Quantum Error Correction Based Entanglement Routing in Socially-Aware Quantum Networks

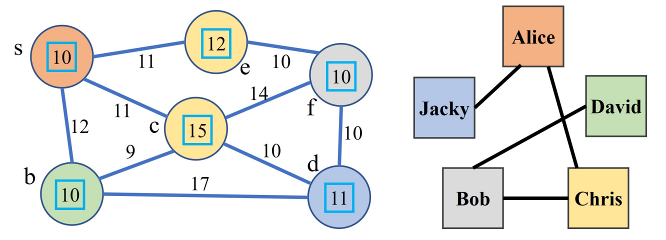
Shao-Min Huang[§†‡], Ming-Huang Chien[§†], Ting-Yuan Wen[‖†], Qian-Jing Wang[‖†], and Jian-Jhih Kuo[*†]
[†]Dept. of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan
[‡]Research Center for Information Technology Innovation, Academia Sinica, Taipei City, Taiwan

***Abstract*—Quantum teleportation enables high-security communications via quantum entanglement. However, decoherence, signal decay, and environmental interference may cause imperfect entangled pairs with low fidelity. Such entangled pairs may easily generate errors when used to teleport data qubits and cause fatal computation errors on quantum computers. Fortunately, data qubits can be encoded via quantum error correction (QEC) code to recover the detected error to some extent. Nevertheless, letting any repeaters process data qubits is dangerous because malicious repeaters may peep at, destroy, or fake the data qubits. Thus, in this paper, we propose a novel QEC-enabled routing framework MOON facilitated with the concept of social networks (SNs) for quantum networks (QNs). MOON selects only trusted repeaters to process data qubits and exhibits elastic routing with QEC to maximize the throughput without errors. Last, simulation results show that our framework can outperform existing approaches by up to 34% on network throughput without errors.**

I. Introduction

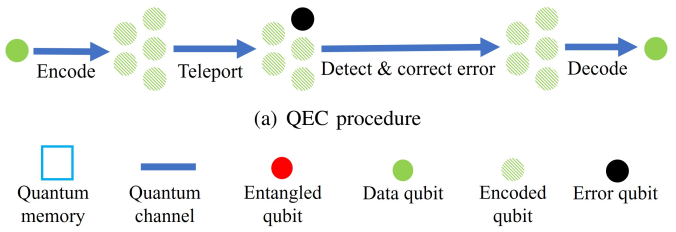
Recently, quantum networks (QNs) have been validated and realized to transmit information to facilitate the technical frontier development [1]. QNs connect quantum computers (QCs) to transmit quantum bits (i.e., *qubits*) and enable distributed quantum computing [2]. A QC is a node in a QN and has a specific number of *quantum memory* to store qubits and prevent qubits from fast decoherence [3], as shown by the blue squares with a number in Fig. 1(a). Any two adjacent nodes are interconnected by a bunch of optical fibers (i.e., *quantum channels*) with the same quality[1] for generating entangled pairs between them, as shown by the lines with a number in Fig. 1(a). By doing so, they can teleport a data qubit via an entangled pair [5] (i.e., *quantum teleportation*) and prevent a third party from eavesdropping due to the "no-cloning theorem [2]."

Further, multiple successive entangled pairs can turn into a long-distance entangled pair (i.e., entangled path) to achieve *end-to-end* teleportation by conducting *entanglement swapping* on nodes [6]–[10]. However, long-distance entangled pairs may suffer from decoherence, signal decay, or environmental interference and thus become imperfectly entangled [2], [11]. To measure how perfectly an entangled pair is entangled, *fidelity* is a classic metric to indicate an entangled pair state [11]. An entangled pair with low fidelity tends to generate an error (e.g., a bit flip or phase flip) when teleported, which causes an unwanted situation [12]. Moreover, the error rate will



(a) Quantum network (b) Social network

Fig. 1. A given quantum network with a social network of node owners.



(a) QEC procedure

Fig. 2. Qubit errors and QEC procedure

accumulate after multiple links are merged into an entangled path. That is, the fidelity will decrease when an entangled link swaps with another to obtain an entangled path [12]–[14]. Fidelity should be carefully examined before teleportation since data qubit errors may cause computation errors on QCs.

Fortunately, data qubits can be encoded with a quantum error correction (QEC) code [2], which has a larger Hilbert space and enables error correction. This paper adopts the five-qubit stabilizer code [2] to encode one qubit into five encoded qubits and recover one encoded qubit's error. Take Fig. 2(a) for example. 1) The data qubit is encoded into five encoded qubits. 2) Then, the encoded qubits are teleported through an imperfect entangled path to another node, and one qubit may be incorrect after teleportation. 3) The node detects the error by checking the error syndrome and then identifies the error if there is an error encoded qubit exactly. Such an error can be corrected according to the error syndrome. 4) The node decodes the encoded qubits back to the original data qubit.

It is practical to assume that all the nodes in the network have the ability to encode qubit, detect and correct errors, and decode qubits to overcome the low fidelity issue in the near future [15]. However, letting any repeaters process data qubits is dangerous because malicious repeaters may peek at, destroy, or fake the data qubits [16]–[18]. Thus, in this paper, we introduce the concept of social network (SN) into QNs to help select trusted repeaters to process data qubits. Fig. 1(b) shows a toy SN, where any two users interconnected with an edge trust each other. Every user may own some nodes in the

[§‖]: equal contributions; [*]: corresponding author (lajacky@cs.ccu.edu.tw)

This work was supported by the National Science and Technology Council under Grants 111-2628-E-001-002-MY3, 111-2628-E-194-001-MY3, 112-2218-E-194-005, 113-2221-E-194-040-MY3, and the Academia Sinica under Thematic Research Grant AS-TP-110-M07-3 in Taiwan.

