

Quantum link bootstrapping using a RuleSet-based communication protocol

Takaaki Matsuo ¹, Clément Durand,² and Rodney Van Meter¹

¹*Keio University Shonan Fujisawa Campus, 5322 Endo, Fujisawa, Kanagawa 252-0882, Japan*

²*École Polytechnique, Route de Saclay, F-91128 Palaiseau, France*



(Received 22 April 2019; published 15 November 2019)

Establishing end-to-end quantum connections requires quantified link characteristics, and operations need to coordinate decision making between nodes across a network. We introduce the RuleSet-based communication protocol for supporting quantum operations over distant nodes to minimize classical packet transmissions for guaranteeing synchronicity. RuleSets are distributed to nodes along a path at connection setup time, and hold lists of operations that need to be performed in real time. We simulate the RuleSet-based quantum link bootstrapping protocol, which consists of recurrent purifications and link-level tomography, to quantify the quantum link fidelity and its throughput. Our Markov-chain Monte Carlo simulation includes various error sources, such as the memory error, gate error, and channel error, modeled on currently available hardware. We found that when two quantum nodes, each with 100 memory qubits capable of emitting photons ideally to the optical fiber, are physically connected with a 10-km MeetInTheMiddle link, the Recurrent Single selection–Single error purification (RSs-Sp) protocol is capable of bringing up the fidelity from an average input $F_r = 0.675$ to around $F_r = 0.865$ with a generation rate of 1106 Bell pairs per second, as determined by simulated tomography. The system gets noisier with longer channels, in which case errors may develop faster than the purification gain. In such a situation, a stronger purification method, such as the double selection-based purification, shows an advantage for improving the fidelity. The knowledge acquired from bootstrapping can later be distributed to nodes within the same network, and used for other purposes such as route selection.

DOI: [10.1103/PhysRevA.100.052320](https://doi.org/10.1103/PhysRevA.100.052320)

I. INTRODUCTION

The Quantum Internet will be a world-wide network interconnecting diverse quantum networks, both small scale and large scale [1–4]. These independent, interconnected networks are utilizing different technologies and managed by different organizations (see Fig. 1), known as Autonomous Systems in the classical Internet. The role of such a network is similar to the classical Internet: to provide to users a quantum information service between arbitrary nodes.

The Quantum Internet brings us new capabilities that fundamentally cannot be reproduced by classical technologies. Some of the well-known examples include *quantum key distribution* (QKD) to securely share strings of random, secret classical bits suitable for encrypting messages [6–9], accurately synchronizing clocks over a network [10,11], distributed computing such as the secure delegate quantum computing service named *quantum blind computing* [12,13], and other cases involving more than one quantum computer working on difficult problems that cannot be solved by classical supercomputers.

Quantum repeaters, introduced by Briegel *et al.* in 1998 [14], are the core idea of a robust quantum network. A quantum repeater node has four main roles. First, a repeater needs to be able to physically create, distribute, and store entangled resources between neighbors [15]. The second role is to manage errors on qubits. Errors can be corrected via quantum error correction [16,17], or detected and discarded from the system via quantum purification [14]. Third, the node connects stored resources to increase the span of entangled

states over a multihop route [18–20], typically via entanglement swapping [21]. Lastly, each node needs to participate in management of the network.

Quantum systems are inherently noisy. Knill and Laflamme discussed the imperfections of quantum computer devices, and introduced a fault-tolerant quantum communication scheme in 1996 [22]. Later, Muralidharan *et al.* categorized quantum repeaters, based on capabilities, into three generation classes [23]. The first-generation quantum repeater network works based on Purify-and-Swap [24–26]. The main task of a repeater node is to perform purification to detect and discard erroneous resources, and to perform entanglement swapping to connect nonadjacent nodes. While this scheme is relatively simple and straightforward, its capability could strictly depend on the distance, mainly limited by classical latencies for receiving acknowledgments regarding purification and entanglement swapping. Its performance is also known to be limited by memory lifetime [27]. The second generation utilizes encoded Bell pairs prepared between adjacent nodes, and performs quantum error correction [16,17,28]. Swapping is done at the logical level for [17], via an extended surface for [16]. In the third generation, quantum states are directly encoded to a block of physical qubits that will be sent through the channel. The receiver node can correct errors using the received physical qubits. The third generation is very similar to the second generation, but requires a very high entanglement success rate. These two generations are semantically identical but temporal behavior of the third generation is more like classical packet forwarding networks.

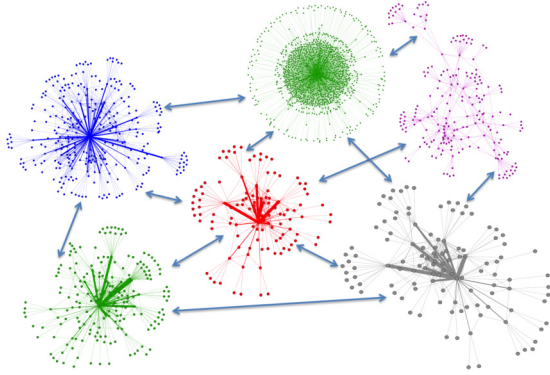


FIG. 1. The Quantum Internet. Different networks are operated under different technologies. That includes the physical link architecture, data link protocol, routing protocol, and any other necessary technology layered as in the OSI reference model [5]. Connecting those networks forms the Quantum Internet.

In all three generations, establishing an end-to-end quantum connection requires knowledge regarding the links, and cannot be accomplished without coordinating quantum operations among the nodes involved [29]. The link bootstrapping protocol is capable of quantifying the achievable link fidelity [30]. Quantified link fidelity can be used for different purposes such as for quantum routing and end-to-end connection setup. For example, the order of performing entanglement swapping impacts the output fidelity, even over the same linear path, when links have different capabilities [19] (see Fig. 2).

In this paper, we introduce the RuleSet-based communication protocol, which we use to enforce consistency in operations over a particular connection without the need to exchange classical messages before each operation. This allows each node to make coordinated decisions autonomously, reducing wait round-trip times which results in greater performance [31]. Using RuleSets, we simulate quantum link bootstrapping that consists of recurrent purifications [32] and density matrix reconstruction via link-level tomography [33], over a quantum repeater network running under specific data link protocols based on [34] with currently available hardware specifications (see Table I in Sec. IV for details). We also estimate the link throughput from the bootstrapping process time relative to the measured Bell pairs. We execute the recurrent purification based on the standard single selection or on double selection [35], and find that switching from double selection to single selection in the middle of the protocol offers an advantage in terms of fidelity and throughput when resources between adjacent nodes have low fidelity.

II. LINK ARCHITECTURES

Jones *et al.* introduced three different quantum data link protocols, MeetInTheMiddle, SenderReceiver, and MidpointSource, for distinct physical layer architectures [34]. In this paper, we leave MidpointSource as future work, because the hardware requirements are noticeably different from the

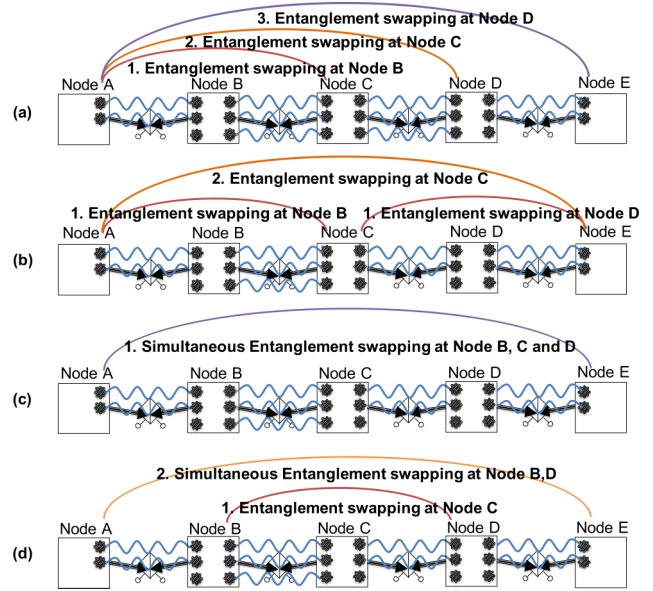


FIG. 2. Examples of entanglement swapping breakdowns without multihop purification (each link may require purification). (a) Perform entanglement swapping from left to right. This keeps the information flow consistent, but requires many steps. (b) Node B and Node D perform entanglement swapping first, and Node C connects the end nodes afterwards. This has one less step than the previous one. (c) All middle nodes perform entanglement swapping simultaneously. This requires only one step, but it may work poorly with low quality links. (d) Ad hoc. Entanglement swapping is performed as soon as resources are available. The number of steps varies from execution to execution.

TABLE I. Default parameters used for all simulations (unless explicitly mentioned).

