IP model. In this paper, we will discuss the 5-layer TCP/IP model and extend this model to a quantum version.

The 5-layer TCP/IP model, including physical layer, link layer, network layer, transport layer and application layer, is an abstract model of real computer network. Different layers are responsible for different tasks using different protocols.

2 Classical Network Architecture

In classical computer world, networks help the computers use common communication protocols over digital interconnections to communicate with each other. Proper protocols are vital for achieving correct communications. By definition, a network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. It's similar to human's communications. Speaking the same language simplifies communication between people, and some actions correspond to specific emotions (for example, smiling means friendly and shaking hands means cooperation).

Network protocol stack model splits the communication process between two network devices into some layers. All the layers are self-contained, and the tasks assigned to them can be executed independently. And based on this model, network protocols divide the communication process into discrete tasks across every layer of the OSI model. One or more network protocols operate at each layer in the communication exchange.

Therefore, different level takes responsible for different tasks, and protocols occur at every level of networks.

Now we will introduce the 5-layer model, and each model's tasks and corresponding protocols.

- Physical layer

  The physical layer provides an electrical, mechanical, and procedural interface to the transmission medium. The shapes and properties of the electrical connectors, the frequencies to broadcast on, the line code to use and similar low-level parameters, are specified by the physical layer.

- Link layer

  The data link layer, or layer 2, is the second layer of the seven-layer OSI model of computer networking. This layer is the protocol layer that transfers data between nodes on a network segment across the physical layer.[2] The data link layer provides the functional and procedural means to transfer data between network entities and may also provide the means to detect and possibly correct errors that can occur in the physical layer.

The data link layer is concerned with local delivery of frames between nodes on the same level of the network. Data-link frames, as these protocol data units are called, do not cross the boundaries of a local area network. Inter-network routing and global addressing are higher-layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration. In this way, the data link layer is analogous to a neighborhood traffic cop; it endeavors to arbitrate between parties contending for access to a medium, without concern for their ultimate destination. When devices attempt to use a medium simultaneously, frame collisions occur. Data-link protocols specify how devices detect and recover from such collisions, and may provide mechanisms to reduce or prevent them.

- Network layer

  The network layer provides the means of transferring variable-length network packets from a source to a destination host via one or more networks. Within the service layering semantics of the OSI network architecture, the network layer responds to service requests from the transport layer and issues service requests to the data link layer.

- Transport layer

  In computer networking, the transport layer is a conceptual division of methods in the layered architecture of protocols in the network stack in the Internet protocol suite and the OSI model. The protocols of this layer provide end-to-end communication services for applications. It provides services such as connection-oriented communication, reliability, flow control, and multiplexing.

- Application layer

  The application layer contains the communications protocols and interface methods used in process-to-process communications across an Internet Protocol (IP) computer network.The application layer only standardizes communication and depends upon the underlying transport layer protocols to establish host-to-host data transfer channels and manage the data exchange in a clientserver or peer-to-peer networking model. Though the TCP/IP application layer does not describe specific rules or data formats that applications must consider when communicating, the original specification (in RFC 1123) does rely on and recommend the robustness principle for application design.

3 Quantum Network

In this section, we will introduce some specific features of quantum network.

First, the transmisson of information is achieved by qubits, which is foundamentionally different from classical bits. The non-cloning theorem enables that qubits cannot be copied, which theoretically garantees the security of transmisson. Additionally, two qubits can share a special relation known as entanglement, even if these two qubits are stored at distant network nodes. This feature enables novel applications.

However, the entanglement is short-lived, and can only be produced over short distances by sending photons over standard telecom fiber [7]. So how to enhance the entanglement and enable long-distance quantum communication is a vital problem.

3.1 Physical Layer

In classical networks, the physical layer is responsible for generating a stream of raw bits, and using physical medium for transmission, like microwave. Similarly, in quantum network, the physical layer enables the basic representation of information: qubits and entanglement between them. Therefore, the physical layer attempts entanglement generation.

Physical layer should build entanglement between two devices, which lays groundwork for designing and implementing control and application protocols in platform independent software in order to build and scale quantum networks. There are already some quantum hardwares available to us, for example, NV platform which has achieved producing heralded entanglement using a solid state platform known as Nitrogen-Vacancy (NV) centres in diamond [8]. However, the same protocol could also be realized directly using Ion Traps [10] or Neutral Atoms [9].

The NV platform is an important platform realizing qubits transmission between two hosts. It is a few qubit quantum computer capable of arbitrary quantum gates, with an optical interface for initialization, measurement and entanglement generation.

The NV center a point defect in the diamond lattice, and its most explored and useful property is its photoluminescence, which allows observers to read out its spin-state. Additionally, NV centers in diamond have emerged as one of the most promising candidates for implementing quantum technologies because they exhibit long-lived spin quantum states and well-defined optical transitions [4]. NV centers are typically produced from single substitutional nitrogen centers (called C or P1 centers in diamond literature) by irradiation followed by annealing at temperatures above 700 řC, and irradiation produces lattice vacancies, which are a part of NV centers. Those vacancies are immobile at room temperature. Key capabilities of the NV platform have already been demonstrated, including qubit lifetimes of 1.46 s, entanglement production faster than it is lost, and using entanglement to teleport qubits between separated NV centres. Other platforms' performance is not yet good enough to generate entanglement faster than it is lost.

Through all these advantages mentioned above, we can see that the NV platform is suitable for the realization of quantum networks. To use the NV platform to generate entanglement between two nodes A and B (capable of storing and manipulating qubits), an intermediate station called the heralding station H should be connected to both A and B over optical fibers, which is responsible for entanglement swapping.

Each node can have two types of qubits: memory qubits as a local memory, and communication qubits with an optical interface. To produce entanglement, a time synchronized trigger is used at both A and B to create entanglement between each communication qubit, and a corresponding traveling qubit (photon). These photons are sent from A and B to H over fiber. When both arrive at H, H performs an automatic entanglement swapping operation which succeeds with some probability. Since H has no quantum memory, both photons must arrive at H at the same time to succeed.