[1]It is reasonable that the fibers in the same bunch have the similar quality at any time since they are in the similar environment [4].

QN, as shown by the circles with the same color as the user in Fig. 1(a), and trusts its nodes and its trusted users' nodes.

With the socially-aware QEC, trusted repeaters are allowed to execute the above processes to foster multiple candidate routing paths. Specifically, any request from a source to a destination can be divided into several *tasks*, each of which starts from the source or a trusted repeater and ends at a trusted repeater or the destination, with or without encoding. Fig. 1(a) shows an example of a request from source s to destination d. Some candidates are illustrated as follows. 1) The entangled path could be end-to-end teleportation generated by gluing an entangled link (s, c) and another (c, d) without encoding. This takes both node s and c two quantum memories, while node d only spends one. 2) The request could be divided into two tasks using (s, c) and (c, d), respectively, storing and forwarding the data qubit via node c. At the first time slot, node s takes two memories, while nodes c takes one. Then, node c uses two memories and node d uses one at the second time slot. 3) Further, the data qubit could be encoded at s, corrected at c, and corrected/decoded at d. Node s has to take five memories to store the encoded data qubits. It may take several time slots to teleport the five encoded qubits to node c. Node c has to wait for all of them to detect and correct the error if needed. Then, node d has to wait for all of them to correct and decode. 4) Similarly, the data qubit could be encoded at s, corrected and decoded at c, and then teleported to d. 5) It could also be teleported to c, encoded at c, and decoded at d. 6) It could be encoded at s. Two encoded qubits are teleported to e, while the others are teleported to c. Then, the five encoded qubits are teleported to d and corrected and decoded there. Overall, there are many possibilities for each request.

The above multiple candidates manifest that QEC enables the more flexible selection of routing paths to increase correct rate. However, new research challenges also appear as we aims to maximize the expected throughput with high correct rate as follows. 1) *Appropriate task length.* To teleport encoded data qubits, the shorter paths the task leverages, the better correct rate after correction. However, a request will be delayed seriously if all its tasks use short paths. 2) *Encoding or not.* Intuitively, encoding may mitigate the correct rate issue, but turning a data qubit into five encoded qubits would consume more network resources for teleportation and incur network congestion. The network condition should be carefully examined to determine whether to use encoding for each task. 3) *Similar path fidelity.* The five encoded qubits for a task can go through different paths to reach a trusted repeater or the destination. Nevertheless, such paths may have different fidelity, and the fidelity difference between the paths should be bounded to guarantee the correction rate. The simplest way to minimize the fidelity difference is to exploit only one path, but the resources on the path may be insufficient to teleport the five encoded qubits at the same time. Thereby, it may leverage five different paths, exploit five copies of resources on a path, or wait for resources to teleport the five encoded qubits. Overall, socially-aware entanglement routing with QEC is challenging because it needs to jointly decide whether, when, and where to encode, detect, correct, and decode the data qubits.

In this paper, we present a novel framework[2] termed Socially-aware entangle**m**ent r**o**uting with Quantum Err**o**r Correctio**n** (MOON) to provide more flexible routing in QNs to maximize correct throughput. To this end, MOON adopts QEC processes and temporary storage at trusted repeaters to improve data qubits' correct rate during teleportation. For the first challenge and the second and third challenges, we design two algorithms named **I**ntermediate **N**odes Selec**t**ion (INT) and **M**ulti-maneuver Rout**i**ng for **C**orrect R**a**te (MICA), respectively. INT determines a suitable set of intermediate nodes to divide each new request into tasks. MICA finds the paths to connects the intermediate nodes to achieve each task. Finally, the simulation results show our method can outperform existing approaches by up to 34% on network throughput.

II. Background and Assumptions

This section describes the preliminary background such as quantum state, entangled pair, entanglement swapping, mixed state, fidelity, and the adopted QEC technique in this paper.

A. Quantum State

The two basic states for a qubit are $|0\rangle$ and $|1\rangle$. A qubit is different from a classic bit. A qubit can be a superposition state $|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$, and the variable a_0 and a_1 are the information of this qubit. When we measure the state $|\psi\rangle$, we will get either 0 with a probability of $|a_0|^2$, or 1 with a probability of $|a_1|^2$, where $|a_0|^2 + |a_1|^2 = 1$. With a two-qubit quantum state, it can be the superposition state of $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, and $|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1$. An entangled pair, or called a bell pair, can be described as $|\beta_{xy}\rangle = \frac{|0y\rangle + (-1)^x |1\bar{y}\rangle}{\sqrt{2}}$, where $\bar{y} = 1 - y$.

