



Received June 15, 2020; revised August 30, 2020; accepted September 14, 2020; date of publication October 1, 2020; date of current version November 16, 2020.

Digital Object Identifier 10.1109/TQE.2020.3028172

# Entanglement Distribution in a Quantum Network: A Multicommodity Flow-Based Approach

KAUSHIK CHAKRABORTY<sup>1,2</sup>  (Member, IEEE),  
DAVID ELKOUS<sup>1</sup>  (Member, IEEE), BRUNO RIJSMAN<sup>1</sup> (Member, IEEE),  
AND STEPHANIE WEHNER<sup>1,2</sup> (Member, IEEE)

<sup>1</sup> QuTech, Delft University of Technology, 2628 Delft, The Netherlands

<sup>2</sup> Kavli Institute of Nanoscience, Delft University of Technology, 2628 Delft, The Netherlands

Corresponding authors: Kaushik Chakraborty and Stephanie Wehner (k.chakraborty@tudelft.nl, s.d.c.wehner@tudelft.nl)

This work was supported by an ERC Starting Grant and the QIA-Project that has received funding from the European Union's Horizon 2020 Research and Innovation Program under Grant 820445.

**ABSTRACT** We consider the problem of optimizing the achievable EPR-pair distribution rate between multiple source-destination pairs in a quantum Internet, where the repeaters may perform a probabilistic Bell-state measurement and we may impose a minimum end-to-end fidelity as a requirement. We construct an efficient linear programming (LP) formulation that computes the maximum total achievable entanglement distribution rate, satisfying the end-to-end fidelity constraint in polynomial time (in the number of nodes in the network). Our LP formulation gives the optimal rate for a class of entanglement generation protocols where the repeaters have very short-lived quantum memories. We also propose an efficient algorithm that takes the output of the LP solver as an input and runs in polynomial time (in the number of nodes) to produce the set of paths to be used to achieve the entanglement distribution rate. Moreover, we point out a practical entanglement generation protocol that can achieve those rates.

**INDEX TERMS** Entanglement distribution, end-to-end fidelity, linear programming, multi-commodity flow, routing, quantum internet.

## I. INTRODUCTION

The quantum Internet will provide a facility for communicating qubits between quantum information processing devices [1]–[4]. It will enable us to implement interesting applications such as quantum key distribution [5], [6], clock synchronization [7], secure multiparty computation [8], and others [4], [9]. To implement a full quantum internet the network needs to be able to produce entanglement between any two end nodes connected to the network [10]–[13]. The performance of such networks will depend on the quantum channels as well as the classical control units [14]. We refer to the book [1] and the papers [15]–[18] for a detailed technical introduction of a quantum Internet.

In this article, we consider the problem of optimizing the achievable rates for distributing EPR-pairs among multiple source-destination pairs in a network of quantum repeaters while keeping a lower bound on the end-to-end fidelity as a requirement. We propose a polynomial time algorithm (in the number of nodes in the network) for solving this problem

and we show that, for a particular class of entanglement distribution protocol and a relevant noise model, our solution is tight and achieves the optimal rate. Our algorithm is inspired by *multicommodity flow* optimization, which is a very well-studied subject and has been used in many optimization problems, including classical Internet routing [19]. In the context of a classical internet, a *flow* is the total number of data packets, transmitted between a source and a destination per unit time (*rate*). In this context, a *commodity* is a *demand*, which consists of a source, destination, and potentially other requirements like the desired end-to-end packet transmission rate, quality of service, etc. In a classical network, a source and a destination can be connected via multiple communication channels as well as a sequence of repeaters and each of the communication channels has a certain *capacity*, which upper bounds the amount of flow it can transmit. In this context, a flow must satisfy another restriction, called *flow conservation*, which says that the amount of flow entering a node (*inflow*), except the source and destination node, equals

the amount of flow leaving the node<sup>1</sup> (*outflow*). With these constraints, one of the goals of a multicommodity flow optimization problem is to maximize the total amount of flows (end-to-end packet transmission rates) in a network given a set of commodities (demands). There exist *linear programming* (LP) formulations for solving this problem and if we allow the flows to be a fraction then this LP can be solved in polynomial time (in the number of nodes) [20].

In a quantum Internet, we abstract the entire network as a graph  $G = (V, E, C)$ , where  $V$  represents the set of repeaters as well as the set of end nodes, and the set of edges  $E$  abstracts the physical communication links. Corresponding to each edge, we define edge capacities  $C : E \rightarrow \mathbb{R}^+$ , which denotes the maximum elementary EPR-pair generation rate. We assume that the fidelity of all the EPR-pairs, generated between any two nodes  $u, v \in V$  such that  $(u, v) \in E$ , is the same (say  $F$ ). We refer to such EPR-pair as an *elementary pair* and the physical communication link via which we create such an elementary pair is called an *elementary link*. We give a brief overview of the elementary EPR-pair generation in Appendix D. Flow in such a network is the EPR-pair generation rate between a source-destination pair. Depending on the applications, the end nodes may need to generate EPR-pairs with a certain fidelity. Keeping the analogy with the classical internet, here we refer to such requirement as a demand (commodity) and it consists of four items, a source  $s \in V$ , a destination  $e \in V$ , end-to-end desired entanglement distribution rate  $r$ , and an end-to-end fidelity requirement  $F_{\text{end}}$ . We denote the set of all such demands (commodities) as  $D$ . In this article, we are interested in computing the maximum entanglement distribution rate (flow). Given a quantum network  $G$  and a set of demands  $D$ , we investigate how to produce a set of paths  $\mathcal{P}_i$ , and an end-to-end entanglement generation rate  $r_i$  (flow), corresponding to each demand  $(s_i, e_i, F_i)$ , such that the total entanglement generation rate  $\sum_{i=1}^{|D|} r_i$  is maximized. In the rest of this article, we refer to this maximization problem as a *rate maximization problem*. What is more, here, we also investigate what type of practical entanglement distribution protocol achieves such rates.

In the case of the quantum Internet, we can use an LP for maximising the total flow  $\sum_{i=1}^{|D|} r_i$ . However, the working principle of quantum repeaters is different, unlike classical networks, the repeaters extend the length of the shared EPR-pairs by performing *entanglement swapping* operations<sup>2</sup> [15]–[18]. However, entanglement swapping

<sup>1</sup> Assuming that the intermediary repeater nodes do not lose packets while processing them.

<sup>2</sup> Entanglement swapping is an important tool for establishing entanglement over long-distances. If two quantum repeaters,  $A$  and  $B$  are both connected to an intermediary quantum repeater  $r$ , but not directly connected themselves by a physical quantum communication channel such as fiber, then  $A$  and  $B$  can nevertheless create entanglement between themselves with the help of  $r$ . First,  $A$  and  $B$  each individually create entanglement with  $r$ . This requires one qubit of quantum storage at  $A$  and  $B$  to hold their end of the entanglement, and two qubits of quantum storage at  $r$ . Repeater  $r$  then performs an entanglement swap, destroying its own entanglement with  $A$  and  $B$ , but instead of creating entanglement between  $A$  and  $B$ . This process

operations might be probabilistic depending on the repeater technology used in the quantum Internet. This implies that the usual flow-conservation property, which we use in classical networks does not hold in the quantum networks, i.e., the sum of the inflow is not always equal to the sum of the outflow. Hence, the standard multicommodity flow-based approach cannot be applied directly. For example, if the repeaters are built using *atomic ensemble* and *linear optics* then they use a probabilistic Bell-state measurement (BSM) for the entanglement swap operation [21]–[23]. Due to the probabilistic nature of the BSM, the entanglement generation rate decays exponentially with the number of swap operations.

Another difficulty for using the standard multicommodity flow-based approach for solving our problem occurs due to the end-to-end fidelity requirement in the demand. In a quantum network, the fidelity of an EPR-pair drop with each entanglement swap operation. This implies that longer path-length results in lower end-to-end fidelity. One can enhance the end-to-end fidelity using *entanglement distillation*. However, some repeater technologies are unable to perform such quantum operations (for instance, the atomic ensemble-based quantum repeaters). Hence, for such cases, one can achieve the end-to-end fidelity requirement only by increasing the fidelity  $F$  of the elementary pairs and reducing the length of the discovered path. The first of these two options, the elementary pair fidelity, depends on the hardware parameters at fabrication. The second option is related to the path length and it is under the control of the routing algorithm that determines the path from the source to the destination. For the routing algorithms, one possible way to guarantee the end-to-end fidelity is to put an upper bound on the discovered path lengths. The standard multicommodity flow-based LP-formulations does not take into account this path-length constraint. However, there exists one class of multicommodity flow-based LP-formulation, called *length-constrained multicommodity flow* [24], which takes into account such constraints. In this article, our proposed LP-formulation is inspired by the length-constrained multicommodity flow problem, and it takes into account the path-length constraint.

Given these differences, one might use the LP-formulation corresponding to the standard multicommodity flow-based approach, which we described before, but this would lead to a very loose upper bound on the achievable entanglement generation rate [25]–[32] in this setting.

In our setting, it is not clear whether one can still have an efficient LP-formulation for the flow maximization problem in a quantum Internet. In fact, recently in [33], the authors mention that multicommodity flow optimization-based routing in a quantum Internet may, in general, be an NP-hard problem. In this article, we show that for some classes of practical entanglement generation protocols, one can still have an efficient LP-formulation, which maximizes the total

can be understood as repeater  $r$  teleporting its qubit entangled with  $A$  onto repeater  $B$  using the entanglement that it shares with  $B$ .

flow for all the commodities in polynomial time (in the number of nodes). Our proposed method gives the optimal rate if due to the short memory storage time the intermediate quantum repeaters have to perform the entanglement swapping operations just after creating EPR-pairs with its neighbors.

The organization of this article is as follows. Section II-A provides a summary of our results. In Section II-C, we give the exact LP-formulation for solving rate maximization problem. In Section III, we prove that our LP formulation solves the desired rate maximization problem. Later, in the same section, we show how one can achieve the entanglement generation rates proposed by the LP-formulation using an entanglement distribution protocol. We also analyze the complexity of our proposed algorithms in Section III. We conclude this article in Section IV.

## II. RESULTS

### A. OUR CONTRIBUTIONS IN A NUTSHELL

In this article, all of our results are directed towards solving the rate maximization problem in a quantum Internet, where given a quantum network and a set of demands, the goal is to produce a set of paths such that the total end-to-end entanglement generation rate is maximized and in addition for each of the demands the end-to-end fidelity of the EPR-pairs satisfy a minimal requirement. In this section, we summarize our contributions.

- 1) In order to solve the maximization problem, we propose an LP-formulation called *edge-based formulation* where both the number of variables and the number of constraints as well as the algorithm for solving such LPs scale polynomially with the number of nodes in the graph. However, it is nontrivial to see whether this formulation provides a valid solution to the problem or not. In this article, by showing the equivalence between the edge-based formulation and another intuitive LP-formulation, called *path-based formulation*, we show that the edge-based formulation provides a valid solution.
- 2) A disadvantage of the solution of the edge-based formulation is that it only gives the total achievable rate, not the set of paths that the underlying entanglement distribution protocol would use to distribute the EPR-pairs to achieve such rate. In this article, we provide an algorithm, called the *path extraction algorithm*, which takes the solutions of the edge-based formulation, and for each of the commodities, it extracts the set of paths to be used and the corresponding entanglement distribution rate along that path. The worst case time complexity of this algorithm is  $O(|V|^4|E||D|)$ , where  $|V|$ ,  $|E|$  denote the total number of nodes and edges in the network graph  $G$  and  $|D|$  denotes the total number of demands. What is more, we point out that there exists a practical entanglement distribution protocol along a path, called the *prepare and swap protocol*, which achieves the rates (asymptotically) proposed by the path extraction algorithm.

### B. FROM THE FIDELITY CONSTRAINT TO THE PATH-LENGTH CONSTRAINT

In a quantum network, the fidelity of the EPR-pairs drops with each entanglement swap operation. The fidelity of the output state after a successful swap operation depends on the fidelity of the two input states. If a mixed state  $\rho$  has fidelity  $F$ , corresponding to an EPR-pair (say  $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ ) then the corresponding Werner state [34] with parameter  $W$  can be written as follows:

$$\rho = W|\phi^+\rangle\langle\phi^+| + \frac{1-W}{4}\mathbb{I}_4$$

where  $\mathbb{I}_4$  is the identity matrix of dimension 4. The fidelity of this state is  $\frac{1+3W}{4}$ .

In this article, we assume that all the mixed entangled states in the network are Werner states. The main reason is that Werner states can be written as mixing with isotropic noise and, hence, form the worst-case assumption. For the Werner states, if a node performs a noise-free entanglement swap operation between two EPR-pairs with fidelities  $F$ , then the fidelity of the resulting state is  $\frac{1+3W^2}{4}$ , which is equal to  $\frac{1}{4} + \frac{3}{4}(\frac{4F-1}{3})^2$  [35]. In general, if a node performs a noise-free entanglement swap operation between two EPR-pairs with Werner parameters  $W_1, W_2$  then the fidelity of the resulting EPR-pair is  $\frac{1+3W_1W_2}{4}$ . This is a well-known result in the community. However, for the completeness in Appendix D, we give the calculation of the value of the Werner parameter after a successful entanglement swap operation.

Here, each demand  $(s_i, e_i, F_i)$  (where  $1 \leq i \leq |D| = k$ ) has  $F_i$  as the end-to-end fidelity requirement. We assume that the fidelity of each of the elementary pairs is lower bounded by a constant  $F > 0.5$ . Note that, in our model, we do not consider entanglement distillation, so in order to have a feasible solution, here we assume that the fidelity requirement of the  $i$ th demand,  $F_i$  is at most the fidelity of the elementary pair  $F$ . Corresponding to a demand  $(s_i, e_i, F_i)$ , if we start generating the EPR-pairs along a path  $p = ((s_i, u_1), (u_1, u_2), \dots, (u_{|p|-1}, e_i))$ , then the total number of required entanglement swap operations is  $|p| - 1$ , where the path length is  $|p|$ . As with each swap operation the fidelity drops exponentially, this implies the end-to-end fidelity will be  $\frac{1+3W^{|p|}}{4} = \frac{1}{4} + \frac{3}{4}(\frac{4F-1}{3})^{|p|}$ . In order to satisfy the demand,  $\frac{1}{4} + \frac{3}{4}(\frac{4F-1}{3})^{|p|}$  should be greater than  $F_i$ , i.e.,  $\frac{1}{4} + \frac{3}{4}(\frac{4F-1}{3})^{|p|} \geq F_i$ . From this relation, we get the following constraint on the length of the path:

$$|p| \leq \left\lceil \frac{\log\left(\frac{4F_i-1}{3}\right)}{\log\left(\frac{4F-1}{3}\right)} \right\rceil. \quad (1)$$

This implies, for the  $i$ th demand all the paths should have length at most  $\left\lceil \frac{\log\left(\frac{4F_i-1}{3}\right)}{\log\left(\frac{4F-1}{3}\right)} \right\rceil$ . In the rest of this article, for the

$i$ th demand we assume

$$l_i := \left\lfloor \frac{\log\left(\frac{4F_i-1}{3}\right)}{\log\left(\frac{4F-1}{3}\right)} \right\rfloor. \quad (2)$$

Using this constraint on the number of intermediate repeaters, we can rewrite the demand set  $D$  in the following way:

$$D = \{(s_1, e_1, l_1), \dots, (s_k, e_k, l_k)\}. \quad (3)$$

### C. LP-FORMULATION

In this section, we construct the LP-formulation for computing the maximum flow in a quantum network. For simplicity, we consider the network  $G = (V, E, C)$  as a directed graph and construct all the LP-formulations accordingly. Note that, one can easily extend our result to an undirected graph, just by converting each of the edges which connects two nodes  $u, v$  in the undirected graph into two directed edges  $(u, v)$  and  $(v, u)$ .

