Received May 24, 2021; accepted June 14, 2021; date of publication June 21, 2021;
date of current version July 15, 2021.

Digital Object Identifier 10.1109/TQE.2021.3090532

Request Scheduling in Quantum Networks

CLAUDIO CICCONETTI, MARCO CONTI, AND ANDREA PASSARELLA

Istituto di Informatica e Telematica-CNR, 56124 Pisa, Italy

Corresponding author: Claudio Cicconetti (c.cicconetti@iit.cnr.it).

ABSTRACT Quantum networking is emerging as a new research area to explore the opportunities of interconnecting quantum systems through end-to-end entanglement of qubits at geographical distance via quantum repeaters. A promising architecture has been proposed in the literature that decouples entanglement between adjacent quantum nodes/repeaters from establishing end-to-end paths by adopting a time slotted approach. Within this model, we destructure further end-to-end path establishment into two subproblems: path selection and scheduling. The former is set to determine the best repeaters to connect two end nodes, provided that all their local entanglements have succeeded. On the other hand, scheduling is concerned with deciding, which pairs of end nodes are served in the current time slot, while the others remain queued for later time slots. Unlike path selection, scheduling has not been investigated so far in the literature, particularly in presence of quantum noise, which makes both problems even more challenging. In this article, we propose to address it via a general framework of heuristic algorithms, for which we propose three illustrative instances with the objective of keeping the application delay small while achieving a good system utilization, in terms of high entanglement rate and fidelity of remotely entangled qubits. The system proposed is evaluated extensively via event-driven quantum network simulations, with noisy repeaters, in different node topologies under a Poisson arrival of requests from quantum applications. The results show the existence of a fundamental tradeoff between system- and application-level metrics, such as fairness versus entanglement and fidelity, which lays the foundations for further studies in this thriving research area.

INDEX TERMS End-to-end entanglement, noisy quantum repeaters, quantum Internet, routing.

I. INTRODUCTION

Quantum computing is pushing the frontiers of computation, and its advantages will be multiplied by the creation of a quantum Internet to interconnect remote quantum computers with one another. The quantum Internet will enable distributed quantum computation [8] and many other applications [36]. A roadmap for its realization is neatly summarized in [43], while a break-down of the technologies involved and their recent status is reported in [2]. The most important building block of quantum networks is the *quantum repeater* (or *repeater* for short),[1] which allows the transfer of quantum states through *entanglement swap* [5]. The case of end-to-end entanglement between two physically separated quantum systems (let us call them *nodes*) via a repeater using quantum sources is illustrated in Fig. 1 [35]. The quantum sources emit locally entangled Einstein–Podolsky–Rosen (EPR) pairs toward both of the end-point quantum systems, called Alice and Carol in the example, and the repeater, i.e., Bob. Once Bob has performed a projective measurement on his two particles ($|\Phi\rangle^0_{\mathrm{Bob}}$ and $|\Phi\rangle^1_{\mathrm{Bob}}$) and has communicated via a traditional communication channel, the outcome of the operation (m^B_0, m^B_1) to, for example, Carol, the latter may perform local Pauli operations in the form of X- and Z-gates to correct its qubit $|\Phi\rangle^0_{\mathrm{Carol}}$, which effectively creates an end-to-end entanglement with the qubit received by Alice $|\Phi\rangle^0_{\mathrm{Alice}}$.

Even though the quantum repeater is not yet available as a commercial device, several studies have proved experimentally the viability of its key components ([4], [38], [39], [45], [46]. Furthermore, by interconnecting repeaters with one another, it is possible to create an end-to-end entanglement between nodes separated by an arbitrary distance, at least in theory. In practice, there are technical limitations due to: 1) signal power attenuation along the transmission, over fiber optic cables, or ground-to-satellite air links [22] and 2) the short coherence times in quantum memories [7]. These limitations create many research challenges for the realization of quantum networks based on entanglement swap. Even though the advancements in the construction processes of the devices involved, as well as in the area of distillation [18], purification [40], and quantum error correction [16], will

[1] Alternative techniques to achieve entanglement distribution in a quantum network have been also proposed and are worthy of further investigation, e.g., *entanglement percolation* [30], but in the research community repeater-based approaches are currently predominant.

gradually reduce the current problems, we do not expect conclusive solutions to appear shortly.

Therefore, in this article, we pursue a line of research that is based on near-future technology and has already attracted some interest in the literature: we assume that the entanglement done locally at every repeater for every link toward an adjacent node can fail with a non negligible probability, but it is heralded by some technology means (beyond the scope of our work) such that the outcome is known to the repeater itself. We then assume that there is a network element that collects all these outcomes and, therefore, has a perfect knowledge of which links between quantum computers can be used to the transfer the status of qubits (the analogous in quantum Internet to transfer of information in legacy Internet). Thus, the controller tries to satisfy a set of demands from the upper layer applications by performing an *ephemeral* routing of multiple requests, which is only valid until the next round of local link entanglements.

Whether this will be a practical setup for a quantum network, on a local or wide-area scale, only time will tell. For now, we build on top of the prior works in this area, which are surveyed in Section II, and analyze one specific aspect that, to the best of our knowledge, has been overlooked in the studies so far: request scheduling. The latter refers to the choice of which end-to-end entanglement requests to satisfy at any given time, based on the latest outcome of local entanglements, provided that it is not possible to grant immediately all the demands pending. The system model is illustrated in Section III, while the problem of request scheduling is analyzed in Section IV, as part of the more general issue of quantum routing. In the same section, we also propose a general framework, called *iterative scheduling*, and three instances to showcase the ability to achieve different objectives, which are then evaluated in Section V. The results show that a fundamental tradeoff exists between the efficiency, in terms of the rate of entanglements and their fidelities, and fairness across applications, which lays the foundations for further research in this area, as discussed in the conclusions (see Section VI).

II. STATE OF THE ART

In this section, we review the state of the art on quantum routing, as introduced in Section I and fully elaborated in Section III and Section IV below. For completeness, we mention that in the literature there is a growing number of studies dealing with sparse aspects of quantum routing, which are, however, only marginally relevant to the specific problem that we address and, thus, are not analyzed in detail. They include, among the others: the definition of protocol stacks [12], [26], [32], the identification of capacity regions [42], studies on tools for performance evaluation of quantum networks [3], multipartite entanglement distribution [27], and the design of networked quantum applications [13].

A relevant area, instead, is that of *path selection*, i.e., deciding along which repeaters to establish end-to-end entanglement between two nodes among a set of possible alternatives in a quantum network. This problem is very much related to selecting a suitable routing metric (or cost), since those typically used in traditional networks do not keep into account quantum-network-specific characteristics, such as local entanglement potential failures and fidelity. This is the starting point in [41], where several possible metrics are defined and evaluated when used in combination with the traditional Dijkstra's shortest path first (SPF) algorithm [15]. Two key findings are that SPF achieves reasonable performance goals, and that the throughput of a single link, in number of Bell pairs generated in the unit of time, performs better than other physical-related metrics, such as the channel loss or the inverse of the channel transmittance. A more sophisticated routing metric has been defined in [6] and proved to be optimal for selecting the path between two nodes, when used in conjunction with a custom routing protocol with polynomial time complexity.

Key difference: All these works did not explore the opportunities and challenges of activating concurrently multiple end-to-end entanglements in the quantum network, which is supported by our system model.

Multipath routing is instead considered in [29], to which our system model inspires (see Section III). The authors propose both a global link state routing algorithm, where decisions are taken by a centralized authority, and a local link state protocols, where repeaters take local uncoordinated decisions. A similar approach has been followed in [28], where entanglement swapping is used as a means to "boost" the capacity that would otherwise be available with local links only. The authors have used a maximum flow problem formulation, which we also adopt to define an idealized quantum routing problem in Section IV-A.

Key difference: We extend the two models previously by considering also the aspect of request scheduling, as well as including decoherence in quantum memories in the performance evaluation, not accounted in those studies.

A different direction has been investigated in other works where the notion of *virtual quantum link* has been introduced. A virtual link is an entanglement between multiple nodes (the intermediate ones acting a repeaters), which can be combined together with other virtual links to create the final entanglement between the end nodes. This abstraction allows the quantum problem to be decoupled in two: first, the virtual links are established to create an arbitrary virtual topology, which is assumed to be regular; then, routing is done on the virtual topology to create the end-to-end entanglements as required. These problems have been investigated in [34], which also proposes a resource-efficient decentralized routing protocol in a ring and sphere topology, and in [9], with ring, grid and recursively generated topologies, i.e., topologies that adapt dynamically by substituting edges in the topology at the previous step. Furthermore, Chakraborty *et al.* [9] elaborated on two alternative models: continuous, where virtual links are created in the background even if no demand for end-to-end entanglement is known, and on demand, which reacts to the arrival of demands with no precreated virtual links available.

Key difference: Virtual link routing is not directly applicable to our system model, but in principle it is a candidate alternative to our proposed contribution, and we plan to compare them in a future work.

Finally, we mention the research area of stochastic routing in wireless sensor networks (WSNs), which has some similarities to quantum routing. In fact, a WSN is typically multihop, i.e., the nodes are expected to forward messages not intended for them, in addition to carrying out their ordinary sensing duties. However, energy is very often a scarce resource, therefore, the nodes alternate between activity and inactivity periods: when they do so in an uncontrolled manner, the routing process cannot rely on all potential links being active at the same time, which is similar to what happens in a quantum network due to a failed local entanglement. The problem has been studied in the literature, e.g., Ribeiro *et al.* [33] proposed a distributed algorithm that finds the optimal routing probabilities based on the physical-layer characteristics of the links, which is then generalized in [1], which also considers the sensing process in addition to multihop data transfer.

Key difference: There is a fundamental difference between stochastic routing in WSN and the quantum routing problem investigated in this article: as we will explain in more details in Section III, we assume that local link entanglement is heralded at each time slot, hence a (reduced) network topology is known with certainty at the time of taking routing decisions. We recognize that a possible further line of research is possible, where routing decisions are taken before knowing the local link entanglement outcomes, but we consider this beyond the scope of this article.

III. SYSTEM MODEL

In this section, we describe the system model used in this article. We first introduce the network architecture and its key elements (see Section III-A) and then describe the time-slotted nature of the approach (see Section III-B).

A. QUANTUM NETWORK ARCHITECTURE

The structure of a quantum network is illustrated in Fig. 2. It consists of the following:

1) *quantum computers:* systems that wish to establish end-to-end entanglement with other computers through the network, e.g., for distributed quantum computation;
2) *quantum repeaters:* devices that interconnect quantum computers and other quantum repeaters for the only purpose of making such entanglement possible beyond the maximum distance under state-of-the-art quantum communication technologies.

