

Asynchronous Entanglement Routing for the Quantum Internet

Zebo Yang,¹ Ali Ghubaish,¹ Raj Jain,¹ Hassan Shapourian,² and Alireza Shabani³¹Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, USA²Cisco Research, San Jose, CA, USA³Cisco Quantum Lab, San Monica, CA, USA

(*Electronic mail: zebo@wustl.edu)

(Dated: 26 October 2023)

With the emergence of the Quantum Internet, the need for advanced quantum networking techniques has significantly risen. Various models of quantum repeaters have been presented, each delineating a unique strategy to ensure quantum communication over long distances. We focus on repeaters that employ entanglement generation and swapping. This revolves around establishing remote end-to-end entanglement through repeaters, a concept we denote as the "quantum-native" repeaters (also called "first-generation" repeaters in some literature). The challenges in routing with quantum-native repeaters arise from probabilistic entanglement generation and restricted coherence time. Current approaches use synchronized time slots to search for entanglement-swapping paths, resulting in inefficiencies. Here, we propose a new set of asynchronous routing protocols for quantum networks by incorporating the idea of maintaining a dynamic topology in a distributed manner, which has been extensively studied in classical routing for lossy networks, such as using a destination-oriented directed acyclic graph (DODAG) or a spanning tree. The protocols update the entanglement-link topology asynchronously, identify optimal entanglement-swapping paths, and preserve unused direct-link entanglements. Our results indicate that asynchronous protocols achieve a larger upper bound with an appropriate setting and significantly higher entanglement rate than existing synchronous approaches, and the rate increases with coherence time, suggesting that it will have a much more profound impact on quantum networks as technology advances.

I. INTRODUCTION

Quantum networks¹⁻³ offer many possibilities for applications that go beyond what classical networks can accomplish. Among these are quantum key distribution (QKD)^{4,5}, sensing⁶, secure communication⁷, clock synchronization⁸, and distributed quantum computing^{9,10}. Quantum repeaters have been developed using various techniques to facilitate long-distance communication for these applications. These methods can be broadly categorized into two groups: the first involves transmitting encoded quantum information^{11,12} and applying quantum error correction¹³, similar to classical repeaters. The second, hereafter referred to as the quantum-native repeaters (or in some literature, first-generation repeaters¹⁴), distributes long-distance entanglement^{14,15} through entanglement swapping¹⁶ (Fig. 1), allowing end-users to carry out desired operations such as key distributions or secure communications. While a direct application of classical networking technologies to quantum networks by transmitting encoded quantum states may seem intuitive, the quantum-native repeater provides a more immediate solution to creating a network that utilizes end-to-end entanglement, as the former approaches necessitate a considerably impractical quantity of qubits for error correction purposes^{17,18}.

The development of direct-link entanglement generation has brought quantum networks closer to realization, with recent experiments demonstrating deterministic entanglement delivery between solid-state memories located in different physical locations¹⁹, facilitated by the use of telecommunication wavelengths to extend the distance between the locations^{20,21}. However, the rate of direct-link entanglement generation still decreases significantly over long distances due

to photon loss and quantum state and operation instability resulting from unavoidable interactions with the environment. Quantum repeaters based on entanglement swapping are thus necessary to extend the entanglement generation (distribution) distance. However, they may not always be capable of achieving the desired fidelity levels required for certain applications^{22,23}. Entanglement purification (see Subsection II B) can improve fidelity, but it has a trade-off. It reduces the number of entangled pairs shared between neighboring nodes on the network and requires a substantial amount of classical communication overhead²⁴. Entanglement swapping and purification form a quantum-native repeater scheme analogous to the link layer of classical networks (see Appendix A).

On top of the quantum-native repeater scheme, a network layer can be established to ensure an effective architecture^{25,26} for quantum networks, thereby enabling the full network capabilities. The quantum network layer plays a crucial role in determining the routing principles, i.e., quantum-native routing, that facilitate requests for generating end-to-end entanglements between remote stations²⁷⁻²⁹. Various metrics have been suggested to direct the routing process^{30,31}, and the entanglement rate — the number of entanglements generated per unit time — is a prevalent metric akin to throughput in classical networks, with entanglements (as the communication resources) replacing bits. In essence, designing a quantum-native routing protocol aims to determine the optimal path while maximizing the entanglement rate between any two nodes under specific constraints, such as operation probability, coherence time, and link capacity. It should be noted that a classical network is assumed to be present in the quantum network, responsible for performing routing computations and broadcasting routing information. In other words, each node in the quantum network must be able to communi-

Asynchronous Entanglement Routing

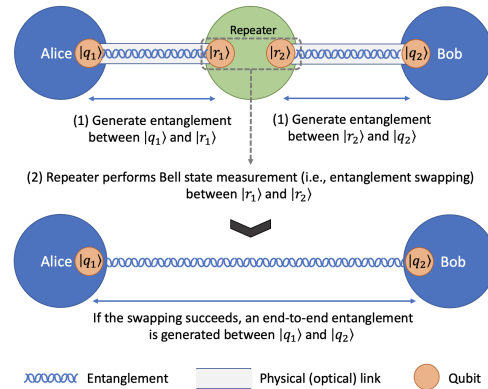


FIG. 1. Single-hop entanglement generation via entanglement swapping. Alice and Bob intend to establish an end-to-end entanglement through a repeater positioned in between. The process involves two steps: **(1)** both parties create an entangled state with the repeater, resulting in two pairs of entangled qubits: Alice's $|q_1\rangle$ paired with the repeater's $|r_1\rangle$ and Bob's $|q_2\rangle$ paired with the repeater's $|r_2\rangle$. **(2)** the repeater conducts entanglement swapping, which essentially performs the Bell state measurement on $|r_1\rangle$ and $|r_2\rangle$ (see Subsection II A) assisted by classical communication, producing an end-to-end entanglement between Alice's $|q_1\rangle$ and Bob's $|q_2\rangle$.

cate with one another via a classical network, c.f., local operations and classical communications (LOCC). This assumption can be safely made given the expectation of a hybrid quantum-classical network and is a common practice in quantum routing techniques^{27–32}. After the paths have been established with the aid of LOCC, the intermediate nodes (repeaters) carry out entanglement swapping and purification. The resulting end-to-end entanglements connecting the endpoints (end users) are then used as the communication resources for any desired operations. The nature of end-to-end entanglement resembles connection-oriented circuit-switching, in contrast to the connection-less packet switching commonly used in classical networks, which was also proposed for quantum networks³³. To avoid confusion, the term "quantum-native" is explicitly used to denote that the technique applies only to quantum networks (to distinguish from the repeater approaches that relay qubits) and is not independent of classical networks.

The routing process in existing approaches^{27–30} is divided into two phases, namely external and internal, each assigned a specific time slot. During the external phase, pairs of network nodes with a direct connection try to establish entanglement using a physical link, resulting in an "instant topology" (Fig. 2a and 2b) consisting of direct-link entanglements. In the internal phase, assuming that nodes are only aware of the direct-link entanglement generation results of their neighbors (see Subsection II C), each repeater node swaps entanglement blindly to reach an end-to-end entanglement between the two endpoints that request to communicate (Fig. 2c and 2d). If

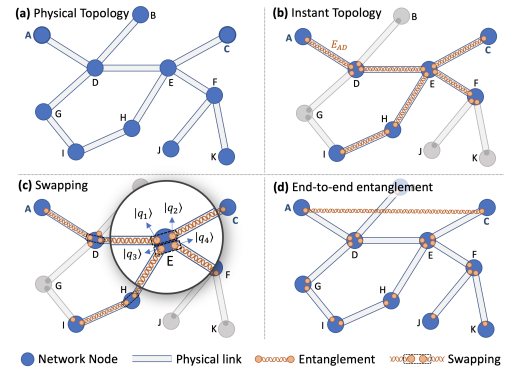


FIG. 2. End-to-end entanglement generation. Nodes A and C in the network intend to establish an end-to-end entanglement. **(a)** The network consists of nodes connected by optical communication links, forming the *physical topology*. **(b)** In the external phase, nodes connected by the same optical link attempt to generate direct-link entanglement, resulting in an *instant topology* of entanglement links (the orange helical links). For example, Nodes A and D are connected by the entanglement link E_{AD} . **(c)** In the internal phase, all repeaters involved in two pairs of entanglements perform entanglement swapping on the qubits closer to the source or destination. Node E, for instance, has four qubits, with each qubit entangled with one of its four neighboring nodes (E_{DE} , E_{HE} , E_{EF} , and E_{EC}). It performs swapping on $|q_1\rangle$ and $|q_2\rangle$ since they are connected to the nodes closer to A and C, respectively. **(d)** By consuming all entanglements in the instant topology, A and C successfully obtain an end-to-end entanglement. Since each node is only aware of itself and its neighbor's entanglement-link statuses, Node E does not know if the path A-D-E-C would succeed. It also performs swapping on $|q_3\rangle$ and $|q_4\rangle$ to increase the success rate of achieving an end-to-end entanglement.

the end-to-end entanglement is not achieved, the two phases are repeated. It is essential to synchronize both phases to ensure that entanglement "links" remain available for the internal phase. However, this synchronization and blind attempts consume all entanglements in each time slot, leading to a low entanglement rate. Furthermore, there are no obvious solutions for connecting networks of networks in these existing approaches.

In this paper, we propose a set of asynchronous routing protocols that can find the path between any two endpoints in a network with a much higher entanglement rate than existing approaches. Also, the entanglement rate increases with coherence time, and there are gateways in the network to connect networks of networks. It removes the need for synchronized executions by iterating an instant topology consisting of entanglement links. The instant topology can be in the form of a graph that can be maintained in a distributed manner, such as a destination-oriented directed acyclic graph (DODAG) [12] or a spanning tree. Such techniques have been well-studied for classical lossy networks, which share the similarity of losing a connection (in a quantum context, an entanglement in the instant topology) with a probability depending on the qual-

Asynchronous Entanglement Routing

3

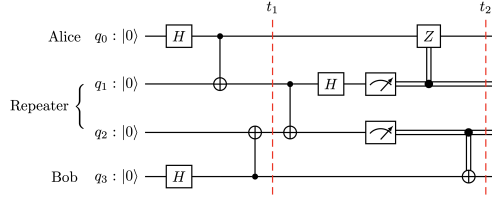


FIG. 3. The circuit of entanglement swapping.

ity of the physical link. Our theoretical analyses and simulations demonstrate that our asynchronous protocols achieve a larger upper bound and significantly higher entanglement rate compared to existing synchronous approaches. The entanglement rate increases as coherence time increases. Additionally, a tree-like instant topology such as DODAG allows for the connection of networks of networks through the root nodes. These findings suggest that our proposed protocols will have a significant impact on the development of a Quantum Internet as technology advances (as coherence time increases and networks expand in scale).

The rest of the paper is organized as follows. First, we define the problem statement of this research by introducing the routing metric and the settings of the existing routing scheme we aim to optimize. Then, we propose the framework and two example algorithms for the asynchronous routing protocols that maintain an instant topology in a distributed fashion. Next, we evaluate the routing performance from various perspectives and provide a detailed explanation of a sample routing result with DODAG. Subsequently, we analyze the simulation results to examine the effectiveness of our scheme under different conditions and establish its superiority. Finally, we conclude the paper by discussing the limitations, the extent of usability, and potential areas for future research and expansion.

II. PRELIMINARIES

This section provides an overview of key concepts and methodologies relevant to our research. It offers essential background to ensure readers grasp the subsequent discussions and the context of our research question.

A. Entanglement swapping

Entanglement swapping, in essence, is to perform a Bell state measurement on two pairs of entangled qubits. In the circuit depicted in Fig. 3, three parties are involved: Alice, Bob, and a repeater. Alice possesses a qubit q_0 , the repeater has qubits q_1 and q_2 , and Bob has a qubit q_3 . Before t_1 , operations (i.e., Hadamard and controlled-not gates) are carried out to produce two pairs of entangled qubits: one between Alice and the repeater, $|q_0q_1\rangle$, and another between the repeater

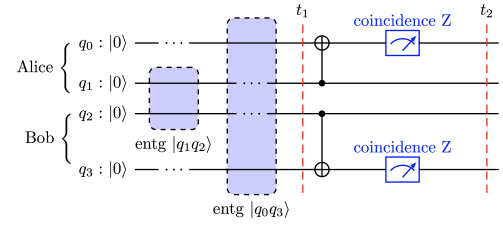


FIG. 4. An example purification circuit with link capacity two.

and Bob, $|q_2q_3\rangle$. Following this, the repeater executes a Bell state measurement, which is used initially to identify the specific Bell state that an entangled pair is in. After that, at t_2 , the two entangled pairs $|q_0q_1\rangle$ and $|q_2q_3\rangle$ are swapped to form a new, single pair of entangled qubits, $|q_0q_3\rangle$, which connects Alice and Bob with an entangled pair (i.e., an entanglement link).

B. Entanglement purification

At time t_1 , Fig. 4 demonstrates that Alice and Bob possess two entangled pairs, namely $|q_1q_2\rangle$ and $|q_0q_3\rangle$, indicating a link capacity of two between them. This implies that two raw entanglements with moderate fidelity will be utilized to produce a single entanglement with higher fidelity. Alice owns q_0 and q_1 , while Bob owns q_2 and q_3 . Both $|q_1q_2\rangle$ and $|q_0q_3\rangle$ are considered low-quality (raw entanglements). Successful purification is determined by coincidence measurements in the Z basis. After the purification process, one of the entangled pairs is consumed while the other becomes a higher-quality entanglement. In this example, $|q_0q_3\rangle$ is consumed, and $|q_1q_2\rangle$ is preserved.

C. Global and local knowledge of direct links

For a quantum routing protocol to be effective, it must be capable of identifying the optimal path as and when required. If a previously selected path becomes unavailable, the proposal must be able to identify an alternative path from the current location to the destination. Since the global knowledge of the entire network's entanglement links is often not up-to-date, the protocol must identify the best adjacent node based on its local knowledge of the entanglement links connected to it. Although the shortest path in an instant topology can be easily identified if a global view is available, it is unrealistic to assume every node has the latest version of it (e.g., to assume every node instantly knows the instant topology created by nodes A, D, E, H, I, F, C in Fig. 2). The status of all direct-link entanglements can only be propagated through classical networks, which is relatively slow. In large networks, the instant topology may have already been altered by the time the global view has been broadcast to every network node. As a result,

Asynchronous Entanglement Routing

4

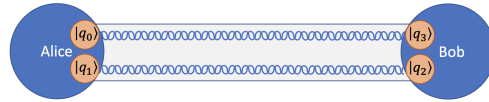


FIG. 5. A link between Alice and Bob with capacity two.

routing should be constrained to ensure that each node is only aware of the entanglement link states of its neighbors (e.g., node D only knows the direct-link entanglement generation results of links $D-A$, $D-B$, $D-G$, and $D-E$. Specifically, $D-A$ and $D-E$ have been successful, while the other two have failed).

D. Finding disjoint paths

In a network, disjoint paths refer to two or more paths between a pair of nodes (i.e., source and destination) that do not share any common node or link/edge except for the source and destination. Here, we consider link-disjoint paths where two paths do not have any link or edge in common. Finding disjoint paths is necessary in reality due to the current low link capacity in quantum networks. Assuming global knowledge of the instant topology, finding even two disjoint shortest paths in many scenarios remains an NP-complete problem, such as the min-min (i.e., to minimize the length of the shorter path of the two disjoint paths), min-max (i.e., to minimize the length of the longer path of the two disjoint paths), and min-sum (i.e., to minimize the sum of the length of the two disjoint paths) problems. Besides single-cost min-sum, others are NP-complete. In addition, it is NP-hard to obtain a ρ -approximation to the optimal solution for any $\rho > 1$. More details can be found in Reference³⁴. This increases the time needed for classical operations to find the shortest paths. Calculating paths and propagating the global view of the instant topology within the entanglement coherence time is unrealistic. However, this problem will be eased in the future if there is a high link capacity.

E. Link capacity and desired fidelity

The maximum number of entanglements that can be generated between two parties is referred to as link capacity. As illustrated in Fig. 5, Alice and Bob each have two qubits, which means they can generate two entangled pairs, leading to resource/qubit allocation and scheduling issues. For more information, please refer to Reference²⁸. However, if there is redundant link capacity, a link in the instant topology can last longer since the other pairs of qubits (besides the one being used) will continually attempt entanglement generation, providing backup direct-link entanglements. Redundant entanglements in one edge can also be utilized to generate multiple paths and support simultaneous connection requests. However, having four qubits does not necessarily mean we

can achieve the maximum of two entanglements since various entanglement-based applications, such as QKD, require a minimum fidelity level. Some entanglements must be used up to attain the desired fidelity, as explained in Subsection II B. Consequently, the actual link capacity should be $C/E(f)$. Assuming homogeneous link and node quality, this is a constant factor applied to the entanglement rate. In the main text, without loss of generality, we set the link capacity $C = 1$ and the number of entanglements for purification $E(f) = 1$ since they are constant as average numbers. This simplifies the routing problem and allows us to focus on the entanglement rate change by maintaining a distributed graph.

III. THE ENTANGLEMENT ROUTING PROBLEM

Quantum networks have several constraints that do not apply to classical networks. One such constraint is the probabilistic nature of entanglement generation and swapping performed by repeaters. Without loss of generality, we consider the generation of the maximally entangled states (Bell states) between adjacent nodes and the operation of Bell state measurement at repeater nodes to swap entanglement. Each instance of generation or swapping has a chance of succeeding or failing, leading to probabilistic and unpredictable routing paths. As pathfinding takes place and direct-link entanglements are generated, the instant topology changes constantly, emphasizing the need for an adaptive routing approach to creating an end-to-end entanglement. Finally, once an end-to-end entanglement is consumed, the network must establish a new one for a new connection request. In short, end-to-end entanglements allow the creation of dedicated unicast channels between node pairs, regardless of their position in the network. As mentioned, this characteristic of entanglement resembles connection-oriented circuit-switching rather than the connection-less packet switching used in classical networking. Here, we present a definition of the quantum-native routing problem.