For the entanglement distribution rate, here we let the achievable rate between two end nodes  $s_i, e_i \in V$  along a repeater chain (or a path)  $p = ((s_i, u_1), (u_1, u_2), \dots, (u_{|p|-1}, e_i))$  be  $r_p$  such that

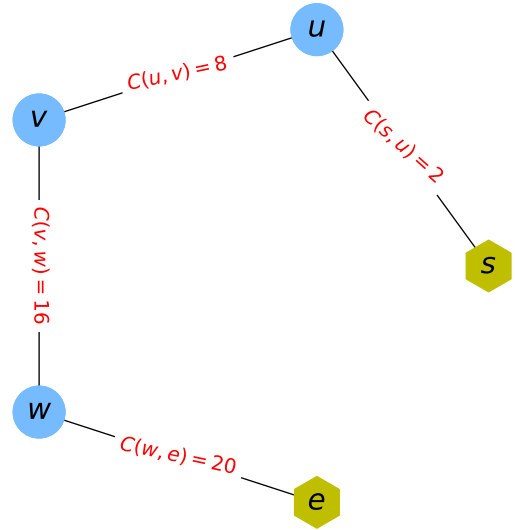
$$r_p \leq (q)^{|p|-1} \min\{C(s_i, u_1), \dots, C(u_{|p|-1}, e_i)\} \quad (4)$$

where  $C(u, v)$  denotes the capacity of the edge  $(u, v) \in E$  and  $|p|$  is the length of the path and  $q$  is the success probability of the BSM. Later, in Section III, we show that there exists a practical protocol called prepare and swap, which achieves this rate requirement along a path. For an idea of such protocol, we refer to the example of Fig. 1. In the following section, we give the LP-construction of the edge-based formulation.

#### 1) EDGE-BASED FORMULATION

In this section, we present the edge-based formulation for solving the rate maximization problem. In this formulation, we assign one variable to each of the edges of the network. As the total number of edges  $|E|$ , in a graph of  $|V|$  nodes scales quadratically with the number of nodes in the graph, the total number of variables is polynomial in  $|V|$ . This makes the edge-based formulation efficient. However, it is challenging to formulate the path-length constraint in this formulation. The main reason is that, an edge can be shared by multiple paths of different lengths, and the variables of the edge-based formulation corresponding to that edge do not give any information about the length of the paths. In this article, we borrow ideas from the length-constrained multicommodity flow [24] to handle this problem. In order to implement the length constraint we need to modify the network graph  $G$  as well as the demand set  $D$ . In the following section, we show how to modify the network graph and the demand set.

*Network Modification.* To implement the length constraint in the edge-based formulation, we define an expanded



**FIGURE 1.** Repeater chain network with three intermediate nodes and one source-destination pair  $s, e$ . Here, there is a demand to create EPR-pairs between source  $s$  and destination  $e$ . The capacity of an edge  $(u, v)$  is given by the function  $C(u, v)$ . All the repeaters use BSM for the entanglement swap. Here, we assume that the success probability of the BSM is  $q = \frac{1}{2}$ . In the prepare and swap protocol, all the intermediate repeaters perform the swap operation at the same time. This implies that expected entanglement generation rate between  $s$  and  $e$  for this protocol would be  $r_{s,e} = (q)^{4-1} \min\{C(s, u), C(u, v), C(v, w), C(w, e)\} = 2(\frac{1}{2})^3 = 0.25$ .

graph  $G' = (V', E', C')$  from  $G = (V, E, C)$  such that it contains  $l_{\max} + 1$  copies of each of the nodes, where  $l_{\max} = \max\{l_1, \dots, l_k\}$  and for all  $1 \leq i \leq k$ ,  $l_i$  denotes the length constraint of the  $i$ th demand  $(s_i, e_i, l_i)$ . For a node  $u \in V$ , we denote the copies of  $u$  as  $u^0, u^1, \dots, u^{l_{\max}}$ . We denote  $V^j$  as the collection of the  $j$ th copy of all the nodes. This implies,  $V' = \bigcup_{j=0}^{l_{\max}} V^j$ . For each edge  $(u, v) \in E$  and for each  $0 \leq j < l_{\max}$ , we connect  $u^j \in V'$  and  $v^{j+1} \in V'$  with an edge, i.e.,  $(u^j, v^{j+1}) \in E'$ . For each edge  $(u^j, v^{j+1}) \in E'$ , we define  $C'(u^j, v^{j+1}) := C(u, v)$ . In Fig. 2, we give an example of this extension procedure corresponding to a network graph of Fig. 3.

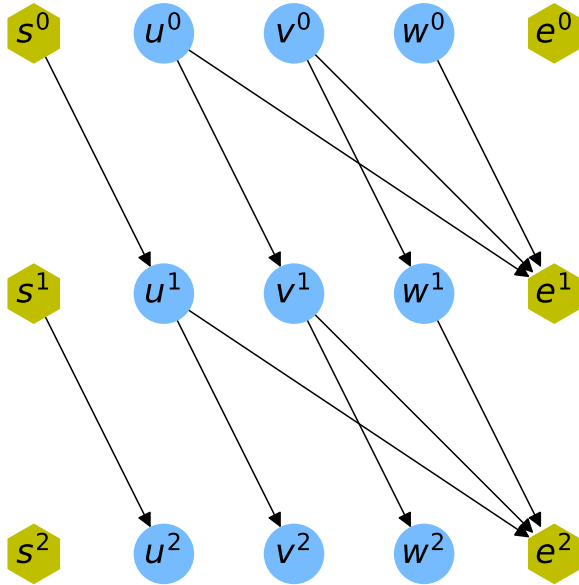
According to this construction, the length of all the paths in  $G'$  from  $s_i^0$  to  $e_i^j$  is exactly  $j$ , and all the paths from  $s_i^0$  to  $e_i^1, \dots, e_i^{l_i}$  have a path length at most  $l_i$ . This implies, for the edge formulation, the  $i$ th demand  $(s_i, e_i, l_i)$  can be decomposed into  $l_i$  demands  $\{(s_i^0, e_i^1), \dots, (s_i^0, e_i^{l_i})\}$ . This implies the total modified demand set would become

$$D_{\text{mod}} := \{D_1, \dots, D_k\} \quad (5)$$

where for each  $1 \leq i \leq k$ ,  $D_i = \{(s_i^0, e_i^1), \dots, (s_i^0, e_i^{l_i})\}$ . Note that, the new demand set  $D_{\text{mod}}$  does not have any length or fidelity constraint.

*Edge-Based Formulation.* Here, we give the exact LP-construction of the edge-based formulation. Note that, one can use a standard LP-solver (in this article, we use Python 3.7 pulp class [36]) to solve this LP (see Fig. 4 for an example).



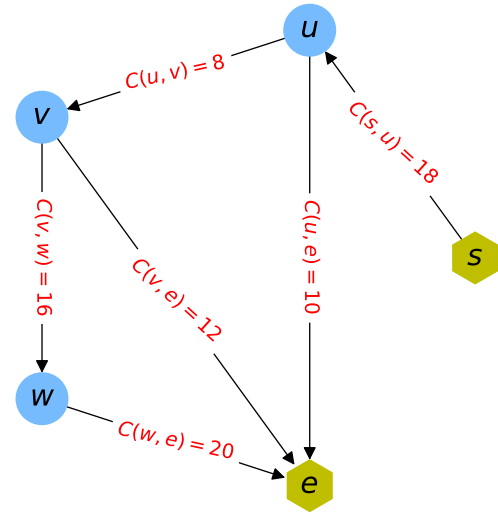


**FIGURE 2.** Extended network graph  $G' = (V', E', C')$  of the original graph in Fig. 3. Here, we are interested in finding the paths between  $s$  and  $e$  with path length at most 2. For the construction of  $G'$ , we create three copies  $u^0, u^1, u^2$  of each of the nodes  $u \in G$ . There are five nodes  $s, u, v, w, e$  in the graph of Fig. 3. This implies, in this modified graph, we have  $(s^0, s^1, s^2), (u^0, u^1, u^2), (v^0, v^1, v^2), (w^0, w^1, w^2), (e^0, e^1, e^2)$ , 15 nodes. In this original graph of Fig. 3, if  $u$  is connected to  $v$ , then in this modified graph, we connect,  $u^0$  with  $v^1$  and  $u^1$  with  $v^2$ . Note that, all the paths from  $s^0$  to  $e^1$  or  $e^2$  has hop length at most 2. In this modified graph, the new demand set corresponding to the demand  $D = \{(s, e)\}$  in the original graph  $G$  in Fig. 3 is  $D_{\text{mod}} = \{(s^0, e^1), (s^0, e^2)\}$ .

**TABLE 1.** Edge-Based Formulation

<p>Maximize</p> $\sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1). \quad (6)$ <p>Subject to:</p> <p>For all <math>1 \leq i \leq k, 1 \leq j \leq l_i,</math>  <math>0 \leq t \leq l_{\max} - 1, (u, v) \in E,</math>  <math>g_{ij}(u^t, v^{t+1}) \geq 0,</math> and <span style="float: right;">(7)</span></p> $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} g_{ij}(u^t, v^{t+1}) \leq C(u, v). \quad (8)$ <p>For all <math>1 \leq i \leq k, 1 \leq j \leq l_i,</math>  <math>u', v', w' \in V' : v' \neq s_i^0, v' \neq e_i^j,</math>  <math display="block">\sum_{u': (u', v') \in E'} g_{ij}(u', v') = \sum_{w': (v', w') \in E'} g_{ij}(v', w'). \quad (9)</math> </p>
---

In this LP-construction, for the  $i, j$ th demand  $(s_i^0, e_i^j) \in D_i$  (where  $D_i \in D_{\text{mod}}$ ), we define one function  $g_{ij} : E' \rightarrow \mathbb{R}^+$ . The value of this function  $g_{ij}(u, v)$ , corresponding to an edge  $(u, v) \in E'$  denotes the flow across that edge for the  $i, j$ th demand. We give the edge-based formulation in Table 1.



**FIGURE 3.** Network graph  $G = (V, E, C)$ , with demand  $D = \{(s, e, 2)\}$  and  $q = \frac{1}{2}$ , i.e., here the source,  $s$  wants to share EPR-pairs with  $e$ . In this network, for each edge  $(u, v) \in E$ , the quantity,  $C(u, v)$  denotes the EPR-pair generation rate corresponding to that edge.

In the objective function (6) of the edge-based formulation, the sum  $\sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1)$  denotes the entanglement distribution rate between  $s_i^0, e_i^j$ , for a fixed  $i, j$ , when  $q = 1$ . Note that, according to the construction of the graph  $G'$ , all the paths between  $s_i^0, e_i^j$  have path-length exactly  $j$ . This implies that  $(q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1)$  denotes the entanglement distribution rate between  $s_i^0, e_i^j$  and  $\sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1)$  denotes the total entanglement distribution rate for all the demands. The condition in (8) represents the capacity constraint and condition (9) denotes the flow conservation property.

Although this construction is efficient, it does not give any intuition about whether it will solve the rate maximization problem. In Section III, we give another intuitive LP-formulation, called *path-based formulation* and we explain the equivalence between the path-based and the edge-based formulation. This proof guarantees that the solution of the edge-based formulation gives a solution for the rate maximization problem.

## 2) PATH-EXTRACTION ALGORITHM

The edge-based formulation, proposed in the last section is a compact LP construction and it can be solved in polynomial time. However, this solution only gives the total achievable rates for all the commodities, it does not give us any information about the set of paths corresponding to each commodity along which one should distribute the EPR-pairs to maximize the entanglement distribution rate. In this section, we give an efficient method for doing so in Algorithm 1, which takes the solution of the edge-based formulation and produces a set of paths as well as the achievable rates across each path for each



**FIGURE 4.** This figure is the graphical representation of the SURFnet network of Fig. 5. We run our rate maximization algorithm on this graph. Here, we consider the demand set as  $D = \{(40, 45, 7), (21, 13, 6), (30, 50, 7), (38, 15, 8)\}$ . The end-nodes are represented using hexagonal boxes. The capacity of each of the link is chosen uniformly at random between  $[1,400]$ -EPR-pairs per second. We assume that the success probability of the bell-state measurement is  $q = 0.5$  and the fidelity of the each of the elementary pairs is  $F = 0.9925$  and the value of the corresponding Werner parameter  $W = 0.99$ . Given such a network and demand set, we run our edge-based formulation proposed in Table 1 and get a total achievable rate of 18.140625 EPR-pairs per second. Next, we feed the solution to the path extraction algorithm, proposed in algorithm 1 and extract the paths for each of the demands. In this figure, all the thick edges, participate in the paths. For the demand,  $(45, 40, 7)$  we get two paths,  $((45, 43), (43, 47), (47, 48), (48, 9), (9, 33), (33, 40))$  and  $((45, 44), (44, 47), (47, 48), (48, 9), (9, 39), (39, 40))$ . Note that each of the paths has path-length 6. Hence the end-to-end fidelity for this demand is  $\frac{1+3(0.99)^6}{4} = 0.955$ . Similarly, for the demand  $(21, 13, 6)$  the extracted path is  $((21, 27), (27, 28), (28, 25), (25, 20), (20, 31), (31, 13))$  and end-to-end fidelity is  $\frac{1+3(0.99)^6}{4} = 0.955$ . For the demand  $(30, 50, 7)$ , there is no path of length smaller than 8. Hence, this demand cannot be satisfied in this model. For the demand  $(38, 15, 8)$ , we extract four paths,  $((38, 23), (23, 31), (31, 14), (14, 15))$ , and  $((38, 39), (39, 31), (31, 14), (14, 15))$ , and  $((38, 39), (39, 9), (9, 48), (48, 46), (46, 15))$ ,  $((38, 39), (39, 9), (9, 6), (6, 48), (48, 47), (47, 43), (43, 15))$  and the end-to-end fidelity for the paths are 0.97, 0.97, 0.9625 and 0.9475. If we use the prepare and swap protocol for generating EPR-pairs, along each of the paths, then we can achieve the total entanglement distribution rate of 18.140625 EPR-pairs per second.

of the demands. Later, in Section III, we show that the set of extracted paths satisfies the path-length constraint for each of the demands and if one uses the prepare and swap method for distributing entanglement across each of the paths then one can achieve the entanglement distribution rate suggested from this algorithm.

### 3) EXAMPLE

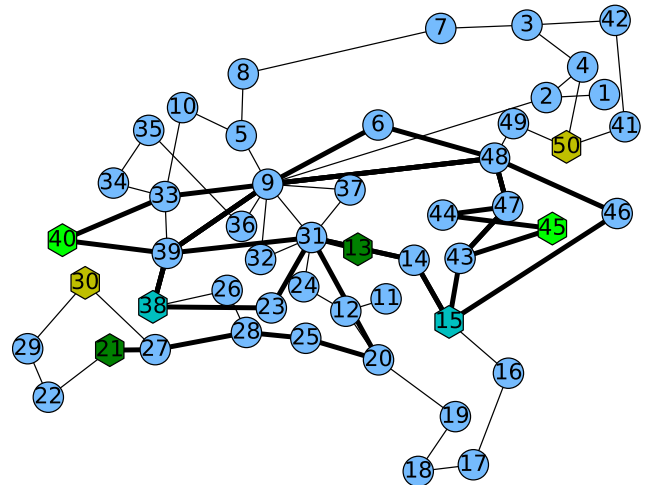
In this section, we give an example of our algorithms on a real-world network topology  $G = (V, E, C)$ . In order to do so, we choose a SURFnet topology from the *Internet*

### Algorithm 1: Path Extraction and Rate Allocation Algorithm.

**Input:** The solution of the edge-based formulation, i.e.,  $\{g_{ij}(u, v')\}_{(u, v') \in E'}\}_{1 \leq i \leq k, 1 \leq j \leq l_i}$ .

**Output:** Set of paths as well as the rate across each of the paths  $\{\mathcal{P}_{i,j}\}_{1 \leq i \leq k, 1 \leq j \leq l_i}$ .