Key	Value
Fiber refractive index	1.44 [44]
Fiber Pauli error rate (per km)	0.03 ^a
Fiber photon loss rate (per km)	0.04501 ^b [44]
Memory Pauli error rate (per second)	1/3 ^a [45]
Memory lifetime	50 ms ^c
Emission probability into the zero phonon line	0.46 ^d [46]
Photon collection efficiency	0.49 ^d [36,47]
Photon detector efficiency	0.8 [47]
Photon detector dark-count rate (per second)	10 [47]
Photon detector recovery time	1 ns [48]
Single-qubit gate error rate	0.0005 ^a [49]
Multiqubit gate error rate	0.02 ^a [50]
Measurement error rate	0.05 ^a [51]
Number of memory qubits (per QNIC)	100

^aIncluding X, Y, and Z error.

^bEquivalent to 0.2 dB/km.

^cMemory lifetime can be, for example, up to the order to seconds [45] depending on the system. We use an artificial value of 50 ms for the simulation. The ratio of excitation and relaxation probability is set to 100:1.

^dPhoton emission probability (memory-to-fiber) is the product of the emission probability into the zero phonon line and the collection efficiency.

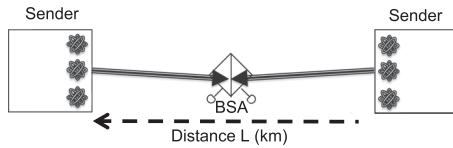


FIG. 3. The MeetInTheMiddle model. In the real world, the position of the BSA could be limited for various reasons, such as geographical constraints.

other two. We briefly explain the two protocols we focus on, MeetInTheMiddle and SenderReceiver, below.

A. MeetInTheMiddle

A link based on MeetInTheMiddle involves two repeater nodes with one Bell state analyzer (BSA) in the middle (see Fig. 3). Each node is connected to the BSA via a quantum channel and a classical channel. The nodes are separated by a distance L , with an arbitrary positioning of the BSA along the channel. This is commonly implemented in the real world [36]. In this model, nodes need to synchronize their operations, so that the emitted photons arrive at the BSA simultaneously. The BSA stochastically succeeds or fails in entangling the qubits, and acknowledges the result to both nodes for each attempt. Nodes consume all entangled qubits every round, reinitialize all of their memory qubits, and restart the cycle from the beginning.

A prototype link-layer protocol for a MeetInTheMiddle link, with few-qubits quantum processors, is discussed by Dahlberg *et al.* [37]. The link-layer protocol is responsible for generating robust resources, and estimating the fidelity. Generated resources will be allocated to the upper layer dedicated for networking.

B. SenderReceiver

SenderReceiver is similar to MeetInTheMiddle, but the BSA module is installed inside one endpoint (see Fig. 4). Therefore, one node acts as a sender, while the other node with the BSA acts as a receiver. The behavior of the nodes is similar to the ones in the MeetInTheMiddle model, but the receiver is capable of resetting memory qubits in real time at every attempt, because it can immediately refer to the entanglement success and failure results locally. Therefore, the receiver can effectively utilize its memories, which will be helpful when its buffer size is smaller than that of the sender's [38,39]. Implementing the BSA in a node directly also reduces the installation cost. On the other hand, because the acknowledgment needs to be sent from the receiver to the sender, compared to MeetInTheMiddle with the same

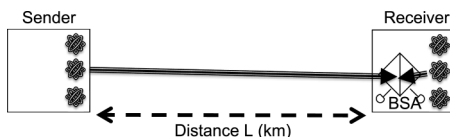


FIG. 4. The SenderReceiver model.

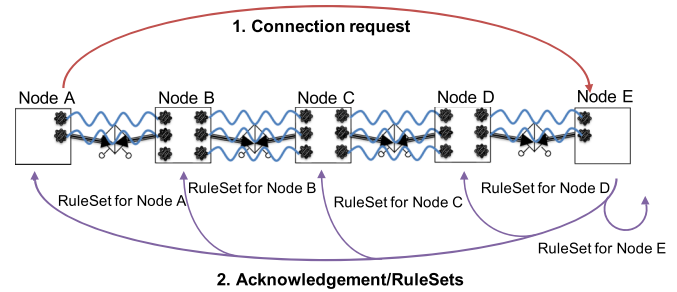


FIG. 5. Node A generating and distributing RuleSets. In this example, there are five nodes in total, which requires a total of five distinct RuleSets.

distance L , classical latency increases by up to a factor of two.

III. PROTOCOL DESIGN

In this section, we will explain the RuleSet-based quantum networking protocol and the data link protocol, we use for the network simulation.

A. RuleSet

Coordinated operation is a prerequisite functionality for quantum networking over both the link layer and the network layer [5]. When nodes perform a particular operation using pre-shared entangled resources, participants are assumed to perform the correct operation targeting the appropriate resources. Such consistency should not be achieved by exchanging messages with each other, especially over long distances due to the latency incurred. In this section, we introduce the concept of *RuleSet* for supporting quantum networking, which allows us to synchronize operations over a network with minimal classical message transmission. If a single connection follows a route involving n nodes, the source node requests the destination node to generate n RuleSets, and distributes them to all n nodes, respectively (see Fig. 5) [29].

A RuleSet is an object consisting of one or more *Rules*, each holding a *Condition* and an *Action*. A Condition may have one or more *Clauses*, each of which is a conditional statement. An Action holds a list of operations that a node needs to perform in order to accomplish a single task, which may be, for example, entanglement swapping, purification, measurement for tomography, etc., and is invoked if and only if each Clause in the corresponding Condition is fulfilled. When the task involves at least one resource, the oldest available resource will be picked from the allocated set. An Action may also generate a message to another node, lock, and reinitialize resources. In our current implementation, a RuleSet also has a *Termination Condition*, which is a simple counter for determining when to discard the RuleSet and discontinue the connection. This may also have multiple Clauses.

RuleSets will be managed and executed by the RuleEngine, which is a software module installed in all quantum nodes that is responsible for interpreting the RuleSet instructions, and executing them in real time. The RuleEngine is event

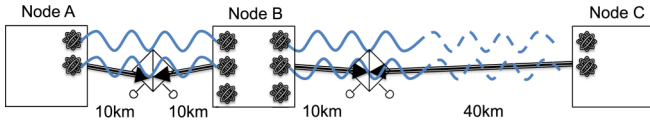


FIG. 6. Bias in knowledge regarding the state of resources. Node C receives the success or failure acknowledgment later than Node A and B.

driven, where the event may be a classical packet arrival, new resource allocation or RuleSet timeout. If a single RuleSet consists of multiple Rules, the RuleEngine refers to each Rule based on top-down strategy, invoking Actions from top to bottom sequentially. That is, if a node completes the action in the first Rule on qubit A, if any, qubit A will be reassigned for the use of the second Rule in the same RuleSet. Therefore, newly generated resources are first assigned to the first Rule in the set. They must complete the first rule before being reassigned to the second Rule. Timeout may be performed by tracking how well resources are upgraded to the upper Rules. This RuleSet level resource allocation can be directly translated into multiplexing schemes, such as buffer space multiplexing [39]. This top-down approach is also a simple solution to overcome the bias in knowledge regarding the resources across a network (see Fig. 6).

All RuleSets for the same connection share the same RuleSet identifier (*RuleSetID*). The RuleSet identifier is generated via a hash function using the time when the RuleSet was generated, the IP address of the node which generated the RuleSet, and a random number as a seed to avoid global conflicts. Rules are also indexed (*RuleID*) inside the RuleSet. Each action holds a counter (*ActionIndex*), where the counter is incremented whenever the action has been performed. These identifiers are used, for example, when a node needs to share the measurement result with another node for link-level tomography. Whenever an Action of a RuleSet requires a node to transmit a classical packet to another node, such as the measurement result, it encloses the identifiers in the packet, so that the receiving node can uniquely identify and pair its own measurement result with the received one accordingly.

An example structure of a RuleSet composed of two Rules is shown in Fig. 7. The pseudocode of a Rule is provided in Sec. IV.

B. Data link protocol

The core concept of the data link protocols introduced in [34] will be kept untouched. For simplicity, in this paper, we only consider two models, MeetInTheMiddle and SenderReceiver. MidpointSource remains as future work. As in [34], we do not elaborate how to synchronize clocks across a network, but will start with the coordination of photon emissions between adjacent nodes.