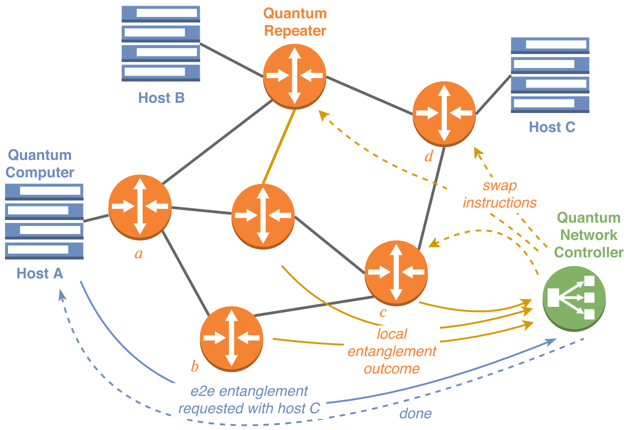
Quantum Repeater
Host B
Host C
Quantum Computer
Host A
a
b
c
d
swap instructions
Quantum Network Controller
local entanglement outcome
e2e entanglement requested with host C
done

FIG. 2. Quantum network architecture.

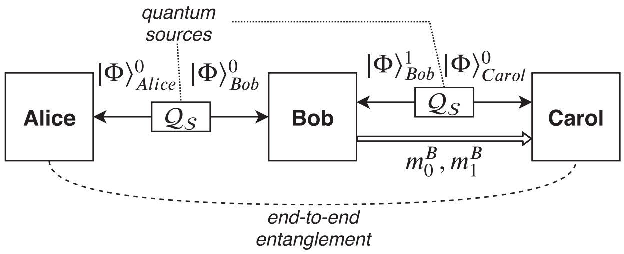

FIG. 1. End-to-end entanglement between Alice and Carol via a quantum repeater (Bob).

As introduced in Section I, end-to-end entanglement between two computers, e.g., Host A and Host C in the example, at a given time is only possible if there is a suitable path, e.g., $\sigma = \{a, b, c, d\}$, and the following:

1) local entanglement between all pairs of repeaters along the path (i.e., ab, bc, and cd) is successful;
2) the intermediate nodes (i.e., b and c) perform a swap operation (like Bob in Fig. 1) via a projective measurement;
3) the results of the measurement are transmitted, via classical communication, to one of the end nodes (either a or d) to perform corrections to the local qubit via an appropriate sequence of X- and Z-gate quantum memory instructions.

Without loss of generality, hereafter we do not distinguish anymore between quantum computers and quantum repeaters and use the generic term *nodes* for both. Two nodes are *neighbors* if the state of a qubit (called *flying qubit*) can be transferred between the two directly. Typically this means that the two nodes are connected by a fiber optic cable to carry quantum state embedded in light pulses. The physical connection between two neighbors is called *link*; it is realistic to assume that the link may also transport classical information, though the two quantum versus classical channels may have different characteristics.

In our model a so-called *quantum network controller*—term borrowed from the context of software-defined networking [23]—is a classical computing device, without any quantum links with the nodes, in charge of the following.

1) Collecting the requests for an end-to-end entanglement from the quantum computers in the network, and notifying them once this has been done.
2) Receiving the local link entanglement outcome (success versus failure) from the repeaters in the network.
3) Choosing which requests to serve (we call this operation *scheduling*, as defined in Section IV) and, for each of them, selecting the best path (*path selection*—Section IV).
4) For each end-to-end entanglement activated on a given path, notifying the intermediate repeaters to perform an entanglement swap operation involving the given links, and also specifying where (= to which end node) to transfer the result of the projective measurement.

As it will be clear in the following section, the steps above occur continuously: the quantum controller has to take real-time decisions in step 3, since the deadline for using the end-to-end entanglement path is the beginning of the next *slot*. Moreover, we note that a controller-based architecture has been already proposed in the context of quantum networks in [14], where the authors study the problem of efficient distribution of monolithic quantum algorithms, in particular for an accelerated variational quantum eigensolver algorithm over arbitrary sized distributed quantum computers.

B. FRAMING STRUCTURE

As in [29], we assume that the time is slotted and all the nodes are synchronized on a common time reference. We note that ideal synchronization is *not* a requirement, since all the repeaters use quantum memories that can compensate small time offsets due to imperfect synchronization, as well as uneven propagation delays across different links. However, we do assume that all the nodes and the controller have the notion of time slots and agree on their boundaries; therefore, the duration of the time slot is a system parameter that in practice will have to be chosen depending on the network geography, hardware characteristics, and application requirements; we consider this provisioning an implementation issue, and do not address it in this article.

The time slot structure is illustrated in Fig. 3. As can be seen in the bottom part, there are two phases in each time slot.

1) In the first phase (gray area in the diagram) end-to-end entanglement is prepared through a sequence of subphases, which are described below. This phase is most relevant to our work.
2) In the second phase (from the end of the gray area until the next time slot boundary), the applications running on the hosts use the end-to-end entangled qubits that have been set up in the first phase for their purposes. This is not discussed further in this article. The request p_i for the activation of an end-to-end path between a pair of qubits is asynchronous with respect to the time boundaries, which do not need to be exposed at all to the quantum applications on the hosts. On the other hand, the end-to-end paths $\tilde{p}_i$ are always granted by the controller at the end of the first phase. It can happen that a request is not fulfilled in the time slot immediately preceding that in which it arrived: for instance, in the example in Fig. 3, p_0 and p_1 arrive in the same time slot, and $\tilde{p}_1$ is granted in the next slot but $\tilde{p}_0$ is granted after two time slots. The reasons for this behavior will be clear in the following.

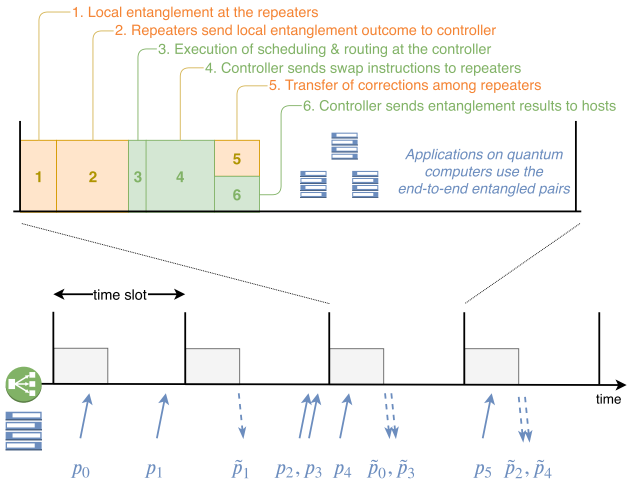
1. Local entanglement at the repeaters
2. Repeaters send local entanglement outcome to controller
3. Execution of scheduling & routing at the controller
4. Controller sends swap instructions to repeaters
5. Transfer of corrections among repeaters
6. Controller sends entanglement results to hosts
1
2
3
4
5
6
Applications on quantum computers use the end-to-end entangled pairs
time slot
time
p_0
p_1
$\tilde{p}_1$
p_2, p_3
p_4
$\tilde{p}_0, \tilde{p}_3$
p_5
$\tilde{p}_2, \tilde{p}_4$

FIG. 3. **Time slot structure.**

We now describe all the subphases that compose the first phase (gray area in Fig. 3, where we zoom in the time slot with a qualitative diagram in the top part of the figure).

1) Local entanglement is performed on every link in the network at the beginning of every time slot. Hereafter, we assume without loss of generality that this operation happens through the use of quantum sources located in between nodes (like in Fig. 1), but the actual implementation can be different without affecting the nature of our contribution. Under this working assumption, as the time slot begins, all synchronized quantum sources emit a pair of entangled flying qubits, which are absorbed by the nodes and identified as successful or failed.
2) The local entanglement outcomes are communicated to the controller via the classical network. The time needed for this operation depends on the geographical size of the network, the position of the controller, as well as the classical communication technologies deployed.
3) The controller performs *routing*, that is it decides, which qubits will have to be measured by the nodes in the network to create end-to-end entanglement paths based on the current demands. Note that the controller

TABLE I **System Model Assumptions**

Assumption	Reason
Nodes are synchronized	We assume that the time is divided into slots and the repeaters and the controller have to be aware of the boundaries to respect alternating the two main phases as illustrated in Fig. 3. Furthermore, for simplicity of illustration, we assume that quantum sources are used for local link entanglement between adjacent nodes, in which case the emission rate of the sources is $1/T$, where T is the time slot duration.
Stable topology	We assume that the logical topology of nodes, i.e., the graph $G(V, E)$ of repeaters and their interconnection links, does not vary at a short time scale, comparable with that of the time slot. Under this assumption it is reasonable to assume that the controller has the full view of the network, which is needed to take routing decisions as described in Sec. IV.
Heralded local entanglement	A key assumption in our study is that a repeater knows the outcome, success vs. failure, of the entanglement. This is required so that the controller can become aware of the reduced subgraph, as defined in Sec. IV, to be used for the end-to-end entanglement paths in the current time slot.

is the only element with a global view of the quantum network as both a logical topology (i.e., the graph (V, E)) and a physical topology (i.e., the length of the physical links interconnecting the nodes). Routing is performed through the execution of classical algorithms and it is the subject of the entire Section IV. The output of this operation is a set of end-to-end paths $\{\sigma_i\}$.

4) Based on that, the controller notifies to each intermediate node in a path (a node is *intermediate* in a path if it is not the first or last element, which are referred to as *end* nodes instead) the list of pairs of quantum memory locations on which to execute entanglement swap to create an end-to-end entanglement as required.
5) Each intermediate node then forwards to one of the end nodes in the path the result (classical information) of the projective measurement used to obtain an entanglement swap. We assume that the result is forwarded directly by the intermediate nodes without involving the controller, but this is merely a protocol implementation detail.
6) After an end node has performed the X-/Z-gate corrections on its local qubit as indicated by the results forwarded by all the intermediate nodes along a given path, the pair of qubits is ready to be used by the applications on the respective remote hosts, until the end of the time slot.

In our model, we assume that a locally entangled pair of qubits is not reused across multiple time slots, i.e., it is discarded if it is not used as part of an end-to-end entanglement path in the current time slot. This is motivated by the current state of technology, which yields a relatively low qubit generation rate and a much faster decoherence rate. However, the solution proposed in this article could be extended to also consider the case of reusing local entanglements in future time slots. Furthermore, we assume for simplicity that all requests are equal, but we can easily imagine several directions of further investigation when the requests differ by their relative priority (as in DiffServ [37]) or by quantum network specific parameters (e.g., minimum number of paths established in the same time slot or minimum fidelity threshold). We will consider the case of unequal entanglement requests, as proposed in [19], in our future work.

We summarize the system model assumptions in Table I.