In the case of success, one of several entangled states may be produced, which can however be converted to one other using local quantum gates at A and B. After a generation attempt, the communication qubit may be moved to a memory qubit, in order to free the communication qubit to produce the next entangled pair. Many parameters influence the success and quality of this process, such as the quality of the qubits themselves, the probability of emission of a photon given a trigger signal, losses in fiber, and quality of the optical elements such as detectors used at H.

To teleport qubits between separated NV centres, in [6], an initial functional allocation of a quantum network stack is proposed, which illustrates the midpoint heralding protocol (MHP) at the physical layer. MHP is a lightweight protocol built directly on top of physical implementations of the form, supplementing them with some additional control information. With minor modifications this MHP can be adapted to other forms of heralded entanglement generation between controllable nodes, even using multiple automated middle nodes.

The MHP protocol provides the two services, and takes the following actions.

- Protocol for Create and Keep (K)

  The parameters given to the MHP with a "yes" response contain the following:

  - An ID for the attempt that is forwarded to H.
  - Generation parameters α(a microwave pulse prepares the communication qubit depending on the parameter α).
  - The device qubits for storing the entanglement.
  - A sequence of operations to perform on the device memory.

  The higher layer may instruct the MHP to perform a gate on the communication qubit depending on the heralding signal from H allowing the conversion from the $|\Psi^-\rangle$ state to the $|\Psi^+\rangle$ state. Entanglement generation is then triggered at the start of the next time interval, including generation parameter α, and a GEN message is sent to H which includes a timestamp, and the given ID. The motivation for including the ID is to protect against errors in the classical control, for example losses.

  The station H uses the timestamp to link the message to a detection window in which the accompanying photons arrived. Should messages from both nodes arrive, the midpoint verifies that the IDs transmitted with the GEN messages match, and checks the detection counts (Figure 3) from the corresponding detection window. The midpoint will then send a REPLY message indicating success or failure, and in the case of success which of the two states $|\Psi^-\rangle$ and $|\Psi^+\rangle$ were produced. The REPLY additionally contains a sequence number uniquely identifying the generated pair of entangled qubits chosen by H, which later enables the EGP to assign unique entanglement identifiers. This REPLY and the ID is forwarded to the link layer for post-processing. Note that the REPLY may be received many MHP cycles later, allowing the potential for emission multiplexing.

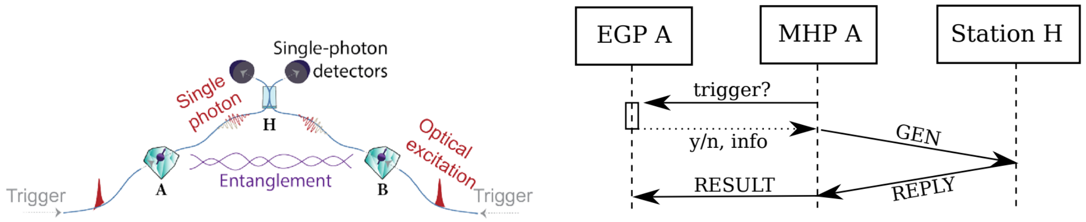


Figure 1: Entanglement Producer and Physical Layer Protocol.

- Protocol for Create and Measure (M) Handling M type requests is very similar, differing only in two ways: Instead of performing a gate on the communication qubit, the "yes" message requests the MHP to perform a measurement on the communication qubit in a specified basis once the photon has been emitted, even before receiving the response from H . The outcome of the measurement and the REPLY are passed back to the EGP. In practice, the communication time from transmitting a GEN message to receiving a REPLY may currently exceed the duration of such a local measurement, and the MHP may choose to perform the measurement only after a successful response is received.

3.2 Link Layer

In classical networks, the link layer provides the functional and procedural means to transfer data between network entities. Additionally, it can detect and possibly correct errors that can occur in the physical layer. Therefore, the existence of the link layer helps make the transmission more reliable.

In quantum networks, entanglement has been already established in the physical layer, which means two devices are already connected to each other. The link layer is then to turn the physical layer making entanglement attempts into a robust entanglement generation service, that can produce entanglement between controllable quantum nodes connected by a (chain of) automated quantum nodes. So we have two tasks in general: 1) enhance the entanglement, and 2) do error correction.

Before describing further protocols, we introduce some exsiting techniques.

3.2.1 Quantum Repeaters

In [2], quantum repeaters is proposed to help on this problem. As we've mentioned above, entanglement is short-lived, and prodused over short distances. The bottleneck for communication between distant nodes is the scaling of the error probability with the length of the channel connecting the nodes. For channels such as an optical fiber, the probability for both absorption and depolarization of a photon (i.e., the qubit) grows exponentially with the length l of the fiber. As a result, the number of trials scales exponentially with l to transmit a photon without absorption, and when a photon arrives, the fidelity of the transmitted state decreases exponentially with l. Standard purification schemes [1] can be used to solve the latter problem, but a certain minimum fidelity $F_{\min}$ is required, which cannot be achieved if l is too large. Furthermore, in any realistic situation, the operations that are part of the purification protocol are themselves imperfect, and this defines a maximum attainable fidelity $F_{\max}$ and limits the efficiency of the scheme.

Recall that in classical computer networks and classical communication, there are a type of devices called repeaters. Repeaters are used to extend transmissions so that the signal can cover longer distances or be received on the other side of an obstruction. The problem of exponential attenuation can be overcome by using repeaters at certain points in the channel, which amplify the signal and restore it to its original shape.

There are different types of repeaters, corresponding to different types of the signal's physical medium. For example, Figure 2 shows the diagrammatic sketch and certain photo of radio repeaters.

Guided by these ideas, for quantum communication between A and B over a long distance, we can divide the channel into N segments with connection points in between. We then create N elementary EPR pairs of fidelity F_1 between nodes A and C_1, C_1 and C_2, ... , C_n and B. Subsequently, we connect these pairs by making Bell measurements at the nodes C_i and classically communicating the results between the nodes as in the schemes for teleportation [14] and entanglement swapping [18]. This is totally similar to what we use in classical communication.

