QUANTUM INTERNET ALLIANCE

D4.4 Design of Quantum Internet Network Stack Report

Document History

Revision Nr	Description	Author	Review	Date
V1	First version	Wojciech Kozlowski	Stephanie Wehner	12-11-2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820445.

Index

1. Abstract

The design of a Quantum Internet Network Stack is reported in three attached papers. A brief summary and their contribution in this context is summarised.

2. Keyword list

Quantum Internet, Quantum Networks, Link Layer, quantum communication

3. Acronyms & Abbreviations

EGP	Entanglement Generation Protocol
MHP	Midpoint Heralding Protocol
QNP	Quantum Network Protocol
MPLS	Multi-Protocol Label Switching

4. Introduction

The design of a Quantum Internet Network Stack is reported in the **three attached papers**. A brief summary and their contribution in this context is summarised below. Although it is not part of this deliverable, it is worth noting that we are currently in the process of integrating a software implementation of the network stack on the NV nodes developed in QIA in D1.3.

5. A Link Layer Protocol for Quantum Networks (Appendix 1)

Proceedings of the ACM Special Interest Group on Data Communication. 2019. 159-173.

In this paper we introduce the quantum network stack (Fig. 1) which is inspired by the classical TCP/IP network stack. The network stack separates different layers of abstraction and allows one to build a complex network system by building on top of the services provided by the lower layers.

Application	
Transport	Qubit transmission
Network	Long distance entanglement
Link	Robust entanglement generation
Physical	Attempt entanglement generation

Figure 1: The Quantum Network Stack

This paper also introduces and evaluates a link layer (Entanglement Generation Protocol – EGP) and physical layer (Midpoint Heralding Protocol – MHP) protocol which satisfy the requirements of the first two layers of the network stack proposed and are the first step towards the design of a quantum network stack, and thus, functional quantum network systems.

6. Designing a Quantum Network Protocol (Appendix 2)

Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies. 2020.

This paper proposes and evaluates a network layer protocol (Quantum Network Protocol – QNP) which satisfies the requirements of the network layer of the quantum network stack proposed in the previous paper. It builds upon the link and physical layers proposed in the previous paper to connect them via entanglement swapping into a long-distance, end-to-end entanglement generation service. It introduces the concept of quantum virtual circuits and is designed to be a building block for more complex quantum network services much like MPLS (Multi-Protocol Label Switching) protocol is for classical networks.

7. An Architecture for Meeting Quality-of-Service Requirements in Multi-User Quantum Networks (Appendix 3)

Manuscript, publication pending the filing of a patent application

This paper builds upon the network stack proposed and developed in the previous two papers and puts forward a network architecture capable of meeting end-to-end quality of service requirements for entanglement generation in a multi-user network. It forms a crucial step towards a functional network (near-term and long-term) by addressing questions that go beyond the physical process of entanglement generation and that require coordination between multiple quantum network nodes and protocols.

8. Note on the transport layer

Through our work on the protocol stack and application development in the QIA project we have concluded that the transport layer, as originally proposed, is not currently necessary. Its functionality is better either directly included in the application above or in the network layer below. However, it may be that a transport layer becomes necessary in the future.

9. Appendix

A Link Layer Protocol for Quantum Networks

Axel Dahlberg*[1,2], Matthew Skrzypczyk*[1,2], Tim Coopmans[1,2], Leon Wubben[1,2], Filip Rozpędek[1,2], Matteo Pompili[1,2], Arian Stolk[1,2], Przemysław Pawełczak[1], Robert Knegjens[1], Julio de Oliveira Filho[1], Ronald Hanson[1,2], and Stephanie Wehner[1,2]
[1]QuTech, Delft University of Technology and TNO, [2]Kavli Institute of Nanoscience
Delft, The Netherlands
s.d.c.wehner@tudelft.nl

ABSTRACT

Quantum communication brings radically new capabilities that are provably impossible to attain in any classical network. Here, we take the first step from a physics experiment to a quantum internet *system*. We propose a functional allocation of a quantum network stack, and construct the first physical and link layer protocols that turn ad-hoc physics experiments producing heralded entanglement between quantum processors into a well-defined and robust service. This lays the groundwork for designing and implementing scalable control and application protocols in platform-independent software. To design our protocol, we identify use cases, as well as fundamental and technological design considerations of quantum network hardware, illustrated by considering the state-of-the-art quantum processor platform available to us (Nitrogen-Vacancy (NV) centers in diamond). Using a purpose built discrete-event simulator for quantum networks, we examine the robustness and performance of our protocol using extensive simulations on a supercomputing cluster. We perform a full implementation of our protocol in our simulator, where we successfully validate the physical simulation model against data gathered from the NV hardware. We first observe that our protocol is robust even in a regime of exaggerated losses of classical control messages with only little impact on the performance of the system. We proceed to study the performance of our protocols for 169 distinct simulation scenarios, including trade-offs between traditional performance metrics such as throughput, and the quality of entanglement. Finally, we initiate the study of quantum network scheduling strategies to optimize protocol performance for different use cases.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Link-layer protocols**; • **Hardware** → **Quantum communication and cryptography**; • **Computer systems organization** → **Quantum computing**;

KEYWORDS

Quantum Internet, Quantum Networks, Link Layer

SIGCOMM '19, August 19–23, 2019, Beijing, China

ACM ISBN 978-1-4503-5956-6/19/09…$15.00
https://doi.org/10.1145/3341302.3342070

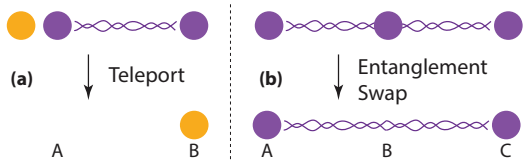


Figure 1: Entanglement enables long-distance quantum communication: (a) once two qubits (purple/dark) are confirmed to be entangled (threaded links between qubits), a data qubit (yellow/light) can be sent deterministically using teleportation [11], consuming the entangled pair; (b) long-distance entanglement can be built from shorter segments: If node A is entangled with B (repeater), and B with C, then B can perform *entanglement swapping* [96] to create long-distance entanglement between the qubits at A and C.

ACM Reference Format:
Axel Dahlberg*[1,2], Matthew Skrzypczyk*[1,2], Tim Coopmans[1,2], Leon Wubben[1,2], Filip Rozpędek[1,2], Matteo Pompili[1,2], Arian Stolk[1,2], Przemysław Pawełczak[1], Robert Knegjens[1], Julio de Oliveira Filho[1], Ronald Hanson[1,2], and Stephanie Wehner[1,2]. 2019. A Link Layer Protocol for Quantum Networks. In *ACM SIGCOMM 2019 Conference (SIGCOMM '19), August 19–23, 2019, Beijing, China.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3341302.3342070

1 INTRODUCTION

Quantum communication enables the transmission of quantum bits (qubits) in order to achieve novel capabilities that are provably impossible using classical communication. As with any radically new technology, it is hard to predict all uses of a future Quantum Internet [54, 90], but several major applications have already been identified depending on the stage of quantum network development [90]. These range from cryptography [10, 37], sensing and metrology [41, 55], distributed systems [9, 33], to secure quantum cloud computing [19, 24].

Qubits are fundamentally different from classical bits, which brings significant challenges both to the physical implementation of quantum networks, as well as the design of quantum network architectures. Qubits cannot be copied, ruling out signal amplification or repetition to overcome transmission losses to bridge great distances. Two qubits can share a special relation known as *entanglement*, even if these two qubits are stored at distant network nodes. Such entanglement is central not only to enable novel applications, but also provides a means to realize a quantum repeater, which enables quantum communication over long-distances (Figure 1).

At present, short-lived entanglement has been produced probabilistically over short distances (≈ 100 km) on the ground by sending photons over standard telecom fiber (see e.g. [36, 49]), as well as

*The first two authors contributed equally to this work.

from space over 1203 km from a satellite [93]. Such systems can allow the realization of applications in the prepare-and-measure stage [90] of quantum networks on point-to-point links, i.e. the stage in where end nodes can only prepare and measure single qubits. However, they cannot by themselves be concatenated to allow the transmission of qubits over longer distances. Using such technology, secure communication links have been realized over short distances on the ground, individually or in chains of trusted nodes [90] - see e.g. [4, 38, 92]). In a chain of trusted nodes, a separate key is produced between each pair of nodes along the chain, and hence compromising any of those nodes leads to a break in security. Importantly, trusted nodes do not enable the end-to-end transmission of qubits.

In order to enable long-distance quantum communication and the execution of complex quantum applications, we would like to produce long-lived entanglement between two quantum nodes that are capable of storing and manipulating qubits. To do so efficiently (Section 3.1), we need to confirm entanglement generation by performing *heralded* entanglement generation. This means that there is a *heralding signal* that can be sent to the two nodes to indicate that entanglement has been successfully generated. The generation of a specific entangled pair is not heralded by default, since it requires the ability to generate such a signal without collapsing the quantum state of the entangled qubits (see e.g. Section 4.4 for a method that achieves this).

The current world distance record for producing heralded entanglement is 1.3 km, which has been achieved using a solid state platform known as Nitrogen-Vacancy (NV) centers in diamond [44]. Intuitively, this platform is a few qubit (as of now 8 [15]) quantum computer capable of executing arbitrary quantum gates and measurements, with an optical interface to connect to other nodes for entanglement generation. Key capabilities of the NV platform have already been demonstrated, including qubit lifetimes of 1.46 s [1], entanglement production faster than it is lost [47], and sending qubits over entanglement using deterministic quantum teleportation [68]. Other hardware platforms exist that are identical on an abstract level (quantum computer with an optical interface), and on which heralded long-lived entanglement generation has been demonstrated (e.g. Ion Traps [61], and Neutral Atoms [45]). Theoretical proposals and early stage demonstrations of individual components also exists for other physical platforms (e.g. quantum dots [32], rare earth ion-doped crystals [84], atomic gases [25, 50], and superconducting qubits [65]), but their performance is not yet good enough to generate entanglement faster than it is lost.

Up to now, the generation of long-lived entanglement has been the domain of highly sophisticated, but arguably ad-hoc physics experiments. We are now on the verge of seeing early stage quantum networks becoming a reality, entering a new phase of development which will require a *joint effort* across physics, computer science and engineering to overcome the many challenges in scaling such networks. In this paper, we take the first step from a physics experiment to a fully-fledged quantum communication *system*.

Design considerations and use cases: We identify general design considerations for quantum networks based on fundamental properties of entanglement, and technological limitations of near-term quantum hardware, illustrated with the example of our NV platform. For the first time, we identify systematic use cases, and employ them to guide the design of our stack and protocols.

Application	
Transport	Qubit transmission
Network	Long distance entanglement
Link	Robust entanglement generation
Physical	Attempt entanglement generation

Figure 2: Functional allocation in a quantum network stack. Entanglement forms an inherent connection already at the physical layer, which contrasts with classical networking where shared state is typically only established at much higher layers.

Functional allocation quantum network stack: We propose a functional allocation of a quantum network stack, and define the service desired from its link layer to satisfy use case requirements and design considerations. In analogy to classical networking, the quantum link layer is responsible for producing entanglement between two nodes that share a direct physical connection (e.g. optical fiber).

First physical and link layer entanglement generation protocols: We proceed to construct the world's first physical and link layer protocols for a quantum network stack that turn ad-hoc physics experiments producing heralded entanglement into a well defined service. This lays the groundwork for designing and implementing control and application protocols in platform-independent software in order to build and scale quantum networks. At the physical layer, we focus primarily on the quantum hardware available to us (NV platform), but the same protocol could be realized directly using Ion Traps or Neutral Atoms, as well as—with minor changes—other means of producing physical entanglement [76]. Our link layer protocol takes into account the intricacies of the NV platform, but is in itself already platform independent.

Simulation validated against quantum hardware: Using a purpose built discrete-event simulator for quantum networks, we examine the robustness and performance of our protocol using more than 169 scenarios totaling 94244 h wall time and 707 h simulated time on a supercomputing cluster. To this end, we perform a complete implementation of our protocols and let them use simulated quantum hardware and communication links. To illustrate their performance, we consider two concrete short and long-distance scenarios based on the NV platform: (1) Lab where the nodes A and B are 2 m apart. Since this setup has already been realized, we can use it to compare the performance of the entanglement generation implemented on real quantum hardware against the simulation to validate its physical model, and (2) a planned implementation of QL2020 where A and B are in two Dutch cities separated by $\approx$25 km over telecom fiber. Next to investigating trade-offs between traditional performance metrics (e.g. throughput or latency) and genuinely quantum ones (fidelity, Section 4.2), we take a first step in examining different quantum network scheduling strategies to optimize performance for different use cases.

2 RELATED WORK

At present there is no quantum network stack connected to quantum hardware, no link layer protocols have been defined to produce

long-lived entanglement, and no quantum networks capable of end-to-end qubit transmission or entanglement production have been realized (see [90] and references therein). Also, we are not aware of any other systematic investigation on use cases informing requirements for such an architecture.

A functional allocation of a stack for quantum repeaters and protocols controlling entanglement distillation (a process of correcting errors in entanglement) has been outlined in [5, 86, 88, 89], which is complementary to this work. This is very useful to ultimately realize entanglement distillation, even though no complete control protocols or connection to a hardware system were yet given. We remark that here we do not draw layers from specific protocols like entanglement distillation, but focus on the service that these layers should provide (a layer protocol may of course choose distillation as a means to realize requirements). An outline of a quantum network stack was also put forward in [69], including an appealing high level quantum information theory protocol transforming multi-partite entanglement. However, this high level protocol does not yet consider failure modes, hardware imperfections, nor the requirements on entanglement generation protocols and the impact of classical control. Plans to realize the physical layer of a quantum network from a systems view were put forward in [58], however development has taken a different route.

In the domain of single-use point-to-point links for quantum key distribution (QKD), software has been developed for trusted repeater networks [90] to make use of such key in e.g. VoIP [56]. However, these do not allow end-to-end transmission of qubits or generation of entanglement, and rely on trust in the intermediary nodes who can eavesdrop on the communication. Control using software defined networks (SDN) to assist trusted repeater nodes has been proposed, e.g. [2, 94]. These QKD-centric protocols however do not address control problems in true quantum networks aimed at end-to-end delivery of qubits, and the generation of long-lived entanglement.

In contrast, classical networking knows a vast literature on designing and analyzing network protocols. Some ideas can indeed be borrowed from classical networking such as scheduling methods, but fundamental properties of quantum entanglement, as well as technological considerations of quantum hardware capabilities (Section 4.5) call for new protocols and methods of network control and management. Naturally, there is a continuous flow of systems papers proposing new networking architectures, e.g. for SDN [16], data center networks [43], content delivery networks [22] or cloud computing [95], to name a few. Yet, we are unaware of any system-level papers proposing a quantum network stack including protocols for concrete hardware implementations.

3 DESIGN CONSIDERATIONS FOR QUANTUM NETWORK ARCHITECTURES

We first discuss design considerations of quantum networks themselves, followed by considerations specific to the quantum physical and link layers (Section 4). These can be roughly subdivided into three categories: (i) fundamental considerations due to quantum entanglement, (ii) technological limitations of near-term quantum hardware, and (iii) requirements of quantum protocols themselves.

3.1 Qubits and Entanglement

We focus on properties of entanglement as relevant for usage and control (see Appendix, and [67, 87]). Teleportation [11] allows entanglement to be used to send qubits (see Figure 1). We will hence also call two entangled qubits an *entangled link* or *entangled pair*. Teleportation consumes the entangled link, and requires two additional classical bits to be transmitted per qubit teleported. Already at the level of qubit transmission we hence observe the need for a close integration between quantum and classical communication. Specifically, we will need to match quantum data stored in quantum devices with classical control information that is sent over a separate physical medium, akin to optical control plane architectures for classical optical networks [81]. To create long-distance entanglement, we can first attempt to produce short-distance entangled links, and then connect them to form longer distance ones [18, 62] via an operation known as entanglement swapping (see Figure 1). This procedure can be used iteratively to create entanglement along long chains, where we remark that the swapping operations can in principle be performed in parallel. From a resource perspective, we note that to store entanglement, both nodes need to store one qubit per entangled link. Proposals for enabling quantum communication by forward communication using quantum error correction also exist, which avoid entanglement swapping [63]. However, these have arguably much more stringent requirements in terms of hardware, putting them in a technologically more distant future: they require the ability to create entangled states consisting of a large number of photons (only ten realized today [40]) and densely placed repeater stations performing near perfect operations [64].

Producing heralded entanglement does however allow long-distance quantum communication without the need to create entanglement consisting of many qubits. Here, the heralding signal (see Figure 3) provides a confirmation that an entanglement generation attempt has succeeded. Such heralding - i.e. confirmed entanglement - allows techniques using entanglement swapping to enable long-distance quantum communication without exponential overheads [18], and without the need for more complex resources [8, 20]. Creating long-distance links between two controllable nodes by means of entanglement swapping (Section 3.2), and executing complex applications requires both nodes to know the state of their entangled links (which qubits belong to which entangled link, and who holds the other qubit of the entangled pair). As illustrated in Figure 1, remote nodes ("B" in the figure) can change the state of such entangled links ("A" and "C" in the figure). Entanglement is an inherently connected element already at the lowest physical level, whereas classical communication typically proceeds by forward communication that does not require information at both the sender and receiver to be used.

3.2 Quantum Network Devices

We focus on a high level summary of devices in a quantum network without delving into detailed physics (for more details, see [6, 76, 90] and Section 4.4). Qubits can be sent optically through standard telecom fiber using a variety of possible encodings, such as polarization [10, 60], time-bin [17], or absence and presence of a photon [20]. Such qubits can be emitted from quantum nodes [12, 13, 75], but in principle also transferred [52, 66, 75] from fiber into the node's local quantum memory. Present day quantum memories have very

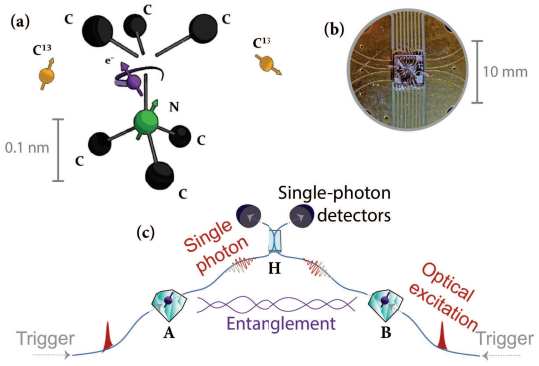


Figure 3: Heralded entanglement generation on the NV platform. (a) NV centers are point defects in diamond with an electronic spin as a communication qubit (purple) and carbon-13 nuclear spins as memory qubits (yellow), realized in custom chips (b). (c) A trigger produces entanglement between the communication qubits of A and B (diamonds) and two qubits (photons) traveling over fiber to the heralding station H. H measures the photons by observing clicks in the left or right detector giving the *heralding signal* s: [failure] (none or both click), [success,$|\Psi^+\rangle$] (left clicks), [success,$|\Psi^-\rangle$] (right clicks). Success confirms one of two types of entangled pairs $|\Psi^+\rangle$ or $|\Psi^-\rangle$ (wiggly purple line). H sends s to A and B (not pictured).

limited lifetimes, making it highly desirable to avoid the exchange of additional control information before the entanglement can be used.

We distinguish two classes of quantum nodes. One, which we will call a *controllable quantum node*, offers the possibility to perform controllable quantum operations as well as storing qubits. Specifically, these nodes enable decision making, e.g. which nodes to connect by entanglement swapping. Such nodes can act as quantum repeaters and decision making routers in the network (e.g. NV platform or other quantum memories combined with auxiliary optics), and—if they support the execution of gates and measurements—function as *end nodes* [90] on which we run applications (e.g. NV centers in diamond or Ion Traps). Others, which we call *automated quantum nodes*, are typically only timing controlled, i.e. they perform the same pre-programmed action in each time step. Such nodes can also support a limited set of quantum operations and measurements, but only those necessary to perform their pre-programmed tasks. Automated nodes are still very useful, for example, to establish entanglement along a chain of quantum repeaters performing the entanglement swapping operations [18, 62] (see again Figure 1). In Section 4.4 we give a concrete example of such a timing controlled element.

3.3 Use Cases

We distinguish five use cases of the stack: one related to producing long-distance entanglement, and four that come from application demands. Since no quantum network has been realized to date, we cannot gain insights from actual usage behavior. Instead we must resort to properties of application protocols known today. Looking into the future, we desire flexibility to serve all use cases, including supporting multiple applications at the same time.

Measure Directly (MD) Use Case: The first application use case comes from application protocols that produce many ($\geq 10^4$) pairs of entangled qubits sequentially, where both qubits are immediately measured to produce classical correlations. As such, no quantum memory is needed to store the entanglement and it is not necessary to produce all entangled pairs at the same time. It follows that applications making use of this use case may tolerate fluctuating delays in entanglement generation. Additionally, it is not essential to deliver error free correlations obtained from entanglement to the application. Such applications will thus already anticipate error fluctuation across the many pairs. This contrasts with classical networking where errors are often corrected before the application layer. Examples of such applications are QKD [37], secure identification [31] and other two-party cryptographic protocols [3, 21, 30, 72, 91] at the prepare-and-measure network stage [90], and device-independent protocols at the entanglement network stage [90].

Create and Keep (CK) Use Case: The second application use case stems from protocols that require genuine entanglement, possibly even multiple entangled pairs to exist simultaneously. Here, we may wish to perform joint operations on multiple qubits, and perform quantum gates that depend on back and forth communication between two nodes while keeping the qubits in local quantum storage. While more applications can be realized with more qubits, this use case differs substantially in that we want to create relatively few (even just one) entangled pairs, but want to store this entanglement. Since we typically want these pairs to be available at the same time, and memory lifetimes are short, we want to avoid delay between producing consecutive pairs, which is superficially similar to constraints in real time classical traffic. Also for CK, many applications can perform well with noisy entangled links and the amount of noise forms a performance metric (fidelity, Section 4.2). Examples of such protocols lie in the domain of sensing [41], metrology [55], and distributed systems [9, 33] which are in the quantum memory network stage and above [90].

Remote State Preparation (RSP) Use Case: For certain application protocols (for example, secure delegated quantum computation [19, 24]), an interpolation between the CK and MD use case can be considered. Here, one of the two qubits is immediately measured as in the MD use case, but the other is stored as in the CK use case. Due to the similarity to the CK use case, we will only distinguish the RSP case in the appendix.

Send Qubit (SQ) Use Case: While many application protocols known to date consume entanglement itself, some—such as distributed quantum computing applications—ask for the transmission of (unknown) qubits. This can be realized using teleportation over any distance as long as entanglement is confirmed between the sender and the receiver. For the quantum link layer, this again does not differ from CK, where we want to produce one entangled pair per qubit to be sent.

Network Layer (NL) Use Case: In analogy to the classical notion of a link layer, we take the quantum link layer to refer to producing entanglement between neighboring nodes (see Section 3.4). The network layer will be responsible for producing entanglement between more distant ones. While usage behavior of quantum networks is unknown, it is expected (due to technological limitations) that routing decisions, i.e. how to form long-distance links from pairwise links, will not be entirely dynamic. One potential approach would

be to first determine a path, and reserve it for some amount of time such that pairwise entanglement can be produced. Producing pairwise entanglement concurrently enables simultaneous entanglement swapping along the entire path with minimal delay to combat limited memory lifetimes. For this, the network layer needs to be capable of prioritizing entanglement production between neighboring nodes.

3.4 Network Stack

Based on these considerations, we propose an initial functional allocation of a quantum network stack (see Figure 2). In analogy to classical networking, we refer to the lowest element of the stack as the physical layer. This layer is realized by the actual quantum hardware devices and physical connections such as fibers. We take the physical layer to contain no decision making elements and keep no state about the production of entanglement (or the transmissions of qubits). The hardware at the physical layer is responsible for timing synchronization and other synchronization, such as laser phase stabilization [47], required to make attempts to produce heralded entanglement (Section 4.4). A typical realization of the physical layer involves two controllable quantum nodes, linked by an (chain of) automated quantum node that attempt entanglement production in well-defined time slots.

The task of the quantum link layer is then to turn the physical layer making entanglement attempts into a robust entanglement generation service, that can produce entanglement between controllable quantum nodes connected by an (chain of) automated quantum node. Requests can be made by higher layers to the link layer to produce entanglement, where robust means that the link layer endows the physical system with additional guarantees: a request for entanglement generation will (eventually) be fulfilled or result in a time-out. This can be achieved by instructing the physical layer to perform many attempts to produce entanglement until success.

Built on top of the link layer rests the network layer, which is responsible for producing long-distance entanglement between nodes that are neither connected directly, nor connected by a chain of automated quantum nodes at the physical layer. This may be achieved by means of entanglement swapping, using the link layer to generate entanglement between neighboring controllable nodes. In addition, it contains an entanglement manager that keeps track of entanglement in the network, and which may choose to pre-generate entanglement to service later requests from higher layers. It is possible that the network layer and entanglement manager may eventually be separated.

To assist the SQ use case, a transport layer takes responsibility for transmitting qubits deterministically (e.g. using teleportation). One may question why this warrants a separate layer, rather than a library. Use of a dedicated layer allows two nodes to pre-share entanglement that is used as applications of the system demand it. Here, entanglement is not assigned to one specific application (purpose ID, Section 4.1.1). This potentially increases the throughput of qubit transmission via teleportation, as teleportation requires no additional connection negotiation, but only forward communication from a sender to the receiver. Implementing such functionality in a library would incur delays in application behavior as entanglement would need to be generated on-demand rather than supplying it from pre-allocated resources.

4 DESIGN CONSIDERATIONS FOR QUANTUM LINK LAYER

4.1 Desired Service

The link layer offers a robust entanglement creation service between a pair of controllable quantum nodes A and B that are connected by a quantum link, which may include automated nodes along the way. This service allows higher layers to operate independently of the underlying hardware platform, depending only on high-level parameters capturing the hardware capabilities.

4.1.1 Requesting entanglement. Our use cases bring specific requirements for such a service. Entanglement creation can be initiated at either A or B by a CREATE request from the higher layer with parameters: (1) *Remote node* with whom entanglement generation is desired if the node is connected directly to multiple others. (2) *Type of request* - create and keep (K), create and measure (M), and remote state preparation (R). The first type of request (K) stores entanglement, addressing the use cases CK and NL (see Section 3.3). The second (M) leads to immediate measurement, supporting the use case MD. The reason for distinguishing these two scenarios is twofold: first, we will show later (Section 4.4) that a higher throughput can for some implementations be achieved for M than for K on the same system. Second, simple photonic quantum hardware without a quantum memory and sophisticated processing capabilities [77] only supports the M mode of operation. In R, a measurement is performed only at one node. Since it behaves like K, we will only expand upon R in the appendix. (3) *Number of entangled pairs to be created.* Allowing the higher layer to request several pairs at once can increase throughput by avoiding additional processing delays due to increased inter-layer communication (as compared to classical networks [57, Table 2]). It also helps the CK use case where an application actually needs several pairs concurrently. (4) *Atomic* is a flag that indicates that the request should be satisfied as a whole without interruption by other requests. (5) *Consecutive* is a flag indicating an OK is returned for each pair made for a request (typical for NL use case). Otherwise, an OK is sent only when the entire request is completed (more common in application use cases). (6) *Waiting time,* $t_{\max}$ (and *time units*) can be used to indicate the maximum time that the higher layer is willing to wait for completion of the request. This allows a general timeout to be set, and enables the NL and CK use case to specify strict requirements since the requested pairs may no longer be desirable if they are delivered too late. (7) A *purpose ID* can be specified which allows the higher layer to tag the entanglement for a specific purpose. For an application use case, this purpose ID may be considered analogous to a port number found in the TCP/IP stack. Including it in the CREATE request allows both nodes to immediately provide the entanglement to the right application and proceed processing without incurring further communication delays. Reducing any additional communication overhead is necessary due to the noisy nature of quantum devices. A purpose ID is also useful to identify entanglement created by the NL use case for a specific long-distance path. We envision that an entanglement manager who may decide to pre-generate entanglement would use a special tag to indicate "ownership" of the requested pairs. For the NL use case for example,

if the entanglement request does not correspond to a pre-agreed path, then the remote node may refuse to engage in entanglement generation. Finally, because quantum resources are scarce, a purpose ID enables rejection of requests from remote nodes based on scheduling or security policies.(8) A *priority* that may be used by a scheduler. Here we use only three priorities in our simulations (use cases NL, MD and CK), but we remark that in the future more fine grained priorities may find use. For now, were merely provision space for such information for traffic engineering purposes. (9) *Random basis choice* to be used for measurements in MD requests. May be used to specify measurements bases that are uniformly sampled from by the local and remote nodes from a set of basis commonly used in QKD (see Appendix). (10) *Measurement basis* for the local and remote nodes should one desire all measurements be performed in a fixed basis. Default is a measurement in the standard basis. Other bases may be specified in terms of rotations around the Bloch sphere axes of a qubit (see appendix). (11) Finally, we allow a specification of a purely quantum parameter (see Appendix), the *desired minimum fidelity*, $F_{\min}$, of the entanglement [67]. Here, it is sufficient to note that the fidelity $0 \leq F \leq 1$ measures the quality of entanglement, where a higher value of F means higher entanglement quality. The ideal target state has $F = 1$, while $F \geq 1/2$ is often desirable [46]. Higher fidelity implies lower quantum bit error rate (QBER), which captures the probability that measurements on the entangled state deviate from the ideal outcomes (see Appendix). The reason for allowing different $F_{\min}$ instead of fixing one for each hardware platform is that the same platform can be used to produce higher or lower fidelity pairs, where a higher fidelity pair costs more time to prepare. An example of this is the use of entanglement distillation [35, 53] where two lower quality pairs are combined into one higher quality one. Another is the choice of bright state population α (see Section 4.4), which can be chosen to trade-off fidelity and throughput. In practice, the necessary minimum fidelity required to execute either long distance entanglement generation or application protocols may be obtained by the requirements for the successful operation of said protocols, and differs significantly across protocols. Such minimum fidelity requirements are typically concluded from an analytical or numerical analysis of such protocols, and are not yet known for many proposed application protocols.

4.1.2 Response to entanglement requests. If entanglement has been produced successfully, an OK message should be returned. In addition, the use cases specified in Section 3.3 desire several other pieces of information, which may also be tracked at higher layer: (1) An entanglement identifier Ent_{ID} unique in the network during the lifetime of the entanglement. This allows both nodes to immediately process the entanglement without requiring an additional round of communication degrading the entanglement due to limited memory lifetimes. (2) A qubit ID for K-type (create and keep) requests which identifies where the local qubit is in the quantum memory device. (3) The "Goodness" G, which for K requests is an estimate (see Appendix) of the fidelity — where $G \geq F_{\min}$ should hold — and for M an estimate of the QBER (see Appendix). (4) The measurement outcome for M type requests. (5) The time of entanglement creation. (6) The time the goodness parameter was established. The goodness may later be updated given fixed information about the underlying hardware platform. Explicit OK messages from the link layer are desired for several reasons which derive from the task of the link layer to turn low probability generation at the physical layer into a robust service: First, before an entanglement swapping or other operation may be performed by the network layer we need to know entanglement has been produced. Second, applications demand knowledge of entanglement identifiers or measurement outcomes to proceed successfully.

Evidently, there are many possibilities of failure resulting in the return of error messages. This includes: (1) Timeout when a request could not be fulfilled in a specific time frame (TIMEOUT). (2) An immediate rejection of the request because the requested fidelity is not achievable in the given time frame (UNSUPP). (3) The quantum storage is permanently (MEMEXCEEDED) or temporarily (OUTOFMEM) too small to simultaneously store all pairs of an atomic request. (4) Refusal by the remote node to participate (DENIED).