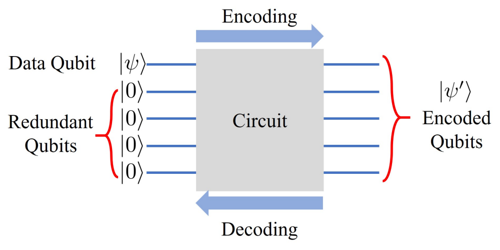
The above shows a pure state. A mixed quantum state is a statistical ensemble of pure states. A mixed state deals with a system which is not fully known. In other words, we can use a mixed state to describe an imperfect quantum state. A trivial mixed state can be described as $\rho = \sum_s p_s |\psi_s\rangle \langle\psi_s|$, and $\sum_s p_s = 1$. This describes that a mixed-state system has a probability of p_s for a pure state $|\psi_s\rangle$. Thus, an imperfect entangled pair can be written as $\rho_e = F_e |\Psi\rangle \langle\Psi| + \sum_{us} p_{us} |\Phi\rangle \langle\Phi|$, where $|\Psi\rangle$ is the pure state of the entangle pair, and it has the probability of F_e in this mixed state; $\sum_{us} p_{us} |\Phi\rangle \langle\Phi|$ is the other unwanted state which may comes from decoherence. We term F_e as the entangled pair's fidelity.

Two qubit pairs such as $(1, 2)$ and $(3, 4)$ can conduct an entangled swapping to change their state. If the swapping process is executed on qubits 1 and 4, then we will get two entangled pairs $(1, 4)$ and $(2, 3)$. Furthermore, if the entangled pairs $(1, 2)$ and $(3, 4)$ with the fidelity of F_1 and F_2, the probability of both pairs in the wanted state is $F_1 \times F_2$. Here we assume the swapping process is perfect. Thus, $F_1 \times F_2$ can be seen as the approximate fidelity of pair $(1, 4)$ [14].

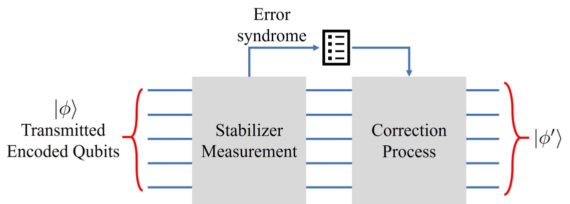
B. Five-qubit Error Correction Code

This paper adopts the five-qubit stabilizer code [2]. This code is $[[5, 1, 3]]$, which means it encodes one qubit to five qubits. It has the distance of three, and thus it can correct an arbitrary

[2]Due to the page limit, the related work is placed in Appendix A in [19].



(a) Encoding and decoding processes



(b) Error detection and correction processes

Fig. 3. Encoding, decoding, and error detection and correction processes.

error on $\lceil \frac{3-1}{2} \rceil = 1$ qubit. Fig. 3(a) demonstrates the encoding and decoding processes. A data qubit is encoded with four redundant qubits via an encoding circuit. It turns the original quantum state $|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$ to $|\psi'\rangle = a_0 |0_L\rangle + a_1 |1_L\rangle$, where $|0_L\rangle$ and $|1_L\rangle$ are five-qubit states. After encoding, the encoded state still carry the information of the initial data qubit, a_0 and a_1. The quantum circuit has the property of reversibility. We can decode the encoded state to the initial state via the same circuit with the opposite direction.

Fig. 3(b) illustrates the error detection and correction processes. A transmitted encoded qubit $|\phi\rangle$ may have errors (i.e., it is different from the encoded state $|\psi'\rangle$). We input $|\phi\rangle$ to a circuit to measure the stabilizer and get the error syndrome, check the look-up table to judge which error occurs, and correct the error by using the corresponding circuit. After the correction process, we can get a new state $|\phi'\rangle$. If there are a lot of errors in $|\phi\rangle$, the error detection and correction processes may not succeed (i.e., $|\psi'\rangle \neq |\phi'\rangle$). In contrast, if there is only one encoded qubit with an error, it can be perfectly recovered.

III. System Model & Framework Design

This section introduces the system model and then proposes a novel framework MOON for QNs.

A. System Model

A QN consists of several quantum nodes, each of which has limited quantum memory and connects to its neighboring nodes with a limited number of quantum channels [20]. Also, every node can perform encoding, decoding, and error handling processes, while storing an (encoded) data qubit on a node takes one quantum memory. This paper assumes that the above procedures are perfect. In other words, errors only occur at the teleportation of data qubits via imperfect entangled paths. Besides, the storage time limit of quantum memory can be ignored because QEC can correct data qubits' decoherence.

An entangled link between two adjacent nodes is created by a pair of entangled qubits on a quantum channel, as shown by the blue lines in Fig. 1(a). Due to decoherence from the environment, entangled links may be imperfect and has a fidelity of $F_e(u,v)$. Let $F_e(u,v)$ denote the fidelity of all the entangled links created on the link (u,v) at any time. This paper assumes that all quantum channels on the same link have the same fidelity since they have a similar environment [4].

This paper assumes the swapping process, gluing multiple entangled links into a longer entangled path, is always perfect. An entangled path is one-width and may contain several entangled links with different fidelity. Thus, the entire entangled path has the fidelity of the multiple of all the entangled links' fidelity, as defined in Definition 1. Then, the most reliable entangled path between two nodes is defined in Definition 2.

Definition 1. The fidelity of a path p, denoted as $F_p(p)$, is the product of all the fidelity of links on the path [12]–[14], i.e., $F_p(p) = \prod_{(u,v)\in p} F_e(u,v)$, where p denotes the path.

Definition 2. The *most reliable path* from node u to node v, denoted by p^*_{uv}, is the path with the highest fidelity of a path from u to v, i.e., $p^*_{uv} = \arg\max_{p\in P_{uv}} F_p(p)$, where P_{uv} denotes the set of all possible paths between u and v.