IV. QUANTUM ROUTING

In this section, we address the quantum routing problem, defined in accordance with the system model described in the aforementioned section. First, we define the problem in a formal manner (see Section IV-A), and then we propose to separate it into two subproblems called *scheduling* and *path selection*, which are tackled individually in Section IV-B and Section IV-C, respectively.

A. ROUTING PROBLEM

According to our system model defined in Section III, routing is addressed by the quantum network controller, with traditional computing resources. The input of the routing problem is as follows.

1) The logical topology of the network as a graph (V, E) and the physical distances between any two nodes with an edge in (V, E).
2) The set of local entanglements S that have succeeded in the first phase of this time slot. End-to-end entanglement paths will have to be established in a *reduced subgraph*$(V,' E')$, defined as follows: E' contains all the quantum network links that are successful, i.e., where the local entanglements in the current time slot have been detected as successful by both repeaters.
3) The list of pairs of nodes that wish to establish an end-to-end entanglement path: this includes both the pairs that have arrived in previous time slots but have not been served due to lack of resources, and the new pairs that have arrived in this time slot.

The *output* of the routing problem is a set of end-to-end entanglement paths; for each path, the following information is required.

1) The list of *swap nodes* and, for each swap node, the positions of the memory slots to be measured.

2) The respective positions of the memory slots on each of the two end nodes.
3) A path identifier, to distinguish different paths coexisting in the same time slot.
4) An identifier of the end node to which to send all the corrections.

All the above-mentioned need to be encoded in messages exchanged via traditional communication channels i) for the swap nodes to trigger measurements, ii) for intermediate nodes to forward the messages with correction bits toward the intended end node for a given path, iii) for the end node to perform the corrections to its local qubit, and iv) for both end nodes to make the local qubit available to upper layers. Even though, we have defined a simple protocol for the purpose of running simulation experiments, whose results will be presented in Section V, we do not delve into this aspect, which is an open research area (see, e.g., [10], [12], and [32]). In the following, we assume that such a protocol exists and it is efficient enough not to introduce a noticeable overhead, in terms of latency, except that of the propagation of messages over the physical links.

In general, a given input may produce several possible outputs; thus, the role of the controller is to select the *best* possible outcome. However, decisions in a given time slot have consequences also in future time slots: pairs that are not assigned a path as part of the current routing algorithm execution remain pending, and will have to be assigned a path later or be dropped, eventually. Therefore, at least in principle, when taking its decisions in a given time slot t, the controller should consider what happened in the previous slots (= the requested pairs that have not been served) and what will happen in the future slots (= the opportunities for better serving the pending requests in the subsequent slots and the further requests that will arrive). We formulate mathematically an *idealized routing problem* as a maximum multicommodity flow problem spanning over a time horizon from $t = 0$ to $t = T$

$$\max \sum_{i=1}^{P} \sum_{t=1}^{T} \sum_{w \in V'(t)} f_{i,t}(s_i, w) \tag{1}$$

s.t.

$$\forall t \in \mathcal{T}\, \forall u, v \in E'(t) : \sum_{i=1}^{P} f_{i,t}(u, v) \leq 1 \tag{2}$$

$$\forall i \in \mathcal{P} : \sum_{t=0}^{a_i} f_{u,t}(u, v) = 0 \tag{3}$$

$$\forall i \in \mathcal{P} : \sum_{t=a_i+1}^{T} f_{i,t}(u, v) \leq 1 \tag{4}$$

$\forall i \in \mathcal{P}\, \forall t \in \mathcal{T}$:

$$\sum_{w \in V'(t)} f_{i,t}(u, w) - \sum_{w \in V'(t)} f_{i,t}(w, u) = 0, u \neq s_i, d_i \tag{5}$$

$$\sum_{w \in V'(t)} f_{i,t}(s_i, w) - \sum_{w \in V'(t)} f_{i,t}(w, s_i) \leq 1 \tag{6}$$

$$\sum_{w \in V'(t)} f_{i,t}(w, d_i) - \sum_{w \in V'(t)} f_{i,t}(d_i, w) \leq 1 \tag{7}$$

where

1) the requests (commodities) are the end-to-end entanglement pairs requested by the quantum applications identified as $p_i = (s_i, d_i, a_i)$, with $s_i \in V$ the source node, $d_i \in V$ the destination node, and $a_i \in \{0, \ldots, T\}$ the time slot in which the request arrives;
2) $f_{i,t}(u, v)$ is a binary variable that is equal to 1 iff the end-to-end entanglement path of pair i is routed through the edge from node u to node v in the reduced subgraph $(V'(t), E'(t))$;
3) $\mathcal{T}$ is the set of all the time slots $\{0, \ldots, T\}$, and $\mathcal{P}$ is the set of all the requests $\{0, \ldots, P\}$.

The problem above-mentioned is to maximize the total number of flows established in all time slots for all request pairs in (1) because, for each pair p_i, $\sum_{t=1}^{T} \sum_{w \in V'(t)} f_{i,t}(s_i, w)$ will be 1 iff the pair has been assigned a path in any time slot, and 0 otherwise. This corresponds to maximizing the entanglement rate. With regard to the set of constraints: (2) guarantees that the same link is not used by more than one path in the same time slot; (3) means that a pair cannot be served before it arrives; (4) means that a pair can be served at most once; (5) guarantees that for every path associated to a pair, the number of incoming flows is equal to the number of outgoing flows, unless the node is the source of the pair, (6) or the destination (7), i.e., they are the flow conservation constraints. The major difference with respect to a classical linear programming flow problem is that the graph may change at every time slot due to failures of local entanglement, as illustrated in a visual manner in the example in Fig. 4 for three time slots. We note that the objective function (1) may be replaced with another that fits better the specific use case of interest, but this does not affect the core of our contribution.

The problem (1)–(7) is merely *ideal* for two reasons as follows:

1) in practice, the time horizon, and hence, the size of the problem, is infinite ($T \to \infty$);
2) finding the optimal solution of the problem assumes that the times of arrivals of application requests are available and the evolution of the reduced subgraphs over the entire time horizon is known exactly.

In practice, the controller takes decision time slot by time slot, i.e., the time fixed, as well as the reduced subgraph and the list of currently pending requests. Even in this form,

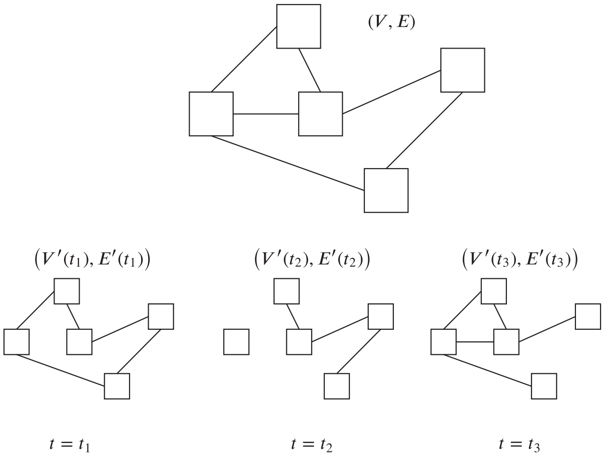

FIG. 4. Example of three reduced subgraphs at different time slots from the same graph.

the problem (1)–(7) is NP-complete [17], thus, it is difficult to solve it exactly even for small instances, and impossible for large ones. We note that solving the problem *as soon as possible* is of paramount importance, because during the time spent by the controller to take its decisions the qubits experience decoherence in the quantum memories of the repeaters, thus decreasing the fidelity of end-to-end entanglements, resulting in the qubit state becoming useless to the application layers. While in principle we could use approximation techniques to reduce the computation time (e.g., [24]), we propose instead to exploit the nature of the problem and split it into two subproblems, called scheduling, and path selection.

1) *Scheduling* is the problem of selecting, which pairs are assigned an end-to-end entanglement path in the current time slot (conversely, which pairs remain unserved in the current time slot and will be either assigned a path in the future or dropped, eventually).
2) *Path selection* is the problem of finding the best path for a single pair of end nodes wishing to establish an end-to-end entanglement path in the reduced subgraph (V, E') in the current time slot.

Decoupling routing in two subproblems entails a clean design and a more efficient implementation, allowing each problem to be solved with its most suitable tool. Such an approach is very common in state-of-the-art wireless resource scheduling (e.g., [25]), which has some similarities to the quantum routing problem since it involves a constrained allocation of resources under fast changing and largely unpredictable time-varying conditions. Furthermore, since path selection has been already investigated in research (as outlined below in Section IV-C), we can make use of the results obtained so far.

However, in quantum routing like in wireless resource scheduling, such a decoupling is merely artificial, as explained below for quantum routing with the help of two simple examples. Let us consider first the situation in Fig. 5,

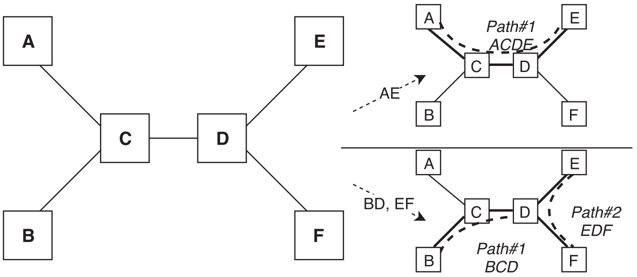

FIG. 5. Example: how scheduling affects path selection.

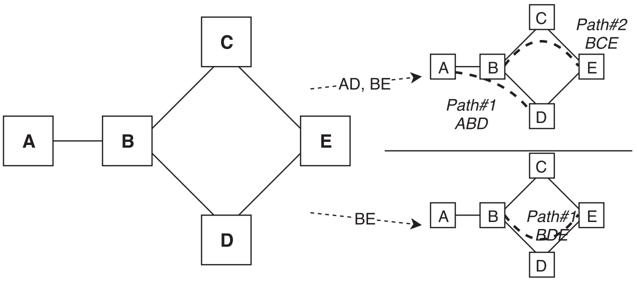

FIG. 6. Example: how path selection affects scheduling.

representing the reduced subgraph (V, E') after pruning the links where local entanglement has failed, and let us assume that the following pairs are pending being assigned a path: AE, BD, EF. The paths that it is possible to select depend on the order in which the pairs are considered, i.e., depend on scheduling: if AE is considered first, then the only possible option is to use ACDE, which leaves a disconnected reduced subgraph and, in particular, does not allow either BD or EF to be served in the same time slot (top outcome in Fig. 5); on the other hand, if BD is considered first, then it can be assigned path BCD, which leaves no options available for the pair AE, but allows the pair EF to be assigned the path EDF (bottom outcome in the same figure).

The opposite can also be true. In Fig. 6, we show another situation and we assume that the pairs pending service are: AD and BE. If BE is considered first, there are two possible paths: BCE and BDE, which (ignoring the physical distance between nodes and any other information the controller might have) are interchangeable from the point of view of path selection only, since they both require exactly one swap (C or D, respectively). However, the choice *does* have an impact on scheduling: if BCE is selected, then AD can be served with path ABD (top outcome in Fig. 6); on the other hand, if BDE is selected, then the pair AD cannot be scheduled since there is no path in the residual reduced subgraph (bottom outcome in the same figure).