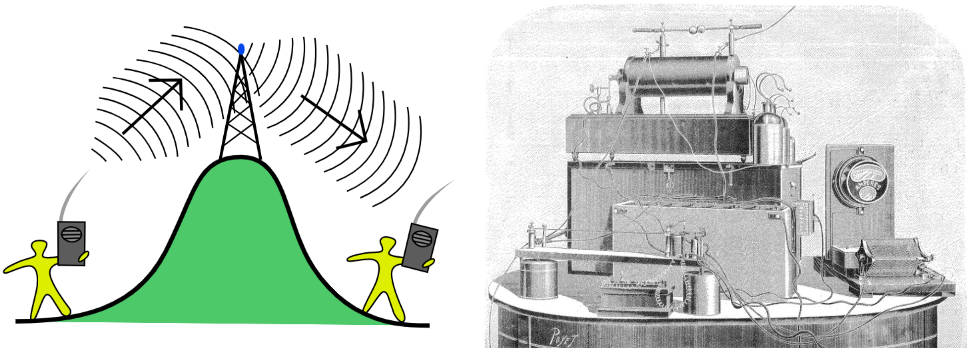


Figure 2: Radio Repeaters.

However, with every connection, the fidelity F_0 of the resulting pair may decrease: the connection process involves imperfect operations which introduce noise; and even for perfect connections, the fidelity decreases. This may lead to an exponential decrease of the fidelity F_N with N of the final pair shared between A and B. Eventually, the value of F_N drops below $F_{\min}$, and therefore it will not be possible to increase the fidelity by purification. Therefore, the number of pairs L that may be truly connected by this method seems therefore to be restricted by the condition $F_L < F_{\min}$.

Therefore, [2] proposed nested purification protocol, which combines the methods of entanglement swapping and purification into a single (meta) protocol that circumvents this restriction. Its core idea is to use nested connection instead of sequential connection. Assume that $N = L^n$ for some integer n. On the first level, we simultaneously connect the pairs (initial fidelity F_1) at all of the checkpoints except at $C_L, C_{2L}, \cdots, C_{N-L}$. As a result, we have N/L pairs of length L and fidelity F_L between A and C_L, C_L and C_{2L}, and so on. We also need to purify and obtain one pair of fidelity $\geq F_1$ on each segment by using M copies that we construct in parallel fashion. Here M depends on the initial fidelity, the degradation of the fidelity under connections, and the efficiency of the purification protocol. The total number of elementary pairs involved in constructing one of the more distant pairs of length L is LM.

On the second level, we connect L of these more distant pairs at every checkpoint $C_{kL}(k = 1, 2, \cdots)$ except at $C_{L^2}, C_{2L^2}, \cdots, C_{N-L^2}$. As a result, we have N/L^2 pairs of length L^2, of fidelity $\geq F_L$. Again, we need M parallel copies of these long pairs to repurify up to the fidelity $\geq F_1$. We iterate the procedure to higher and higher levels, until we reach the nth level. As a result, we have obtained a final pair between A and B of length N and fidelity $\geq F_1$. In this way, the total number R of elementary pairs will be $(LM)^n$. Using this method, we are able to avoid the decrease of fidelity.

In general, the basic elements of the nested purification protocol are a) nested pair connections and b) purification.

Quantum repeaters allow the creation of an entangled (EPR) pair of particles over arbitrary large distances with a tolerability of errors in the percent region.

3.2.2 Quantum Error Correction

In the physical layer, we have introduced different error types on qubits, and a typical coding method Shor code, using 9 qubits to code a single-qubit.

Actually, Shor code and many existing codes defined on qubits are examples of stabilizer codes. A detailed overview is presented in [13], which reviews the theory of fault-tolerance and quantum error-correction, discuss examples of various codes and code constructions. Stabilizer codes are attractive as (i) they are the straightforward quantum generalization of classical binary linear codes, (ii) their logical operators and distance are easily determined, and it is relatively simple to (iii) understand how to construct universal sets of logical gates and (iv) execute a numerical analysis of the code performance. The main idea of stabilizer codes is to encode k logical qubits into n physical qubits using a subspace, the code space, $\mathcal{L} \subset (\mathbb{C}^2)^{\otimes n}$ spanned by states $|\psi\rangle$ that are invariant under the action of a stabilizer group $\mathcal{S}$,

$$\mathcal{L} = \left\{ |\psi\rangle \in \left(\mathbb{C}^2\right)^{\otimes n} : P|\psi\rangle = |\psi\rangle \quad \forall P \in \mathcal{S} \right\}.$$

Here $\mathcal{S}$ is an Abelian subgroup of the Pauli group, such that $-I \notin S$.

The physical layer may apply different encoding methods, and during the transmission, errors may occur at different stages. Then, we can use error-correction method to recover the qubits. Recall that in the physical layer is responsible for encoding, and the recovery after transmission happens in the link layer. Therefore, the physial layer and link layer together accomplish the quantum error correction task.

3.3 Link Layer Protocol

In this section, we will introduce some considerations on the design of link layer, and give a sketch of the link layer protocol according to [6].

3.3.1 Desired Service

The link layer offers a robust entanglement creation service between a pair of controllable quantum nodes A and B that are connected by a quantum link. The process can be divided into two stages:

- Requesting entanglement.

  Entanglement creation can be initiated at either A or B by a CREATE request from the higher layer with parameters. The parameters includes source node ID, purpose node ID, type of request, number of entangled pairs to be created, priority, the desired minimum fidelity and some other additional tags.

- Response to entanglement requests.

  If entanglement has been produced successfully, an OK message should be returned. In addition, different use cases may desire several other pieces of information, which may also be tracked at higher layers. For example, if the request is CREATE, then the addtional information is a qubit ID for which identifies where the local qubit is in the quantum memory device; a priority value can also be set. In general, the link layer should response to the requests according to the instructions from higher layers.

  Evidently, these requests may suffer failures due to different reasons. There are many possibilities of failure resulting in the return of error messages. This includes:

  - Timeout when a request could not be fulfilled in a specific time frame (TIMEOUT).
  - An immediate rejection of the request because the requested fidelity is not achievable in the given time frame (UNSUPP).
  - The quantum storage is too small to simultaneously store all pairs of an atomic request, permanently (MEMEXCEEDED) or temporarily (OUTOFMEM).
  - Refusal by the remote node to participate (DENIED).