Moreover, each node belongs to a specific owner. An owner may send a data qubit via its node s to another node d in QNs. Additionally, owners trust each other *iff* they have an edge in SN, as shown in Fig. 1(b). For each owner, the repeaters from its trusted owners can assist its requests to conduct the following processes: 1) storing data (encoded) qubits, 2) encoding data qubits, 3) detecting and correcting encoded data qubits, and 4) decoding encoded data qubits. In order to coordinate all nodes and execute phases, the global information of the SN and QN is periodically collected by the central controller. It is reasonable since the one-way propagation delay takes at most tens of milliseconds [21].

B. MOON in a Nutshell

MOON aims to maximize the number of requests served correctly without error. We assume that MOON uses a central controller to manage the QN in a time-synchronous manner (i.e., time is divided into time slots). The controller periodically performs the four phases (i.e., P1$-$P4) at each time slot [6], [7] , briefly described in the following subsections. Before them, the controller precalculates five candidate paths for every request and stores the results in a table (detailed in Section IV).

1) P1 – Request Dealing Phase: When a new request arrives, the source node needs to take one quantum memory to store and admit the data qubit. Once admitted, the controller has to serve the request. If the source node has insufficient quantum memory to store the qubit, then the request will fail.

2) P2 – Intermediate Node Selection Phase: P2 determines which trusted repeaters for the new request should act as *intermediate node* to deal with the data qubit. P2 jointly considers SN and correct rate to choose the intermediate nodes. P2 has to select the set of sequential intermediate nodes for each request, and the data qubit will transiently stay at these intermediate nodes in order until it reaches the destination. Then, it divides each request into tasks by the selected intermediate nodes. Note that P2 does not decide the actual paths for tasks. Later, INT will be introduced to facilitate P2 in Section IV-A.

3) P3 – Path Decision and Resource Allocation Phase: P3 aims to choose a suitable way to deal with each task. To this end, P3 has to decide whether to encode the data qubit and allocate the path for each task. If a task is determined to transmit five encoded data qubits, then the current node will try to encode the data qubit and use five quantum memory, if needed. Otherwise, the current node will attempt to decode it and use one quantum memory, if needed. Later, MICA will be introduced to complete P3 in Section IV-B.

4) P4 – Physical Process Phase: P4 conducts all the physical processes such as generating entangled paths, teleporting data qubits, detecting and correcting errors, and storing qubits. After that, P4 initializes all the resources of the entangled paths to ease synchronization, and those unfinished tasks will be served at the following time slots.

IV. Algorithm Design — INT and MICA

Our intuitive idea is to divide each new request into tasks as reliable as possible, exploit only the paths with the highest fidelity to connect the source or a trusted repeater to the destination or a trusted repeater for each task, and must adopt encoding if the correct rate can be improved by QEC. We term it Fidelity-First. It is trivial that Fidelity-First can guarantee the highest correct rate for each request. However, Fidelity-First may seriously delay the tasks when the network is congested since it only focuses on fidelity and never changes the choices of the routing paths and QEC processes, even if the resources become insufficient. In contrast, an efficient algorithm should react elastically according to the network condition. Thus, we propose two algorithms termed INT and MICA for P2 and P3 to determine the intermediate nodes, physical paths, and QEC processes to maximize the correct throughput while consuming fewer time slots for each request. Particularly, INT decides which intermediate nodes each request's data qubit should stay at transiently based on the path fidelity of the QN and their trust relationships. MICA further introduces and exploits a subtly-designed *threshold* to filter and find paths with high fidelity and sufficient resources for teleportation to ensure a moderate waiting time. The details of INT and MICA are described in Sections IV-A and IV-B. Moreover, the time complexity of these algorithms is described in Appendix B in [19]. Later, Section V-B5 will show INT and MICA can shorten the waiting time for resources compared with Fidelity-First. Furthermore, the pseudocodes of INT and MICA are respectively provided in Algorithms 1 and 2 of Appendix C in [19].

Before the details, to reduce the redundant computation time, MOON precalculates the *top-5 reliable paths* for all pairs of nodes in the original QN G and stores them in a table T for later use. Specifically, it makes a clone of the original QN G, denoted by G', sets each link's weight in G' to $-\ln F_e(u,v)$, and runs the Dijkstra algorithm to find the top-5 reliable paths for each pair of nodes in G'. Note that the second path will be found in the residual graph after the resources of the first path is reserved, and so on. Similarly, MOON iteratively finds the next path in the residual graph until this pair has five paths. Then, it will reset all resources in G' and continue finding five paths for the next pair of nodes. The five paths and their fidelity for each node pair will be stored together in table T. For ease of presentation, we let $T(u,v)$ denote the set of top-5 reliable paths between u and v in table T.

A. ***I****ntermediate* ***N****odes Selection (INT)*

For each new request, INT decides which trusted repeaters the request's data qubit should stay at transiently (i.e., find the intermediate nodes) based on the link fidelity to address the first challenge. INT first builds a *virtual graph* G_t, which is an edge-weighted complete graph and contains only the source, destination, and trusted repeaters. Each virtual edge (u,v) in the virtual graph G_t represents the most reliable path between node u and node v physically on the original QN G. Then, INT finds the most reliable virtual path between the source and destination of the new request on the virtual graph G_t. Note that each node on such a virtual path must be trusted nodes for the request, and the request's data qubit can be stored and stay there transiently during the teleportation. Later, MICA will employ T to recover each edge of virtual paths in the virtual graph G_t to the physical path(s) in the original QN G.