Let us consider a graph $G(V, E)$ that represents the physical topology of the quantum network (Fig. 2a). In this graph, each node $v \in V$ represents a quantum-native repeater, while each edge $e \in E$ denotes an optical link that connects two repeater nodes. The instant topology (Fig. 2b) is denoted as $G'(V', E')$, which is a subgraph of G . E' is the set of direct-link entanglements, and V' are the nodes that are connected by the entanglements in E' . Additionally, each edge (link) e can support multiple entanglement generations with multiple qubit pairs between adjacent nodes, where the link capacity $C(e)$ of link e is defined as the maximum number of entangled pairs generated between adjacent nodes connected by e . Each node contains a finite number of qubits (depicted as orange dots in Fig. 2), denoted as node capacity $C(v) \geq \sum_{e \in N(v)} C(e)$ where $N(v)$ refers to the set of links connecting node v . To achieve a desired fidelity f , we need $E(f)$ entanglements for purification, and each entanglement has an average coherence time, T_{co} . For simplicity, we assume that the link capacity C and purification number $E(f)$ are both equal to 1. This implies that, as depicted in Fig. 2, only one qubit is present on

one side of the physical link. Nonetheless, varying link capacities and purification numbers will not impact the evaluation of the routing scheme as they simply introduce a constant multiplier to the entanglement rate. The entanglement rate will increase linearly relative to the actual link capacity, as detailed in Subsection II E. Given our assumption that $C/E(f) = 1$, we can omit the $C/E(f)$ multiplier from each entanglement rate equation.

In addition, for each entanglement generation between a direct link e , there is a probability $p(e)$ that the generation will succeed, while each entanglement swapping at a repeater v has a probability $q(v)$ of success. For convenience, we assume homogeneous entanglement generation probability, denoted by p , and entanglement swapping probability, denoted by q , on all network links and nodes. Finally, the entanglement rate, denoted by ξ , is the performance metric indicating the number of end-to-end entanglements (Bell states) that can be generated in a unit time T . T must satisfy $T \leq T_{co}$. Existing approaches usually consider the time required for both internal and external phases (as discussed in the Introduction Section) to be one unit time and assume it to be longer than the coherence time. To facilitate comparison, we set the unit time as $T = \frac{T_{co}}{n}$, where $n \in \mathbb{Z}$ and $n \geq 1$.

With these preliminaries, the problem of quantum-native routing is to design an effective algorithm that maximizes the entanglement rate under the constraint that nodes only possess local knowledge of adjacent links in the instant topology (For example, Node E in Fig. 2b only knows the direct-link generation results of E_{ED} , E_{EH} , E_{EF} , and E_{EC}). Note that it is unrealistic to assume that a global view of the instant topology is known, which may allow an algorithm to determine the shortest paths in the instant topology. In existing protocols such as^{27,28} that depend on global knowledge of the instant topology, entanglements need to possess longer coherence times as the network size (i.e., the number of network nodes) expands. This is because direct-link entanglement information takes a relatively long time to traverse the entire network. Consequently, in a large-scale quantum network, direct-link entanglements may decay by the time all direct-link generation results are collected. Thus, a distributed approach is necessary to find optimal paths with local knowledge of the instant topology.

Nonetheless, even if we assume to have a global view of the instant topology, it is still NP-complete to find multiple disjoint shortest paths (for robustness) in many scenarios (see Subsection II D), given the restricted resources available at each link³⁵. However, many existing approaches assume that the global view of the instant topology is known, such as^{28–30,32}. A recent work by Pant et al.²⁷ proposes a blind search routing strategy with local knowledge of the instant topology, which, however, requires synchronized phases and consumes all direct-link entanglements in each attempt, as shown in Fig. 2c and 2d, resulting in a low entanglement rate. To ensure efficiency, entanglement swapping within a repeater should only occur when a path that includes that node is selected. Our approach, therefore, involves maintaining a distributed graph and reserving entanglements not used in the current paths for future use.

IV. ASYNCHRONOUS ROUTING SCHEME

In essence, our method of asynchronous routing functions on an event-driven basis. It establishes and maintains the instant topology as a distributed graph (such as DODAG or spanning tree) and reacts to connection requests (events) by following the path determined in the distributed graph, such as tracing the path through the most current DODAG's root. The crux of the matter is the former service, which outlines how to preserve the instant topology as a distributed graph that determines a route for any given pair of nodes. In particular, we illustrate the protocol using two sample graphs: DODAG and spanning tree. Protocol 1 presents the algorithmic steps of the routing processes, including quantum operations (QO), such as entanglement generation and swapping, and classical operations (CO), such as event listening and request handling. Moreover, the maintenance of the two example graphs (DODAG and spanning tree) is illustrated later in Protocol 1.1 and Protocol 1.2, which are replaceable by any distributed routing algorithms.

Algorithm 1 Asynchronous routing scheme

(N_{prev}, N_{next} : the previous and next node in the path.)
(N_{dest} : destination.)

▷ **Resource:** System Interface and Variable

- 1: $CO \leftarrow$ A set of classical operations
- 2: $QO \leftarrow$ A set of quantum operations
- 3: $N_c \leftarrow$ The current node that is running a protocol instance

▷ **On-demand Service:** Topology Maintenance and Listener

- 4: $GRAPHUPDATE(QO, CO, N_c)$ ▷ Refer to Protocol 1.1 or 1.2
- 5: $CO.LISTEN('connection_request', N_{prev}, N_{dest})$

▷ **Request Handler:** Path Navigation and Swapping

- 6: **function** NAVIGATE(N_{prev}, N_{dest})
 - 7: $N_{next} \leftarrow$ NEXTHOPDETERMINATION(N_c, N_{dest})
 - 8: **if** N_{next} is not null **then**
 - 9: $bool\ i \leftarrow$ QO.ENTANGLEMENTSWAPPING(N_{prev}, N_{next})
 - 10: **if** i is true **then**
 - 11: $CO.EMITCONNREQUESTTO(N_{next}, (N_c, N_{dest}))$
 - 12: **else** ▷ When the quantum operation fails
 - 13: $GRAPHUPDATE(QO, CO, N_c)$
 - 14: NAVIGATE(N_{prev}, N_{dest})
 - 15: **end if**
 - 16: **else** ▷ When the current node is not in the graph
 - 17: $GRAPHUPDATE(QO, CO, N_c)$
 - 18: NAVIGATE(N_{prev}, N_{dest})
 - 19: **end if**
 - 20: **end function**
 - 21: $CO.CONNECTIONHANDLER \leftarrow$ NAVIGATE(N_{prev}, N_{dest})
-

Protocol 1 is executed on each network node. To join the network, any node has to only abide by the protocol without requiring synchronization of operations and communications across all nodes. Lines 1 and 2 of Protocol 1 define the interfaces for executing classical operations (i.e., CO) and conducting quantum operations (i.e., QO). Line 3 assigns the identifier of the current node running the protocol instance to N_c . Starting from Line 4, an on-demand service is initiated to

update the instant topology continuously, based on the rules specified in the distributed graph, such as Protocol 1.1 or 1.2 in the subsequent subsections. Line 5 is responsible for listening to incoming connection requests. It is worth noting that this is a self-propagating entanglement swapping process, whereby each node that receives the connection request only has to focus on connecting itself with the destination node. Once a connection request arrives, Line 21 processes it by redirecting the request to the NAVIGATE function, which begins at Line 6. The NAVIGATE function verifies if both the current and destination nodes are present in the distributed graph. If it finds that both nodes are present in the graph, it proceeds to conduct entanglement swapping between the qubit connecting to the previous node N_{prev} and the qubit connecting to the next node N_{next} . Assuming a successful entanglement swapping, the NAVIGATE function will send a connection request to the next node, requesting the establishment of a connection between the current and destination nodes. If the entanglement swapping is unsuccessful, or either the source or the destination node is absent from the distributed graph, the node will wait for an update to the graph before attempting to navigate again.

In short, to initiate the creation of end-to-end entanglement between a source and a destination node, the source node issues a connection request, similar to Line 11, to begin the navigation process. Each intermediate node requests the next node along the path toward the destination to perform entanglement swapping until the destination is reached. In the subsequent two subsections, we present two illustrations of the distributed graph: DODAG and spanning tree.

A. DODAG protocol

A DODAG³⁶ is a graph with a tree-like structure formed by selecting a root node and then growing by the exchange of control messages, as illustrated in Fig. 6. The routing protocol revolves around the utilization of one or more DODAGs, employing designated *routing metrics* and an *objective function* to assist in selecting the optimal path to the root node. Subsequently, communication between the two endpoints is established via the root node. The specific graph of a DODAG at any given time is referred to as a DODAG version. If needed, a network can support multiple DODAGs with different roots.

The *routing metrics* are the set of factors used to evaluate a candidate hop that determines a preferred path. These factors can vary, and here we include elements such as link quality, which is determined by link capacity (C) and the probability of direct-link entanglement generation (p), as well as node quality, which is influenced by distance to the root node, number of qubits, probability of entanglement swapping (q), and coherence time (T_{co}) — these metrics aid in creating a routing strategy that maximizes the entanglement rate. To focus on the primary concern of pathfinding, we assume that node and link qualities are homogeneous. However, in actual control messages, such as DIS (DODAG Information Solicitation), DIO (DODAG Information Object), and DAO (Destination Advertisement Object) in Fig. 6, distinct node and link qualities may

be included to assist in selecting or eliminating parents and children. For example, if a link or node falls short of the minimum requirement (e.g., fidelity $f < 0.5$), it will be removed from the DODAG, along with its branches, and newer nodes will select stronger parents in their selection process. It is important to note that if a direct-link entanglement is lost (either due to decoherence or removal), the branch that was disconnected from the DODAG tree will remain a floating DODAG branch. During this time, the disconnected node will act as the temporary root node for the branch until it can successfully rejoin the main DODAG. Once it has rejoined, the floating branch will become a part of the main DODAG once again.

The *objective function* for a DODAG establishes how nodes select and optimize the structure of the graph. It translates routing metrics into *ranks* that determine rules such as parent selection and loop avoidance (see Appendix B). In other words, the node's rank in a DODAG indicates its connectivity with the root and is determined as a scalar measure based on the routing metrics. As the node and link qualities are considered homogenous in this paper, the ranks should increase monotonically as the nodes get farther from the root, as shown in Fig. 6. It is worth noting that a DODAG node can have multiple parents as long as there are no loops in the graph, which is ensured by the objective function when calculating rank for new nodes. Consequently, routing can select preferred paths based on factors such as fewer hops or stronger nodes and links when dealing with distinct node and link qualities. An initial objective function is provided below for rank computation and parent selection.

Suppose a node v_{new} wants to join a DODAG. It must calculate the rank step between itself and each of its neighboring nodes in the DODAG, representing the amount by which the rank should be increased along a specific link. Let us represent a neighboring DODAG node as v_{nbr} , and the rank step between v_{new} and v_{nbr} as $S(v_{new}, v_{nbr})$. The value of $S(v_{new}, v_{nbr})$ is determined by the difference between the connectivity of node v_{new} (relative to the root), denoted as $\gamma(v_{new})$, and the connectivity of node v_{nbr} , denoted as $\gamma(v_{nbr})$. If we denote the length (i.e., the number of links) of the path from a node v to the root as $L(v)$, then $\gamma(v)$ can be calculated by:

$$\gamma(v) = p^{L(v)} q^{L(v)-1} T_{co} \quad (1)$$

With that, we can compute $\gamma(v_{nbr})$ by $L(v_{nbr})$ and compute $\gamma(v_{new})$ based on the fact $L(v_{new}) = L(v_{nbr}) + 1$. With that, the rank step $S(v_{new}, v_{nbr})$ can be calculated as follows:

$$S(v_{new}, v_{nbr}) = |\gamma(v_{nbr}) - \gamma(v_{new})| \quad (2)$$

The rank step can also be represented as $S(v_{new}, v_{nbr}) = |\gamma(v_{new}) (\frac{1}{pq} - 1) T_{co}|$ since we assume uniform p , q and T_{co} . A stronger node (i.e., a node with higher connectivity) would have a smaller rank step. Next, node v_{new} selects the neighboring node with the minimum rank step (suppose it is v_{nbr}) to become its parent node. This enables node v_{new} to determine its rank value by following v_{nbr} , denoted as $\rho(v_{new})$, when it successfully joins the DODAG:

$$\rho(v_{new}) = \rho(v_{nbr}) + S(v_{new}, v_{nbr}) \quad (3)$$

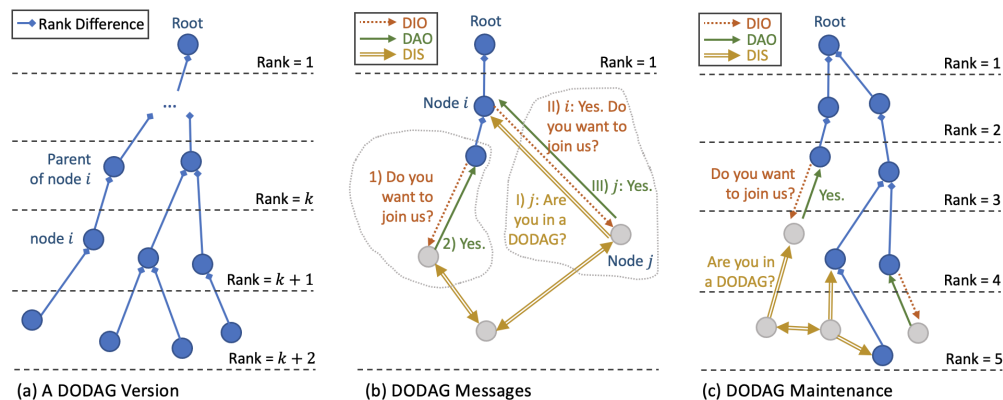


FIG. 6. **DODAG and its control messages.** (a) This represents a version of a DODAG at any given time. The rank values of nodes in the graph increase as their distance from the root node increases. When a node is directly connected to another node and has a lower rank value, it is considered the parent of that node. Note that the rank increment here is set as 1 for simplicity, but it can be any positive real number indicating the cost metric. (b) Each node outside the DODAG communicates with its neighboring nodes by sending DIS messages to inquire if they belong to the DODAG. When a neighboring node responds with a yes (as Node i does to Node j in the figure) using DIO messages, the node (Node j) sends DAO messages asking to join the DODAG. Then, the neighboring node (Node i) computes a rank value for the new node and allows it to join. If the neighboring node does not respond, the node continues to send DIS messages until it receives a yes response or is asked to join the DODAG by a DODAG neighbor using DIO (as in the case of the dotted area on the left). (c) A DODAG expands itself and maintains its version using control messages indicated in (b).

While a DODAG is constantly updated according to the rules above (refer to Appendix B for information on loop avoidance), the routing process in the DODAG is as simple as moving up to the root and then descending to the intended destination by selecting the parent with the lowest rank value among all parents (or both source and destination ascend to the root). The maintenance of the DODAG is outlined in Protocol 1.1.

B. Distributed spanning tree

As a result of having access solely to local knowledge of the instant topology of a quantum network, constructing a spanning tree — a subgraph of a connected undirected graph that covers all vertices using the least number of edges — necessitates a distributed approach. Numerous distributed algorithms for spanning tree construction have been developed, including the renowned Gallagher-Humblet-Spira (GHS) algorithm³⁷. At the beginning of a GHS algorithm, each node is considered a fragment, and these fragments progressively fuse to form larger ones. To combine with another fragment, a fragment must locate its least-weight outgoing edge and work collaboratively with other nodes to find it in a distributed fashion. Over time, the number of fragments reduces until only one fragment remains, which is the minimum spanning tree (MST). A spanning tree guarantees the existence of a path between any two nodes. Nevertheless, using a path will disconnect parts of the tree, requiring all nodes to update it con-

tinuously. Our assumption of uniform p , q , and T_{co} implies that all edge weights are equal in our situation.

However, we must address the instant topology's probabilistic and decoherent entanglements (links). Fig. 7 illustrates our modified version of the GHS algorithm. In short, it addresses the separation of a fragment caused by the decoherence of entanglements and continuous attempts to generate an outgoing edge due to the probabilistic nature of entanglement generation. We achieve this by incorporating a *split* event in addition to the *merge* and *absorb* processes of the GHS algorithm. The following paragraphs outline how the modified GHS algorithm operates within a quantum network.