Finally, we allow an EXPIRE message to be sent, indicating that the entanglement is no longer available. This in principle can be indicated by a quantum memory manager (see Appendix, Section 5.2.2) instead of the protocol, but we will show that this allows for recovery from unlikely failures.

4.1.3 Fixed hardware parameters. Not included in these request or response messages are parameters that are fixed for the specific hardware platform, or change only very infrequently. As such, these may be obtained by the higher-level software by querying the low level system periodically, similarly to some classical network architectures (e.g. [59]). Such parameters include: (1) The number of available qubits. (2) The qubit memory lifetimes. (3) Possible quantum operations. (4) Attainable fidelities and generation time. (5) The class of states that are produced. The latter refers to the fact that more information about that state than just the fidelity allows optimization at layers above the link layer.

4.2 Performance Metrics

Before designing any protocols that adhere to these requirements, we consider the performance metrics that such protocols may wish to optimize. Standard metrics from networking also apply here, such as *throughput* (entangled pairs/s), and the *latency*. We distinguish between: (1) Latency per request (time between submission of a CREATE request and its successful completion at a requesting node). (2) Latency per pair (time between CREATE and OK at requesting node). (3) Latency per request divided by the number of requested pairs (which we denote as the *scaled latency*). Given that requests may originate at both A and B, we also demand *fairness*, i.e., the metrics should be roughly independent of the origin of the request. Here, we also care about genuinely quantum quality metrics, specifically the fidelity F (at least $F_{\min}$).

The non-quantum reader may wonder about the significance of F, and why we do not simply maximize throughput (e.g. [16, 80]) or minimize latency (e.g. [22, 34]). For instance, QKD (a MD use case as listed in Section 3.3), requires a minimum quantum bit error rate (QBER) between measurement outcomes at A and B (related to F, see Appendix). A lower F results in a larger QBER, allowing less key to be produced per pair. We may thus achieve a higher throughput, but a lower number of key bits per second, or key generation may become impossible.

4.3 Error Detection

Link layer protocols for classical communication typically aim to correct or detect errors, e.g. using a CRC. In principle, there exists an exact analogy at the quantum level: We could use a checksum provided by a quantum error correcting code (QECC) [67, 83] to detect errors. This is technologically challenging and experimental implementations of QECC are in very early stages [26, 27, 74]: to use a QECC for information traveling 5̃km, we would need to create highly entangled quantum states of many qubits, combined with quantum operations of extremely high precision [7]. Yet, apart from technological limitations, future quantum link layer protocols may not use quantum checksums due to different use case requirements. We typically only demand some minimum fidelity F_{min} with high confidence that may also fluctuate slightly for pairs produced within a time window. That is, the applications do not expect all errors to be corrected for them.

As we thus allow imperfect pairs to be delivered to an application, we instead use a different mechanism: we intersperse test rounds during entanglement generation (for details, see Appendix) to verify the quality of the link, by estimating the fidelity of the generated entanglement. Such test rounds are easy to produce without the need for complex gates or extra qubits. Evidently, there exists an exact analogy in the classical networking world, where we would transmit test bits to measure the current quality of transmission, e.g. a direct analogy to network profiling [59] to gain confidence that the non-test bits are also likely to be transmitted with roughly the same amount of error. Yet, there we typically care about correctness of a specific data item, rather than an enabling resource like entanglement.

4.4 Physical Entanglement Generation

Let us now explain how heralded entanglement generation is actually performed between two controllable nodes A and B (see Appendix for details). As an example, we focus on the hardware platform available to us (NV in diamond, Figure 3), but analogous implementations have been performed using remote Ion Traps [61] and Neutral Atoms [45].

In all cases (NV, Ion Trap, Neutral Atom, and others), processing nodes A and B are few-qubit quantum computers, capable of storing and manipulating qubits. They are connected to an intermediate station called the *heralding station* H over optical fibers. This station is a much simpler automated node, built only from linear optical elements. Each node can have two types of qubits: *memory qubits* as a local memory, and *communication qubits* with an optical interface, that can be entangled with a photon. To produce entanglement, a time synchronized trigger is used at both A and B to create entanglement between each communication qubit, and a corresponding traveling qubit (photon). These photons are sent from A and B to H over fiber. When both arrive at H, H performs an automatic entanglement swapping operation which succeeds with some probability. Since H has no quantum memory, both photons must arrive at H at the same time to succeed. Success or failure is then transmitted back from H to the nodes A and B over a standard classical channel (e.g. 100Base-T). In the case of success, one of several entangled states may be produced, which can however be converted to one other using local quantum gates at A or B. The heralding signal is used to indicate which state was produced. After a generation attempt, the communication qubit may be moved to a memory qubit, in order to free the communication qubit to produce the next entangled pair. Many parameters influence the success and quality of this process, such as the quality of the qubits themselves, the probability of emission of a photon given a trigger signal, losses in fiber, and quality of the optical elements such as detectors used at H (Figure 3).

To understand this process in more detail, consider the NV platform (Figure 3) (see e.g. [47] for details on this process, and [23] for an overview of the NV platform in general). Two different schemes for producing entanglement have been implemented, that differ in how the qubits are encoded into photons (time-bin [8], or presence/absence of a photon [20]). While physically different, both of these schemes fit into the framework of our physical and link layer protocols.

To evaluate the performance of the protocol (Section 6) and provide intuition of timings, we compare to data from the setup [47] which uses presence/absence of a photon as encoding. A microwave pulse prepares the communication qubit depending on a parameter α, followed by a laser pulse to trigger photon emission (total duration 5.5μs). A pair ($|\Psi^+\rangle$ or $|\Psi^-\rangle$) is successfully produced with fidelity $F \approx 1 - \alpha$ with probability $p_{\text{succ}} \approx 2\alpha p_{\text{det}}$, where $p_{\text{det}} \ll 1$ is, given that a photon was emitted, the probability of heralding success. The parameter α thus allows a trade-off between the rate of generation (p_{succ}), and the quality metric F. Other factors that impact the fidelity are memory decoherence, detector dark-counts, phase instabiliy, losses, imperfect operations and more (see Appendix). For K type requests, we may store the pair in the communication qubit, or move to a memory qubit (gate duration 1040μs for the qubit considered). The quality of this qubit degrades as we wait for H to reply. For M type requests, we may choose to measure immediately before receiving a reply (here readout takes 3.7μs). Important is the time of an attempt t_{attempt} (time preparing the communication qubit until receiving a reply from H, and completion of any post-processing such as moving to memory), and the maximum attempt rate r_{attempt} (maximum number of attempts that can be performed per second not including waiting for a reply from H or post-processing). The rate r_{attempt} can be larger than $1/t_{\text{attempt}}$: (1) for M the communication qubit is measured before receiving the reply from H and thus allows for multiple attempts to overlap and (2) for K, if the reply from H is failure, then no move to memory is done.

For performance evaluation we consider two physical setups as an example (see Appendix) with additional parameters hereafter referred to as the Lab scenario and the QL2020 scenario. The Lab scenario already realized [47] with 1 m distance to the station from both A and B (communication delay to H negligible), $p_{\text{succ}} \approx \alpha \cdot 10^{-3}$ (F vs. α, Figure 8). For M requests, we act the same for Lab and QL2020 and always measure immediately before parsing the response from H to ease comparison (thus $t_{\text{attempt}} = 1/r_{\text{attempt}} = 10.12$ μs which includes electron readout 3.7 μs, photon emission 5.5 μs and a 10 % extra delay to avoid race conditions). For K requests in Lab, $t_{\text{attempt}} = 1045$ μs but $1/r_{\text{attempt}} \approx 11$ μs as memory qubits need to be periodically initialized (330 μs every 3500 μs). The QL2020 scenario has not been realized and is based on a targeted implementation connecting two Dutch cities by the end of 2020 ($\approx 10km$ from A to H with a communication delay of 48.4μs in fiber, and $\approx 15km$

from B to H with a 72.6μs delay). Frequency conversion of 637nm to 1588nm needs to be performed on the photons emitted in our modeled NV center, where fiber losses at 1588nm are taken to be 0.5 dB/km (values for deployed QL2020 are 0.43-0.47 db/km). We assume the use of optical cavities to enhance photon emission [14, 73] giving a probability of success $p_{\text{succ}} \approx \alpha \cdot 10^{-3}$. F is worse due to increased communication times from H. For QL2020, $t_{\text{attempt}} = 145$ μs for M (trigger, wait for reply from H) and $t_{\text{attempt}} = 1185$ μs for K (trigger, wait for reply from H, swap to carbon). Maximum attempt rates are $1/r_{\text{attempt}} = 10.120$ μs (M) and $1/r_{\text{attempt}} \approx 165$ μs (K).

4.5 Hardware Considerations

Quantum hardware imposes design considerations for any link layer protocol based on top of such experiments for generating entanglement.

Trigger generation: Entanglement can only be produced if both photons arrive at the heralding station at the same time. This means that the low level system requires tight timing control; such control (ns scale) is also required to keep the local qubits stable. This imposes hard real time constraints at the lowest level, with dedicated timing control (AWG) and software running on a dedicated microcontroller (Adwin ProII). We expect that a physical layer protocol built on heralded entanglement without the use of additional quantum memories would operate over distances up to 100km. As such, providing timing synchronization at the required level may be done using existing techniques such as White Rabbit [78]. Timing constraints to perform entanglement swapping over larger distances at higher layers, or using automated nodes with memories are less stringent. When considering a functional allocation between the physical and link layer in the quantum network stack, this motivates taking all timing synchronization to happen at the physical layer. At this layer, we may then also timestamp classical messages traveling to and from H, to form an association between classical control information and entangled pairs.

Scheduling and flow control: Consequently, we make the link layer responsible for all higher level logic, including scheduling, while keeping the physical layer as simple as possible. An example of scheduling other than priorities, is flow control which controls the speed of generation, depending on the availability of memory on the remote node to store such entanglement.

Note that depending on the number of communication qubits, and parallelism of quantum operations that the platforms allows, a node also needs a global scheduler for the entire system and not only the actions of the link layer.

Noise due to generation: One may wonder why one does not continuously trigger entanglement generation locally whenever the node wants a pair, or why one does not continuously produce pairs and then this entanglement is either discarded or otherwise made directly available. In the NV system, triggering entanglement generation causes the memory qubits to degrade faster [51, 71]. As such we would like to achieve agreement between nodes to avoid triggering unless entanglement it is indeed desired.

This consideration also yields a security risk: if an attacker could trick a node into triggering entanglement generation, without a matching request on the other side, this would allow a rapid destruction of contents of the nodes' local quantum memory. For this reason, we want classical communication to be authenticated which can be achieved using standard methods.

Memory allocation: Decisions on which qubits to use for what purpose lies in the domain of higher level logic, where more information is available. We let such decisions be taken by a global quantum memory manager (QMM), which can assist the link layer to make a decision on which qubits to employ. It can also translate logical qubit IDs into physical qubit IDs in case multiple qubits are used to redundantly form one logical storage qubit.

5 PROTOCOLS

We now present our protocols satisfying the requirements and considerations set forth in Sections 3 and 4. The entanglement generation protocol (QEGP) at the link layer, uses the midpoint heralding protocol (MHP) at the physical layer. Classical communication is authenticated, and made reliable using standard methods (e.g. 802.1AE [48], authentication only).

5.1 Physical Layer MHP

Our MHP is a lightweight protocol built directly on top of physical implementations of the form of Section 4.4, supplementing them with some additional control information. With minor modifications this MHP can be adapted to other forms of heralded entanglement generation between controllable nodes, even using multiple automated middle nodes [42].

The MHP is meant to be implemented directly at the lowest level subject to tight timing constraints. Protocol execution is divided into time slots, which are synchronized between the two neighboring nodes (Section 4.4). In each time slot, the MHP polls the higher layer (Figure 4, the link layer QEGP) to determine whether entanglement generation is required in this slot. A batched operation is possible, should the delay incurred by the polling exceed the minimum time to make one entanglement generation attempt - *the MHP cycle* - and hence dominate the throughput. MHP keeps no other state. Upon polling, the higher layer may respond "no" in which case no attempt to produce entanglement will be made or with "yes", additionally providing parameters to use in the attempt. These parameters include the type of request (M, measure) or (K, store) passed on from the higher layer, for which the MHP takes the following actions.

5.1.1 Protocol for Create and Keep (K). The parameters given to the MHP with a "yes" response contain the following: (1) An ID for the attempt that is forwarded to H, (2) Generation parameters (α, Section 4.4), (3) The device qubits for storing the entanglement, (4) A sequence of operations to perform on the device memory [1]. The higher layer may instruct the MHP to perform a gate on the communication qubit depending on the heralding signal from H allowing the conversion from the $|\Psi^-\rangle$ state to the $|\Psi^+\rangle$ state, before returning completion to the higher layer. Entanglement generation is then triggered at the start of the next time interval, using the generation parameter α, and a GEN message is sent to H which includes a timestamp, and the given ID. The motivation for including the ID is to protect against errors in the classical control, for example losses.

[1] Less abstractly, by specifying microwave and laser pulse sequences controlling the chip (see Appendix).

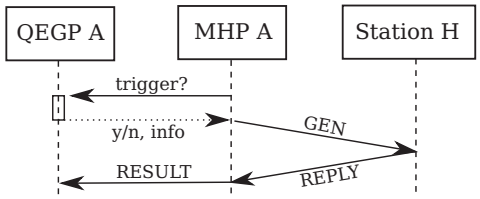


Figure 4: Timeline of the MHP polling higher layers to see if entanglement should be produced.

The station H uses the timestamp to link the message to a detection window in which the accompanying photons arrived. Should messages from both nodes arrive, the midpoint verifies that the IDs transmitted with the GEN messages match, and checks the detection counts (Figure 3) from the corresponding detection window. The midpoint will then send a REPLY message indicating success or failure, and in the case of success, which of the two states, $|\Psi^+\rangle$ and $|\Psi^-\rangle$, was produced. The REPLY additionally contains a sequence number uniquely identifying the generated pair of entangled qubits chosen by H, which later enables the QEGP to assign unique entanglement identifiers. This REPLY and the ID is forwarded to the link layer for post-processing. Note that the REPLY may be received many MHP cycles later, allowing the potential for emission multiplexing (Section 5.2).

5.1.2 Protocol for Create and Measure (M). Handling M type requests is very similar, differing only in two ways: Instead of performing a gate on the communication qubit, the "yes" message requests the MHP to perform a measurement on the communication qubit in a specified basis once the photon has been emitted, even before receiving the response from H. The outcome of the measurement and the REPLY are passed back to the QEGP. In practice, the communication time from transmitting a GEN message to receiving a REPLY may currently exceed the duration of such a local measurement (3.7 μs vs. communication delay Lab 9.7 ns, and QL2020 145 μs). The MHP may thus choose to perform the measurement immediately (communication delay exceeds measurement delay), or only after receiving the response (measurement delay exceeds communication delay).

5.2 Link Layer QEGP

Here we present an implementation of a link layer protocol, dubbed QEGP (quantum entanglement generation protocol), satisfying the service requirements put forth in Section 4 (see Appendix for details and message formats). We build up this protocol from different components:

5.2.1 Distributed Queue. Both nodes that wish to establish entangled link(s) must trigger their MHP devices in a coordinated fashion (Section 4.4). To achieve such agreement, the QEGP employs a distributed queue comprised of synchronized local queues at the controllable nodes. These local queues can be used to separate requests based on priority, where here we employ 3 queues for the different use cases (CK, NL, MD). Due to low errors in classical communication (estimated $< 4\times10^{-8}$ on QL2020, see Appendix), we let one node hold the master copy of the queue, and use a simple two-way handshake for enqueing items, and a windowing mechanism to ensure fairness. Queue items include a *min_time* that specifies the earliest possible time a request is deemed ready for processing by both nodes (depending on their distance). Specifying *min_time* prevents either node from beginning entanglement generation in different timesteps. We note that while the distributed queue requires timing synchronization for such functionality, the timing constraints are looser than those found at the physical layer. Hence, sufficient synchronization may be obtained by piggy-backing on the mechanisms used at the physical layer, or by using PTP [79].

One may wonder why we employ a distributed queue to coordinate entanglement rather than utilizing classical discussion after entanglement has been generated. Recall from Section 4.5 that the memory lifetimes of qubits are very short. By agreeing on coordination in advance, we reduce the amount of noise introduced into the qubits before they are used by applications. An alternative design choice worthwhile exploring would be to employ the heralding midpoint as the master of the distributed queue. Such a construction may allow coordination of entanglement generation between several endnodes connected to a common midpoint station.

5.2.2 Quantum Memory Management (QMM). The QEGP uses the node's QMM (Section 4.5) to determine which physical qubits to use for generating or storing entanglement.

5.2.3 Fidelity Estimation Unit (FEU). In order to provide information about the quality of entanglement, the QEGP employs a fidelity estimation unit. This unit is given a desired quality parameter F_{min}, and returns generation parameters (such as α) along with an estimated minimal completion time. Such a fidelity estimate can be calculated based on known hardware capabilities such as the quality of the memory and operations. To further improve this base estimate the QEGP intersperses test rounds.

5.2.4 Physical Translation Unit (PTU). The link layer protocol processes CREATE requests in a hardware-independent manner. To resolve physical gate instructions that must be provided to the MHP and underlying platform, a physical translation unit that converts hardware-independent instruction descriptions into hardware-dependent instructions is used. For example, the PTU may convert the Euler decomposition of a single-qubit gate or a pair of physical qubit ids for a two-qubit gate (such as moving the state of one to the other) into a sequence of physical instructions that should be issued to the hardware below. This unit also converts entanglement generation parameters like α supplied by the FEU into the corresponding physical instruction (here, a specific microwave pulse).

5.2.5 Scheduler. The QEGP scheduler decides which request in the queue should be served next. In principle, any scheduling strategy is suitable as long as it is deterministic, ensuring that both nodes select the same request locally. This limits two-way communication, which adversely affects entanglement quality due to limited memory lifetimes.

5.2.6 Protocol. Figure 5 presents an architecture diagram visualizing the operation. The protocol begins when a higher layer at a controllable node issues a CREATE operation to the QEGP specifying a desired number of entangled pairs along with F_{min} and t_{max} (Section 4.1.1). Upon receipt of a request the QEGP will query the FEU to obtain hardware parameters (α), and a minimum completion time (depending on α). If this time is larger than t_{max}, the QEGP immediately rejects the request (UNSUPP). Should the request pass this evaluation, the local node will compute a fitting *min_time* specifying the earliest MHP polling cycle the request may begin processing. The node then adds the request into the distributed

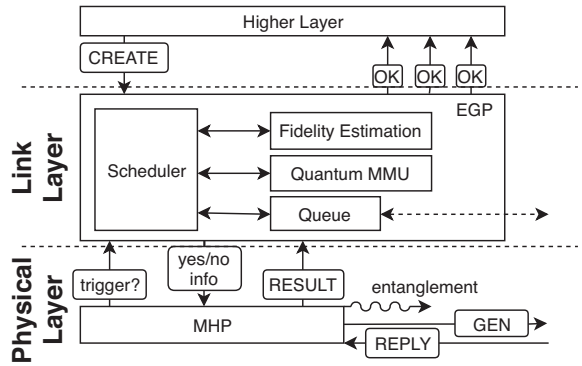


Figure 5: Flow diagram of the MHP and QEGP operation. The QEGP handles CREATE requests and schedules entanglement generation attempts are issued to the MHP. Replies from the midpoint are parsed and forwarded to the QEGP from request management.

queue shared by the nodes. This request may be rejected by the peer should the remote node have queue rules that do not accept the specified purpose ID. Then, the QEGP locally rejects the request (DENIED).

The local scheduler selects the next request to be processed, given that there exists a ready one (as indicated by *min_time*). The QMM is then used to allocate qubits needed to fulfill the specified request type (create and keep K or create and measure M). The QEGP will then again ask the FEU to obtain a current parameter α due to possible fluctuations in hardware parameters during the time spent in the queue. The scheduler then constructs a "yes" response to the MHP containing α from the FEU, along with an ID containing the unique queue ID of the request in the distributed queue, and number of pairs already produced for the request. This response is then forwarded to the local MHP upon its next poll to the QEGP. If no request is ready for processing, a "no" response is returned to the MHP . At this point the MHP behaves as described in the previous section and an attempt at generating entanglement is made.

Whenever a REPLY and ID is received from the MHP, the QEGP uses the ID to match the REPLY to an outstanding request, and evaluates the REPLY for correctness. Should the attempt be successful, the number of outstanding pairs in the request is decremented, and an OK message is propagated to higher layers containing the information specified in Section 4.1.2, where the Goodness is obtained from the FEU.

In the Appendix, we consider a number of examples to illustrate decisions and possible pitfalls in the QEGP. One such example is the possibility of *emission multiplexing [85]:* The QEGP can be polled by the MHP before receiving a response from the MHP for the previous cycle. This allows the choice to attempt entanglement generation multiple times in succession before receiving a reply from the midpoint, e.g., in order to increase the throughput for the MD use case. Errors such as losses on the classical control link can lead to an inconsistency of state (of the distributed queue) at *A* and *B* from which we need to recover. Inconsistencies can also affect the higher layer, e.g. with node *A* issuing an OK to higher layer, but not node *B*. Since the probability of e.g. losses is extremely low, we choose not to perform additional two-way discussion to further curb all inconsistencies at the link layer. Instead, the QEGP can issue an EXPIRE message for an OK already issued if inconsistency is detected later, e.g. when the remote node never received an OK for this pair.

6 EVALUATION

We investigate the performance of our link layer protocol using a purpose built discrete event simulator for quantum networks (NetSquid [70], Python/C++) based on DynAA [39] (see Appendix for details and more simulation results). Both the MHP and QEGP are implemented in full in Python, running on simulated nodes that have simulated versions of the quantum hardware components, fiber connections, etc. All simulations were performed on the supercomputer *Cartesius* at SURFsara [82], in a total of 2578 separate runs using a total of 94244 core hours, and 707 hours time in the simulation (~250 billion MHP cycles). One simulated second currently takes about two core minutes on average, since in each entanglement generation attempt (every 10.12 μs for type MD) multiple events are scheduled and handled and the 16×16-matrix representing the state of the two photons and electrons is updated based on multiple sources of noise and gate operations. The code used for the simulation can be found at [28] and complete data at [29].

We conduct the following simulation runs:

- Long runs: To study robustness of our protocol, we simulate the 169 scenarios described below for an extended period of time. Each scenario was simulated twice for 120 wall time hours, simulating 502 – 13437 seconds. We present and analyze the data from these runs in sections 6.1, 6.2 and the Appendix.
- Short runs: We perform the following simulations for a shorter period of time (24 wall time hours, reaching 67 – 2356 simulated seconds):
  - Performance trade-offs: To study the trade-off between latency, throughput and fidelity we sweep the incoming request frequency and the requested minimum fidelity, see Figure 6.
  - Metric fluctuations: To be able to study the impact of different scheduling strategies on the performance metrics, we run 4 scenarios, 102 times each. The outcomes of theses simulation runs are discussed in section 6.3.

To explore the performance at both short and long distances, the underlying hardware model is based on the Lab and QL2020 scenarios, where we validate the physical modeling of the simulation against data collected from the quantum hardware system of the Lab scenario already realized (Figure 8). For the quantum reader we note that while our simulations can also be used to predict behavior of physical implementations (such as QL2020), the focus here is on the performance and behavior of the link layer protocol.

We structure the evaluation along the three different use cases (NL, CK, MD), leading to a total of 169 different simulation scenarios. First, we use different kinds of requests: (1) *NL* (K type request, consecutive flag, priority 1 (highest), store qubit in memory qubit), (2) *CK*, an application asking for one or more long-lived pairs (K type request, immediate return flag, priority 2 (high), store qubit in memory qubit) and (3) (*MD*) measuring directly (M type request, consecutive flag, priority 3 (lowest)). For an application such as QKD, one would not set the immediate return flag in practice for efficiency, but we do so here to ease comparison to the other two

scenarios. Measurements in *MD* are performed in randomly chosen bases X, Z and Y (see Appendix).

In each MHP cycle, we randomly issue a new CREATE request for a random number of pairs k (max $k_{\max}$), and random use case $P \in \{NL, CK, MD\}$ with probability $f_P \cdot p_{\text{succ}}/(E \cdot k)$, where p_{succ} is the probability of one attempt succeeding (Section 4.4), f_P is a fraction determining the load in our system of kind P, and E is the expected number of MHP cycles to make one attempt ($E = 1$ for MD and $E \approx 1.1$ for NL/CK in Lab due to memory re-initialization and post-processing. $E \approx 16$ for NL/CK in QL2020 due to classical communication delays with H ($145\mu s$)). In the long runs, we first study single kinds of requests (only one of MD/CK/NL), with $f_P = 0.7$ (Low), 0.99 (High) or 1.5 (Ultra). For the long runs, we fix one target fidelity $F_{\min} = 0.64$ to ease comparison. For each of the 3 kinds (MD/CK/NL), we examine (1) $k_{\max} = 1$, (2) $k_{\max} = 3$, and (3) only for *MD*, $k_{\max} = 255$. For Ultra the number of outstanding requests intentionally grows until the queue is full (max 256), to study an overload of our system. To study fairness, we take 3 cases of CREATE origin for each single kind (MD/CK/NL) scenario: (1) all from A (master of the distributed queue), (2) all from B, (3) A or B are randomly chosen with equal probability. To examine scheduling, we additionally consider long runs with mixed kinds of requests (Appendix, e.g. Figure 7).

6.1 Robustness

To study robustness, we artificially increase the probability of losing classical control messages (100 Base T on QL2020 fiber $< 4 \times 10^{-8}$ (see Appendix)), which can lead to an inconsistency of state of the QEGP, but also higher layers (Section 5.2). We ramp up loss probabilities up to 10^{-4} (see Appendix) and observe our recovery mechanisms work to ensure stable execution in all cases (35 runs, 281 - 3973 s simulated time), with only small impact to the performance parameters (maximum relative differences [2] to the case of no losses, fidelity (0.005), throughput (0.027), latency (0.629), number of OKs (0.026) with no EXPIRE messages). We see a relatively large difference for latency, which may however be due to latency not reaching steady state during the simulation (70×70 core hours).

6.2 Performance Metrics

We first consider runs including only a single kind of request (MD/CK/NL). In addition to the long runs, we conduct specific short runs examining the trade-off between latency and throughput for fixed target fidelity $F_{\min}$ (Figure 6(a)), and the trade-off between latency (throughput) and the target fidelity in Figure 6(b) (Figure 6(c)). As described in section 4.4, the probability of successful entanglement generation, and therefore throughput, is directly proportional to one minus fidelity of the generated pair.

Below we present the metrics extracted from the long runs with single kinds of requests:

Fidelity: As a benchmark, we began by recording the average fidelity F_{avg} in all 169 scenarios with fixed minimum fidelity. We observe that F_{avg} is independent of the other metrics but does depend on the distance, and whether we store or measure: $0.745 < F_{\text{avg}} < 0.757$ (*NL/CK* Lab), $0.626 < F_{\text{avg}} < 0.653$ (*NL/CK* QL2020), $0.709 < F_{\text{avg}} < 0.779$ (*MD* Lab), $0.723 < F_{\text{avg}} < 0.767$ (*MD* QL2020)

[2] Relative difference between m_1 and m_2 is $|m_1 - m_2| / \max(|m_1|, |m_2|)$

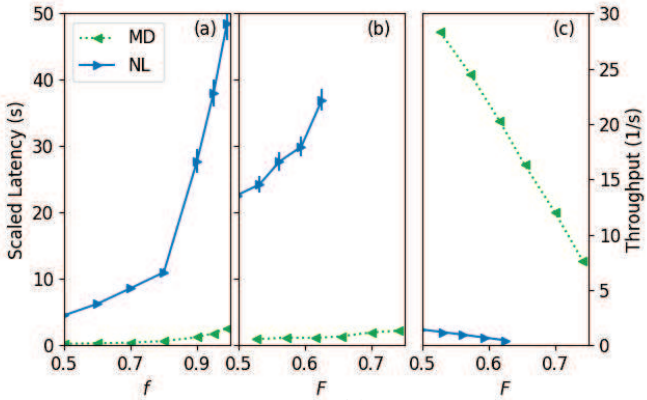


Figure 6: Performance trade-offs. (a) Scaled latency vs. f_P determining fraction of throughput (b) Scaled latency vs. fidelity F_{min}. Demanding a higher F_{min} lowers the probability of success (Section 4.4), meaning (c) throughput directly scales with F_{min} (each point averaged over 40 short runs each 24 h, 93 − 2355 s simulated time, QL2020, $k_{\max} = 3$, for (b,c) $f_P = 0.99$). Higher F_{min} not possible for NL in (b).

(Fidelity *MD* extracted from QBER measurements, see Appendix). This is to be expected since (1) we fix one $F_{\min}$ and (2) we consider an NV platform with only 1 available memory qubit so no change in quality is observed by using different memory qubits (Lab).

Throughput: All scenarios High and Ultra in Lab reach an average throughput th_{avg} (1/s) of $6.05 < th_{\text{avg}} < 6.47$ *NL/CK* and $6.51 < th_{\text{avg}} < 7.09$ for *MD*. It is expected that *MD* has higher throughput, since no memory qubit needs to be initialized. The time to move to memory ($1040\mu s$) is less significant since many MHP cycles are needed to produce one pair, but we only move once. As expected for Low the throughput is slightly lower in all cases, $4.44 < th_{\text{avg}} < 4.72$ *NL/CK*, and $4.86 < th_{\text{avg}} < 5.22$ *MD*. For QL2020, the throughput for *NL/CK* is about 14 times lower, since we need to wait ($145\mu s$) for a reply from H before MHP can make a new attempt.

Latency: The scaled latency highly depends on the incoming request frequency as the longer queue causes higher latency. However, from running the same scenarios many (102) times for a shorter period (24 wall time hours) (see Section 6.3), we see that the average scaled latency fluctuates a lot, with a standard deviation of up to 6.6 s in some cases. For QL2020 with *NL* requests specifying 1-3 pairs from both nodes, we observe an average scaled latency of 10.97 s Low, 142.9 s High and 521.5 s Ultra. For *MD* requests, 0.544 s Low, 3.318 s High and 32.34 s Ultra. The longer scaled latency for *NL* is largely due to longer time needed to create a pair, and not that the queues are longer (average queue length for *NL*: 3.83 Low, 56.3 High, 214 Ultra), and for *MD*: 3.23 Low, 22.4 High and 219 Ultra).

Fairness: For 103 scenarios of the long runs (single kinds of requests (MD/CK/NL) randomly from A and B), we see only slight differences in fidelity, throughput or latency between requests from A and B. Maximum relative differences do not exceed: fidelity 0.033, throughput 0.100, latency 0.073, number of OKs 0.100 (for Ultra).

6.3 Scheduling

We take a first step studying the effect of scheduling strategies on the performance when using mixed kinds of requests. Part of simulating the performance of a scheduling strategies can certainly

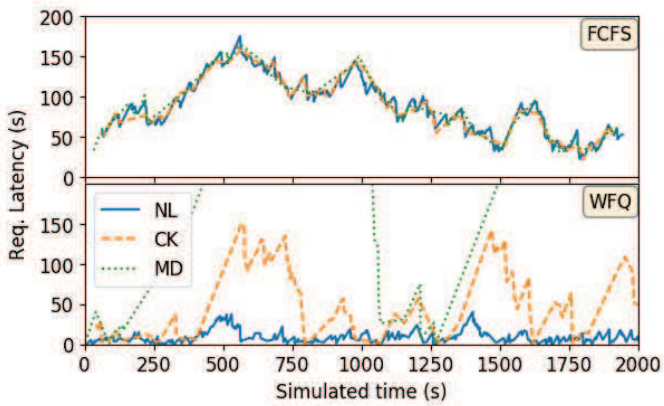


Figure 7: Request latency vs. time for two scheduling scenarios (long runs simulated 120 h wall time). As expected the max. latency for *NL* is decreased due to strict priority. In this scenario, there are more incoming *NL* requests ($f_{NL} = 0.99 \cdot 4/5$, $f_{CK} = 0.99 \cdot 1/5$ and $f_{MD} = 0.99 \cdot 1/5$).