Specifically, INT selects the trusted nodes according to the SN and builds a virtual graph G_t for each request because only the source's trusted nodes with a sufficient memory (i.e., more than five) can store, process, and teleport the five encoded data qubits. Recall that the virtual edge between nodes u and v in the virtual graph G_t represents the path from u and v on the QN G. Moreover, the data qubit can be teleported with or without encoding. Thus, the weight of each virtual edge (u,v) in G_t, denoted by $w(u,v)$, is set based on 1) the *correct rate* (see Definition 3) of using the most reliable path between u and v stored in table T and 2) the correct rate of using the top-5 reliable paths stored in T with encoding. To make the weight *additive*, INT takes the logarithm of weight $w(u,v)$ and multiply it by -1. Then, INT acquires the trusted nodes on the most reliable path between the source and destination of the request in the virtual graph G_t with Dijkstra algorithm. Thus, the request will be divided into tasks, each of which starts from the source or a trusted node and ends at the destination or a trusted node. Last, INT admits the request and its tasks.

Definition 3. Let $F_c(\{p_1,...,p_5\})$ denote the *correct rate* of a data qubit after teleporting its five encoded qubits via a set of five paths (i.e., p_1 to p_5) to intermediate nodes, detecting and correcting errors, and decoding them back to the origin data qubit. Due to the five-qubit stabilizer code can correct one qubit with an arbitrary error, it can be defined as

$$F_c(\{p_1,...,p_5\}) = \prod_{i=1}^{5} F_p(p_i) + \sum_{i=1}^{5} \frac{1-F_p(p_i)}{F_p(p_i)} \times \prod_{i=1}^{5} F_p(p_i)$$

B. ***M****ulti-maneuver Rout****i****ng for* ***C****orrect R****a****te (MICA)*

For each request's current task (i.e., teleport the data qubit to the intermediate node or destination), MICA determines 1) whether to encode the data qubit and 2) whether to wait for the resources to address the second and third challenges. For the former, MICA compares the data qubit's correct rates before and after using QEC and chooses the one with a higher correct rate. Meanwhile, for the latter, MICA examines whether the corresponding paths in table T have sufficient resources and

TABLE I
RUNNING TIME ON SERVER WITH INTEL XEON GOLD 6248R CPU (SEC)

β	1	5	10	15	20
MOON	0.0013	0.0023	0.0036	0.0050	0.0061
GREEDY	0.0019	0.0057	0.0085	0.1041	0.1152
Q-CAST	0.0220	0.2007	0.5319	0.9231	2.6289
REPS	0.1301	0.4871	1.3767	2.4660	2.7089

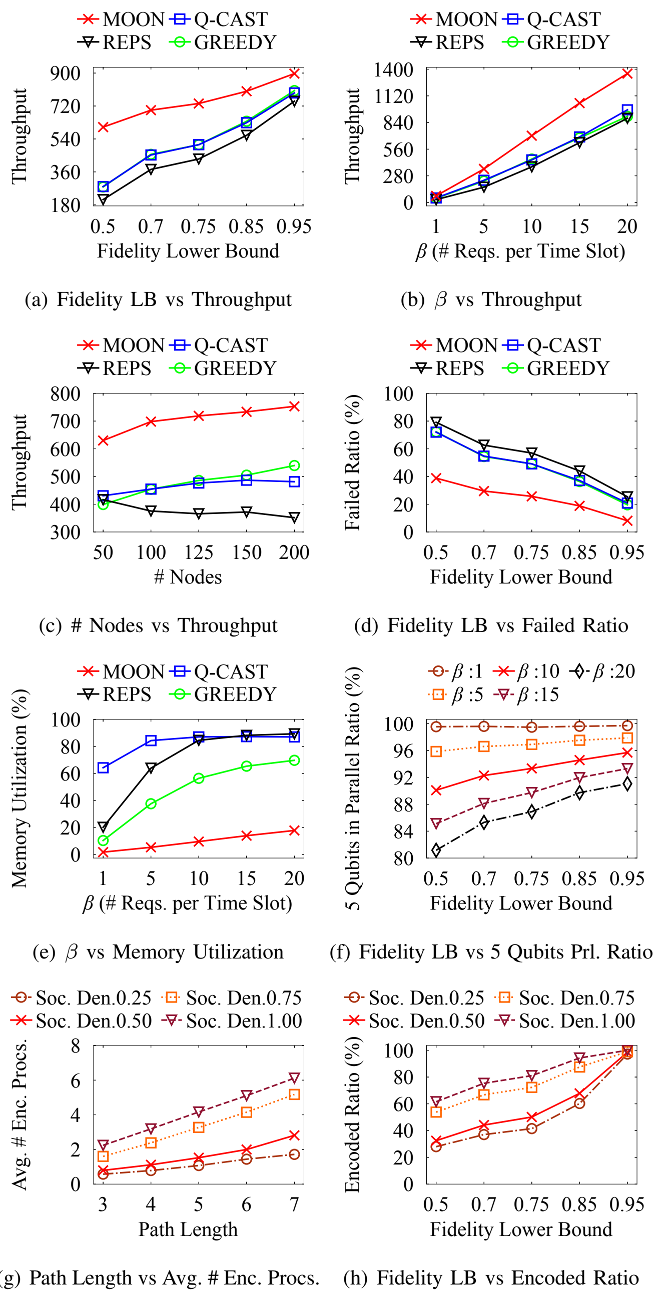
calculates a threshold to judge whether to postpone the request. If the correct rate achievable by the current paths is less than the threshold, MICA will delay the request to the next time slot. Otherwise, it will choose the way (i.e., using encoding or not) with a better correct rate for the request's task.