  Finally, we allow an EXPIRE message to be sent, indicating that the entanglement is no longer available.

- Fixed hardware parameters

  There are parameters fixed for the specific hardware platform, or change only very infrequently. As such, these may be obtained by high-level software by querying the low level system periodically, similarly to some classical network architectures. These may include the number of available qubits, the qubit memory lifetimes and so on.

3.3.2 Error Detection

Proposals for enabling quantum communication by forward communication using quantum error correction also exist, which avoid entanglement swapping. However, these have arguably much more stringent requirements in terms of hardware, putting them in a technologically more distant future: they require the ability to create entangled states consisting of a large number of photons (only 10 realized today) and densely placed repeater stations performing near perfect operations.

We typically demand some minimum fidelity $F_{\min}$ with high confidence that may also fluctuate slightly for pairs produced within a time window.

we intersperse test rounds during entanglement generation to verify the quality of the link. Such test rounds are easy to produce without the need for complex gates or extra qubits. Evidently, there exists an exact analogy in the classical networking world, where we would transmit test bits to measure the current quality of transmission, e.g. a direct analogy to network profiling to gain confidence that the non-test bits are also likely to be transmitted with roughly the same amount of error. Yet, there we typically care about correctness of a specific data item, rather than an enabling resources like entanglement.

We now explain the test used to gain confidence in transmission quality, specifically to estimate the quality parameter fidelity F. The following is a standard procedure in quantum information to estimate the fidelity of a state ρ to the entangled target state $|\Psi^-\rangle$. We emphasize that it is not possible to measure F from a single copy of the state ρ. The matrices ρ are a mathematical description of an underlying quantum system, and not a matrix that one can read or access like classical information.

The fidelity F measures how close a realized state ρ is to an ideal target state $|\Psi^-\rangle$. The fidelity of a state with the target state can be written as

$$
\begin{aligned}
F\left[\Psi^-\right] &= \left\langle \Psi^- | \rho | \Psi^- \right\rangle && (1)\\
&= 1 - \frac{\mathrm{QBER}_X + \mathrm{QBER}_Y + \mathrm{QBER}_Z}{2}, && (2)
\end{aligned}
$$

where for a fixed basis (say Z) the QBER (here QBER_Z) is the probability of receiving equal measurement outcomes.

Let us first assume a simpler scenario, in which n identical noisy entangled states ρ are produced in succession and we want to estimate F . We remark that when using imperfect quantum devices it is evidently a highly idealized situation that all states ρ are exactly identical. The testing protocol works as follows:

- Node A randomly chooses an n element string $r = r_1, \cdots, r_n \in \{X, Z, Y\}$ and sends it to Node B.
- Nodes A and B now perform the following procedure for n rounds: a) Node A produces one entangled pair ρ with Node B, b) Nodes A and B both measures their respective qubits in the basis r_j and record outcomes x_j^A (Node A) and x_j^B (Node B) respectively.
- Node $B(A)$ transmits the outcome string $x_i^B(x_i^A)$ to Node $A(B)$.
- Both nodes estimate the error rates:

$$
\begin{aligned}
QBER_Z &\approx \frac{\#\left\{j \mid x_j^A = x_j^B, r_j = Z\right\}}{\#\left\{j \mid r_j = Z\right\}}, && (3)\\
QBER_X &\approx \frac{\#\left\{j \mid x_j^A = x_j^B, r_j = X\right\}}{\#\left\{j \mid r_j = X\right\}} && (4)\\
QBER_Y &\approx \frac{\#\left\{j \mid x_j^A = x_j^B, r_j = Y\right\}}{\#\left\{j \mid r_j = Y\right\}} && (5)
\end{aligned}
$$

3.3.3 Link Layer Protocol

For the link layer, [6] presents an implementation of link layer protocol called dubbed EGP, which is built up from different components:

- Distributed Queue

  The EGP employs a distributed queue comprised of synchronized local queues at the controllable nodes. These local queues can be used to separate requests based on priority, where here we employ 3 for the different use cases (CK, NL, MD). We let one node hold the master copy of the queue, and use a simple two-way handshake for enqueing items, and a windowing mechanism to ensure fairness.

- Quantum Memory Management (QMM)

  The EGP uses the nodes quantum memory manager (QMM) to determine which physical qubits to use for generating or storing entanglement.

- Fidelity Estimation Unit (FEU)

  In order to provide information about the quality of entanglement, the EGP employs a fidelit estimation unit. This unit is given a desired quality parameter $F_{\min}$, and returns generation parameters (such as α) along with an estimated minimal completion time. Such a fidelity estimate can be calculated based on known hardware capabilities such as the quality of the memory and operations. To further improve this base estimate the EGP intersperses test rounds.

- Scheduler

  The EGP scheduler decides which request in the queue should be served next. In principle, any scheduling strategy is suitable as long as it is deterministic, ensuring that both nodes select the same request locally. This limits two-way communication, which adversely affects entanglement quality due to limited memory lifetimes.

- Protocol

  Figure 3 presents an architecture diagram visualizing the operation. The protocol begins when a higher layer at a controllable node issues a CREATE operation to the EGP specifying a desired number of entangled pairs along with $F_{\min}$ and $t_{\max}$. Upon receipt of a request the EGP will query the FEU to obtain hardware parameters (α), and a minimum completion time.

  If this time is larger than $t_{\max}$, the EGP immediately rejects the request (UNSUPP). Should the request pass this evaluation, the local node will compute a fitting min_time specifying the earliest MHP polling cycle the request may begin processing. The node then adds the request into the distributed queue shared by the nodes. This request may be rejected by the peer should the remote node have queue rules that do not accept the specified purpose ID. Then, the EGP locally rejects the request (DENIED).