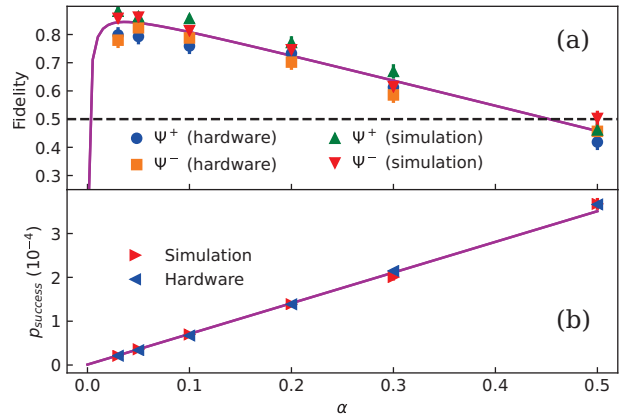


Figure 8: Validation against data from NV hardware (Lab scenario). Fidelity (a) and probability an attempt succeeds (b) in terms of α (Section 4.4) shows good agreement between hardware and simulation points (each at least 300 pairs averaged, 5s–117s simulated time, 500k–10.000k attempts, 122 hours wall time). Theoretical model [47] as visual guide (solid line).

be done without implementing all details of the physical entanglement generation. However, since we do simulate these details we can first confirm that different scheduling strategies below do not affect the average fidelity in these scenarios. Here, we examine two simple scheduling strategies: (1) first-come-first-serve (FCFS) and (2) a strategy where *NL* (priority 1) has a strict highest priority, and use a weighted fair queue (WFQ) for *CK* (priority 2) and *MD* (priority 3), where *CK* has 10 times the weight of *MD*. With these scheduling strategies, we simulate two different request patterns ((i) uniform and (ii) no *NL* more *MD*), 102 times over 24 wall time hours each and extract the performance metrics of throughput and scaled request latency (Table 1).

As expected we see a drastic decrease of the average scaled latency for *NL* when giving it strict priority: 10.3 s with FCFS and 3.5 s with WFQ. For *CK* there is similarly a decrease in average scaled latency, however smaller than for *NL*, of 10.1 s (FCFS) and 6.5 s (WFQ). For *MD* the average scaled latency goes up in both cases when using WFQ instead of FCFS, by factors of 2.49 (uniform) and 1.28 (no *NL* more *MD*).

Table 1: Throughput (T) and scaled latency (SL) using scheduling strategies FCFS and WFQ for two request patterns: (i) with $f_{NL} = f_{CK} = f_{MD} = 0.99 \cdot 1/3$, i.e. a uniform load of the different priorities and (ii) with $f_{NL} = 0$, $f_{CK} = 0.99 \cdot 1/5$ and $f_{MD} = 0.99 \cdot 4/5$, i.e. no *NL* and more *MD*. The physical setup: QL2020 and number of pairs per request: 2 (*NL*), 2 (*CK*), and 10 (*MD*). Each value average over 102 short runs each 24 h, with standard error in parentheses.

T (1/s)	*NL*	*CK*	*MD*
(i) FCFS	0.146 (0.003)	0.144 (0.003)	2.464 (0.056)
(i) WFQ	0.154 (0.003)	0.156 (0.003)	2.130 (0.063)
(ii) FCFS	-	0.086 (0.003)	5.912 (0.033)
(ii) WFQ	-	0.096 (0.003)	5.829 (0.049)

SL (s)	*NL*	*CK*	*MD*
(i) FCFS	10.272 (0.654)	10.063 (0.631)	1.740 (0.120)
(i) WFQ	3.520 (0.085)	6.548 (0.361)	4.331 (0.336)
(ii) FCFS	-	5.659 (0.313)	0.935 (0.062)
(ii) WFQ	-	2.503 (0.100)	1.194 (0.093)

We observe that the throughput gets less affected by the scheduling strategy than the latency for these scenarios. The maximal difference between the throughput for FCFS and WFQ is by a factor of 1.16 (for *MD* in the scenario of no *NL* and more *MD*). Furthermore, we see that the total throughput for all requests goes down from 2.75 (5.99) 1/s for FCFS to 2.44 (5.92) 1/s for WFQ in the case of uniform (no *NL* more *MD*).

7 CONCLUSION

Our top down inventory of design requirements, combined with a bottom up approach based on actual quantum hardware allowed us to take quantum networks a step further on the long path towards their large-scale realization. Our work readies QL2020, and paves the way towards the next step, a robust network layer control protocol. The link layer may now be used as a robust service without detailed knowledge of the physics of the devices. Due to the relatively small size of initial quantum networks, close attention was paid to application use cases even at the link layer. We expect that in the future, the network layer will have a similar interface to higher layers as the link layer itself, and nodes internal to the network will not run applications themselves. Scheduling strategies catering to different use cases may at this stage be applied primarily at the network layer at the level of long-distance links, which are then directly passed to applications running at the end nodes requesting long-distance entanglement. We expect that at the network layer, and when considering larger quantum memories, smart scheduling strategies will be important not only to combat memory lifetimes but also to coordinate actions of different nodes in time, calling for significant effort in computer science and engineering.

ACKNOWLEDGEMENTS

We thank Kenneth Goodenough for comments on earlier drafts. This work was supported by ERC Starting Grant (SW), ERC Consolidator Grant (RH), EU Flagship on Quantum Technologies, Quantum Internet Alliance (No. 820445), NWO VIDI (SW), and Marie Skłodowska-Curie action Spin-NANO (No. 676108). The research in this paper poses no ethical issues.

REFERENCES

[1] Mohamed H. Abobeih, Julia Cramer, Michiel A. Bakker, Norbert Kalb, Matthew Markham, Daniel J. Twitchen, and Tim H. Taminiau. 2018. One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment. *Nature Communications* 9, 1 (Dec 2018), 2552. https://doi.org/10.1038/s41467-018-04916-z arXiv:1801.01196

[2] Alejandro Aguado, Emilio Hugues-Salas, Paul Anthony Haigh, Jaume Marhuenda, Alasdair B. Price, Philip Sibson, Jake E. Kennard, Chris Erven, John G. Rarity, Mark Gerard Thompson, Andrew Lord, Reza Nejabati, and Dimitra Simeonidou. 2017. Secure NFV Orchestration Over an SDN-Controlled Optical Network With Time-Shared Quantum Key Distribution Resources. *J. Lightwave Technol.* 35, 8 (Apr 2017), 1357–1362. http://jlt.osa.org/abstract.cfm?URI=jlt-35-8-1357

[3] Dorit Aharonov, Amnon Ta-Shma, Umesh V. Vazirani, and Andrew C. Yao. 2000. Quantum bit escrow. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing (STOC '00)*. ACM, New York, NY, USA, 705–714. https://doi.org/10.1145/335305.335404

[4] Romain Alléaume, Cyril Branciard, Jan Bouda, Thierry Debuisschert, Mehrdad Dianati, Nicolas Gisin, Mark Godfrey, Philippe Grangier, Thomas Langer, Norbert Lutkenhaus, Christian Monyk, Philippe Painchault, Momtchil Peev, Andreas Poppe, Thomas Pornin, John Rarity, Renato Renner, Gregoire Ribordy, Michel Riguidel, Louis Salvail, Andrew Shields, Harald Weinfurter, and Anton Zeilinger. 2014. Using Quantum Key Distribution for Cryptographic Purposes: a Survey. *Theoretical Computer Science* 560 (2014), 62–81.

[5] Luciano Aparicio, Rodney Van Meter, and Hiroshi Esaki. 2011. Protocol Design for Quantum Repeater Networks. In *Proceedings of the 7th Asian Internet Engineering Conference (AINTEC '11)*. ACM, New York, NY, USA, 73–80. https://doi.org/10.1145/2089016.2089029

[6] David D Awschalom, Ronald Hanson, Jörg Wrachtrup, and Brian B Zhou. 2018. Quantum technologies with optically interfaced solid-state spins. *Nature Photonics* 12, 9 (2018), 516.

[7] Koji Azuma, Kiyoshi Tamaki, and Hoi-Kwong Lo. 2015. All-photonic quantum repeaters. *Nature Communications* 6 (2015), 6787.

[8] Sean D Barrett and Pieter Kok. 2005. Efficient high-fidelity quantum computation using matter qubits and linear optics. *Physical Review A* 71, 6 (2005), 060310.

[9] Michael Ben-Or and Avinatan Hassidim. 2005. Fast Quantum Byzantine Agreement. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing (STOC '05)*. ACM, New York, NY, USA, 481–485. https://doi.org/10.1145/1060590.1060662

[10] Charles H. Bennett and Gilles Brassard. 2014. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.* 560 (2014), 7–11.

[11] Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. 1993. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters* 70, 13 (1993), 1895.

[12] Hannes Bernien, Bas Hensen, Wolfgang Pfaff, Gerwin Koolstra, Machiel S. Blok, Lucio Robledo, Tim Taminiau, Matthew Markham, Daniel J. Twitchen, Lilian Childress, and Ronald Hanson. 2013. Heralded entanglement between solid-state qubits separated by three metres. *Nature* 497, 7447 (2013), 86.

[13] Boris B. Blinov, David L. Moehring, Luming Duan, and Christopher Monroe. 2004. Observation of entanglement between a single trapped atom and a single photon. *Nature* 428, 6979 (2004), 153.

[14] Stefan Bogdanović, Suzanne B van Dam, Cristian Bonato, Lisanne C Coenen, Anne-Marije J Zwerver, Bas Hensen, Madelaine SZ Liddy, Thomas Fink, Andreas Reiserer, Marko Lončar, and Ronald Hanson. 2017. Design and low-temperature characterization of a tunable microcavity for diamond-based quantum networks. *Applied Physics Letters* 110, 17 (2017), 171103.

[15] Conor E. Bradley, Joe Randall, Mohamed H. Abobeih, Remon Berrevoets, Maarten Degen, Michiel A. Bakker, Raymond F. L. Vermeulen, Matthew Markham, Daniel J. Twitchen, and Tim H. Taminiau. 2019. A 10-qubit solid-state spin register with quantum memory up to one minute. (May 2019). arXiv:quant-ph/1905.02094 https://arxiv.org/pdf/1905.02094.pdf

[16] Anat Bremler-Barr, Yotam Harchol, and David Hay. 2016. OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions. In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*. ACM, New York, NY, USA, 511–524. https://doi.org/10.1145/2934872.2934875

[17] Jürgen Brendel, Nicolas Gisin, Wolfgang Tittel, and Hugo Zbinden. 1999. Pulsed energy-time entangled twin-photon source for quantum communication. *Physical Review Letters* 82, 12 (1999), 2594.

[18] Hans J. Briegel, Wolfgang Dür, Juan I. Cirac, and Peter Zoller. 1998. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters* 81, 26 (1998), 5932.

[19] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. 2009. Universal Blind Quantum Computation. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '09)*. IEEE Computer Society, Washington, DC, USA, 517–526. https://doi.org/10.1109/FOCS.2009.36

[20] Carlos Cabrillo, Juan I. Cirac, Pablo Garcia-Fernandez, and Peter Zoller. 1999. Creation of entangled states of distant atoms by interference. *Physical Review A* 59, 2 (1999), 1025.

[21] Andre Chailloux and Iordanis Kerenidis. 2011. Optimal Bounds for Quantum Bit Commitment. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science (FOCS '11)*. IEEE Computer Society, Washington, DC, USA, 354–362. https://doi.org/10.1109/FOCS.2011.42

[22] Fangfei Chen, Ramesh K. Sitaraman, and Marcelo Torres. 2015. End-User Mapping: Next Generation Request Routing for Content Delivery. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, New York, NY, USA, 167–181. https://doi.org/10.1145/2785956.2787500

[23] Lilian Childress and Ronald Hanson. 2013. Diamond NV centers for quantum computing and quantum networks. *MRS Bulletin* 38, 2 (2013), 134–138. https://doi.org/10.1557/mrs.2013.20

[24] Andrew M. Childs. 2005. Secure Assisted Quantum Computation. *Quantum Info. Comput.* 5, 6 (Sep 2005), 456–466. http://dl.acm.org/citation.cfm?id=2011670.2011674

[25] Chin-Wen Chou, Hugues de Riedmatten, Daniel Felinto, Sergey V. Polyakov, Steven J. Van Enk, and Harry Jeffrey Kimble. 2005. Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature* 438, 7069 (2005), 828.

[26] Antonio D Córcoles, Easwar Magesan, Srikanth J Srinivasan, Andrew W Cross, Matthias Steffen, Jay M Gambetta, and Jerry M Chow. 2015. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature communications* 6 (2015), 6979.

[27] Julia Cramer, Norbert Kalb, M Adriaan Rol, Bas Hensen, Machiel S Blok, Matthew Markham, Daniel J Twitchen, Ronald Hanson, and Tim H Taminiau. 2016. Repeated quantum error correction on a continuously encoded qubit by real-time feedback. *Nature communications* 7 (2016), 11526.

[28] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawłczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. 2019. Code used in simulations. https://github.com/SoftwareQuTech/QLinkLayerSimulations. (2019).

[29] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawłczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. 2019. Data from simulations. https://dataverse.nl/dataverse/QLinkLayer. (2019).

[30] Ivan B Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. 2008. Cryptography in the bounded-quantum-storage model. *SIAM J. Comput.* 37, 6 (2008), 1865–1890.

[31] Ivan B Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. 2014. Secure identification and QKD in the bounded-quantum-storage model. *Theoretical Computer Science* 560 (2014), 12 – 26. https://doi.org/10.1016/j.tcs.2014.09.014

[32] Aymeric Delteil, Zhe Sun, Wei-bo Gao, Emre Togan, Stefan Faelt, and Ataç Imamoğlu. 2016. Generation of heralded entanglement between distant hole spins. *Nature Physics* 12, 3 (March 2016), 218–223. https://doi.org/10.1038/nphys3605 arXiv:1507.00465

[33] Vasil S Denchev and Gopal Pandurangan. 2008. Distributed quantum computing: A new frontier in distributed systems or science fiction? *ACM SIGACT News* 39, 3 (2008), 77–95.

[34] Fahad R. Dogar, Thomas Karagiannis, Hitesh Ballani, and Antony Rowstron. 2014. Decentralized Task-aware Scheduling for Data Center Networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 431–442. https://doi.org/10.1145/2619239.2626322

[35] Wolfgang Dür and Hans J Briegel. 2007. Entanglement purification and quantum error correction. *Reports on Progress in Physics* 70, 8 (2007), 1381.

[36] James F. Dynes, Hiroki Takesue, Zhiliang L. Yuan, Andrew W. Sharpe, Ken-Ichi Harada, Toshimori Honjo, Hidehiko Kamada, Osamu Tadanaga, Yoshiki Nishida, Masaki Asobe, and Andrew J. Shields. 2009. Efficient entanglement distribution over 200 kilometers. *Optics express* 17, 14 (2009), 11440–11449.

[37] Artur K. Ekert. 1991. Quantum cryptography based on Bell's theorem. *Physical Review Letters* 67, 6 (1991), 661.

[38] Chip Elliott, David Pearson, and Gregory Troxel. 2003. Quantum Cryptography in Practice. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*. ACM, New York, NY, USA, 227–238. https://doi.org/10.1145/863955.863982

[39] Julio Filho, Zoltan Papp, Relja Djapic, and Job Oostveen. 2013. Model-based Design of Self-adapting Networked Signal Processing Systems. In *Proc. SASO*. IEEE, Philadelphia, PA, USA, 41–50. https://doi.org/10.1109/SASO.2013.16

[40] Wei-Bo Gao, Chao-Yang Lu, Xing-Can Yao, Ping Xu, Otfried Gühne, Alexander Goebel, Yu-Ao Chen, Cheng-Zhi Peng, Zeng-Bing Chen, and Jian-Wei Pan. 2010. Experimental demonstration of a hyper-entangled ten-qubit Schrödinger cat state. *Nature physics* 6, 5 (2010), 331.

[41] Daniel Gottesman, Thomas Jennewein, and Sarah Croke. 2012. Longer-baseline telescopes using quantum repeaters. *Physical Review Letters* 109, 7 (2012), 070503.

[42] Saikat Guha, Hari Krovi, Christopher A Fuchs, Zachary Dutton, Joshua A Slater, Christoph Simon, and Wolfgang Tittel. 2015. Rate-loss analysis of an efficient quantum repeater architecture. *Physical Review A* 92, 2 (2015), 022357.

[43] Mark Handley, Costin Raiciu, Alexandru Agache, Andrei Voinescu, Andrew W. Moore, Gianni Antichi, and Marcin Wójcik. 2017. Re-architecting Datacenter Networks and Stacks for Low Latency and High Performance. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17)*. ACM, New York, NY, USA, 29–42. https://doi.org/10.1145/3098822.3098825

[44] Bas Hensen, Hannes Bernien, Aanaïs E. Dréau, Andreas Reiserer, Norbert Kalb, Machiel S. Blok, Just Ruitenberg, Raymond F. L. Vermeulen, Raymond N. Schouten, Carlos Abellán, Waldimar Amaya, Valerio Pruneri, Morgan W. Mitchell, Matthew Markham, Daniel J. Twitchen, David Elkouss, Stephanie Wehner, Tim H. Taminiau, and Ronald Hanson. 2015. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* 526, 7575 (Oct 2015), 682–686. https://doi.org/10.1038/nature15759 arXiv:1508.05949

[45] Julian Hofmann, Michael Krug, Norbert Ortegel, Lea Gérard, Markus Weber, Wenjamin Rosenfeld, and Harald Weinfurter. 2012. Heralded entanglement between widely separated atoms. *Science* 337, 6090 (2012), 72–75.

[46] Michał Horodecki and Paweł Horodecki. 1999. Reduction criterion of separability and limits for a class of distillation protocols. *Physical Review A* 59, 6 (1999), 4206.

[47] Peter C. Humphreys, Norbert Kalb, Jaco P.J. Morits, Raymond N. Schouten, Raymond F.L. Vermeulen, Daniel J. Twitchen, Matthew Markham, and Ronald Hanson. 2018. Deterministic delivery of remote entanglement on a quantum network. *Nature* 558 (2018), 268–273. https://doi.org/10.1038/s41586-018-0200-5

[48] IEEE 802.1 working group. 2015. 802.1AE - Media Access Control (MAC) Security. (2015).

[49] Takahiro Inagaki, Nobuyuki Matsuda, Osamu Tadanaga, Masaki Asobe, and Hiroki Takesue. 2013. Entanglement distribution over 300 km of fiber. *Optics express* 21, 20 (2013), 23241–23249.

[50] Brian Julsgaard, Alexander Kozhekin, and Eugene S Polzik. 2001. Experimental long-lived entanglement of two macroscopic objects. *Nature* 413, 6854 (2001), 400.

[51] Norbert Kalb, Peter C. Humphreys, Jesse J. Slim, and Ronald Hanson. 2018. Dephasing mechanisms of diamond-based nuclear-spin memories for quantum networks. *Physical Review A* 97 (Feb 2018), 1–11. https://doi.org/10.1103/PhysRevA.97.062330

[52] Norbert Kalb, Andreas Reiserer, Stephan Ritter, and Gerhard Rempe. 2015. Heralded storage of a photonic quantum bit in a single atom. *Physical Review Letters* 114, 22 (2015), 220501.

[53] Norbert Kalb, Andreas A. Reiserer, Peter C. Humphreys, Jacob J. W. Bakermans, Sten J. Kamerling, Naomi H. Nickerson, Simon C. Benjamin, Daniel J. Twitchen, Matthew Markham, and Ronald Hanson. 2017. Entanglement distillation between solid-state quantum network nodes. *Science* 356, 6341 (Jun 2017), 928–932. https://doi.org/10.1126/science.aan0070 arXiv:1703.03244

[54] Harry Jeffrey Kimble. 2008. The quantum internet. *Nature* 453, 7198 (2008), 1023.

[55] Peter Komar, Eric M Kessler, Michael Bishof, Liang Jiang, Anders S Sørensen, Jun Ye, and Mikhail D Lukin. 2014. A quantum network of clocks. *Nature Physics* 10, 8 (2014), 582.

[56] Bo Liu, Baokang Zhao, Ziling Wei, Chunqing Wu, Jinshu Su, Wanrong Yu, Fei Wang, and Shihai Sun. 2013. Qphone: A Quantum Security VoIP Phone. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13)*. ACM, New York, NY, USA, 477–478. https://doi.org/10.1145/2486001.2491696

[57] Yunpeng James Liu, Peter Xiang Gao, Bernard Wong, and Srinivasan Keshav. 2014. Quartz: A New Design Element for Low-latency DCNs. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 283–294. https://doi.org/10.1145/2619239.2626332

[58] Seth Lloyd, Jeffrey H. Shapiro, Franco N. C. Wong, Prem Kumar, Selim M. Shahriar, and Horace P. Yuen. 2004. Infrastructure for the Quantum Internet. *SIGCOMM Comput. Commun. Rev.* 34, 5 (Oct. 2004), 9–20.

[59] Ilias Marinos, Robert N. M. Watson, and Mark Handley. 2013. Network Stack Specialization for Performance. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII)*. ACM, New York, NY, USA, 9:1–9:7. https://doi.org/10.1145/2535771.2535779

[60] Klaus Mattle, Harald Weinfurter, Paul G Kwiat, and Anton Zeilinger. 1996. Dense coding in experimental quantum communication. *Physical Review Letters* 76, 25 (1996), 4656.

[61] David L. Moehring, Peter Maunz, Steve Olmschenk, Kelly C. Younge, Dzmitry N. Matsukevich, Luming Duan, and Christopher Monroe. 2007. Entanglement of single-atom quantum bits at a distance. *Nature* 449, 7158 (2007), 68.

[62] William J Munro, Koji Azuma, Kiyoshi Tamaki, and Kae Nemoto. 2015. Inside quantum repeaters. *IEEE Journal of Selected Topics in Quantum Electronics* 21, 3 (2015), 78–90.

[63] William J. Munro, Ashley M. Stephens, Simon J. Devitt, Keith A. Harrison, and Kae Nemoto. 2012. Quantum communication without the necessity of quantum memories. *Nature Photonics* 6, 11 (2012), 777.

[64] Sreraman Muralidharan, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2014. Ultrafast and fault-tolerant quantum communication across long distances. *Physical Review Letters* 112, 25 (2014), 250501.

[65] Anirudh Narla, Shyam Shankar, Michael Hatridge, Zaki Leghtas, Katrina M. Sliwa, Evan Zalys-Geller, Shantanu O. Mundhada, Wolfgang Pfaff, Luigi Frunzio, Robert J. Schoelkopf, and Michel H. Devoret. 2016. Robust concurrent remote entanglement between two superconducting qubits. *Physical Review X* 6, 3 (2016), 031036.

[66] Kae Nemoto, Michael Trupke, Simon J Devitt, Burkhard Scharfenberger, Kathrin Buczak, Jörg Schmiedmayer, and William J Munro. 2016. Photonic Quantum Networks formed from NV- centers. *Scientific reports* 6 (2016), 26284.

[67] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information* (10th anniversary edition ed.). Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511976667

[68] Wolfgang Pfaff, Bas J. Hensen, Hannes Bernien, Suzanne B. van Dam, Machiel S. Blok, Tim H. Taminiau, Marijn J. Tiggelman, Raymond N. Schouten, Matthew Markham, Daniel J. Twitchen, and Ronald Hanson. 2014. Unconditional quantum teleportation between distant solid-state quantum bits. *Science* 345, 6196 (aug 2014), 532–535. https://doi.org/10.1126/science.1253512 arXiv:1404.4369

[69] Alexander Pirker and Wolfgang Dür. 2019. A quantum network stack and protocols for reliable entanglement-based networks. *New Journal of Physics* 21, 3 (Mar 2019), 033003. https://doi.org/10.1088/1367-2630/ab05f7

[70] QuTech. 2018. NetSQUID. https://netsquid.org/. (2018).

[71] Andreas Reiserer, Norbert Kalb, Machiel S. Blok, Koen J.M. van Bemmelen, Tim H. Taminiau, Ronald Hanson, Daniel J. Twitchen, and Matthew Markham. 2016. Robust Quantum-Network Memory Using Decoherence-Protected Subspaces of Nuclear Spins. *Physical Review X* 6, 2 (Jun 2016), 021040. https://doi.org/10.1103/PhysRevX.6.021040 arXiv:1603.01602

[72] Jérémy Ribeiro and Frédéric Grosshans. 2015. A tight lower bound for the bb84-states quantum-position-verification protocol. (2015). arXiv:quant-ph/1504.07171 https://arxiv.org/pdf/1504.07171.pdf

[73] Daniel Riedel, Immo Söllner, Brendan J Shields, Sebastian Starosielec, Patrick Appel, Elke Neu, Patrick Maletinsky, and Richard J Warburton. 2017. Deterministic enhancement of coherent photon generation from a nitrogen-vacancy center in ultrapure diamond. *Physical Review X* 7, 3 (2017), 031040.

[74] Diego Riste, Stefano Poletto, Myles Huang, Alessandro Bruno, Visa Vesterinen, Olli-Pentti Saira, and Leonardo DiCarlo. 2015. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nature communications* 6 (2015), 6983.

[75] Stephan Ritter, Christian Nölleke, Carolin Hahn, Andreas Reiserer, Andreas Neuzner, Manuel Uphoff, Martin Mücke, Eden Figueroa, Joerg Bochmann, and Gerhard Rempe. 2012. An elementary quantum network of single atoms in optical cavities. *Nature* 484, 7393 (2012), 195.

[76] Nicolas Sangouard, Christoph Simon, Hugues De Riedmatten, and Nicolas Gisin. 2011. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics* 83, 1 (2011), 33.

[77] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. 2009. The security of practical quantum key distribution. *Reviews of modern physics* 81, 3 (2009), 1301.

[78] Jéssica L. Serrano, Pedro F. B. Álvarez, Matthieu Cattin, Emilio G. Cota, John F. Lewis, Paulo Moreira, Tomasz Włostowski, Georg Gaderer, Patrick Loschmidt, Jiří Dědič, Ralph C. Bär, Tiago Fleck, Megan C. Kreider, Celso Prados, and Susan Rauch. 2009. The white rabbit project. In *Proceedings of ICALEPCS*. TUC004, Kobe, Japan, 3. https://www.ohwr.org/project/white-rabbit

[79] Vinay Shankarkumar, Laurent Montini, Time Frost, and Greg Dowd. 2017. *Precision Time Protocol Version 2 (PTPv2) Management Information Base*. RFC 8173. RFC Editor. 1–64 pages. http://www.rfc-editor.org/rfc/rfc8173.txt

[80] Rachee Singh, Manya Ghobadi, Klaus-Tycho Foerster, Mark Filer, and Phillipa Gill. 2018. RADWAN: Rate Adaptive Wide Area Network. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM, New York, NY, USA, 547–560. https://doi.org/10.1145/3230543.3230570

[81] John Strand, Angela L. Chiu, and Robert Tkach. 2001. Issues For Routing In The Optical Layer. *IEEE Comm. Mag.* 39, 2 (2001), 81–87.

[82] SURFsara. 2018. Cartesius. https://userinfo.surfsara.nl/systems/cartesius. (2018).

[83] Barbara M Terhal. 2015. Quantum error correction for quantum memories. *Reviews of Modern Physics* 87, 2 (2015), 307.

[84] Raju Valivarthi, Marcelli Grimau Puigibert, Qiang Zhou, Gabriel H. Aguilar, Varun B. Verma, Francesco Marsili, Matthew D. Shaw, Sae Woo Nam, Daniel Oblak, and Wolfgang Tittel. 2016. Quantum teleportation across a metropolitan fibre network. *Nature Photonics* 10, 10 (Oct 2016), 676–680. https://doi.org/10.1038/nphoton.2016.180 arXiv:1605.08814

[85] Suzanne B van Dam, Peter C Humphreys, Filip Rozpędek, Stephanie Wehner, and Ronald Hanson. 2017. Multiplexed entanglement generation over quantum networks using multi-qubit nodes. *Quantum Science and Technology* 2, 3 (2017), 034002.

[86] Rodney Van Meter. 2012. Quantum networking and internetworking. *IEEE Network* 26, 4 (2012), 59–64.

[87] Rodney Van Meter. 2014. *Quantum Networking* (1st ed.). Wiley-IEEE Press, Hoboken, NJ, USA.

[88] Rodney Van Meter, Thaddeus D. Ladd, William J. Munro, and Kae Nemoto. 2009. System Design for a Long-Line Quantum Repeater. *IEEE/ACM Transactions on Networking* 17, 3 (Jun 2009), 1002–1013. https://doi.org/10.1109/TNET.2008.927260 arXiv:0705.4128

[89] Rodney Van Meter and Joe Touch. 2013. Designing quantum repeater networks. *IEEE Communications Magazine* 51, 8 (Aug 2013), 64–71. https://doi.org/10.1109/MCOM.2013.6576340

[90] Stephanie Wehner, David Elkouss, and Ronald Hanson. 2018. Quantum internet: A vision for the road ahead. *Science* 362, 6412 (Oct 2018), eaam9288. https://doi.org/10.1126/science.aam9288

[91] Stephanie Wehner, Christian Schaffner, and Barbara M Terhal. 2008. Cryptography from noisy storage. *Physical Review Letters* 100, 22 (2008), 220502.

[92] Quantum Xchange. 2019. Quantum Xchange. https://quantumxc.com. (2019).

[93] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, Guang-Bing Li, Qi-Ming Lu, Yun-Hong Gong, Yu Xu, Shuang-Lin Li, Feng-Zhi Li, Ya-Yun Yin, Zi-Qing Jiang, Ming Li, Jian-Jun Jia, Ge Ren, Dong He, Yi-Lin Zhou, Xiao-Xiang Zhang, Na Wang, Xiang Chang, Zhen-Cai Zhu, Nai-Le Liu, Yu-Ao Chen, Chao-Yang Lu, Rong Shu, Cheng-Zhi Peng, Jian-Yu Wang, and Jian-Wei Pan. 2017. Satellite-based entanglement distribution over 1200 kilometers. *Science* 356, 6343 (Jun 2017), 1140–1144. https://doi.org/10.1126/science.aan3211 arXiv:1707.01339[quant-ph]

[94] Wanrong Yu, Baokang Zhao, and Zhe Yan. 2018. Software defined quantum key distribution network. *2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017* 2018-Janua (2018), 1293–1297. https://doi.org/10.1109/CompComm.2017.8322751

[95] Liang Zheng, Carlee Joe-Wong, Chee Wei Tan, Mung Chiang, and Xinyu Wang. 2015. How to Bid the Cloud. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, New York, NY, USA, 71–84. https://doi.org/10.1145/2785956.2787473

[96] Marek Zukowski, Anton Zeilinger, Michael A. Horne, and Aarthur K. Ekert. 1993. "Event-ready-detectors" Bell experiment via entanglement swapping. *Physical Review Letters* 71 (1993), 4287–4290.

Designing a Quantum Network Protocol

Wojciech Kozlowski, Axel Dahlberg, and Stephanie Wehner
QuTech, Delft University of Technology
Kavli Institute of Nanoscience, Delft University of Technology
Delft, The Netherlands
{w.kozlowski,s.d.c.wehner}@tudelft.nl

ABSTRACT

The second quantum revolution brings with it the promise of a quantum internet. As the first quantum network hardware prototypes near completion new challenges emerge. A functional network is more than just the physical hardware, yet work on scalable quantum network systems is in its infancy. In this paper we present a quantum network protocol designed to enable end-to-end quantum communication in the face of the new fundamental and technical challenges brought by quantum mechanics. We develop a quantum data plane protocol that enables end-to-end quantum communication and can serve as a building block for more complex services. One of the key challenges in near-term quantum technology is decoherence — the gradual decay of quantum information — which imposes extremely stringent limits on storage times. Our protocol is designed to be efficient in the face of short quantum memory lifetimes. We demonstrate this using a simulator for quantum networks and show that the protocol is able to deliver its service even in the face of significant losses due to decoherence. Finally, we conclude by showing that the protocol remains functional on the extremely resource limited hardware that is being developed today underlining the timeliness of this work.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Network layer protocols**; • **Hardware** → **Quantum communication and cryptography**.