Specifically, MICA checks the paths on T and extracts the paths that have enough resources on the residual graph, denoted as $\mathcal{P}_s$. After finding $\mathcal{P}_s$, MICA calculates the correct rate of the encoding mode, denoted as F'. Note that $\mathcal{P}_s$ may not have enough five paths due to the limited resource. In such cases, some path(s) in $\mathcal{P}_s$ will be exploited twice or more to teleport the five encoded qubits. For example, $\mathcal{P}_s$ may have only three paths, p_1, p_2, p_3, and assume $F_p(p_1) \geq F_p(p_2) \geq F_p(p_3)$ without loss of generality. To teleport five encoded qubits with encoding, the two paths with higher fidelity (i.e., p_1 and p_2) will be used twice, while the remaining path (i.e., p_3) will be used one time. Thereby, the teleportation will take two time slots. On the other hand, for the correct rate without encoding, denoted by F^*, MICA will only use the path with the highest fidelity in $\mathcal{P}_s$ (i.e., p_1 for the above example). Afterward, MICA sets a threshold according to the paths in T to filter out the paths with low fidelity. Once both F' and F^* are less than the threshold, the request will wait for the next time slot. In contrast, MICA encodes the task's data qubit into five encoded qubits if F' has a higher correct rate and use the paths in $\mathcal{P}_s$; otherwise, it directly teleports the data qubit via the best path in $\mathcal{P}_s$ without encoding. After that, the path(s) chosen by above method for each task j will be put in path set $\mathcal{K}_j$ and to be processed in P4. Note that $\mathcal{K}_j$ of each task j will be reset to empty when the data qubit reaches the local destination of the task (i.e., the destination node or the intermediate node).

V. PERFORMANCE EVALUATION

A. Simulation Settings

We implement GREEDY [22], Q-CAST [6], and REPS [7] to compare with MOON. The QNs are generated by the Waxman model [23], which was also used in [6], [7]. The SNs are generated by MEDICI [24]. The generated quantum nodes in each QN are randomly placed in a 2000km $\times$ 4000km rectangle field. They are randomly assigned to 20 owners in the SN. In default setting, any two nodes have an edge with probability $\delta e^{-l(u,v)/\epsilon L}$ in the Waxman node, where L is the largest distance between any two nodes and $l(u,v)$ is the distance between node u and node v. We set $\delta = 0.85$ and $\epsilon = 0.1$ for the Waxman model to generate 100 quantum nodes. The density of social network links is set to 50% in MEDICI. The number of qubits of each node is assigned randomly from 28 to 32, and the number of quantum channels of each edge is assigned randomly from 3 to 7 [6]. The fidelity of quantum channels is evenly distributed between $[0.70, 0.95]$



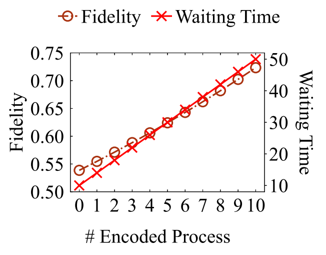
(a) Fidelity LB vs Throughput (b) β vs Throughput (c) # Nodes vs Throughput (d) Fidelity LB vs Failed Ratio (e) β vs Memory Utilization (f) Fidelity LB vs 5 Qubits Prl. Ratio (g) Path Length vs Avg. # Enc. Procs. (h) Fidelity LB vs Encoded Ratio

Fig. 4. Effect of different parameters on different metrics.

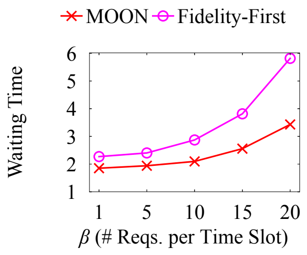
[4]. Each simulation runs 100 time slots. For each time slot, $\beta = 10$ source-destination pairs are randomly selected to generate requests. The key metric, *throughput*, presents the number of requests served correctly with no qubit error. Each result is averaged over 50 trials.

B. Numerical Results

Overall, MOON has the best performance among all frameworks. TABLE I shows that MOON runs the fastest among all the frameworks. Fig. 4 shows MOON enhances the correct rate and improves the throughput with superior resource efficiency. It is worth noting that we implement REPS with Gurobi, the fastest linear programming (LP) solver in the world, but solving the LP of REPS still takes a long time. Q-CAST is also time-



(a) # Enc. vs Corr. Rate & Wait. Time



(b) β vs Waiting Time

Fig. 5. Comparisons of Fidelity-First and MOON with INT and MICA

TABLE II
EFFECT OF # REQS. PER TIME SLOT ON THROUGHPUT WITHOUT ERRORS

β	1	5	10	15	20
MOON	71.38	350.14	700.68	1038.86	1366.3
Fidelity-First	70.98	348.3	699.5	1024.64	1296.06

consuming since it checks many combinations of nodes to find the recovery path. In contrast, GREEDY uses a shortest-path-based algorithm, while MOON employs the precalculated all-pair top-5 reliable path table to select proper paths quickly.