- 1: **for**  $(i = 1; i \leq k; i++)$  **do**
- 2:     **for**  $(j = 1; j \leq l_i; j++)$  **do**
- 3:          $m = 0$ .
- 4:          $F_{i,j} = g_{ij}$ .
- 5:          $\mathcal{P}_{i,j} = \emptyset$ .
- 6:         **while**  $\sum_{v:(s_i^0, v^1) \in E'} F_{i,j,m}(s_i^0, v^1) > 0$  **do**
- 7:             Find a path  $p_{j,m}$  from  $s_i^0$  to  $e_i^j$  such that,
- 8:              $\forall (u, v') \in p_{j,m}, F_{i,j,m}(u, v') > 0$
- 9:              $\tilde{r}_{p_{j,m}} = (q)^{j-1} \{\min_{(u, v') \in p_{j,m}} \{F_{i,j,m}(u, v')\}\}$
- 10:              $\forall (u, v') \in p_{j,m},$
- 11:              $F_{i,j,m+1}(u, v') = F_{i,j,m}(u, v') - \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}$ .
- 12:              $\mathcal{P}_{i,j} = \mathcal{P}_{i,j} \cup (p_{j,m}, \tilde{r}_{p_{j,m}})$ .
- 13:              $m = m + 1$ .
- 14:         **end while**
- 15:     **end for**
- 16: **end for**



**FIGURE 5.** Pictorial view of the Dutch SURFnet network, taken from Internet topology zoo [37]. In this article, we suppose that this is a quantum network and all the nodes in this network are quantum repeater nodes and some of them are the end nodes. We also assume that the repeaters can generate EPR-pairs using the communication links, shown in this figure. We run our proposed edge-based formulation and path-extraction algorithm on this network topology for maximizing the total end-to-end EPR-pairs generation rate. We refer to Fig. 4 for detailed description.

*topology zoo* [37]. This is a publicly available example of a dutch classical telecommunication network, with 50 nodes (see Fig. 5). In this network, we assume that each of the nodes in the network is an atomic ensemble and linear optics-based quantum repeater. These types of repeaters can generate elementary pairs almost deterministically [22], [23], due to their multiplexing abilities. Here, we assume that the elementary

pair generation is a deterministic process. The elementary pair generation rate depends only on the entanglement source and its efficiency. Here, we choose the elementary pair generation rate uniformly randomly from 1 to 400 EPR-pairs per second. The success probability for the BSM,  $q$  is considered to be 0.5 for all the nodes. We also assume that the memory efficiency is one and all the memories are on-demand memories, i.e., they can retrieve the stored EPR-pairs whenever required [22].

We additionally assume that the minimum storage time of all the memories is the maximum round trip communication time between any two nodes in the network that are directly connected by an optical fiber. In the SURFnet network, the maximum length of the optical fiber connecting any two nodes is 50 km. Hence, the minimum storage time is  $\frac{2 \times 50000}{c} = 500 \mu\text{s}$ , where  $c$  is the speed of light in a telecommunication fiber, which is approximately  $c \approx 2 \times 10^8$  m/s. In this example, we consider the fidelity of all elementary pairs to be  $F = 0.9925$ . We generate the demands uniformly at random, i.e., we choose the sources and the destinations uniformly at random between 1 and 50. We also choose the end-to-end fidelity randomly from 0.93 to 0.99 for each of the demands. Substituting these fidelity constraints in (2), we obtain a maximum path length  $l_{\max} = 8$ . Here, we have generated only four demands and we assume that all the entanglement distribution tasks are performed in parallel. We optimize the total achievable rate using the LP solver available in the Python 3.7 pulp class [36]. The rates and the paths corresponding to the four demands are shown in Fig. 4. An overview of the entire procedure is given in Algorithm 2 and the code of this implementation is publicly available in [38] and the dataset can be found in [39].

### III. METHODS

In this section, we provide more details of the results, presented in the last section. First, we propose an intuitive LP-construction, called path-based formulation for solving the rate maximization problem. Later, we give an idea about how to prove the equivalence between both of the proposed LP-constructions. Next, we explain the prepare and swap protocol in detail and show that using this protocol one can achieve the optimized entanglement generation rates along the paths from Algorithm 1. We finish this section with the complexity analysis of the edge-based formulation and Algorithm 1.

#### A. PATH-BASED FORMULATION

In this formulation, for each path  $p$  corresponding to each source-destination pair  $s_i, e_i$ , we define one variable  $r_p$ . This  $r_p$  denotes the achievable rate between  $s_i, e_i$  along the path  $p$ . From (4), we have, for all  $(u, v) \in p$

$$r_p \leq (q)^{|p|-1} C(u, v). \quad (10)$$

Note that, for the  $i$ th demand the total rate  $r_i$  can be achieved via multiple paths. Let  $\mathcal{P}_i$  be the set of all possible

---

#### Algorithm 2: Method to Solve the Rate Maximisation Problem.

---

**Input:** Set of demands  $D$ , Network Graph  $G = (V, E, C)$ .

**Output:** Set of paths  $\mathcal{P}_i$  for the  $i$ -th demand and rate  $r_p$ , across each of the path  $p \in \mathcal{P}_i$ .

- 1: Convert the fidelity requirement  $F_i$  of the  $i$ -th demand  $(s_i, e_i, F_i) \in D$  into a path-length constraint  $l_i$  (use equation 2).
  - 2: Compute the modified demand set  $D_{\text{mod}}$  from  $D$  according to the path-length constraint which we compute at the previous step (see subsection II-C1 for details).
  - 3: Compute the extended network  $G'$  from  $G$  using the procedure, described in subsection II-C1.
  - 4: Implement the edge-based LP-formulation, proposed in Table 1 and compute the total maximum achievable rate  $\sum_{i=1}^k r_i$  using the LP solver available in the Python 3.7 pulp class [36].
  - 5: For the  $i$ -th demand, extract the set of paths  $\mathcal{P}_i$  and compute the required rate  $r_p$  across each of the paths  $p \in \mathcal{P}_i$ , such that  $r_i = \sum_{p \in \mathcal{P}_i} r_p$  using the algorithm 1.
- 

paths, which connect  $s_i$  and  $e_i$ . Hence

$$r_i = \sum_{p \in \mathcal{P}_i} r_p. \quad (11)$$

From (3), we have that for all the paths  $p \in \mathcal{P}_i$ ,  $|p| \leq l_i$ . We give the exact LP-formulation, which takes into account all these constraints in Table 2.

Note that, in this LP-formulation, as we introduce one variable corresponding to each path so the total number of variables is of the order  $O(|V|!)$ . This scaling stops us from using the path-based formulation for solving the maximum flow problem. However, this formulation helps us to prove that the edge-based formulation gives a solution to the rate maximization problem. In the following section, we give an idea of the proof. The full details of the proof are given in the supplementary material.

#### B. EQUIVALENCE BETWEEN THE TWO FORMULATIONS

The idea of the proof of equivalence is that, first we try to construct a solution of the edge-based formulation from the solution of the path-based formulation and then we try to construct a solution of the path-based formulation from the edge-based formulation. If both of the constructions are successful, then we can conclude that both of the formulations are equivalent.

In Table 2, we provide the path-based formulation on the basis of the network graph  $G = (V, E, C)$  and the demand set  $D$ , whereas in Table 1, we propose the edge-based formulation using the network graph  $G' = (V, E', C')$  and the demand set  $D_{\text{mod}}$ . In order to show the equivalence between

**TABLE 2. Path-Based Formulation.**

$$\begin{aligned} & \text{Maximize} \\ & \sum_{i=1}^k \sum_{p \in \mathcal{P}_i} r_p \quad (12) \\ & \text{Subject to :} \\ & \sum_{i=1}^k \sum_{\substack{p \in \mathcal{P}_i: \\ (u,v) \in p}} \frac{r_p}{(q)^{|p|-1}} \leq C(u,v), \forall (u,v) \in E, \quad (13) \\ & r_p \geq 0, \forall i \in \{1, \dots, k\}, \forall p \in \mathcal{P}_i, \\ & |p| \leq l_i \forall i \in \{1, \dots, k\}, \forall p \in \mathcal{P}_i. \end{aligned}$$

both of the formulations, we first need to rewrite the path-based formulation using the network graph  $G' = (V', E', C')$  and the demand set  $D_{\text{mod}}$ . The following section focuses on this. After that, we focus on proving the equivalence.

### 1) PATH-BASED FORMULATION ON THE MODIFIED NETWORK

In this section, we construct the path-based formulation on the basis of the new demand set  $D_{\text{mod}}$ , defined in (5) and the modified network  $G' = (V', E', C')$ . In this demand set, we use the term  $i, j$ th demand to denote the  $j$ th source destination pair  $(s_i^0, e_i^j)$  of the  $i$ th demand  $(s_i, e_i) \in D$ . We denote the set of all possible paths for the  $i, j$ th demand as  $\mathcal{P}_{i,j}$  and we assign a variable  $r_p$ , corresponding to each path  $p \in \mathcal{P}_{i,j}$ . From (4), we have, for all the edges  $(u', v') \in E'$  in a path  $p$

$$r_p \leq (q)^{|p|-1} C'(u', v'). \quad (14)$$

Note that, from the construction of  $G'$  and the new demand set  $D_{\text{mod}}$ , the length of all the paths in  $\mathcal{P}_{i,j}$  is  $j$ . This implies, if  $\mathcal{P}_i$  denotes the set of all possible paths for the  $i$ th demand  $(s_i, e_i, l_i) \in D$ , then  $\mathcal{P}_i = \bigcup_{j=1}^{l_i} \mathcal{P}_{i,j}$ . For a fixed source-destination pair  $(s_i, e_i)$ , if along a path  $p$ , the achievable rate is  $r_p$  then

$$r_i = \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} r_p. \quad (15)$$

We give the exact formulation in Table 3.

### 2) PATH-BASED FORMULATION TO EDGE-BASED FORMULATION

In this section, we construct a solution of the edge-based formulation from the solution of the path-based formulation. In the edge-based formulation, we construct a new demand set  $D_{\text{mod}}$ , where the  $i$ th demand  $(s_i, e_i, l_i)$  in the original demand set  $D$  is decomposed into  $l_i$ -demands  $(s_i^0, e_i^1), \dots, (s_i^0, e_i^{l_i})$ . Recall that, the quantity  $l_i$  denotes the upper bound on the discovered path-length, which reflects the lower bound on

**TABLE 3. Path-Based Formulation on the Modified Network.**

$$\begin{aligned} & \text{Maximize} \\ & \sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} r_p \quad (16) \\ & \text{Subject to :} \\ & \sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\text{max}}-1} \sum_{\substack{p \in \mathcal{P}_{i,j}: \\ (u^t, v^{t+1}) \in p}} \frac{r_p}{(q)^{|p|-1}} \leq C(u,v), \forall (u,v) \in E, \quad (17) \\ & \forall i \in \{1, \dots, k\}, \forall j \in \{1, \dots, l_i\}, \forall p \in \mathcal{P}_{i,j}, \\ & r_p \geq 0. \quad (18) \end{aligned}$$

the required end-to-end fidelity of the EPR-pairs generated between  $s_i$  and  $e_i$  (see Section II-B for the details). Here, each of the  $(s_i^0, e_i^j)$  are the nodes in the modified graph  $G'$ . From the construction of  $G'$ , it is clear that all the paths from  $s_i^0$  to  $e_i^j$  have length  $j$ . If we have the solutions of the path formulation, proposed in Table 3, then from there, for each edge  $(u', v') \in E'$  and for the  $i, j$ th demand  $(s_i^0, e_i^j)$ , we define the value of  $\tilde{g}_{ij}(u, v)$  as

$$\tilde{g}_{ij}(u', v') := \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u', v') \in p}} \frac{r_p}{(q)^{j-1}}. \quad (19)$$

One can easily check that the definition of  $\tilde{g}_{ij}$ , defined in (19) satisfies all the constraints of the edge-based formulation, proposed in (7)–(9). Moreover, with this definition of the  $\tilde{g}_{ij}$ , the objective function (6) of the edge-based formulation becomes same as the objective function of the path-based formulation. This shows that the optimal value of the edge-based formulation is at least as good as the solution of the path-based formulation.

### 3) EDGE-BASED FORMULATION TO PATH-BASED FORMULATION

Here, we construct a solution of the path-based formulation from the solution of the edge-based formulation. We use the algorithm, proposed in Algorithm 1 for extracting the paths and corresponding rate along that path. In the algorithm, we compute the rate  $\tilde{r}_{p_{j,m}}$  corresponding to a path  $p_{j,m}$  for a demand  $(s_i^0, e_i^j) \in D_{\text{mod}}$  as follows:

$$\tilde{r}_{p_{j,m}} := (q)^{j-1} \min_{(u', v') \in p_{j,m}} \{F_{i,j,m}(u', v')\} \quad (20)$$

where the function  $F_{i,j,m}(u', v')$  is related to  $g_{i,j}(u', v')$  and  $\tilde{r}_{p_{j,m}}$ . Here,  $F_{i,j,0} = g_{i,j}$  and for all  $m \geq 0$  we compute



$F_{i,j,m+1}$  as follows:

$$\forall (u, v) \in p_{j,m} \quad F_{i,j,m+1}(u, v) = F_{i,j,m}(u, v) - \frac{\tilde{r}_{p,j,m}}{(q)^{j-1}}. \quad (21)$$

We give a detailed proof of the fact that the paths as well as the allocated rate corresponding to each path, extracted from Algorithm 1 correspond to a feasible solution of the path-based formulation in Appendix D. Moreover, if we consider (20) as the definition  $\tilde{r}_{p,j,m}$ , then the objective function of the edge-based formulation is the same as the objective function of the edge-based formulation. This shows that this is a valid solution to the path-based formulation. In the last section, we showed that the solution of the path-based formulation is at least as good as the solution of the edge-based formulation. Hence, the solutions of both the formulations are equivalent.

### C. PREPARE AND SWAP PROTOCOL AND THE LP-FORMULATIONS

Here, we explain the prepare and swap protocol and show that with this protocol one can achieve the entanglement distribution rate along a path, proposed by Algorithm 1. We refer to the Appendices D and D for a brief idea about the elementary EPR-pair generation and the entanglement swapping operation. In the following section, we explain the protocol for a repeater chain with a single demand. After that, we extend the protocol for the case for multiple demands.

#### 1) PREPARE AND SWAP PROTOCOL FOR A REPEATER CHAIN

Suppose in a repeater chain  $u_0 = s, u_1, \dots, u_n, u_{n+1} = e$ , where for all  $0 \leq i \leq n$ , the nodes  $u_i, u_{i+1}$  are neighbors of each other and  $s$  would like to share EPR-pairs with  $e$ . In this protocol, first, all the repeaters generate entanglement with its neighbors/neighbor in parallel and store the entangled links in the memory. Here, we assume that entanglement generation across an elementary link is a deterministic event, i.e., the entanglement generation probability per each attempt is one. An intermediate node  $u_i$ , which resides between  $u_{i-1}$  and  $u_i$ , performs the swap operation when both of the EPR-pairs between  $u_{i-1}, u_i$  and  $u_i, u_{i+1}$  are ready. As we assume that each of the swap operations is probabilistic, so the entanglement generation rate with this protocol is lower. However, due to the independent swap operations, the protocol does not need a long storage time. This makes the protocol more practical.

We give an example of such an entanglement generation protocol on a repeater chain with three intermediate nodes and one source-destination pair in Fig. 1. In the following lemma, we derive an analytical expression of the end-to-end entanglement generation rate in a repeater chain network for the prepare and swap protocol. Note that, a variant of the proof of Lemma 1 can be found in the literature [23]. For completeness, in the appendix, we include the proof of this lemma.

*Lemma 1:* In a repeater chain network with  $n + 1$  repeaters  $\{u_0, u_1, \dots, u_n\}$ , if the probability of generating an elementary pair per attempt is one, the probability of a successful BSM is  $(q)$ , the capacity of an elementary link  $(u_i, u_{i+1})$  (for  $0 \leq i \leq n - 1$ ) is denoted by  $C_i$  and if the repeaters follow the prepare and swap protocol for generating EPR-pairs, then the expected end-to-end entanglement generation rate  $r_{u_0, u_n}$  is

$$r_{u_0, u_n} = (q)^{n-1} \min\{C_0, \dots, C_{n-1}\}. \quad (22)$$

Notice that, the EPR-pair generation rate for this protocol is exactly the same as the EPR-pair generation rate, proposed in (4), which we use for the path-based formulation. In the aforementioned sections, we show that both of the path-based and the edge-based formulations are equivalent. This implies, the rates extracted from Algorithm 1 can be achieved with this prepare and swap protocol.