In the physical topology $G(V, E)$, each node starts as a *fragment*, and the objective is to connect all nodes with minimum weight (i.e., the least number of edges in our case) with no loop. In this context, an edge represents a direct-link entanglement in a quantum network. A fragment F is a subset of V and is assigned a non-decreasing integer level $L_v(F)$ that begins with 0. Each node in a fragment must have the same level value as its fragment, i.e., $L_v(v) = L_v(F) \forall v \in F$. Fragments with a non-zero level are assigned a unique name or fragment ID, denoted as $FID(v) = FID(F) \forall v \in F$. Every node chooses a neighbor node to form a core edge under certain constraints, such as the neighbor node with the minimum weight. Each fragment has only one core edge, and the core edge selection process is beyond this text's scope but can be found in Reference³⁷. The two nodes connected by a core edge are responsible for broadcasting the fragment ID and level to all other nodes in the fragment. After receiv-

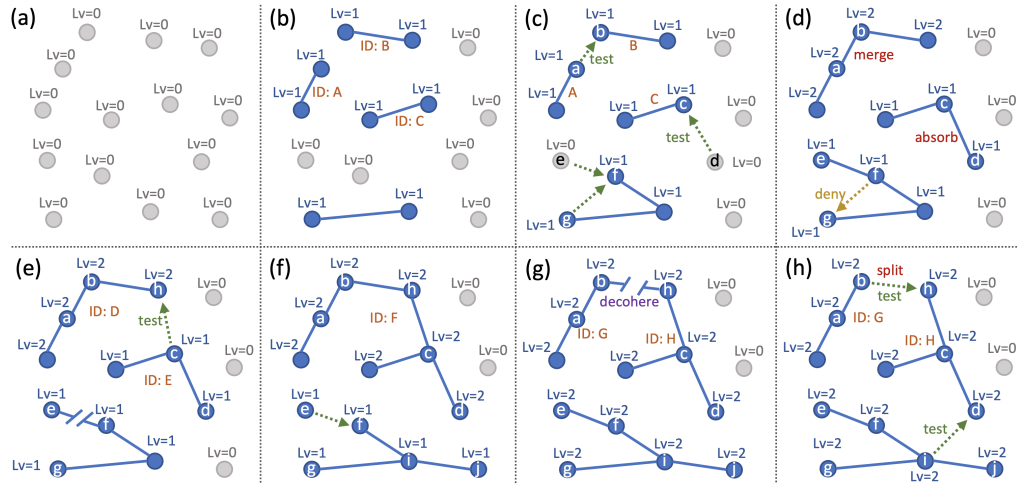


FIG. 7. Distributed spanning tree by the modified GHS algorithm. The figure illustrates the maintenance of instant topology in a stepwise manner as a spanning tree without including the physical links. (a) Initially, all nodes are considered as fragments with a level value of 0. (b) Each node sends a "test" message to its adjacent nodes, resulting in some successful and some failed connections (i.e., entanglement links). Since all level values are the same at the start, any successful connection leads to a merge process where the levels are incremented by 1, and the connecting edge becomes the core edge of that fragment. The fragments are denoted as A , B , and C . (c) Nodes connected by core edges send "test" messages to their adjacent nodes (not every "test" message is shown for brevity). For example, Node a sends a message to Node b , and Node d sends a message to Node c . (d) Fragments A and B have the same level, so they merge, and their levels are incremented by 1. Fragment C absorbs another node, and its level remains the same. Note that Node f declines to connect with Node g because they have the same Fragment ID. (e) Node c sends a "test" message to Node h . (f) Fragment D absorbs Fragment E and becomes Fragment F . (g) Edge $b-h$ breaks, causing Fragment F to split into Fragments G and H . (h) The split event is triggered, and Nodes b and h send "test" messages to their adjacent nodes.

ing a broadcast message from the core, every node searches for its minimum-weight outgoing incident edge. For instance, if node v receives a broadcast, it selects its minimum weight edge and sends a "test" message to node v' on the other side (assuming $v' \in F'$) with its fragment's ID and level. Then, node v' determines whether the edge is an outgoing edge and sends a message back to inform node v of the result. The decision is based on the following criteria:

1. if $FID(v) = FID(v')$, the edge is not outgoing because they are in the same fragment;
2. if $FID(v) \neq FID(v')$ and $Lv(v) \leq Lv(v')$, the edge is outgoing;
3. if $FID(v) \neq FID(v')$ and $Lv(v) > Lv(v')$, the algorithm cannot reach a conclusion because the two nodes may already belong to the same fragment, but node v' has not yet discovered this due to the delay of a broadcast message. In this case, the algorithm allows node v' to postpone its response until its level becomes higher than or equal to the level it received from node v . Refer to Appendix C for scenarios involving broadcast delay in the GHS algorithm.

When an outgoing edge is present, the fragments will begin

to merge through the *merge* and *absorb* processes (direct-link entanglement generation on the outgoing edge). If $Lv(F) = Lv(F')$, the fragments will merge through the merge process, resulting in a new fragment with an increased level of $Lv(F) + 1$. On the other hand, if $Lv(F) < Lv(F')$, the fragments will merge through the absorb process, and the combined fragment will have the same level as F' . When an edge within a fragment decoheres, it triggers a *split* event, where the two adjacent nodes generate new fragment IDs for their own fragments and broadcast them to the rest of the fragment. The levels of the new fragments remain the same. The increase in level during the merge process is to prevent the formation of an edge to a fragment that has already been combined by the other outgoing edge, which can cause a loop (see Appendix C). The split event does not combine fragments and thus does not affect the merge and absorb processes or the levels of any fragments.

As the rules dictate, a distributed spanning tree undergoes continuous updates. During the routing process within the spanning tree, the objective is to determine the correct direction toward the destination from the multiple available directions. Whenever an outgoing edge is detected, the relative position of each node in the tree, along with the new level and fragment ID, will be broadcast. This relative position is used

Asynchronous Entanglement Routing

to determine the correct direction toward the destination. Protocol 1.2 provides an outline of the distributed spanning tree's update process. To be concise, we have omitted the broadcasting and listening of events that update the fragment ID and level within a fragment.

C. Time complexity

The time complexity of asynchronous pathfinding algorithms can vary depending on the specifics of the algorithm and the nature of the network. Here, we analyze the time complexity of the asynchronous protocols in terms of link availability probability p , node availability probability q , and consider a grid network with node size n (i.e., $\sqrt{n} \times \sqrt{n}$ grid) where the root is at the center of the grid.

The time complexity of DODAG can be analyzed in two parts: the time complexity for constructing the DODAG and the time complexity of pathfinding within the DODAG. In the worst case, constructing a DODAG from a network can have a time complexity of $O(n)$ if we consider all possible nodes and links. However, since each link and node has a probability p and q of being available, respectively, on average, we are dealing with $p\sqrt{n}$ nodes and $q\sqrt{n}$ links (assuming a full mesh). So, the complexity becomes $O(pqn)$ for creating the DODAG. However, this decreases the probability of the pathfinding part of the protocol. In other words, the reduction

Algorithm 1.1 DODAG maintenance

($N_{neighbor}$: the node that the current node is communicating with)

```

1: function NEXTHOPDETERMINATION( $N_c, N_{dest}$ )
2:   return  $N_c.next$   $\triangleright$  relative position is updated when receiving
   a broadcast
3: end function
4: function GRAPHUPDATE( $QO, CO, N_c$ )
5:    $CO.DIS\_HANDLER \leftarrow DIS(N_{neighbor}, CO)$ 
6:    $CO.DIO\_HANDLER \leftarrow DIO(N_{neighbor}, QO, CO)$ 
7:   while true do
8:     if  $N_c$  is in DODAG then
9:        $CO.BROADCASTDIO(N_c)$ 
10:       $CO.LISTEN('DIS', N_{neighbor})$ 
11:     else
12:        $CO.BROADCASTDIS(N_c)$ 
13:       $CO.LISTEN('DIO', N_{neighbor})$ 
14:     end if
15:   end while
16: end function
17: function DIS( $N_{neighbor}, CO$ )
18:    $CO.SENDDIO(N_{neighbor})$ 
19: end function
20: function DIO( $N_{neighbor}, QO, CO$ )
21:   bool  $j \leftarrow CO.CHECKMINIRANK$ 
22:   if  $j$  is true then
23:      $CO.SENDDAO(N_{neighbor})$ 
24:     bool  $k \leftarrow false$ 
25:     while ! $k$  do
26:        $k \leftarrow QO.ENTANGLEMENTGENERATION(N_c, N_{neighbor})$ 
27:     end while
28:   end if
29: end function

```

9

Algorithm 1.2 Distributed spanning tree

($N_{neighbor}$: the node that the current node is communicating with)

```

1: function NEXTHOPDETERMINATION( $N_c, N_{dest}$ )
2:   return  $N_c.next$   $\triangleright$  relative position is updated when receiving
   a broadcast
3: end function
4: function GRAPHUPDATE( $QO, CO, N_c$ )
5:    $CO.TESTMESSAGE\_HANDLER \leftarrow$ 
   TESTMESSAGE( $N_{neighbor}, QO, CO$ )
6:   while true do
7:      $CO.BROADCASTTESTMESSAGE(N_c)$ 
8:      $CO.LISTEN('test', N_{neighbor})$ 
9:   end while
10: end function
11: function TESTMESSAGE( $N_{neighbor}, QO, CO$ )
12:   if  $CO.FID(N_{neighbor}) \neq CO.FID(N_{neighbor})$  then
13:     if  $CO.Lv(N_c) \leq CO.Lv(N_{neighbor})$  then
14:       bool  $r \leftarrow false$ 
15:       while ! $r$  do
16:          $r \leftarrow QO.ENTANGLEMENTGENERATION(N_c, N_{neighbor})$ 
17:       end while
18:       if  $CO.Lv(N_c) == CO.Lv(N_{neighbor})$  then  $\triangleright$  merge
19:         char  $id \leftarrow CO.GENERATEID$ 
20:         int  $lv \leftarrow CO.Lv(N_c) + 1$ 
21:          $CO.SETID(N_c, id)$ 
22:          $CO.SETID(N_{neighbor}, id)$ 
23:          $CO.SETLV(N_c, lv)$ 
24:          $CO.SETLV(N_{neighbor}, lv)$ 
25:       else  $\triangleright$  absorb
26:         char  $id \leftarrow CO.FID(N_{neighbor})$ 
27:         int  $lv \leftarrow CO.Lv(N_{neighbor})$ 
28:          $CO.SETID(N_c, id)$ 
29:          $CO.SETLV(N_c, lv)$ 
30:       end if
31:        $CO.BROADCASTNEWFRAGMENTINFO(N_c, id, lv)$ 
32:        $CO.BROADCASTNEWFRAGMENTINFO(N_{neighbor},$ 
    $id, lv)$ 
33:     else
34:       TESTMESSAGE( $N_{neighbor}, QO, CO$ )
35:     end if
36:   end if
37: end function

```

of time complexity of DODAG construction of the algorithms (by the decrease of p and q) increases the success probability of pathfinding. For pathfinding, the paths from the source or destination to the root are planned by the rank values, which means the complexity corresponds to the length of the planned path. The worst-case time complexity of DODAG pathfinding is thus the summation of the furthest possible distance between source (i.e., \sqrt{n}) and root and that between destination and root (i.e., \sqrt{n}). Therefore, the overall complexity is $O(\sqrt{n})$. Note that these are the worst-case time complexities of the algorithm since we consider DODAG construction and pathfinding as separate steps, but they are happening simultaneously in practice. Thus, the true complexity can be dynamic and less than this in practice.

The time complexity of the spanning tree protocol can be similarly analyzed in two parts: constructing the distributed spanning tree and pathfinding in the spanning tree. Gener-

ally, a distributed algorithm to construct a minimum spanning tree (like the GHS algorithm) can have a time complexity of $O(n \log n)$ ³⁷. In the worst case, all fragments are combined with "Merge" operations, so the number of fragments decreases by half at each level. Therefore, the maximum number of levels is $O(\log n)$. Each level takes $O(n)$ time (for information exchange in broadcast). Hence, the time complexity is $O(n \log n)$. Since a spanning tree is a tree, and trees are acyclic and have only one path between any two nodes, locating a path in a tree (given pointers to the start and end nodes) can be done in $O(n)$ in the worst case, as you may need to traverse the entire tree (in a skewed tree scenario). But if it is reasonably balanced, the path could be found much faster, in $O(\log n)$ time. Note that the discussion of message complexity of both algorithms is out of the scope of this article but can be found in³⁸.

V. PERFORMANCE ANALYSIS

To assess the effectiveness of our asynchronous routing protocol, we utilize the entanglement rate (ξ) as a metric. To simplify matters, we focus on a 2D grid topology (as illustrated in Fig. 8a), denoted by $G(V, E)$, and define an instant topology $G'(V', E')$ as a subset of this grid. Each node in G is only aware of the status of entanglements in the four direct links connected to it and has no knowledge of the larger instant topology beyond that.

In Section IV A, we used $L(v)$ to denote the length (number of links) of a path from node v to the root of a DODAG. It is possible to extend this notation to represent the length of the path between any two nodes (e.g., source node s and target node t) as $L(s, t)$, but this value only applies to a specific instance of the instant topology. To calculate the entanglement rate in a more general case, we can estimate this value by approximating the mean length of all paths between two nodes in the physical topology. Specifically, we use $l_{s,t}$ to represent the mean length of all possible paths between nodes s and t in G . $l_{s,t}$ is of interest in percolation theory for a 2D square lattice, but it cannot be determined analytically. However, an approximation is possible^{39,40} (see Appendix D).

A. Synchronous protocol analysis

Let us denote the average number of disjoint paths found between s and t as $n_{s,t}$, and in a 2D grid topology, we have $0 \leq n_{s,t} \leq 4$ since the maximum number of neighboring links that can be used by a node is four. Assuming we have only one possible entanglement in a direct link (i.e., $C/E(f) = 1$), we can estimate the average entanglement rate ξ_{syn} (per time slot) between nodes s and t for existing synchronous protocols as follows:

$$\xi_{\text{syn}}(s, t) = p^{l_{s,t}} q^{l_{s,t}-1} n_{s,t} \quad (4)$$

Let us denote the length of the shortest path from s to t as $\hat{l}_{s,t}$. With the range of $n_{s,t}$, we can establish the bounds for

$\xi_{\text{syn}}(s, t)$ (the length of the shortest path is the minimum length for each of the four paths):

$$0 \leq \xi_{\text{syn}}(s, t) \leq 4p^{\hat{l}_{s,t}} q^{\hat{l}_{s,t}-1} \quad (5)$$

It is noteworthy that our examination of the entanglement rate pertains to exclusive connection requests. For an analysis of link capacity allocation in the context of a batch of requests, we refer interested readers to Reference²⁸.

B. Asynchronous protocol analysis

In essence, the idea of asynchronous protocols is to preserve the direct-link entanglements that do not fall on the route between the two communicating parties, thus enabling the upkeep of a distributed graph such as DODAG and a spanning tree for future use. Specifically, we take the DODAG protocol as an example and examine its performance in this subsection. Besides s and t , the root node of the DODAG is denoted by r . We define $l'_{s,r}$ as the average number of direct-link entanglements already in the path between the root node r and node s , and $l'_{r,t}$ as that between nodes r and t . Note that $l'_{s,r}$ and $l'_{r,t}$ are proportional to T_{co} because higher values of T_{co} result in more entanglement links that remain un-decohered in G' . With this, we can estimate the entanglement rate ξ_{DODAG} for the DODAG protocol as:

$$\xi_{\text{DODAG}}(s, t) = p^{l'_{s,r} + l'_{r,t} - l'_{s,r} - l'_{r,t}} q^{l'_{s,r} + l'_{r,t} - 1} n_{s,t} \quad (6)$$

To facilitate comparison, we adopt a unit time for the entanglement rate that is equivalent to one time slot in synchronous approaches. As mentioned earlier, we define the unit time as $T = \frac{T_{co}}{n}$, where $n \in \mathbb{Z}$ and $n \geq 1$. For brevity, we disregard the symbol T and express T_{co} as n unit times. Since $l'_{s,r} \propto T_{co}$ and $l'_{r,t} \propto T_{co}$, it follows that $\xi_{\text{DODAG}}(s, t)$ increases as the coherence time grows. Nonetheless, by definition, we know that $\xi_{\text{syn}}(s, t)$ is independent of T_{co} .

The range of $n_{s,t}$ in our asynchronous approach is the same as in the synchronous approach, as a node can have up to four paths to the destination (having four parent nodes). Let us denote the length of the shortest path from s to r and from r to t as $\hat{l}'_{s,r}$ and $\hat{l}'_{r,t}$, respectively. Then, we can establish the bounds of $\xi_{\text{DODAG}}(s, t)$ as follows (assuming s and t need to go through the root r):

$$0 \leq \xi_{\text{DODAG}}(s, t) \leq 4q^{\hat{l}'_{s,r} + \hat{l}'_{r,t} - 1} \quad (7)$$

The upper bound is attained when both s and t are already present in the DODAG, meaning no direct-link entanglement generation (expressed by p) is necessary. It should be noted that our assumption for Equation 7 is that s and t must pass through the root node r . However, in reality, if s and t are located on the same branch, they can pass through a shared ancestor instead. As such, the right-hand side of Equation 7 represents a worst-case upper bound, and in reality, the upper bounds for most specific scenarios would be greater. We demonstrate that even in the worst-case scenario where nodes need to traverse the root node, the DODAG protocol provides

Asynchronous Entanglement Routing

11

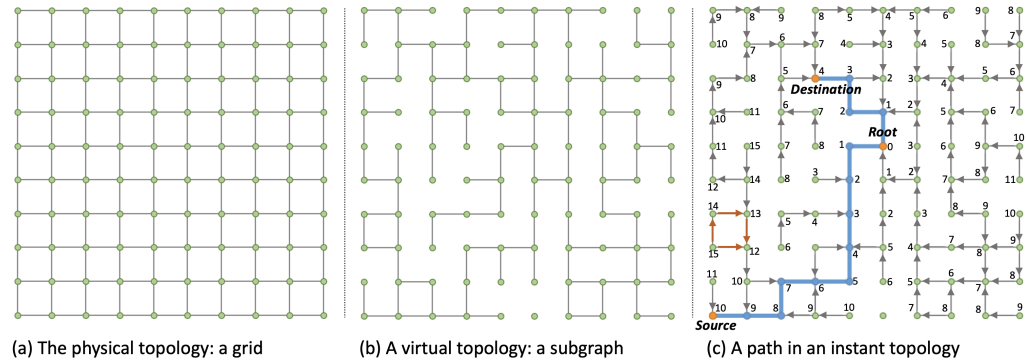


FIG. 8. **Grid topology.** (a) A 2D grid represents the physical topology $G(V, E)$. (b) An instant topology $G'(V', E')$ is a subset of this grid. (c) A DODAG path can be identified within an instant topology, with the connected graph forming a DODAG with respect to the root node. The arrows indicate the rank differences (parent-child relationships), while small digits beside nodes indicate their rank value. It is important to note that the small squares in (b) and (c) do not represent loops but occur when nodes select multiple parents, as shown in the example of orange arrows in (c). These integers are solely to provide an example that illustrates the relative difference of rank values. Real values would be more fine-grained (i.e., can be any positive real numbers).

a higher upper bound (for most node pairs s and t) than existing synchronous approaches. On the other hand, the lower bound happens when there are no direct-link entanglements in G' , and s and t cannot be incorporated into the DODAG within a single unit time.