KEYWORDS

quantum internet, quantum networks, quantum communication

ACM Reference Format:
Wojciech Kozlowski, Axel Dahlberg, and Stephanie Wehner. 2020. Designing a Quantum Network Protocol. In *The 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20), December 1–4, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3386367.3431293

1 INTRODUCTION

The second quantum revolution is currently unfolding across the scientific world [27]. It brings with it the promise of a quantum internet, a global network capable of transmitting quantum data [54, 92].

CoNEXT '20, December 1–4, 2020, Barcelona, Spain

ACM ISBN 978-1-4503-7948-9/20/12.
https://doi.org/10.1145/3386367.3431293

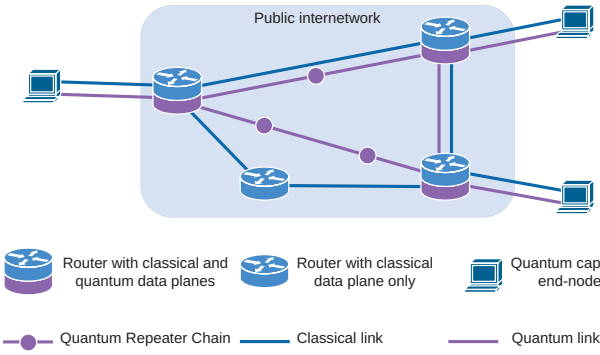


Figure 1: Quantum networks will use existing network infrastructure to exchange classical messages for the purposes of running quantum protocols as well as the control and management of the network itself. Long-distance links will be built using chains of automated quantum repeaters.

Quantum networks will enhance non-quantum (classical) networks (Fig. 1) and they will execute protocols that are provably impossible to do classically or that are more efficient than what is possible classically. This new paradigm enables new possibilities such as quantum secure communications [7, 32], distributed quantum computation [21], secure quantum computing in the cloud [11, 34], clock synchronisation [50], and quantum-enhanced measurement networks [36, 38]. This technology is developing rapidly with the first inter-city network planned to go online in the next few years [4].

Quantum communication has been actively researched for many years. Its most well-known application, quantum key distribution (QKD) is a protocol used for secure communications [7, 32]. Short-distance QKD networks are already being deployed and studied in metropolitan environments (e.g. [64, 77, 81, 91]) and are even commercially available (e.g. [26, 33, 36, 45]). Longer distance QKD networks are currently possible provided all intermediate nodes are trusted and physically secure [19, 74, 77, 78]. However, whilst these nodes are capable of exchanging quantum bits (qubits) with their neighbours, they are not capable of forwarding them (including by means of entanglement swapping, a method explained later in this paper). As a result such networks are unable to transmit qubits end-to-end and thus do not offer end-to-end security.

The next step is to enable long-distance end-to-end communication of qubits. There are three key challenges in realising this objective: transmission losses, decoherence, and the no-cloning theorem. Decoherence is the loss of quantum information due to interactions with the environment and it limits the lifetime of quantum memories. Typical memory lifetimes in quantum networking hardware range from a few microseconds to just over a second [1]

Application	
Transport	Qubit transmission
Network	Long distance entanglement
Link	Robust entanglement generation
Physical	Attempt entanglement generation

Figure 2: Functional allocation in a quantum network stack from Ref. [22]. The structure is inspired by the TCP/IP stack.

though lifetimes of up to a minute have been observed in devices disconnected from a network [8]. The no-cloning theorem states that arbitrary quantum data cannot be copied. Therefore, it is impossible to use standard techniques of amplification or retransmission to compensate for transmission or decoherence losses. Quantum error-correcting techniques for quantum repeaters exist [35, 59, 60] which eventually would be able to compensate for both types of losses [61], but they are extremely demanding in terms of resources and will likely not be feasible for a few more decades.

An alternative to directly transmitting qubits relies on distributing entangled pair states. Quantum entanglement is a special state of two or more qubits in which the individual qubits cannot be described independently of the others, in principle, across arbitrary distances. It is the key ingredient for long-distance communication, because one can use an entangled pair of qubits to teleport an arbitrary data qubit. This bypasses the problem of losses and the no-cloning theorem, because the entangled pairs can easily be regenerated when lost as they need only be delivered from a small set of particular states called the Bell states. Furthermore, this method overcomes transmission losses as long-range entanglement can be created by "stitching" shorter-range pairs together through a process called entanglement swapping [10] which means that it is not necessary to transmit qubits directly along the entire path. Entanglement generation between two directly connected nodes with a quantum memory has been demonstrated at distances of up to 1.3 km [41] and work is underway to build a three-node setup and extend the inter-node distances to several kilometres [28, 83].

In this paper we design a quantum network protocol capable of generating end-to-end entanglement marking the next step in the development of long-distance quantum communication networks. The starting point for our work is a recently proposed protocol for generating link-level entanglement [22]. Going from link-level entanglement to end-to-end entanglement is a significant leap in complexity as it requires many new mechanisms that do not exist at the link level. In our protocol we develop solutions for: (i) coordinating entanglement swapping between multiple nodes in order to "stitch" link-level entanglement into long-range end-to-end entanglement, (ii) reducing the amount of decoherence experienced by qubits stored in quantum memory, (iii) compensating for qubits lost due to decoherence, (iv) ensuring that the final entangled pair is of sufficient quality to be useful in an application, and many other problems. The result of our work is a quantum data plane protocol capable of creating end-to-end entanglement thus enabling long-distance quantum networks. In particular, our design focuses on ensuring efficient entanglement generation in the face of short

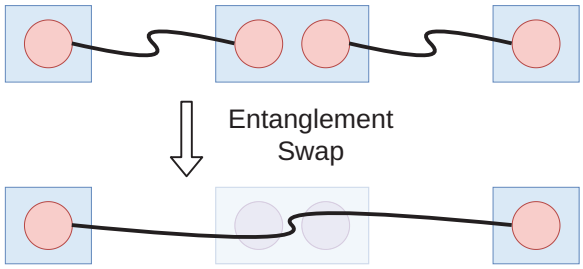


Figure 3: Quantum repeaters create long-distance entanglement by connecting short-distance entangled pairs. Initially two entangled pairs each have one qubit in the memory of the middle repeater. An entanglement swap is performed on these qubits which destroys the entanglement of the two pairs, but as a result the remote qubits become entangled.

memory lifetimes. At the same time we ensure scalability by designing the protocol to be a building block for more complex quantum network services rather than a complete all-in-one solution. Our key research contributions are:

(1) We design a protocol for generating end-to-end entangled pairs in the face of decoherence that fulfils the role of a quantum network layer.
(2) We outline the architecture for the construction of quantum network services and design our protocol to fulfil the role of the building block in this scheme.
(3) We evaluate the effectiveness of the proposed protocol against decoherence in a quantum network simulator.
(4) We show that it remains functional on extremely limited near-term hardware justifying its timeliness.

2 BACKGROUND AND MOTIVATION

Here, we provide the motivation and justify the timeliness of a quantum network protocol. We only provide an introduction to the quantum mechanical concepts necessary to understand the protocol design. Nevertheless, quantum networks are not new in literature and good introductions to the subject can be found in Refs. [22, 54, 85, 88, 92].

2.1 Motivation

So far, the generation of long-lived entanglement has been the domain of highly sophisticated physics experiments. However, real deployments of quantum networks are around the corner with the first inter-city network scheduled to go online within the next few years [4]. Much essential work is being done to build quantum hardware to make this possible [8, 58, 75, 83, 92] and we are now entering a new phase of development where we need to learn how to build quantum communication *systems*. Work in this field has been slowly emerging over the last few years (see e.g. [13, 49, 57, 58, 76, 86]). Recently, a proposal for a quantum network stack inspired by TCP/IP has been put forward (Fig. 2) along with a link layer protocol that provides a robust entanglement generation service between directly connected nodes [22]. Here, we go one level up this network stack and achieve the next step in quantum connectivity, a quantum network layer protocol capable of providing long distance end-to-end entanglement between any pair of nodes in the network.

2.2 Entanglement Swapping

In light of the the no-cloning theorem, decoherence, and transmission losses how can entangled qubits be practically distributed if we cannot use amplification or retransmissions? In 1998 Briegel et al. [10] proposed a solution whereby *quantum repeaters* create long-distance entanglement by connecting a string of short-distance entangled pairs of qubits through a process called entanglement swapping, shown in Fig. 3. Therefore, a practical scheme for distributing entanglement may combine a scheme for generating short-distance entangled pairs, such as a quantum link layer protocol [22] which wraps the physical mechanism [43, 46, 68] for pair generation, with entanglement swapping at quantum repeaters.

Despite the quantum nature of the underlying physical processes, quantum networks will require classical connectivity between all the quantum nodes as shown in Fig. 1 for the exchange of control messages. Most notably, entanglement swapping as shown in Fig. 3 requires the middle node to send a message to at least one of the other nodes for the entanglement to be useful[1]. Furthermore, just like classical networks, quantum networks will need control and management protocols which will also use the classical channels.

2.3 Fidelity and Decoherence

Next to standard measures like throughput and latency, a key parameter in a quantum network is a quantity called *fidelity* [22]. Fidelity is a purely quantum metric with no classical equivalent. Its value lies between 0 and 1 and it quantifies the quality of the state in terms of how "close" it is to the desired state (a fidelity of 1 means it is in the desired state, a value below 0.5 means that the state is no longer usable). It is important to note that unlike in classical networks where data must be delivered error-free, quantum applications are able to operate with imperfect quantum states — as long as the fidelity is above an application-specific threshold (for basic QKD the threshold fidelity is about 0.8).

Decoherence is the gradual degradation of qubit quality over time and will cause the fidelity to decrease. Decoherence is one of the key challenges in quantum networks as it puts extremely stringent limits on how long qubits can be held in memory before they need to be used. In current experimental hardware, these times are of the order of few milliseconds [22, 43], but memories in similar devices disconnected from a network have shown lifetimes of up to one minute [8].

Quantum state fidelity in a network is lost in several ways:

(P1) Short-range pairs generated on a link are imperfect.
(P2) Swapping imperfect pairs results in a pair of lower fidelity even if the physical operations are noiseless.
(P3) Imperfect implementations of quantum gates reduce fidelity whenever any qubit is processed.
(P4) Decoherence degrades a quantum state's fidelity while the qubits are stored in memory.

Whilst the fidelity of a short-distance pair generated on a link (P1) is ultimately the property of the hardware, some implementations are able to vary the fidelity of the produced pairs though higher fidelities come at the cost of reduced rates [22]. The issue in (P2) is a fundamental property of entanglement swapping and the only way to ensure that the output state is sufficiently good is to feed sufficiently high quality states into the swap. (P3) is similar, but can also be addressed by improving the hardware which is out of scope for a network protocol. Finally, decoherence (P4) can be addressed at the protocol level by minimising the time qubits spend idling in memory. Therefore, in our design we focus on addressing two key questions: (i) how does the protocol know what fidelity to request on the individual links to ensure a sufficiently high end-to-end fidelity after all the operations complete, and more importantly (ii) how to minimise decoherence by reducing the amount of time qubits sit idly in memory.

[1]The entanglement swap results in one of four possible entangled states, but which state is produced is fundamentally random. The node that performed the entanglement swap will obtain two bits of information indicating which state was produced. Without this information the remote nodes do not know what state they share rendering it useless to any application.

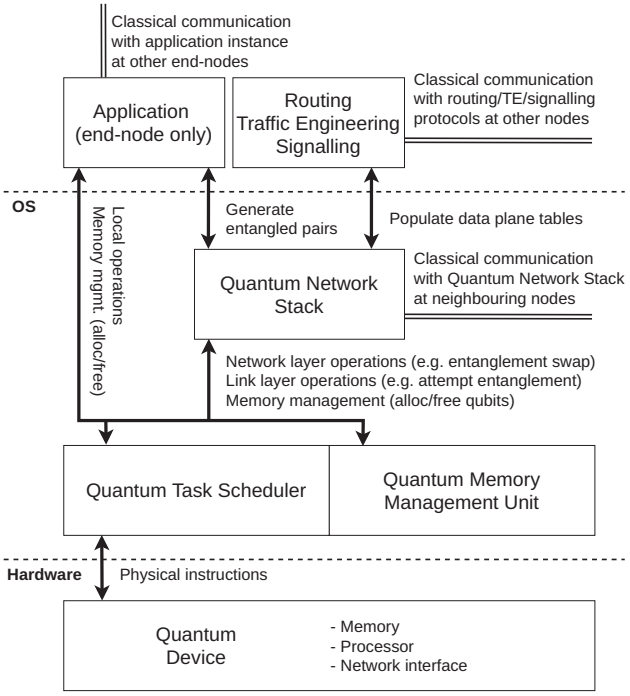


Figure 4: Local system components of a quantum node. The quantum memory, processor, and network interface are all one hardware component on current platforms. Local gate and network operations are performed on qubits in the main memory. Management and arbitration of local hardware resources belong to local operating system components such as a quantum task scheduler and a quantum memory management unit.

2.4 Quantum Node Architecture

We first define the high-level architecture of a quantum node, shown in Fig. 4. The network stack is expected to be part of a local operating system (OS). The stack is responsible for managing operations relating to the generation of entangled pairs which it executes with the help of local OS services.

Upon receiving an entanglement request (from an application or an upstream node) the network stack will need to do two things: (i) coordinate with neighbouring quantum nodes and (ii) issue local

instructions to generate entangled link-pairs and perform entanglement swaps. The network stack coordinates with its neighbours by exchanging classical messages (all nodes are connected classically, see Fig. 1). Just like in classical networks, certain tasks such as path computation happen outside of the network stack itself. These tasks are delegated to other protocols which communicate their decisions to the local network stack by means of populating relevant data plane structures. Additionally, the network stack will have to issue instructions to the local quantum device in order to generate link-pairs and perform entanglement swaps. In currently available hardware, unlike in classical devices, there is no distinction between the processor and the network interface and they both operate directly on qubits in the main memory. Though, in general, they are not able to operate on any arbitrary qubit on the device. The precise nature of these limitations strongly depends on the hardware implementation, but at a high level the qubits are split into *communication qubits*, those that can participate in networked operations, and *storage qubits*, those that can store quantum information but cannot be used for entanglement generation [22]. The network stack relies on other OS components such as a quantum task scheduler and a quantum memory manager for arbitrating access to hardware.

3 THE QUANTUM NETWORK LAYER

3.1 Use Cases

Currently, no quantum networks exist so it is impossible to derive any use cases based on real usage statistics. However, Ref. [22] identifies two categories of use cases that represent application demands of quantum application protocols known to date: "measure directly" and "create and keep".

Measure directly Applications in this category are characterised by the fact that they consume the delivered pairs (by measuring them) as soon as they are available and do not store them. Therefore, they can tolerate fluctuations in the rate of delivery as the qubits never sit idly in memory where they would decohere. This use case is relevant for applications that use the entangled pairs to produce stronger than classical correlations such as QKD [32], secure identification [23], other two-party cryptographic protocols [3, 14, 24, 69, 93], and other applications in the prepare-and-measure stage of quantum networks [92].

Create and keep Applications in this category are characterised by their need for storage, possibly of multiple entangled pairs simultaneously. This use case is relevant for applications that may want to send qubits deterministically (via teleportation), perform joint operations on multiple qubits, or perform operations that depend on back and forth communication with another node. Due to decoherence, these applications cannot tolerate large delays between successive pairs. Examples of such applications include sensing [38], metrology [50], and quantum distributed systems [6, 25].

3.2 Service Delivered to Higher Layers

Here, we explain the key aspects of the quantum network layer service delivered to the higher layers.

Entangled pair identifier Logically, the network delivers an entangled pair. Physically, the network delivers one entangled qubit to each of the two end-nodes. This means that the network must track the entanglement swaps that connect the individual link-pairs into a long-range pair such that at the end it can identify which qubits at the end-nodes belong to the same pair. When delivering the qubits, it provides this by means of an entangled pair identifier.

Entangled pair state Entangled pairs come in four variants called the Bell states. They are all equally usable, but the recipient must know which one it has received. Due to the fundamental randomness of quantum mechanics, the state of each pair produced by entanglement swaps is not known a priori, but is revealed to the swapping node upon the swap's completion. The network must collect these announcements, infer the state, and deliver this information to the application.

Class of service: fidelity States do not have to be perfect to be usable as long as they are above an application-specific threshold. Since more time is needed to produce better states, applications can sacrifice fidelity in exchange for higher rates (or vice-versa). Therefore, the user must specify a minimum fidelity threshold, F, on each request. The network then attempts to deliver these states. A strict guarantee is not required, because end-to-end quantum security proofs do not rely on a trustworthy source of entanglement.

Class of service: time The application must be able to quantify its desired fidelity-vs-rate trade-off, especially in light of the use cases described in Sec. 3.1. For the "measure directly" use case, the application can specify its requirement as either (i) N pairs by deadline T or (ii) a rate of R pairs per unit time. For the "create and keep" use case the application specifies that it requires N pairs by deadline T such that the last pair is delivered at most Δt after the first. In both cases T may be set to zero to indicate no deadline.

3.3 Network Layer Architecture

Delivering the full network layer service cannot be accomplished with one protocol alone. Instead, we envisage a situation similar to the one that exists in classical networks where a variety of different services are built from simpler building blocks such as the IP datagram or MPLS virtual circuits. In this paper, we propose a *quantum data plane* protocol that aims to provide such a building block for quantum networks. However, our protocol requires support from at least two external services: a signalling protocol and a routing protocol. In this paper we only propose a quantum data plane protocol, but we first outline the roles of the supporting protocols.

Routing protocol Before any end-to-end entangled pair can be generated the optimal path must be determined. Just like in a classical network this is expected to be done by a separate routing protocol. However, routing in quantum networks is more complicated because it must compute the paths not only based on path length, cost, and throughput, but it must also take into account the desired end-to-end fidelity. Higher fidelity paths will require links that can produce higher fidelity link-pairs and nodes with longer lasting memories. Furthermore, higher fidelity link-pairs require more time to produce which must be taken into account when determining available bandwidth. Routing algorithms for quantum networks are an emerging field of study [12, 15, 16, 39, 40, 44, 56, 79, 80, 87].

Signalling Protocol Our protocol is connection-oriented. It requires a fixed path, called a virtual circuit, to be established between the end-nodes prior to its operation. Installing virtual circuits will be the task of a signalling protocol. This is similar to how RSVP-TE

is used to install MPLS virtual circuits in classical networks. However, allocating a path with sufficient resources is not enough. In a quantum network the upstream and downstream links at each node must generate their link-pairs sufficiently close in time that they do not decohere before swapping. The routing component is responsible for choosing a path based on available resources, but does not decide how to use them. On the other hand, the quantum data plane protocol's worldview will be limited to that of a single virtual circuit. We propose that the signalling protocol is best suited to the task of schedule management. It is an open question how best to perform scheduling at a quantum node [89, 90]. In early-stage network this synchronisation will have to be very precise and may need to allocate dedicated time bins to each circuit.

These protocols can be implemented in a distributed or centralised fashion. Researchers have considered both distributed [16] as well as centralised routing protocols [12, 15, 56, 87] in quantum networks. Our design does not assume either architecture.

3.4 Quantum Data Plane Protocol

In analogy to classical networks, where the task of delivering connectivity once all state has been installed is the responsibility of a data plane protocol, in this paper we propose a *quantum data plane* protocol. We define the quantum data plane protocol as the component that is responsible for coordinating the generation of link-level entanglement and the subsequent entanglement swapping along a path between two distant nodes while minimising the losses due to decoherence and compensating for the losses that do happen. The focus of our work are mechanisms for the quantum data plane, that is, local quantum operations and the classical messaging to coordinate these operations. It is important to note that we do include classical message exchange that is necessary to coordinate quantum operations in the definition of the quantum data plane. However, it is not within the quantum data plane's domain to perform any resource management, routing, or any other long-term state management. Therefore, in this work we will assume the existence of suitable routing and signalling protocols and focus on defining what information we expect them to provide to the quantum data plane. This is in contrast to existing works on quantum routing which focus on control plane aspects and algorithms while working with an abstract model of the data plane. Nevertheless, the quantum data plane protocol is expected to participate in policing and shaping of the traffic to meet the use case requirements outlined in Sec. 3.1. We expect the such a protocol to have three tasks:

Link-pair generation management To create a long-distance pair, link-pairs must be first generated along the entire path. The network layer is not expected to manage the physical process directly, but instead will rely on a link layer protocol [22] to deliver these link-pairs as per the quantum network stack design shown in Fig. 2. However, it is the network layer's responsibility to manage the link layer service at each node along the path such that a sufficient amount of link-pairs of suitable fidelity are produced.

Entanglement swapping and tracking Once the link-pairs are generated, the repeaters must perform entanglement swaps to create long-distance entangled pairs. In addition to performing the physical operation, the protocol must also track the swaps that were involved in producing each end-to-end pair. This is done for two reasons (outlined in Sec. 3.2): to correctly identify which qubits that belong to the same end-to-end pair and which Bell state they are in. Therefore, the network protocol needs a mechanism to collect the entanglement swap outcomes and deliver them to the end-nodes so that the final Bell state of the end-to-end pair can be inferred and delivered to the recipient.

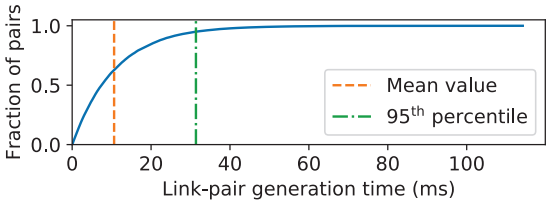


Figure 5: The cumulative distribution function for the time taken to generate a link-pair of fidelity 0.95 over a 2 m long fibre with the same hardware parameters as used in Sec. 5. The y-axis denotes the fraction of pairs generated in less than the time indicated on the x-axis. We see that on average we have to wait 10 ms and that 95% of link-pairs are generated within 30 ms.

Quality of service management Whilst the quantum data plane protocol cannot guarantee the quality of service on its own, it is expected to provide basic mechanisms that will allow the supporting protocols to achieve this goal. This includes at least (i) confidence that the fidelity is above the threshold, (ii) policing requests by rejecting any that cannot be fulfilled, and (iii) shaping traffic by delaying requests that can be fulfilled later.

3.5 The Link Layer Service

The link layer protocol interacts with the physical layer and exposes a meaningful link entanglement generation service to the network layer. It is meaningful in the sense that it is responsible for batching and multiplexing entanglement attempts across a link in order to either deliver an entangled pair to the higher layer with suitable identifiers or notify it of failure. Since the probability of success on each entanglement generation attempt is generally low, the link layer is expected to include a retry mechanism to increase reliability. Fig. 5 shows how long it takes to create a link-pair.

A single link layer request is simply an asynchronous request made at one end of the link which returns entangled qubits at both ends. Our network protocol requires four properties from the link layer. (i) A link-unique request identifier can be assigned to each link layer request. This identifier must accompany all qubits delivered as part of this request at both ends — this allows the network protocol to coordinate its own actions across a link (Purpose ID in Ref. [22]). (ii) An identifier must be assigned to each pair that uniquely identifies it within the particular link layer request — the network layer needs this for entanglement tracking purposes (Entanglement ID in Ref. [22]). (iii) The link layer must inform the network layer which of the four Bell states the qubits are delivered — this information is needed for entanglement tracking in order to infer the final state of the end-to-end pair. (iv) The caller must be able to specify relevant quality of service parameters: minimum fidelity and time restrictions — this allows the network protocol to fine-tune its own quality of service properties.

4 QUANTUM NETWORK PROTOCOL

4.1 Protocol Design

Here, we present the main design aspects of our quantum data plane network protocol, the Quantum Network Protocol (QNP). A more detailed description is available online [52].

Principle of operation The QNP becomes operational once a virtual circuit (VC) is installed into the network by the signalling protocol using the path provided by the routing protocol. A VC is defined as a fixed path between two end-nodes with the necessary data plane state installed into the local network stack data structures. The circuit is directed with a head-end node at the upstream end and a tail-end node at the downstream end. It is up to the signalling protocol to determine which direction is upstream and which is downstream. Whilst the entangled pairs are directionless this distinction is used to give upstream nodes the right to initiate pair-wise activities, such as link-pair generation.

The QNP starts when a request is received at the head-end node (for simplicity we currently require the tail-end node to forward user requests to the head-end node). This triggers a FORWARD message sent downstream towards the tail-end node initiating link-pair generation for this particular VC on each link along the path. Once two link-pairs are generated at the same intermediate node, one on the upstream and one on the downstream link, an entanglement swap is immediately performed (without any further classical communication). The swap outcomes are collected by two TRACK messages, one going downstream and one upstream. Once the TRACK messages reach the end-nodes the pair is delivered to the application. Some applications can consume the qubits before the TRACK messages arrive which we discuss later.

Virtual circuits The central property of our protocol is that it is connection-oriented. That is, a connection, in the form of a VC installed by the signalling protocol, must be established prior to the QNP's operation. This decision is driven by the fact that link-pair generation and entanglement swapping are parallelisable. Link-pairs themselves are completely independent of each other until they are connected via an entanglement swap so they can all be generated at the same time. Furthermore, the order in which the entanglement swaps are executed also does not matter. VCs enable parallelisation as they allow us to dedicate resources on each link along the path for a particular end-to-end connection. Since link-pair generation is not necessarily a fast process (rates in laboratory setups are of the order of few tens of Hz [43]) this is a significant performance optimisation. Short memory lifetimes further compound the benefits of parallelisation as it allows the protocol to minimise the decoherence experienced by the qubits — they will not have to wait as long for a matching qubit to become available for swapping.

VCs are installed by the signalling protocol. It achieves this in a similar manner to MPLS: by assigning a link-unique label, called the link-label, to each link on the path of the circuit. The network protocol then uses this label as its request identifier when issuing requests to the link layer service. Entanglement swaps are performed as soon as pairs with labels for the same VC are available on the upstream and downstream links.

It is worth noting some works [63, 80] on routing entanglement in quantum networks assume a different model for the quantum data plane. Instead, they build upon an abstract model of the network whereby some subset of (or all) links in the network attempt to generate entanglement in pre-defined time slots. Swapping decisions are then made by each node based on their knowledge of global topology combined with information about which of the nearby links succeeded in generating entanglement in that time slot. These quantum data plane models show good performance when used in conjunction with the aforementioned routing protocols. However, they rely on networks that are sufficiently big that they are able to support multiple paths between the relevant source and destination pairs which will not be the case for near-term deployments. Our quantum data plane protocol does not have this problem as it is designed to be operational on single paths. However, as our protocol is inspired by MPLS VCs, generalising it to multipath scenarios (for redundancy or bandwidth purposes) will be straightforward at which point it may also be used with multipath entanglement routing protocols.

Swap records As explained in Sec. 3.4 the protocol must track the entanglement in addition to performing entanglement swaps. That is, it must (i) correctly identify which qubits at the end-nodes are part of the same entangled pair and (ii) collect all the entanglement swap results to infer the final Bell state of the end-to-end pair. For this reason, as soon as an entanglement swap completes, a temporary swap record is logged at the node. This record must contain the following information: (i) the link-unique identifiers (Entanglement ID) for the two pairs involved in the swap and (ii) the two-bit output of the entanglement swap. Provided the Bell states of the input pairs are known, the two-bit output uniquely identifies the Bell state of the output pair which now spans between the two remote qubits of the two input pairs (see Fig. 3).

Lazy entanglement tracking The swap records generated after every entanglement swap must be collected and sent to the end-nodes so that they can deliver the end-to-end pair with the correct identifier and Bell state information. The QNP achieves this by sending an entanglement tracking message (TRACK) from the head-end to the tail-end along the VC which collects the records at each node it visits, waiting if a swap has not completed yet. A similar message is sent in the reverse direction so that the head-end can also receive this information.

An individual swap record is sufficient to identify the Bell state of the output entangled pair provided the input Bell states are known. The problem is that in a VC with multiple intermediate nodes where the ordering of the entanglement swaps is not defined, the swapping nodes do not actually know what the input states are (as other swaps along the VC may have already happened changing the state of the input pairs) so they are unable to infer the output state from their swap record on their own. However, we do not need the swapping nodes to know this information — it is only needed at the end-nodes. The TRACK messages collect the swap records one by one from one end-node all the way to the other end-node. As the ordering of the entanglement swaps does not matter, logically, a TRACK message can be thought of as reconstructing the final entangled pair state as if the swaps happened in the order it collects these swap records. In this context, the TRACK message effectively carries information about the input state for the next swap. When it collects a new swap record, it uses the two-bit swap outcome contained within to infer the new input state for the swap

at the next node. When it reaches the final end-node this "next input state" is actually the final entangled pair state. This logical picture works for TRACK messages in both directions (upstream and downstream) as the ordering of swaps does not matter.

We call this lazy entanglement tracking, because the protocol does not keep track of any of the intermediate pairs created throughout the process. The swaps do not necessarily happen in the order the TRACK messages collect the records so they do not represent the intermediate states as they really happened. The only pair the TRACK message is guaranteed to know the state of is the final pair. This allows: (i) quantum operations to proceed regardless of classical control messages being communicated and (ii) individual nodes to discard decohered qubits (discussed later) without having to separately communicate this with the rest of the VC.

The ability to do lazy entanglement tracking is an advantage of the connection-oriented approach as opposed to a hop-by-hop model where each node makes a swapping decision without any prior agreement. In that case it would be necessary to keep track of all intermediate pairs in order to know what pair will result from the next swap. This would introduce additional latency and synchronisation issues as the protocol would need to constantly update its entanglement information database. In the worst case this will block entanglement swaps until the protocol completes synchronising this information which is highly undesirable, especially in the presence of decoherence.

Cutoff time When memory lifetimes are short (as will be the case for near-future hardware), it often happens that a qubit may decohere too much by the time a suitable pair on another link is available. To counteract this, we adopt the cutoff mechanism from repeater chain protocols [49, 55, 71, 73]. The protocol discards qubits that have not been swapped, but have reached a cutoff deadline. The tighter the deadline the less likely it is that two links will be able to generate link-pairs at the same time, but when they do manage to be generated within the cutoff window the qubits will have suffered from less decoherence leading to a higher end-to-end fidelity. Therefore, we allow the external routing protocols to specify the cutoff value as well. These timeouts can then be distributed by the signalling protocol when setting up the circuit.

When a qubit is discarded, the node must log a temporary discard record. When an entanglement tracking message arrives at the node, it will check for the discard record if it cannot find a swap record. If the discard record is present, the tracking message will be sent back to its origin to notify that end-node of the broken chain. The cutoff timer is not used at the end-nodes as we found this to result in a window condition where one end-node delivers its end of the pair to the application whilst the other end-node discards the other qubit. Therefore, the end-nodes instead discard their qubits upon receipt of this expiry notification.

Policing and shaping If circuits are used with a resource reservation mechanism they will also be allocated a maximum end-to-end rate (EER), i.e. bandwidth. The routing protocol computes a path that can support a given EER and the signalling protocol provides the head-end node with this EER value so that the QNP can police (reject) and shape (delay) incoming requests. The head-end node calculates a request's minimum EER, compares it to its available bandwidth and decides if the request can be satisfied by the specified deadline T. Our service definition from Sec. 3.2 requires applications to always specify their minimum rate in its request which we use as its minimum EER (measure directly: N/T, R, or 0 if T not set; create and keep: $N/\Delta t$). Note that these checks are only made at the end-nodes and we do not implement any further in-network mechanisms. It is the role of the resource reservation protocols to ensure that network resources are not over-subscribed as long as the end-nodes fulfil their part of the contract by policing and shaping the incoming requests.