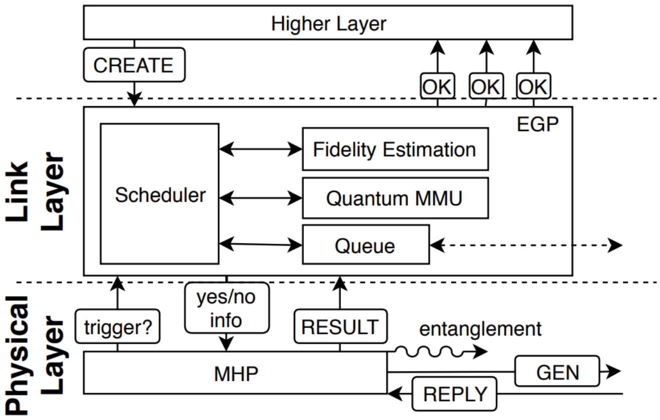


Figure 3: Flow diagram of the MHP and EGP operation. The EGP handles CREATE requests and schedules entanglement generation attempts are issued to the MHP. Replies from the midpoint are parsed and forwarded to the EGP from request management.

The local scheduler selects the next request to be processed, given that there exists a ready one (as indicated by min_time). The QMM is then used to allocate qubits needed to fulfill the specified request type (create and keep K or create and measure M). The EGP will then again ask the FEU to obtain a current parameter α due to possible fluctuations in hardware parameters during the time spent in the queue. The scheduler then constructs a “yes” response to the MHP containing α from the FEU, along with an ID containing the unique queue ID of the request in the distributed queue, and number of pairs already produced for the request. This response is then forwarded to the local MHP upon its next poll to the EGP. If no request is ready for processing, a “no” response is returned to the MHP . At this point the MHP behaves as described in the previous section and an attempt at generating entanglement is made. Whenever a REPLY and ID is received from the MHP, the EGP uses the ID to match the REPLY to an outstanding request, and evaluates the REPLY for correctness. Should the attempt be successful, the number of pairs remaining to be generated for the request is decremented and an OK message is propagated to higher layers.

In [11], a single-link multipath routing (SLMP) algorithm is proposed and in [3] a virtual-path based greedy routing algorithm in ring and grid networks is proposed. The algorithms are both designed for entanglement routing in grid networks, and the latter one also contributes on formal restrictions of network layer considerations.

3.4 Network Layer

In the physical layer and link layer, we have described protocols that can make robust entanglement as well as error correction. Recall that in classical computing networks, the network layer is responsible for long-distance transmission of data, where routing is a vital problem.

3.4.1 Network Topology

For the network layer and higher layers, we do not seriously focus on how the entanglement is established physically, or the physical material of the links or channels. Instead, we abstract all the network components into a network topology, or a a multigraph $G = \langle V, E, C \rangle$. Here, V is the set of n nodes. Each node u is a quantum processor or quantum repeater. A quantum processor is similar to an end host in classical networks, and is equipped with a limited number Q_u of qubits. All quantum processors are connected via the classical Internet and are able to freely exchange classical information. And quantum repeaters are used to support long-distances entanglement sharing via quantum swapping. E is the set of edges in the graph, and an edge existing between two nodes means that the two nodes share one or more quantum channels. A quantum channel connecting two nodes supports the transmission of qubits. The physical material of quantum channels may be polarization-maintaining optical fibers. However, a quantum channel is inherently lossy: the success rate of each attempt to create an entanglement of a quantum channel c is p_c , which decreases exponentially with the physical length of the channel: $p_c = e^{-\alpha L}$, where L is the physical length of the channel and α is a constant depending on the physical media. If an attempt is successful, the two quantum processors share an entanglement pair, and there is a quantum link on this channel. The number of channels on an edge is called the width W of the edge. C is the set of all quantum channels, each of which is identified by its two end nodes. The number of channels on an edge is called the width W of the edge.

There are also some restrictions on the abstract topology. A node can assign/bind each of its quantum memory qubits to a quantum channel, such that no qubit is assigned to more than one channel, and no channel is assigned more than one qubits at the same end of it. Channels that are assigned qubits at its both ends are bound channels, other channels are unbound channels. There could be more than one bound channels between two nodes. And two neighbor nodes may share multiple quantum links. To create a quantum entanglement, two neighbor nodes make a number of quantum entanglement attempts at the same time on the bound channels connecting them.

Similar to the classical network, in every attempt of connction, there must be a source node and a destination node. For convience, we will denote them as a S-D pair.

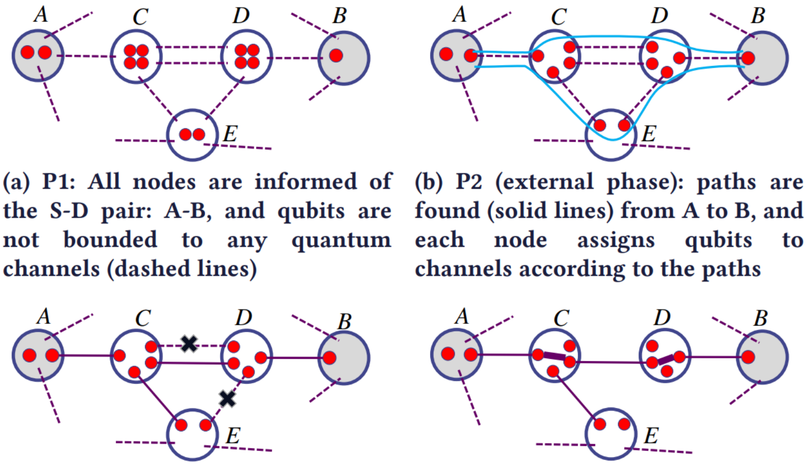
3.4.2 Time Slots

For multi-hop entanglement swapping, all nodes on the path need to establish and hold quantum entanglements with its predecessor and successor at the same time. Hence, some level of time synchronization among all nodes is necessary, which can be achieved by existing current synchronization protocols via Internet connections. Time is loosely synchronized in time slots. Each time slot is a device-technology-dependent constant and set to an appropriate duration such that the established entanglements do not discohere within one time slot. The global network topology $G = \langle V, E, C \rangle$, which is relatively stable, should be common knowledge for all nodes before any time slot.