#### 2) PREPARE AND SWAP PROTOCOL FOR AN ARBITRARY NETWORK

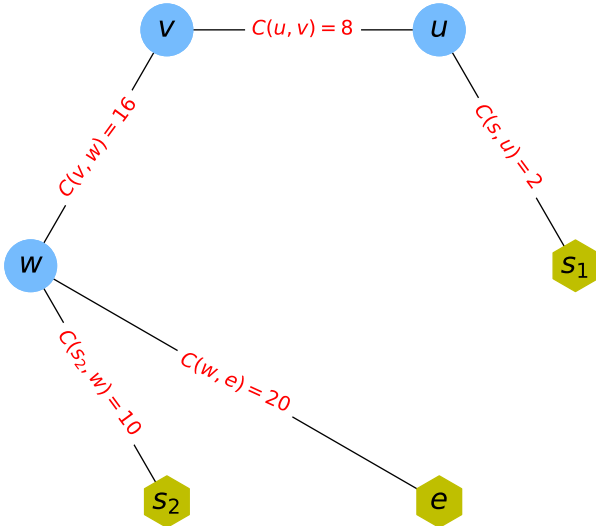
In a quantum network, if there are multiple demands then one link might be shared between multiple paths. From Algorithm 1, we get the set of paths and the desired EPR-pair generation rate  $r_p$ , across each of the paths  $p$ , passing through an edge  $(u, v)$ . In this scenario, we use the prepare and swap protocol for each of the paths in a sequential manner or round-robin manner.<sup>3</sup> From Lemma 1, we get that, to achieve the rate  $r_p$ , we need to generate on average  $\frac{r_p}{(q)^{|p|-1}}$  elementary EPR-pairs per second across each of the elementary links  $(u, v)$  along the path  $p$ . Here, we also assume that the elementary link generation is a deterministic event, i.e., the two nodes  $u$  and  $v$  connecting the elementary link  $(u, v)$  can generate exactly  $C(u, v)$  EPR-pairs per second. Hence, each of the paths uses the elementary link  $(u, v)$  for  $\frac{r_p}{(q)^{|p|-1}C(u, v)}$  seconds (on average) for generating the required elementary EPR-pairs<sup>4</sup>. For generating  $r_p$  EPR-pairs per second, all the nodes along the path  $p$  need to use the elementary links for at most  $\max_{(u,v) \in p} \left\{ \frac{r_p}{(q)^{|p|-1}C(u, v)} \right\}$  seconds, which is equal to  $\frac{r_p}{(q)^{|p|-1} \min_{(u,v) \in p} \{C(u, v)\}}$  seconds. This gives an upper bound on the average storage time of the quantum memory for generating the EPR-pairs along the path  $p$ . In Fig. 6, we give an example of allocating EPR-pair generation resources for multiple demands. For more sophisticated resource allocation techniques we refer to the techniques that are proposed in [33].

### D. COMPLEXITY ANALYSIS

In this section, we analyze the complexity of the LP formulations as well as the path extraction and rate allocation

<sup>3</sup>One can use a more sophisticated scheduling algorithm for allocating the EPR-pair generation resources across an elementary link. The details of this scheduling algorithm are beyond the scope of this article.

<sup>4</sup>Note that, if  $\frac{r_p}{(q)^{|p|-1}}$  is a rational number then one can achieve this average rate. For an irrational value of  $\frac{r_p}{(q)^{|p|-1}}$  we need to approximate it to a rational number.

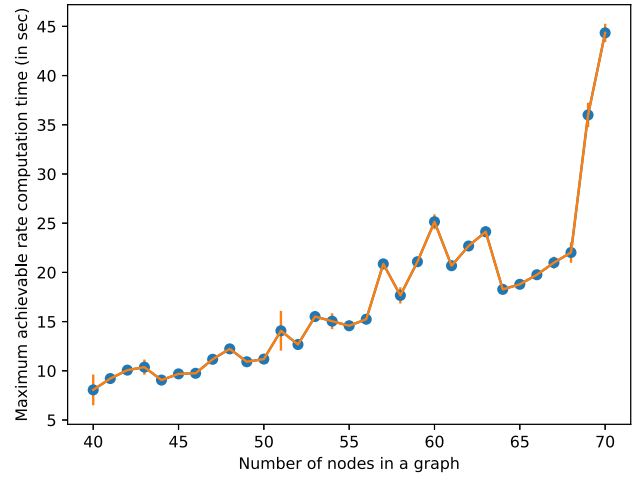


**FIGURE 6.** Repeater network  $G = (V, E, C)$  with the demand set  $D = \{(s_1, e), (s_2, e)\}$ . All the repeaters use BSM for the entanglement swap. Here we assume that the success probability of the BSM is  $q = \frac{1}{2}$ . From the Algorithm 1, we get a path  $p_1 = ((s_1, u), (u, v), (v, w), (w, e))$  for the demand  $(s_1, e)$  and a path  $p_2 = ((s_2, w), (w, e))$  for the demand  $(s_2, e)$ . The desired rate across  $p_1$  is  $r_{p_1} = 0.25$  EPR-pairs per second and the desired rate across  $p_2$  is  $r_{p_2} = 5$  EPR-pairs per second. If we use the prepare and swap protocol along both of the paths separately then we get the desired EPR-pairs generation rate. However, here, the elementary link  $(w, e)$  is being shared by both of the paths. In this case, a simple scheduling technique would be to distribute the EPR-pairs sequentially. For example, at the beginning the elementary link  $(w, e)$  generates the EPR-pairs for the demand  $(s_1, e)$ . For generating on average 0.25 EPR-pairs across the path  $p_1$ , the elementary link  $(w, e)$  has to generate  $0.25 \times 2^{|p_1|-1} = 2$  EPR-pairs per second. Hence, it can generate 2 EPR-pairs within  $\frac{2}{20} = 0.1$  seconds (on average). Then, it can start generating the EPR-pairs for the demand  $(s_2, e)$ . For this demand, the elementary link  $(w, e)$  has to generate  $5 \times 2^{|p_2|-1} = 10$  EPR-pairs per second. Hence, it will take on average  $\frac{10}{20} = 0.5$  seconds to generate 10 EPR-pairs.

algorithm (Algorithm 1). The edge-based LP-formulation, proposed in this article is based on the modified network graph  $G'$  and modified demand set  $D_{\text{mod}}$  and the running time of the edge-based LP-formulation solver depends on the size of this network and modified demand set. In the following lemma, we give an upper bound on the size of  $G'$  and  $D_{\text{mod}}$ .

**Lemma 2:** The edge-based formulation, proposed in (6)–(9), has at most  $N = |D||E||V|$  variables and  $M = |V|^2|E||D| + |V||E| + |V|^3|D|$  constraints, where  $|V|$ ,  $|E|$ ,  $|D|$  denote the total number of repeater nodes, total number of edges in the network graph  $G$  and size of the demand set  $D$ , respectively.

*Proof:* The edge-based formulation, is based on the modified network  $G' = (V', E', C')$ , which we construct from the actual Internet network  $G = (V, E, C)$ . In  $G'$ , we create at most  $l_{\text{max}}$  copies of each of the nodes and edges. This implies,  $|V'| \leq l_{\text{max}}|V|$  and  $|E'| \leq l_{\text{max}}|E|$ . As,  $l_{\text{max}} = \max\{l_1, \dots, l_{|D|}\}$ , hence,  $l_{\text{max}} \leq |V|$ . This implies,  $|V'| \leq |V|^2$  and  $|E'| \leq |E||V|$ . In the construction of the edge-based formulation, for each demand, we introduce one variable



**FIGURE 7.** This plot shows the running time of the edge-based formulation that is proposed in Table 1 as a function of the number of nodes in a graph. Here, we consider the Erdős–Rényi random graphs with parameter  $p = 0.6$  as network graphs. We vary the number of nodes in the graph from  $n = 40$  to  $n = 70$ . For each value of  $n$ , we generate five sample random graphs and for each graph we generate four random demands. The capacity of each of the link is chosen uniformly at random between  $[1,400]$ -EPR-pairs per second. We assume that the success probability of the bell-state measurement is  $q = 0.5$  and the fidelity of the each of the elementary pairs is  $F = 0.9925$  and the value of the corresponding Werner parameter  $W = 0.99$ . We compute the running time of the edge-based formulation for computing the maximum achievable rates for each of the sample graphs and plot the average computation time (in seconds) on the Y-axis of the plot. For the simulation we use a server with CPU Intel Xeon Gold 6230 @ 80x 3.9 GHz and RAM 4784MiB/193114MiB.

corresponding to each edge of  $E'$ . Hence, the total number of variables for this formulation is  $N = |D||E'| = |D||E||V|$ .

In the edge-based formulation, the constraint (7) ( $g_{ij}(u, v) \geq 0$ ) holds for all  $1 \leq i \leq |D|$ , for all  $1 \leq j \leq l_i$  and for all edge  $(u, v) \in E'$ . This implies, the total number of constraints corresponding to (7) is  $|V|^2|E||D|$ . Similarly, the constraint (8) holds for all the edges  $|E'|$ . This implies, there are at most  $|V||E|$  constraints corresponding to (8). The constraint (9) should be satisfied by all the nodes in  $V'$  and for all  $1 \leq i \leq |D|$ , for all  $1 \leq j \leq l_i$ . This implies, the total number of constraints corresponding to that equation is  $|V|^3|D|$ . Hence, the total number of constraints in the edge-based formulation is  $M = |V|^2|E||D| + |V||E| + |V|^3|D|$ . ■

The previous lemma implies that the total number of variables and constraints in the LP-formulation corresponding to the maximum flow problem scales polynomially with the number of nodes in the network graph  $G$ . This implies, the time complexity of the LP solvers for this problem also scales polynomially with the number of nodes in the network graph  $G = (V, E, C)$ . Hence, one can compute the maximum achievable rate for the quantum Internet in polynomial time. For a better understanding of the running time of the LP-formulation, we run simulations on the Erdős–Rényi random graphs with parameter  $p = 0.6$  [40]. We refer to Fig. 7 for the details of the simulation.

Now, we focus on the complexity of Algorithm 1. The Algorithm 1 uses the solution of the edge-based formulation for extracting the set of paths for each of the demands. In the following proposition, we show that the size of the set of extracted paths corresponding to each demand is upper bounded by  $|V||E|$ . Then, we use this result for computing the running time of Algorithm 1.

*Proposition 1:* In Algorithm 1

$$|\mathcal{P}_{i,j}| \leq |E||V|. \quad (23)$$

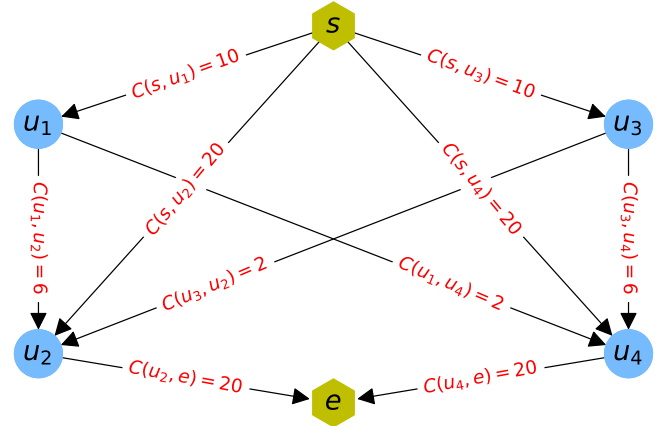
*Proof:* Due to the flow conservation property of the edge-based formulation, if for some neighbor of  $s_i^0$ ,  $F_{i,j,m}(s_i^0, v^1) > 0$ , then there exist a path  $p_{j,m}$  from  $s_i^0$  to  $e_j^1$  such that  $F_{i,j,m}(u, v) > 0$  for all  $(u, v) \in p_{j,m}$ . Note that, at each step  $m$  of the Algorithm 1 there exist at least one edge  $(u, v) \in E'$  in the discovered path  $p_{j,m}$ , such that  $F_{i,j,m+1}(u, v) = 0$ . As there are in total,  $|E'|$  number of edges and the algorithm runs until  $\sum_{v^1: (s_i^0, v^1) \in E'} F_{i,j,m+1}(s_i^0, v^1) = 0$ , so the maximum value of  $m$  could not be larger than  $|E'|$ . From the construction of the modified network,  $G'$  we have  $|E'| \leq |E||V|$ . This implies,  $|\mathcal{P}_{i,j}| \leq |E'| \leq |E||V|$ . ■

In the worst case, the total number of edges in a graph are of the order  $O(|V|^2)$ , hence, from Proposition 1, we conclude that the total number of paths extracted by Algorithm 1 for each demand scales polynomially with the number of nodes. In the following theorem, we show that, running time of Algorithm 1 is  $O(|D||V|^4|E|)$ .

*Theorem 1:* The Algorithm 1 takes the solution of the edge-based LP-formulation and extract the set of paths in  $O(|D||V|^4|E|)$  time, where  $|D|$  is the size of the demand set,  $|V|$ ,  $|E|$  denote the total number of nodes and edges in the network  $G = (V, E, C)$ .

*Proof:* In Algorithm 1, we compute the paths based on the modified network  $G' = (V', E', C')$ , which we construct from the original network  $G = (V, E, C)$ . In this modified network  $|V'| \leq |V|l_{\max} \leq |V|^2$ . In Algorithm 1 at step 11, we compute a path in the graph  $G'$ . Note that, in the worst case, it takes  $O(|V'|) = O(|V|^2)$  time to find a path between a source-destination pair in a network. According to Proposition 1, we have that for a fixed  $1 \leq i \leq |D|$  and a fixed  $1 \leq j \leq l_i$  the total number of paths discovered by Algorithm 1 is upper bounded by  $O(|V||E|)$ . Hence, for that  $i, j$ , the running time of Algorithm 1 is  $O(|V|^3|E|)$ . As,  $i \leq |D|$  and  $j \leq l_i \leq l_{\max} \leq |V|$ , so in the worst case scenario, the total running time of Algorithm 1 is upper bounded by  $O(|D||V|^4|E|)$ . ■

Note that, one can use the path-based formulation directly and apply a naive algorithm, like finding a shortest-path and compare the rate of generation along that path with the computed maximized rate so far. However, in general, there may be an exponential (in the number of nodes) number of paths between a source and a destination node in a graph. So, this type of naive algorithm can have higher time complexity. In our case, we use Algorithm 1 after solving the edge-based formulation that is described in Table 1. The edge-based



**FIGURE 8.** Repeater network  $G = (V, E, C)$  with the demand set  $D = \{(s, e, 3)\}$ . All the repeaters use BSM with success probability  $q = \frac{1}{2}$  for the entanglement swapping operation. By solving the edge-based formulation we get the values of the  $g$  function. The value of the function is zero for all of the edges, except  $g(s, u_2) = g(s, u_4) = g(u_2, e) = g(u_4, e) = 20$ . From the values of the  $g$  function we use the Algorithm 1 and get two paths  $p_1 = ((s, u_2), (u_2, e))$  and  $p_2 = ((s, u_4), (u_4, e))$ . The desired rate across both  $p_1$  and  $p_2$  is  $r_{p_1} = r_{p_2} = 10$  EPR-pairs per second. If we use the naive algorithm, where we consider all of the paths and choose the paths along, which the achievable rates are maximum, then in this example we will end up computing the rates across all six possible paths.

formulation in Table 1 is a linear program and it gives the values of  $g_{i,j}(u, v)$  corresponding to each demand across an edge  $(u, v)$ . Intuitively, the value of  $g_{i,j}(u, v)$  across an edge is the summation of the rates across multiple paths between a fixed source-destination pair  $(s_i, e_j^1)$ , i.e., information that is related to multiple paths is packed inside the value of  $g_{i,j}(u, v)$ . The job of Algorithm 1 is to unpack that information. In Fig. 8, we describe the difference between Algorithm 1 and the naive algorithm using an example.

#### IV. CONCLUSION

In this article, we use techniques from the length constrained multicommodity flow theory for developing a polynomial-time algorithm for maximizing achievable expected entanglement generation rate between multiple source-destination pairs in a quantum Internet. Here, we have maximized the end-to-end entanglement distribution rate, satisfying a constraint on the end-to-end fidelity for each of the demand. We have shown that our LP-formulation provides a maximal solution if it exists. Our path extraction algorithm produces a set of paths and the achievable rates along each of the paths. The path-extraction algorithm has high running time as a function of the path length. Here, we consider the worst case scenarios, where we assume that the length of the discovered path scales with  $|V|$ . In practical scenarios, without distillation, the end-to-end fidelity of the distributed EPR-pairs would drop drastically with the path length. Hence, it is fair to consider that the length of the allowed path increases slowly with the size of the network node set. This would make the path-extraction algorithm faster.