The worst-case upper bound of our approaches is represented by $\xi_{DODAG}^{UB}(s, t) = 4q^{\hat{l}_{s,r} + \hat{l}_{r,t} - 1}$. Initially, let us examine the case where $p = q$, then the upper bound of synchronous approaches is $\xi_{syn}^{UB}(s, t) = 4q^{2\hat{l}_{s,t} - 1}$. Therefore, the difference between $\xi_{DODAG}^{UB}(s, t)$ and $\xi_{syn}^{UB}(s, t)$ is attributed to the difference between $2\hat{l}_{s,t}$ and $\hat{l}_{s,r} + \hat{l}_{r,t}$. Let us denote the angle formed by s , r , and t with respect to r as θ , for instance, θ_1 , θ_2 , and θ_3 in Fig. 9. As demonstrated in Fig. 9, we can infer the difference between $2\hat{l}_{s,t}$ and $\hat{l}_{s,r} + \hat{l}_{r,t}$ by the relationship of sides in a triangle:

1. If θ is obtuse (e.g., θ_3), $\hat{l}_{s,r} + \hat{l}_{r,t} < 2\hat{l}_{s,t}$ (i.e., the sum of the two sides adjacent to an obtuse triangle is less than twice the length of the other side. See the next subsection).
2. If θ is a right angle and the three nodes form an isosceles right triangle (e.g., θ_1 with s_2 and t_2), $\hat{l}_{s,r} + \hat{l}_{r,t} = 2\hat{l}_{s,t}$. If θ is a right angle and the three nodes do not form an isosceles right triangle (e.g., θ_1 with s_2 and t_4), $\hat{l}_{s,r} + \hat{l}_{r,t} < 2\hat{l}_{s,t}$.
3. If θ is acute (e.g., θ_2), there are two cases depending on the relative position of the three nodes. Assuming s is the one closer to r (i.e., t is farther from the root. Note that these are also valid when treating t as the closer one to the root):

- (a) When $\hat{l}_{s,r} \leq \alpha \hat{l}_{s,t}$ (See the next subsection for the value of α with respect to θ), $\hat{l}_{s,r} + \hat{l}_{r,t} \leq 2\hat{l}_{s,t}$ (e.g.,

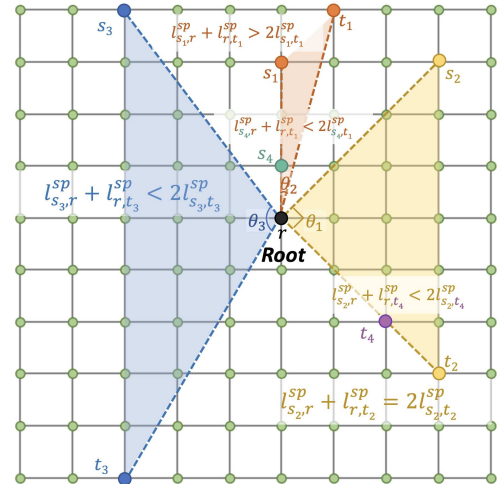


FIG. 9. **The location scenarios in a DODAG.**

θ_2 with s_1 and t_1). That is, our approach is better when s is close to the root but far from t . In the case of equality, the three nodes form an equilateral triangle.

- (b) When $\hat{l}_{s,r} > \alpha \hat{l}_{s,t}$, $\hat{l}_{s,r} + \hat{l}_{r,t} < 2\hat{l}_{s,t}$ (e.g., θ_2 with s_4 and t_1). That is, when s is close to t but far from the root, our approach is not providing the optimal route.

In summary, only if the source and destination nodes are located in proximity and far from the root (only case 3.b above), our worst-case upper bound is lower than $\xi_{syn}^{UB}(s,t)$. Nevertheless, in this scenario, they are likely to be on the same branch of the DODAG and thus have a higher probability of having a much greater rate since they do not need to pass through the root. This hypothesis is verified through simulation in the subsequent section. Additionally, in the case where the three nodes are aligned in a line, if r is between s and t , $\hat{l}_{s,r} + \hat{l}_{r,t} = \hat{l}_{s,t}$. If r is not in between, they can always establish communication without going through the root. When traveling towards the root, the node farther away from the root would reach the closer node before reaching the root. However, any entanglement swapping between the closer node and the root would be pointless. Therefore, we can apply $\hat{l}_{s,r} + \hat{l}_{r,t} = \hat{l}_{s,t}$ in this scenario as well.

With that, it is obvious that $\xi_{DODAG}^{UB}(s,t) > \xi_{syn}^{UB}(s,t)$ holds when $p < q$.

Now, let us see the case where $p > q$. From $\xi_{syn}^{UB}(s,t)$ and $\xi_{DODAG}^{UB}(s,t)$, we have the relationship between \hat{l} and $\hat{l}_{s,r} + \hat{l}_{r,t}$:

1. If $\hat{l}_{s,t} \geq \beta(\hat{l}_{s,r} + \hat{l}_{r,t})$ where $\beta < 1$ (See the next subsection for the value of β with respect to p and q), $\xi_{syn}^{UB}(s,t) \leq \xi_{DODAG}^{UB}(s,t)$.
2. If $\hat{l}_{s,t} < \beta(\hat{l}_{s,r} + \hat{l}_{r,t})$, $\xi_{syn}^{UB}(s,t) > \xi_{DODAG}^{UB}(s,t)$.

Even though the second case above seems to suggest that the synchronous method has a higher upper bound on the entanglement rate, the above relationships are valid only because $\hat{l}_{s,r} + \hat{l}_{r,t} \geq \hat{l}_{s,t}$. However, we find that, in reality, it is most likely $\hat{l}_{s,r} + \hat{l}_{r,t} \approx \hat{l}_{s,t}$ when the distance between the source and destination is large. As shown in our simulation in Fig. 10, the paths' lengths (hops) tend to converge as the distance between the source and destination increases. That is, $\hat{l}_{s,r} + \hat{l}_{r,t} \approx \hat{l}_{s,t}$ when the distance between source and destination is large, which means that our approach has a significantly larger upper bound than asynchronous approaches when the distance between source and destination is large. This phenomenon can be explained by Fig. 9 to some extent (i.e., as the distance between the two communicating nodes increases, the two nodes and the root tend to align in a linear configuration if the root is in the center of the network, that is, the three nodes are more likely to form a line as distance increases. It is important to note that although the mean path length tends to decrease with larger distances between nodes, it will not be lower than the actual distance for obvious reasons. The number of network nodes influences the observed trend in a grid network. When approaching the maximum achievable distance between the source and destination, the mean path lengths for both approaches always converge. This convergence is demonstrated in Fig. 10 (a) and (b).). Also, this distance value of the convergence point becomes smaller when T_{co} becomes larger. With this fact, Equations 4 and 6 imply $\xi_{DODAG}(s,t) \gtrsim \xi_{syn}(s,t)$ when the distance between source and destination is large, which makes $\hat{l}_{s,r} + \hat{l}_{r,t} \approx \hat{l}_{s,t}$.

Note that Fig. 10 only represents the average length of paths found by the two protocols in an instant topology. It does not take into account the probability of finding a path. Path length

alone does not fully determine the entanglement rate. We must also consider the probability of finding a path in an instant topology to determine the entanglement rate. In our simulations, the number of successful path findings by asynchronous protocols is significantly higher than synchronous protocols, given the same number of simulation iterations. The averages in Fig. 10 are calculated by taking only the iterations with successful path findings. Therefore, despite potentially having longer paths, asynchronous protocols are more likely to find a path, leading to a higher entanglement rate in the end. Results pertaining to entanglement rate versus distances are presented in the next subsection.

Additionally, the simulations presented in Fig. 10 only account for scenarios where the source and destination nodes must pass through the root to establish a connection. However, in reality (and in subsequent simulations), it is common for closely located source and destination nodes to bypass the root and instead traverse a common ancestor. As a result, the mean path length for our approach is expected to be considerably lower than depicted in the figure (as demonstrated in the simulation results in Subsection VI).

Moreover, if we assume that there is at least one available path (where p and q are sufficiently high to exceed the probability threshold in bond percolation, such as $p > 0.5$ and $q = 1$ for a 2D square lattice⁴¹), we can establish lower bounds for $\xi_{syn}^{LB}(s,t) = p^{l_{s,t}} q^{l_{s,t}-1}$ and $\xi_{DODAG}^{LB}(s,t) = p^{l_{s,r} + l_{r,t}} q^{l_{s,r} + l_{r,t} - 1}$. If no direct-link entanglement is available in the instant topology, using a DODAG approach results in a worse lower bound since it requires two paths to achieve, the sum of which is often larger than the shortest path achievable in the existing synchronous approach. However, as the distance between the source and destination increases (network scale increases), the two approaches have similar lower bounds.

C. Relative positions of the three nodes in DODAG

1. Sides of an obtuse triangle

As shown in Fig. 11, let us denote the sides of the triangle formed by the three nodes as a , b and c . We show that the sum of the two sides adjacent to the obtuse angle of an obtuse triangle is less than twice the length of the third side. Since it is an obtuse triangle and c is the longest side, we have

$$a^2 + b^2 < c^2 \Rightarrow (a+b)^2 < c^2 + 2ab < 4c^2 \Rightarrow a+b < 2c \quad (8)$$

which means $\hat{l}_{s,r} + \hat{l}_{r,t} < 2\hat{l}_{s,t}$.

2. The value of α

As shown in Fig. 11, we denote the sides of the triangle formed by the three nodes as a' , b' and c' (when θ is acute). Based on the ASS (Angle-Side-Side) Theorem, we have

$$b' = a' \cos \theta \pm \sqrt{a'^2 - c'^2 \sin^2 \theta} \quad (9)$$

Asynchronous Entanglement Routing

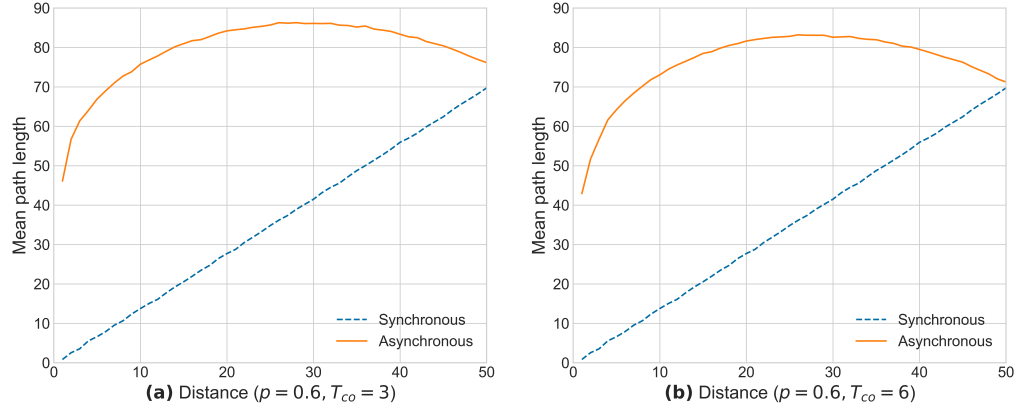


FIG. 10. **Mean path length when source and destination must pass through the root.** (a) With $p = 0.6, T_{co} = 3$ (with source and destination locations randomly generated), the average number of hops in a path in an asynchronous approach increases rapidly when the L1 distance (see Appendix D) between source and destination is small and becomes gentle. When the distance further increases, the mean hops decrease gradually. This happens because the two nodes and the root tend to form a line (setting the root in the center of the network) as the distance between the two communicating nodes increases. It is important to note that although the mean path length tends to decrease with larger distances between nodes, it will not be lower than the actual distance for obvious reasons. This trend scales as the number of network nodes increases, and at the maximum distance between the source and destination, the mean number of path lengths for both approaches always converges. In this figure, the network size is 26×26 , where the maximum distance is 50. Note that the value of q is irrelevant in this context as the graph solely illustrates the pathfinding in an instant topology without yet reaching the phase of implementing entanglement swapping. (b) The plot with the value of T_{co} set to 6.

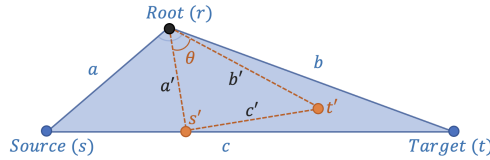


FIG. 11. **The triangle formed by r, s, t .** To simplify the symbols, we denote the sides of the triangle formed by the three nodes as a, b , and c .

Together with $a' + b' = 2c'$ (for the boundary that makes the relationship of $a' + b'$ and $2c'$ different) and solve for a' , we have

$$a' = c' \frac{2 \cos \theta + \sqrt{2} \sqrt{2 \cos^2 \theta + \cos \theta - 1} + 2}{2(\cos \theta + 1)} = \alpha c' \quad (10)$$

3. **The value of β**

Since we want to know the boundary of $\xi_{syn}^{UB}(s, t) = 4p^{\hat{l}_{s,t}} q^{\hat{l}_{s,t}-1}$ and $\xi_{DODAG}^{UB}(s, t) = 4q^{\hat{l}_{s,r} + \hat{l}_{r,t}-1}$ with respect to p and q . We solve for $\hat{l}_{s,t}$ by

$$4p^{\hat{l}_{s,t}} q^{\hat{l}_{s,t}-1} = 4q^{\hat{l}_{s,r} + \hat{l}_{r,t}-1} \quad (11)$$

Thus, we have

$$\hat{l}_{s,t} = (\hat{l}_{s,r} + \hat{l}_{r,t}) \frac{\log q}{\log p + \log q} = \beta (\hat{l}_{s,r} + \hat{l}_{r,t}) \quad (12)$$

VI. SIMULATION RESULTS

We conduct network simulations on the two approaches to support our observation of the theoretical performance analysis. We conduct simulation with single-path scenarios using NetworkX⁴², where we only find one optimal path for the source and destination in each unit time. We did the single-path scenarios since it is much faster to simulate in a large grid topology. A multi-path scenario is also shown at the end. The locations of the source and destination are generated randomly. The root node for DODAG is set at the center of the grid. To simulate the probabilistic behaviors of direct-link entanglement generation, we employ a grid network with edges that have assigned probabilities for availability. Subsequently, we simulate the entanglement swapping process within the probabilistic vertices of the network. Within this simulated grid network, we conduct 50,000 iterations of each protocol for random pairs located at the same distance, aiming to achieve end-to-end entanglement.

As shown in Fig. 12, the entanglement rates of asynchronous protocols (both DODAG and spanning tree) decay slower than synchronous approaches. We can tell that

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819

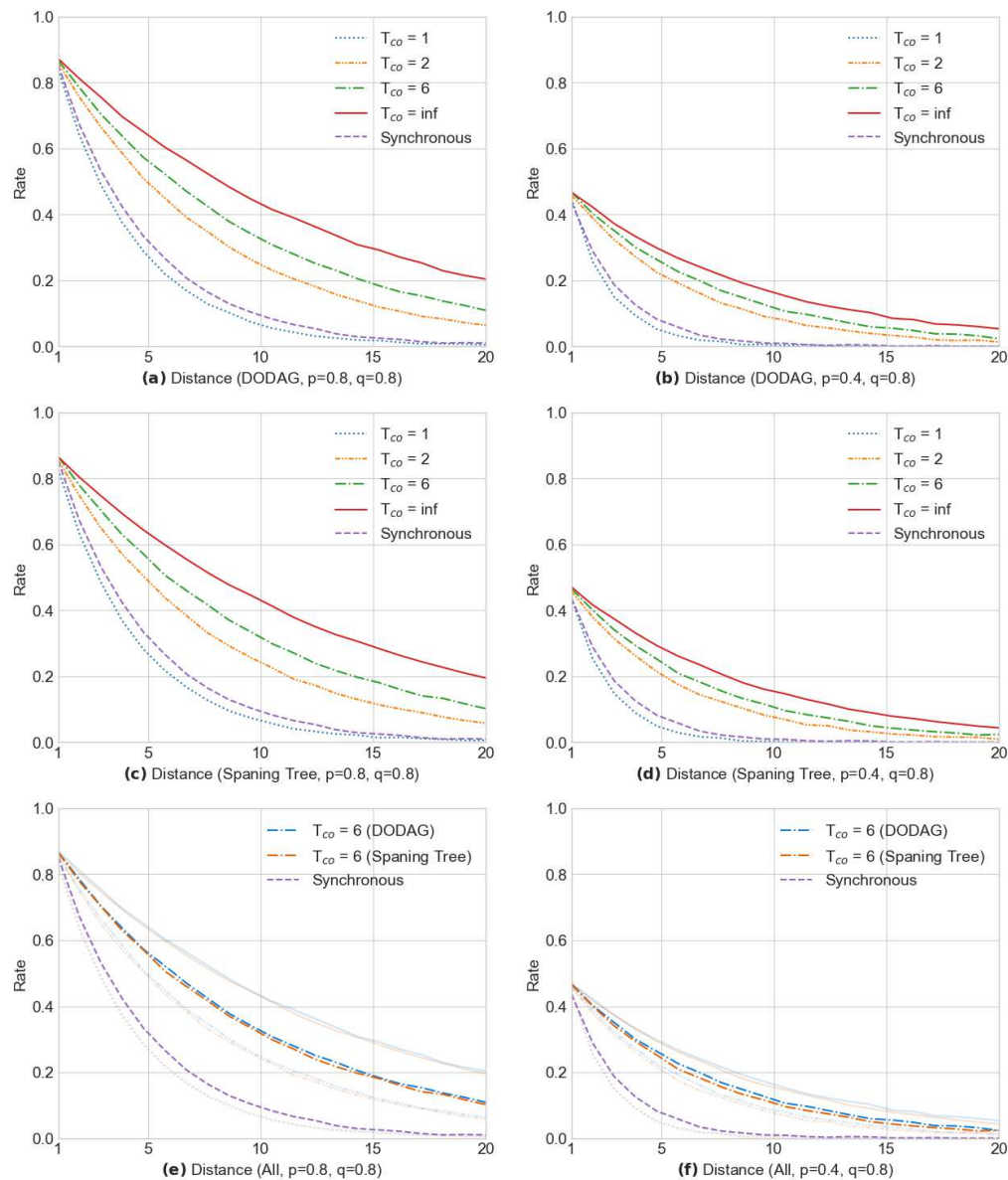


FIG. 12. **Simulation results: rate vs. distance.** (a), (c), and (d) show the entanglement rate behaviors when $p = q = 0.8$, and (b), (d), and (f) show those when $p = 0.4$ and $q = 0.8$ (with local knowledge of the instant topology). The first row, represented by (a) and (b), showcases the DODAG performance, while the second row, (c) and (d), highlights the spanning tree performance. These are compared to the performance of synchronous protocols, denoted by the dashed purple lines. The third row, (e) and (f), includes all protocols together and highlights asynchronous ones when $T_{co} = 6$.