The establishment of the connection can be viewed as 4 phases in a time slot. Each phase has its own task:

1. Phase One (P1)

   All nodes receive the information of the current S-D pairs that need to establish long-distance entanglements.



(a) P1: All nodes are informed of the S-D pair: A-B, and qubits are not bounded to any quantum channels (dashed lines)

(b) P2 (external phase): paths are found (solid lines) from A to B, and each node assigns qubits to channels according to the paths

(c) P3: each channel may fail or succeed to build an entanglement (solid line). These link states are shared within *k* hops.

(d) P4 (internal phase): each node locally decides swapping (bold solid lines in C and D). Then, A and B share a remote entanglement.

Figure 4: Phases in a time slot.

2. Phase Two (P2)

   In Phase Two, also called the external phase, paths are found for the S-D pairs, according to an identical routing algorithm running on all nodes that produces consistency results. During P2, each channel can make a number $n_c \geq 1$ of attempts, until a link is built or timeout. After P2, some quantum links may be created. We call the information of these links as link states. Compared to the same term in link-state routing of classical networks, the quantum link states are highly dynamic and non-deterministic.

3. Phase Three (P3)

   Each node knows its own link states via classical communications with its neighbors and shares its link states via the classical network. Since entanglements will quickly decay, each node can only exchange the link states with a subset of other nodes.

4. Phase Four (P4)

   In Phase 4, also called the internal phase, nodes perform entanglement swapping to establish long-distance quantum entanglement using the successful quantum links. Each node locally determines the swapping of successful entanglements, which can be considered as placing an internal link between two qubits. Each swapping succeeds at a device-dependent probability q. A and B can successfully share an entanglement qubit pair (an ebit) if there is an end-to-end path with both external and internal links.

After P4, the secret bit can be teleported from the source to the destination. Eavesdropping attempts at any repeater will be detected hence the confidentiality is preserved.

3.4.3 Routing Problem

We now give a formal definition of the entanglement routing problem.

Given a network topology $G = \langle V, E, C\rangle$, and a number of source-destination (S-D) pairs $\langle s_1, d_1\rangle$, $\langle s2,$ $d2\rangle, \cdots, \langle s_m, d_m\rangle$. The number of memory qubits of a node $u \in V$ is Q_u , and each edge $e \in E$ consists of one or more channels from C. For each bound channel c, a link is successfully built at a probability p_c in P2. In P3, each node gets the link-state information of its k-hop neighbors. Each node decides the swapping of its internal qubits in P4 locally, and each swapping succeeds in probability q.

Our purpose of entanglement routing is to maximize the number of qbits delivered for all S-D pairs in each time slot, also the throughput in one time slot.

We can define a routing metric E_t, which indicates the expected expected throughput (EXT). For a (W, h)-path P with k hops and width W, suppose the success rate of a single channel on the i-th hop is p_i. We denote the probability of the k-th hop on the path having exactly i successful links as Q_k^i, and the probability of each of the first k hops of P has $\geq i$ successful links as P_k^i. Then we get the recursive formula set, for $i \in \{1, 2, \cdots, W\}$ and $k \in \{1, 2, \cdots, h\}$:

$$Q_k^i = \left(\begin{array}{c} W \\ i \end{array} \right) p_k^i \left(1 - p_k\right)^{W-i} \tag{6}$$

$$P_1^i = Q_1^i \tag{7}$$

$$P_k^i = P_{k-1}^i \cdot \sum_{l=i}^{W} Q_k^l + Q_k^i \cdot \sum_{l=i+1}^{W} P_{k-1}^l. \tag{8}$$

With each entanglement swapping successful probability q, the EXT is:

$$E_t = q^h \cdot \sum_{i=1}^{W} i \cdot P_h^i, \tag{9}$$

and we want to maximize it.

3.4.4 Routing Algorithms

A single-link multipath routing algorithm is proposed in [11]. It is a circuit-switching style protocol. It uses a greedy algorithm to choose the internal links: consider the subgraph induced by the successful external links and the repeater nodes (at the end of the external phase), and find in it the shortest path connecting Alice and Bob. If no connected path between Alice and Bob exists, no shared ebits are generated in that time slot. If a shortest path of length k_1 is found, all internal links along the nodes of that path are attempted, and the probability a shared ebit is generated by this path is the probability that all $k_1 1$ internal link attempts were successful, i.e., $q^{k_1 1}$. We then remove all the (external and internal) links of the above path from the subgraph, and find a shortest path connecting Alice and Bob in the pruned subgraph. Note that instead of removing the links of the first path from the subgraph, we could simply search for a shortest path in the original subgraph but one that is edge disjoint from the previous path. If such a path exists, we again attempt all internal links at the nodes of this path, so the probability the path contributes to the generation of an ebit between Alice and Bob is $q^{k_2 1}$, where k_2 is the length of the second path. And finally, the entanglement generation rate achieved using this greedy algorithm is the sum of expected rates from these paths.

In [12], two concurrent routing algorithms are proposed: Q-PASS and Q-CAST.

The Q-PASS algorithm follows the four-phase time slot model with an additional offline phase. The core idea of Q-PASS is to pre-compute potential “good” paths between all possible S-D pairs based on the network topology G. The offline computation happens at the system initialization and after the network topology change. Then in each time slot, every node uses an online algorithm to make qubit-to-channel assignments based on the precomputed paths of current S-D pairs and make local swapping decisions based on local link states. The design includes both offline and online algorithms.

However, the offline algorithm in Q-PASS has two fundamental disadvantages. 1) It has to compute candidate paths for $n(n-1)/2$ pairs because it does not know the runtime S-D pairs, where n is the number of nodes. 2) The candidate paths exhibit a low utilization rate due to severe resource contention among them. As an opptimization, Q-CAST algorithm does not require any offline computation and always finds the paths if only paths exist in the residual graph.

Phase 1 and phase 3 only include standard processes and do not have special algorithmic designs. In phase 2, Q-CAST selects major paths for each S-D pair, without resource contention. Besides, contention-free recovery paths are also selected in phase 2. Phase 4 takes the major paths and recovery paths from Phase 2 and the link states from Phase 3 to compute the swapping decisions.