We now move on exploring individually the scheduling and path selection problems.

B. SCHEDULING

We begin with the scheduling subproblem, i.e., deciding which of the pending pairs to serve in this time slot, provided

Algorithm 1: Iterative Scheduling Algorithm.

```
$(V, E')$ is the reduced subgraph after the links where
  local entanglement has failed have been removed
$\mathcal{P}$ is the set of pending pairs sorted on chronological
  order (older first); ties are broken arbitrarily
$\mathcal{S} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$
for $p \in \mathcal{P}$ do
    $\sigma = path\_selection(V, E')$
    if $\sigma = \emptyset \vee skip(\sigma)$ then
        $\mathcal{S} \leftarrow \mathcal{S} \cup p$
    else
        $E' \leftarrow E' \setminus \{e \in \sigma\}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \sigma$
        $\mathcal{P} \leftarrow \mathcal{P} \setminus p$
    end
end
for $p \in \mathcal{S}$ do
    $\sigma = path\_selection(V, E')$
    if $\sigma \neq \emptyset$ then
        $E' \leftarrow E' \setminus \{e \in \sigma\}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \sigma$
        $\mathcal{P} \leftarrow \mathcal{P} \setminus p$
    end
end
return $\mathcal{C}$
```

that there are sufficient resources (successful local entanglements) for them. To the best of our knowledge, this is a new problem that has not been investigated so far in the literature of quantum networking under these settings.

In this article, we take a first step along the road by defining a general framework for iterative scheduling algorithms based on *skipping*. The algorithm is illustrated by means of the pseudocode in Algorithm 1. Basically, we perform exactly two passes on the list of pending pairs, sorted in FIFO order. In the first pass, for every pair p we execute path selection [discussed below in Section IV-C)], which returns the best path found, according to any metric of interest for the system, or an empty value. If a valid path σ is found, then we evaluate whether it should be "skipped" or not.

1) If it is skipped, then the pair p is put aside in a list of skipped pairs $\mathcal{S}$, which keeps FIFO ordering.
2) If it is not skipped, then the path σ is confirmed and the controller will have to notify all involved nodes in the quantum network about swapping and correcting as already explained; furthermore, the pair p is removed from the list of pending pairs $\mathcal{P}$ and the reduced subgraph (V, E') is updated by pruning the edges along the path σ.

After all the pairs have been considered, we do a second pass only on the pair $\mathcal{S}$, which have been skipped in the first pass: this time we never skip valid paths.

This iterative scheduling algorithms has the following properties.

1) It favors older pairs: this is a desirable property because otherwise there would be a risk of *starvation:* if the chances of a pair to be served decreased over time, then a given request would be either served soon or likely remain in the controller queue until eventually dropped while newer requests come and go. We mention that the issue of prioritization and load balancing has been also noted in [21], where the authors have proposed a model of resource consumption of quantum repeaters and related optimization framework, which is however not directly applicable to our system model (see Section III).
2) It is efficient, in terms of time complexity, since each pair is evaluated at most twice.
3) It allows to put aside, for the moment, paths that are not deemed efficient under the current reduced subgraph.
4) However, if there are resources available after all the efficient pairs have been served, these are used for the nonefficient pairs until the resources are completely depleted.

Furthermore, the system behavior can be controlled by defining different policies on whether to skip a given (valid) path. For illustration purposes, we propose three such skipping policies that can be used in combination with the iterative scheduling algorithm above, which will be evaluated and compared in Section V.

Policy#1 – Strict FIFO: Never skip paths.

The Strict FIFO policy gives no preference to one pair compared to another based on the resulting "best" path selected. This can be a desirable feature if one wishes to maintain *fairness* among the application requests, that is all the requests are treated alike regardless of the network structure and other variable environment conditions, but can be detrimental to performance, as is well known in the context of scheduling in wireless communication systems (e.g., [25]). For instance, in a cellular system, the user terminals are generally spread all over the cell, whose radius is determined by the base station coverage; user terminals that are closer to the base station (or anyway which enjoy better channel conditions due to the lack of physical impediments or any other reason) can transmit/receive data at a faster speed than user terminals that are farther or blocked. It is in the interest of both the mobile network operator and the individual users that the wireless resources (e.g., bandwidth channels, time slots) are allocated to terminals having currently better conditions to improve the overall so-called *efficiency* (typically measured in bits/s/Hz). The best case is when a terminal experiencing poor conditions now (hence, not served) experiences better conditions shortly afterward (thus, it can be served with good efficiency). However, if this does not happen, i.e., if a user terminal is consistently experiencing poor channel conditions for a long time, we usually want to allocate it *some* resources from time to time to avoid starvation. This has led to a plethora of studies investigating the fundamental tradeoff between different types of fairness versus efficiency in various scenarios. We speculate that the same path will be traveled in the next years by quantum networking in the research community, since similar fundamental tradeoffs seem to exist, though the overall setting is, obviously, very different.

Policy#2 – Best FIFO: Skip path σ if its cost, *in terms of the same metric used by path selection*, is greater than the long-term average path cost.

The Best FIFO policy discourages paths that are relatively expensive by always putting them aside in the first pass. We decide how expensive is a path, based on whichever metric will be used during the subsequent phase selection phase. If the load is sustained, then there are high chances that no valid path at all will be found in the second pass for the paths skipped in the first pass, which makes this policy quite discriminatory against expensive paths. Indeed, this might penalize too much the pairs whose paths are *inherently* more expensive than the average: e.g., if an application wishes to establish an end-to-end entanglement path between two nodes on the opposite sides of the quantum network, then the corresponding pairs may *always* end up being served in the second pass, irrespective of the actual local entanglement successes in the current time slot. This is similar to what happens in a wireless communication system to a user terminal that is at the edge of a cell: no matter the other channel conditions, its data rate will persistently be lower than average.

Policy#3 – Random FIFO: Skip path σ with probability P_{skip}:

$$P_{\text{skip}} = \max\left\{0, 1 - \frac{E[c]}{c(\sigma)}\right\} \tag{8}$$

where $c(\sigma)$ is the cost of the path σ, *in terms of the same metric used by path selection*, and $E[c]$ is the long-term average path cost.

The random FIFO policy smooths a bit the aggressive behavior of Best FIFO: the more expensive a path is, the higher becomes the probability that the path is skipped, but if a path's cost is below the average then it is never skipped. Thus, the Random FIFO scheduling policy is always less aggressive than Best FIFO. To visualize the probability that a path is skipped, we plot in Fig. 7 the value of P_{skip} for some average cost path values.

C. PATH SELECTION

Path selection is one of the most investigated areas within the nascent topic of quantum networking. For this reason, we do not linger on this specific subproblem, except for some considerations that are relevant to the system model in Section III and the performance evaluation in Section V. We consider as a strength of the scheduling framework proposed in Section IV-B, which is the most novel contribution of this article, that different (known or future) path selection solutions can be plugged in it to achieve different objectives.

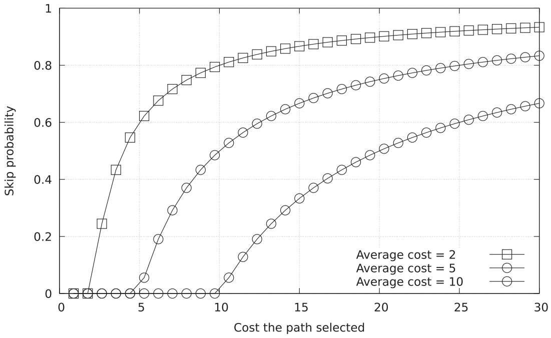

FIG. 7. Example of skip probability functions with some values of the average path cost $E[c] \in \{2, 5, 10\}$.

Even though no conclusive solutions applicable to all cases have been found, significant attention has been devoted so far to SPF using the well-known Dijkstra or Bellman–Ford algorithms [41], which both run with a polynomial number of iterations on the number of vertices and edges, or variations as in [6], [20]. Provided that some form of SPF is used, the problem is moved to the cost function to be used. For those works where path selection is done *before* knowing, which local entanglements have succeeded, an expectation on whether a given path will actually be usable or not must be factored in: this tends to prefer shorter paths (in physical distance) or paths with "better" links, i.e., where the quantum channel is expected to yield a lower loss probability due to technical capabilities of the equipment used. However, with the two-phase time slotted model that we adopt, there is no such need since we only consider links in the reduced subgraph (V, E') for which we know with certainty that local entanglement has succeeded. This simplifies the problem of finding a good metric, and in fact the authors of [29], who have proposed originally this model, in their work use the hop count as the only metric.

However, decoherence was not included in [29], and this opens the door to new considerations. In fact, as already explained, fidelity is the key figure of merit to assess the quality of an entangled pair of qubits, and the latter basically depends on the *time* that will be required for the correction bits to reach one of the end nodes. Minimizing the number of swaps, i.e., the hop count between the two end nodes in the reduced subgraph (V, E'), does not always minimize the time required for all the corrections to reach one end node, for two reasons: 1) the time depends on the physical distance, rather than the distance on the logical topology of nodes; 2) messages using classical communications can use *all* links in (V, E), not only those in the reduced subgraph (V, E').

Let us consider the simple example in Fig. 8, where the full topology is on the left, but links CB and DB fail due to unsuccessful local entanglement, yielding the reduced subgraph

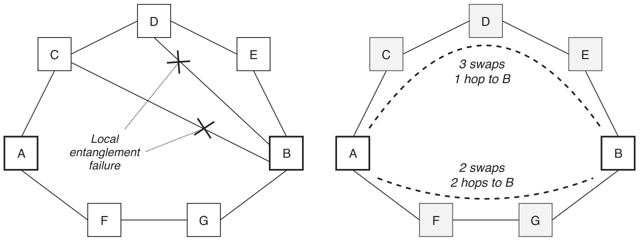

FIG. 8. **Example of end-to-end entanglement between A and B with lower latency on longer path.**

on the right. If the pair for which a path is to be selected is AB, then clearly there are two possible paths: AFGB (bottom path) requires only two swaps but the corrections from F to B need to be forwarded by G; on the other hand, ACDEB (top path) needs three swaps but all intermediate nodes can reach B with a single hop in the full graph. Thus, assuming that in the example all the links have the same physical distance, the longest path ACDEB will require *less time* for the entanglement to complete (hence, a higher fidelity with high probability). On the other hand, if the physical distance of CB or DB or EB is less than the sum of the physical distances of FB and GB, then path AFGB will require less swaps *and* a shorter time to complete entanglement. This observation leads to the definition of the following path selection strategy aimed at minimizing the entanglement latency, hence maximizing the fidelity.