One can use any entanglement generation protocol for distributing EPR-pairs across the paths that are discovered by the path-extraction algorithm. However, our LP-formulation is inspired by the atomic ensemble and linear optics based quantum repeaters, where the storage time is very short and the entanglement swap operation is probabilistic in nature [21]–[23]. Here, we have also pointed out that, there exists a practical protocol, called prepare and swap protocol, which can be implemented using atomic ensemble based repeaters, and if one uses this protocol for distributing entanglement across each of the paths, then one can generate EPR pairs with the rate proposed by our path-extraction algorithm.

In this article, we focus on maximizing the end-to-end entanglement generation rate. However, one can easily extend our results for other objective functions, like minimising the weighted sum of congestion at edges.

In future work, it would be interesting to include the more realistic parameters like the bounded storage capacity, time to perform the swap operation, etc., in our model and modify our current formulations to come up with more sophisticated routing algorithms.

The proposed LP-formulations give an optimal achievable EPR-pairs distribution rate with respect to prepare and swap protocol. This protocol is practical and requires very less amount of quantum storage time. However, there exists a more sophisticated protocols that can achieve higher EPR-pairs distribution rates [41] but require higher quantum storage time. Another interesting future research direction would be to find out a protocol for distributing EPR-pairs along a chain that achieves the optimal EPR-pair generation rate and find an LP-formulation for such protocol.

## APPENDIX A: OUTLINE

In the first part of the appendix, we focus on giving a detailed proof of the equivalence of the path-based and the edge-based formulation. In the second part of the appendix, we show how the prepare and swap protocol can achieve the entanglement distribution rate, which we get as an output from the LP solver. Before going to the detailed proof, in the following appendix first, we define again some of the notations, which we use in the proof. One can find the equivalence of the edge-based formulation and the path-based formulation in Appendix D. More precisely, for the clarity, in Appendices D and D, we rewrite the path-based formulation and the edge-based formulations. We prove the equivalence between both of the formulations by showing that one can construct the solution of the edge-based formulation from the path-based formulation (see Appendix D) and vice-versa (see Appendix D). In Appendix D, we give a detailed calculation of the fidelity scaling with each entanglement swap operation. In Appendix D, we describe the entanglement distribution rate for the prepare and swap protocol across a path.

## APPENDIX B: NOTATIONS

In this section, we define again some of the notations we are going to use later in the proofs. We first start with the network

graph  $G = (V, E, C)$ , which is a directed graph, and it abstracts the quantum network. Here,  $V$  denotes the set of quantum repeaters,  $E$  denotes the set of quantum communication links, and  $C : E \rightarrow \mathbb{R}^+$  denotes the entanglement generation capacity of an edge  $(u, v) \in E$ , i.e., how many EPR-pairs the nodes  $u, v$  can generate per second. A path  $p$  between a source node and a destination node in the graph is a finite sequence of edges, which joins a sequence of distinct vertices. The path-length of a path  $p$  between a source-destination pair  $(s, e)$  is denoted by  $|p|$ . Next, we define the set of demands  $D = \{(s_1, e_1, l_1), \dots, (s_k, e_k, l_k)\}$ , where the  $i$ th element of this set (or  $i$ th demand) is a triplet  $(s_i, e_i, l_i)$  and  $s_i$  would like to share EPR-pairs with  $e_i$  using the multiple paths, whose path lengths are at most  $l_i$ . Here, we assume the size of the demand set  $|D| = k$ .

## APPENDIX C: EQUIVALENCE OF THE PATH-BASED AND EDGE-BASED FORMULATION FOR THE PREPARE AND SWAP PROTOCOL

In this section, we prove the equivalence between the path-based formulation and the edge-based formulation. Before the proof, for clarity, in the following two sections, we rewrite the path-based formulation and the edge-based formulation. Note that for the edge-based formulation of the LP construction, we use the modified network graph  $G' = (V', E', C')$ , which is constructed from  $G = (V, E, C)$ . For the clarity, here, again we rewrite the construction of  $G'$ . First, from the demand set,  $D$  we compute  $l_{\max} = \max\{l_1, \dots, l_k\}$ , where each of the  $l_i$  is related to the length constraint of the  $i$ th demand  $(s_i, e_i, l_i)$ . Then, for each node  $u \in V$ , we create  $l_{\max} + 1$  copies of  $u$ . We denote them as  $u^0, u^1, \dots, u^{l_{\max}}$ . We denote  $V^j$  as the set of the  $j$ th copy of all the nodes in  $V$ , i.e.,  $V^j := \{u^j : u \in V\}$ . For  $G'$ , the set of nodes  $V' = \bigcup_{j=0}^{l_{\max}} V^j$ . For each edge  $(u, v) \in E$  and for each  $0 \leq j < l_{\max}$ , we define,  $(u^j, v^{j+1}) \in E'$ . For each edge  $(u^j, v^{j+1}) \in E'$ , we define  $C'(u^j, v^{j+1}) = C(u, v)$ .

Note that, by construction, the path length of all the paths from  $s_i^0$  to  $e_i^j$  is exactly  $j$ . For the  $i$ th demand, we are interested in finding the paths between  $s_i$  and  $e_i$  with path length at most  $l_i$ . Hence, finding paths for the  $i$ th demand in  $G$  is same as finding paths from  $s_i^0$  to  $e_i^1, \dots, e_i^{l_i}$  in the modified network  $G'$ . For this reason, in the edge-based formulation, we decompose the  $i$ th demand  $(s_i, e_i, l_i)$  into  $l_i$  demands  $\{(s_i^0, e_i^1), \dots, (s_i^0, e_i^{l_i})\}$  and construct a new demand set called,  $D_{\text{mod}}$ . It is defined as follows:

$$D_{\text{mod}} := \{D_1, \dots, D_k\} \quad (24)$$

where for each  $1 \leq i \leq k$ ,  $D_i = \{(s_i^0, e_i^1), \dots, (s_i^0, e_i^{l_i})\}$ .

## A. PATH-BASED FORMULATION

In this section, we rewrite again the path-based formulation based on the new demand set  $D_{\text{mod}}$  and the modified network  $G'$ . Here, for the  $i, j$ th demand  $(s_i^0, e_i^j) \in D_{\text{mod}}$  we denote  $\mathcal{P}_{i,j}$  as the set of all possible paths from  $s_i^0$  to  $e_i^j$  and for each



**TABLE 4. Path-Based Formulation on the Modified Network.**

<p>Maximize</p> $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} r_p \quad (25)$ <p>Subject to :</p> $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \sum_{\substack{p \in \mathcal{P}_{i,j}: \\ (u^t, v^{t+1}) \in p}} \frac{r_p}{(q)^{ p -1}} \leq C(u, v), \forall (u, v) \in E, \quad (26)$ $\forall i \in \{1, \dots, k\}, \forall j \in \{1, \dots, l_i\}, \forall p \in \mathcal{P}_{i,j}, \quad (27)$ $r_p \geq 0.$
---

**TABLE 5. Edge-Based Formulation on the Modified Network.**

<p>Maximize</p> $\sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1). \quad (28)$ <p>Subject to :</p> <p>For all <math>1 \leq i \leq k, 1 \leq j \leq l_i,</math></p> $0 \leq t \leq l_{\max} - 1, (u, v) \in E,$ $g_{ij}(u^t, v^{t+1}) \geq 0. \quad (29)$ $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} g_{ij}(u^t, v^{t+1}) \leq C(u, v). \quad (30)$ <p>For all <math>1 \leq i \leq k, 1 \leq j \leq l_i,</math></p> <p>For all <math>u', v', w' \in V' : v' \neq s_i^0, v' \neq e_i^j,</math></p> $\sum_{u': (u', v') \in E'} g_{ij}(u', v') = \sum_{w': (v', w') \in E'} g_{ij}(v', w'). \quad (31)$
--

path  $p \in \mathcal{P}_{i,j}$  we define one variable  $r_p \in \mathbb{R}^+$ . The variable  $r_p$  denotes the flow between  $s_i^0$  to  $e_i^j$  via the path  $p$ . The aim of the path-based formulation is to maximize the sum  $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} r_p$ .

We give the exact formulation in Table IV.

## B. EDGE-BASED FORMULATION

In this section, for clarity, we rewrite the edge formulation. Here, for the  $i, j$ th demand  $(s_i^0, e_i^j) \in D_i$  (where  $D_i \in D_{\text{mod}}$ ), we define one function  $g_{ij} : E' \rightarrow \mathbb{R}^+$ . We give the edge formulation in Table D.

## C. FROM THE PATH-BASED FORMULATION TO THE EDGE-BASED FORMULATION

In this section, we show that the solution of the edge-based formulation is at least as good as the solution of the path-based formulation. In order to do so, we assume that we have the solution of the path-based formulation proposed in Table IV. From this solution, we construct a solution of the edge-based formulation, proposed in Table D. For the  $i, j$ th demand  $(s_i^0, e_i^j) \in D_{\text{mod}}$ , for each edge  $(u, v) \in E'$ , we define a function  $\tilde{g}_{ij} : E \rightarrow \mathbb{R}^+$ , as follows:

$$\tilde{g}_{ij}(u, v) := \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u, v') \in p}} \frac{r_p}{(q)^{j-1}}. \quad (32)$$

Here, we show that this  $\tilde{g}_{ij}$  is a valid solution for the edge-based formulation. In order to do so, first, we need to show that if we assume  $\tilde{g}_{ij}$  as a solution of the edge-based formulation then the objective function of the path-based formulation is the same as the objective function of the edge-based formulation.

*Proposition 2:* For all  $(u, v) \in E'$ , if we consider (32) as the definition of the function  $\tilde{g}_{ij} : E' \rightarrow \mathbb{R}^+$  then

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} r_p = \sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} \tilde{g}_{ij}(s_i^0, v^1). \quad (33)$$

*Proof:* According to the definition of  $\tilde{g}_{ij}$  in (32), we get

$$\tilde{g}_{ij}(s_i^0, v^1) = \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (s_i^0, v^1) \in p}} \frac{r_p}{(q)^{j-1}}. \quad (34)$$

This implies

$$\sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (s_i^0, v^1) \in p}} r_p = (q)^{j-1} \tilde{g}_{ij}(s_i^0, v^1). \quad (35)$$

In the path-based formulation, for the  $(i, j)$ th demand  $(s_i^0, e_i^j)$  the total entanglement generation rate is  $\sum_{p \in \mathcal{P}_{i,j}} \frac{r_p}{(q)^{j-1}}$ . As  $s_i^0$  is the source of all of those paths  $p \in \mathcal{P}_{i,j}$ , so we can write the total entanglement generation rate as follows:

$$\sum_{p \in \mathcal{P}_{i,j}} \frac{r_p}{(q)^{j-1}} = \sum_{v^1: (s_i^0, v^1) \in E'} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (s_i^0, v^1) \in p}} \frac{r_p}{(q)^{j-1}}. \quad (36)$$

Substituting the definition of  $(q)^{j-1} \tilde{g}_{ij}(s_i^0, v^1)$  from (35) to the right-hand side of (36), we get

$$\sum_{p \in \mathcal{P}_{i,j}} \frac{r_p}{(q)^{j-1}} = \sum_{v^1: (s_i^0, v^1) \in E'} (q)^{j-1} \tilde{g}_{ij}(s_i^0, v^1). \quad (37)$$

By taking the summation over all  $1 \leq i \leq k$  and  $1 \leq j \leq l_i$  on the both side of the above-mentioned equation, we can prove this proposition.  $\blacksquare$

In the rest of this section, we show that  $\tilde{g}_{ij}$  satisfies all the constraints from (29)–(31). Note that,  $\tilde{g}_{ij}$  satisfies the first constraint of the edge-formulation by construction. In the following proposition, we show that  $\tilde{g}_{ij}$  satisfies the constraint equations:  $\forall(u, v) \in E$  and for all  $0 \leq t \leq l_{\max} - 1$ ,  $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} g_{ij}(u^t, v^{t+1}) \leq C(u, v)$ .

**Proposition 3:** For all  $(u, v) \in E'$ , if we consider (32) as the definition of the function  $\tilde{g}_{ij} : E' \rightarrow \mathbb{R}^+$  then, for all  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ ,  $0 \leq t \leq l_{\max} - 1$ ,  $\forall(u, v) \in E$

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \tilde{g}_{ij}(u^t, v^{t+1}) \leq C(u, v). \quad (38)$$

*Proof:* For any  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ , and  $0 \leq t \leq l_{\max} - 1$  and for any edge  $(u, v) \in E$ , we define the function  $\tilde{g}_{ij}$  in (32) as follows:

$$\tilde{g}_{ij}(u^t, v^{t+1}) = \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u^t, v^{t+1}) \in p}} \frac{r_p}{(q)^{j-1}}. \quad (39)$$

By taking the sum over all the values of  $i, j$ , and  $t$ , we get

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \tilde{g}_{ij}(u^t, v^{t+1}) = \sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u^t, v^{t+1}) \in p}} \frac{r_p}{(q)^{j-1}}. \quad (40)$$

From the constraint (26) of the path formulation, we get

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u^t, v^{t+1}) \in p}} \frac{r_p}{(q)^{|p|-1}} \leq C(u, v). \quad (41)$$

Substituting this inequality in (42), we get

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \tilde{g}_{ij}(u^t, v^{t+1}) \leq C(u, v). \quad (42)$$

In the following proposition, we prove that  $\tilde{g}_{ij}$  satisfies the constraint proposed in (31), which is, for all  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ ,  $u, v, w' \in V' : v' \neq s_i^0, v' \neq e_i^j$

$$\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v') = \sum_{w':(v',w') \in E'} \tilde{g}_{ij}(v', w').$$

**Proposition 4:** For all  $(u', v') \in E'$ , if we consider (32) as the definition of the function  $\tilde{g}_{ij} : E' \rightarrow \mathbb{R}^+$  then, for all  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ ,  $v' \in V' : v' \neq s_i^0, v' \neq e_i^j$

$$\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v') = \sum_{w':(v',w') \in E'} \tilde{g}_{ij}(v', w').$$

*Proof:* From the definition of  $\tilde{g}_{ij}$  in (32), we have

$$\tilde{g}_{ij}(u', v') = \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p}} \frac{r_p}{(q)^{j-1}}. \quad (43)$$

Using this relation, for any node  $v' \in V'$ , such that  $v' \neq s_i^0, v' \neq e_i^j$ , we can rewrite the expression  $\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v')$  in the following manner:

$$\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v') = \sum_{u':(u',v') \in E'} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p}} \frac{r_p}{(q)^{j-1}}. \quad (44)$$

For each edge  $(u', v') \in E'$  (where  $v' \in V' \setminus \{s_i^0, e_i^j\}$ ) can be part of multiple paths  $p \in \mathcal{P}_{i,j}$ . This implies

$$\sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p}} \frac{r_p}{(q)^{j-1}} = \sum_{w':(v',w') \in E'} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p \\ (v',w') \in p}} \frac{r_p}{(q)^{j-1}}.$$

Substituting the above-mentioned value of  $\sum_{p \in \mathcal{P}_{i,j}: \frac{r_p}{(q)^{j-1}}}$  in (44), we get

$$\sum_{\substack{u': \\ (u',v') \in E'}} \tilde{g}_{ij}(u', v') = \sum_{\substack{u': \\ (u',v') \in E'}} \sum_{\substack{w': \\ (v',w') \in E'}} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p \\ (v',w') \in p}} \frac{r_p}{(q)^{j-1}}. \quad (45)$$

At the right-hand side of the above-mentioned equation, by interchanging the summation over  $u'$  and  $w'$ , we get

$$\sum_{\substack{u': \\ (u',v') \in E'}} \tilde{g}_{ij}(u', v') = \sum_{\substack{w': \\ (v',w') \in E'}} \sum_{\substack{u': \\ (u',v') \in E'}} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (u',v') \in p \\ (v',w') \in p}} \frac{r_p}{(q)^{j-1}}.$$

As for an intermediate node  $v'$ , the total number of paths, incoming to it same as the total number of paths leaving it, so we can rewrite the above-mentioned expression as

$$\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v') = \sum_{w':(v',w') \in E'} \sum_{\substack{p \in \mathcal{P}_{i,j}, \\ (v',w') \in p}} \frac{r_p}{(q)^{j-1}}.$$

According to the definition of  $\tilde{g}_{ij}$ , we have,  $\tilde{g}_{ij}(v', w') = \sum_{p \in \mathcal{P}_{i,j}: \frac{r_p}{(q)^{j-1}}}$ . By substituting this relation on the right-hand side of the above-mentioned expression, we get

$$\sum_{u':(u',v') \in E'} \tilde{g}_{ij}(u', v') = \sum_{w':(v',w') \in E'} \tilde{g}_{ij}(v', w').$$

Propositions 3 and 4 certifies that  $\tilde{g}_{ij}$  all the constraints, proposed in the edge-based formulation and Proposition 2 proves that  $\tilde{g}_{ij}$  corresponds to the objective function of the edge formulation. This implies,  $\tilde{g}_{ij}$  is a valid solution to the edge-based formulation. In the following section, we show how to construct the path-based formulation from the edge-based formulation.