Continuous link generation Discarding qubits due to decoherence will be the norm rather than the exception in early-stage networks. Therefore, an efficient retry mechanism is necessary. For this reason, the quantum network protocol simply requests the link layer service to produce a continuous stream of pairs until the end-nodes signal the completion of the request. To allow the link layer to multiplex requests from different circuits, the network layer must provide some information about the desired rate. The link-pair rate (LPR) must necessarily be higher than the EER as some link-pairs will be discarded due to decoherence. The routing component will have calculated the necessary LPRs for each link when determining which path can support a given EER [15]. The QNP will request the maximum LPR on each link unless the only active requests are rate-based ("measure directly" requests that specify R) in which case it requests a suitable fraction of the circuit's LPR based on the fraction of its EER that these requests need.

Early delivery For the "measure directly" use case the application may benefit from acting on its entangled pair as soon as possible to minimise decoherence. Some applications can start operating on the qubit at their end-node before all entanglement swaps complete — the effect will be propagated with the swaps to the other end. Thus, they may choose to have the QNP perform a measurement as soon as its end of the pair is available or have it delivered before the protocol can confirm the pair's creation. In the case of a measurement, the protocol simply withholds the result until the tracking messages arrive so that only results from successful pairs are delivered. If the pair was delivered early, the application must take over the responsibility of handling any error messages such as notifications about discarded pairs. It will also have to wait for the final entanglement tracking information of the entangled pair to correctly post-process its results.

Aggregation Entangled pairs generated between the same two end-nodes for the same fidelity threshold are, for application purposes, indistinguishable. Therefore, the QNP may aggregate such requests onto the same VC. Aggregation is an important feature of the protocol that enables scalability, because (i) it reduces the amount of state the network needs to manage by reducing the total number of circuits, and (ii) it improves resource sharing at entanglement swapping nodes. To explain the second point, we note that a repeater node may only swap two entangled pairs if they belong to the same circuit. Without aggregation, a node would have to wait for two pairs allocated to the same request before swapping. With aggregation the nodes do not have to distinguish between individual requests if they share the same VC.

Aggregation means that the VC does not keep track of any request information. Therefore, demultiplexing, i.e. assigning a VC's pairs to requests, must be done by the end-nodes. There are many ways to do this. The QNP only requires that the end-nodes agree on a method which can be negotiated when the VC is set up. The

end-nodes may use a distributed queue, have the head-end node make all the decisions and communicate them on the TRACK messages, or use some other algorithm. We do not specify the strategy as part of the protocol. However, we do provide two mechanisms to aid in this task. (i) Epochs: an epoch is the set of currently active requests. A new epoch is created (but does not activate) whenever a request is received or completed. The head-end advances the active epoch by setting the value of the next one on each TRACK message. Once the entangled pair corresponding to that TRACK message is delivered the epoch indicated by that message becomes active. (ii) TRACK messages carry information about which request they were assigned to by the end-node that originated the message. Due to the cutoff strategy, qubits along the VC may be suddenly discarded which leads to window conditions where the end-nodes may not agree on which request the pair was assigned to. This information allows the end-nodes to perform a cross-check and discard such qubits if necessary (if a qubit was not delivered early it may even be possible to reassign it).

Routing table To communicate all the routing decisions to the quantum data plane protocol, we require a routing table entry at each node for each VC. This entry must contain: (i) the next downstream node, (ii) the next upstream node (TRACKs are also sent upstream), (iii) the downstream link-label, (iv) the upstream link-label, (v) the downstream link minimum fidelity, (vi) the downstream maximum LPR, and (vii) the circuit maximum EER. The fidelity threshold for a link will be higher than the end-to-end fidelity to account for losses due to entanglement swapping and decoherence. The nodes are also provided with the circuit maximum EER so that the QNP can scale its LPR if the VC's maximum EER is not required. We delegate the responsibility for choosing the fidelity and LPR values to a routing protocol for two reasons: (i) choosing them requires knowledge of the entire path — the longer the path, the higher must they be on each link to compensate for various losses — and (ii) their exact values depend on the hardware parameters of all the nodes and links on the path of the VC.

It is worth noting that the LPR and link fidelity values do not have to be identical for every link along the path of a particular VC. In fact, this is likely to be the case in heterogeneous networks where the different links have different rate-fidelity trade-offs and the nodes have different decoherence rates. Assuming isotropic noise (i.e. the worst case scenario) so that the entangled pairs can be expressed as Werner states [94] it can be shown that the fidelity, F', of an entangled pair produced by combining two pairs with fidelities F_1 and F_2 is given by

$$F' = F_1 F_2 + \frac{(1 - F_1)(1 - F_2)}{3}.$$

This expression is associative and thus variations in link fidelity do not affect our key assumption that entanglement swaps can be performed in any order. Therefore, in heterogeneous networks it is conceivable that the fidelity is "budgeted" differently across the different links as necessary to improve the end-to-end rates.

Fidelity test rounds It is physically impossible for the protocol to peek or measure the delivered pairs to evaluate their fidelity. However, we need a mechanism to provide some confidence that the states delivered to the application are above the fidelity threshold. We apply the same method as proposed in Ref. [22] for individual links, but instead test end-to-end pairs. In summary, the method relies on creating a number of pairs as test rounds which are then measured (and thus consumed). The statistics of the measurement outcomes can be used to estimate the fidelity of the non-test pairs.

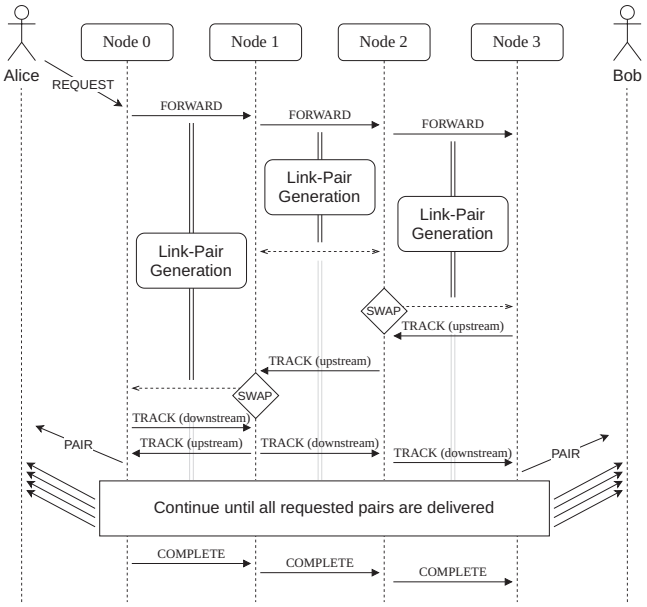


Figure 6: Example sequence of the QNP.

Classical communication and link reliability The protocol requires that all its control messages are transmitted reliably and in order. It is designed to not depend strongly on the classical messaging latency so that we may simply rely on a transport protocol to provide these guarantees (e.g. TCP or QUIC). Every VC establishes its own transport connection between every pair of nodes along its path for this purpose. The transport's liveness mechanism can then be used to monitor the classical channel liveness and tear down the VC if the connection goes down. The quantum link layer is also expected to support a liveness check mechanism (Ref. [22] does in the form of fidelity testing rounds). If a circuit goes down due to loss of connectivity, the protocol aborts all requests and notifies applications of the failure.

4.2 Example Sequence

Fig. 6 illustrates a sequence diagram of a sample flow. Upon receiving a request, a FORWARD message is sent along the VC to initiate link-pair generation. Entanglement swaps execute as soon as an upstream and downstream pair are available for the same circuit and a swap record is generated upon its completion. Each end-node initiates a TRACK message as soon as their link-pairs are available. The TRACK messages proceed along the circuit collecting swap records, waiting for the corresponding pair's swap to complete if necessary. Once the TRACK messages arrive at the destination end-nodes, the final identifier and Bell state information are delivered together with the qubit itself, if not delivered early. Once all pairs are delivered, a COMPLETE message is sent along the circuit to terminate/update the link layer requests.

4.3 Entanglement Distillation

Entanglement distillation is a process through which two or more imperfect pairs are consumed to produce a higher fidelity pair with some finite probability [30, 48]. However, because entanglement distillation has higher hardware requirements than entanglement swapping, it is not the solution to extremely fast decoherence. Nevertheless, it will be necessary to overcome poor quality links and the fundamental loss of fidelity due to entanglement swapping which ultimately limits the achievable path length.

We decided not to incorporate distillation into the protocol at this stage of development of quantum networks, because it is still an open research question as to what the right distillation strategy is: should distillation happen as soon as link-pairs are generated, after every swap, after N swaps, at the ends only, etc. Furthermore, there are many different methods available for performing distillation, each with its own trade-offs [72]. However, the QNP was designed to be used as a building block for more complex quantum network services and entanglement distillation offers a particularly interesting example of such a service. Therefore, we instead illustrate how distillation could be implemented on top of our protocol.

To implement distillation using the QNP we rely on the observation that this process consumes two or more entangled pairs between the same pair of nodes. Therefore, one can implement distillation in a layered fashion. We run the network protocol between a pair of intermediate nodes which deliver entangled pairs to a distillation module. Once distilled, the module passes the higher fidelity pair to another circuit that only runs between the distillation end-points and that sees all the nodes in between as one virtual link. This proposal is similar to some of the early quantum network architecture proposals [88].

5 EVALUATION

To evaluate the performance of the QNP we have implemented it on top of a purpose-built discrete event simulator for quantum networks called NetSquid (Python/C++) [66]. The simulator is responsible for the accurate representation of the physical hardware including decoherence, propagation delay, fibre losses, quantum gate operations and their time dependence. The protocol itself is implemented in Python and runs on top of the link layer implementation from Ref. [22].

As our work is focused on quantum data plane processes we keep the control plane as simple as possible. For routing purposes we implement a rudimentary algorithm that runs in a central controller and assumes all links and nodes are identical. It calculates a network path together with link fidelities as a function of end-to-end requirements by simulating the worst case scenario where every link-pair is swapped just before its cutoff timer pops. The routing information is installed by a source-routed signalling protocol. We also implement a simple swapping and link scheduling algorithm. Links function independently of each other and schedule requests using a weighted round-robin scheme where the number of pairs generated for a particular VC is proportional to its LPR and inversely proportional to the average time per pair. This mechanism ensures that: (i) circuits get an equal share of the link's time regardless of fidelity (higher fidelity VCs need more time to achieve the same rate), (ii) when under-subscribed the excess capacity is distributed proportionally to demand, (iii) when over-subscribed the available capacity is distributed proportionally to demand. At each node, each VC will maintain two logical queues of link-pairs (upstream and downstream) ready for swapping. Note that these queues are only logical and they must all share a limited number of physical qubits for storage purposes — we do not pre-allocate qubits to particular VCs. For queuing entanglement swaps we employ the first in, first out strategy with the caveat that qubits may expire due to the cutoff timer. That is, entanglement swaps always prefer the oldest unexpired upstream and downstream pairs that correspond to the same VC. We do not perform any resource management (all VCs are admitted regardless of available bandwidth) as it is an open research question beyond the scope of this paper. Instead, we examine the protocol's performance under different loads and draw conclusions as to how resources could be managed.

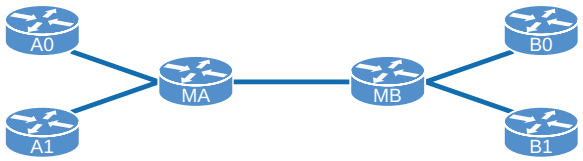


Figure 7: The evaluation topology. MA-MB is a bottleneck link between the A nodes and the B nodes. All links consist of a quantum and a classical channel.

For the evaluation we consider the topology shown in Fig. 7 which has six nodes in total, four of which we use as end-nodes (A0, A1, B0, B1), and with one bottleneck link (MA-MB). The dumbbell topology is complex enough that it is functionally beyond the capabilities of repeater chain protocols and requires the ability to merge and split entangled pair flows. At the same time it is simple enough that the control plane does not have to make any difficult routing decisions allowing us to focus our evaluation on the quantum data plane processes. Our simulation is based on a simplified model of nitrogen vacancy centre repeater platform [1, 8, 20, 43, 48, 67, 70, 82, 96]. We simplify the model by allowing for arbitrary quantum gates and increasing the number of communication qubits from one per node to two per link (not shared between links). The exact hardware parameters used are listed in Appendix B. For the entire evaluation except for Sec. 5.3 we consider parameters that are slightly better than currently achievable. The parameters were chosen to produce higher fidelities, but retain rates comparable to current hardware. The links between the nodes are 2 m in length and we do not convert the photons to telecom wavelength. We set the cutoff timeout to the time it takes a link-pair to lose approximately 1.5% of its initial fidelity. We run each simulation 100 times and calculate the average values of the quantities. Error bars are not shown as they are comparable to, or smaller than, the plot markers, unless stated otherwise.

5.1 Throughput and Latency

To gain some intuition about the protocol, before we study the effect of major decoherence, we evaluate it on devices with long memory lifetimes of one minute (current record on nitrogen vacancy platform not connected to a network [8]). We first investigate how the protocol shares resources in the network when multiple VCs have

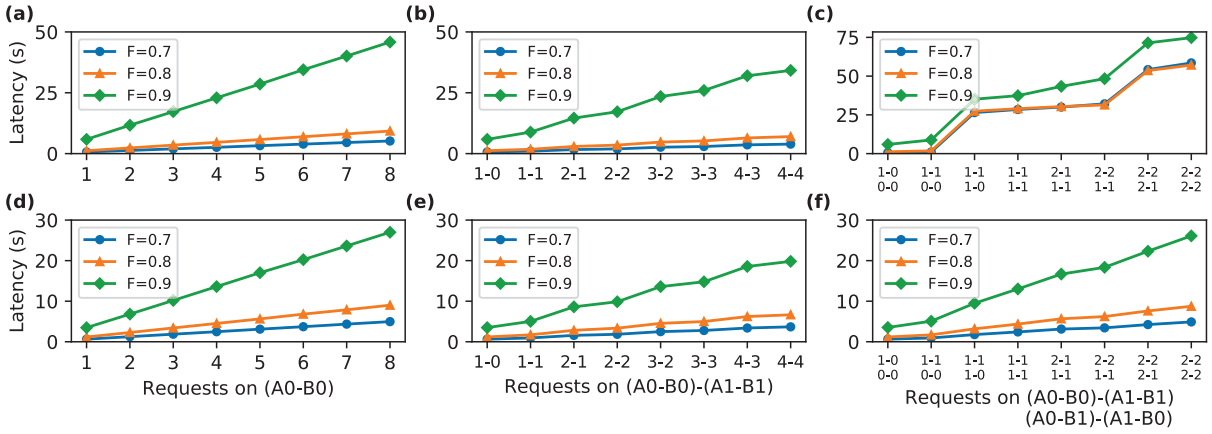


Figure 8: Average latency of requests on the A0-B0 circuit when 1–8 simultaneous requests, each for 100 pairs, are issued across (a,d) 1 circuit (A0-B0), (b,e) 2 circuits (A0-B0, A1-B1), and (c,f) 4 circuits (A0-B0, A1-B1, A0-B1, A1-B0). We consider a long (a-c) and short (d-f) cutoff time (see main text). Linear growth in (a,b,d,e) shows that circuits are efficiently shared across multiple requests. A shorter cutoff allows the routing algorithm to use a tighter bound on the decoherence and thus to relax the fidelity requirements on each link improving their rates. In (c) the 4 circuits struggle to share the bottleneck link when the cutoff time is long. Our scheduling algorithm is too simple and often generates pairs which do not have a matching pair on the same circuit on another link. Reducing the cutoff time (f) alleviates this problem as pairs that cannot be swapped are discarded faster.

to compete for resources. We investigate the end-to-end latency of multiple requests issued across multiple circuits that all share the MA-MB bottleneck link. We simultaneously issue between 1–8 requests for 100 pairs each split across up to four circuits. We consider three scenarios: one circuit only (A0-B0), two circuits (A0-B0, A1-B1), and four circuits (A0-B0, A1-B1, A0-B1, A1-B0). We vary two parameters: the end-to-end fidelity and the cutoff time. Normally we set the cutoff time to a value determined by the memory lifetime, but here we are using a relatively long-lived memory so we will also consider a "shorter cutoff" set to the time it takes for a link to have a 0.85 probability of generating a link-pair (see Fig. 5). The requests are equally distributed across the circuits in a round-robin manner. For example, in the four circuit scenario with six requests, the circuit A0-B0 handles the 1^{st} and 5^{th} requests, circuit A1-B1: the 2^{nd} and 6^{th}, A0-B1: the 3^{rd}, and A1-B0: the 4^{th}. All VCs are set up with the same max-LPR on the bottleneck link so they all get the same share of that link's time regardless of how many requests they carry. The average end-to-end request latency of requests issued on the A0-B0 circuit are shown in Fig. 8. It is immediately obvious that higher end-to-end fidelity pairs take longer to generate.

In Fig. 8 (a,b,d,e), we also see that when requests are split across up to two circuits, the latency scales linearly with the number of requests across the bottleneck link. However, Fig. 8c shows that the network struggles to multiplex four circuits (a "quantum congestion collapse"). Our scheduling algorithm is too simple: it assumes the links are independent, but they are not. A pair on an upstream link must wait for a pair on the downstream link to be generated for the same VC. Therefore, with four circuits and only two qubits per link, it can happen that no VC has matching pairs in their upstream and downstream queues and with no free qubits in the quantum memory the links cannot generate more pairs. The requests complete, because eventually the pairs decohere and are

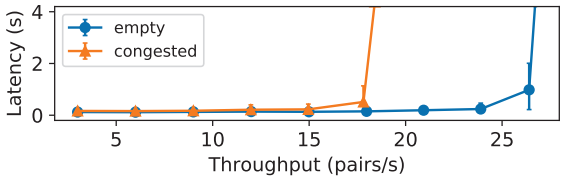


Figure 9: Average latency vs. throughput of A0-B0 circuit as we increase the rate of 3-pair requests over A0-B0. In the "empty" case, there is no other traffic in the network. In the "congested" case, there is a long running flow on A1-B1 at the same time competing for the bottleneck link. Error bars denote 5th and 95th percentile of the measured latency.

discarded. This problem can be solved by either not admitting this many circuits or by improving the scheduling and/or queuing at the nodes. Fig. 8f shows that reducing the cutoff value (effectively modifying the local scheduling strategy) alleviates the problem. A shorter cutoff improves throughput as any pairs that are using up memory slots without swapping are discarded sooner. Nevertheless, more research is required as to what the best scheduling strategy might be. We also note that the 1- and 2-circuit cases benefit from the shorter cutoff time. This is because a shorter cutoff allows the routing algorithm to use a tighter bound on the time qubits spend idling and as a result it can relax the fidelity requirements on each link leading to improved rates.

In the previous example, all requests were using their share of the bottleneck to the fullest. To evaluate how request latency scales with throughput we issue a series of smaller requests, each for three pairs, at an increasing frequency at regular intervals. This time, we only consider two circuits: A0-B0 and A1-B1 and we use the shorter cutoff. We send the small requests over the A0-B0 circuit and measure their latency and the VC's throughput. We run this scenario

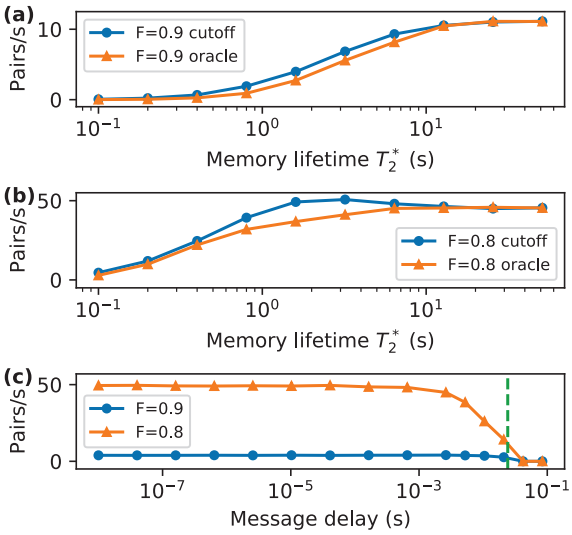


Figure 10: Robustness against decoherence. (a,b) show the effects of short memory lifetime on the throughput of the two competing circuits. Note that the F=0.9 with cutoff throughput becomes low, but not zero. (c) shows the effects of classical message delay (time from sending from one node to processing at next node). The dashed vertical line is the qubit cutoff value.

in an empty network (A1-B1 is idle) and in a congested network (A1-B1 is constantly busy with a long running request). We run the simulations for 50 simulated seconds and measure the latency of requests issued after the 40 s mark (a saturated equilibrium). Fig. 9 shows the average request latency vs the VC throughput. The latency is constant until the link saturates. The A0-B0 VC in the congested case saturates at more than half the value of the empty case. Whilst counter-intuitive, this has a simple explanation: the MA-MB link is shared by two circuits and thus generates pairs for each circuit slower than the non-congested links. Therefore, the other links will have a higher probability of having a pair ready for a swap by the time the MA-MB pair is ready.

5.2 Decoherence

We evaluate the two mechanisms for handling decoherence: the cutoff timer and not forcing quantum operations to wait for control messages. Here, we evaluate the protocol by running two circuits: A0-B0 for a fidelity of 0.9 and A1-B1 for a fidelity of 0.8. We use different fidelity values for the two VCs as lower fidelity requests suffer less from decoherence as the link-pairs are generated faster and can tolerate longer idle times. We issue one long running request for each circuit. The bottleneck link will round-robin between the two circuits allocating the same amount of time to each. Since the 0.8 fidelity circuit requires less time to generate each link-pair it will operate at a faster rate. We stop the simulation after 20 s of simulated time and calculate the throughput.

Cutoff timer Fig. 10 (a,b) shows the throughput of both VCs against the memory lifetime parameterised by T_2^*, the dephasing time of a qubit [62]. We see that as the memory lifetime decreases so does the throughput due to an increased rate of qubits being discarded. Higher fidelity VCs are affected more significantly as it takes longer to generate the link-pairs and thus they have a smaller window for swapping. In both cases we compare the performance of the protocol to a simpler protocol which instead of using a cutoff in the network discards end-to-end pairs that are below our required fidelity threshold. However, knowing which pairs are below the fidelity threshold is highly non-trivial as it is not possible to simply read it out from a pair. It would require a fidelity tracking mechanism that understands noise models of every device along the VC. Thus, the "simpler" protocol is implemented using an oracle: we use the simulation to give us the fidelity. The QNP does not use this backdoor mechanism as it is not available outside of simulations. We remark that Fig. 10 shows that the cutoff timer is more efficient than an end-node only strategy even with the physically impossible direct access to the fidelity.

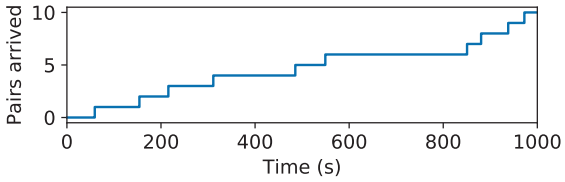


Figure 11: The number of pairs produced as a function of time on a near-future network. The protocol is able to deliver entanglement despite the limited resources.

Message delays As memory lifetimes get shorter, the effect of message delays becomes a concern. The QNP is designed such that quantum operations like swapping never block waiting for control messages. To demonstrate the effectiveness of this strategy in Fig. 10c we plot the throughput of the two VCs as we introduce artificial delays to increase the time between the sending of any QNP message to the moment that message is processed at the next node. We perform the simulations for a memory lifetime of about 1.6 s (approximately the middle of Fig. 10a) as it corresponds to achievable lifetimes in current hardware [1]. We see that the delay has no effect until it starts approaching the cutoff timeout. Once classical control messages are delayed beyond this threshold the delivered pairs have insufficient fidelity.

5.3 Near-Future Hardware Performance

So far, we considered a network that whilst not infeasible is still beyond our capabilities. We demonstrate that the protocol remains functional even with near-future hardware [1, 43] which highlights the timeliness of our work (hardware model and parameters are described in Appendix B). Fig. 11 shows the arrival times of 10 pairs requested over a linear network of three nodes with an inter-node separation of 25 km in a single simulation run. We request a fidelity of 0.5 which is sufficient to demonstrate quantum entanglement. In addition to more realistic parameters there are other constraints. The nodes have only one communication qubit and thus may only use one link at a time. As a result, a pair must be moved into storage before another pair can be created to swap with. Furthermore, the act of generating the next entangled pair further degrades the stored qubits due to the dephasing of nuclear spins [47]. Yet despite the enormous differences in the operating environment the

QNP remains functional: it exposes the right knobs to the control plane which an operator that understands the limitations can properly tune. As our routing protocol does not work well in this environment we manually populate the routing tables. We set the link-fidelities as high as possible to compensate for poor hardware quality and the nuclear dephasing and we tune the cutoff timer to ensure we meet the end-to-end fidelity threshold.

6 DISCUSSION

In this paper we have proposed a connection-oriented quantum data plane protocol for delivering end-to-end entanglement across a quantum network. However, whilst our work marks an important step on the way to large-scale quantum networks it is only one component of a complete quantum network architecture. Here, we briefly outline possible future directions of work.

QNP services We have designed the QNP using a VC approach inspired by MPLS as a building block for more complex quantum network services such as the entanglement distillation example described in Sec. 4.3. Other potential services include (i) services inspired by classical MPLS such as multipath support or failure recovery and (ii) services that take advantage of new features that are not present in classical networks such as the ability to pre-generate and store entangled pairs in preparation for future demand [16].

Control plane design In our paper we focused entirely on the quantum data plane and considered only a simplified control plane. Control plane protocols are also an emerging field in quantum network research, especially in the area of routing [12, 15, 16, 39, 40, 44, 56, 79, 80, 87]. However, more work is needed for a complete quantum network control plane. In particular, there is scope for further work on resource reservation, signalling, and more generally traffic engineering in quantum networks. Furthermore, there is also the question of software architecture for control planes: whether it is distributed or centralised. For example, software-defined architectures have been considered for QKD networks [2] and more recently have also been proposed for quantum repeater networks [53].

Relation to Internet protocol design It has been shown that classical network protocol stacks may be holistically analysed and systematically designed as distributed solutions to some optimisation problems (i.e. generalised network utility maximisation) [17]. It is conceivable that it is also possible to apply a similar "Layering as Optimisation Decomposition" approach to the quantum network protocol stack to improve its design.

Heterogeneous networks In this paper we focused on homogeneous networks based on a single hardware platform as that is the focus for near-future experimental work. However, a future quantum internet will inevitably consist a wide variety of physical platforms resulting in very different parameters for decoherence and quantum state fidelity for the quantum nodes and links. Therefore, more work is needed to understand the performance of quantum network protocols on hybrid quantum networks.

7 RELATED WORK

Quantum data plane protocols Three other proposals for end-to-end entanglement generation protocols that operate within our definition of a quantum data plane have been put forward [42, 57, 95]. Ref. [95] proposes a scheme inspired by classical UDP/TCP based on quantum error correction which is currently beyond hardware capabilities both in terms of required state quality and number of qubits. Ref. [42] does not consider decoherence. Ref. [57] combines what we would define as a quantum data plane protocol and a signalling protocol into one "RuleSet" based protocol, but the authors only study two-node networks with a single link.

Repeater chain protocols Since many long-distance links in the quantum internet will be built by chaining many quantum repeaters, protocols for such constructions have received significant attention [9, 10, 18, 29, 31, 37, 49, 58, 71, 73, 75, 76, 86]. However, these protocols are limited in scope to individual chains: they cannot handle non-linear topologies and do not have mechanisms for merging and splitting flows. Nevertheless, since a circuit in our network protocol is in some ways like a repeater chain, we use many ideas from this line of research, such as cutoff times [49, 55, 71, 73].

Network stacks Our paper fits into the network stack architecture proposed in Ref. [22]. The authors in Ref. [22] have also designed a link layer protocol, but they did not develop a network layer protocol. A complementary functional allocation for a quantum network stack for entanglement distillation also exists [5, 84, 88] though no concrete protocols have been given. An alternative outline for a quantum network stack has also been put forward in Ref. [65], but it does not account for many crucial low-level details such as hardware imperfections or classical control.

8 CONCLUSIONS

In this paper we have taken another step towards large-scale quantum networks. We have designed a quantum data plane network protocol for creating long-distance end-to-end entangled pairs, the key resource for distributed quantum applications. Quantum networks are complex systems and will require sophisticated resource management and scheduling strategies. We designed the Quantum Network Protocol to be the building block for constructing such higher-level services much like MPLS and IP datagrams have been for classical networks. We have ensured the protocol is efficient despite the extreme noise intrinsic to quantum systems by leveraging virtual circuits, building upon a robust link layer protocol, and adopting a cutoff timer. We also ensure that our protocol is scalable and can remain usable in the future once more capable hardware becomes available by leaving out tasks that require detailed knowledge of the hardware parameters of the nodes and links in the network to supporting protocols. This allows the core building block, the Quantum Network Protocol, to remain the same whilst giving the control plane the flexibility to evolve together with the network capabilities and requirements.

ACKNOWLEDGMENTS

We would like to thank Matthew Skrzypczyk, Carlo Delle Donne, Przemysław Pawełczak, Tim Coopmans, and Bruno Rijsman for the many useful discussions that helped us in this work and their detailed feedback on earlier versions of this draft. We would also like to acknowledge Kaushik Chakraborty and Kenneth Goodenough for further technical discussions.

The authors acknowledge funding received from the EU Flagship on Quantum Technologies, Quantum Internet Alliance (No. 820445), an ERC Starting Grant (SW), and an NWO VIDI Grant (SW).

REFERENCES

[1] Mohamed H Abobeih, Julia Cramer, Michiel A Bakker, Norbert Kalb, Matthew Markham, Daniel J Twitchen, and Tim H Taminiau. 2018. One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment. *Nature communications* 9, 1 (2018), 1–8.

[2] Alejandro Aguado, Victor López, Juan Pedro Brito, Antonio Pastor, Diego R López, and Vicente Martin. 2020. Enabling Quantum Key Distribution Networks via Software-Defined Networking. In *2020 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 1–5.

[3] Dorit Aharonov, Amnon Ta-Shma, Umesh V Vazirani, and Andrew C Yao. 2000. Quantum bit escrow. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. 705–714.

[4] Quantum Internet Alliance. 2018. https://quantum-internet.team

[5] Luciano Aparicio, Rodney Van Meter, and Hiroshi Esaki. 2011. Protocol design for quantum repeater networks. In *Proceedings of the 7th Asian Internet Engineering Conference*. 73–80.

[6] Michael Ben-Or and Avinatan Hassidim. 2005. Fast quantum Byzantine agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. 481–485.

[7] Charles H Bennett and Gilles Brassard. 1984. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the International Conference on Computers, Systems and Signal Processing*.

[8] C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau. 2019. A Ten-Qubit Solid-State Spin Register with Quantum Memory up to One Minute. *Physical Review X* 9 (Sep 2019), 031045. Issue 3.

[9] Sebastiaan Brand, Tim Coopmans, and David Elkouss. 2020. Efficient computation of the waiting time and fidelity in quantum repeater chains. *IEEE Journal on Selected Areas in Communications* (2020).

[10] H-J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. 1998. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters* 81, 26 (1998), 5932.

[11] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. 2009. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 517–526.

[12] Marcello Caleffi. 2017. Optimal routing for quantum networks. *IEEE Access* 5 (2017), 22299–22312.