1) Effect of Request Scheduling and QEC: Figs. 4(a)−4(c) show the throughput with different lower bounds of fidelity (default is 0.7), number of requests per round, and number of quantum nodes. In general, when the number of requests and the fidelity increase, the throughput grows. MOON outperforms GREEDY, Q-CAST, and REPS by up to 31%, 31%, and 41%, respectively. Fig. 4(d) further shows MOON achieves the lowest ratio of unrecoverable errors to teleported qubits, implying it can address imperfect links and raise the efficiency of QNs. It is worth noting that REPS and Q-CAST may have a lower throughput as the number of nodes increases since they both exhaust resources, causing more low-fidelity long paths.

2) Effect of Resource Allocation: Fig. 4(e) shows that the utilization grows as the number of requests increases. MOON is resource-efficient since it can select appropriate intermediate nodes to process data qubits rather than find redundant paths.

3) Effect of Maneuver Diversity: Fig. 4(f) shows the ratio of sending five qubits within a time slot with different fidelity lower bound and number of requests per time slot. In most cases, a trusted node can get all five encoded qubits within a time slot in MOON, leading to a short average waiting time of a task. Besides, as the number of requests increases or the fidelity lower bound decreases, MOON moderately employs the paths with secondary fidelity to avoid a long waiting time for the paths with the highest fidelity, addressing the third challenge.

4) Effect of Social Density: Fig. 4(g) shows the relation between the path length and the number of encoding processes with different social densities. Long-distance requests tend to be divided into more tasks by MOON since longer paths have lower fidelity. By doing so, more trusted repeaters can help correct the data qubit to address the first challenge. The number of encoding processes also grows as the social density increases because MOON has more trusted repeaters for selection. Fig. 4(h) shows that the higher social density and fidelity encourage MOON to encode more, which indicates MOON can adaptively process data qubits to handle the second and third challenges.

5) Why not Fidelity-First: Fig. 5(a) shows the effect of number of encoding processes on correct rate and waiting time for a request divided into ten tasks, each of which has a path with the fidelity of 0.94. As the number of encoding processes grows, the correct rate increases. Thus, Fidelity-First uses the most encoding processes, waiting too long for the most reliable path, as shown in Fig. 5(b) and Table II, but MOON can select other paths to decrease waiting time and improve throughput.

VI. CONCLUSION

This paper proposes a novel framework MOON to select trusted repeaters via SNs as intermediate nodes to process data qubit to enhance performance. To this end, MOON employs INT and MICA to divide each request into tasks based on the selected appropriate intermediate nodes, decide whether to process the data qubit for each task, and allocate the resources. Finally, simulation results show that MOON outperforms existing approaches on correct throughput by up to 34%.

REFERENCES

[1] C. Elliott, “Building the quantum network,” *New J. Phys.*, vol. 4, p. 46, 2002.
[2] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. American Association of Physics Teachers, 2002.
[3] N. Sangouard *et al.*, “Quantum repeaters based on atomic ensembles and linear optics,” *Rev. Mod. Phys.*, vol. 83, p. 33, 2011.
[4] Y. Zhao *et al.*, “E2E fidelity aware routing and purification for throughput maximization in quantum networks,” in *IEEE INFOCOM*, 2022.
[5] Chen *et al.*, “General form of genuine multipartite entanglement quantum channels for teleportation,” *Physical Review A*, p. 032324, 2006.
[6] S. Shi and C. Qian, “Concurrent entanglement routing for quantum networks: Model and designs,” in *ACM SIGCOMM*, 2020.
[7] Y. Zhao and C. Qiao, “Redundant entanglement provisioning and selection for throughput maximization in quantum networks,” in *IEEE INFOCOM*, 2021.
[8] S.-M. Huang *et al.*, “Socially-aware concurrent entanglement routing with path decomposition in quantum networks,” in *IEEE GLOBECOM*, 2022.
[9] ——, “Socially-aware concurrent entanglement routing in satellite-assisted multi-domain quantum networks,” in *IEEE ICC*, 2023.
[10] ——, “Socially-aware opportunistic routing with path segment selection in quantum networks,” in *IEEE GLOBECOM*, 2023.
[11] J.-W. Pan *et al.*, “Entanglement purification for quantum communication,” *Nature*, vol. 410, pp. 1067–1070, 2001.
[12] R. Van Meter, *Quantum Networking*. John Wiley & Sons, Ltd, 2014.
[13] J. Li *et al.*, “Fidelity-guaranteed entanglement routing in quantum networks,” *IEEE Trans. Commun.*, 2022.
[14] W. J. Munro *et al.*, “Inside quantum repeaters,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 21, pp. 78–90, 2015.
[15] S. Muralidharan *et al.*, “Optimal architectures for long distance quantum communication,” *Sci. Rep.*, vol. 6, pp. 1–10, 2016.
[16] Y.-Y. Fei *et al.*, “Quantum man-in-the-middle attack on the calibration process of quantum key distribution,” *Sci. Rep.*, vol. 8, pp. 1–10, 2018.
[17] S. Song *et al.*, “Secure quantum network code without classical communication,” *IEEE Trans. Inf. Theory*, vol. 66, pp. 1178–1192, 2019.
[18] R. V. Meter and J. Touch, “Designing quantum repeater networks,” *IEEE Commun. Mag.*, vol. 51, pp. 64–71, 2013.
[19] S.-M. Huang *et al.*, “Quantum error correction based entanglement routing in socially-aware quantum networks (technical report),” Aug. 2024. [Online]. Available: https://github.com/510117/24-GC-QEC/blob/main/24_GC_QEC.pdf
[20] H. J. Kimble, “The quantum internet,” *Nature*, vol. 453, pp. 1023–1030, 2008.
[21] I. Parvez *et al.*, “A survey on low latency towards 5G: RAN, core network and caching solutions,” *IEEE Commun. Surv. Tutor.*, vol. 20, pp. 3098–3130, 2018.
[22] M. Pant *et al.*, “Routing entanglement in the quantum internet,” *NPJ Quantum Inf.*, vol. 5, pp. 1–9, 2019.
[23] B. Waxman, “Routing of multipoint connections,” *IEEE J. Sel. Areas Commun.*, vol. 6, pp. 1617–1622, 1988.