MinMax path selection: for a given pair of nodes $src, dst \in V$ (where we assume that the intermediate nodes will be instructed to send their corrections to dst), we select the path σ that minimizes the maximum physical distance of any intermediate node along the path to reach dst; we break ties between same-max-distance paths by taking that with the smaller number of hops in the logical reduced subgraph (V, E').

We sketch a possible implementation of the MinMax path selection in the appendix.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our contribution in two scenarios with different network topologies: nodes arranged regularly in a grid (see Section V-B), and nodes dropped random on a bidimensional circular surface (see Section V-C). Before delving into the evaluation, we describe the methodology and tools used for the analysis (see Section V-A).

A. METHODOLOGY AND ASSUMPTIONS

The results, in this section, have been obtained with the network simulator for quantum information using discrete events (NetSquid)[2], written in Python and free to download and use. The tool has been used, for instance, to run the experiments published in [12]. We have used the NetSquid libraries to develop the modules required to realize the system model described in Section III; to allow interested researchers to replicate and extend our work, the software developed is available as *open source* on GitHub repository[3], which also includes the scenarios and postanalysis scripts, as well as the full output of the simulations.

The most important building blocks in the simulations are illustrated in Fig. 9, which shows three nodes (Alice, Bob, Carol) interconnected by means of quantum and classical links, both having the same length; such length may vary for different pairs of nodes depending on the type of experiments, as described later. At the middle of every quantum link there is a *quantum source*, which emits EPR-entangled pairs toward the nodes it is connected to. The quantum channel introduces a constant propagation latency $\delta = \frac{d}{c'}$ for the traversal of a link of length d (in km), where c' is the propagation speed (set to 200 000 km/s in the simulations). Furthermore, the quantum channel may cause the loss of a qubit with the following probability:

$$p_{\text{loss}} = 1 - (1 - p_{\text{init}}) \cdot 10^{-\eta d/10} \tag{9}$$

where p_{init} is the probability that the qubit is lost immediately after generation, due to nonideal implementation of the physical equipment performing the generation/entanglement, and η is an attenuation factor along the optical fiber (in dB/km). Instead, we assume that the classical channel is error-free, and that it only introduces a propagation delay $\delta = \frac{d}{c'}$, which are reasonable assumptions considering the state of the art of high-speed communications on optical fibers, as long as the messages exchanged contain small amount of data.

The nodes are modeled according to the blueprint in Fig. 10: each flying qubit from a quantum source is immediately absorbed into its dedicated quantum memory slot and the node can detect the successful/failed status of the local entanglement with its peer node. Qubits in memory are subject to decoherence through a *dephasing noise*, which is implemented in the simulator through applying the Pauli Z-gate stochastically with probability:

$$p_{\text{dephase}} = 1 - e^{\Delta t \cdot R_{\text{dephase}}} \tag{10}$$

where R_{dephase} is the dephasing rate (in Hz) and Δt is the time elapsed since the qubit has been absorbed into the quantum memory slot. The quantum processor operates on the qubits stored in the quantum memory to perform entanglement swap, which is implemented through a Bell measurement on two qubits, and qubit correction via X- and Z-gates (depending on the corrections bit received). Every operation requires a constant time equal to 10 ns. Furthermore, the X- and Z-gate instructions introduce noise modeled via a *depolarizing noise*, which is realized through applying stochastically Pauli X-, Y-, and Z-gates with the following probability:

$$p_{\text{depol}} = 1 - e^{\Delta T \cdot R_{\text{depol}}} \tag{11}$$

where R_{depol} is the depolarizing rate (in Hz) and ΔT is the time required for the execution of the instruction. The

[2] https://netsquid.org/ (version 0.9.8).

[3] https://github.com/ccicconetti/netsquid

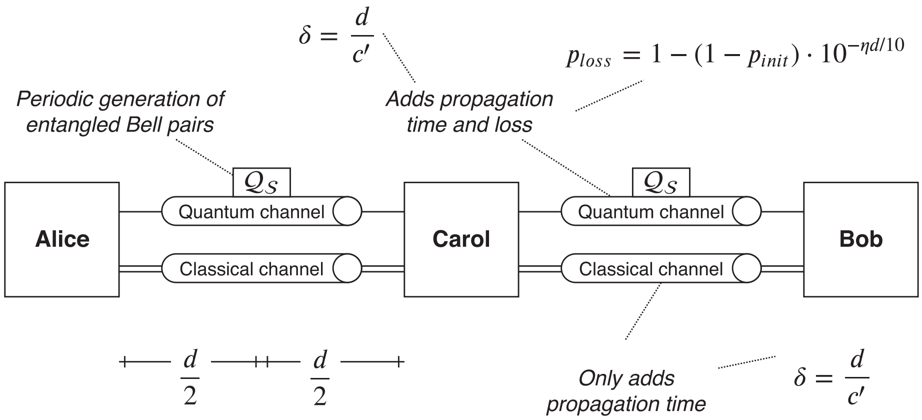

FIG. 9. Simulation basic blocks.

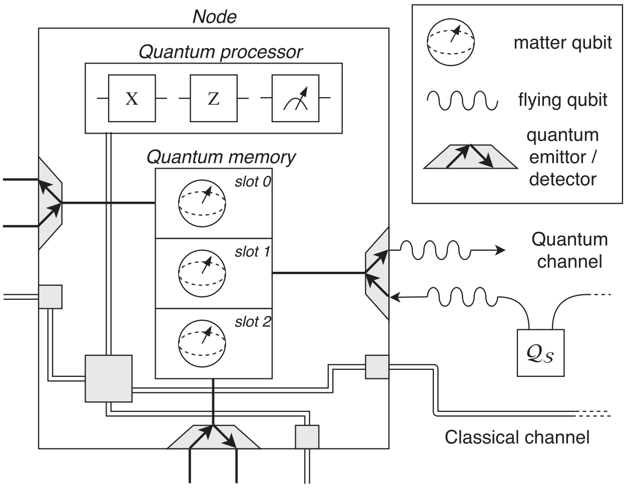

FIG. 10. System model: node.

gate operations to implement the depolarizing and dephasing noise are stochastic and mathematically equivalent to the following error operator $\varepsilon(\rho)$:

$$\varepsilon(p) = \sum_i p_i O_i \rho O_i^\dagger \tag{12}$$

where O_i is the ith quantum operator and p_i its weight $(\sum p_i = 1)$.

As far as the quantum applications are concerned, we simulate a random arrival of requests based on a Poisson process, with $1/\lambda$ being the average number of new pairs for which end-to-end entanglement is requested in a time slot. Each request is then selected uniformly from the set of all possible $u, v \in V$ pairs, with replication. As already mentioned, there is no SLA associated to an application request, e.g., minimum fidelity or maximum delay or minimum number of paths, which is part of our future work. To enforce system stability and avoid excessive delays, the controller drops application requests as soon as they have been queued for more than 10 time slots.

Every replication is deterministic, i.e., running it multiple times does not change even minimally the results. The experiments whose results depend on an initial random configuration (e.g., Section V-C) have been repeated multiples with different initialization seeds of the pseudorandom number generators. When relevant and non-negligible, 95% t-Student confidence intervals have been added to the mean values in plots.

The configuration parameters have been set based on a combination of a review of the state of the art and an initial calibration campaign described in the following with nodes evenly distributed across a chain topology. In a first group of simulations there are only three nodes (i.e., end-to-end entanglement always requires one swap) but the internode distance is increased from 1 km to 7.5 km; in the second group, we have kept the total length distance equal to 15 km and increased the number of nodes from 3 to 15. Note the last simulation of the first group is the same as the first one of the second group. We have then run two calibration scenarios per group. In the first scenario, we have tested decoherence only, i.e., we have set $p_{\text{init}} = 0$, $\eta = 0$ dB/km, $R_{\text{depol}} \in \{1, 10\}$ kHz, $R_{\text{dephase}} \in \{0.1, 1, 10\}$ MHz. The results are reported in the top two plots in Fig. 11, in terms of the fidelity, which decreases linearly with the internode distance, but hyperbolically with the number of swapping nodes. We also note that the contribution of the dephasing noise does not depend on the distance between the nodes or the number of nodes, whereas the depolarizing noise effects increase significantly with both, since they affect the time for the end-to-end entanglement to be fully established. In the second calibration scenario, we have measured the entanglement success probability only (irrespective of the resulting fidelity), which is plotted in the bottom part of Fig. 11 for $p_{\text{init}} \in \{0.1, 0.2\}$ and $\eta \in \{0.1, 0.2\}$ dB/km. As can be seen, the success probability goes exponentially to 0 as the number of nodes increases: end-to-end entanglement fails when *any* of the local entanglement along the chain fails. Note that the case of nodes in a chain has been well studied in the literature: the interested reader is referred to, e.g., [31],

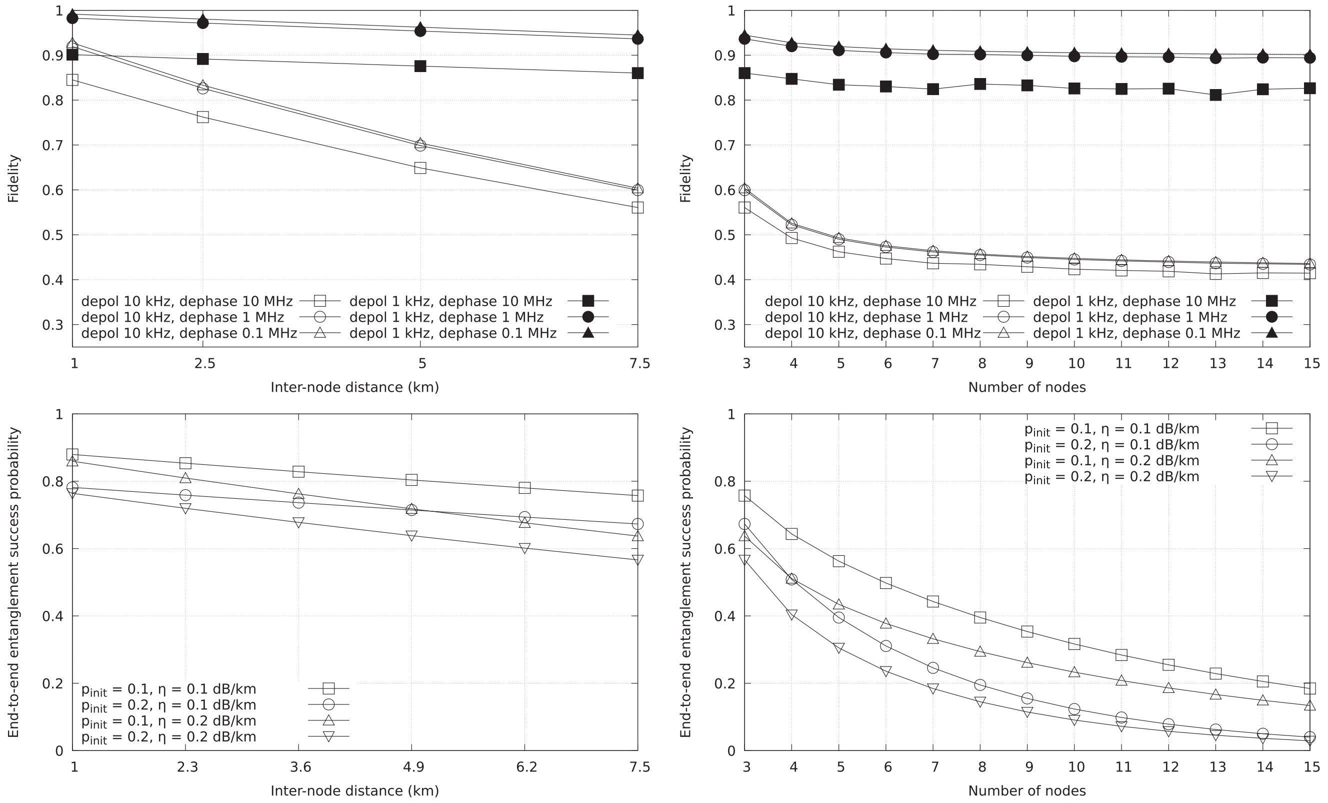

FIG. 11. Calibration experiment results.