#### D. FROM THE EDGE-BASED FORMULATION TO THE PATH-BASED FORMULATION

In this section, we show that the path-based formulation is at least as good as the edge-based formulation. We assume that

---

**Algorithm 3:** Path Extraction and Rate Allocation Algorithm.
 

---

**Input:** The solution we obtain from the edge-based formulation, i.e.,  $\{\{g_{ij}(u, v)\}_{(u, v) \in E'}\}_{1 \leq i \leq k, 1 \leq j \leq l_i}$ .

**Output:** Set of paths as well as the rate across each of the paths  $\{\mathcal{P}_{i,j}\}_{1 \leq i \leq k, 1 \leq j \leq l_i}$ .

```

1:  for ( $i = 1; i \leq k; i++$ )do
2:    for ( $j = 1; j \leq l_i; j++$ )do
3:       $m = 0$ .
4:       $F_{i,j} = g_{ij}$ .
5:       $\mathcal{P}_{i,j} = \emptyset$ .
6:      while  $\sum_{v:(s_i^0, v^1) \in E'} F_{i,j,m}(s_i^0, v^1) > 0$  do
7:        Find a path  $p_{j,m}$  from  $s_i^0$  to  $e_i^j$  such that,
8:         $\forall (u, v) \in p_{j,m}, F_{i,j,m}(u, v) > 0$ 
9:         $\tilde{r}_{p_{j,m}} = (q)^{j-1} \min_{(u,v) \in p_{j,m}} \{F_{i,j,m}(u, v)\}$ 
10:        $\forall (u, v) \in p_{j,m}$ , we define  $F_{i,j,m+1}(u, v)$ 
           as,
11:        $F_{i,j,m+1}(u, v) := F_{i,j,m}(u, v) - \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}$ .
12:        $\mathcal{P}_{i,j} = \mathcal{P}_{i,j} \cup (p_{j,m}, \tilde{r}_{p_{j,m}})$ .
13:        $m = m + 1$ .
14:     end while
15:   end for
16: end for

```

---

we have the solution of the edge-based formulation, defined in Appendix B. From this solution, we extract a solution for the path-based formulation. We use Algorithm 3 for extracting the paths and the achievable rates for the path-based formulation. In Algorithm 3, at step 8, for the  $i, j$ th demand  $(s_i^0, e_i^j) \in D_{\text{mod}}$ , we compute the entanglement distribution rate  $\tilde{r}_{p_{j,m}}$  across a path  $p_{j,m} \in \mathcal{P}_{i,j}$ . In order to be a valid solution of the path-based formulation, proposed in Table IV, we need to show that the extracted rates should satisfy the constraints in (26) and (27). We also need to show that, the objective function, which is computed from these extracted rates should be the same as the objective function of the edge-based formulation. In order to do that, first, we need to prove some properties of the function  $F_{i,j,m}$ , used in Algorithm 3. In the following proposition, we show that for all the edges  $(u, v) \in E'$ , the value of the function  $F_{i,j,m} \geq 0$ .

**Proposition 5:** In Algorithm 3 for all  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ ,  $m \geq 0$ ,  $(u, v) \in E'$

$$F_{i,j,m}(u, v) \geq 0. \quad (46)$$

*Proof:* In the Algorithm 3, after each iteration over  $m$ , we compute  $F_{i,j,m}(u, v) = F_{i,j,m-1}(u, v) - \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}}$ , where  $\tilde{r}_{p_{j,m-1}} = (q)^{j-1} \min_{(u,v) \in p_{j,m-1}} \{F_{i,j,m-1}(u, v)\}$ . This implies, at least for one edge  $(u, v) \in E'$ ,  $F_{i,j,m}(u, v) = 0$  and according to the path-extraction algorithm that edge will not be part of any other paths. For the other edges  $(u, v) \in p_{j,m-1}$

$$\tilde{r}_{p_{j,m-1}} \leq F_{i,j,m-1}(u, v)(q)^{j-1}.$$

This implies

$$F_{i,j,m-1}(u, v) - \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}} \geq 0.$$

Hence

$$F_{i,j,m}(u, v) \geq 0. \quad (47)$$

In the following proposition, we show that for each demand  $i, j$  the total number of paths  $|\mathcal{P}_{i,j}|$  is upper bounded by  $|E'|$ .

**Proposition 6:** In Algorithm 3

$$|\mathcal{P}_{i,j}| \leq |E'|. \quad (48)$$

*Proof:* Due to the flow conservation property of the edge-based formulation, if for some neighbor of  $s_i^0$ ,  $F_{i,j,m}(s_i^0, v^1) > 0$  then there exist a path  $p_{j,m}$  from  $s_i^0$  to  $e_i^j$  such that  $F_{i,j,m}(u, v) > 0$  for all  $(u, v) \in p_{j,m}$ . Note that, at each step  $m$  of the Algorithm 3 there exist at least one edge  $(u, v) \in E'$  in the discovered the path  $p_{j,m}$ , such that  $F_{i,j,m+1}(u, v) = 0$ . As there are in total,  $|E'|$  number of edges and the algorithm runs until  $\sum_{v^1:(s_i^0, v^1) \in E'} F_{i,j,m+1}(s_i^0, v^1) = 0$ , so the maximum value of  $m$  could not be larger than  $|E'|$ . ■

In the edge-based formulation, we have the flow conservation for each  $g_{ij}$ . In the following proposition, we show that the flow conservation also holds for all  $F_{i,j,m}$ .

**Proposition 7:** In Algorithm 3 for all  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ ,  $m \geq 0$ , and  $\forall v' \in V' \setminus \{s_i^0, e_i^j\}$

$$\sum_{\substack{u': \\ (u', v') \in E'}} F_{i,j,m}(u, v') = \sum_{\substack{w': \\ (v, w') \in E'}} F_{i,j,m}(v, w'). \quad (49)$$

In Algorithm 3, for the  $i, j$ th demand, we discover a path with each iteration over  $m$ . We denote the set of discovered paths up to the  $m$ th iteration as  $\mathcal{P}_{i,j,m}$ . After each discovery of the path, we allocate the rate  $r_{p_{j,m}}$  across that path using a function  $F_{i,j,m}$  and compute the value of the new function  $F_{i,j,m+1}$  by subtracting the allocated rate from  $F_{i,j,m}$ . For  $m = 0$ , we have  $F_{i,j,0} = g_{ij}$ . This implies, for every iteration  $m$ , we can rewrite  $g_{ij}$  as a function of  $F_{i,j,m}$  and the sum of the allocated rates so far. In the edge formulation, the function  $g_{ij}$  satisfies the flow conservation property. Here, we use this relation and substitute  $g_{ij}$  with the function of  $F_{i,j,m}$ , then by doing some simple algebraic manipulation, we could show that  $F_{i,j,m}$  also satisfies the flow conservation property.

**Proof.** In Algorithm 3, suppose for any  $m > 0$ , the set of discovered paths are  $\mathcal{P}_{i,j,m-1}$ . This implies, for any edge  $(u, v) \in E'$

$$F_{i,j,m}(u, v) = g_{ij}(u, v) - \sum_{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}}.$$

By exchanging the position of  $g_{ij}(u, v)$  and  $F_{i,j,m}(u, v)$  in the abovementioned equation, we get

$$g_{ij}(u, v) = F_{i,j,m}(u, v) + \sum_{\substack{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}: \\ (u, v) \in p_{j,m-1}}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}}. \quad (50)$$

From the flow conservation property (31) of the edge formulation, we have

$$\sum_{\substack{u': \\ (u, v) \in E'}} g_{ij}(u, v) = \sum_{\substack{w': \\ (v, w') \in E'}} g_{ij}(v, w').$$

Substituting the value of  $g_{ij}(u, v)$  from (50), we get

$$\begin{aligned} & \sum_{u: (u, v) \in E'} \left( F_{i,j,m}(u, v) + \sum_{\substack{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}: \\ (u, v) \in p_{j,m-1}}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}} \right) \\ &= \sum_{w': (v, w') \in E'} \left( F_{i,j,m}(v, w') + \sum_{\substack{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}: \\ (v, w') \in p_{j,m-1}}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}} \right). \end{aligned} \quad (51)$$

As for an intermediate node  $v$ , the number of the incoming paths to it is the same as the number of outgoing paths from it. This implies

$$\sum_{\substack{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}: \\ (u, v) \in p_{j,m-1}}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}} = \sum_{\substack{p_{j,m-1} \in \mathcal{P}_{i,j,m-1}: \\ (v, w') \in p_{j,m-1}}} \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}}.$$

By substituting this relation in (51), we get

$$\sum_{u: (u, v) \in E'} F_{i,j,m}(u, v) = \sum_{w': (v, w') \in E'} F_{i,j,m}(v, w').$$

In the following proposition, we show that, the rates we compute in Algorithm 3 satisfies the condition (26) of the path-based formulation.

**Proposition 8:** For all  $(u, v) \in E$

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \sum_{\substack{p_j \in \mathcal{P}_{i,j}: \\ (u, v^{t+1}) \in p_j}} \frac{\tilde{r}_{p_j}}{(q)^{j-1}} \leq C(u, v) \quad (52)$$

where  $\tilde{r}_{p_{j,m}}$  is defined in Algorithm 3 (see step 8).

In Algorithm 3, for the  $i, j$ th demand, we discover a path with each iteration over  $m$ . We denote the set of discovered paths up to the  $m$ th iteration as  $\mathcal{P}_{i,j,m}$ . After each discovery of the path, we allocate the rate  $\tilde{r}_{p_{j,m}}$  across that path using a function  $F_{i,j,m}$  and compute the value of the new function  $F_{i,j,m+1}$  by subtracting the allocated rate from  $F_{i,j,m}$ . For  $m = 0$ , we have  $F_{i,j,0} = g_{ij}$ . This implies, for every iteration  $m$ , we can rewrite  $g_{ij}$  as a function of  $F_{i,j,m}$  and the sum of the allocated rates so far. In the edge formulation, for

every edge  $(u, v) \in E$  and for every  $0 \leq t \leq l_{\max} - 1$ , we have that  $\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} g_{ij}(u, v^{t+1}) \leq C(u, v)$ . Here, we use this relation and substitute  $g_{ij}$  with the function of  $F_{i,j,m}$ , then by doing some simple algebraic manipulation we could show that the sum of the extracted rate is also upper bounded by the capacity of that edge.

*Proof.* In Algorithm 3, suppose for any  $m \geq 0$  and for any  $1 \leq i \leq k$ ,  $1 \leq j \leq l_i$ , and  $0 \leq t \leq l_{\max} - 1$ , the set of discovered paths is  $\mathcal{P}_{i,j,m}$ . This implies, for any edge  $(u, v) \in E$ , and for any  $0 \leq t \leq l_{\max} - 1$

$$F_{i,j,m+1}(u, v^{t+1}) = g_{ij}(u, v^{t+1}) - \sum_{\substack{p_{j,m} \in \mathcal{P}_{i,j,m}: \\ (u, v^{t+1}) \in p_{j,m}}} \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}.$$

From Proposition 5, we have that for all  $0 \leq m \leq |E'|$  and for all the edges  $(u, v^{t+1}) \in E'$ ,  $F_{i,j,m}(u, v^{t+1}) \geq 0$ . This implies, when  $\sum_{v: (s_i^0, v) \in E'} F_{i,j,m}(s_i^0, v) = 0$ , then

$$\begin{aligned} g_{ij}(u, v^{t+1}) - \sum_{\substack{p_j \in \mathcal{P}_{i,j}: \\ (u, v^{t+1}) \in p_j}} \frac{\tilde{r}_{p_j}}{(q)^{j-1}} &\geq 0 \\ g_{ij}(u, v^{t+1}) &\geq \sum_{\substack{p_j \in \mathcal{P}_{i,j}: \\ (u, v^{t+1}) \in p_j}} \frac{\tilde{r}_{p_j}}{(q)^{j-1}}. \end{aligned} \quad (53)$$

From the edge-based formulation, we have

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} g_{ij}(u, v^{t+1}) \leq C(u, v)$$

for all the edges  $(u, v) \in E$  and for all  $0 \leq t \leq l_{\max} - 1$ . Substituting this relation in (53), we get

$$\sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{t=0}^{l_{\max}-1} \sum_{\substack{p_j \in \mathcal{P}_{i,j}: \\ (u, v^{t+1}) \in p_j}} \frac{\tilde{r}_{p_j}}{(q)^{j-1}} \leq C(u, v). \quad (54)$$

We finish this section by showing the equivalence of the objective functions for both of the formulations.

**Proposition 9:** In Algorithm 3

$$\sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v: (s_i^0, v) \in E'} g_{ij}(s_i^0, v) = \sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in \mathcal{P}_{i,j}} \tilde{r}_p. \quad (55)$$

Due to the flow conservation property of the function  $F_{i,j,m}$ , the Algorithm 3 runs until,  $F_{i,j,m+1}(s_i^0, v) = 0$ , for all the neighbors of the source node  $s_i^0$ . In the previous propositions, we establish a relation between  $g_{ij}$  and  $F_{i,j,m+1}$  and the set of discovered paths, i.e.,  $F_{i,j,m+1}(s_i^0, v) = g_{ij}(s_i^0, v) - \sum_{\substack{p_{j,m} \in \mathcal{P}_{i,j,m}: \\ (s_i^0, v) \in p_{j,m}}} \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}$ . If all the paths are discovered, then the value of  $F_{i,j,m+1}(s_i^0, v)$  becomes zero and  $g_{ij}$  will only be a



function of all the discovered paths. Here, we use this relation to prove the equivalence of the objective functions.