We add some supplemental components for the network simulation. Coordinating the arrival time of photons at the BSA via messaging between the two nodes with memory is difficult. Instead, we make the BSA node responsible for coordinating the entanglement generation. To start, an

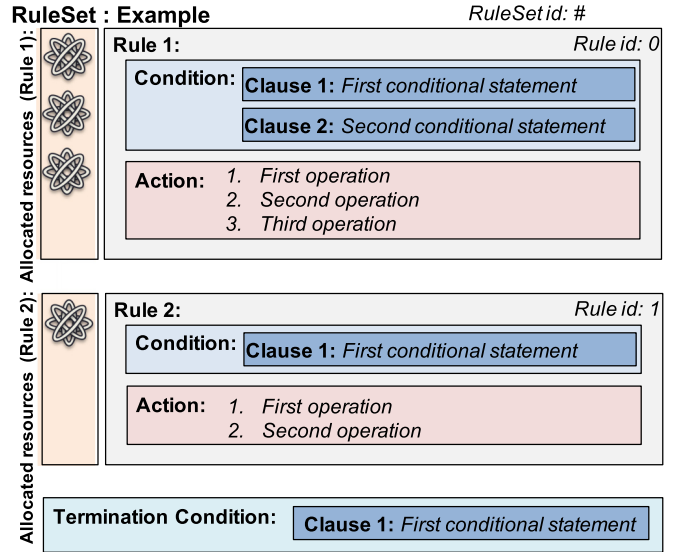


FIG. 7. An example structure of a RuleSet. The resource allocated in the second Rule has been already passed through the first.

activated quantum node classically transmits a *Boot Up Notification* to all neighboring BSA nodes. For a SenderReceiver link, the sender transmits the message towards the receiver. Once received, the BSA node calculates the photon emission timings and the corresponding burst rate for neighboring nodes, based on the quantum channel lengths and its own single photon detector recovery time. Such information will be classically forwarded to the neighboring quantum nodes to start the entanglement generation. The quantum nodes also need to classically notify the BSA node of the end of the burst. Reception triggers the BSA to recalculate the emission timing for the next round, and returns the information together with a list of *success or failure for transmission i*. We buffer the success or failure results, and send them as a single packet to prevent overflowing the classical channel. As in Fig. 8, each end of a classical channel is connected to a network interface card (NIC), and each end of a quantum channel is connected to a quantum network interface card (QNIC), which holds the memory qubits. Optical qubits arrive at the beamsplitter from QNICs via quantum channels, and success or failure results will be accumulated by the BSA controller, which

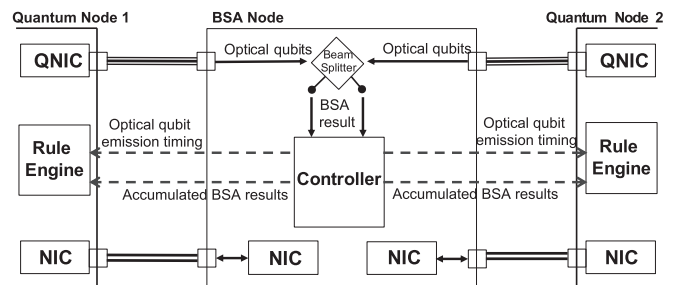


FIG. 8. BSA node architecture. Classical communications, shown as dashed arrows, will be sent through the classical channel via NICs.

also controls the packet transmissions. For SenderReceiver links, the BSA module is implemented inside the receiver’s QNIC.

IV. ANALYSIS

In this section, we study the capability of a first generation quantum repeater link, with a limited amount of resources, and under noisy conditions generated by various hardware imperfections. Simulated error sources include quantum memories, quantum channels, quantum gates, and single photon detector dark counts. Quantum memories and quantum channels include Pauli errors as well as loss of photons and carriers. Our Markov-chain Monte Carlo simulation uses similar parameters as in [40] but with larger memory buffers (see Table I for details). We use the memoryless Markov chain in the simulation to dynamically model errors on memory qubits independently, based on a given lifetime T_1 [41] (for details, see Appendix). Other error sources, such as the gate error, are simulated via a random selection with static probabilities. In order to reduce the computation time, we only propagate Pauli errors through circuits as in [42,43]. The simulated purification circuit, therefore, is incapable of stochastically detecting excited, relaxed, and completely mixed errors. Hence, our simulation generates a pessimistic output fidelity, and an optimistic output resource generation rate. To concretely identify the real impact of imperfect quantum systems, we assume ideal classical communication channels. Because classical communication latencies cover the majority of the simulated time, we assume negligible gate times. Furthermore, multiqubit operations can also be performed between arbitrary qubits across QNICs within the same node. For each data point in the figures, a total of 25 simulations has been performed to quantify the average behavior.

A. Performing simple tomography with different numbers of measurements

Performing quantum state tomography with more measurements typically results in a greater accuracy of the reconstructed density matrix. Although the required accuracy may depend on the demanded precision of a particular application, it is unlikely to perform an extremely large set of measurements solely for the reconstruction for purposes such as real time quantum channel monitoring. We first investigate how the number of measurements for the tomography N_M impacts the accuracy of the *reconstructed fidelity* F_r between adjacent nodes. Here, $F_r = \text{Tr}[\rho_r \rho_i]$, where ρ_i is the ideal density matrix of a Bell pair $|\Phi^+\rangle$ and ρ_r is the density matrix reconstructed through the RuleSet-based quantum link-level tomography. This particular RuleSet consists of a single Rule composed of two Clauses and an Action. The first Clause, *MeasurementConditionClause*, tracks how many measurements have been performed. The second Clause, *ResourceConditionClause*, checks the availability of resources (see Algorithm 1). The Action performs the measurement for the link-level tomography (see Algorithm 2). In this case, the Termination Condition also tracks the number of performed measurements, and stops the execution when the requirement is fulfilled.

Algorithm 1 ResourceConditionClause

This Clause checks if enough resources are available for the corresponding Action.
 Input: resourceList \leftarrow List of allocated resources for the Rule. A resource may be locked when it is waiting for the classical packet to arrive, such as for purification.
 Output: enoughResources \leftarrow A boolean value.

```

1: procedure RESOURCECONDITIONCLAUSE(resourceList)
2:   numRequired  $\leftarrow$  Number of required resources for the
   Action
3:   numFree = 0
4:   enoughResources = false
5:   for each resource in resourceList do
6:     if resource is not locked then
7:       numFree++
8:     end if
9:     if numFree  $\geq$  numRequired then
10:      enoughResources = true
11:      break
12:     end if
13:   end for
14:   return enoughResources
15: end procedure

```

Algorithm 2 TomographyAction

This Action performs measurement on an entangled qubit. The measurement result and the selected basis are stored, and sent to the partner node as a message.
 Input: resourceList \leftarrow List of allocated resources for the Rule.
 Output: msg \leftarrow A message for another node.

Require: ResourceConditionClause == true && MeasurementConditionClause == true

```

1: Procedure TOMOGRAPHYACTION(resourceList)
2:   self_addr  $\leftarrow$  Node address of this RuleSet owner.
3:   partner_addr  $\leftarrow$  Node address of tomography partner.
4:   resource = Select a free resource from resourceList
5:   basis = RandomBasisSelect({X,Y,Z})
6:   outcome = resource.Measure(basis)
7:   removeResourceFromList(resourceList,resource)
8:   Data.result = {outcome, basis}
9:   Data.ruleSetId = this.RuleSetID
10:  Data.ruleId = this.RuleID
11:  Data.actionIndex = this.ActionIndex
12:  save(Data)
13:  msg.destination = partner_addr
14:  msg.source = self_addr
15:  msg.data = Data
16:  this.ActionIndex++
17:  return msg
18: end procedure

```

The simulated result of a repeater network composed of two nodes based on a MeetInTheMiddle link, each 10 km from the BSA node ($L = 20$ km), is shown in Fig. 9. Error bars represent one standard deviation uncertainty (σ) from the average F_r . Max F_r and Min F_r are the highest and lowest fidelity observed within the 25 trials. Purple data points

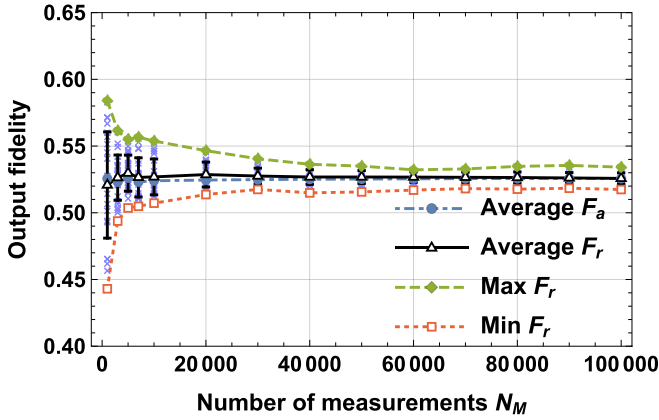


FIG. 9. Impact of the number of measurements on the reconstructed fidelity error over the MeetInTheMiddle link. F_r is the reconstructed fidelity and F_a is the actual fidelity.

marked with cross between the minimum and the maximum are individual simulation results.

As in Fig. 9, the reconstructed fidelity has poor accuracy with only 1000 measurements, and slightly underestimates the value simply due to the lack of samples— $\sigma \approx 0.040$. In order to achieve $\sigma < 0.015$, at least 7000 measurement shots are required. A total of 20 000 measurements is capable of reconstructing the fidelity with a standard deviation less than 1%, and roughly converges to $\sigma \approx 0.004$ when $N_M > 60\,000$.