DODAG and spanning tree approaches behave similarly (DODAG slightly better) in simulation. Both provide higher rates than synchronous approaches when the coherence time is greater than one unit time. We try different combinations of p and q in the simulation. In all combinations, asynchronous protocols have a higher entanglement rate and decay slower than synchronous ones when the coherence time is greater than one unit time. When coherence time is equal to one, asynchronous protocols perform similarly to synchronous protocols when the distance is small but slightly worse when the distance increases. The solid lines in the figure show the estimated upper bounds of the entanglement rates of an asynchronous protocol under the particular parameter settings, which is obtained by using infinite coherence time. The figure also shows that asynchronous protocol has a higher rate even in short-distance scenarios. This verifies our speculation in Section V B that close-by node pairs have a high chance of not needing to go through the root but only through a common ancestor. Note that the simulation results represent the average rate across multiple iterations, which introduces some inherent noise. Additionally, intermediate entanglements (i.e., the intermediate entanglements established towards end-to-end entanglements) set up through entanglement swapping will refresh the coherence time and are retained in asynchronous routing processes. Consequently, the rates in simulations at distance d may exceed q^d when using asynchronous protocols.

As shown in Fig. 13, the entanglement rate of asynchronous routing increases as coherence time increases. The performance of existing synchronous protocols such as²⁷ is independent of the coherence time. This indicates that when technology improves, asynchronous protocols will have a much more impact than synchronous protocols. Since the performance of DODAG and spanning tree protocols is similar in simulation, we only show the DODAG graph in this figure for concise. The plot also demonstrates the estimated upper bound of the asynchronous protocol under a particular q when the coherence time is infinite. For example, the upper bounds of the entanglement rate of an asynchronous protocol when $q = 0.8$ or $q = 0.4$ are shown with the solid line in Fig. 13. Nonetheless, we can see if we have $p = q = 1$, we do not need any asynchronous or synchronous protocols running in the case of having local knowledge of the instant topology. With that, we can run any shortest path algorithm using the physical topology. However, asynchronous protocols will still be useful, considering a quantum mobile network with dynamic physical topology (i.e., nodes may be moving). Moreover, with $T_{co} = 1$, the curves of asynchronous and synchronous protocols look just like the percolation probability in percolation theory and get increasingly obvious (more and more rapid transition around the threshold) when q gets larger.

In this article, we equate quantum state lifetime with the coherence time of entanglement, considering them synonymous due to the dependency of entanglement coherence time on qubit lifetime. As entanglement is sustained by quantum states, its coherence period is determined by how long the states remain coherent. Additionally, we posit that the usable lifetime of qubit—its duration of reusability for entanglement

generation—is sufficiently long. Utilizing unit time, which conveys a relative rather than absolute value, prevents us from specifying a minimum quantum state lifetime, prompting us to assume it to be suitably long.

Finally, we show the simulation results in multi-path scenarios (Fig. 14). Multi-path simulation means we search for more than one path (disjoint) for a pair of source and destination. We can tell asynchronous protocols have a higher entanglement rate than synchronous protocols in all scenarios when $T_{co} > 1$.

With the simulation results, we can establish the contributions of our asynchronous protocols:

1. The entanglement rate of asynchronous routing increases with the coherence time.
2. The entanglement rate of asynchronous routing decays slower than synchronous approaches.
3. The entanglement rate of asynchronous routing is larger than synchronous approaches in almost all scenarios.

VII. CONCLUSION

We envision quantum-native routing (i.e., routing for end-to-end entanglement) as the key component of the Quantum Internet and identify the disadvantages of synchronization used in existing quantum-native routing schemes. Thus, we proposed the design of the asynchronous routing scheme and benchmarked its performance under different parameter settings. Here, we highlight the differences between our asynchronous routing protocols and existing asynchronous protocols. Under the condition that a node only knows the direct-link entanglement states of its adjacent links, existing approaches use synchronized time slots, including two execution phases to collaborate all nodes to search for a path connecting the source and destination nodes. This method consumes all direct-link entanglements, trying to find multiple paths between them. However, the consumption of all direct-link entanglements counters the entanglement rate. Our method removes such synchronization of execution phases by introducing graph maintenance in a distributed manner. We treat all network nodes as vertices in a graph that try to connect their neighbors using direct-link entanglements. No central control and no global knowledge of the direct-link entanglements are needed in our method. Whenever a pair of nodes wants to create an end-to-end entanglement, they go through the distributed graph that is continuously updated to find each other. With analysis and simulation, we found that the entanglement rate of our asynchronous protocols increases with the coherence time, decays slower than and is generally larger than that of asynchronous approaches. Furthermore, asynchronous protocols can simultaneously handle multiple user pairs, which remains an open problem in synchronous schemes²⁷. Exploring the performance of multi-user scenarios holds significant prospects for future research.

Moreover, existing approaches treat repeater nodes equally. However, repeaters have different performances (they can

Asynchronous Entanglement Routing

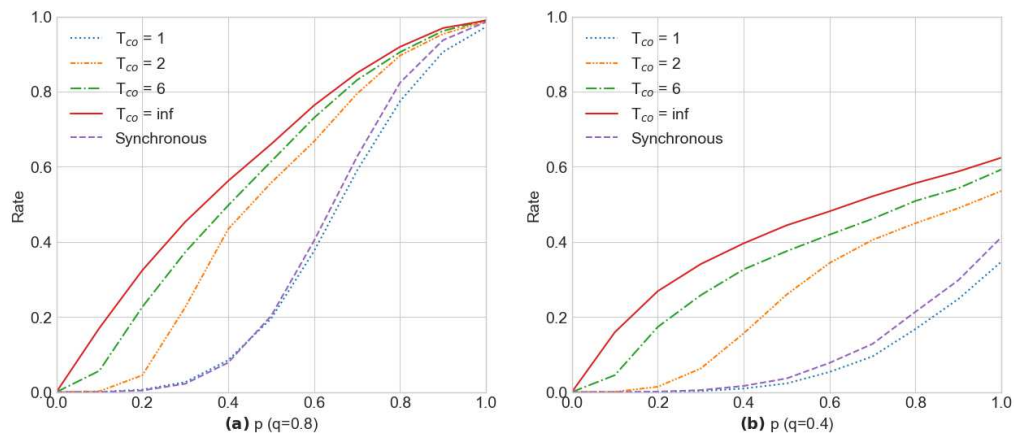


FIG. 13. **Simulation results: rate vs. p .** The two diagrams show the relationship between entanglement rate and p in the two types of protocols. For $q = 0.8$, this relationship is depicted in (a), and for $q = 0.4$, it is presented in (b). The coherence time T_{co} is varied across the set $\{1, 2, 6, \infty\}$. When the coherence time is greater than one, the asynchronous protocol has higher rates in all values of p . When the coherence time equals one unit time, the asynchronous protocol performs similarly to the synchronous protocol, and its entanglement rate is slightly smaller as p increases. In both situations, the entanglement rate of asynchronous routing increases as coherence time increases, but the synchronous protocol does not.

have weights for edges, e.g., transmissivity, for routing, but no preference for using competent nodes for more important purposes). Even though this is not the primary purpose, DODAG can reflect the node robustness by preferring to set robust nodes as root nodes. Also, DODAG roots can serve as well-fit gateways to connect multiple networks (i.e., networks of networks). Since DODAG has such benefits that a spanning tree does not have and has similar performance, we recommend DODAG to be the primary protocol for asynchronous routing, which is also well-studied and standardized for low-cost wireless networks. Additionally, the scalability of the DODAG network can be enhanced by constructing hierarchical DODAGs, which involve creating trees of trees. This direction holds potential for future exploration.

There are other aspects of the protocol that warrant further investigation. For example, the center of a network is a good position for the root node, but the impact of different root positions remains future work. Moreover, the strategies to arrange multiple roots' positions and to determine how fine-grain (the density) the partitioned networks should be (distance between adjacent root nodes) remain future work. Last but not least, other aspects of entanglement-swapping-based routing are also important and under investigation, such as the use of untrusted repeaters, the minimal trust between networks (i.e., no knowledge of the internals of another network), the possible presence of malicious nodes, and the order of entanglement swapping along a path, etc. Furthermore, exploring the performance of asynchronous protocols in non-grid networks is worthwhile. Through this work, we aim to raise awareness and foster advancements in networking techniques

for the Quantum Internet.

ACKNOWLEDGMENTS

This work was supported in part by Cisco University Research Gift #86944165.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

Appendix A: Quantum network stack

We have identified seven layers within the technical framework for quantum research, providing a schematic to grasp the diverse elements essential to quantum technology, as illustrated in Fig. 15. At the foundational level, the stack is anchored by diverse technologies, which include photon-based and semiconductor-based approaches³, critical for quantum

Asynchronous Entanglement Routing

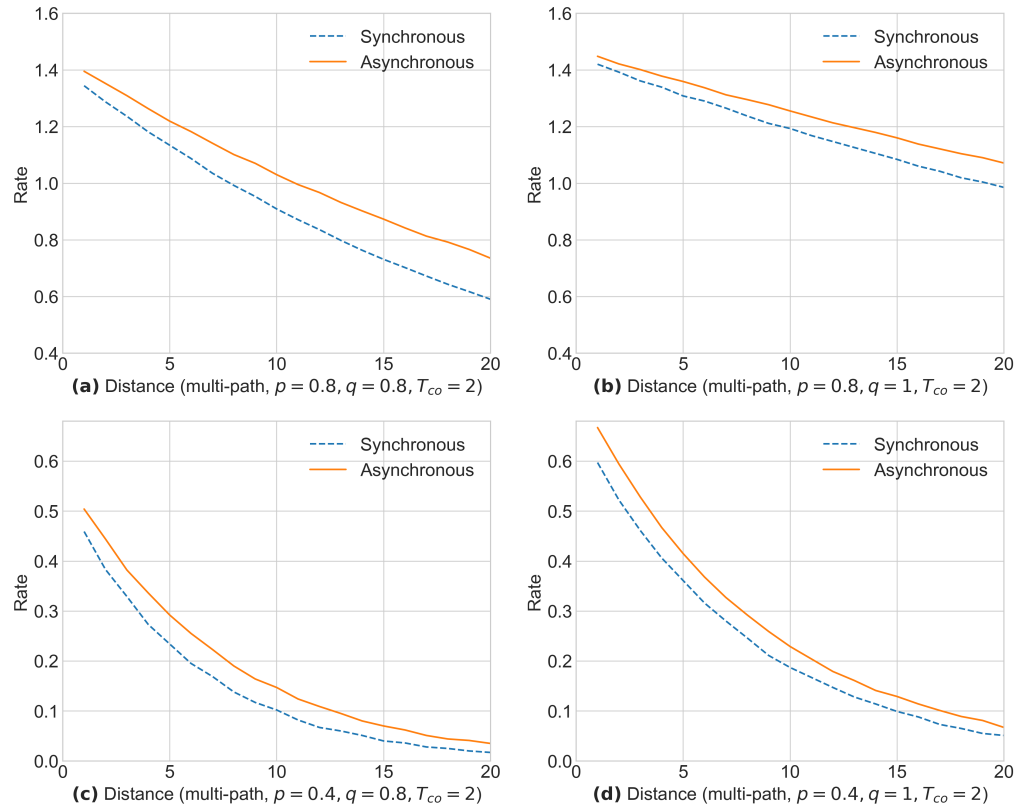


FIG. 14. **Multi-path simulation results: rate vs. p.** Previous diagrams demonstrate the performance of single-path scenarios. This figure shows the performance comparisons of multi-path scenarios of the protocols. In varying scenarios with different p , q , and T_{co} values, asynchronous protocols consistently outperform synchronous ones in terms of entanglement rate. Specifically, for $T_{co} = 2$ and $p = q = 0.8$, the result is shown in (a). The scenarios " $p = 0.8, q = 1$," " $p = 0.4, q = 0.8$," and " $p = 0.4, q = 1$ " are illustrated in (b), (c), and (d) respectively. Overall, our protocol provides larger rates than synchronous approaches in all scenarios when $T_{co} > 1$. Note that when $q = 1$ and $q > 0.5$ (the bond percolation threshold for the square lattice⁴⁰), the distance affects less on the rate. The rate is not distance independent (it actually is when considering only one path) because there are fewer alternative paths as distance increases. This behavior conforms to what is expected in percolation theory.

states and operations. These technologies interface with various media, such as optical fiber and free space, such as satellites, providing the physical medium for quantum communication and interactions. Nodes in the quantum network, which encompass end nodes and repeater Nodes, form the fundamental infrastructure, facilitating quantum computation and communication processes. In the computation realm, universal quantum computing (i.e., gate model) and Quantum Annealing are the primary mechanisms that execute quantum algorithms and solve complex problems. Systems and protocols are the backbone of any quantum network. While TCP/IP represents conventional networking standards, the stack in-

troduces quantum-native protocols with unique functionalities like entanglement swapping and asynchronous routing. At the Applications layer, solutions such as quantum cloud services, quantum blockchains, quantum artificial intelligence (AI), and quantum cryptography showcase the potential real-world uses of quantum technologies^{5,43-46}. Lastly, at the markets layer, the stack emphasizes the wide-ranging sectors that quantum technologies can influence, e.g., computing, networking, clouds, FinTech, medical, and sensing, underlining the vast impact quantum advancements can have across industries.

Our observation shows that while substantial progress has

Asynchronous Entanglement Routing

Markets	Computing, Networking, Cloud Computing, FinTech, Medical, Environment Sensing, ...			
Applications	Quantum cloud services	Quantum-Native Blockchain	Quantum-Native AI	Quantum-Native Cryptography
Systems and Protocols	TCP/IP (Forwarding)	Quantum-Native Networking Protocols		Security Management
	Quantum ECC	Quantum-Native Routing	Entanglement Swapping + Purification	
Computation	Universal Quantum Computing (Gate)		Quantum Annealing	Security Management
Nodes	End Nodes, Repeater Nodes, ...			
Media	Optical Fiber, Free Space, Satellites, ...			
Technologies	Photon-based, Semiconductor-based, ...			

FIG. 15. 7-layer stack of quantum computing research.

been made in the lower layers, the upper layers have received comparatively less attention. These layers reflect research initiatives in physics and electrical engineering^{1,6}. In contrast, the upper layers indicate endeavors in software, encompassing network protocols and applications. In particular, within the layer of systems and protocols, there are two distinct approaches: one involves using quantum repeaters that transmit quantum information (left of the red dashed line). The transmission of quantum information through repeaters requires error correction using a significant number of error-correcting qubits, which is restricted by the current capabilities of quantum computers. The other involves distributing entanglement as a communication resource (i.e., quantum-native routing), which is our primary interest and, in our view, deserves greater attention from the research community.

Appendix B: DODAG loop avoidance

When nodes select routes in a graph, loops can cause the routing process not to conclude, wasting time and resources. Therefore, they must be detected and avoided. While a directed acyclic graph (DAG) does not form closed loops, a DODAG may do so if a node is greedy and wants to select as many parents as possible. To prevent this, a rank-based non-greedy mechanism is used to detect loops before allowing a node to join the DODAG.

If a node in the DODAG is greedy and wants to move deeper (closer to the root) by increasing its number of parent nodes, there is a risk of creating a loop. Consider Fig. 16a, which depicts the initial state of a branch of a DODAG, where node *A* has two children, *B* and *C*. Let us assume that *A* is the closest node to the root among the three, making it the preferred parent of *B* and *C*. If nodes *B* and *C* are both eager to select more parents, a loop may occur. For example, node *C* may want to choose both *A* and *B* as parents, making it deeper than both *A* and *B*, as shown in Fig. 16b. Suppose node *B* is also greedy and decides to select *C* as a parent against the non-greedy rule. In this case, node *B* will intentionally leave the DODAG and then rejoin at a lower rank, taking both *A* and *C* as parents, as illustrated in Fig. 16c. As a result, node *B* will become deeper than both *A* and *C*. Node *C*, who is also greedy, will then leave and rejoin at a lower rank to get two

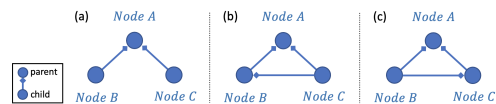


FIG. 16. DODAG greedy parent selection. (a) The initial state of a branch of a DODAG, where node *A* has two children, *B* and *C*. (b) Node *C* chooses both *A* and *B* as parents. (c) Node *B* intentionally leaves the DODAG and then rejoins at a lower rank, taking both *A* and *C* as parents. In this case, (b) and (c) would create an infinite cycle.

parents and have a lower rank than both of them. Node *B* will repeat the process by leaving and rejoining at a lower rank, which will cause the DODAG to alternate between Fig. 16b and c, creating an infinite cycle. This situation can be prevented by ensuring nodes *B* and *C* do not select parents from deeper nodes or leave the DODAG intentionally to acquire more parents. Instead, they should maintain a reasonable rank relative to their preferred parent, *A*. Additionally, they should not process parenting requests from deeper nodes. In the example, as long as node *B* does not depart from the DODAG solely to acquire two parents, the branch of the DODAG will remain in the state depicted in Fig. 16b.