[24] D. F. Nettleton, S. Nettleton, and M. C. i. Farriol, "MEDICI: A simple to use synthetic social network data generator," in *MDAI*, 2021.

Appendix A
Related Work

Elliott *et al.* introduced QNs to realize secure communications [1]. Meter *et al.* proposed a large QN architecture with layered recursive repeaters, where repeaters may not trust each other, and then designed new protocol layers to support quantum sessions to ensure robustness and interoperable communication [18]. Muralidharan *et al.* presented new quantum nodes can execute the QEC processes and classified existing quantum nodes into three generations [15]. Pant *et al.* presented a greedy algorithm to create entangled paths by gluing multiple established entangled links [22]. Shi *et al.* derived a routing method Q-CAST based on the Dijkstra algorithm to find paths and recovery paths to mitigate the effect of entanglement failures. [6]. Zhao *et al.* presented an LP based algorithm REPS to maximize throughput in QNs with a central controller [7]. Zhao *et al.* considered entangled links' fidelity and exploited quantum purification to enhance link fidelity [4]. Overall, implementation works adopted trusted repeaters and focused on feasibility, while optimization works neglected trusted repeaters and aimed at throughput maximization. However, none of them considers QEC-enabled nodes and trust relationships among owners to maximize throughput while ensuring data qubits are robust for further calculation.

Appendix B
Time Complexity

Time Complexity 1. The time complexity of building table T is $O(|V|^3 \log |V| + |V|^2|E|))$ because there are $O(|V|^2)$ pairs of nodes and each pair needs time $O(|V| \log |V| + |E|)$ to conduct five times of Dijkstra algorithm. ■

Time Complexity 2. The time complexity of INT is $O(|V|^2)$ for each new request since INT only conducts one time of the Dijkstra algorithm on the virtual graph. ■

Time Complexity 3. The time complexity of MICA is $O(|V|)$ for each request because MICA only needs to check the resources of the nodes on the paths and calculates the correct rate of data qubits after teleportation. ■

Appendix C
Pseudocode

Algorithm 1 Intermediate Nodes Selection

```
Input: New request $i$ with source $s_i$ and destination $d_i$, residual QN $G = (V, E)$, SN $G_s = (V_s, E_s)$, all-pair top-5 shortest paths table $T$, remaining tasks $R$
Output: Task queue $Q$
 1: $Q \leftarrow R$;
 2: $c_k \leftarrow$ the remaining memory size of each node $k \in V$;
 3: $V(s_i) \leftarrow$ trusted repeaters of $s_i$ in $G_s$;
 4: Initialize a virtual graph $G_t = (V_t, E_t)$, where
 5: $V_t \leftarrow \{k \mid k \in V(s_i) \cup \{s_i, d_i\}, c_k > 5\}$ and
 6: $E_t \leftarrow \{(u, v) \mid u, v \in V_t, u \neq v\}$;
 7: $w(u, v) \leftarrow -\ln \max\{\max_{p \in p(u,v)} F_p(p), F_c(T(u, v))\}$;
 8: $M_i \leftarrow$ the nodes on the shortest virtual path from $s_i$ to $d_i$ in $G_t$;
 9: $Q \leftarrow Q \cup (s_i, d_i, M_i)$;
10: return $Q$;
```

Algorithm 2 Multi-maneuver Routing for Correct Rate

```
Input: Residual QN $G(V, E)$, task queue $Q$, the path set $\mathcal{K}_j$ for every task $j \in Q$ to be processed
Output: Task queue $Q$, $\mathcal{K}_j$
 1: for $j \in Q$ with $\mathcal{K}_j = \emptyset$ do
 2:   $t \leftarrow$ the node that stores the qubit(s) of $j$;
 3:   $n \leftarrow$ the next node that will store the qubit(s) of $j$;
 4:   $\mathcal{P}_s \leftarrow \{$the paths with sufficient resources $\in T(t, n)\}$;
 5:   $F' \leftarrow F_c(\mathcal{P}_s)$;
 6:   $F^* \leftarrow \max_{p \in \mathcal{P}_s} F_p(p)$;
 7:   $F_{threshold} \leftarrow \max\{F_c(T(t, n)), \max_{p \in T(t,n)} F_p(p)\}$;
 8:   if $\max\{F^*, F'\} \geq F_{threshold}$ then
 9:     if $F' \geq F^*$ then
10:       Task $j$ exploits encoding;
11:       $\mathcal{K}_j \leftarrow \mathcal{P}_s$;
12:     else if $F' < F^*$ then
13:       Task $j$ does not exploit encoding;
14:       $\mathcal{K}_j \leftarrow \{\arg\max_{p \in P_s} F_p(p)\}$;
15:     Reserve the resources for $\mathcal{K}_j$ in the corresponding time slots;
16:   else
17:     Task $j$ is not served at this time slot;
18:     $\mathcal{K}_j \leftarrow \emptyset$;
19: return $\mathcal{K}_j$ for all $j \in Q$;
```