[13] Marcello Caleffi and Angela Sara Cacciapuoti. 2020. Quantum switch for the quantum internet: Noiseless communications through noisy channels. *IEEE Journal on Selected Areas in Communications* 38, 3 (2020), 575–588.

[14] André Chailloux and Iordanis Kerenidis. 2011. Optimal bounds for quantum bit commitment. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE, 354–362.

[15] Kaushik Chakraborty, David Elkouss, Bruno Rijsman, and Stephanie Wehner. 2020. Entanglement Distribution in a Quantum Network, a Multi-Commodity Flow-Based Approach. (2020). arXiv:2005.14304 [quant-ph]

[16] Kaushik Chakraborty, Filip Rozpedek, Axel Dahlberg, and Stephanie Wehner. 2019. Distributed Routing in a Quantum Internet. (2019). arXiv:1907.11630 [quant-ph]

[17] Mung Chiang, Steven H Low, A Robert Calderbank, and John C Doyle. 2007. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE* 95, 1 (2007), 255–312.

[18] Juan Ignacio Cirac, Peter Zoller, H Jeff Kimble, and Hideo Mabuchi. 1997. Quantum state transfer and entanglement distribution among distant nodes in a quantum network. *Physical Review Letters* 78, 16 (1997), 3221.

[19] Rachel Courtland. 2016. China's 2,000-km quantum link is almost complete [News]. *IEEE Spectrum* 53, 11 (2016), 11–12.

[20] Julia Cramer, Norbert Kalb, M Adriaan Rol, Bas Hensen, Machiel S Blok, Matthew Markham, Daniel J Twitchen, Ronald Hanson, and Tim H Taminiau. 2016. Repeated quantum error correction on a continuously encoded qubit by real-time feedback. *Nature communications* 7, 1 (2016), 1–7.

[21] Claude Crépeau, Daniel Gottesman, and Adam Smith. 2002. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 643–652.

[22] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. 2019. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication* (Beijing, China) *(SIGCOMM '19)*. ACM, New York, NY, USA, 159–173.

[23] Ivan B Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. 2007. Secure identification and QKD in the bounded-quantum-storage model. In *Annual International Cryptology Conference*. Springer, 342–359.

[24] Ivan B Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. 2008. Cryptography in the bounded-quantum-storage model. *SIAM J. Comput.* 37, 6 (2008), 1865–1890.

[25] Vasil S Denchev and Gopal Pandurangan. 2008. Distributed quantum computing: A new frontier in distributed systems or science fiction? *ACM SIGACT News* 39, 3 (2008), 77–95.

[26] Eleni Diamanti, Hoi-Kwong Lo, Bing Qi, and Zhiliang Yuan. 2016. Practical challenges in quantum key distribution. *npj Quantum Information* 2 (2016), 16025.

[27] Jonathan P Dowling and Gerard J Milburn. 2003. Quantum technology: the second quantum revolution. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 361, 1809 (2003), 1655–1674.

[28] Anaïs Dréau, Anna Tchebotareva, Aboubakr El Mahdaoui, Cristian Bonato, and Ronald Hanson. 2018. Quantum frequency conversion of single photons from a nitrogen-vacancy center in diamond to telecommunication wavelengths. *Physical Review Applied* 9, 6 (2018), 064031.

[29] L-M Duan, MD Lukin, J Ignacio Cirac, and Peter Zoller. 2001. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* 414, 6862 (2001), 413.

[30] Wolfgang Dür and Hans J Briegel. 2007. Entanglement purification and quantum error correction. *Reports on Progress in Physics* 70, 8 (2007), 1381.

[31] W Dür, H-J Briegel, JI Cirac, and P Zoller. 1999. Quantum repeaters based on entanglement purification. *Physical Review A* 59, 1 (1999), 169.

[32] Artur K Ekert. 1991. Quantum cryptography based on Bell's theorem. *Physical Review Letters* 67, 6 (1991), 661.

[33] A Extance. 2017. *Fibre Systems*. www.fibre-systems.com/feature/quantum-security

[34] Joseph F Fitzsimons and Elham Kashefi. 2017. Unconditionally verifiable blind quantum computation. *Physical Review A* 96, 1 (2017), 012303.

[35] Austin G Fowler, David S Wang, Charles D Hill, Thaddeus D Ladd, Rodney Van Meter, and Lloyd CL Hollenberg. 2010. Surface code quantum communication. *Physical Review Letters* 104, 18 (2010), 180503.

[36] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. 2004. Quantum-enhanced measurements: beating the standard quantum limit. *Science* 306, 5700 (2004), 1330–1336.

[37] Alexander M Goebel, Claudia Wagenknecht, Qiang Zhang, Yu-Ao Chen, Kai Chen, Jörg Schmiedmayer, and Jian-Wei Pan. 2008. Multistage entanglement swapping. *Physical Review Letters* 101, 8 (2008), 080403.

[38] Daniel Gottesman, Thomas Jennewein, and Sarah Croke. 2012. Longer-baseline telescopes using quantum repeaters. *Physical Review Letters* 109, 7 (2012), 070503.

[39] Laszlo Gyongyosi and Sandor Imre. 2017. Entanglement-gradient routing for quantum networks. *Scientific reports* 7, 1 (2017), 14255.

[40] Laszlo Gyongyosi and Sandor Imre. 2018. Decentralized base-graph routing for the quantum internet. *Physical Review A* 98, 2 (2018), 022310.

[41] Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenberg, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, Waldimar Amaya, Valerio Pruneri, Morgan W Mitchell, Michael Markham, Daniel J Twitchen, David Elkouss, Stephanie Wehner, Tim H Taminiau, and Ronald Hanson. 2015. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* 526, 7575 (2015), 682.

[42] Bernardo A Huberman and Bob Lund. 2019. A Quantum Router For The Entangled Web. *Information Systems Frontiers* (2019), 1–7.

[43] Peter C Humphreys, Norbert Kalb, Jaco PJ Morits, Raymond N Schouten, Raymond FL Vermeulen, Daniel J Twitchen, Matthew Markham, and Ronald Hanson. 2018. Deterministic delivery of remote entanglement on a quantum network. *Nature* 558, 7709 (2018), 268.

[44] Sandor Imre and Laszlo Gyongyosi. 2012. *Advanced quantum communications: an engineering approach*. John Wiley & Sons.

[45] Takahiro Inagaki, Nobuyuki Matsuda, Osamu Tadanaga, Masaki Asobe, and Hiroki Takesue. 2013. Entanglement distribution over 300 km of fiber. *Optics express* 21, 20 (2013), 23241–23249.

[46] IV Inlek, C Crocker, M Lichtman, K Sosnova, and C Monroe. 2017. Multispecies trapped-ion node for quantum networking. *Physical Review Letters* 118, 25 (2017), 250502.

[47] Norbert Kalb, Peter C Humphreys, JJ Slim, and Ronald Hanson. 2018. Dephasing mechanisms of diamond-based nuclear-spin memories for quantum networks. *Physical Review A* 97, 6 (2018), 062330.

[48] Norbert Kalb, Andreas A Reiserer, Peter C Humphreys, Jacob JW Bakermans, Sten J Kamerling, Naomi H Nickerson, Simon C Benjamin, Daniel J Twitchen, Matthew Markham, and Ronald Hanson. 2017. Entanglement distillation between solid-state quantum network nodes. *Science* 356, 6341 (2017), 928–932.

[49] Sumeet Khatri, Corey T Matyas, Aliza U Siddiqui, and Jonathan P Dowling. 2019. Practical figures of merit and thresholds for entanglement distribution in quantum networks. *Physical Review Research* 1, 2 (2019), 023032.

[50] Peter Komar, Eric M Kessler, Michael Bishof, Liang Jiang, Anders S Sørensen, Jun Ye, and Mikhail D Lukin. 2014. A quantum network of clocks. *Nature Physics* 10, 8 (2014), 582.

[51] Wojciech Kozlowski, Axel Dahlberg, and Stephanie Wehner. 2020. [CoNEXT 2020] Artifacts for "Designing a Quantum Network Protocol". https://doi.org/10.34894/2P1P91

[52] Wojciech Kozlowski, Axel Dahlberg, and Stephanie Wehner. 2020. Designing a Quantum Network Protocol. (2020). arXiv:2010.02575 [cs.NI]

[53] Wojciech Kozlowski, Fernando Kuipers, and Stephanie Wehner. 2020. A P4 Data Plane for the Quantum Internet. In *Proceedings of the 3rd P4 Workshop in Europe (EuroP4'20)*.

[54] Wojciech Kozlowski and Stephanie Wehner. 2019. Towards Large-Scale Quantum Networks. In *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*. ACM.

[55] Boxi Li, Tim Coopmans, and David Elkouss. 2020. Efficient optimization of cut-offs in quantum repeater chains. (2020). arXiv:2005.04946 [quant-ph]

[56] Changhao Li, Tianyi Li, Yi-Xiang Liu, and Paola Cappellaro. 2020. Effective routing design for remote entanglement generation on quantum networks. (2020). arXiv:2001.02204 [quant-ph]

[57] Takaaki Matsuo, Clément Durand, and Rodney Van Meter. 2019. Quantum link bootstrapping using a RuleSet-based communication protocol. *Physical Review A* 100, 5 (2019), 052320.

[58] William J Munro, Koji Azuma, Kiyoshi Tamaki, and Kae Nemoto. 2015. Inside quantum repeaters. *IEEE Journal of Selected Topics in Quantum Electronics* 21, 3 (2015), 78–90.

[59] William J Munro, Ashley M Stephens, Simon J Devitt, Keith A Harrison, and Kae Nemoto. 2012. Quantum communication without the necessity of quantum memories. *Nature Photonics* 6, 11 (2012), 777.

[60] Sreraman Muralidharan, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2014. Ultrafast and fault-tolerant quantum communication across long distances. *Physical Review Letters* 112, 25 (2014), 250501.

[61] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2016. Optimal architectures for long distance quantum communication. *Scientific reports* 6 (2016), 20463.

[62] Michael A Nielsen and Isaac L Chuang. 2000. Quantum information and quantum computation. *Cambridge: Cambridge University Press* 2, 8 (2000), 23.

[63] Mihir Pant, Hari Krovi, Don Towsley, Leandros Tassiulas, Liang Jiang, Prithwish Basu, Dirk Englund, and Saikat Guha. 2019. Routing entanglement in the quantum internet. *npj Quantum Information* 5, 1 (2019), 25.

[64] M Peev, C Pacher, R Alléaume, C Barreiro, J Bouda, W Boxleitner, T Debuisschert, E Diamanti, M Dianati, J F Dynes, S Fasel, S Fossier, M Fürst, J-D Gautier, O Gay, N Gisin, P Grangier, A Happe, Y Hasani, M Hentschel, H Hübel, G Humer, T Länger, M Legré, R Lieger, J Lodewyck, T Lorünser, N Lütkenhaus, A Marhold, T Matyus, O Maurhart, L Monat, S Nauerth, J-B Page, A Poppe, E Querasser, G Ribordy, S Robyr, L Salvail, A W Sharpe, A J Shields, D Stucki, M Suda, C Tamas, T Themel, R T Thew, Y Thoma, A Treiber, P Trinkler, R Tualle-Brouri, F Vannel, N Walenta, H Weier, H Weinfurter, I Wimberger, Z L Yuan, H Zbinden, and A Zeilinger. 2009. The SECOQC quantum key distribution network in Vienna. *New Journal of Physics* 11, 7 (jul 2009), 075001.

[65] Alexander Pirker and Wolfgang Dür. 2019. A quantum network stack and protocols for reliable entanglement-based networks. *New Journal of Physics* 21, 3 (2019), 033003.

[66] QuTech. 2020. *NetSquid*. https://netsquid.org/

[67] Andreas Reiserer, Norbert Kalb, Machiel S Blok, Koen JM van Bemmelen, Tim H Taminiau, Ronald Hanson, Daniel J Twitchen, and Matthew Markham. 2016. Robust quantum-network memory using decoherence-protected subspaces of nuclear spins. *Physical Review X* 6, 2 (2016), 021040.

[68] Andreas Reiserer and Gerhard Rempe. 2015. Cavity-based quantum networks with single atoms and optical photons. *Reviews of Modern Physics* 87, 4 (2015), 1379.

[69] Jérémy Ribeiro and Frédéric Grosshans. 2015. A Tight Lower Bound for the BB84-states Quantum-Position-Verification Protocol. (2015). arXiv:1504.07171 [quant-ph]

[70] Daniel Riedel, Immo Söllner, Brendan J Shields, Sebastian Starosielec, Patrick Appel, Elke Neu, Patrick Maletinsky, and Richard J Warburton. 2017. Deterministic enhancement of coherent photon generation from a nitrogen-vacancy center in ultrapure diamond. *Physical Review X* 7, 3 (2017), 031040.

[71] Filip Rozpędek, Kenneth Goodenough, Jeremy Ribeiro, Norbert Kalb, V Caprara Vivoli, Andreas Reiserer, Ronald Hanson, Stephanie Wehner, and David Elkouss. 2018. Parameter regimes for a single sequential quantum repeater. *Quantum Science and Technology* 3, 3 (2018), 034002.

[72] Filip Rozpędek, Thomas Schiet, Le Phuc Thinh, David Elkouss, Andrew C. Doherty, and Stephanie Wehner. 2018. Optimizing practical entanglement distillation. *Phys. Rev. A* 97 (Jun 2018), 062333. Issue 6. https://doi.org/10.1103/PhysRevA.97.062333

[73] Filip Rozpędek, Raja Yehia, Kenneth Goodenough, Maximilian Ruf, Peter C. Humphreys, Ronald Hanson, Stephanie Wehner, and David Elkouss. 2019. Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission. *Physical Review A* 99 (May 2019), 052330. Issue 5. https://doi.org/10.1103/PhysRevA.99.052330

[74] Louis Salvail, Momtchil Peev, Eleni Diamanti, Romain Alléaume, Norbert Lütkenhaus, and Thomas Länger. 2010. Security of trusted repeater quantum key distribution networks. *Journal of Computer Security* 18, 1 (2010), 61–87.

[75] Nicolas Sangouard, Christoph Simon, Hugues De Riedmatten, and Nicolas Gisin. 2011. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics* 83, 1 (2011), 33.

[76] Siddhartha Santra, Liang Jiang, and Vladimir S Malinovsky. 2019. Quantum repeater architecture with hierarchically optimized memory buffer times. *Quantum Science and Technology* 4, 2 (2019), 025010.

[77] Masahide Sasaki, M Fujiwara, H Ishizuka, W Klaus, K Wakui, M Takeoka, S Miki, T Yamashita, Z Wang, A Tanaka, K Yoshino, Y Nambu, S Takahashi, A Tajima, A Tomita, T Domeki, T Hasegawa, Y Sakai, H Kobayashi, T Asai, K Shimizu, T Tokura, T Tsurumaru, M Matsui, T Honjo, K Tamaki, H Takesue, Y Tokura, J F Dynes, A R Dixon, A W Sharpe, Z L Yuan, A J Shields, S Uchikoga, M Legréx, S Robyr, P Trinkler, L Monat, J-B Page, G Ribordy, A Poppe, A Allacher, O Maurhart, T Länger, M Peev, others, and A. Zeilinger. 2011. Field test of quantum key distribution in the Tokyo QKD Network. *Optics express* 19, 11 (2011), 10387–10409.

[78] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. 2009. The security of practical quantum key distribution. *Reviews of Modern Physics* 81 (Sep 2009), 1301–1350. Issue 3.

[79] Eddie Schoute, Laura Mancinska, Tanvirul Islam, Iordanis Kerenidis, and Stephanie Wehner. 2016. Shortcuts to quantum network routing. (2016). arXiv:1610.05238 [cs.NI]

[80] Shouqian Shi and Chen Qian. 2020. Concurrent Entanglement Routing for Quantum Networks: Model and Designs. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 62–75.

[81] D Stucki, M Legré, F Buntschu, B Clausen, N Felber, N Gisin, L Henzen, P Junod, G Litzistorf, P Monbaron, L Monat, J-B Page, D Perroud, G Ribordy, A Rochas, S Robyr, J Tavares, R Thew, P Trinkler, S Ventura, R Voirol, N Walenta, and H Zbinden. 2011. Long-term performance of the SwissQuantum quantum key distribution network in a field environment. *New Journal of Physics* 13, 12 (dec 2011), 123001.

[82] Tim H Taminiau, Julia Cramer, Toeno van der Sar, Viatcheslav V Dobrovitski, and Ronald Hanson. 2014. Universal control and error correction in multi-qubit spin registers in diamond. *Nature nanotechnology* 9, 3 (2014), 171.

[83] Anna Tchebotareva, Sophie L. N. Hermans, Peter C. Humphreys, Dirk Voigt, Peter J. Harmsma, Lun K. Cheng, Ad L. Verlaan, Niels Dijkhuizen, Wim de Jong, Anaïs Dréau, and Ronald Hanson. 2019. Entanglement between a Diamond Spin Qubit and a Photonic Time-Bin Qubit at Telecom Wavelength. *Physical Review Letters* 123 (Aug 2019), 063601. Issue 6.

[84] Rodney Van Meter. 2012. Quantum networking and internetworking. *IEEE Network* 26, 4 (2012), 59–64.

[85] Rodney Van Meter. 2014. *Quantum networking*. John Wiley & Sons.

[86] Rodney Van Meter, Thaddeus D Ladd, William J Munro, and Kae Nemoto. 2008. System design for a long-line quantum repeater. *IEEE/ACM Transactions on Networking* 17, 3 (2008), 1002–1013.

[87] Rodney Van Meter, Takahiko Satoh, Thaddeus D Ladd, William J Munro, and Kae Nemoto. 2013. Path selection for quantum repeater networks. *Networking Science* 3, 1-4 (2013), 82–95.

[88] Rodney Van Meter and Joe Touch. 2013. Designing quantum repeater networks. *IEEE Communications Magazine* 51, 8 (2013), 64–71.

[89] Gayane Vardoyan, Saikat Guha, Philippe Nain, and Don Towsley. 2019. On the Capacity Region of Bipartite and Tripartite Entanglement Switching and Key Distribution. In *QCRYPT 2019*.

[90] Gayane Vardoyan, Saikat Guha, Philippe Nain, and Don Towsley. 2019. On the stochastic analysis of a quantum entanglement switch. *ACM SIGMETRICS Performance Evaluation Review* 47, 2 (2019), 27–29.

[91] Shuang Wang, Wei Chen, Zhen-Qiang Yin, Hong-Wei Li, De-Yong He, Yu-Hu Li, Zheng Zhou, Xiao-Tian Song, Fang-Yi Li, Dong Wang, Hua Chen, Yun-Guang Han, Jing-Zheng Huang, Jun-Fu Guo, Peng-Lei Hao, Mo Li, Chun-Mei Zhang, Dong Liu, Wen-Ye Liang, Chun-Hua Miao, Ping Wu, Guang-Can Guo, and Zheng-Fu Han. 2014. Field and long-term demonstration of a wide area quantum key distribution network. *Optics Express* 22, 18 (Sep 2014), 21739–21756.

[92] Stephanie Wehner, David Elkouss, and Ronald Hanson. 2018. Quantum internet: A vision for the road ahead. *Science* 362, 6412 (2018).

[93] Stephanie Wehner, Christian Schaffner, and Barbara M Terhal. 2008. Cryptography from noisy storage. *Physical Review Letters* 100, 22 (2008), 220502.

[94] Reinhard F. Werner. 1989. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Phys. Rev. A* 40 (Oct 1989), 4277–4281. Issue 8. https://doi.org/10.1103/PhysRevA.40.4277

[95] Nengkun Yu, Ching-Yi Lai, and Li Zhou. 2019. Protocols for Packet Quantum Network Intercommunication. (2019). arXiv:1903.10685 [quant-ph]

[96] Sebastian Zaske, Andreas Lenhard, Christian A Keßler, Jan Kettler, Christian Hepp, Carsten Arend, Roland Albrecht, Wolfgang-Michael Schulz, Michael Jetter, Peter Michler, et al. 2012. Visible-to-telecom quantum frequency conversion of light from a single quantum emitter. *Physical Review Letters* 109, 14 (2012), 147404.

APPENDIX

A ARTIFACTS

The source code for the implementation of the Quantum Network Protocol in NetSquid [66] and the raw data used to produce the plots in this paper has been made available at https://doi.org/10.34894/2P1P91 [51].

The artifact directory contains:

- the source code to run the simulations and reproduce the results,
- the raw data used to plot the figures in the paper.

The zip file contains a directory within which are contained:

- README.md — description of the contents as well as instructions to install and set up the simulations,
- EXPERIMENTS.md — instructions to run the experiments described in this paper and reproduce all the data.

Note that compatibility on all platforms is not guaranteed. For this reason a Dockerfile is also provided which should make it possible to execute the artifacts on all platforms that support Docker containers. Instructions for using the container are included in README.md.

B HARDWARE PARAMETERS

The simulations in this paper are based on the nitrogen vacancy centre (NV-centre) platform for quantum repeaters. Experimental results for this platform are available in Refs. [1, 8, 20, 43, 48, 67, 70, 82, 96]. An in-depth introduction to the quantum physics and operation of this platform including noise modelling and the definitions of the different hardware parameters can be found in Appendix D of Ref. [22]. Parameter values used for simulations in this paper are given in Tables 1 and 2. The near-term values are based on references to the aforementioned experimental papers and Ref. [22].

Simulation parameters All of the simulations in the paper except for the near-future hardware example were done in an optimistic configuration with hardware parameters beyond what is currently possible in the laboratory. These parameters are shown in Tables 1 and 2 where they are also compared to the currently achievable parameters. Additionally, we made a few simplifications that go beyond hardware parameter values.

We did not distinguish between so-called communication (electron) qubits and memory (carbon) qubits. In an NV-centre architecture only one qubit, the communication qubit, can participate in link-pair generation at any one time. This means that only one link of every node can be active at any one time. The quantum network protocol, as demonstrated in the near-future hardware simulations, can cope with this scenario, but for larger networks requires a more sophisticated resource management and scheduling approach which is beyond the scope of this work. Therefore, for the purposes of our simulations (except for the near-future hardware case) all qubits are treated as communication (electron) qubits and can participate in link-pair generation.

Furthermore, a major source of noise in NV-centres is the dephasing of nuclear spins (memory qubits) due to the resetting of the communication qubit during entanglement generation attempts. Since we only consider communication qubits in our simulations we also do not consider this noise in our simulations. However, from the point of view of our protocol this noise can be treated like normal decoherence — it is a process that degrades the quality of idle qubits over time. Nevertheless, this requires a more sophisticated approach to correctly calculate the cutoff timeout values for idle qubits which is also beyond the scope of this paper. However, our near-future hardware example in the main text, where we hand-picked a timeout value, shows that the cutoff time of the protocol is a suitable mechanism for handling this noise.

Optical fibres The channels that carry photons and classical messages between the nodes (both classical and quantum channels) are standard telecom optical fibres. For the near-term hardware simulation we considered fibres of 25 km length between each node which requires frequency conversion for the photons used in entanglement generation (to achieve 0.5 db/km losses). For the rest of the simulations we used parameters closer to a lab scenario, 2 m fibres, as they do not need frequency conversion (losses of 5 dB/km) leading to faster generation rates. We do not simulate losses for classical messages, because (i) they are extremely low, (ii) protocol communication happens over TCP so lost packets would just be resent, (iii) in the main text we already consider the effects of arbitrary processing and communication delays which can arise from TCP retransmission.

	Simulation		Near-term (Fig. 11)	
	Fidelity	Duration	Fidelity	Duration
Electron single-qubit gate	1.0	5 ns	1.0	5 ns
Two-qubit gate (E-C controlled $\sqrt{X}$-gate for near-term)	0.998	500 μs	0.992	500 μs
Carbon Rot-Z gate	—	—	1.0	20 μs
Electron initialisation in $\|0\rangle$	0.99	2 μs	0.99	2 μs
Carbon initialisation in $\|0\rangle$	—	—	0.95	300 μs
Electron readout $\|0\rangle$	0.998	3.7 μs	0.95	3.7 μs
Electron readout $\|1\rangle$	0.998	3.7 μs	0.995	3.7 μs

Table 1: Quantum gate parameters. Explanation of each parameter can be found in Appendix D of Ref. [22].

	Simulation	Near-term (Fig. 11)
Electron T_1	>1 h	>1 h
Electron T_2^*	60 s	1.46 s
Carbon T_1	—	> 6 m
Carbon T_2^*	—	60 s
$\Delta\omega$	—	$2\pi \times 377$ kHz
τ_d	—	82 ns
τ_w	25 ns	25 ns
τ_e	6.0 ns	6.48 ns
$\Delta\phi$	2.0°	10.6°
$p_{\text{double\_excitation}}$	0.00	0.04
$p_{\text{zero\_phonon}}$	0.75	0.46
Collection efficiency	$20.0 \cdot 10^{-3}$	$4.38 \cdot 10^{-3}$
Dark count rate	20 s^{-1}	20 s^{-1}
$p_{\text{detection}}$	0.8	0.8
Visibility (distinguishability)	1.0	0.9

Table 2: Other hardware parameters. Explanation of each parameter can be found in Appendix D of Ref. [22].

An Architecture for Meeting Quality-of-Service Requirements in Multi-User Quantum Networks

ABSTRACT

Quantum communication can enhance internet technology by enabling novel applications that are provably impossible classically. The successful execution of such applications relies on the generation of quantum entanglement between different users of the network that meets stringent performance requirements. Alongside traditional metrics such as throughput and jitter, one must ensure the generated entanglement is of sufficiently high quality. Meeting such performance requirements demands a careful orchestration of many devices in the network, giving rise to a fundamentally new scheduling problem. Furthermore, technological limitations of near-term quantum devices impose significant constraints on scheduling methods hoping to meet performance requirements. In this work, we propose the first end-to-end design of a centralized quantum network with multiple users that orchestrates the delivery of entanglement which meets quality-of-service (QoS) requirements of applications. We achieve this by using a centrally constructed schedule that manages usage of devices and ensures the coordinated execution of different quantum operations throughout the network. We use periodic task scheduling and resource-constrained project scheduling techniques, including a novel heuristic, to construct the schedules. Our simulations of four small networks using hardware-validated network parameters, and of a real-world fiber topology using futuristic parameters illustrate trade-offs between traditional and quantum performance metrics.

1 INTRODUCTION

Recent progress in developing networked quantum devices (see e.g. [28, 30, 41, 51, 62, 63]) motivates the emerging field of quantum network architecture. Quantum networks promise to significantly enhance internet technology by enabling new applications that are impossible to achieve using classical (non-quantum) communication [39, 60]. Key to enabling quantum applications is the creation of end-to-end entanglement between two nodes in the network. Entanglement is a special property of two quantum bits (qubits) held by two nodes in the quantum network. As such, one might think of entanglement as a form of virtual or - *entangled - link* between the two qubits [53].

Application performance in quantum networks depends on several dimensions of network service. On one hand, there are traditional performance metrics such as the *throughput* of entanglement delivery as well as *jitter* (variance in inter-delivery times) for more complex applications [16]. On the other hand, there is a genuinely quantum performance metric, namely the quality, or *fidelity*, of the entanglement delivered to users [16]. Meeting application requirements and maximizing network utility motivates the design of quantum network architectures that support quality-of-service (QoS) guarantees on the distributed entanglement.

Entanglement, or *entangled links*, may be established between quantum network devices that are directly connected via a physical medium such as optical fiber (see e.g. [30, 41, 63]), or free-space communication (see e.g. [62]). We refer to two such devices as *connected*. In multi-hop quantum networks, where not all devices are connected, entanglement distribution can be accomplished with the help of intermediary nodes using a procedure known as *entanglement swapping* (see Section 3.1). Such intermediary quantum devices are often referred to as a *quantum repeater*. In general, quantum repeater protocols that establish entanglement over long distances can be formed by combining several types of operations in addition to entanglement swapping (see Section 3.1). The fidelity requirement on entanglement distribution is satisfied by the exact combination of these operations, as allowed by the underlying quantum hardware. Throughput requirements are met by executing quantum repeater protocols frequently enough to distribute entanglement at the desired rate while jitter requirements are met by regulating the inter-delivery times of entanglement from the quantum repeater protocols.

Even if only two users in the network wish to communicate, the successful execution of a quantum repeater protocol requires the coordinated execution of different operations at intermediary nodes in the network (see Section 3.1). If many users wish to generate entanglement simultaneously, we also require coordination between the actions of two disjoint repeater protocols at the level of their component operations. This gives rise to a novel scheduling problem that is fundamental to the design of quantum networks.

What is more, near-term technological limitations impose very strict demands on any coordination mechanism hoping to meet QoS requirements. Specifically, near-term quantum hardware (sometimes also referred to as noisy intermediate-scale quantum devices, NISQ [49]) offers limited memory lifetimes (at most seconds [1, 9]), which means that entangled links cannot be stored for a long time. Furthermore, a limited storage space means that the number of entangled links

that can be stored simultaneously is small (only recently 2 [25]). These limitations impose both real-time and resource constraints on the creation of entangled links.

Here, we propose a novel time-division multiple access (TDMA) network architecture for quantum networks that supports QoS requirements of entanglement generation for applications. Our centralized architecture achieves this by encoding quantum repeater protocols into schedules that are distributed across the network. Fixed-duration time slots in the schedule encode the different operations of quantum repeater protocols. The encoded protocols are selected to succeed with high probability and meet fidelity requirements, while the schedule is constructed such that the frequency of entanglement delivery meets throughput and jitter requirements (see Section 5.2 for details). To this end, we introduce the novel problem of constructing schedules of quantum repeater protocols and provide several methods, including a new heuristic, for solving the resulting scheduling problem. We benchmark our new heuristic against existing heuristics adaptable to this setting on several small near-term quantum network topologies as well as a futuristic setting based on a real-world fiber topology in the Netherlands and show that comparable performance can be obtained while reducing runtime complexity. We additionally find that the choice of scheduling heuristic can be used to trade off higher network throughput for lower jitter. We emphasize that the use of a centralized architecture to coordinate entanglement distribution has no effect on quantum security applications (e.g. quantum key distribution [7, 21]) as the network in its entirety is treated as untrusted in their security analysis.

We design our architecture to fit into an existing quantum network stack, and build upon the previously proposed quantum network protocols [16]. It has the following defining properties: (1) encoded repeater protocols are connection-oriented and can be tailored per application, (2) the centrally constructed schedule provides contention-free usage of network devices, (3) dynamic update of the schedule allows the system to accompany network demands at runtime.

2 RELATED WORK

Several functional allocations of quantum network stacks have been proposed in [4, 56–58] that formulate layers based on specific protocols such as entanglement distillation. In contrast, Dahlberg et al. take a different approach in [16] and focus on the type of service each layer should provide. The authors complement their functional allocation with physical and link layer protocols that take practical considerations, such as hardware imperfections and communication overhead, into account. Kozlowski et al. [40] build upon this work and design a network layer protocol suitable for the network stack model of [16]. In [43], Matsuo et al. present and simulate a RuleSet-based quantum link bootstrapping protocol that may be used to install rules to provide flexibility when establishing connections in quantum networks.

Quantum network architectures that break down entanglement distribution into discrete time steps have been studied in [17, 47, 55]. In contrast to our paper, these works do not describe the network infrastructure that realizes their architecture nor do they detail any scheduling strategies for network coordination. Furthermore, they depend on the production of high quality entanglement between connected devices in order to meet high fidelity requirements between distant nodes in the network and as a result are not suitable for near-term networks that we consider here. In [3], Aparicio et al. evaluate the usage of time-division multiplexing (TDM) in comparison to other multiplexing strategies. Their TDM scheme differs from ours in that it assigns end-to-end flows to time slots without detailing the quantum repeater protocol operations to execute. As a result, their scheme does not coordinate the operation of connected devices that are limited to establishing one entangled link with another connected device at a time. Vardoyan et al. study the performance of a quantum network switch within a star topology in [59] and find that correctly configured scheduling policies can outperform TDM for smaller star topologies, but that the relative improvement reduces as the number of users grows.