Appendix C: GHS delay scenarios

In Fig. 17a, (1) Node *a* sends a "test" message to Node *b*, but they cannot determine which ID is the latest because $Lv(a) > Lv(b)$. (2) Consequently, they put the edge on hold. (3) Meanwhile, another node, *k*, initiates a "test" message to Node *n*. (4) Since $Lv(k) < Lv(n)$, an outgoing edge is established. (5) Node *k* propagates the new level and ID to other nodes in the same fragment. (6) Node *b* now knows that its latest level is 2 and its latest ID is *A*. (7) Nodes *a* and *b* also realize they have the same ID and leave the edge alone. In Fig. 17b, a similar process leads to finding an outgoing edge successfully. These processes use level increases to avoid broadcast delay, which may cause loops or unnecessary connections. Thus, fragment splitting while maintaining the level value will not affect the merge or absorb operations in the GHS algorithm. If a link breaks during broadcasting, the on-hold edge will remain on hold until the adjacent nodes broadcast the breaking message. Then, it will release the on-hold status and start the next "test" message. If edge 'k-f' breaks in Figures 17a and b, only process (2) will remain on hold until the node knows it is isolated. On the other hand, if edge 'i-j' breaks, Fragment A splits into two fragments with different IDs, but it will not impact any operations from (1) to (7).

Appendix D: Mean path length approximation

We can examine the physical 2D grid topology by representing it as a 2D square lattice *G* comprising of *n* nodes where $\sqrt{n} \in \mathbb{Z}$. We consider the L1 distance between every

Asynchronous Entanglement Routing

19

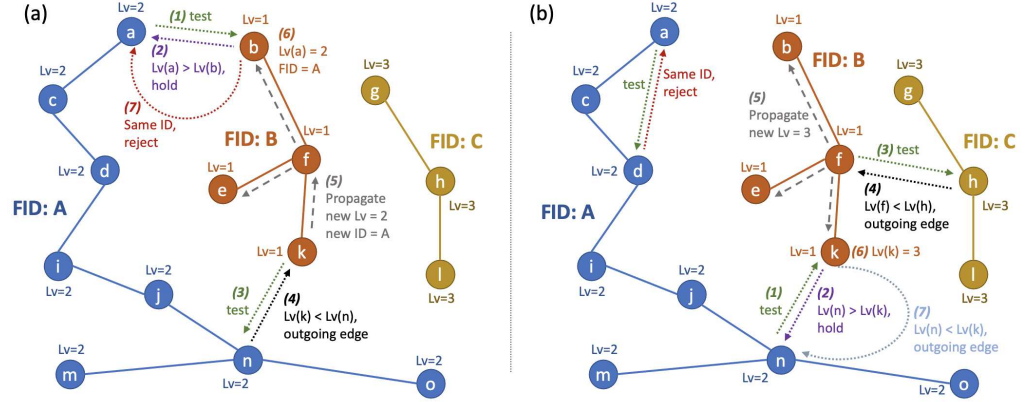


FIG. 17. **Fragment combination in the GHS algorithm.** The procedures labeled (1) through (7) operate following the rules outlined in Section IV B. Two scenarios are described as follows: (a) The “test” message from Node a to b is rejected due to the outgoing edge determined at (4). However, if no level value is available to determine whether an edge is outgoing, a loop will be formed through nodes ‘a-b-f-k-n-j-i-d-c-a.’ (b) If the level of the node that initiates a “test” message is less than or equal to the level of the recipient node, then it is safe to consider the edge in-between them outgoing. No loop will be formed in this case. An entanglement link can safely be created for this edge.

two nodes within graph G . The L1 distance for nodes s and t with coordinates $s = (x_s, y_s)$ and $t = (x_t, y_t)$ is determined as follows:

$$d(s, t) = d((x_s, y_s), (x_t, y_t)) = |x_s - x_t| + |y_s - y_t| \quad (\text{D1})$$

The length of the m th shortest self-avoiding path between s and t is as follows (c.f., Reference³⁹):

$$l_{s,t}^m = d(s, t) + 2(m - 1) \quad (\text{D2})$$

Let us denote the total number of self-avoiding paths of length l_m between s and t as $N_{l_m}^{s,t}$. This value can be obtained by the piecewise path enumeration algorithm introduced in Reference³⁹. Then, we can represent the mean length of all the possible paths with maximum length l_m between s and t as

$$l_{s,t} = \frac{\sum_{i=1}^m l_i N_{l_i}^{s,t}}{\sum_{i=1}^m N_{l_i}^{s,t}} \quad (\text{D3})$$

Assuming l_m represents the longest path length between nodes s and t in G , the above expression corresponds to the average length of all possible paths connecting them. Nevertheless, to simplify the computation without compromising the outcome, we can assign a sufficiently large fixed value to m without necessarily determining the precise average length of all possible paths. Note that the maximum value of m is proportional to n .

¹H. J. Kimble, “The quantum internet,” *Nature* **453**, 1023–1030 (2008).

²S. Wehner, D. Elkouss, and R. Hanson, “Quantum internet: A vision for the road ahead,” *Science* **362**, eaam9288 (2018), publisher: American Association for the Advancement of Science.

³Z. Yang, M. Zolanvari, and R. Jain, “A survey of important issues in quantum computing and communications,” *IEEE Communications Surveys & Tutorials* **25**, 1059–1094 (2023).

⁴A. K. Ekert, “Quantum cryptography based on Bell’s theorem,” *Physical Review Letters* **67**, 661–663 (1991), publisher: American Physical Society.

⁵H.-K. Lo, M. Curty, and K. Tamaki, “Secure quantum key distribution,” *Nature Photonics* **8**, 595–604 (2014).

⁶C. L. Degen, F. Reinhard, and P. Cappellaro, “Quantum sensing,” *Reviews of Modern Physics* **89**, 035002 (2017), publisher: American Physical Society.

⁷N. Gisin and R. Thew, “Quantum communication,” *Nature Photonics* **1**, 165–171 (2007).

⁸P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin, “A quantum network of clocks,” *Nature Physics* **10**, 582–587 (2014).

⁹R. Beals, S. Brierley, O. Gray, A. W. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather, “Efficient distributed quantum computing,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **469**, 20120686 (2013), publisher: Royal Society.

¹⁰R. Van Meter and S. J. Devitt, “The path to scalable distributed quantum computing,” *Computer* **49**, 31–42 (2016).

¹¹K. Azuma, K. Tamaki, and H.-K. Lo, “All-photonic quantum repeaters,” *Nature Communications* **6**, 6787 (2015).

¹²D. Niu, Y. Zhang, A. Shabani, and H. Shapourian, “All-photonic one-way quantum repeaters,” (2022), arXiv:2210.10071.

¹³D. A. Lidar and T. A. Brun, *Quantum error correction* (Cambridge University Press, 2013) google-Books-ID: XV9sAAAAQBAJ.

¹⁴S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, “Optimal architectures for long distance quantum communication,” *Scientific Reports* **6**, 20463 (2016).

¹⁵H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, “Quantum repeaters: the role of imperfect local operations in quantum communication,” *Physical Review Letters* **81**, 5932–5935 (1998), publisher: American Physical Society.

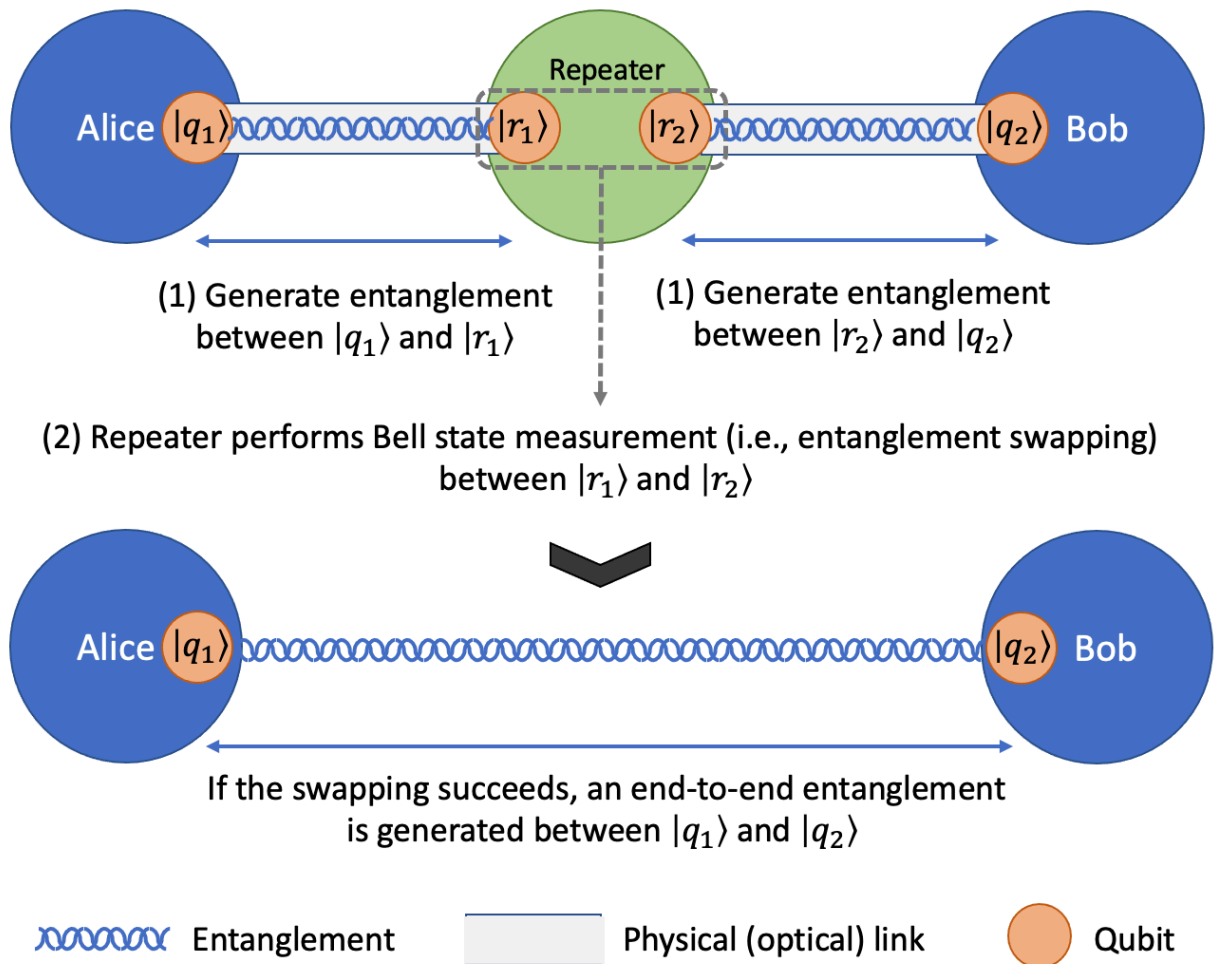
¹⁶J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger, “Experimental entanglement swapping: entangling photons that never interacted,” *Physical Review Letters* **80**, 3891–3894 (1998), publisher: American Physical Society.

¹⁷B. M. Terhal, “Quantum error correction for quantum memories,” *Reviews of Modern Physics* **87**, 307–346 (2015), publisher: American Physical Society.

- ¹⁸J. Preskill, "Quantum computing in the NISQ era and beyond." *Quantum* **2**, 79 (2018), publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.
- ¹⁹P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, "Deterministic delivery of remote entanglement on a quantum network," *Nature* **558**, 268–273 (2018).
- ²⁰A. Tchebotareva, S. L. N. Hermans, P. C. Humphreys, D. Voigt, P. J. Harmsma, L. K. Cheng, A. L. Verlaan, N. Dijkhuizen, W. de Jong, A. Dréau, and R. Hanson, "Entanglement between a diamond spin qubit and a photonic time-bin qubit at telecom wavelength," *Physical Review Letters* **123**, 063601 (2019), publisher: American Physical Society.
- ²¹A. Dréau, A. Tchebotareva, A. E. Mahdaoui, C. Bonato, and R. Hanson, "Quantum frequency conversion of single photons from a Nitrogen-Vacancy center in diamond to telecommunication wavelengths," *Physical Review Applied* **9**, 064031 (2018), publisher: American Physical Society.
- ²²S. Wengerowsky, S. K. Joshi, F. Steinlechner, J. R. Zichi, B. Liu, T. Scheidl, S. M. Dobrovolskiy, R. v. d. Molen, J. W. N. Los, V. Zwiller, M. A. M. Versteegh, A. Mura, D. Calonico, M. Inguscio, A. Zeilinger, A. Xuereb, and R. Ursin, "Passively stable distribution of polarisation entanglement over 192 km of deployed optical fibre," *npj Quantum Information* **6**, 1–5 (2020).
- ²³S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, "Quantum repeaters and quantum key distribution: Analysis of secret-key rates," *Physical Review A* **87**, 052315 (2013), publisher: American Physical Society.
- ²⁴W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, "Quantum repeaters based on entanglement purification," *Physical Review A* **59**, 169–181 (1999), publisher: American Physical Society.
- ²⁵A. Acín, J. I. Cirac, and M. Lewenstein, "Entanglement percolation in quantum networks," *Nature Physics* **3**, 256–259 (2007).
- ²⁶S. Perseguers, M. Lewenstein, A. Acín, and J. I. Cirac, "Quantum random networks," *Nature Physics* **6**, 539–543 (2010).
- ²⁷M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Information* **5**, 1–9 (2019).
- ²⁸C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *npj Quantum Information* **7**, 1–12 (2021).
- ²⁹A. Patil, M. Pant, D. Englund, D. Towsley, and S. Guha, "Entanglement generation in a quantum network at distance-independent rate," *npj Quantum Information* **8**, 1–9 (2022).
- ³⁰S. Shi and C. 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*, SIGCOMM '20 (Association for Computing Machinery, New York, NY, USA, 2020) pp. 62–75.
- ³¹M. Caleffi, "Optimal routing for quantum networks," *IEEE Access* **5**, 22299–22312 (2017).
- ³²F. Hahn, A. Pappa, and J. Eisert, "Quantum network routing and local complementation," *npj Quantum Information* **5**, 1–7 (2019).
- ³³S. DiAdamo, B. Qi, G. Miller, R. Kompella, and A. Shabani, "Packet switching in quantum networks: A path to the quantum Internet," *Physical Review Research* **4**, 043064 (2022), publisher: American Physical Society.
- ³⁴D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, "On the complexity of and algorithms for finding the shortest path with a disjoint counterpart," *IEEE/ACM Transactions on Networking* **14**, 147–158 (2006).
- ³⁵K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," (2019), arXiv:1907.11630.
- ³⁶R. Alexander, A. Brandt, J. P. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," Request for Comments RFC 6550 (Internet Engineering Task Force, 2012).
- ³⁷R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and systems (TOPLAS)* **5**, 66–77 (1983), publisher: ACM New York, NY, USA.
- ³⁸E. Korach, S. Moran, and S. Zaks, "Tight lower and upper bounds for some distributed algorithms for a complete network of processors," in *Proceedings of the third annual ACM symposium on Principles of distributed computing*, PODC '84 (Association for Computing Machinery, New York, NY, USA, 1984) pp. 199–207.
- ³⁹O. Malik, X. Meng, S. Havlin, G. Korniss, B. K. Szymanski, and J. Gao, "Concurrence percolation threshold of large-scale quantum networks," *Communications Physics* **5**, 1–11 (2022).
- ⁴⁰S. R. Broadbent and J. M. Hammersley, "Percolation processes: I. Crystals and mazes," *Mathematical Proceedings of the Cambridge Philosophical Society* **53**, 629–641 (1957), publisher: Cambridge University Press.
- ⁴¹H. Kesten, "The critical probability of bond percolation on the square lattice equals 1/2," *Communications in Mathematical Physics* **74**, 41–59 (1980).
- ⁴²A. Hagberg, D. Schult, and P. Swart, "NetworkX reference," Release 3.0 (2023), https://networkx.org/documentation/stable/_downloads/networkx_reference.pdf.
- ⁴³Z. Yang, H. Alfauri, B. Farkiani, R. Jain, R. Di Pietro, and A. Erbad, "A survey and comparison of post-quantum and quantum blockchains," *IEEE Communications Surveys & Tutorials* (2023), 10.1109/COMST.2023.3325761.
- ⁴⁴Z. Yang, T. Salman, R. Jain, and R. D. Pietro, "Decentralization using quantum blockchain: A theoretical analysis," *IEEE Transactions on Quantum Engineering* **3**, 1–16 (2022).
- ⁴⁵M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "Trust xai: Model-agnostic explanations for ai with a case study on iiot security," *IEEE Internet of Things Journal* **10**, 2967–2978 (2023).
- ⁴⁶T. Renduchintala, H. Alfauri, Z. Yang, R. D. Pietro, and R. Jain, "A Survey of Blockchain Applications in the FinTech Sector," *Journal of Open Innovation: Technology, Market, and Complexity* **8**, 185 (2022), number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
- ⁴⁷D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," in *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, *Fundamental Theories of Physics*, edited by M. Kafatos (Springer Netherlands, Dordrecht, 1989) pp. 69–72.
- ⁴⁸P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing* **26**, 1484–1509 (1997), publisher: Society for Industrial and Applied Mathematics.