In phase 2, Q-CAST searches multiple contention-free paths for online S-D pairs using a greedy algorithm, which works as follows: Step 1) For every S-D pair, it uses the Extended Dijkstras algorithm to find the best path in terms of the routing metric EXT between this pair. Step 2) Among the best paths of all SD pairs, it further selects the path with the highest EXT and reserve the resources (qubits and channels) of this path, and the network topology is updated to the residual graph by removing the reserved resources.

Steps 1) and 2) are repeated with the residual graph, until no more path can be found, or the number of paths exceeds a value limiting the number of paths to avoid unnecessary computation.

Some extensions are also discussed to achieve better fairness and pioritized routing in [12].

3.5 Transport Layer Protocol

In classical networks, the transport layer is usually responsible for providing reliable data transfer, and some additional services, such as congestion control. In classical networks, we have the TCP protocol, which provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network.

In [17], the first transport layer protocols are proposed for quantum networks, to be referred to as Quantum Transport Protocols (or QTPs). Additionally, in [15], a quantum version of TCP is proposed, which is called Quantum TCP. The QTPs are designed to handle the packet loss problem in transport, and further provide reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on quantum hosts communicating via an quantum network.

We've mentioned the no-cloning theorem, which enables security during data transfer, but also prevents copying a qubit and re-transfer when encountering with packet loss. Therefore, the Q-TCP is proposed for dealing with packet loss problem uses quantum secret sharing techniques. As in detail, the Q-TCP use a quantum version three-way handshake protocol for connection establishment, and use secret sharin technique to enable retransmission protocol.

The kernel of the retransmission protocol is a recursive use of the $(2, 3)$ threshold scheme. The principle and technique of quantum secret sharing is proposed in [5]. The definition of (m, n) threshold scheme can be generally explained as: use n qubits to encode 1 qubit, and only taking $\geq m$ qubits enables the recovery of the original qubit. This protocol chooses the $(2, 3)$ threshold scheme for illustration of quantum retransmission, and the $(2, 3)$ threshold scheme is qutrit-based (a three level quantum system); we can simply represent a qutrit by two qubits without using the fourth level.

The protocol works as follows:

- A quantum message A is encoded into three shares $A_1A_2A_3$ and at least two shares are required to recover the message.
- The shares A_2 and A_3 are sent sequentially, depending on the acknowledgment of the previous share. If the first share A_2 is acknowledged, the share A_3 is directly sent. If A_3 is also acknowledged, we are done since A_2A_3 is enough for recovering A.
- If A_2 gets lost, A_1 has to be sent carefully. So A_1 is then encoded into three shares by the $(2, 3)$ threshold scheme and the procedure proceeds recursively.

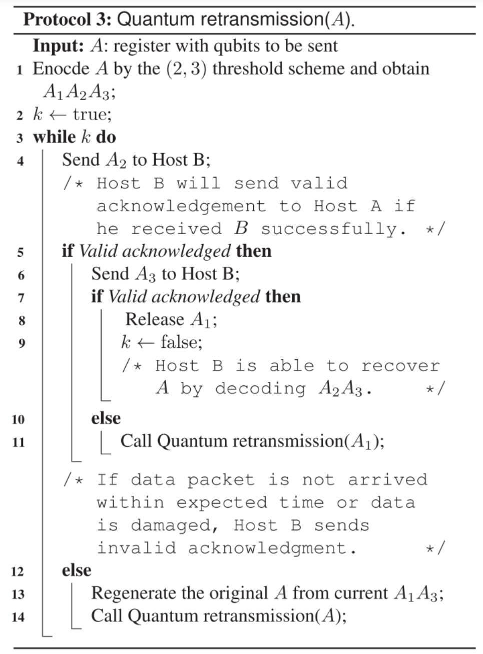
A pseudocode is shown in Fig.5.

Based on the retransmission protocol and teleportation protocol, [16] proposes a transport layer protocol Tele-QTP, which takes the AIMD algorithm as the underlying algorithm.

In a Teleportation QDN(Tele-QDN), two parties use the teleportation protocol for exchanging quantum information. Alice needs to first establish W entanglement circuits in a time slot with Bob, if Alice wants to teleport W qubits (because only one qubit can be teleported per circuit). This means Alice and Bob each needs to use at least W qubits of quantum memory, and at least $2W$ qubits of quantum memory at each of the intermediate repeaters. Unlike in a classical network, where congestion may occur during the actual data transmissions of TCP segments for example, here, the main challenge becomes how to establish these circuits given the limited quantum memory at each repeater, since once these circuits are established, teleportation of actual data qubits will not experience any congestion.

In a typical implementation of a distributed reservation approach, each source sends a request (in the classical network) for W circuits along a path to the corresponding destination (Bob), where the path is determined by a quantum routing protocol. Such a distributed protocol typically has two phases: a forward phase and a backward phase. In the forward phase, each request is processed hop-by-hop by local controllers co-located with the quantum repeaters to see if W entanglement links can be established and then forwarded to the next hop (controller) for further processing until it reaches the destination. At this time, the forward phase ends and the backward phase will start.

Recall that in a AIMD based TCP congestion control mechanism, the basic idea is to allow each source to request say W circuits in one time slot and $W + 1$ circuits in the next if possible. If/when there isn't a sufficient amount of quantum memory at a given hop to satisfy all the requests from different sources,



```
Protocol 3: Quantum retransmission($A$).
Input: $A$: register with qubits to be sent
Enocde $A$ by the $(2, 3)$ threshold scheme and obtain
  $A_1A_2A_3$;
$k \leftarrow$ true;
while $k$ do
    Send $A_2$ to Host B;
    /* Host B will send valid
       acknowledgement to Host A if
       he received $B$ successfully.  */
    if Valid acknowledged then
        Send $A_3$ to Host B;
        if Valid acknowledged then
            Release $A_1$;
            $k \leftarrow$ false;
            /* Host B is able to recover
               $A$ by decoding $A_2A_3$.      */
        else
            Call Quantum retransmission($A_1$);
    /* If data packet is not arrived
       within expected time or data
       is damaged, Host B sends
       invalid acknowledgment.        */
    else
        Regenerate the original $A$ from current $A_1A_3$;
        Call Quantum retransmission($A$);
```

Figure 5: Quantum retransmission protocol.

the protocol will cut the number circuits granted to a given source by half. In order to facilitate the presentation, hereafter, we use the terms common in TCP congestion control such as "sending window size" and "congestion encountered", respectively to refer to the number of circuits to be established, and to indicate whether the total number of requested circuits through a given repeater has exceeded the number of quantum memory it has or not. Note that other heuristics can also be used, and we leave them for future studies.