*Proof:* From Proposition 7, we have that the functions  $F_{i,j,m}$  follow the flow conservation. From the Proposition 5, we have that  $F_{i,j,m}(u, v) \geq 0$  for all  $0 \leq m \leq |E'|$ . This implies, for a fixed  $i, j$ , for any value of  $m$ , we can find one path  $p_{j,m}$  with nonzero  $\tilde{r}_{p_{j,m}}$  until  $\sum_{v^1: (s_i^0, v^1) \in E'} F_{i,j,m}(s_i^0, v^1) = 0$ . According to the Algorithm 3 after each iteration if an edge  $(s_i^0, v^1) \in p_{j,m}$

$$F_{i,j,m+1}(s_i^0, v^1) = F_{i,j,m}(s_i^0, v^1) - \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}. \quad (56)$$

If for some  $i, j, m$ ,  $F_{i,j,m+1}(s_i^0, v^1) = 0$  then from (56), we have

$$F_{i,j,m}(s_i^0, v^1) = \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}.$$

Using the recurrence relation of (56), we can rewrite the abovementioned expression as

$$F_{i,j,m-1}(s_i^0, v^1) - \frac{\tilde{r}_{p_{j,m-1}}}{(q)^{j-1}} = \frac{\tilde{r}_{p_{j,m}}}{(q)^{j-1}}.$$

If we continue like this until  $m = 0$ , then we get

$$F_{i,j,0}(s_i^0, v^1) = \sum_{m'=0}^m \frac{\tilde{r}_{p_{j,m'}}}{(q)^{j-1}}.$$

Note that,  $F_{i,j,0} = g_{ij}$ . This implies

$$g_{ij}(s_i^0, v^1) = \sum_{m'=0}^m \frac{\tilde{r}_{p_{j,m'}}}{(q)^{j-1}} \\ \sum_{m'=0}^m \tilde{r}_{p_{j,m'}} = (q)^{j-1} g_{ij}(s_i^0, v^1). \quad (57)$$

This is true for all  $m$ . This implies

$$(q)^{j-1} g_{ij}(s_i^0, v^1) = \sum_{\substack{p \in P_{i,j} \\ (s_i^0, v^1) \in p}} \tilde{r}_p. \quad (58)$$

By taking sum on both sides on  $i$  and  $j$ , we get

$$\sum_{i=1}^k \sum_{j=1}^{l_i} (q)^{j-1} \sum_{v^1: (s_i^0, v^1) \in E'} g_{ij}(s_i^0, v^1) = \sum_{i=1}^k \sum_{j=1}^{l_i} \sum_{p \in P_{i,j}} \tilde{r}_p. \quad (59)$$

#### APPENDIX D: NOISE MODEL AND THE CAPACITY OF THE ELEMENTARY LINKS

In this section, we calculate the capacity of the elementary links using a noise model for elementary EPR-pair generation. Moreover, we also explain how to approximate deterministic elementary EPR-pair generation.

In this model, the EPR-pairs are distributed between two quantum memories within an elementary link by means of entangled photon pair sources and linear-optic Bell state

measurements. The setup is shown in Fig. 9. The photon pair sources on both ends of an elementary link generate entangled photon pairs and send one photon to the middle Bell-state measurement station, and send the other photon to their quantum memories. In the middle Bell-state measurement station, an entanglement swapping operation is performed. Note that, here for a successful entanglement swapping operation it is necessary for both of the photons to arrive at the middle station at the same time. The result of the Bell state measurement is heralded by the middle station to the corresponding two quantum repeaters. Due to the losses, other noise in the communication channel, and the probabilistic nature of the entanglement swapping operations the EPR-pair generation is a probabilistic process. We can abstract the impact of all of the losses using a term  $0 \leq P_{\text{gen}} \leq 1$ . It denotes the probability of generating an EPR-pair in one attempt. To boost the success probability of a single attempt, we consider that the entangled photon pair sources can generate multiple entangled photon pairs with different frequency modes in a single attempt. Suppose the source can generate  $m$  such frequency modes and each of them can create an EPR-pair with probability  $P_{\text{gen}}$ . We can create an elementary EPR-pair if the middle station manages to perform a successful Bell state measurement operation for at least one out of  $m$  modes. For this scenario, the probability of generating an EPR-pair per attempt is at least  $(1 - (1 - P_{\text{gen}})^m)$ . Here, we assume that the photon pair source can attempt to generate the  $m$  entangled photon pairs at a rate  $\mu_s$  per second and each of such attempts can create an elementary EPR-pair with probability at least  $(1 - (1 - P_{\text{gen}})^m)$ . Hence, the capacity of an elementary link is  $\mu_s(1 - (1 - P_{\text{gen}})^m)$ .

In this article, for the prepare and swap protocol, we assume that we can choose  $m$  high enough such that the success probability  $(1 - (1 - P_{\text{gen}})^m)$  is as close to one as desired. More concretely, for any  $\epsilon > 0$  we choose  $m$  such that

$$(1 - (1 - P_{\text{gen}})^m) = 1 - \epsilon.$$

From this equation, we get the number of required modes as

$$m = \frac{\log \epsilon}{\log(1 - P_{\text{gen}})}.$$

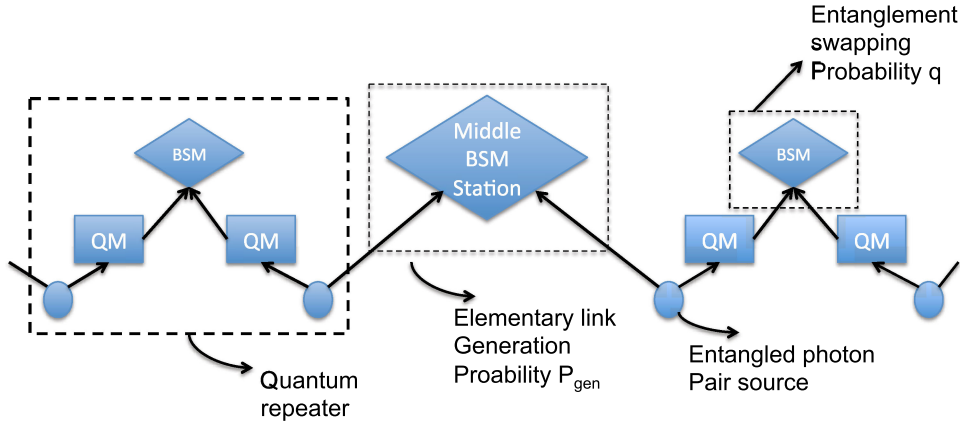
If  $(1 - (1 - P_{\text{gen}})^m) \approx 1$ , then the capacity of an elementary link  $(u, v)$  is

$$C(u, v) = \mu_s. \quad (60)$$

Note that, during the elementary link creation, all of the operations may introduce some noise in the shared EPR-pairs. In this article, we abstract the impact of the noise using the elementary EPR-pair fidelity  $F$ .

#### APPENDIX E: QUANTUM MEMORIES AND THE ENTANGLEMENT SWAPPING OPERATION

This section gives a brief overview of the entanglement swapping operation and how to compute its success probability  $q$ .



**FIGURE 9.** Abstract view of the elementary links, quantum repeaters and the success probabilities. Please see Appendices D and E for a detailed description.

In our model, a quantum repeater stores its part of the shared EPR-pairs inside its quantum memories. An intermediate quantum repeater retrieves its part of the shared EPR-pairs from the memories in a repeater chain network if it manages to create EPR-pairs with both of its neighbors. After successfully retrieving the photons from the memories, an entanglement swapping operation is performed using a BSM (see Fig. 9). All of the operations involved—retrieving the photons from memory, transmitting them to the Bell-state measurement station, and the measurements themselves—can be probabilistic. We abstract the total success probability of the combined operation using the parameter  $q$ .

#### APPENDIX F: FIDELITY SCALING AS A FUNCTION OF THE NUMBER OF SWAP OPERATIONS

This section shows for completeness how the fidelity of the EPR-pairs drops with the number of entanglement swapping operations. First, we give a bit more technical details of the entanglement swapping operations. Note that, here, we assume that the reader has a basic understanding of quantum information, i.e., the reader is familiar with the concept of bra-ket notation, projective measurements, etc.

Suppose, there are two EPR-pairs, one of them is of the form  $|\phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)_{AB}$  (the subscript  $AB$  denotes that the first qubit is stored in the quantum memory  $A$  and the second qubit is stored inside the quantum memory  $B$ ) and the other one could be of the form  $|\phi^\pm\rangle_{B'C}$  or  $|\psi^\pm\rangle_{B'C}$ . Here,  $|\phi^\pm\rangle_{B'C} = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)_{B'C}$  and  $|\psi^\pm\rangle_{B'C} = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)_{B'C}$ . In the introduction, we mention that, if we perform an entanglement swapping operation between the two qubits stored in the memories  $B, B'$ , then the qubits in the memories  $A$  and  $C$  (also qubits in  $B, B'$ ) share an EPR-pair. More technically, entanglement swapping operation is a two-qubit projective measurement (Bell-state measurement), where the projective operator is of the form  $\{|\psi^\pm\rangle_{BB'}\langle\psi^\pm|, |\phi^\pm\rangle_{BB'}\langle\phi^\pm|\}$ . After the measurement the state of the qubits in the memories  $B, B'$  is randomly projected onto one of the Bell-states  $|\psi^\pm\rangle, |\phi^\pm\rangle$ . After the

measurement, depending on the state in the memories  $B, B'$ , we get another EPR-pair in the memories  $A, C$ . We explain the step by step process with an example.

If  $A, B$  and  $B', C$  share EPR-pairs of the form  $|\phi^+\rangle_{AB} \otimes |\phi^+\rangle_{B'C}$ , then we can write the joint state as

$$(|\phi^+\rangle \otimes |\phi^+\rangle)_{ABB'C} = \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle)_{ABB'C}. \quad (61)$$

After interchanging the qubits that are stored in the memories, we get

$$(|\phi^+\rangle \otimes |\phi^+\rangle)_{ACBB'} = \frac{1}{2}(|0000\rangle + |0110\rangle + |1010\rangle + |1111\rangle)_{ACBB'}. \quad (62)$$

It is easy to verify that one can write  $|00\rangle = \frac{1}{\sqrt{2}}(|\phi^+\rangle + |\phi^-\rangle)$ ,  $|01\rangle = \frac{1}{\sqrt{2}}(|\psi^+\rangle + |\psi^-\rangle)$ ,  $|10\rangle = \frac{1}{\sqrt{2}}(|\psi^+\rangle - |\psi^-\rangle)$ ,  $|11\rangle = \frac{1}{\sqrt{2}}(|\phi^+\rangle - |\phi^-\rangle)$ . Substituting these relations for the qubits that are stored in the memories  $B, B'$  in (62), we get

$$(|\phi^+\rangle \otimes |\phi^+\rangle)_{ACBB'} = \frac{1}{2\sqrt{2}}(|00\rangle \otimes (|\phi^+\rangle + |\phi^-\rangle) + |01\rangle \otimes (|\psi^+\rangle + |\psi^-\rangle) + |10\rangle \otimes (|\psi^+\rangle - |\psi^-\rangle) + |11\rangle \otimes (|\phi^+\rangle - |\phi^-\rangle))_{ACBB'}.$$

We can rewrite the abovementioned expression and get the following form:

$$(|\phi^+\rangle \otimes |\phi^+\rangle)_{ACBB'} = \frac{1}{2}(|\phi^+\rangle \otimes |\phi^+\rangle + |\psi^+\rangle \otimes |\psi^+\rangle + |\psi^-\rangle \otimes |\psi^-\rangle + |\phi^-\rangle \otimes |\phi^-\rangle)_{ACBB'}.$$

From the right-hand side of the above-mentioned expression, it is clear that after the Bell-state measurement on the qubits in  $B, B'$  the qubits in  $A, C$  share one of the four EPR-pairs, and its form depends on the BSM outcome. If  $A, C$  would like to share  $|\phi^+\rangle$  after the BSM, then we need to

apply some single qubit Pauli operations on the qubits that are stored in the memories  $A, C$ . Of course, this correction depends on the measurement outcome of the BSM. If we keep the same set of corrections corresponding to the outcome of the BSM and if we vary the initial shared EPR-pair between the memories  $B, C$  then after the BSM, the shared state  $|\Psi\rangle_{AC}$  between  $A, C$  changes. Using the same technique that we explain in the last paragraph, one can verify that  $|\Psi\rangle_{AC}$  can have the following forms:

$$|\Psi\rangle_{AC} = \begin{cases} |\phi^+\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\phi^+\rangle)_{A,B,B',C} \\ |\psi^+\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\psi^+\rangle)_{A,B,B',C} \\ |\psi^-\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\psi^-\rangle)_{A,B,B',C} \\ |\phi^-\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\phi^-\rangle)_{A,B,B',C}. \end{cases} \quad (63)$$

In the following lemma, we use these techniques for computing the Werner parameter of the shared state after performing entanglement swapping operations between multiple Werner states.

**Lemma 3:** The Werner parameter  $W_n$  of the shared EPR-pair after performing  $n \geq 1$  successful entanglement swap operations between  $n + 1$  Werner states with parameter  $W$  is

$$W_n = W^n. \quad (64)$$

*Proof:* We prove this lemma by induction. We denote the set of  $n + 1$  Werner states

$$\{\rho_{X_0, X_1}, \rho_{X_1, X_2}, \dots, \rho_{X_n, X_{n+1}}\}$$

where each of the states are of the form  $W|\phi^+\rangle\langle\phi^+| + \frac{1-W}{4}\mathbb{I}_4$ . Here, the state  $\rho_{X_i, X_{i+1}}$  (or  $\rho_{X'_i, X_{i+1}}$ , for  $1 \leq i \leq n$ ) is shared between two quantum memories  $X_i$  ( $X'_i$ , for  $1 \leq i \leq n$ ) and  $X_{i+1}$  for  $0 \leq i \leq n$ . We consider that they are swapped in sequence. First  $\rho_{X_0, X_1}$  is swapped with  $\rho_{X_1, X_2}$ . If the entanglement swap is successful it results in a state that we denote  $\rho_{X_0, X_2}$ . Then, the state  $\rho_{X_0, X_2}$  is swapped with  $\rho_{X_2, X_3}$ , etc.

For the base case  $n = 1$ , we perform the entanglement swapping operation between two EPR-pairs that are stored in the memories  $X_1$  and  $X'_1$ . The joint state  $\rho_{X_0, X_1} \otimes \rho_{X'_1, X_2}$  is of the form

$$\rho_{X_0, X_1} \otimes \rho_{X'_1, X_2} = \left( W|\phi^+\rangle\langle\phi^+| + \frac{1-W}{4}\mathbb{I}_4 \right)_{X_0, X_1, X'_1, X_2}^{\otimes 2} \quad (65)$$

$$= \left( W^2|\phi^+\rangle\langle\phi^+| \otimes |\phi^+\rangle\langle\phi^+| + Y \right)_{X_0, X_1, X'_1, X_2} \quad (66)$$

where

$$Y_{X_0, X_1, X'_1, X_2} = \left( W \frac{1-W}{4} (|\phi^+\rangle\langle\phi^+| \otimes \mathbb{I}_4 + \mathbb{I}_4 \otimes |\phi^+\rangle\langle\phi^+|) \right)_{X_0, X_1, X'_1, X_2} + \left( \frac{1-W}{4} \right)^2 \mathbb{I}_{16}. \quad (67)$$

Note that, from (63), we get that after performing the entanglement swapping operation between two EPR-pairs the resulting state  $|\Psi\rangle_{X_0, X_2}$  looks like

$$|\Psi\rangle_{X_0, X_2} = \begin{cases} |\phi^+\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\phi^+\rangle)_{X_0, X_1, X'_1, X_2} \\ |\psi^+\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\psi^+\rangle)_{X_0, X_1, X'_1, X_2} \\ |\psi^-\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\psi^-\rangle)_{X_0, X_1, X'_1, X_2} \\ |\phi^-\rangle & \text{Input state: } (|\phi^+\rangle \otimes |\phi^-\rangle)_{X_0, X_1, X'_1, X_2}. \end{cases} \quad (68)$$

Note that, we can write

$$\mathbb{I}_4 = \frac{1}{4} (|\psi^+\rangle\langle\psi^+| + |\phi^+\rangle\langle\phi^+| + |\phi^-\rangle\langle\phi^-| + |\psi^-\rangle\langle\psi^-|).$$

If the input state (unnormalized) is  $Y_{X_0, X_1, X'_1, X_2}$ , then using the relation in (68), we get the resulting state (unnormalized) after a successful entanglement swapping operation between the qubits in memories  $X_1$  and  $X'_1$  as

$$Y_{X_0, X_2} = 2W \frac{1-W}{4} \mathbb{I}_4 + 4 \left( \frac{1-W}{4} \right)^2 \mathbb{I}_4 = \frac{1-W^2}{4} \mathbb{I}_4. \quad (69)$$

From the first case of (68), we get that if  $X_1$  performs a successful entanglement swapping operation between two states of the form  $|\phi^+\rangle_{X_0, X_1} |\phi^+\rangle_{X'_1, X_2} \langle\phi^+|$  then the resulting state is of the form  $|\phi^+\rangle_{X_0, X_2} \langle\phi^+|$ . Hence, after the successful entanglement swapping operation the shared state between  $X_0, X_2$  is of the form

$$\rho_{X_0, X_2} = W^2 |\phi^+\rangle\langle\phi^+| + Y_{X_0, X_2}. \quad (70)$$

By substituting the expression of  $Y_{X_0, X_2}$  in the above-mentioned expression from (69), we get the following form of  $\rho_{X_0, X_2}$ :

$$\rho_{X_0, X_2} = W^2 |\phi^+\rangle\langle\phi^+| + \frac{1-W^2}{4} \mathbb{I}_4. \quad (71)$$

Hence, the Werner parameter of the state  $\rho_{X_0, X_2}$  is  $W^2$ . This proves the base case of the lemma.