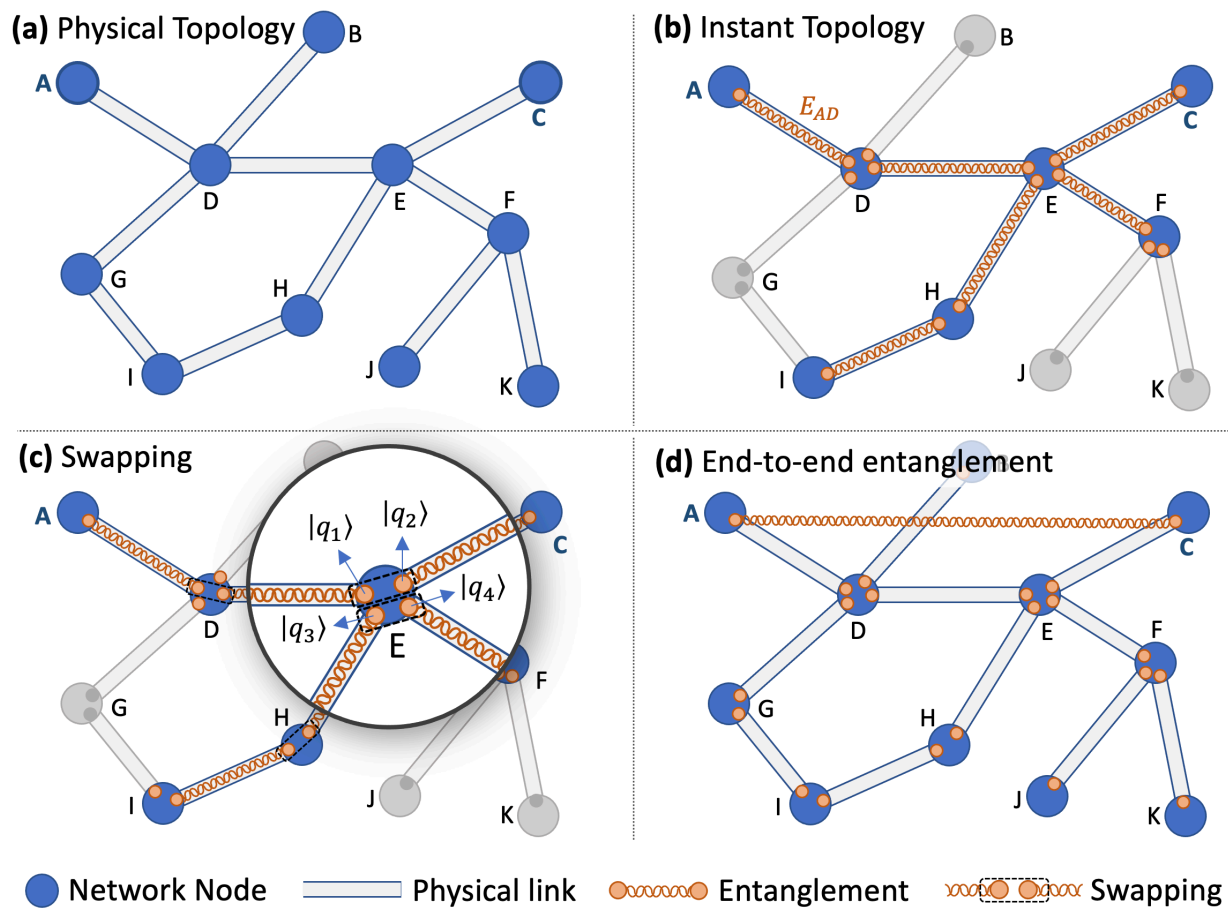
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



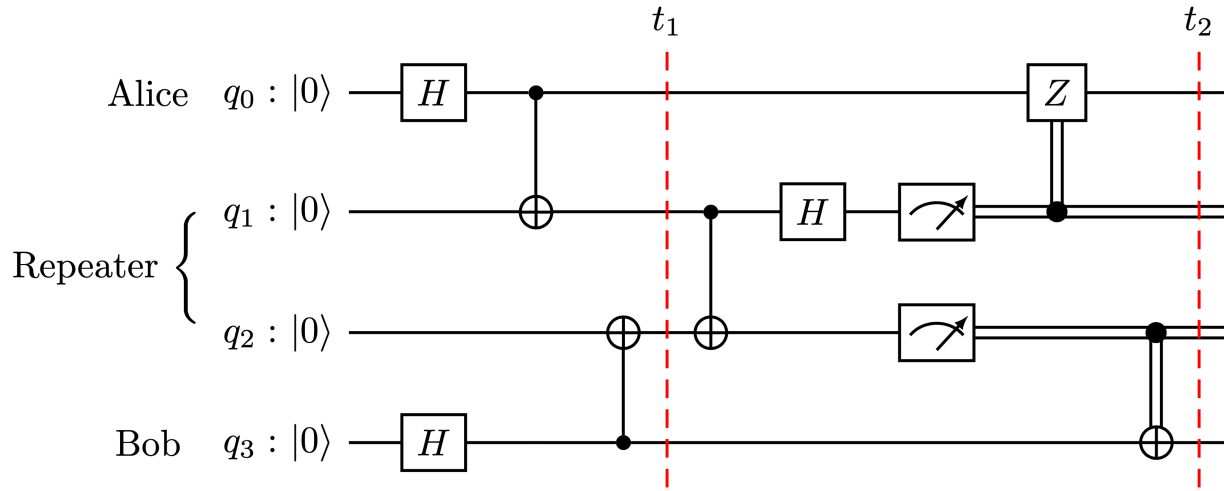
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



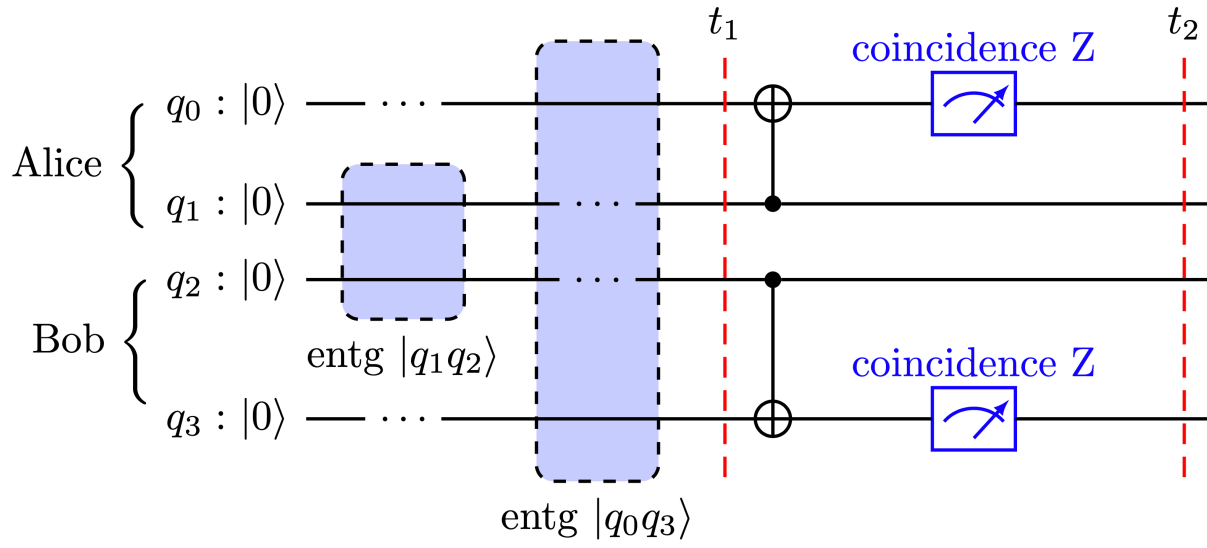
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



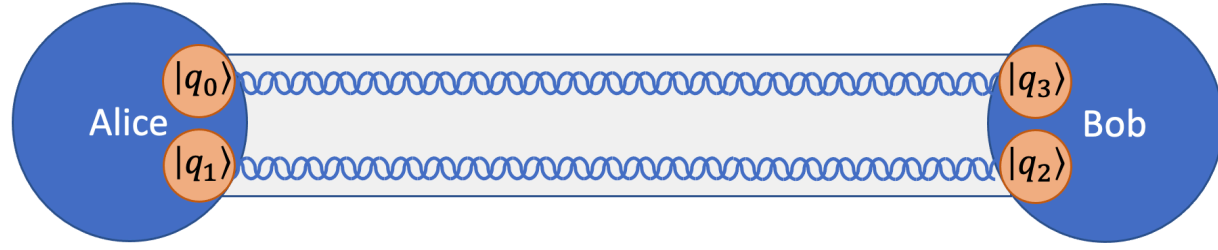
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



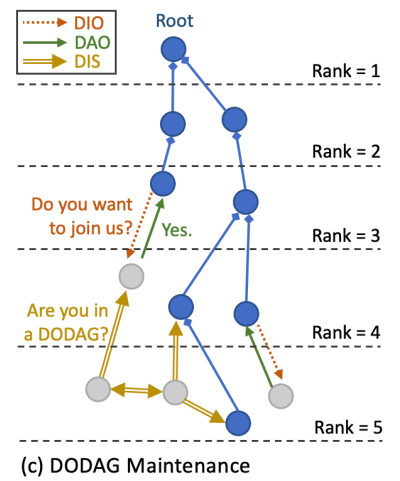
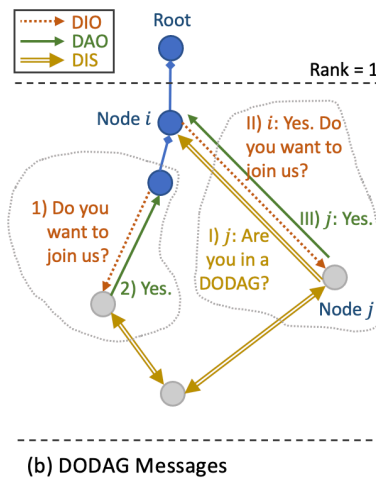
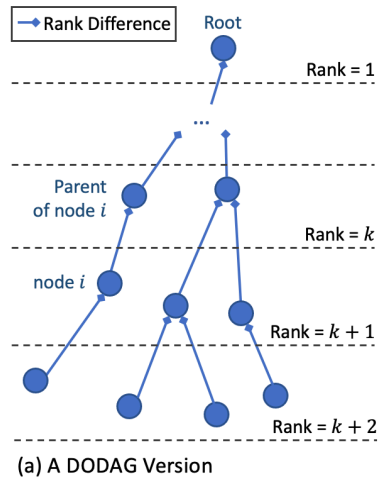
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



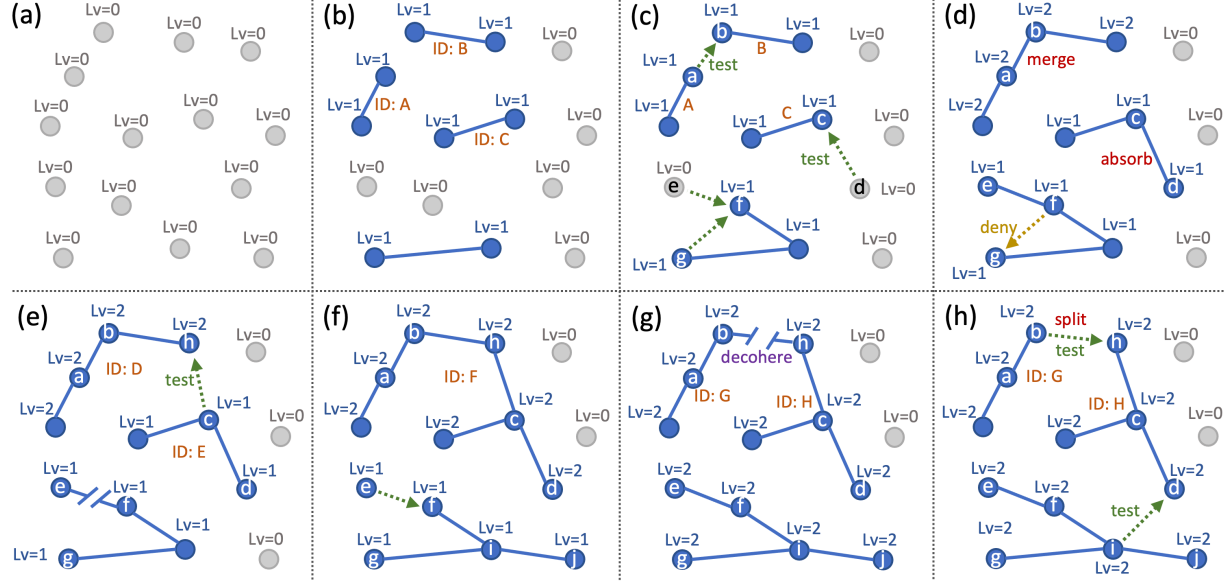
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



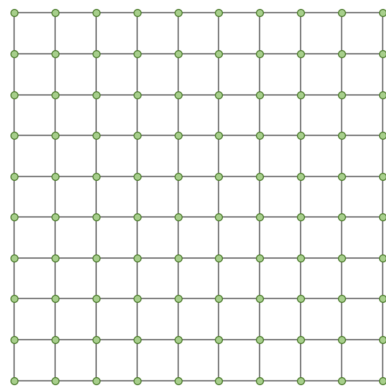
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819

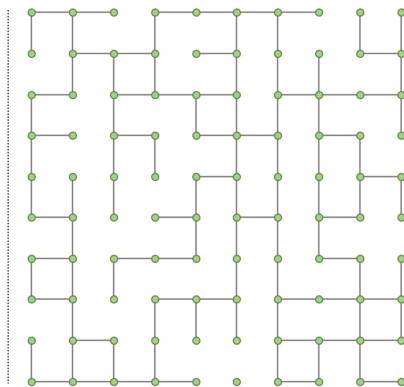


This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

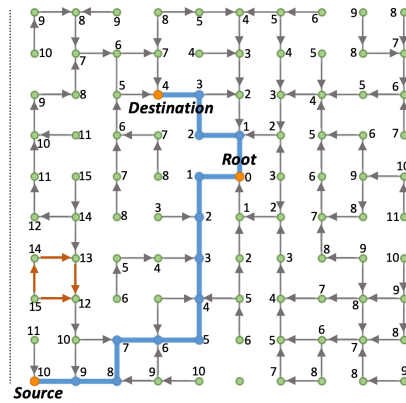
PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



(a) The physical topology: a grid



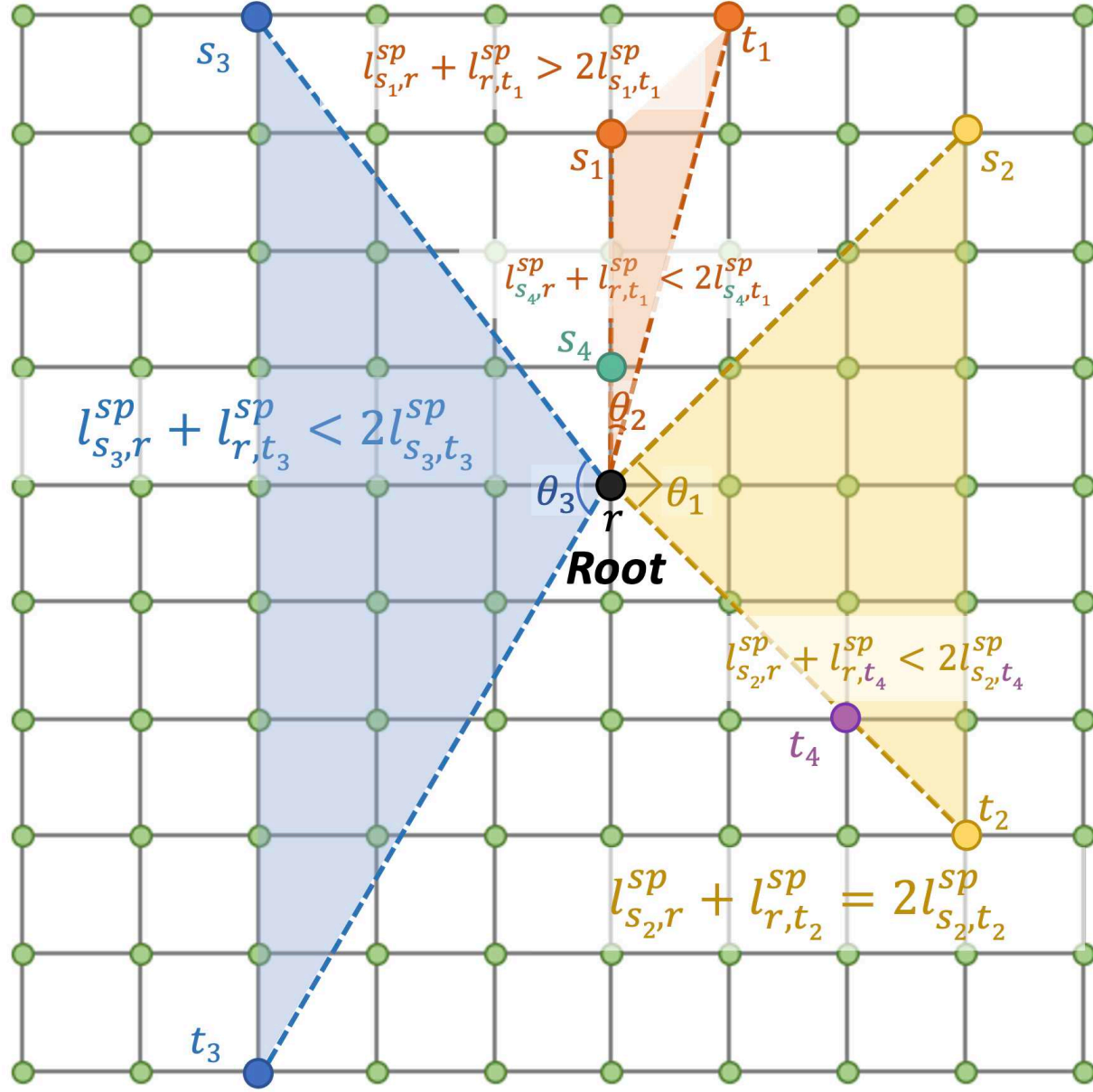
(b) A virtual topology: a subgraph



(c) A path in an instant topology

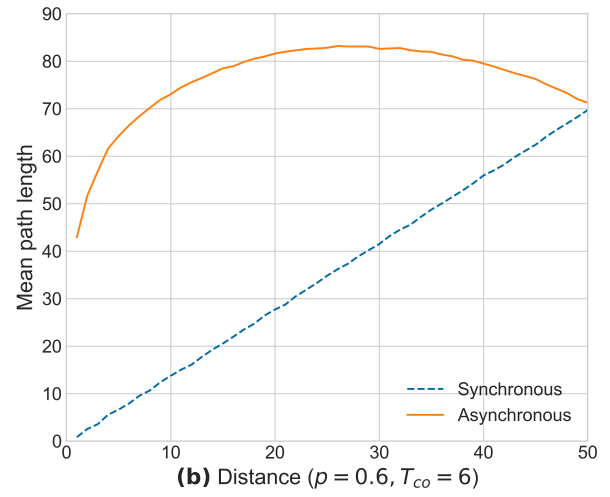
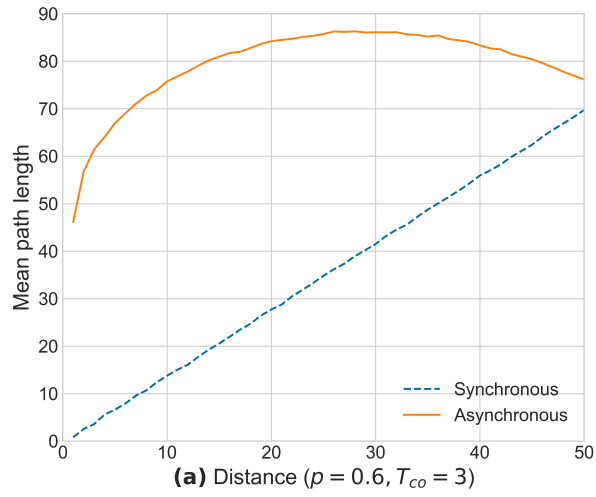
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/1.50172819