which also covers the case of entanglement distillation. In summary, in the simulations below we have: $p_{\rm init} = 0.1$, $\eta = 0.1$ dB/km, $p_{\rm depol} = 5$ kHz, $p_{\rm dephase} = 1$ MHz. For comparison, we have run experiments with all the scheduling policies in Section IV-B (Strict FIFO, Best FIFO, Random FIFO), which are reported in the remainder of this section.

B. GRID TOPOLOGY

In this section, we report the results obtained in square grid logical topologies, which has been also used in [9] and[29]. Such a configuration is of potential interest because it reflects the need to cover a large area with the provisioning of evenly (or almost evenly) quantum repeaters in a regular structure. We have replicated the same set of simulations in two arrangements: 1) *regular* physical grid, where the internode distance was set to exactly 1.8 km; 2) *irregular* physical grid, where the internode distance was drawn from a uniform r.v. in [1.08, 2.52] km. The number of nodes was set to 25, and the offered load $1/\lambda$ has been increased from 1 to 10 in different experiments to simulate increasing load conditions. We have simulated 10 000 time slots for each configuration.

We begin by showing the *normalized entanglement rate*, which we define as the ratio between the number of paths established and the number of requests arrived (i.e., it is always smaller than or equal to 1). This metric is reported in Fig. 12 with Strict FIFO scheduling only as the load increases for the three cases of: 1) fidelity greater than 0.8, 2) fidelity greater than 0.9, and 3) any fidelity. The results show that the quantum network is lightly loaded until $1/\lambda$ is smaller than 4, since the *Any F* curves are very close to 1 (the regular and irregular curves overlap in the plot). Even under such a light load, with $F > 0.8$ and $F > 0.9$ the entanglement rate is much smaller: this is due to entanglement requests for pairs or nodes that are physically distant. With $1/\lambda$ greater than 4, all the curves start to decrease (especially *Any F*), which means that the resources are not anymore sufficient to serve most of incoming requests. Irrespective of the load, with $F > 0.8$ and $F > 0.9$ a regular grid exhibits a non-negligible performance improvement over an irregular one, which suggests that the *physical* topology of nodes plays an important role in determining the overall performance; recall

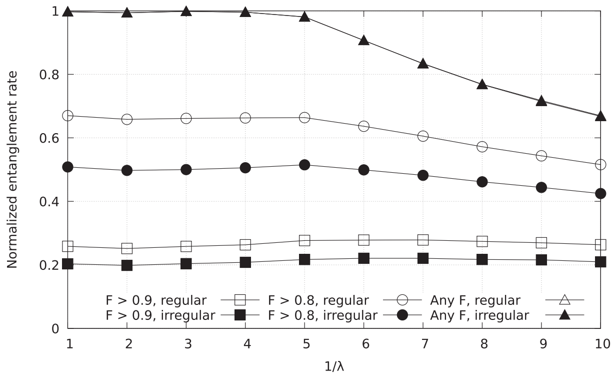

FIG. 12. Grid topology: normalized entanglement rate, Strict FIFO only.

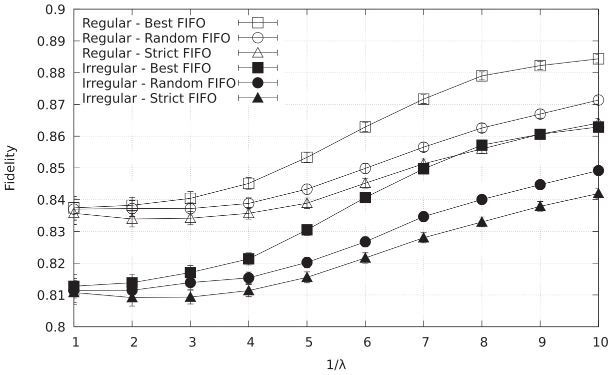

FIG. 13. Grid topology: fidelity.

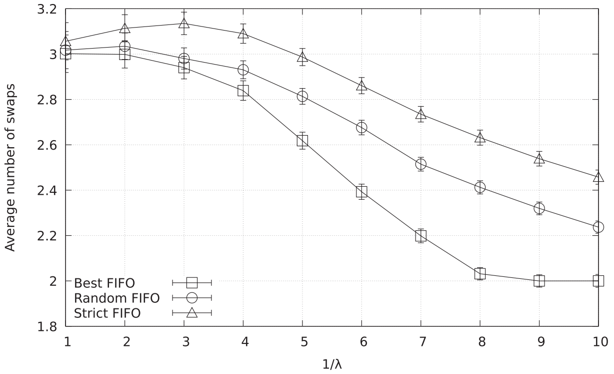

FIG. 14. Grid topology (irregular): average number of swaps.

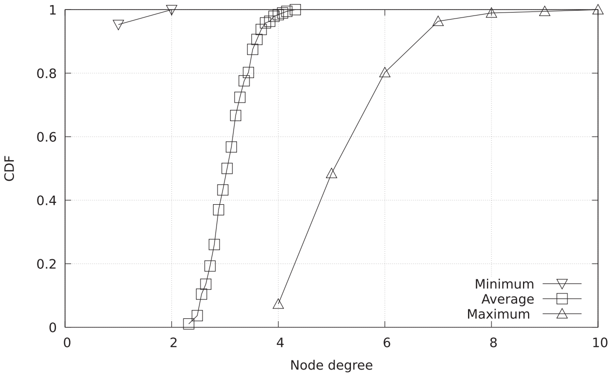

FIG. 15. Random topology: distribution of the logical topology minimum, average, and maximum graph degree.

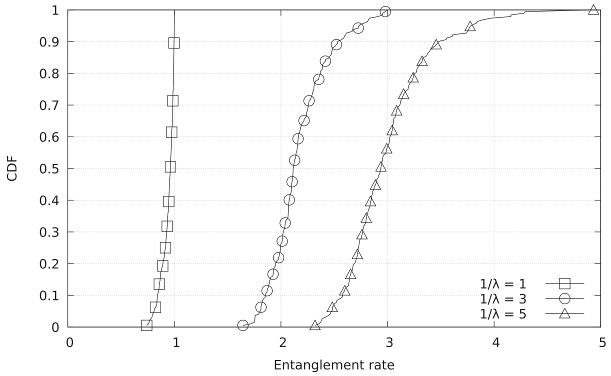

FIG. 16. Random topology: entanglement rate distribution, with Strict FIFO.

that regular and irregular scenarios have an identical logical topology.

We compare the scheduling policies in Fig. 13, which reports the average fidelity of entanglement on the established paths. All curves show the same trend: while the network is lightly loaded (i.e., $1/\lambda \leq 4$) increasing the load marginally *degrades* the average fidelity, since an increasing number of applications requests will have to share the same resources and (most) all get served anyway. On the other hand, as the quantum network resources become scarce (i.e., $1/\lambda > 4$) then the fidelity *improves* because the requests that would require a longest path get dropped more and more often: this phenomenon is exacerbated by the Random FIFO and Best FIFO policies, which are more aggressive in dropping requests that would require a higher-than-average number of swaps.

To confirm the intuition above-mentioned, we show in Fig. 14, the average number of swaps of the entanglement paths, only for irregular grids. As can be seen, all the curves decrease steeply after $1/\lambda > 4$, with Best FIFO requiring the least number of swaps, then Random FIFO, and finally Strict FIFO. In the following, we delve deeper into understanding the properties of our proposed routing scheme in a random topology.

C. RANDOM TOPOLOGY

In this section, we report the results obtained in random topologies. Unlike the grid topologies investigated in Section V-B previously, this scenario mimics an unplanned/spontaneous deployment of nodes. In particular, each scenario has been generated by dropping 25 nodes in a flat disc surface with 6 km radius in a uniform manner. After determining the node positions, a link is established between any two nodes whose distance is smaller than 2 km. If the resulting logical topology is disconnected, i.e., there are two nodes u, v for which no path can found, then the drop is discarded and nodes are dropped again, until a connected topology is found. The offered load $1/\lambda$ has been increased from 1 to 5. For each configuration, we have run 200 independent replications, each with a duration of 1000 time slots.

To give an idea of how connected the result logical graphs are, we report in Fig. 15, the distribution of the logical graph' degree, in terms of its minimum, average, and maximum value.

We begin by showing in Figs. 16 and 17, the distribution of the *entanglement rate*, defined as the number of successful entanglement paths per time slot. The two plots differ only by the minimum fidelity considered, which is 0 in Fig. 16 and 0.9 in Fig. 17. Every point in a curve corresponds to one experiment with a given seed, i.e., one drop of nodes in the

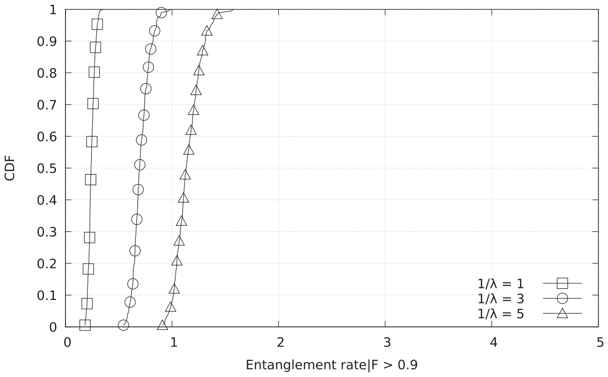

FIG. 17. Random topology: entanglement rate distribution, only with fidelity above 0.9, with Strict FIFO.