F_a is the actual fidelity, accessible to us because this is a simulation, but of course hidden in the real world. The change in the accuracy of the reconstructed fidelity compared to the actual fidelity is shown in Fig. 10. The average difference between the reconstructed fidelity and the actual fidelity for a given N_M is

$$\bar{F}_{|r-a|} = \frac{1}{N} \sum_{i=1}^N |F_{r(i)} - F_{a(i)}|, \quad (1)$$

where $N = 25$ is the total number of simulation trials. The plot shows an upper bound of the 1-norm distance between two mixed states.

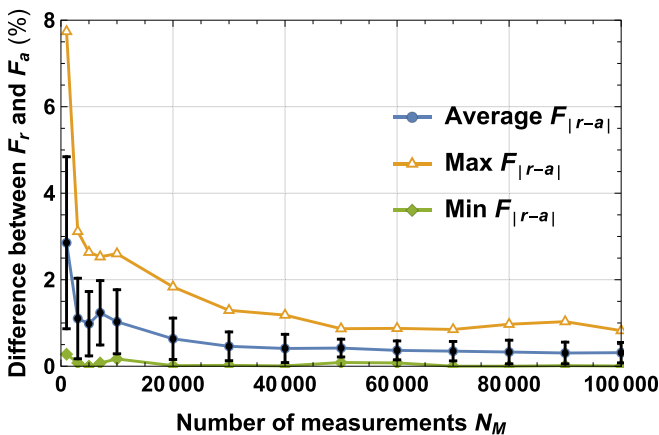


FIG. 10. Change in the accuracy of the reconstructed fidelity relative to the actual fidelity.

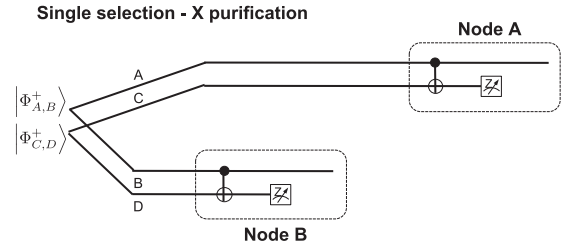


FIG. 11. Single selection–Single error (X) purification (Ss-Sp). Consumes a single Bell pair $|\Psi_{C,D}^+\rangle$ to detect the X error on $|\Psi_{A,B}^+\rangle$.

The fidelity reconstruction process using only 1000 measurement outcomes will result in about 3% error from the actual value on average. The accuracy improvement converges to $\Delta F_{|r-a|} > 0.3\%$ when $N_M \geq 40\,000$, in which case not much benefit can be gained from solely increasing the number of measurements.

The MeetInTheMiddle link and the SenderReceiver link have similar behavior regarding the fidelity, with sufficiently long memory lifetime relative to the idle time caused by classical latencies. Nevertheless, even over a single hop, the throughput strongly depends on the physical link architecture and the adopted data link protocol. Accumulating more measurements will increase the tomography time linearly, for both links, as the operation here consists only of measuring entangled qubits sequentially. The entanglement generation rate over the MeetInTheMiddle link is twice what the SenderReceiver link achieves, simply because the BSA in the MeetInTheMiddle sits half-way through the link. With the given parameter settings, architecture, and protocols, the MeetInTheMiddle link’s Bell pair generation rate is around 6741/s (3368/s for the SenderReceiver link), when qubits are immediately measured when the system recognizes that they are entangled.

B. Performing tomography with single shot purification over different distances

Lengthening the channel between nodes increases the chance for optical qubits and memory qubits to experience errors. One way to tolerate such errors is to perform quantum purification.

The simplest purification scheme is the Single selection–Single error purification (Ss-Sp) [14,35]. As in Fig. 11, the Ss-Sp requires two Bell pairs along the channel to be purified—one Bell pair is consumed to detect the presence of an X error (or Z error) on the other Bell pair. The resource is only purified successfully when the measurement outcomes of the consumed qubits (qubit C and qubit D in Fig. 11) coincide. Otherwise, both nodes discard their resources.

By extending Ss-Sp, we can also try to purify both errors (see Fig. 12). The Single selection–Double error purification (Ss-Dp) is similar to Ss-Sp, but requires another Bell pair to detect the Z error on the resource. However, because we perform the Z error purification after the X, the X error can propagate to the purified resource from the consumed Bell pair. In Fig. 11, the X error from qubit E and qubit F will propagate to qubit A and qubit B through the CNOT gate accordingly.

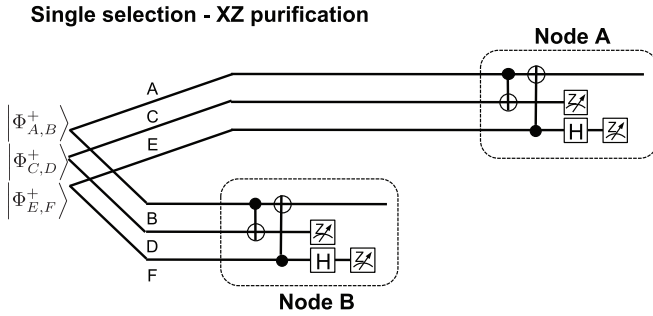


FIG. 12. Single selection–Double error (XZ) purification (Ss-Dp). Consumes two Bell pairs, $|\Psi_{C,D}^+\rangle$ and $|\Psi_{E,F}^+\rangle$ to detect X error and Z error accordingly.

The double selection method introduced by Fujii and Yamamoto in 2009 [35] is a purification method utilizing an additional resource to improve the accuracy of post-selection by applying double verification. The Double selection–Single error purification (Ds-Sp) does the same job as Ss-Sp, but with an additional resource to detect the presence of a Z error on the consumed resource used for the X error purification (see Fig. 13). This minimizes the error propagation to the purified resource. Notice that any Z error that is originally present on the purified resource will not be detected.

The double selection scheme can also be applied to the double error purification circuit (Ds-Dp). The concept of this circuit is similar to Ds-Sp, except we perform the double verification process on both X and Z purification to the resource we keep. Thus, we can also minimize the error propagations that happen during the purification. For the circuit diagram, see Fig. 14.

In this subsection, we describe simulated link-level tomography with 7000 measurements, with and without Rules that perform one of the purification methods above beforehand, over different distances. The simulated results of the fidelity reconstruction is shown in Figs. 15 and 16. The fidelity for the SenderReceiver link is slightly worse than that of the MeetInTheMiddle link, but the overall behavior is approximately the same.

As shown by the results in Fig. 15, performing Ss-Sp once slightly improves the fidelity compared to the case without, only over relatively shorter distances, but with a significant

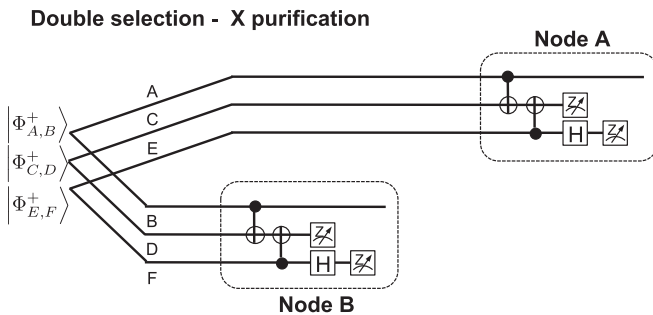


FIG. 13. Double selection–Single error (X) purification (Ds-Sp). Consumes a single Bell pair $|\Psi_{C,D}^+\rangle$ to detect the X error on $|\Psi_{A,B}^+\rangle$, and another Bell pair $|\Psi_{E,F}^+\rangle$ to avoid the Z error propagation from $|\Psi_{C,D}^+\rangle$ to $|\Psi_{A,B}^+\rangle$.