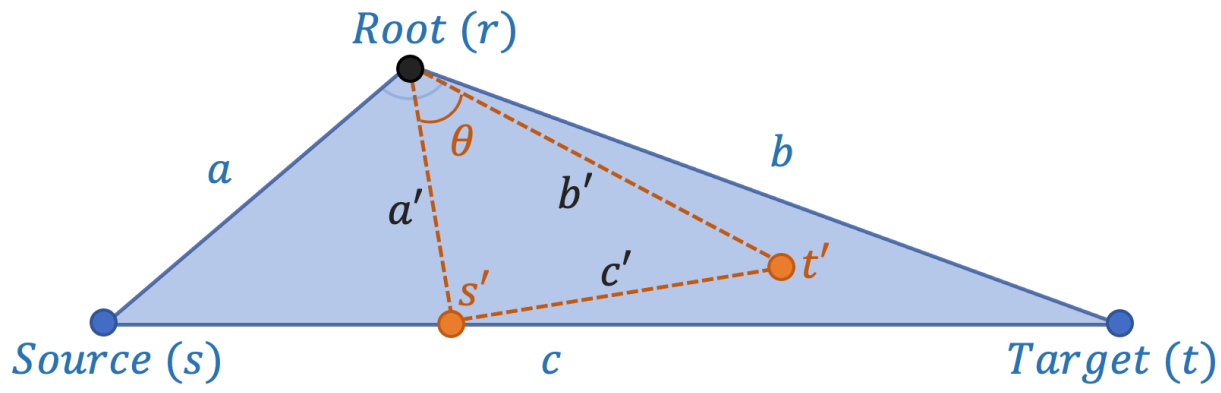
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



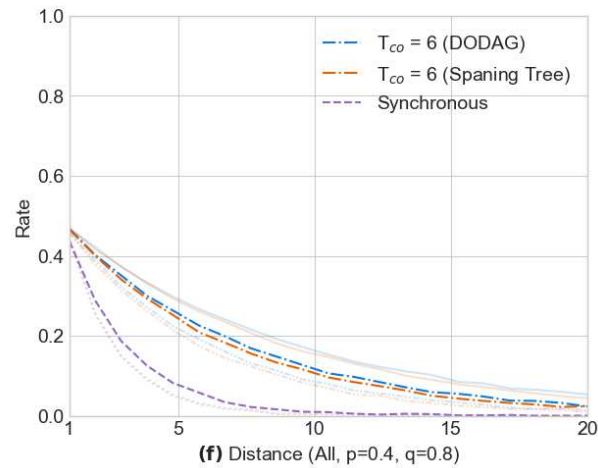
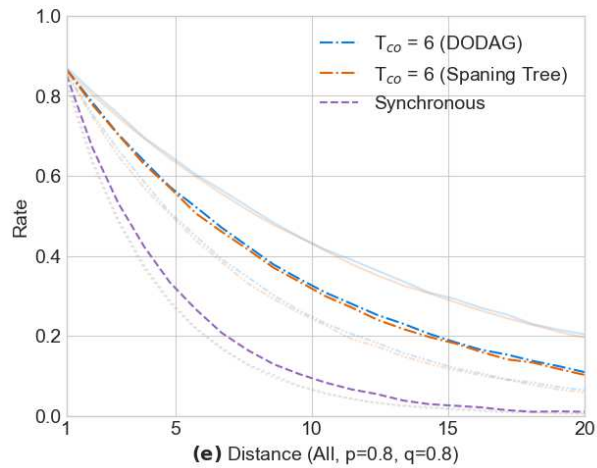
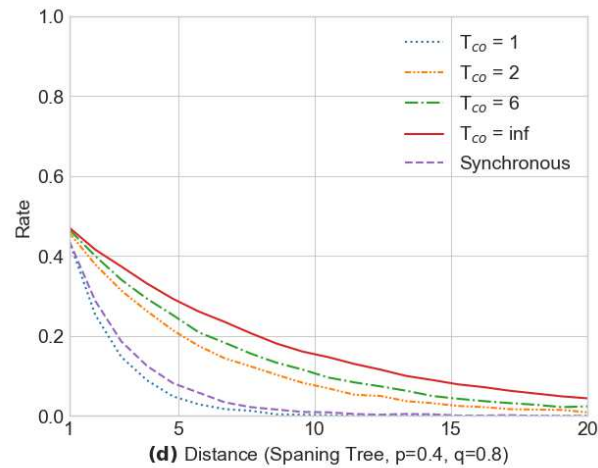
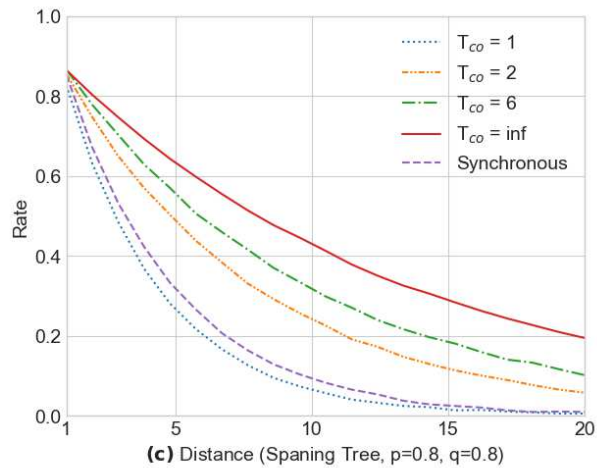
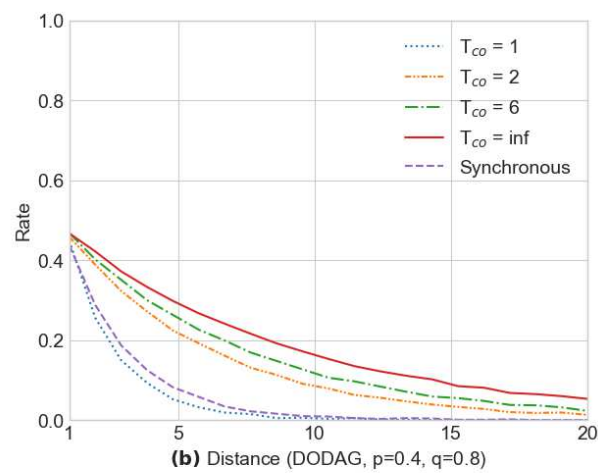
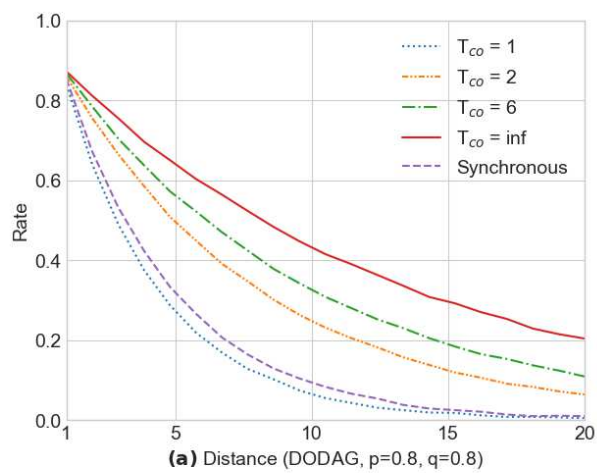
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



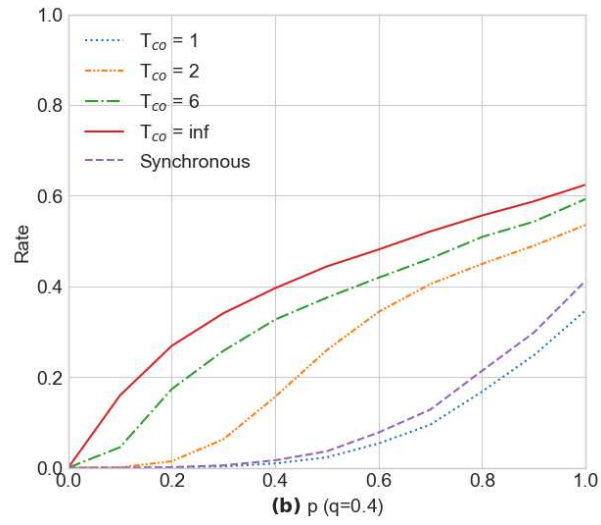
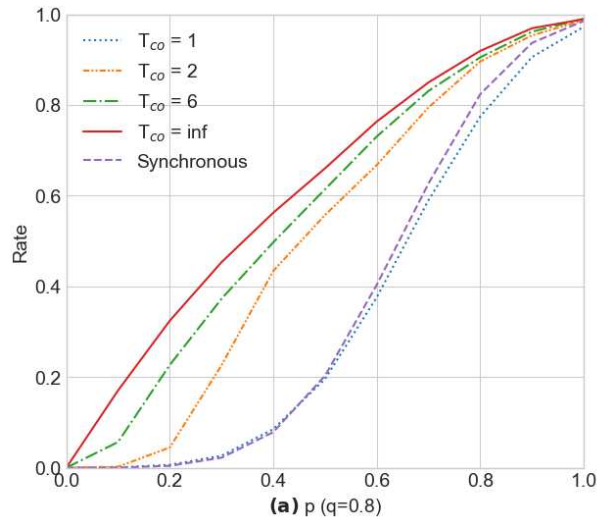
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



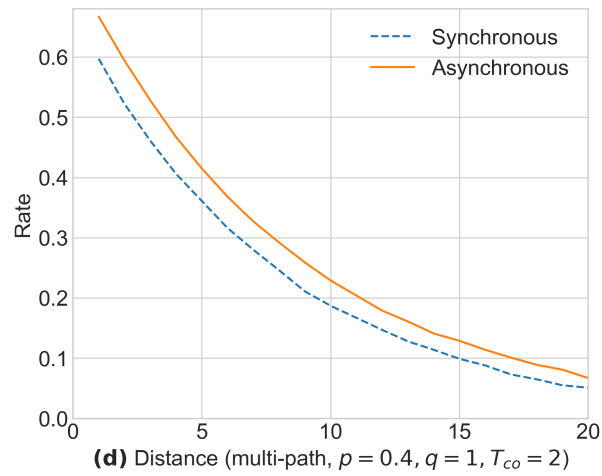
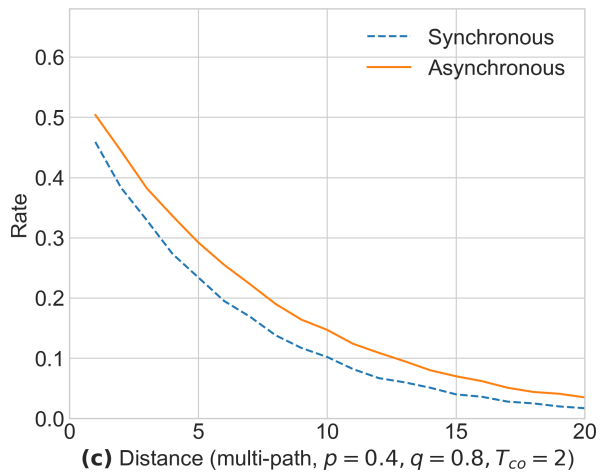
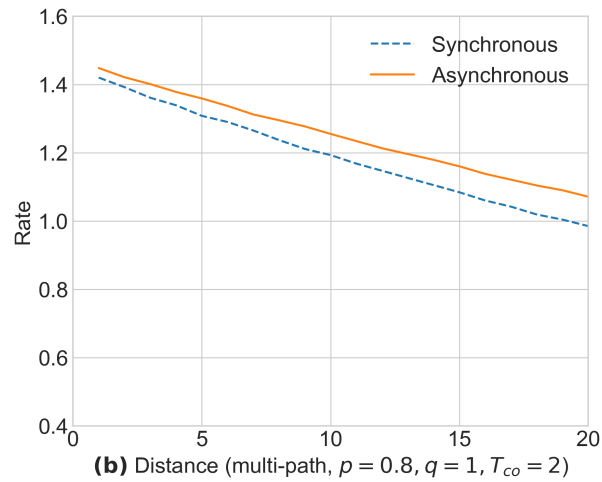
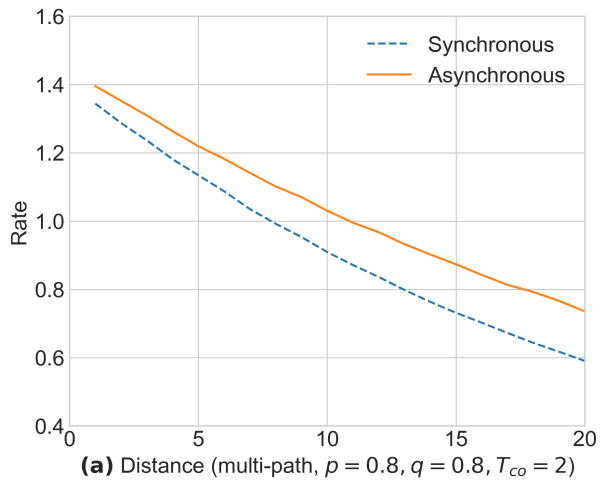
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



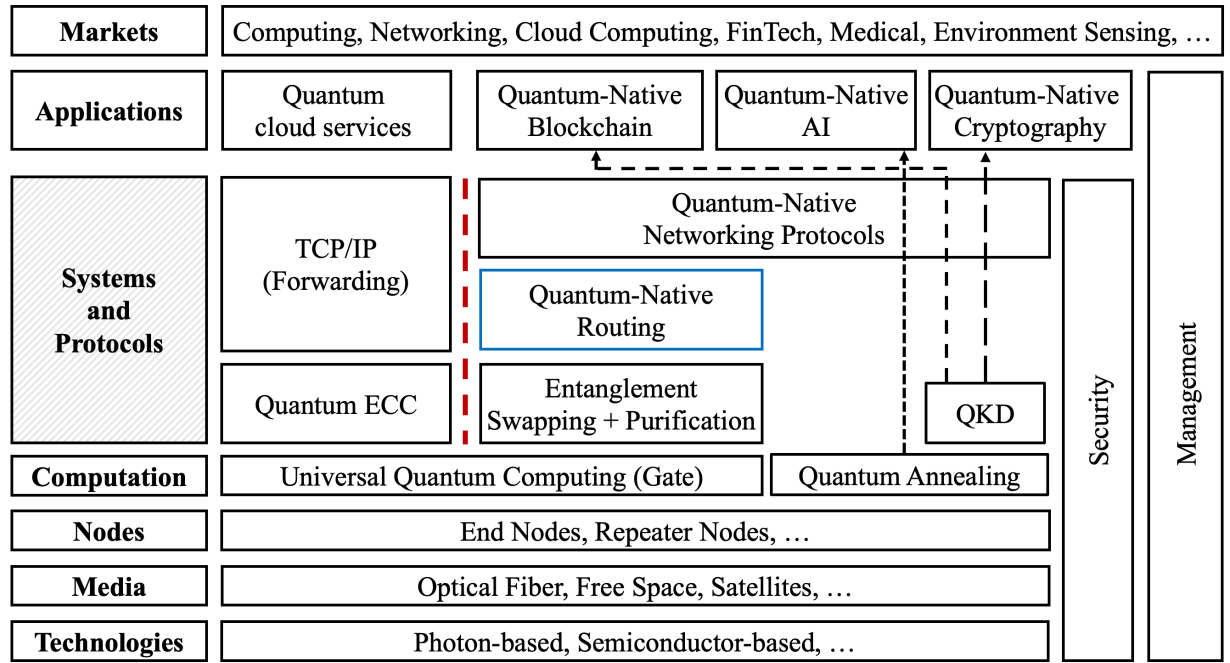
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



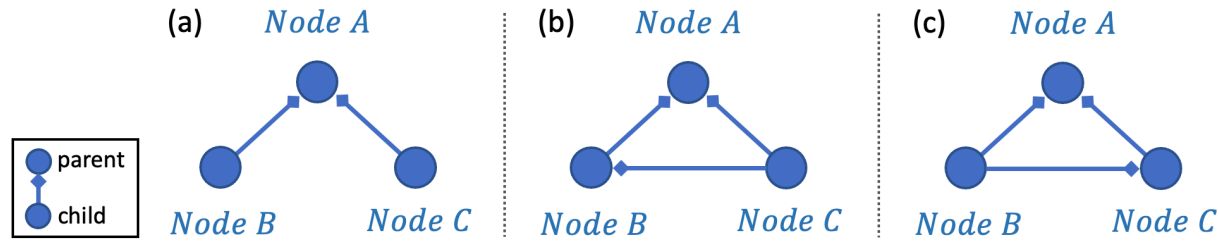
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819



This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1116/5.0172819