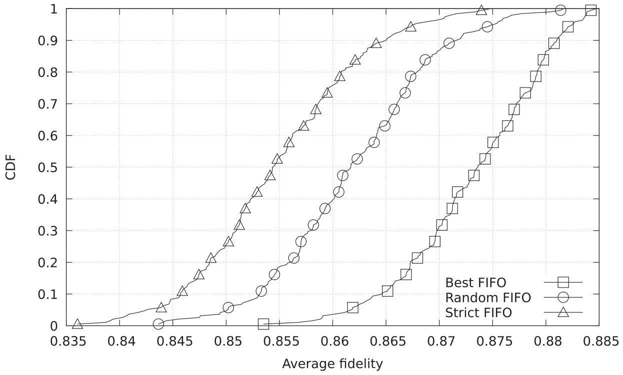

FIG. 18. Random topology: average fidelity distribution, with $1/\lambda = 5$.

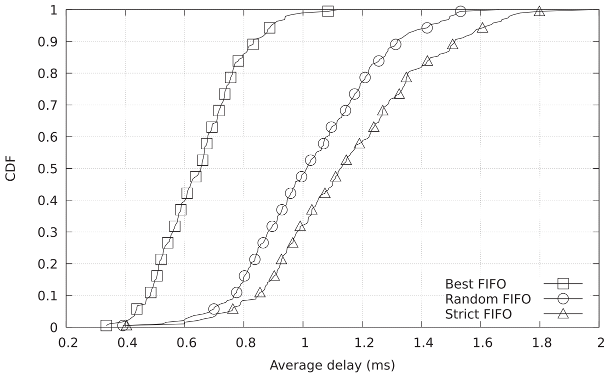

FIG. 19. Random topology: average delay distribution, with $1/\lambda = 5$.

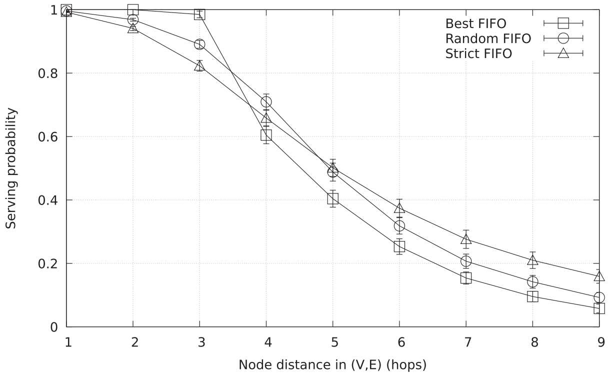

FIG. 20. Random topology: average serving probability for a given node distance, with $1/\lambda = 5$.

target area. Unlike for grid topologies, even at the relatively low load of $1/\lambda = 1$, there is a non-negligible fraction of node drops in which the entanglement rate is smaller than the offered load, despite the two topologies have the same number of nodes: this confirms that the topology can have a significant impact on performance.

In Fig. 18, we show the distribution of the average fidelity with the three scheduling policies. As can be seen, Best FIFO always perform better than Random FIFO, which in turn always perform better than Strict FIFO: this is because by skipping, deterministically or randomly, paths that are longer than the long-term average leads to i) a better utilization of the resources available in the current time slot, and ii) a preference of application requests for nodes that are not too distant.

This is reflected in a manifest manner by the average delay, whose distribution is reported in Fig. 19. Note that the delay includes the time a request is queued while waiting to be served by the controller. Best FIFO outperforms significantly both Random FIFO and Strict FIFO.

However, while a better utilization of the resources is always desirable, aggressively postponing requests requiring *in this time slot* a high number of entanglement swaps might lead to discrimination against requests for nodes that are distant *in all time slots*, because of the structure of the network. This is shown in Fig. 20, which reports the average serving probability for requests of a given distance in the full logical graph. As can be seen, all scheduling policies (including Strict FIFO) tend to prefer shorter paths, since it is more likely that resources in a time slot are available for them compared to requests involving nodes farther apart. However, this effect is much more prominent with Best FIFO, which serves with probability almost 1 all requests within three hops, but then drops the serving probability for all others until it becomes almost negligible for the requests with longest paths. Instead, Random FIFO induces a behavior that is smoother: until node distance 5, it is slightly better than Strict FIFO, afterward it becomes slightly worse.

To summarize, we confirm that the fundamental tradeoff fairness versus efficiency identified in Section IV-B can be found in the scenarios simulated, and we claim that further research will be needed to better understand its implications on realistic quantum networks and application requirements, as technology will progress along the Quantum Internet road ahead.

VI. CONCLUSION AND FUTURE WORK

In this article, we studied the problem of quantum routing in a time slotted system model, separated into *path selection*, i.e., assigning a path to a given application request, and *scheduling*, i.e., deciding, which application requests to serve in the current time slot. While path selection was already investigated in the literature, scheduling is a new problem, and we proposed for it a generic framework, for which we have defined three instances: Best FIFO and Random FIFO: prefer shorter paths to longer ones by putting them aside in a first pass in a deterministic or stochastic manner, respectively; Best FIFO simply serves application requests in the order in which they arrive. Extensive simulation results highlighted that scheduling has a major impact on performance, in terms of entanglement rate, fidelity, and delay. Furthermore, a tradeoff between fairness of treatment of the application requests and system-level metrics has come to light, which opens the way to further investigations and a better understanding of the requirements of quantum networks and its applications.

In addition to the future work already mentioned, we foresee the following open research directions in the context of this article: extension to multipartite entanglement, e.g., along the lines of [42]; integration with communication protocols, e.g., [26], and link layer stacks, e.g., [12]; analysis of the relation between distributed quantum applications and underlying interconnection network [14]; study of the fairness versus efficiency tradeoff in local link state routing protocols, e.g., [9]; extension to multiple local entanglement per node pair, such as in [29]; definition of a SLA for distributed quantum applications [11] and service differentiation [19]; and further simulations with more general topologies and application request models.

APPENDIX
MINMAX IMPLEMENTATION

One possible implementation of the MinMax path selection algorithm is as follows. For a given logical undirected reduced subgraph (V, E') and a destination node dst of end-to-end entanglement, we can create a *helper graph*(V_H, E_H) that has the same edges as (V, E'), but the cost of directed edge e_{uv} from node u to node v is assigned the following quantity:

$$c(e_{uv}) = \Omega \cdot \text{dist}^{\text{phys}}_{(V,E)}(v, dst) + \text{dist}_{(V,E')}(v, dst) \qquad (13)$$

where $\text{dist}^{\text{phys}}_{(V,E)}()$ is the distance between nodes on the full graph (where the cost of edges is the physical distance between adjacent nodes), $\text{dist}_{(V,E')}()$ is the distance in number of hops on the reduced subgraph (i.e., the number of swaps/measurements), and Ω is a constant factor that is big enough that the physical distance always dominates over the number of hops in the reduced subgraph. Once the helper graph has been created, it is sufficient to use Dijkstra's algorithm, but using a $\max(\cdot)$ operator to combine the costs of edges instead of the $+(\cdot)$ operator typically employed. Yang and Wang [44] proved that Dijkstra's algorithm is optimal for any metric that exhibits the properties of right-monotonicity and right-isotonicity, which are trivial to prove for the $\max(\cdot)$ operator: in short, right-monotonicity means that the cost of a path cannot *decrease* if another edge is *added* to it, whereas right-isotonicity means that the cost of a path does not decrease when appended by another path.

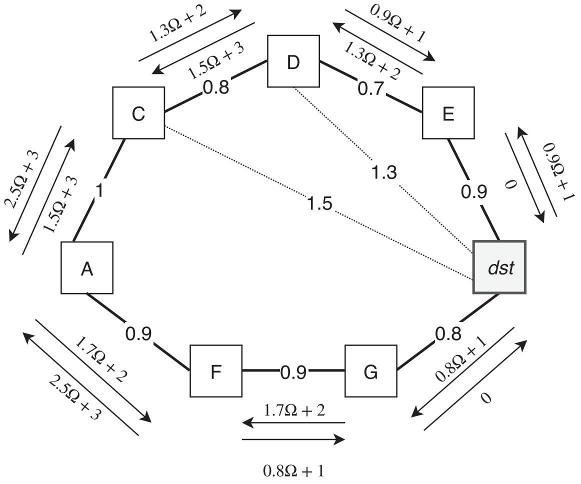

FIG. 21. Example of helper graph construction for MinMax path selection.

In Fig. 21, we show the construction of the directed (V_H, E_H) helper graph in the example already introduced by Fig. 8. Only, we now assume the physical distances between nodes reported on the edges for the full graph (assuming without loss of generality that distances are symmetric). Note that we also report the distances for edges between C, D, and dst, even though local entanglement has failed in those links in this example, because those edges can be used for transferring traditional information, i.e., the correction bits toward dst. If A is the source node, then the path selected by MinMax is ACDEdst, because its cost $(1.5\,\Omega + 3)$ is smaller than that in the alternative path AFG dst (which would cost $1.7\,\Omega + 2$), provided that $\Omega >> 1$ as assumed previously.

REFERENCES

[1] R. Arroyo-Valles, A. Simonetto, and G. Leus, "Consistent sensor, relay, and link selection in wireless sensor networks," *Signal Process.*, vol. 140, pp. 32–44, 2017, *arXiv: 1705.02298.*

[2] D. Awschalom *et al.*, "Development of quantum interconnects (QuICs) for next-generation information technologies," *PRX Quantum*, vol. 2, no. 1, Feb. 2021, Art. no. 017002, doi: 10.1103/PRXQuantum.2.017002.

[3] B. Bartlett, "A distributed simulation framework for quantum networks and channels," Aug. 2018, *arXiv:1808.07047*.

[4] M. K. Bhaskar *et al.*, "Experimental demonstration of memory-enhanced quantum communication," *Nature*, vol. 580, no. 7801, pp. 60–64, Apr. 2020.

[5] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum internet: Networking challenges in distributed quantum computing," *IEEE Netw.*, vol. 34, no. 1, pp. 137–143, Jan. 2020, doi: 10.1109/MNET.001.1900092.

[6] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22299–22312, 2017, doi: 10.1109/ACCESS.2017.2763325.

[7] M. Caleffi and A. S. Cacciapuoti, "Quantum switch for the quantum internet: Noiseless communications through noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 3, pp. 575–588, Mar. 2020, doi: 10.1109/JSAC.2020.2969035.

[8] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi, "Quantum internet: From communication to distributed computing," in *Proc. 5th ACM Int. Conf. Nanoscale Comput. Commun.*, 2018, pp. 1–4, doi 10.1145/3233188.3233224.

[9] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," 2019, *arXiv:1907.11630.*

[10] X. Chen and P. Qiu, "The design of quantum frame via time division multiplexing," in *Proc. IEEE Globecom Workshops*, 2017, pp. 1–5.