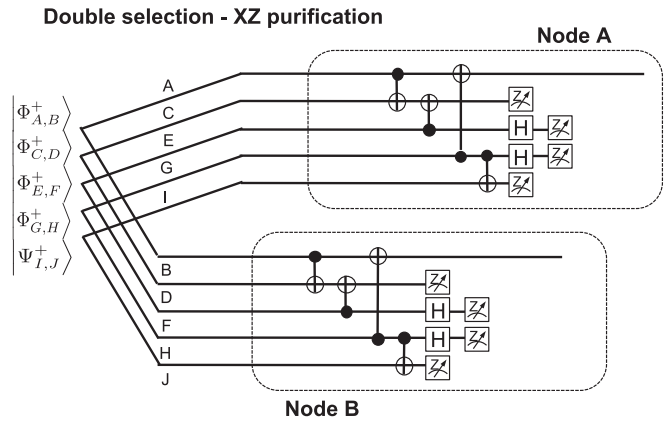


FIG. 14. Double selection–Double error (XZ) purification (Ds-Dp). Consumes two Bell pairs, $|\Psi_{C,D}^+\rangle$ and $|\Psi_{G,H}^+\rangle$, to detect X error and Z error on $|\Psi_{A,B}^+\rangle$. Each consumed Bell pair will also be verified through purification using $|\Psi_{E,F}^+\rangle$ and $|\Psi_{I,J}^+\rangle$ accordingly.

penalty on the throughput for both link architectures (see Fig. 16). With a total distance $L = 10$ km, Ss-Sp suppresses about 5.6% of the X errors and 5.4% of Y errors, but the Z error rate increases approximately 8.8% due to the error propagation through the purification circuit. Overall, the fidelity improves roughly by 3%. While the fidelity did not change dramatically, the error distribution developed significantly different after performing the operation, which we can still take advantage of in the recurrence purification protocol (see Sec. IV C). By implementing double selection to the same circuit (Ds-Sp), the output fidelity improved significantly—about 6.4% increase from the case without performing any purification. The second verification decreased the Z error rate by about 4.6% compared to the case of Ss-Sp with $L = 10$ km. The Ss-Dp protocol purifies the X error first, and then the Z error afterwards. In this case, over $L = 10$ km distance, the Z error rate decreases by 3.3% and the Y error rate by 5.7%—the X error rate increases by 4.3% due to propagation. Simply

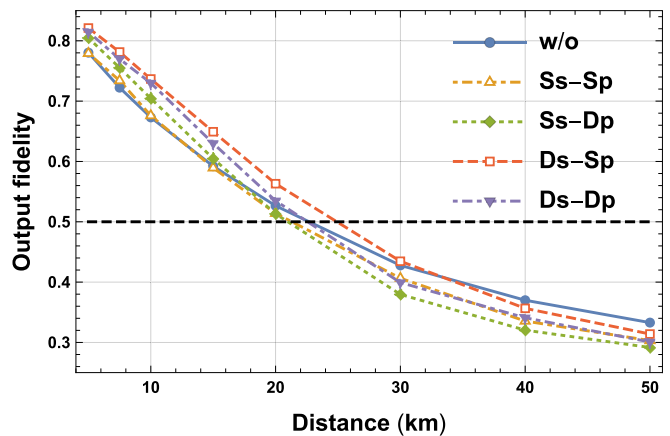


FIG. 15. Impact of the MeetInTheMiddle channel length on the reconstructed fidelity with 7000 measurement outcomes w/o, with purification. The steep drop in the fidelity is due to the high error rate of 1% each X, Y, and Z errors per km in the quantum channel chosen for this simulation.

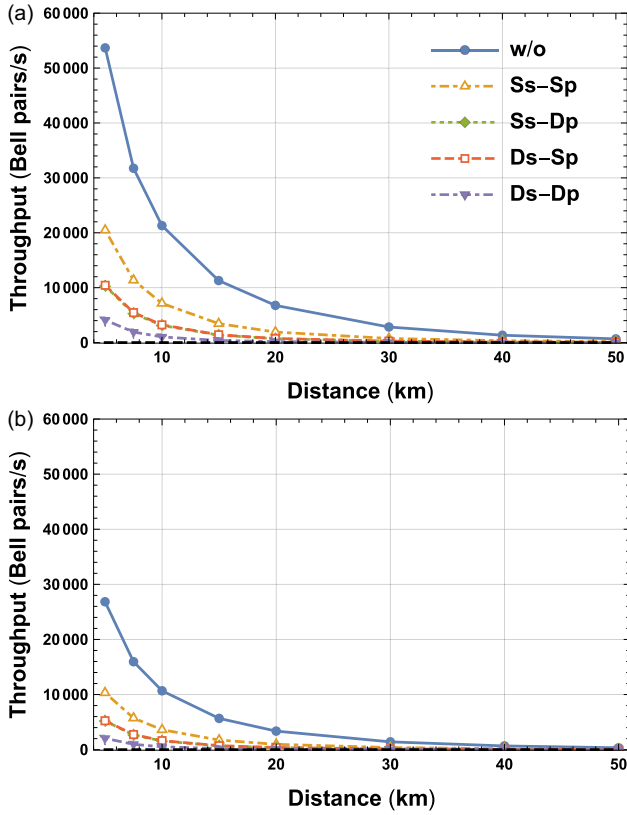


FIG. 16. Impact of the channel distance on the channel throughput with and without performing purification. The throughput has been calculated as N_M/T , where T is the tomography process time to complete N_M measurements. (a) Throughput over the MeetInTheMiddle link. (b) Throughput over the SenderReceiver link.

applying double selection to this (Ds-Dp) improved the fidelity by approximately 2.6% from Ss-Dp. Both purifications based on double selection are capable of purifying resources under longer distances, which is roughly 20 km.

With the given hardware qualities in Table I, Ds-Sp obtains the highest output fidelity. The output fidelity for Ds-Dp is lower than that of Ds-Sp, mainly because of larger KQ with noisy gates, and the increase waiting time for resources, especially over long distances, where KQ is the product of the circuit depth (K) and the number of qubits (Q) [52]. Under this scenario, because Ds-Sp only requires three resources at once, the throughput is also higher than that of Ds-Dp.

C. Bootstrapping using RuleSet-based recurrence purification and tomography

Network operations, such as routing, often require knowledge regarding the links, e.g., fidelity, throughput, etc. Therefore, the channel characteristics to be shared need to be acquired beforehand via quantum link bootstrapping and characterization. In quantum networking, we commonly focus on optimizing the link fidelity for realizing a robust connection, where the goal may be achieved by performing nested

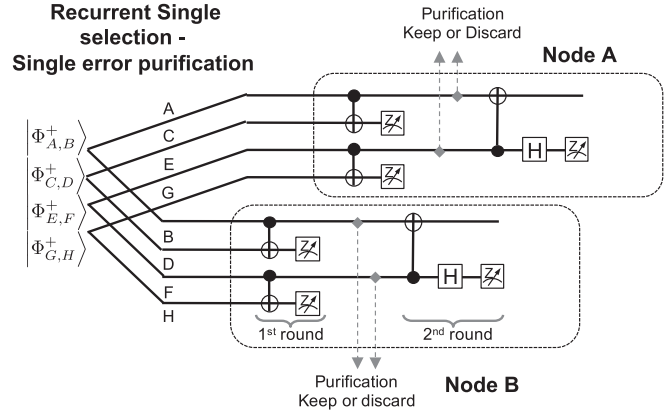


FIG. 17. Recurrence purification protocol based on single selection X purification and Z purification. Each round of purification is a Rule in a RuleSet.

purifications. The recurrence protocol [32] performs quantum purification on top of pre-purified resources to effectively improve the fidelity—the required number of resources for the operation, on the other hand, increases exponentially in the number of purification rounds N_p , which alone linearly affects the total idle time due to classical communication latencies.

In this subsection, we describe our simulation of quantum link bootstrapping to quantify the achievable link fidelity and its corresponding throughput, using RuleSet-supported tomography with recurrence purification based on one of the four circuits shown in Sec. IV B; recurrent single selection single error purification (RSs-Sp), recurrent single selection double error purification (RSs-Dp), recurrent double selection single error purification (RDS-Sp), and recurrent double selection double error purification (RDS-Dp). Each round of purification is a separate Rule in the same RuleSet. In the given example of a two-round RSs-Sp circuit shown in Fig. 17, the first Rule performs X purifications to resources. In the second Rule, we use those purified resources to perform another Z error purification. Each round, we alternate the X purification and the Z purification to suppress the error propagation. For double error purifications, we alternate the XZ purification and the ZX purification. With probabilistic generation of Bell pairs and scarce memory qubits, operations can stall in the middle, in which case the Rule waits for new resources to arrive. Using RuleSets, the whole task is completed autonomously over a network. The simulation continues reconstructing the density matrix every round, and terminates the process when the same operation results in a lower reconstructed fidelity, or due to timeout. For practical reasons, each RuleSet timeout is set to two minutes.

We assume perfect photon emissions from memories to fiber, where in the real world, similar effects may be observed by increasing the number of memory qubits. CNOT gates are also assumed to be ideal to solidify the real merit of recurrence protocols. We first focus on a repeater network with a relatively short channel, where the total length is set to 10 km. Secondly, we perform similar simulations over a longer distance, $L = 20$ km. For the MeetInTheMiddle link, each node is the same distance away from the BSA node.