Dynamic TDMA protocols have been studied in the context of many network technologies: ad-hoc networks [36], wireless ATM networks [23], wireless powered communication networks [38], and satellite communication [48]. An extensive amount of literature on TDMA schemes ranging from centralized to distributed models [29, 52] exists. In contrast, our TDMA architecture for quantum networks multiplexes the execution of operations for quantum repeater protocols, of which there may be many on a single node just to establish a single end-to-end entangled link, rather than multiplexing access to classical channels.

Construction of TDMA schedules is a long-studied problem in several types of networks. Traditionally, TDMA schedules are constructed in order to enable data transmission between nodes in networks over shared communication mediums. Studied methods include formulating schedule construction as a graph coloring problem [50] and using heuristics [37] or branch-and-bound search [8] techniques to compute a solution. These existing formulations are not suitable for NISQ device quantum networks as end-to-end transmission in classical networks may be achieved by preventing collisions on common communication channels whereas establishing end-to-end entanglement with sufficient fidelity requires coordinating operations along the entire path through the network.

A related problem to schedule construction is that of scheduling task graphs to parallel processor systems [2] where parallel programs represented by directed acyclic graphs are

allocated to a set of homogeneous processors. This also differs from our scheduling problem as operations in repeater protocols must be allocated to specific network nodes in order to meet QoS requirements.

3 DESIGN CONSIDERATIONS

Quantum networks pose several new challenges to address when designing the network's architecture. Design considerations for producing entangled links between connected nodes can be found in [16]. We hence focus mainly on the challenges posed by long-distance entanglement generation as relevant to the design of our TDMA architecture.

3.1 Devices and Establishing Entanglement

A quantum network consists of *end nodes* [60] that are connected to the network in order to run specific applications. In addition, a quantum network may include nodes, known as *repeater nodes*, that facilitate the generation of entanglement between two unconnected nodes. End nodes can also act as repeater nodes, but QoS demands only originate from applications at end nodes. Here, we focus primarily on quantum nodes (end nodes or repeaters) known as processing nodes. A processing node is a few-qubit quantum computer with an optical interface, of which there have been implementations in nitrogen vacancy (NV) centers in diamond [26], ion traps [44] and neutral atoms [28]. We emphasize however that our work can also be adapted to other platforms, including systems based on atomic ensembles [14].

To produce long-distance entanglement, near-term quantum repeater protocols [45] may employ a variety of operations (Fig. 1) which need to be scheduled in a coordinated manner (see e.g. Fig 2). Two directly connected nodes can establish an *elementary entangled link* between them (Fig. 1a). Entanglement generation at the physical layer is probabilistic, and will often require several attempts before an entangled link is created. Here, we focus on using a robust link layer protocol [16] based around a physical layer entanglement generation scheme that has a *heralding signal* confirming the success/failure. This link layer protocol allows for a deliberate trade-off between the fidelity and the throughput of generating elementary links, depending on the capabilities of the underlying hardware. Coordinating establishment of a single entangled link between connected nodes already requires timing synchronization and agreement [16] (up to ns precision using e.g. White Rabbit [54]).

In multi-hop quantum networks, entanglement distribution can be accomplished with the help of intermediary nodes using an operation known as *entanglement swapping* (see Fig. 1c): two nodes that share no physical connection (A and C) first produce an elementary entangled link with the intermediary node (B) via their shared physical medium. The fidelity requirement for such elementary links can be computed from the end-to-end QoS requirements. When both entangled links are ready, a measurement (the swapping operation) can be performed on both qubits at node B, which produces end-to-end entanglement between two unconnected devices A and C. This measurement consumes the original entangled links. Since entanglement generation is a probabilistic process, forming long distance entangled links efficiently asks for B to have a quantum memory in which the qubit of one entangled link (say the link A-B) can be stored until the second link (say B-C) is ready. Entanglement swapping operations can either be probabilistic, or deterministic depending on the underlying physical implementation. While our work can also be extended to systems in which swapping operations are probabilistic, we here focus on the case of deterministic swaps as allowed by processing nodes built from eg. NV centers in diamond, ion traps, or neutral atoms. When producing entanglement between two end nodes that are separated by one (or more) intermediary node(s), failure to achieve QoS requirements in any one entangled link in the chain leads to a failure to achieve overall end-to-end QoS. It is important that the swapping operation is applied to the correct entangled links, requiring the careful allocation of qubits to protocol operations across the network. Otherwise, we may establish a link between incorrect pairs of users (or none at all) resulting in application failure.

To meet fidelity requirements, quantum repeater protocols may also employ entanglement distillation operations [18] which turn multiple low-fidelity links into a fewer number of higher quality links. Entanglement distillation has been demonstrated using NV centers in diamond [35]. This operation is in general probabilistic, but a heralding signal is produced to indicate success or failure [35]. For all such repeater operations, the nodes need to exchange classical information, and we hence take a classical network supporting entanglement generation as a given.

Several technological limitations impose stringent demands on any coordination mechanism used to produce end-to-end entanglement. First, due to limited lifetimes of quantum memories, the fidelity of an entangled link decreases exponentially with the storage time (at most seconds [1]). Repeater nodes that lack the ability to store entanglement, or have very short storage times therefore must establish the needed links close in time. Any schedule that does not ensure that the links are produced close in time (i.e. missing deadlines) for any single hop in the chain connecting the two end nodes will hence lead to a failure in end-to-end entanglement generation with the desired QoS requirements.

Second, NISQ devices can only store a limited amount of quantum information at a time. This limits the number of entangled links that a node can hold simultaneously (demonstrated 2 [25]), posing additional resource allocation challenges. Processing nodes typically have different types of

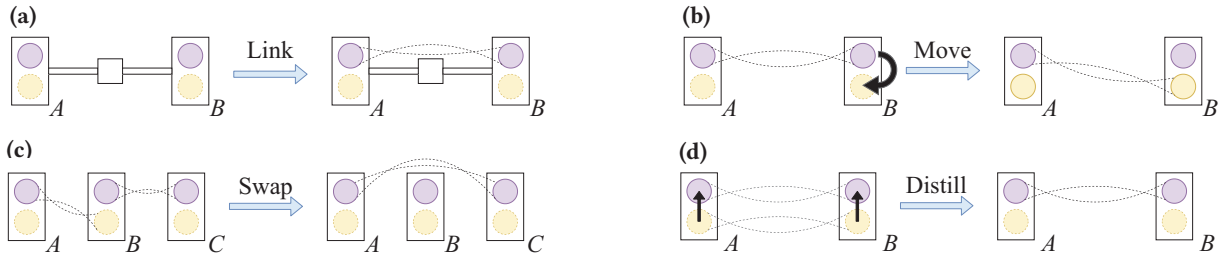
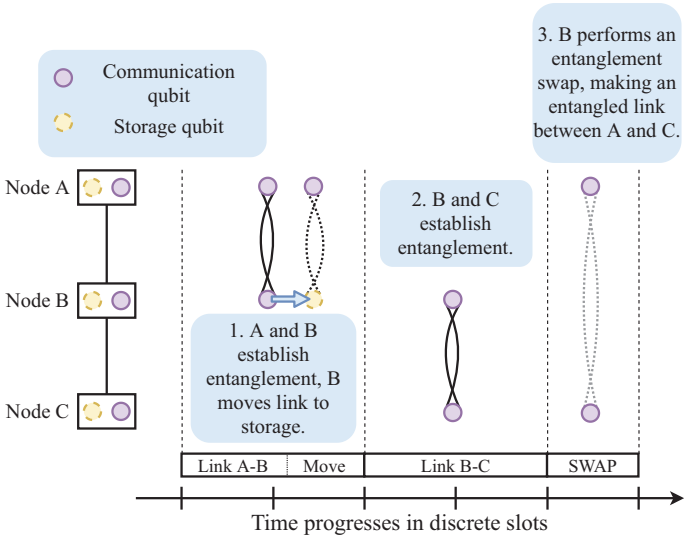


Figure 1: Basic quantum repeater operations to produce long-distance entanglement. a) Generation of an elementary entangled link between two devices connected by a physical medium. b) Memory operation: Certain quantum devices may move the qubit to a different location in memory in order to generate further entangled links. c) Entanglement swapping can be used to produce an entangled link between two unconnected quantum devices. d) Entanglement distillation can be used to produce one (or more) entangled links with a higher fidelity (quality) from two (or more) links of lower quality.

Figure 2: Temporal visualization of a quantum repeater protocol establishing one entangled link between two nodes A and C via one *intermediary node* B (e.g. a quantum repeater). All nodes perform operations according to a schedule that allocates operations to specific time slots. The first operation produces an elementary entangled link between A and B, followed by a move operation to the storage qubit. Operations are assigned into time slots of sufficient length to produce entanglement w.h.p. Due to limited parallelism at device B, the elementary entangled link $B-C$ can only be produced in the subsequent time period. Once the time slots of producing both elementary links w.h.p. and executing the move have elapsed, an entanglement swapping operation is executed, consuming the original entanglement to produce one entangled link $A-C$.



qubits [16]: communication qubits with an optical interface for entanglement generation with connected nodes, as well as storage qubits which can solely be used for storing and manipulating qubits in memory. As such, quantum repeater protocols may necessitate a move operation in memory (Fig 1b).

Third, several network device platforms for processing end nodes (see e.g.[30]) are limited to either processing qubits (e.g. swapping or distilling operations), or establishing entanglement exclusively at any time but not both. This imposes additional timing constraints in any schedule.

3.2 Application Requirements

Much like applications in traditional networks, applications in quantum networks may observe different traffic patterns and quality-of-service (QoS) requirements in order to execute correctly. Applications of the *Measure Directly* (*MD*) use case [16] produce many end-to-end entangled links, but do not require them to be stored nor produced at the same time. This provides flexibility in choosing throughput and jitter requirements. In contrast, applications of the *Create and Keep* (*CK*) use case [16] may require storing multiple entangled links at the same time. Since memory lifetimes are short, applications of *CK* use cases need strict jitter requirements in order to ensure that sufficiently many entangled links can be produced within the same time window. In all use cases, requirements on entanglement fidelity vary depending on the error tolerance of applications, providing flexibility in the choice of quantum repeater protocols for delivering entanglement. Designing quantum networks that meet varying levels of QoS requirements thus increases the number of supported applications and consequently its utility.

4 TDMA ARCHITECTURE DESIGN

4.1 Centralized Control

Due to the design considerations posed by near-term quantum devices, we opt for a centralized architecture in which a central controller is responsible for setting a network-wide

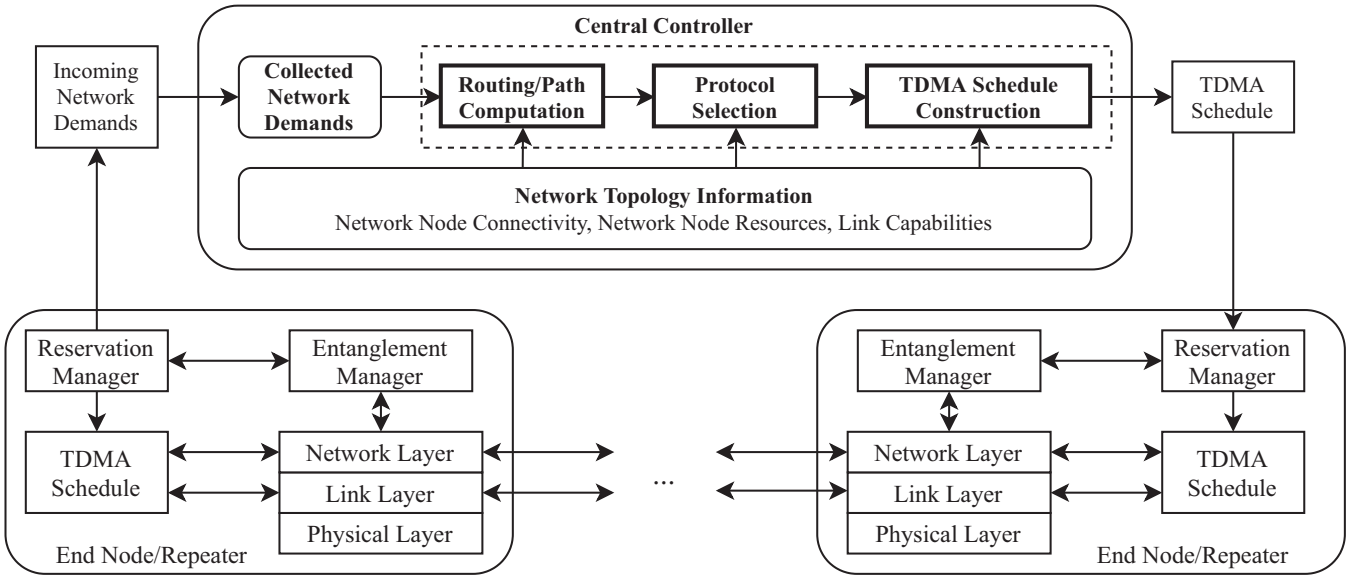


Figure 3: Interaction diagram of software components of end nodes and repeater nodes in the quantum network. Ellipses denote additional network nodes with the same software while arrows denote communication between software components. A reservation manager acts as an interface between applications and the central controller for expressing network demands while the entanglement manager tracks delivered entanglement and provides it to requesting applications. Network demands are forwarded to a central controller where they are subject to admission control and used to produce a new network schedule. The reservation manager installs these schedules for use by the local quantum network stack.

schedule for end-to-end entanglement generation based on demands communicated by the end nodes (see below). This allows us to mitigate limited memory lifetimes imposing strict deadlines on the schedule of repeater protocol operations while maximizing network usage for many users. In addition, a network-wide schedule provides a ready means of tracking which entangled links are used in swapping operations, ensuring entanglement is created between the correct nodes. This removes the need for real-time discussion among network nodes to coordinate quantum operations and reduces the complexity of implementation.

Before any quantum communication takes places, the nodes engage in a discussion with the central controller who acts as a repository of all information required to schedule repeater protocols. The controller holds information such as the network topology and link capabilities, i.e., the available choices of fidelity, throughput and latency at which an elementary link can be produced for each pair of connected nodes [16]. Such topology information may be acquired using link bootstrapping protocols (see e.g. [16, 43]) for characterizing QoS capabilities between connected nodes. What is more, the information includes hardware capabilities of the individual nodes themselves, i.e. their available communication and storage qubits, the quality and speed of their operations affecting end-to-end QoS capabilities, as well as their availability for producing entanglement in given time slots. This information should be updated intermittently to keep the central controller up-to-date as near-term NISQ devices may require intermittent calibration and the capabilities of links may drift with time. The central controller has no quantum capabilities and does not participate in any quantum repeater protocol, though it requires timing synchronization with network nodes for coordinating changes to the network schedule. To prevent disruptions in network service, such a central controller should be realized using a fault-tolerant distributed computing architecture [31].

4.2 Starting Communication

Before entanglement generation for a specific application on end nodes A and B commences, A and B form a classical connection to agree on QoS requirements that entanglement generation should obey. These demands for entanglement are then communicated to the controller along with a maximal time that the end nodes are willing to wait before entanglement production starts. End nodes may also request the central controller to exclude slots so as to allow time for processing entanglement between scheduled operations. The controller then produces a schedule that captures QoS requirements. If demands exceed network capabilities or cannot be fulfilled within the desired time, the controller rejects the new demands. Once entanglement generation starts, the central controller will schedule the requested demand

continuously until the end nodes ask to stop entanglement production.

Integration into the quantum network stack of [16] can be achieved as in Fig. 3: User applications at the end nodes A and B communicate their requirements using a *reservation manager* that acts as an interface between the end node and the central controller (Fig. 3). This reservation manager provides a service interface akin to the link layer interface in [16], allowing applications to specify requirements such as fidelity F, the desired number of entangled pairs N (or throughput R), and constraints on jitter J. If needed, the reservation manager translates N into a rate R. This manager is responsible for submitting all application specific network demands (A, B, F, R, J) to the central controller, and for installing schedules for local use. The reservation manager may additionally supplement the demands with a specification of time needed to process entanglement, allowing slots to be excluded from the schedule so that end nodes can process entanglement before subsequent repeater protocol operations. If network demands are accepted by the central controller, an *entanglement manager* serves the created entangled links to the applications in accordance with their requests, whenever entanglement according to the central controller's schedule becomes ready. The network layer [40] is responsible for executing both swapping and distillation operations and must communicate the associated control information, including e.g. measurement outcomes (see Section 3.1), from these operations to other network nodes. The link layer [16] is used to produce elementary entangled links with connected nodes, where the distributed queue of [16] is replaced with the central network schedule. When the desired number of entangled links has been produced (or an applications no longer desire entanglement), end nodes immediately cease to participate in the operations scheduled to serve the specific application, and the reservation manager updates the central controller.

4.3 Schedule Overview

To construct a schedule meeting QoS requirements, several actions are needed: First, the controller determines one (or several) options on how end-to-end entanglement generation can at all be realized such that the network demand submitted by two end nodes is satisfied. This includes a path selection (see Routing) and a selection of repeater protocols for each path (see Protocol Selection). The choice of protocol determines which operations (see Fig. 1) need to be executed at which nodes along the path, as well as the dependencies of said operations and timing constraints.

Second, given the selection of path and protocols, the central controller maps the protocol operations (and associated information) of all network demands into a joint network-wide schedule composed of fixed-duration time slots. Operations can span a single, or multiple consecutive slots, allowing additional time to be allocated to operations that require more time than others. Since operations occupy an integer number of slots, the size of slots should be chosen so that the excess amount of time allocated to operations is limited so as to limit reduction of fidelity from storing entangled links between operations. Our scheduling structure is beneficial for several reasons. Flexibility in time allocation to operations allows us to spend appropriate amounts of time creating each elementary entangled link, which depends on the capabilities of the individual connected nodes as well as their distance in fiber (or free-space link). It also allows us to account for the highly varying capabilities in the different network nodes, where different platforms have different timing requirements and quality for the operations (see e.g. [30, 44]. These may even differ between two different nodes with the same underlying physical system due to the nature of early quantum hardware. Scheduling operations also ensures that qubits are exclusively allocated to each operation, granting contention-free usage of network devices to support network operation. To guarantee consistent network operation, all quantum network nodes should be time-synchronized to boundaries of slots in the schedule. Such schedule synchronization can be achieved by building upon the existing synchronization mechanisms used by network nodes for heralded entanglement generation [16].

Finally, the controller periodically installs the new schedule at all network nodes. New schedules can be installed in a synchronized fashion using a periodic reconfiguration period and having the controller instruct network nodes to switch to new schedules at pre-specified times. The period at which new schedules are installed is a design choice that depends on the desired responsiveness of the network and the minimum amount of time required to communicate the schedules to the network nodes. As time progresses, the schedule then directs network node behavior: it dictates when network nodes execute the operations in a repeater protocol.

4.4 Constructing Schedules

We now detail the demand processing pipeline of our central controller as shown in Fig. 3. We remark that in order to produce a schedule quickly and permit a modular design, we here separate the pipeline into distinct steps. Of course, one could envision that a global optimization could yield slightly better overall network performance, at the expense of a significantly higher computational effort increasing the latency.

Routing. The demand processing pipeline begins by first collecting demands submitted by network nodes. Depending

on the desired responsiveness of the network, the central controller may be designed to process new network demands periodically or once a sufficient number of them have been aggregated. Collected demands are fed into a routing stage that determines paths of quantum network nodes to use for satisfying each network demand. At this stage, the central controller can assign paths to each network demand based on several different strategies. For example, paths may be assigned based on estimates of the achievable fidelity and throughput or in order to balance load across network resources. We remark that the problem of routing has been previously studied in several works [12, 13, 47, 55] and that the choice of algorithm lies out of the scope of this paper.

Protocol Selection. After routing, quantum repeater protocols consisting of a combination of operations (see Fig. 1) are chosen for each demand and associated path. This must take into account the capabilities of all the nodes along the paths (available qubits, quality of their memories and operations, time needed to execute operations), as well as the capabilities (fidelity, throughput, latency) for producing elementary links between connected nodes along the paths. The selection includes which type of qubits (communication or storage qubit) to use at which node. To provide an accurate estimate of the end-to-end fidelity, operations must be mapped onto qubits to determine 1) whether a sufficient number of qubits exist to execute the protocol, 2) determine any reduction in fidelity due to storage of entangled links between operations, and 3) the quality of the operations themselves. We define the latency of a quantum repeater protocol to be the amount of time that elapses between the start of the first operation and the completion of the final operation in the protocol, and we require that the latency of repeater protocols is small enough to meet throughput requirements. Mapping operations to qubits allows the controller to determine these quantities and find appropriate repeater protocols. An example of such a mapping for the protocol in Fig. 2 can be seen in Fig. 4. Similarly to routing, protocol selection has been studied previously ([5, 24, 34], Appendix A.1.1).

The specification of an operation in the schedule includes resource requirements (communication and storage qubits to be used), as well as QoS requirements (fidelity, throughput) of generating elementary links between connected nodes using a link layer protocol [16], allowing the nodes to correctly execute the operations. Timing constraints take into account the memory quality of the network nodes involved, to ensure that entangled links are produced sufficiently close in time to allow for end-to-end entanglement generation fulfilling network demands within the limited memory lifetimes.

Whenever an operation is probabilistic, sufficient time slots are allocated such that the operation(s) succeeds with high probability. Entanglement generation of an elementary link is achieved robustly using the link layer protocol in [16], where the distributed queue of [16] is replaced by the schedule set by the controller. Here, a sufficiently large time period is given to the link layer protocol, in order to succeed almost surely in producing an elementary link within that time frame. For larger end-to-end quantum repeater protocols, the probability that entanglement is successfully delivered depends on the creation of all elementary entangled links close in time. Here, one can use standard concentration bounds [27] to determine the amount of time allocated to each elementary entangled link such that the overall protocol succeeds with high probability. Meeting throughput and jitter requirements becomes increasingly difficult if repeater protocols fail with non-negligible probability.

Despite scheduling probabilistic operations for sufficient time to succeed with high probability, entanglement generation may still fail. Such failures are communicated by the link and network layer to the end nodes, who then act in accordance with the application protocol to wait for the next link to succeed (*MD* use cases, some *CK* use cases, see Section 3.2), or restart (most *CK* use cases). No other action is taken and the nodes continue to follow the schedule.

Schedule Construction. The final stage of the central controller is to combine protocols for each pair of end nodes in order to construct the joint network schedule. Here, the chosen repeater protocols are scheduled so that the delivery of entanglement respects throughput and jitter requirements, where the controller may make a choice of different protocols identified to meet end node demands. This stage no longer considers the end-to-end fidelity requirements as the protocols have been chosen to satisfy minimum end-to-end fidelity. The schedule is constructed to be of a finite-length and is executed cyclically, meaning that the schedule repeats from the beginning once the end has been reached. By using a cyclic schedule, the central controller need only broadcast a new schedule to the network when demands are changed, thereby reducing the amount of communication to keep the network running. The length of the schedule is chosen by the controller based on the network demands. Once the schedule has been produced, the central controller distributes it to the network nodes and each node's reservation manager installs the schedule for local use. In Section 5, we will detail our approach to constructing these schedules.

When end nodes no longer require entanglement they contact the central controller to remove their network demands. If applications fail to remove their demands, the schedule will still retain the corresponding repeater protocol, potentially starving new applications. To avoid this, the central controller may employ a heartbeat mechanism where end nodes must regularly inform the central controller to keep

their demand, allowing the controller to remove demands that it deems as inactive.

5 SCHEDULING REPEATER PROTOCOLS

In this section, we present our approach to the schedule construction step of the central controller. We describe our model for the repeater protocols selected in the protocol selection stage and detail two methods for constructing the network schedule. A more in-depth overview of our methodology can be found in Appendix section A.1.

5.1 Repeater Protocol Model

We assume that repeater protocols chosen by the protocol selection stage are provided to the schedule construction stage along with their timing and qubit requirements. This includes specifying the relative timing between each operation as well as the qubits that are used in each operation. We remark that this assumption is reasonable as estimating the end-to-end fidelity and success probability of a repeater protocol requires knowledge of timing between protocol operations. We refer to a quantum repeater protocol that provides these details as *concrete* and represent them using a directed acyclic graph (DAG) $P(A, I, M, Q)$. The set of vertices A represents protocol operations, the set of edges I shows dependency relationships between operations, the map M specifies the start and end times of operations and the map Q specifies the qubits for each operation. An example may be visualized in Fig. 4.

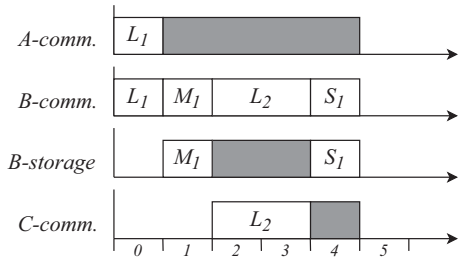


Figure 4: Schedule for the quantum repeater protocol in Fig. 2 with operations L (link generation), M (move), and S (entanglement swapping). Communication and storage qubits for each node are labeled on the left while time proceeds to the right. Shaded regions indicate where qubits are holding entangled links.

We remark that an alternative representation may specify the start times of only elementary link generation operations while move, entanglement swap, and entanglement distillation are performed as-soon-as-possible internally.

5.2 Scheduling Methods

The problem of constructing TDMA schedules of quantum repeater protocols can be stated as follows. Given a set of network demands $\mathcal{D}$ containing $(src, dst, F_{min}, R_{min}, J_{max})$ tuples detailing the source, destination, minimum fidelity, minimum throughput, and maximum jitter requirements along with their corresponding set of concrete repeater protocols $\mathcal{P}$ containing (A, I, M, Q) tuples for each demand, produce a schedule $\mathcal{S}$ with a time duration of L in seconds that maps each (demand, protocol) pair (D_i, P_i) to a set of start times $\mathcal{S}(D_i, P_i)$ such that

(1) no two operations $a_1, a_2 \in \cup_{P_i \in \mathcal{P}} A_i$ use the same qubit at the same time,
(2) operations in A_i are scheduled to respect the relative offset mapping M_i of concrete repeater protocol $P_i(A_i, I_i, M_i, Q_i)$. That is, for any operations a, b in A_i, the jth start times $s_{a,j}, s_{b,j}$ satisfy $s_{a,j} - s_{b,j} = M_i(a) - M_i(b)$ for all i and j,
(3) no distinct operations $a, b, c \in \cup_{P_i \in \mathcal{P}} A_i$ with $b \rightarrow c$ are assigned start times $s_{a,j}, s_{b,k}, s_{c,k}$ such that $s_{b,k} \leq s_{a,j} \leq s_{c,k}$ and $Q(a) \cap Q(b) \cap Q(c) \neq \emptyset$ for any j, k,
(4) the entanglement delivered by each protocol P_i is at least F^i_{min},
(5) the average rate of entanglement delivery for demand D_i satisfies minimum rate requirements, ie. $\frac{|\mathcal{S}(D_i,P_i)|}{L} \geq R^i_{min}$, and
(6) the variance in inter-delivery times for a demand D_i is below J^i_{max}, ie. $\frac{\sum_{j=1}^{|\mathcal{S}(D_i,P_i)|}(x_{i,j}-\bar{x}_i)^2}{|\mathcal{S}(D_i,P_i)|} \leq J^i_{max}$,

where the first condition ensures that protocol operations are scheduled in accordance with their concrete repeater protocols so that the protocol meets minimum fidelity requirements, the next two conditions ensure contention-free usage of qubits, and the final three conditions ensure that the frequency of repeater protocol execution meets QoS requirements. Here, $|\mathcal{S}(D_i, P_i)|$ indicates the number of times P_i has been scheduled while $x_{i,j}$ indicates the inter-delivery time between the jth and $(j+1)$th execution of P_i and $\bar{x}_i$ is the average inter-delivery time. Note that our definition of QoS will be satisfied with high probability. The case where entanglement delivery always succeeds may be recovered by delivering a classically-correlated quantum state to applications such that the average entanglement fidelity satisfies fidelity requirements [30].

While our architecture is suitable for handling the timing-related QoS requirements, we limit the scope of our scheduling methods to handle throughput requirements while fulfilling jitter requirements on a best-effort basis. The scheduling methods we present here are motivated by the real-time constraints imposed by near-term quantum hardware and we propose the use of a novel heuristic that combines the two scheduling methods presented below. We benchmark our heuristic against other known heuristics from these scheduling methods 6.2 and expand upon these methods further in Section A.1.

5.2.1 Periodic Task Scheduling. Given the problem definition above, the first method we consider for constructing the schedule is through non-preemptive periodic task scheduling [15, 42]. In this method, each concrete repeater protocol is transformed into a periodic task with execution requirements reflecting the protocol's latency and the QoS requirements of the corresponding demand. We choose to use non-preemptive scheduling due to two reasons: 1) permitting preemption of a protocol mid-execution introduces delays between operations that increase the latency of the quantum repeater protocol and also reduce the end-to-end fidelity, preventing it from meeting QoS requirements and 2) qubits may hold entangled links at the time of preemption, meaning they are either unavailable for use by the preempting protocol or the entangled links must be discarded, resulting in failure of the previously executing protocol. We remark that the use of preemptive strategies can offer higher scheduling flexibility and permit additional demands to be satisfied, though this comes at the cost of ensuring that the entanglement created before the period of preemption is still present in order to resume the protocol afterwards. Furthermore, preemption of repeater protocols introduces delays between operations which may not be permitted due to memory lifetimes. By determining a schedule for the set of periodic tasks, a corresponding network schedule can be extracted that specifies when each concrete repeater protocol $P_i(A_i, I_i, M_i, Q_i)$ starts. The start and end times for each repeater protocol's operations are then obtained by using the associated map M_i. We provide a more in-depth description of this method as well as our explored heuristics in Section A.1.2 and our data repository [22].

In general, determining a valid schedule for a set of non-preemptable periodic tasks is NP-hard [32], thus the choice of algorithm impacts the overhead in producing a new schedule and the network's responsiveness to changes in network demands. To achieve lower overhead, scheduling heuristics can be used to construct schedules more quickly than an exhaustive search at the cost of not scheduling some of the tasks. With respect to the class of work-conserving scheduling techniques, non-preemptive earliest-deadline first (NP-EDF) [33] is known to be optimal, meaning that NP-EDF can schedule a set of tasks if there exists a non-preemptive work-conserving heuristic capable of scheduling the same set of tasks. Our implementation of NP-EDF has a runtime complexity of $O(N \log N)$ where N is the number of scheduling decisions made.

While the periodic task scheduling method is simple, it comes at the cost of lower network throughput. By hiding fine-grained scheduling decisions on individual qubits, a lower runtime complexity can be achieved for producing a schedule. However, under high network load this results in treating the network as a single resource which prevents any concurrent execution of quantum repeater protocols that operate on disjoint paths in the network. We next describe a second method that can be used to achieve higher network performance at the cost of additional complexity in schedule construction.

5.2.2 Resource-Constrained Project Scheduling. Creating the schedule can be achieved using a second method based on the non-preemptive resource-constrained project scheduling problem (RCPSP) [11]. The goal of RCPSP is to schedule activities of a project under scarce resource constraints and precedence relations. In this method, quantum repeater protocols are encoded into an activity-on-node network representing a project for RCPSP [11]. Scheduling the activity-on-node network provides a scheduling of all quantum repeater protocol operations for the network schedule.

Similarly to periodic task scheduling, constructing a schedule for the RCPSP problem is NP-hard [6] and heuristic methods can be used to find schedules more quickly [46]. Due to the added complexity from scheduling individual resources, RCPSP heuristic solvers observe a higher complexity than periodic task scheduling heuristic algorithms. The trade-off with this higher complexity is that using RCPSP to represent the scheduling problem allows finer-grained scheduling decisions at the resource level, permitting higher levels of parallelism between repeater protocol operations.