[11] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, "Towards a distributed quantum computing ecosystem," *IET Quantum Commun.*, May 2020, doi: 10.1049/iet-qtc.2020.0002.

[12] A. Dahlberg *et al.*, "A link layer protocol for quantum networks," in *Proc. ACM Special Int. Group Data Commun.*, 2019, pp. 159–173, doi: 10.1145/3341302.3342070.

[13] A. Dahlberg and S. Wehner, "SimulaQron-a simulator for developing quantum internet software," *Quantum Sci. Technol.*, vol. 4, no. 1, Sep. 2018, Art. no. 015001, doi: 10.1088/2058-9565/aad56e.

[14] S. DiAdamo, M. Ghibaudi, and J. Cruise, "Distributed quantum computing and network control for accelerated VQE," *IEEE Trans. Quantum Eng.*, vol. 2, Feb. 2021, Art. no. 3100921, doi: 10.1109/TQE.2021.3057908.

[15] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[16] W. Dür and H. J. Briegel, "Entanglement purification and quantum error correction," *Rep. Prog. Phys.*, vol. 70, no. 8, pp. 1381–1424, Aug. 2007, doi: 10.1088/0034-4885/70/8/R03.

[17] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *Proc. 16th Annu. Symp. Found. Comput. Sci.*, 1975, pp. 184–193, doi: 10.1109/SFCS.1975.21.

[18] S. Guha *et al.*, "Rate-loss analysis of an efficient quantum repeater architecture," *Phys. Rev. A*, vol. 92, no. 2, Aug. 2015, Art. no. 022357, doi: 10.1103/PhysRevA.92.022357.

[19] L. Gyongyosi and S. Imre, "Entanglement availability differentiation service for the quantum internet," *Sci. Rep.*, vol. 8, no. 1, Dec. 2018, Art. no. 10620, doi: 10.1038/s41598-018-28801-3.

[20] L. Gyongyosi and S. Imre, "Opportunistic entanglement distribution for the quantum internet," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, Art. no. 2219, doi: 10.1038/s41598-019-38495-w.

[21] L. Gyongyosi and S. Imre, "Resource prioritization and balancing for the quantum internet," *Sci. Rep.*, vol. 10, no. 1, Dec. 2020, Art. no. 22390, doi: 10.1038/s41598-020-78960-5.

[22] S. Khatri, A. J. Brady, R. A. Desporte, M. P. Bart, and J. P. Dowling, "Spooky action at a global distance: Analysis of space-based entanglement distribution for the quantum internet," *NPJ Quantum Inf.*, vol. 7, no. 1, Dec. 2021, doi: 10.1038/s41534-020-00327-5.

[23] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[24] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas, "Fast approximation algorithms for multicommodity flow problems," *J. Comput. Syst. Sci.*, vol. 50, no. 2, pp. 228–243, 1995, doi: 10.1006/jcss.1995.1020.

[25] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *Proc. IEEE 22nd Annu. Joint Conf. IEEE Comput. Commun. Societies*, 2003, vol. 2, pp. 1106–1115, doi: 10.1109/INFCOM.2003.1208947.

[26] T. Matsuo, C. Durand, and R. Van Meter, "Quantum link bootstrapping using a RuleSet-based communication protocol," *Phys. Rev. A*, vol. 100, no. 5, Nov. 2019, Art. no. 052320, doi: 10.1103/PhysRevA.100.052320

[27] P. Nain, G. Vardoyan, S. Guha, and D. Towsley, "On the analysis of a multipartite entanglement distribution switch," *Proc. ACM Measurement Analysis Comput. Syst.*, vol. 4, no. 2, pp. 1–39, Jun. 2020, doi: 10.1145/3392141.

[28] J. Notzel and S. DiAdamo, "Entanglement-enhanced communication networks," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, Denver, CO, USA, 2020, pp. 242–248, doi: 10.1109/QCE49297.2020.00038.

[29] M. Pant *et al.*, "Routing entanglement in the quantum Internet," *NPJ Quantum Inf.*, vol. 5, no. 1, Dec. 2019, Art. no. 25, doi: 10.1038/s41534-019-0139-x.

[30] S. Perseguers, J. I. Cirac, A. Acín, M. Lewenstein, and J. Wehr, "Entanglement distribution in pure-state quantum networks," *Phys. Rev. A*, vol. 77, no. 2, Feb. 2008, Art. no. 022308, doi: 10.1103/PhysRevA.77.022308.

[31] S. Pirandola, "Capacities of repeater-assisted quantum communications," 2017, *arXiv:1601.00966*.

[32] A. Pirker and W. Dür, "A quantum network stack and protocols for reliable entanglement-based networks," *New J. Phys.*, vol. 21, no. 3, Mar. 2019, Art. no. 033003, doi: 10.1088/1367-2630/ab05f7.

[33] A. Ribeiro, N. D. Sidiropoulos, and G. B. Giannakis, "Optimal distributed stochastic routing algorithms for wireless multihop networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4261–4272, Nov. 2008, doi: 10.1109/T-WC.2008.070511.

[34] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner, "Shortcuts to quantum network routing," 2016, *arXiv:1610.05238*.

[35] S.-T. Cheng, C.-Y. Wang, and M.-H. Tao, "Quantum communication for wireless wide-area networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 7, pp. 1424–1432, Jul. 2005, doi: 10.1109/JSAC.2005.851157.

[36] A. Singh, K. Dev, H. Siljak, H. D. Joshi, and M. Magarini, "Quantum internet- applications, functionalities, enabling technologies, challenges, and research directions," 2021, *arXiv:2101.04427*.

[37] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," *Comput. Commun. Rev.*, vol. 29, no. 4, pp. 81–94, 1999, doi: 10.1145/316194.316208.

[38] N. Tomm *et al.*, "A bright and fast source of coherent single photons," *Nat. Nanotechnol.*, vol. 16, pp. 399–403, Jan. 2021, doi: 10.1038/s41565-020-00831-x.

[39] J. A. W. van Houwelingen, A. Beveratos, N. Brunner, N. Gisin, and H. Zbinden, "Experimental quantum teleportation with a three-bell-state analyzer," *Phys. Rev. A*, vol. 74, no. 2, Aug. 2006, Art. no. 022303, doi: 10.1103/PhysRevA.74.022303.

[40] R. Van Meter, T. D. Ladd, W. J. Munro, and K. Nemoto, "System design for a long-line quantum repeater," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 1002–1013, Jun. 2009, doi: 10.1109/TNET.2008.927260.

[41] 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, Dec. 2013, doi: 10.1007/s13119-013-0026-2.

[42] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the capacity region of bipartite and tripartite entanglement switching," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 48, no. 3, pp. 45–50, Dec. 2020, doi: 10.1145/3453953.3453963.

[43] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, Oct. 2018, Art. no. 9288, doi: 10.1126/science.aam9288.

[44] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *Proc. IEEE 27th Conf. Comput. Commun.*, 2008, pp. 1615–1623, doi: 10.1109/INFOCOM.2008.222.

[45] J. Yin *et al.*, "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, Jun. 2017, doi: 10.1126/science.aan3211.

[46] Y. Yu *et al.*, "Entanglement of two quantum memories via fibres over dozens of kilometres," *Nature*, vol. 578, no. 7794, pp. 240–245, Feb. 2020, doi: 10.1038/s41586-020-1976-7.

Claudio Cicconetti received the Ph.D. degree in information engineering in 2007 and Laurea degree in computer science engineering from the University of Pisa, Pisa, Italy, in 2003.

He has been working in Intecs S.p.a. (Italy) from 2009 to 2013 as an R&D Manager and in MBI s.r.l. (Italy) from 2014 to 2018 as a Principal software engineer. He is currently a Researcher with the Ubiquitous Internet group of IIT-CNR, Italy. He has been involved in several international R&D projects funded by the European Commission and the European Space Agency and is involved in the organization and technical program of several international conferences. He has an Erdös Number 3. He has coauthored 60+ papers published in international journals and peer-reviewed conference proceedings and two international patents.

Marco Conti received the Laurea degree (magna cum laude) in computer science in 1987 from the University of Pisa, Pisa, Italy. He is the Director of IIT-CNR. He was the coordinator of FET-open project "Mobile Metropolitan Ad hoc Network (MobileMAN)" (2002–2005), and he has been the CNR Principal Investigator (PI) in several projects funded by the European Commission: FP6 FET HAGGLE (2006–2009), FP6 NEST MEMORY (2007–2010), FP7 EU-MESH (2008–2010), FP7 FET SOCIALNETS (2008–2011), FP7 FIRE project SCAMPI (2010–2013), FP7 FIRE EINS (2011–2015), and CNR Co-PI for the FP7 FET project RECOGNITION (2010–2013). He is currently the CNR PI for the H2020 SCC REPLICATE project. He has authored and coauthored in journals and conference proceedings more than 300 research papers related to design, modeling, and performance evaluation of computer and communications systems, and their use for decentralized solutions for self-organizing networks.

Andrea Passarella received the Ph.D. degree in information engineering in 2005 and Laurea degree (magna cum laude) in computer science engineering in 2001 from the University of Pisa, Pisa, Italy. He is currently a Research Director with IIT-CNR, Pisa, Italy, and Head of the Ubiquitous Internet Group. Before joining UI-IIT, he was a Research Associate with the Computer Laboratory, University of Cambridge, U.K. He has authored and coauthored 150+ papers in international journals and conferences, receiving four best paper awards, including at IFIP Networking 2011, and IEEE WoWMoM 2013. His research interests include design and performance evaluation of Future Internet networks with focus on fog computing and mobile cloud technologies, content-centric networks, offloading, and more generally on self-organizing mobile networks.

Dr. Passarella is a member of NetWorld 2020 ETP Expert Group. He was CNR Principal Investigator (PI) in the H2020 AUTOWARE, FP7 MOTO, EIT Digital MOSES and Efficient IoT projects, and CNR co-PI in the FP7 SOCIALNETS, SCAMPI, RECOGNITION, and H2020 REPLICATE and SoBigData projects. He was Chair/Co-Chair of several IEEE and ACM conferences/workshops (including IFIP IoP 2016, ACM CHANTS 2014, and IEEE WoWMoM 2011 and 2019) and Member of the organizing committees of several conferences (including ACM MobiSys, IEEE PerCom, IEEE WoWMoM, IEEE INFOCOM). He is the founding Associate Editor-in-Chief of the Elsevier journal *Online Social Networks and Media*, and Area Editor for the Elsevier *Pervasive and Mobile Computing* journal and Inderscience *International Journal of Autonomous and Adaptive Communications Systems*. He was Co-Editor of several special sections in international journals. He is the Chair of the IFIP WG 6.3 Performance of Communication Systems.