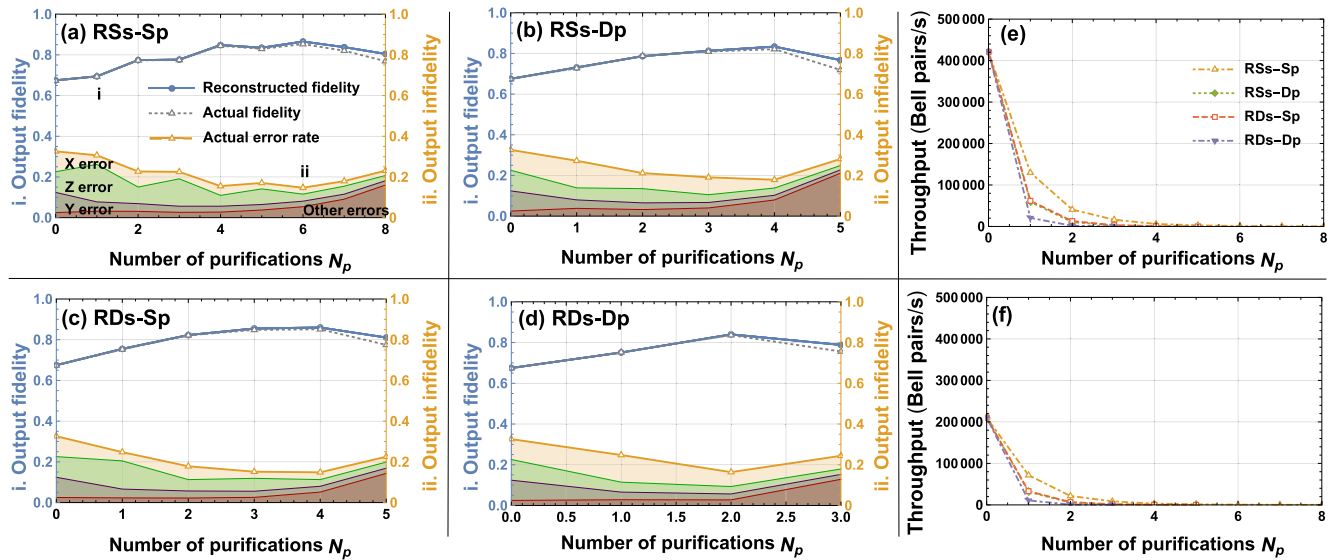


FIG. 18. Reconstructed fidelity with actual fidelity and actual error rates, and estimated throughput of protocols over $L = 10$ km MeetInTheMiddle link. "Other errors" include memory excitation or relaxation error, and completely mixed error due to photon detector dark counts. (a) Simulation result of the RSs-Sp protocol. (b) Simulation result of the RSs-Dp protocol. (c) Simulation result of the RDs-Sp protocol. (d) Simulation result of the RDs-Dp protocol. (e) Protocol throughputs over the MeetInTheMiddle link. (f) Protocol throughputs over the SenderReceiver link.

1. Closely spaced repeater nodes

We start by investigating how recurrent purification benefits over a link between closely located repeater nodes, $L = 10$ km. The simulated result over the MeetInTheMiddle link is shown in Fig. 18. Notice that each protocol has a different number of purification rounds performed.

As in the figure, the given system generates resources with an average fidelity $F_r \approx 0.675$, but the RDs-Dp is capable of bringing up the quality up to $F_r \approx 0.840$ through a two-round purification, which is the fastest in terms of required rounds. The RSs-Sp, on the other hand, starts with the slightest fidelity improvement since applying X purification inherently results in a significant penalty to the Z error rate, as we discussed in Sec. IV B. While this looks operationally impractical, by alternating X and Z purification multiple rounds, we can purify resources while minimizing error propagations—this appears as the stair-step-like curve in Fig. 18(a). RSs-Sp gradually improves the fidelity as we perform more purifications, and produces resources with optimal average fidelity, $F_r \approx 0.865$ ($F_r \approx 0.852$ for the SenderReceiver link) at $N_p = 6$. About 3.1% of the errors are X errors, 2.7% are Y errors, and 3.4% are Z errors; 5.6% of the outputs consist of unfixable errors, which can mainly be avoided by improving the memory lifetime, shortening the distance between nodes, or adjusting the algorithm and the goal. RDs-Sp is also capable of producing high fidelity resources, $F_r \approx 0.860$, with more rounds of purifications than RDs-Dp but fewer than RSs-Sp. The fidelity starts declining for all protocols in the end, due to critical waiting time of memories relative to its own lifetime. The reconstructed fidelity also gets overestimated, because the randomness of measuring unentangled qubits due to errors, stochastically contributes to the fidelity estimation.

The number of rounds does not necessarily determine how fast the protocol processes resources, especially with a

limited number of qubits. As shown in Fig. 18, the RDs-Dp protocol over the MeetInTheMiddle link reaches its highest fidelity with a throughput of roughly 1565/s, completing 7000 purified Bell pair measurements in around 37.1 s. The RSs-Sp protocol produces 1106 resources per second with the local optimal fidelity, but also achieves similar fidelity as RDs-Dp, $F_r \approx 0.848$, at $N_p = 4$, which obtains a higher generation rate of 5923/s. The RDs-Sp protocol and the RSs-Dp protocol have a throughput of 816/s and 520/s, respectively. Although both protocols consume the same amount of resources per purification, the RSs-Dp took longer in total because at each step, RSs-Dp faces higher error propagation probability, resulting in greater purification failure rate. As provided in Figs. 18(e) and 18(f), the throughput over the SenderReceiver link is approximately half of what the MeetInTheMiddle link achieves. In this case, RSs-Dp shows no advantage over RDs-Sp.

The quality of a link can also be increased by installing larger sets of memory qubits. As in Fig. 19, with 700 memory qubits, a six-round RSs-Sp can generate resources with $F_r \approx 0.879$. Compared to the case with only 100 qubits, the fidelity at $N_p = 4$ also increases by around 0.6%. For $N_p < 4$, the fidelity stays unchanged because having 100 qubits is more than sufficient for a four-round RSs-Sp—the throughput will, however, improve. Notice that a linear increase in memory buffer size will not provide an effective solution to extend the performable rounds. Over a 10-km MeetInTheMiddle link, owning 700 memory qubits is still not enough for an eight-round RSs-Sp.

2. Distantly spaced repeater nodes

Errors on qubits have more time to develop when two repeater nodes are located farther away. Hence, how well errors are detected at each round, especially with noisy

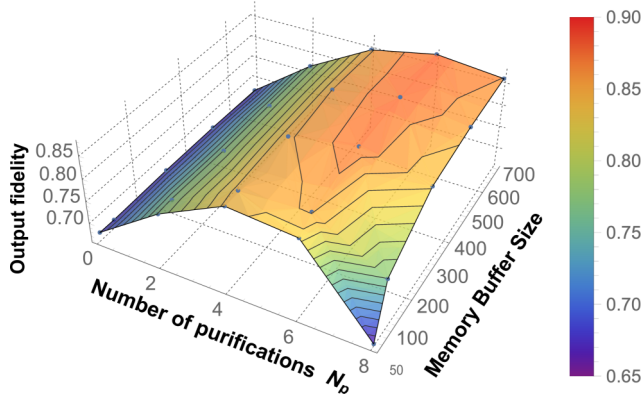


FIG. 19. Impact of the memory buffer size to the maximum fidelity purified through the RSs-Sp protocol over a 10-km MeetInTheMiddle link.

inputs, becomes an important factor for an effective recurrence purification protocol. The simulation results over a repeater network with channel length $L = 20$ km are provided in Fig. 20.

Unlike the case of Fig. 18, neither RSs-Sp nor RSs-Dp improves the fidelity effectively, regardless of the number of purification rounds. As shown in the corresponding error distributions, performing purifications with the given input resources, in such a way done by RSs-Sp and RSs-Dp, ends up failing because the loss lead by error propagations and memory errors is greater than the gain. Resources used for RSs-Sp suffer more from memory errors as they go through more rounds of purifications, resulting in longer average idle time per resource. The other two protocols implemented with double selection, however, are capable of effectively purifying resources. The RDs-Sp protocol produces the highest fidelity among all, with an output value $F_r \approx 0.667$ and a rate of

325/s (168/s with $F_r \approx 0.651$ for the SenderReceiver link) at $N_p = 3$, whereas RDs-Dp also produces similar quality, $F_r \approx 0.637$, or $F_r \approx 0.626$ for the SenderReceiver link, within two rounds. The corresponding throughput is 116/s for the MeetInTheMiddle link, and 60/s for the SenderReceiver link.

Under this scenario, consuming another Bell pair via the double selection shows advantage in terms of fidelity. However, from the previous discussion, we know that RSs-Sp generates resources with higher fidelity and higher rate, given slightly better average input resources and sufficiently long-lived memories relative to its average idle time. Thus, for longer distances, we can also perform double selection-based purification at the beginning to raise the fidelity at a certain level, then change to single selection purification afterwards to maximize the fidelity. Below in Fig. 21 are the simulation results of two cases; Case A that switches to RSs-Sp after performing a single round of RDs-Sp, and Case B that switches to RSs-Sp after performing two rounds of RDs-Sp.