During the first step (i.e., the forward phase), Alice announces the desired sending window size, W, (or equivalently a request to reserve W circuits) along a pre-selected path by routing protocols. After receiving this announcement (or request), every repeater along the path takes note of the requested window size, and waits for similar announcements from other QTP sessions to arrive. Note that every source who wants to establish entanglement circuits in this time slot must either send out their announcements at the start of the time slot (or at the beginning of the next time slot). Assuming that it takes at most W units of time for an announcement from Alice to reach Bob, then no other destination can start Step Two earlier (even though it has already received all the announcements from other ingress nodes).

In Step Two (the backward phase), Bob initializes a response message and sends it back to Alice (along the same path but in reverse direction). The response message contains a Congestion Experienced (CE) flag to indicate if there is congestion along the selected path. The sending window size will be cut to if and only if the CE flag is set by any node.

More specifically, when a node receives (or Bob initializes) a response message, it will first check if the CE flag has been set already or not. If so, it forwards the response message to its upstream node, and tries to set up $W/2$ entanglement links with its upstream node (towards Alice). If the CE flag is not yet set, the node will determine if it can accommodate all circuits requested by all sources that will go through this node. In other words, for each and every QTP session requesting W circuits going through this node, it will determine if it could allocate a total of $2W$ units of quantum memory for the QTP session. If it could, it will establish W entanglement links with its downstream quantum node (towards Bob) and forwards the response message without setting the CE flag. Otherwise, this QTP sessions sending window size will be cut to half (as to be explained, Tele-QTP has built-in mechanisms to ensure that all nodes along the path can handle at least $W/2$ entanglement links per hop for this QTP session during this time slot. Accordingly, none of the other upstream repeater nodes will need to cut the sending window size any further. More specifically, this repeater node will do the following: (i). set the CE flag, and send the response to an upstream node; (ii). allocate $W/2$ units of quantum memory for each direction (upstream and downstream); and (iii). establish $W/2$ entanglement links with both its upstream and downstream nodes. In any case, once an entanglement link with its upstream node has been established, the repeater will perform a BSM operation and send the BSM result to Bob. Such an operation can take place while other upstream repeaters process the response message. Meanwhile, Bob will store the BSM result, along with the BSM results from all other repeater along the path, and use them when Alice performs a teleportation and sends her BSM results to Bob.

References

[1] Charles H Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A Smolin, and William K Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical review letters*, 76(5):722, 1996. 3.2.1

[2] H-J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters*, 81(26):5932, 1998. 3.2.1, 3.2.1

[3] Kaushik Chakraborty, Filip Rozpedek, Axel Dahlberg, and Stephanie Wehner. Distributed routing in a quantum internet. *arXiv preprint arXiv:1907.11630*, 2019. 3.3.3

[4] Lilian Childress and Ronald Hanson. Diamond nv centers for quantum computing and quantum networks. *MRS bulletin*, 38(2):134–138, 2013. 3.1

[5] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648, 1999. 3.5

[6] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, et al. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 159–173. 2019. 3.1, 3.3, 3.3.3

[7] James F Dynes, Hiroki Takesue, Zhiliang L Yuan, Andrew W Sharpe, K Harada, Toshimori Honjo, Hidehiko Kamada, Osamu Tadanaga, Yoshiki Nishida, Masaki Asobe, et al. Efficient entanglement distribution over 200 kilometers. *Optics express*, 17(14):11440–11449, 2009. 3

[8] Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenberg, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, et al. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575):682–686, 2015. 3.1

[9] Julian Hofmann, Michael Krug, Norbert Ortegel, Lea Gérard, Markus Weber, Wenjamin Rosenfeld, and Harald Weinfurter. Heralded entanglement between widely separated atoms. *Science*, 337(6090):72–75, 2012. 3.1

[10] David L Moehring, Peter Maunz, Steve Olmschenk, Kelly C Younge, Dzmitry N Matsukevich, L-M Duan, and Christopher Monroe. Entanglement of single-atom quantum bits at a distance. *Nature*, 449(7158):68–71, 2007. 3.1

[11] Mihir Pant, Hari Krovi, Don Towsley, Leandros Tassiulas, Liang Jiang, Prithwish Basu, Dirk Englund, and Saikat Guha. Routing entanglement in the quantum internet. *npj Quantum Information*, 5(1):1–9, 2019. 3.3.3, 3.4.4

[12] Shouqian Shi and Chen Qian. Concurrent entanglement routing for quantum networks: Model and designs. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 62–75, 2020. 3.4.4

[13] Barbara M Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307, 2015. 3.2.2

[14] Lev Vaidman. Teleportation of quantum states. *Physical Review A*, 49(2):1473, 1994. 3.2.1

[15] Nengkun Yu, Ching-Yi Lai, and Li Zhou. Protocols for packet quantum network intercommunication. *IEEE Transactions on Quantum Engineering*, 2:1–9, 2021. 3.5

[16] Yangming Zhao and Chunming Qiao. Quantum transport protocols for distributed quantum computing. *arXiv preprint arXiv:2105.08109*, 2021. 3.5

[17] Yangming Zhao and Chunming Qiao. Redundant entanglement provisioning and selection for throughput maximization in quantum networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021. 3.5

[18] Marek Zukowski, Anton Zeilinger, Michael A Horne, and Aarthur K Ekert. " event-ready-detectors" bell experiment via entanglement swapping. *Physical Review Letters*, 71(26), 1993. 3.2.1