For the induction step, we assume the lemma is true for any  $n = n'$ . This implies

$$\rho_{X_0, X_{n'}} = W^{n'} |\phi^+\rangle\langle\phi^+| + \frac{1-W^{n'}}{4} \mathbb{I}_4. \quad (72)$$

For the case of  $n = n' + 1$ , an entanglement swapping operation is being performed between the qubits stored in the memories  $X_{n'}$  and  $X'_{n'}$ . The joint state can be written as

$$\rho_{X_0, X_{n'}} \otimes \rho_{X'_{n'}, X_{n'+1}} = \left( W^{n'} |\phi^+\rangle\langle\phi^+| + \frac{1-W^{n'}}{4} \mathbb{I}_4 \right)_{X_0, X_{n'}} \otimes \left( W |\phi^+\rangle\langle\phi^+| + \frac{1-W}{4} \mathbb{I}_4 \right)_{X'_{n'}, X_{n'+1}}$$

$$= \left( W^{n'+1} |\phi^+\rangle \langle \phi^+| \otimes |\phi^+\rangle \langle \phi^+| \right)_{X_0, X_{n'}, X'_{n'}, X_{n'+1}} + Y_{X_0, X_{n'}, X'_{n'}, X_{n'+1}} \quad (73)$$

where

$$Y_{X_0, X_{n'}, X'_{n'}, X_{n'+1}} = \left( W \frac{1 - W^{n'}}{4} \mathbb{I}_4 \otimes |\phi^+\rangle \langle \phi^+| \right)_{X_0, X_{n'}, X'_{n'}, X_{n'+1}} + \left( W^{n'} \frac{1 - W}{4} |\phi^+\rangle \langle \phi^+| \otimes \mathbb{I}_4 \right)_{X_0, X_{n'}, X'_{n'}, X_{n'+1}} + \frac{(1 - W)(1 - W^{n'})}{16} \mathbb{I}_{16}.$$

Using the relation in (63) and substituting  $\mathbb{I}_4 = \frac{1}{4}(|\psi^+\rangle \langle \psi^+| + |\phi^+\rangle \langle \phi^+| + |\phi^-\rangle \langle \phi^-| + |\psi^-\rangle \langle \psi^-|)$  for the  $Y_{X_0, X_{n'}, X'_{n'}, X_{n'+1}}$  part of the shared state, we get the resulting state (unnormalized) after a successful entanglement swapping operation as

$$Y_{X_0, X_{n'+1}} = \left( W^{n'} \frac{1 - W}{4} + W \frac{1 - W^{n'}}{4} \right) \mathbb{I}_4 + 4 \left( \frac{1 - W^{n'}}{4} \right) \left( \frac{1 - W}{4} \right) \mathbb{I}_4 = \frac{1 - W^{n'+1}}{4} \mathbb{I}_4. \quad (74)$$

Hence, the resulting state after  $X_{n'}$  performs the entanglement swapping operation is given by

$$\rho_{X_0, X_{n'+1}} = W^{n'+1} |\phi^+\rangle \langle \phi^+| + \frac{1 - W^{n'+1}}{4} \mathbb{I}_4. \quad (75)$$

Hence, the Werner parameter of the state  $\rho_{X_0, X_{n'+1}}$  is  $W^{n'+1}$ . This proves the induction step. Hence, by the principle of mathematical induction, we conclude that the statement Werner parameter  $W_n$  after the  $n$  successful swap operation is  $W^n$  is true for any  $n \geq 1$ . ■

## APPENDIX G: PREPARE AND SWAP PROTOCOL

In this appendix, we prove the EPR-pair generation rate across a repeater chain using prepare and swap protocol.

*Lemma 4:* In a repeater chain network with  $n + 1$  repeaters  $\{u_0, u_1, \dots, u_n\}$ , if the probability of generating an elementary pair per attempt is one and the probability of a successful BSM is  $(q)$  and the capacity of an elementary link  $(u_i, u_{i+1})$  (for  $0 \leq i \leq n - 1$ ) is denoted by  $C_i$  and if the repeaters follow the prepare and swap protocol for generating EPR-pairs, then the expected end-to-end entanglement generation rate  $r_{u_0, u_n}$  is

$$r_{u_0, u_n} = (q)^{n-1} \min\{C_0, \dots, C_{n-1}\}. \quad (76)$$

*Proof:* In the entanglement generation protocol, first, the repeaters start generating the elementary pairs in parallel. As a elementary pair generation is a deterministic event, so each of the node  $u_i$  can generate  $C_i$  EPR-pairs with its neighbor  $u_{i+1}$  ( $0 \leq i \leq n - 1$ ) per second. After generating the elementary pairs, the intermediate nodes perform the swap operations independently of each other. This implies, if all the swap operations are deterministic then the end-to-end entanglement generation is  $\min\{C_1, \dots, C_n\}$ . However, each of the swap operations succeeds with probability  $(q)$ . The end-to-end entanglement generation probability is equal to the probability that all elementary links are successfully swapped, which is  $(q)^n$  and there are  $\min\{C_1, \dots, C_n\}$  such elementary links. This implies, the expected end-to-end entanglement generation rate is  $r_{u_0, u_n} = (q)^{n-1} \min\{C_0, \dots, C_{n-1}\}$ . ■

## ACKNOWLEDGMENT

The authors would like to thank W. Kozłowski for many stimulating discussions and M. Skrzypczyk for giving useful feedback on the draft.

## REFERENCES

- [1] R. Van Meter, *Quantum Networking*. Hoboken, NJ, USA: Wiley, 2014, doi: [10.1002/9781118648919](https://doi.org/10.1002/9781118648919).
- [2] S. Lloyd, J. H. Shapiro, F. N. C. Wong, P. Kumar, S. M. Shahriar, and H. P. Yuen, "Infrastructure for the quantum internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 5, pp. 9–20, 2004, doi: [10.1145/1039111.1039118](https://doi.org/10.1145/1039111.1039118).
- [3] H. J. Kimble, "The quantum Internet," *Nature*, vol. 453, no. 7198, 2008, Art. no. 1023, doi: [10.1038/nature07127](https://doi.org/10.1038/nature07127).
- [4] S. Wehner, D. Elkouss, and R. Hanson, "Quantum Internet: A vision for the road ahead," *Science*, vol. 362, 2018, Paper eaam9288, doi: [10.1126/science.aam9288](https://doi.org/10.1126/science.aam9288).
- [5] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, no. P1, pp. 7–11, 2014, doi: [10.1016/j.tcs.2014.05.025](https://doi.org/10.1016/j.tcs.2014.05.025).
- [6] A. K. Ekert, "Quantum cryptography based on Bell theorem," *Phys. Rev. Lett.*, vol. 67, no. 6, p. 661, 1991, doi: [10.1103/PhysRevLett.67.661](https://doi.org/10.1103/PhysRevLett.67.661).
- [7] P. Komar et al., "A quantum network of clocks," *Nat. Phys.*, vol. 10, no. 8, pp. 582–587, 2014, doi: [10.1038/nphys3000](https://doi.org/10.1038/nphys3000).
- [8] D. Coppersmith, D. Gamarnik, and M. Sviridenko, "The diameter of a long range percolation graph," in *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2002, pp. 329–337, doi: [10.1007/978-3-0348-8211-8-9](https://doi.org/10.1007/978-3-0348-8211-8-9).
- [9] B. C. Britt, "Modeling viral diffusion using quantum computational network simulation," *Quantum Eng.*, vol. 2, no. 1, p. e29, 2020, doi: [10.1002/que2.29](https://doi.org/10.1002/que2.29).
- [10] R. Van Meter, T. Satoh, T. D Ladd, W. J. Munro, and K. Nemoto, "Path selection for quantum repeater networks," *Netw. Sci.*, vol. 3, no. 1–4, pp. 82–95, 2013, doi: [10.1007/s13119-013-0026-2](https://doi.org/10.1007/s13119-013-0026-2).
- [11] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22299–22312, 2017, doi: [10.1109/ACCESS.2017.2763325](https://doi.org/10.1109/ACCESS.2017.2763325).
- [12] Mihir Pant et al., "Routing entanglement in the quantum Internet," *NPJ Quantum Inf.*, vol. 5, p. 25, 2019, doi: [10.1038/s41534-019-0139-x](https://doi.org/10.1038/s41534-019-0139-x).
- [13] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum Internet," 2019, *arXiv:1907.11630*.
- [14] Fengyou Sun, "Performance analysis of quantum channels," *Quantum Eng.*, vol. 2, no. 2, p. e35, 2020, doi: [10.1002/que2.35](https://doi.org/10.1002/que2.35).
- [15] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, "Inside quantum repeaters," *IEEE J. Sel. Top. Quantum Electron.*, vol. 21, no. 3, pp. 78–90, May/Jun. 2015, doi: [10.1109/JSTQE.2015.2392076](https://doi.org/10.1109/JSTQE.2015.2392076).
- [16] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Phys. Rev. Lett.*, vol. 70, no. 13, p. 1895, 1993, doi: [10.1103/PhysRevLett.70.1895](https://doi.org/10.1103/PhysRevLett.70.1895).



- [17] M. Zukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, “Event-ready-detectors” Bell experiment via entanglement swapping,” *Phys. Rev. Lett.*, vol. 71, pp. 4287–4290, 1993, doi: [10.1103/PhysRevLett.71.4287](https://doi.org/10.1103/PhysRevLett.71.4287).
- [18] A. M. Goebel *et al.*, “Multistage entanglement swapping,” *Phys. Rev. Lett.*, vol. 101, no. 8, 2008, Art. no. 080403, doi: [10.1103/PhysRevLett.101.080403](https://doi.org/10.1103/PhysRevLett.101.080403).
- [19] T. C. Hu, “Multi-commodity network flows,” *Oper. Res.*, vol. 11, no. 3, pp. 344–360, 1963, doi: [10.1287/opre.11.3.344](https://doi.org/10.1287/opre.11.3.344).
- [20] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proc. 16th Annu. ACM Symp. Theory Comput.*, 1984, pp. 302–311, doi: [10.1145/800057.808695](https://doi.org/10.1145/800057.808695).
- [21] N. Sangouard, C. Simon, H. De Riedmatten, and N. Gisin, “Quantum repeaters based on atomic ensembles and linear optics,” *Rev. Modern Phys.*, vol. 83, no. 1, p. 33, 2011, doi: [10.1103/RevModPhys.83.33](https://doi.org/10.1103/RevModPhys.83.33).
- [22] M. Gündoğan, M. Mazzer, P. M. Ledingham, M. Cristiani, and H. de Riedmatten, “Coherent storage of temporally multimode light using a spin-wave atomic frequency comb memory,” *New J. Phys.*, vol. 15, no. 4, 2013, Art. no. 045012, doi: [10.1088/1367-2630/15/4/045012](https://doi.org/10.1088/1367-2630/15/4/045012).
- [23] Neil Sinclair *et al.*, “Spectral multiplexing for scalable quantum photonics using an atomic frequency comb quantum memory and feed-forward control,” *Phys. Rev. Lett.*, vol. 113, no. 5, 2014, Art. no. 053603, doi: [10.1103/PhysRevLett.113.053603](https://doi.org/10.1103/PhysRevLett.113.053603).
- [24] A. Ridha Mahjoub and S. Thomas McCormick, “Max flow and min cut with bounded-length paths: Complexity, algorithms, and approximation,” *Math. Program.*, vol. 124, no. 1/2, pp. 271–284, 2010, doi: [10.1007/s10107-010-0366-6](https://doi.org/10.1007/s10107-010-0366-6).
- [25] S. Bäuml, K. Azuma, G. Kato, and D. Elkouss, “Linear programs for entanglement and key distribution in the quantum Internet,” *Commun. Phys.*, vol. 3, no. 1, pp. 1–12, 2020, doi: [10.1038/s42005-020-0318-2](https://doi.org/10.1038/s42005-020-0318-2).
- [26] S. Pirandola, “Capacities of repeater-assisted quantum communications,” 2016, *arXiv:1601.00966*.
- [27] S. Pirandola, “End-to-end capacities of a quantum communication network,” *Commun. Phys.*, vol. 2, no. 1, pp. 1–10, 2019, doi: [10.1038/s42005-019-0147-3](https://doi.org/10.1038/s42005-019-0147-3).
- [28] S. Pirandola, “Bounds for multi-end communication over quantum networks,” *Quantum Sci. Technol.*, vol. 4, no. 4, 2019, Art. no. 045006, doi: [10.1088/2058-9565/ab3f66](https://doi.org/10.1088/2058-9565/ab3f66).
- [29] K. Azuma, A. Mizutani, and H.-K. Lo, “Fundamental rate-loss trade-off for the quantum internet,” *Nat. Commun.*, vol. 7, no. 1, pp. 1–8, 2016, doi: [10.1038/ncomms13523](https://doi.org/10.1038/ncomms13523).
- [30] K. Azuma and G. Kato, “Aggregating quantum repeaters for the quantum Internet,” *Phys. Rev. A*, vol. 96, no. 3, 2017, Art. no. 032332, doi: [10.1103/PhysRevA.96.032332](https://doi.org/10.1103/PhysRevA.96.032332).
- [31] L. Rigovacca, G. Kato, S. Bäuml, M. S. Kim, W. J. Munro, and K. Azuma, “Versatile relative entropy bounds for quantum networks,” *New J. Phys.*, vol. 20, no. 1, 2018, Art. no. 013033, doi: [10.1088/1367-2630/aa9fcf](https://doi.org/10.1088/1367-2630/aa9fcf).
- [32] S. Bäuml and K. Azuma, “Fundamental limitation on quantum broadcast networks,” *Quantum Sci. Technol.*, vol. 2, no. 2, 2017, Art. no. 024004, doi: [10.1088/2058-9565/aa6d3c](https://doi.org/10.1088/2058-9565/aa6d3c).
- [33] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, “Effective routing design for remote entanglement generation on quantum networks,” 2020, arXiv:2001.02204.
- [34] R. F. Werner, “Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model,” *Phys. Rev. A*, vol. 40, no. 8, p. 4277, 1989, doi: [10.1103/PhysRevA.40.4277](https://doi.org/10.1103/PhysRevA.40.4277).
- [35] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, “Quantum repeaters: The role of imperfect local operations in quantum communication,” *Phys. Rev. Lett.*, vol. 81, no. 26, p. 5932, 1998, doi: [10.1103/PhysRevLett.81.5932](https://doi.org/10.1103/PhysRevLett.81.5932).
- [36] M. Stuart and C.-M. Duquesne, “A python linear programming API,” PYGLPK Roy, J.S. and Franco Peschiera YAPOSIB bindings, 2020. [Online]. Available: <http://coin-or.github.io/pulp/>
- [37] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughtan, “The Internet topology zoo,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011, doi: [10.1109/JSAC.2011.111002](https://doi.org/10.1109/JSAC.2011.111002).
- [38] K. Chakraborty, “Routing-via-multi-commodity-flow,” 2020. [Online]. Available: <https://github.com/kaushikchakraborty9/Routing-via-Multi-Commodity-Flow/>
- [39] K. Chakraborty, “Surfnet graph, data for exploring the problem routing in a quantum Internet,” 2020. [Online]. Available: <https://doi.org/10.4121/uuid:4a0afe1d-5d96-4a90-abc8-4b4e61967ba3>
- [40] A. Hagberg, P. Swart, and D. S. Chult, “Exploring network structure, dynamics, and function using networkx,” Tech. Rep. LA-UR-08-05495 LA-UR-08-5495, Los Alamos National Lab., Los Alamos, NM, USA, 2008.
- [41] W. Dai, T. Peng, and M. Z. Win, “Optimal remote entanglement distribution,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 3, pp. 540–556, Mar. 2020, doi: [10.1109/JSAC.2020.2969005](https://doi.org/10.1109/JSAC.2020.2969005).