Whether this method is beneficial or not very much depends on the situation, and the switching timing as shown. Under this scenario, Case A produces higher fidelity, $F_r = 0.680$, with a higher rate of 396 Bell pairs per second, relative to the optimal case of RDs-Sp in Fig. 20. Case B results in worsening the fidelity to $F_r = 0.666$, and the throughput to 218/s.

V. CONCLUSION

In this paper, we introduced our RuleSet-based communication protocol. The RuleSet-based system is a key element of the network architecture, and supports the operational coordination among distant nodes. Nodes are capable of dynamically generating Rules for a particular connection, which provides us flexibility when establishing a connection compared to predefined hardcoded procedures—how much to purify and where to perform entanglement

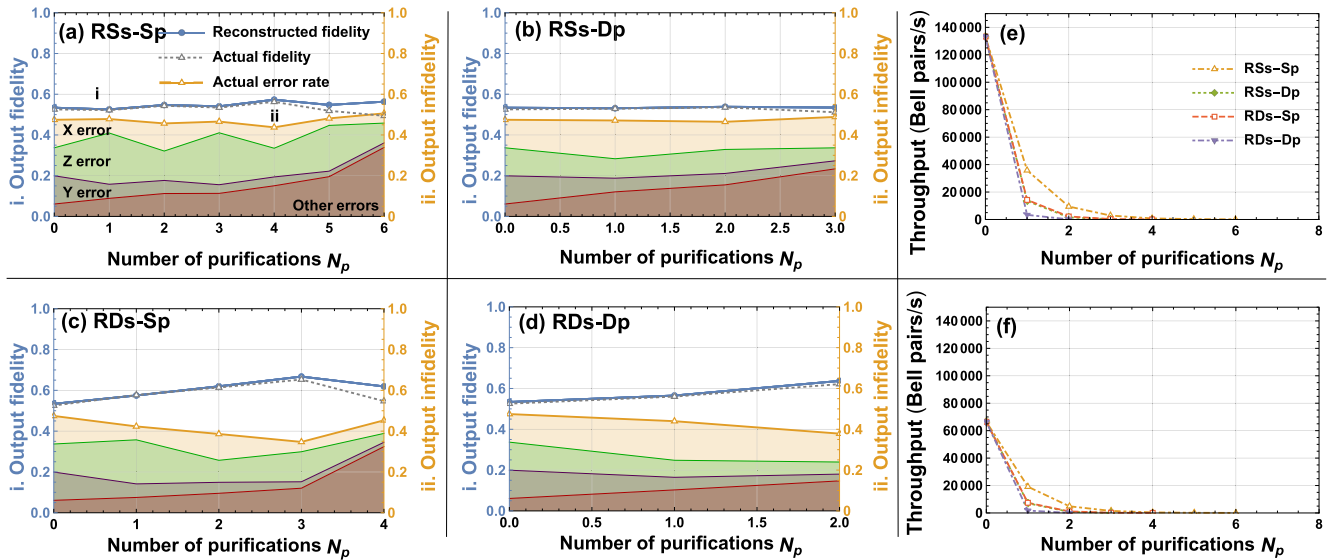


FIG. 20. Reconstructed fidelity with actual fidelity and actual error rates, and estimated throughput of protocols over $L = 20$ km MeetInTheMiddle link. "Other errors" include memory excitation or relaxation error, and completely mixed error due to photon detector dark counts. (a) Simulation result of the RSs-Sp protocol. (b) Simulation result of the RSs-Dp protocol. (c) Simulation result of the RDs-Sp protocol. (d) Simulation result of the RDs-Dp protocol. (e) Protocol throughputs over the MeetInTheMiddle link. (f) Protocol throughputs over the SenderReceiver link.

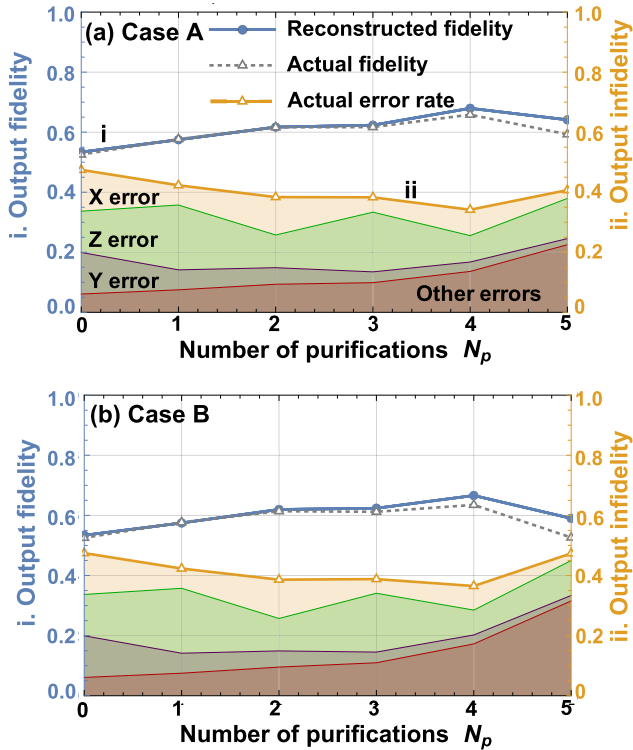


FIG. 21. Switching RDs-Sp to RSs-Sp over $L = 20$ km. (a) Pattern 1 consists of one round RDs-Sp and four rounds of RSs-Sp. Purification methodology switches at $N_p = 1$. (b) Pattern 2 consists of two rounds RDs-Sp and three rounds of RSs-Sp. Purification methodology switches at $N_p = 2$.

swapping. We adapted the technology to conduct quantum link bootstrapping in a network simulation. In total, we ran four different recurrent purification schemes to quantify the achievable link fidelity with its corresponding throughput via link-level tomography, using Markov-chain Monte Carlo simulation with a noisy quantum repeater system. Such information may be distributed across the network, and used for different purposes such as quantum routing. Notice that the RuleSets generation algorithm depends on the firmware installed in the destination node. Therefore, other operations such as the entanglement swapping and error correction, may also be implemented using the RuleSet-based communication protocol for connecting multihop nodes.

The RSs-Sp requires more purification rounds to optimize the fidelity compared to other schemes, which results in a longer average idle time for purified resources. Given 100 link memory qubits each with a 50-ms lifetime, we found that RSs-Sp is still stronger than other simulated schemes

in terms of the achievable fidelity and the throughput, when the distance between nodes is kept short ($L \approx 10$ km). In contrast, because the system is noisier with longer channels ($L \approx 20$ km), RSs-Sp alone may be incapable of purifying the link, as errors can evolve faster than the purification, in which case schemes such as the double selection become more preferable. Our simulations show that the bootstrapping process, therefore, must be able to search and choose the most effective purification method for that specific channel. Changing from double selection to single selection in the middle indeed pushed the link limit, but finding the optimal changeover timing may add an additional complexity to the process. Other methods that detect errors with a higher probability requiring fewer rounds, such as RDs-Dp, may be more advantageous when a larger set of memory qubits is available for each link, relative to its demand.

The purification procedure is not unique to this paper. What is more important is the missing formal definition of the communication decision-making process, which we tackle using the RuleSet-based communication protocols. The focus is not only the output fidelity itself, but also the control process, the network architecture, and engineering. We showed that our RuleSet-based system is capable of running different operations, and even use that to compare different processes.

While this work focuses on link fidelity optimization, connecting source to destination multihops away also requires an adequately high generation rate, or possibly a synchronized generation timing of Bell pairs. Such an optimization problem of the fidelity-throughput tradeoff remains an open question. In the future, we will extend the simulation to a higher layer, simulating with routing policies over a complex network.

ACKNOWLEDGMENTS

We thank Takahiko Satoh for valuable discussions. This work was supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JPMXS0118067285.

APPENDIX: MEMORY ERROR SIMULATION

We use a row vector to describe the present state of a qubit, which is one of the seven distinguishable states—no error, X error, Z error, Y error, excited, relaxed, or completely mixed. For example, a Bell pair with no error as an initial input state can be described as in Eq. (A1).

$$\vec{\pi}(0) = (\text{Clean} \quad \text{Xerror} \quad \text{Zerror} \quad \text{Yerror} \quad \text{Excited} \quad \text{Relaxed} \quad \text{Mixed}) = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0). \quad (\text{A1})$$

Accordingly, the infinitesimal generator for our continuous-time Markov chain [53] we use for the memory error simulation also consists of seven states,

$$\bar{Q} = \begin{pmatrix} \text{Clean} & \text{Xerror} & \text{Zerror} & \text{Yerror} & \text{Excited} & \text{Relaxed} & \text{Mixed} \\ (1 - \Sigma_0) & P_X & P_Z & P_Y & P_E & P_R & 0 \\ P_X & (1 - \Sigma_1) & P_Y & P_Z & P_E & P_R & 0 \\ P_Z & P_Y & (1 - \Sigma_2) & P_X & P_E & P_R & 0 \\ P_Y & P_Z & P_X & (1 - \Sigma_3) & P_E & P_R & 0 \\ 0 & 0 & 0 & 0 & (1 - \Sigma_4) & P_R & 0 \\ 0 & 0 & 0 & 0 & P_E & (1 - \Sigma_5) & 0 \\ 0 & 0 & 0 & 0 & P_E & P_R & (1 - \Sigma_6) \end{pmatrix}. \quad (\text{A2})$$

Element P_X , P_Y , and P_Z correspond to Pauli X , Y , and Z error accordingly. P_E and P_R are memory excitation and relaxation errors, which are used to represent T_1 time and the Boltzmann thermal distribution. Σ_i is the sum of every other element in row i . Mixed state occurs due to dark counts, or when the paired qubit gets excited or relaxed. The probability distribution of a qubit output state after time t can be

found by

$$\vec{\pi}(t) = \vec{\pi}(0)\vec{Q}^t. \quad (\text{A3})$$

Each element in the output vector $\vec{\pi}(t)$ represents the probability for the qubit being in the corresponding state. The qubit state can be determined via a random selection based on the output probability distribution, and the density matrix is constructed on demand.

-
- [1] H. J. Kimble, *Nature (London)* **453**, 1023 (2008).
- [2] R. Van Meter, *Quantum Networking*, 1st ed. (Wiley-IEEE Press, New York, 2014).
- [3] S. Wehner, D. Elkouss, and R. Hanson, *Science* **362**, eaam9288 (2018).
- [4] W. Kozłowski, *Architectural Principles for a Quantum Internet*, Internet draft draft-irtf-qirg-principles-00 (Internet Engineering Task Force, Fremont, 2019), work in progress, expires September 10, 2019.
- [5] H. Zimmermann, *IEEE Trans. Commun.* **28**, 425 (1980).
- [6] C. H. Bennett and G. Brassard, *Theor. Comput. Sci.* **560**, 7 (2014).
- [7] C. H. Bennett, G. Brassard, and N. D. Mermin, *Phys. Rev. Lett.* **68**, 557 (1992).
- [8] A. K. Ekert, *Phys. Rev. Lett.* **67**, 661 (1991).
- [9] M. Takeoka, S. Guha, and M. M. Wilde, *Nat. Commun.* **5**, 5235 (2014).
- [10] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams, *Phys. Rev. Lett.* **85**, 2010 (2000).
- [11] I. L. Chuang, *Phys. Rev. Lett.* **85**, 2006 (2000).
- [12] A. Broadbent, J. Fitzsimons, and E. Kashefi, Measurement-based and universal blind quantum computation, in *Formal Methods for Quantitative Aspects of Programming Languages: 10th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2010, Bertinoro, Italy, June 21–26, 2010, Advanced Lectures*, edited by A. Aldini, M. Bernardo, A. Di Pierro, and H. Wiklicky (Springer, Berlin/Heidelberg, 2010), pp. 43–86.
- [13] C.-H. Chien, R. Van Meter, and S.-Y. Kuo, *J. Emerg. Technol. Comput. Syst.* **12**, 9:1 (2015).
- [14] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Phys. Rev. Lett.* **81**, 5932 (1998).
- [15] C. K. Hong, Z. Y. Ou, and L. Mandel, *Phys. Rev. Lett.* **59**, 2044 (1987).
- [16] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. L. Hollenberg, *Phys. Rev. Lett.* **104**, 180503 (2010).
- [17] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin, *Phys. Rev. A* **79**, 032325 (2009).
- [18] S. Nagayama, B.-S. Choi, S. Devitt, S. Suzuki, and R. Van Meter, *Phys. Rev. A* **93**, 042338 (2016).
- [19] R. Van Meter, T. Satoh, T. D. Ladd, W. J. Munro, and K. Nemoto, *Networking Science* **3**, 82 (2013).
- [20] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, [arXiv:1708.7142](https://arxiv.org/abs/1708.7142).
- [21] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, *Phys. Rev. Lett.* **71**, 4287 (1993).
- [22] V. I. Yudson, *Phys. Rev. Lett.* **77**, 1564 (1996).
- [23] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Sci. Rep.* **6**, 20463 (2016).
- [24] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, *Phys. Rev. A* **59**, 169 (1999).
- [25] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Nature (London)* **414**, 413 (2001).
- [26] L. Jiang, J. M. Taylor, N. Khaneja, and M. D. Lukin, *Proc. Natl. Acad. Sci.* **104**, 17291 (2007).
- [27] L. Hartmann, B. Kraus, H.-J. Briegel, and W. Dür, *Phys. Rev. A* **75**, 032310 (2007).
- [28] Y. Li, S. D. Barrett, T. M. Stace, and S. C. Benjamin, *New J. Phys.* **15**, 023012 (2013).
- [29] R. Van Meter and T. Matsuo, *Connection Setup in a Quantum Network*, Internet draft draft-van-meter-qirg-quantum-connection-setup-00 (Internet Engineering Task Force, Fremont, 2019), work in progress, expires September 12, 2019.
- [30] T. Oka, T. Satoh, and R. Van Meter, in *Proceedings of 2016 IEEE Globecom Workshops, GC Wkshps 2016* (Institute of Electrical and Electronics Engineers, Piscataway, 2017).
- [31] S. E. Vinay and P. Kok, *Phys. Rev. A* **99**, 042313 (2019).
- [32] W. Dür and H. J. Briegel, *Rep. Prog. Phys.* **70**, 1381 (2007).
- [33] J. B. Altepeter, E. R. Jeffrey, and P. G. Kwiat, *Adv. At., Mol., Opt. Phys.* **52**, 105 (2005).
- [34] C. Jones, D. Kim, M. T. Rakher, P. G. Kwiat, and T. D. Ladd, *New J. Phys.* **18**, 083015 (2016).
- [35] K. Fujii and K. Yamamoto, *Phys. Rev. A* **80**, 042308 (2009).
- [36] B. Hensen, H. Bernien, A. Dréau, A. Reiserer, N. Kalb, M. Blok, J. Ruitenberg, R. Vermeulen, R. Schouten, C. Abellán *et al.*, *Nature (London)* **526**, 682 (2015).
- [37] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, and S. Wehner, *SIGCOMM '19 Proceedings of the ACM Special Interest Group on Data Communication* (ACM, New York, USA, 2019).
- [38] W. J. Munro, K. A. Harrison, A. M. Stephens, S. J. Devitt, and K. Nemoto, *Nat. Photonics* **4**, 792 (2010).
- [39] L. Aparicio and R. Van Meter, *Proc. SPIE-Int. Soc. Opt. Eng.* **8163**, 816308 (2011).
- [40] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, *Phys. Rev. A* **99**, 052330 (2019).
- [41] R. V. Meter, T. Matsuo, C. Durand, and T. Satoh (unpublished).
- [42] T. Satoh, F. Le Gall, and H. Imai, *Phys. Rev. A* **86**, 032331 (2012).
- [43] T. Matsuo, T. Satoh, S. Nagayama, and R. Van Meter, *Phys. Rev. A* **97**, 062328 (2018).
- [44] R. Paschotta, <https://rp-photonics.com/fibers.html>.

- [45] P. C. Maurer, G. Kucsko, C. Latta, L. Jiang, N. Y. Yao, S. D. Bennett, F. Pastawski, D. Hunger, N. Chisholm, M. Markham, D. J. Twitchen, J. I. Cirac, and M. D. Lukin, *Science* **336**, 1283 (2012).
- [46] D. Riedel, I. Söllner, B. J. Shields, S. Starosielec, P. Appel, E. Neu, P. Maletinsky, and R. J. Warburton, *Phys. Rev. X* **7**, 031040 (2017).
- [47] S. Bogdanović, M. S. Z. Liddy, S. B. van Dam, L. C. Coenen, T. Fink, M. Lončar, and R. Hanson, *APL Photonics* **2**, 126101 (2017).
- [48] C. Natarajan, L. Zhang, H. Coldenstrodt-Ronge, G. Donati, S. Dorenbos, V. Zwiller, I. Walmsley, and R. Hadfield, *Opt. Express* **21**, 893 (2013).
- [49] A. Reiserer, N. Kalb, M. S. Blok, K. J. M. van Bemmelen, T. H. Taminiou, R. Hanson, D. J. Twitchen, and M. Markham, *Phys. Rev. X* **6**, 021040 (2016).
- [50] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, *Nature (London)* **558**, 268 (2018).
- [51] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, *Science* **356**, 928 (2017).
- [52] A. M. Steane, *Phys. Rev. A* **68**, 042322 (2003).
- [53] L. Kleinrock, *Queueing Systems*, Vol. I, Theory (Wiley, New York, 1974).