We evaluate two heuristics for the RCPSP scheduling method. The first heuristic is based on the EDF heuristics from Section 5.2.1 while the second heuristic combines the periodic task scheduling and RCPSP methods together into a heuristic we call full-protocol reservation (FPR). This novel heuristic approximates the activity-on-node network to a fixed-size that is independent of the number of operations in the protocol. Doing so provides significant reductions in runtime complexity compared to the EDF RCPSP heuristic while still achieving higher network performance than periodic task scheduling. Our RCPSP-NP-EDF and RCPSP-NP-FPR heuristics have runtime complexities $O(N^2S^2|K|\log(NS))$ and $O(N^2|K|\log(N)$ respectively where N is the same as in the periodic task scheduling method, $|K|$ is the total number of qubits, and S is the maximum number of operations in any concrete repeater protocol to schedule. A more in-depth description of our RCPSP method as well as our explored scheduling heuristics is provided in Section A.1.3.

6 EVALUATION

At present, the largest network of processing nodes consists of three nodes [25], prohibiting an interesting hardware validation. We instead focus on an evaluation in simulation using some hardware validated parameters. The objective of our architecture is to enable such early stage networks to scale, which is why we focus on 4 different few node networks

to evaluate the performance of our methodology. We additionally study a futuristic scenario using a real-world fiber topology based on the core network of SURFnet, a network provider for Dutch educational and research institutions. We implement the demand processing pipeline of our TDMA architecture using Python to evaluate the performance of several heuristics at the schedule construction step. As the performance of routing and protocol selection algorithms lie outside of the scope of this paper, we implement these stages using a shortest-path routing algorithm (edge weights corresponding to link length) and our own extension to a previously studied protocol selection algorithm known as entanglement swapping search scheme (ESSS, Appendix A.1) [5]. We do not expect that the choice of these algorithms significantly affects our evaluation, though we remark that more refined methods for choosing a repeater protocol could be used (see e.g. [24, 34]).

6.1 Hardware and Protocol Selection

Our small network simulations use hardware-validated parameters of processing nodes based on NV centers in diamond [16, 26], since this platform is presently the only one in which at least three nodes are connected [25]. In our simulations, each quantum network node has a single communication qubit, and three storage qubits for storing entangled links. Link capabilities between network nodes are found by simulating entanglement establishment using software provided by the authors of [16] in Table 1. For our SURFnet simulation, we assume entanglement may be generated with fidelity $F = 0.999$ at a rate of 1.4 kHz between directly connected nodes. Timing of the operations in our simulations are: swapping operation 1 ms, distillation 526 μs, and move 961 μs [25, 35]. For simplicity in protocol selection, we here assume operations and memory are perfect (i.e. do not further reduce the quality of the entanglement generated), and that the entangled links correspond to a worst-case quantum state (see Appendix A.1.1). Since this does not impact the evaluation of the scheduling algorithms: protocol selection makes sure that end-to-end fidelity requirements are met. For lower fidelity, repeater protocols can meet QoS requirements using elementary link generation and entanglement swapping whereas higher fidelity requirements will include distillation.

Link Length	5 km
Link Capabilities[16] (Fidelity, Rate in Hz)	(0.88, 14.16), (0.83, 20.84), (0.79, 27.83), (0.75, 33.98), (0.7, 39.18), (0.66, 45.6), (0.62, 51.26), (0.57, 57.73)
# Comm. Qubits	1
# Storage Qubits	3

Table 1: Physical device parameters for small network simulations.

We also make the simplifying assumption that the nodes may perform any operation on different qubits in parallel. This does not hold for the NV platform, but again simplifies the protocol selection phase without impacting the study of scheduling procedures. Finally,we let distillation succeed with unit probability: this merely impacts the number of repetitions until end-to-end entanglement is achieved and while affecting the absolute achievable throughput, it does not impact the comparison of the throughput achieved by different scheduling methods as we aim to study here.

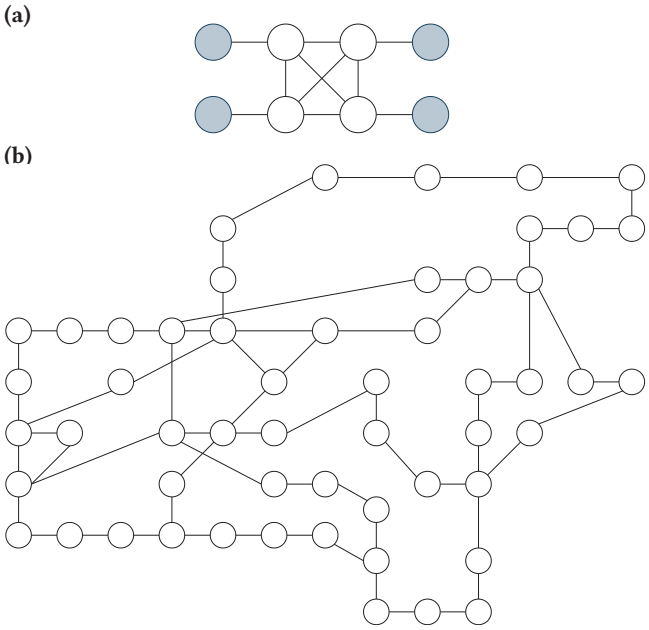


Figure 5: a) Network topology in scheduler evaluation. As reference, a single repeater protocol delivering entanglement of $F = 0.55$ ($F = 0,85$) to two end nodes (grey) via the repeaters (white) operates with latency of 170 (292) ms and throughput of 5.88 (0.342) $\frac{ebit}{s}$. b) SURFnet repeater topology in scheduler evaluation. Each repeater pictured (circle) is connected to an end-node that lies 5 km away (end node not pictured).

6.2 Scheduler Evaluation

We numerically evaluate the scaling of achieved network throughput with network load, as well as the trade-off between throughput and jitter of our implemented scheduling heuristics. Prior studies have shown that throughput decreases as the desired fidelity of an entangled link between connected devices increases [16]. Here we are interested in the relationship between the fidelity requirements of network demands and the achievable network throughput as well as the observed jitter in entanglement delivery. To gain additional insight on the effects of network structure on

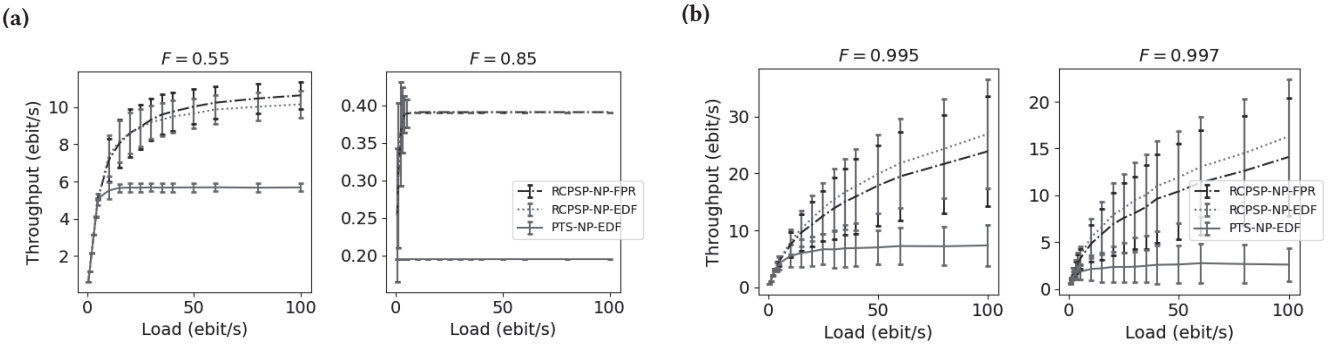


Figure 6: Average network throughput vs. network load for several fidelity requirements F on a) the small network in Fig. 5a and b) the SURFnet topology in figure 5b. Data points averaged over 1000 simulations, error bars one standard deviation. RCPSP heuristics beat the throughput of the trivial schedule obtained by scheduling one single repeater protocol after the other.

scheduler performance, we simulate the schedule construction on three additional small networks (additional data analysis may be found online at [omitted for blinding] [22]).

Time Slot Size	10 ms
Fidelity (small nets.)	0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9
Fidelity (SURFnet)	0.98, 0.985, 0.99, 0.995, 0.997
Throughput ($\frac{ebit}{s}$)	12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.1953125

Table 2: Slot duration and permitted QoS levels of network demands for scheduling simulations.

6.2.1 Simulation Parameters. Table 2 shows the set of QoS parameters used for our simulations while Fig. 5a shows one of the networks used for our simulations. We choose this topology due to its symmetric structure: for each path between two end nodes there exists a disjoint path and subsequently a disjoint repeater protocol connecting the remaining nodes.

We generate a batch of network demands by randomly sampling pairs of end nodes to be source and destination. Each batch of network demands has a fixed fidelity and is randomly assigned a rate from those in Table 2. We restrict rates to a fixed set to reduce the length of the schedule. We use a slot size of 10 ms, for which timing synchronization to slot boundaries across network nodes may be reasonably achieved, resulting in a schedule with a maximum length of 512 slots (equivalent to 5.12 s). After generating a batch of demands, we compute the path for each demand and generate concrete repeater protocols using our modified ESSS algorithm. The set of demands and their concrete repeater protocols are then fed to our scheduling heuristics.

6.2.2 Scaling with Network Load. We evaluate how the network throughput of the scheduling heuristics scales with the cumulative throughput requirements of all network demands (henceforth referred to as *network load*). Evaluating schedulers on this basis is important as achieving higher network throughput shows how well each scheduler extracts parallelism from the network. In these simulations, we generate batches of demands for several network loads and apply our scheduling heuristics. Increasing the number of demands in the system provides schedulers an opportunity to squeeze additional throughput from the network if some portions of the network are not fully utilized.

Fig. 6 shows the average network throughput achieved by each scheduler for several choices of end-to-end fidelity. Both RCPSP scheduling heuristics achieve higher network throughput than our periodic task scheduling heuristic under high load and show comparable performance for low

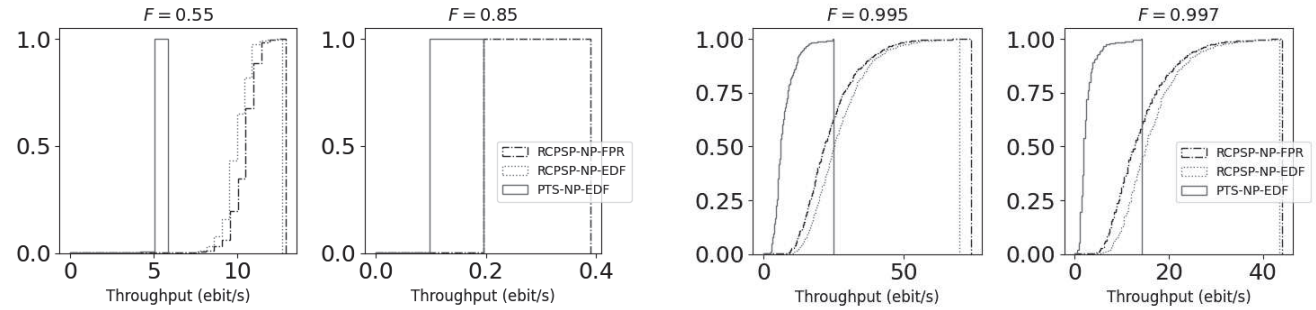


Figure 7: CDF of network throughput for a load of $100\frac{ebit}{s}$ on a) the small network of Fig. 5a and b) the SURFnet topology.

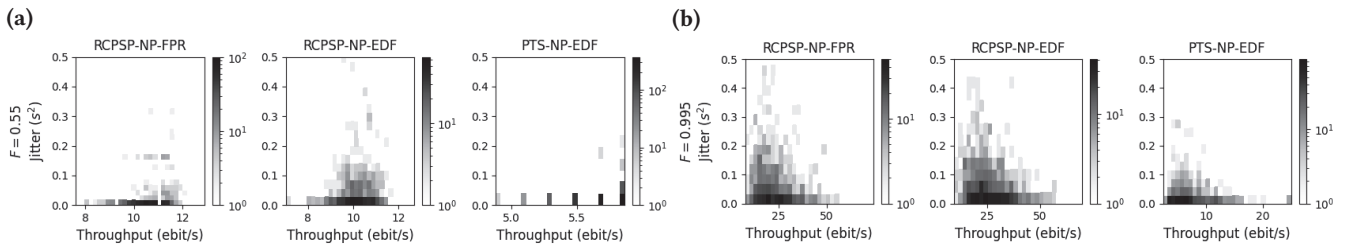


Figure 8: Histograms of throughput and jitter for a) the small network with fidelity requirement $F = 0.55$ and b) SURFnet with a fidelity requirement of $F = 0.995$ under a network load of $100\frac{ebit}{s}$.

network loads. This confirms our expectations of achieving higher network performance when using the RCPSP scheduling method. We observe that the achievable network throughput decreases as fidelity requirements are increased which agrees with the results in [16]. These results additionally suggest that the central controller may choose to use RCPSP heuristics for higher network load while using periodic task scheduling heuristics for low network load in order to reduce scheduling overhead while maintaining network performance. The large variance observed in the average throughput in our SURFnet simulations is the result of network demands that highly congest a common repeater, resulting in lower network throughput.

From our simulations we also learn how well our novel RCPSP-NP-FPR heuristic compares to the PTS-NP-EDF and RCPSP-NP-EDF heuristics. Fig. 7 shows the CDF of achieved network throughput when the network load is fixed to be $100\frac{ebit}{s}$. Here, we see that the throughput distribution of RCPSP-NP-FPR is higher than that for PTS-NP-EDF and comparable to that of the RCPSP-NP-EDF heuristic. This shows that our heuristic can maintain high performance while reducing computational complexity as compared to RCPSP-NP-EDF. These results are further supported by our simulations on other small networks [22].

6.2.3 Throughput/Jitter Trade-Off. In addition to network throughput, we extract the jitter in entanglement delivery. High amounts of jitter result in irregular entanglement delivery and may affect applications that fall into the *Create and Keep* use case [16] negatively. While the evaluated scheduling heuristics are not targeted at minimizing jitter, it is instructive to characterize the trade-off to the achieved throughput. We remark that any demand $D_i = (A, B, F_{min}, R_{min}, J_{max})$ with $J_{max} \leq \frac{1}{(R_{min})^2}$ is guaranteed to be satisfied using either the periodic task scheduling or RCPSP methods we propose.

Fig. 8 shows the distribution of throughput/jitter pairs for our scheduling heuristics when network load is $100\frac{ebit}{s}$. From our results we learn that our periodic task scheduling heuristic observe smaller amounts of jitter compared to the RCPSP heuristics in our small network simulations, suggesting that achieving high network throughput comes at a cost of increased variance in inter-delivery times of entanglement. However, we see that for the SURFnet topology, where the path length between pairs of end-nodes varies, that the periodic task scheduling and RCPSP methods show comparable levels of jitter. This is further supported by our additional simulations of small networks.

6.3 Summary

Our simulation results show that our architecture can be used to accommodate different levels of network performance by controlling the choice of scheduling strategy. We show that flexibility in the choice of scheduling heuristic allows network operators to control trade-offs between scheduling overhead, achievable network throughput, and observed jitter in entanglement delivery. When end-to-end fidelity requirements can be met with low-latency repeater protocols, RCPSP-based heuristics are more suitable for achieving higher network throughput whereas periodic task-scheduling heuristics are more suitable for achieving lower jitter. In this realm, we also show that our novel RCPSP-FPR heuristic can be used to reach comparable levels of performance to RCPSP-NP-EDF while reducing scheduling complexity. However, when repeater protocols have high latency, due to spending more time producing entangled links, the distinction between these scheduling methods with respect to network throughput and jitter becomes less pronounced.

Our results on other network topologies [22] further support these observations and additionally provide insight into the effects of network topology on network throughput. We find that network structures with non-uniform path lengths between end nodes result in larger amounts of variance in the achievable network throughput and that periodic task scheduling heuristics achieve network throughput comparable to RCPSP heuristics on a star topology.

In this work we have presented the first end-to-end design of a centralized quantum network architecture that supports varying levels of QoS requirements among multiple pairs of users. Our architecture may be used to scale near-term networks and may additionally be used to manage smaller networks to form a larger, cluster-based network of quantum devices. Our results may additionally be used as a baseline to evaluate the performance of alternative architecture designs that may target a more distributed approach to network coordination.

This work does not raise any ethical issues.

REFERENCES

[1] Mohamed H Abobeih, Julia Cramer, Michiel A Bakker, Norbert Kalb, Matthew Markham, Daniel J Twitchen, and Tim H Taminiau. 2018. One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment. *Nature communications* 9, 1 (2018), 1–8.

[2] Ishfaq Ahmad, Yu-Kwong Kwok, and Min-You Wu. 1996. Analysis, evaluation, and comparison of algorithms for scheduling task graphs on parallel processors. In *Proceedings Second International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'96)*. IEEE, 207–213.

[3] Luciano Aparicio and Rodney Van Meter. 2011. Multiplexing schemes for quantum repeater networks. In *Quantum Communications and Quantum Imaging IX*, Vol. 8163. International Society for Optics and Photonics, 816308.

[4] Luciano Aparicio, Rodney Van Meter, and Hiroshi Esaki. 2011. Protocol design for quantum repeater networks. In *Proceedings of the 7th Asian Internet Engineering Conference*. 73–80.

[5] Tan Bacinoglu, Burhan Gulbahar, and Ozgur B Akan. 2010. Constant fidelity entanglement flow in quantum communication networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, 1–5.

[6] Martin Bartusch, Rolf H Möhring, and Franz J Radermacher. 1988. Scheduling project networks with resource constraints and time windows. *Annals of operations Research* 16, 1 (1988), 199–240.

[7] Charles H Bennett and Gilles Brassard. 1984. Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing.

[8] Alan A Bertossi, Giancarlo Bongiovanni, and Maurizio Bonuccelli. 1987. Time slot assignment in SS/TDMA systems with intersatellite links. *IEEE Transactions on Communications* 35, 6 (1987), 602–608.

[9] Conor Bradley, Joe Randall, Mohamed Abobeih, Remon Berrevoets, Maarten Degen, Michiel Bakker, Matthew Markham, Daniel Twitchen, and Tim Taminiau. 2019. A ten-qubit solid-state spin register with quantum memory up to one minute. *Physical Review X* 9, 3 (2019), 031045.

[10] Hans J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. 1998. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters* 81, 26 (1998), 5932.

[11] Peter Brucker, Andreas Drexl, Rolf Möhring, Klaus Neumann, and Erwin Pesch. 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. *European journal of operational research* 112, 1 (1999), 3–41.

[12] Kaushik Chakraborty, David Elkouss, Bruno Rijsman, and Stephanie Wehner. 2020. Entanglement Distribution in a Quantum Network, a Multi-Commodity Flow-Based Approach. *arXiv preprint arXiv:2005.14304* (2020).

[13] Kaushik Chakraborty, Filip Rozpedek, Axel Dahlberg, and Stephanie Wehner. 2019. Distributed Routing in a Quantum internet. *arXiv preprint arXiv:1907.11630* (2019).

[14] Chin-Wen Chou, Hugues de Riedmatten, Daniel Felinto, Sergey V Polyakov, Steven J Van Enk, and H Jeff Kimble. 2005. Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature* 438, 7069 (2005), 828–832.

[15] Edgar Frank Codd. 1960. Multiprogram scheduling: parts 1 and 2. introduction and theory. *Commun. ACM* 3, 6 (1960), 347–350.

[16] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. 2019. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication*. 159–173.

[17] Siddhartha Das, Sumeet Khatri, and Jonathan P Dowling. 2018. Robust quantum network architectures and topologies for entanglement distribution. *Physical Review A* 97, 1 (2018), 012335.

[18] Wolfgang Dür and Hans Briegel. 2007. Entanglement purification and quantum error correction. *Rep. Prog. Phys.* 70 (2007), 1381.

[19] Wolfgang Dür and Hans J Briegel. 2003. Entanglement purification for quantum computation. *Physical review letters* 90, 6 (2003), 067901.

[20] Wolfgang Dür, Hans J Briegel, Juan Ignacio Cirac, and Peter Zoller. 1999. Quantum repeaters based on entanglement purification. *Physical Review A* 59, 1 (1999), 169.

[21] Artur K Ekert. 1991. Quantum cryptography based on Bell's theorem. *Physical review letters* 67, 6 (1991), 661.

[22] Omitted for blinding. 2021. An Architecture for Meeting Quality-of-Service Requirements in Multi-User Quantum Networks. *DataverseNL, V1* (2021). https://doi.org/10.34894/5EOGAF

[23] Jean-François Frigon, Victor CM Leung, and Henry Chan Bun Chan. 2001. Dynamic reservation TDMA protocol for wireless ATM networks. *IEEE Journal on Selected Areas in Communications* 19, 2 (2001), 370–383.

[24] Kenneth Goodenough, David Elkouss, and Stephanie Wehner. 2020. Optimising repeater schemes for the quantum internet. (2020). arXiv:2006.12221 [quant-ph]

[25] R. Hanson. 2021. Personal communication. (2021).

[26] Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenberg, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, et al. 2015. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* 526, 7575 (2015), 682–686.

[27] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer, 409–426.

[28] Julian Hofmann, Michael Krug, Norbert Ortegel, Lea Gérard, Markus Weber, Wenjamin Rosenfeld, and Harald Weinfurter. 2012. Heralded entanglement between widely separated atoms. *Science* 337, 6090 (2012), 72–75.

[29] Ekram Hossain and Vijay K Bhargava. 2001. A centralized TDMA-based scheme for fair bandwidth allocation in wireless IP networks. *IEEE Journal on Selected Areas in Communications* 19, 11 (2001), 2201–2214.

[30] Peter C Humphreys, Norbert Kalb, Jaco PJ Morits, Raymond N Schouten, Raymond FL Vermeulen, Daniel J Twitchen, Matthew Markham, and Ronald Hanson. 2018. Deterministic delivery of remote entanglement on a quantum network. *Nature* 558, 7709 (2018), 268–273.

[31] Pankaj Jalote. 1994. *Fault tolerance in distributed systems*. PTR Prentice Hall Englewood Cliffs, New Jersey.

[32] Kevin Jeffay, Richard Anderson, and Charles Martel. 1988. *On optimal, non-preemptive scheduling of periodic and sporadic tasks*. Department of Computer Science, University of Washington.

[33] Kevin Jeffay, Donald F Stanat, and Charles U Martel. 1991. On non-preemptive scheduling of periodic and sporadic tasks. In *IEEE real-time systems symposium*. US: IEEE, 129–139.

[34] Liang Jiang, Jacob M Tayolr, Navin Khaneja, and Mikhail D. Lukin. 2007. Optimal approach to quantum communication using dynamic programming. *Proceedings of the National Academy of Sciences* 104, 44 (2007), 17291–17296.

[35] Norbert Kalb, Andreas A Reiserer, Peter C Humphreys, Jacob JW Bakermans, Sten J Kamerling, Naomi H Nickerson, Simon C Benjamin, Daniel J Twitchen, Matthew Markham, and Ronald Hanson. 2017. Entanglement distillation between solid-state quantum network nodes. *Science* 356, 6341 (2017), 928–932.

[36] SM Kamruzzaman and Mohammad Shamsul Alam. 2010. Dynamic TDMA slot reservation protocol for cognitive radio ad hoc networks. In *2010 13th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 142–147.

[37] Hui Kang, Ya-nan Zhao, and Fang Mei. 2013. A graph coloring based tdma scheduling algorithm for wireless sensor networks. *Wireless personal communications* 72, 2 (2013), 1005–1022.

[38] Xin Kang, Chin Keong Ho, and Sumei Sun. 2014. Optimal time allocation for dynamic-TDMA-based wireless powered communication networks. In *2014 IEEE Global Communications Conference*. IEEE, 3157–3161.

[39] Jeff H Kimble. 2008. The quantum internet. *Nature* 453, 7198 (2008), 1023–1030.

[40] Wojciech Kozlowski, Axel Dahlberg, and Stephanie Wehner. 2020. Designing a quantum network protocol. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*. 1–16.

[41] V Krutyanskiy, M Meraner, J Schupp, V Krcmarsky, H Hainzer, and Ben P Lanyon. 2019. Light-matter entanglement over 50 km of optical fibre. *npj Quantum Information* 5, 1 (2019), 1–5.

[42] Chung Laung Liu and James W Layland. 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)* 20, 1 (1973), 46–61.

[43] Takaaki Matsuo, Clément Durand, and Rodney Van Meter. 2019. Quantum link bootstrapping using a RuleSet-based communication protocol. *Physical Review A* 100, 5 (2019), 052320.

[44] David L Moehring, Peter Maunz, Steve Olmschenk, Kelly C Younge, Dzmitry N Matsukevich, L-M Duan, and Christopher Monroe. 2007. Entanglement of single-atom quantum bits at a distance. *Nature* 449, 7158 (2007), 68–71.

[45] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail D. Lukin, and Liang Jiang. 2016. Optimal architectures for long distance quantum communication. *Scientific Reports* 6, 1 (2016). https://doi.org/10.1038/srep20463

[46] Klaus Neumann and Jürgen Zimmermann. 1999. Methods for resource-constrained project scheduling with regular and nonregular objective functions and schedule-dependent time windows. In *Project Scheduling*. Springer, 261–287.

[47] Mihir Pant, Hari Krovi, Don Towsley, Leandros Tassiulas, Liang Jiang, Prithwish Basu, Dirk Englund, and Saikat Guha. 2019. Routing entanglement in the quantum internet. *npj Quantum Information* 5, 1 (2019), 1–9.

[48] Dionysia K Petraki, Markos P Anastasopoulos, Athanasios D Panagopoulos, and Panayotis G Cottis. 2007. Dynamic resource calculation algorithm in MF-TDMA satellite networks. In *2007 16th IST Mobile and Wireless Communications Summit*. IEEE, 1–5.

[49] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.

[50] Ram Ramanathan. 1997. A unified framework and algorithm for (T/F/C) DMA channel assignment in wireless networks. In *Proceedings of INFOCOM'97*, Vol. 2. IEEE, 900–907.

[51] Andreas Reiserer, Norbert Kalb, Machiel S Blok, Koen JM van Bemmelen, Tim H Taminiau, Ronald Hanson, Daniel J Twitchen, and Matthew Markham. 2016. Robust quantum-network memory using decoherence-protected subspaces of nuclear spins. *Physical Review X* 6, 2 (2016), 021040.

[52] Theodoros Salonidis and Leandros Tassiulas. 2005. Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. 145–156.

[53] Eddie Schoute, Laura Mancinska, Tanvirul Islam, Iordanis Kerenidis, and Stephanie Wehner. 2016. Shortcuts to quantum network routing. (2016). https://arxiv.org/abs/1610.05238.

[54] Javier Serrano, M Lipinski, T Wlostowski, E Gousiou, Erik van der Bij, M Cattin, and G Daniluk. 2013. The white rabbit project. (2013).

[55] Shouqian Shi and Chen Qian. 2020. Concurrent Entanglement Routing for Quantum Networks: Model and Designs. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 62–75.

[56] Rodney Van Meter. 2014. *Quantum networking*. John Wiley & Sons.

[57] Rodney Van Meter, Thaddeus D Ladd, William J Munro, and Kae Nemoto. 2008. System design for a long-line quantum repeater. *IEEE/ACM Transactions on Networking* 17, 3 (2008), 1002–1013.

[58] Rodney Van Meter and Joe Touch. 2013. Designing quantum repeater networks. *IEEE Communications Magazine* 51, 8 (2013), 64–71.

[59] Gayane Vardoyan, Saikat Guha, Philippe Nain, and Don Towsley. 2019. On the Capacity Region of Bipartite and Tripartite Entanglement Switching. *arXiv preprint arXiv:1901.06786* (2019).

[60] Stephanie Wehner, David Elkouss, and Ronald Hanson. 2018. Quantum internet: A vision for the road ahead. *Science* 362, 6412 (2018), eaam9288.

[61] Reinhard F Werner. 1989. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Physical Review A* 40, 8 (1989), 4277.

[62] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, Guang-Bing Li, Qi-Ming Lu, Yun-Hong Gong, Yu Xu, Shuang-Lin Li, Feng-Zhi Li, Ya-Yun Yin, Zi-Qing Jiang, Ming Li, Jian-Jun Jia, Ge Ren, Dong He, Yi-Lin Zhou, Xiao-Xiang Zhang, Na Wang, Xiang Chang, Zhen-Cai Zhu, Nai-Le Liu, Yu-Ao Chen, Chao-Yang Lu, Rong Shu, Cheng-Zhi Peng, Jian-Yu Wang, and Jian-Wei Pan. 2017. Satellite-based entanglement distribution over 1200 kilometers. *Science* 356, 6343 (2017), 1140–1144. https://doi.org/10.1126/science.aan3211

[63] Yong Yu, Fei Ma, Xi-Yu Luo, Bo Jing, Peng-Fei Sun, Ren-Zhou Fang, Chao-Wei Yang, Hui Liu, Ming-Yang Zheng, Xiu-Ping Xie, Wei-Jun Zhang, Li-Xing You, Zhen Wang, Teng-Yun Chen, Qiang Zhang, Xiao-Hui Bao, and Jian-Wei Pan. 2020. Entanglement of two quantum memories via fibres over dozens of kilometres. *Nature* 587 (2020), 240–245.

A APPENDIX

In this Appendix we briefly describe our methodology. Additional information, code, and data analysis may be found online at [omitted for blinding] [22].

A.1 Methodology

A.1.1 Concrete Repeater Protocol Generation. In our simulations we have assumed that the state of entangled links are of the Werner form [61] which corresponds to a "worst-case" quantum state that is a probabilistic mixture of the perfectly entangled state $|\Phi^+\rangle$ and a separable (not entangled) state $\mathbb{I}$.

We extend the ESSS algorithm [5] to consider entanglement swapping distillation at the pivot node level. When a quantum repeater protocol is constructed by performing an entanglement swap of two protocols on either side of a pivot P, we consider quantum repeater protocols that produce several such links and then perform entanglement distillation to reach fidelity requirements on the protocol. Our search assumes two-to-one entanglement distillation is performed and considers two styles of entanglement distillation known as *entanglement pumping* and *nested entanglement pumping* [10, 19, 20].

Our modified ESSS algorithm represents specifications in the form of a directed acyclic graph (DAG) $P(A, I)$. The set of vertices, A, represents the set of operations in the quantum repeater protocol whereas the set of edges, I, represents the dependency relation between operations in the protocol. Each operation $a = (a_{ID}, a_V, a_F, a_R) \in A$ specifies an operation type, a_{ID}, the set of nodes performing the operation, a_V, the fidelity of the link upon performing the operation, a_F, and the amount of time to allocate to the operation a_R. For elementary entangled link generation, a_R is chosen as the expected amount of time to generate the link with fidelity a_F.

Since operations in a quantum repeater protocol $P(A, I)$ have a consumer/producer relationship and the dependency relationship on operations takes the form of a tree structure, we determine the resource mapping Q and relative offset mapping M for the concrete repeater protocol $P(A, I, M, Q)$ by performing a post-order traversal of the protocol. When a source (link establishment operation) node a is reached, the operation is mapped to communication qubits and storage qubits held by the network nodes described by a_V. In our model (Section 5), we assume that any communication qubit may move a link into any storage qubit held by the same node. We thus always assign a vacant communication qubit and a vacant storage qubit to the operation so that a newly established link may immediately be stored and free the communication qubit for subsequent link establishment. When no storage qubits are available, the entangled link remains in the communication qubit until it is freed by an entanglement swap or entanglement distillation that consumes the held link. $Q(a)$ for a link establishment operation a is then the set of communication qubits and storage qubits chosen for an operation a. The assigned qubits are propagated through the protocol DAG to entanglement swaps and entanglement distillation operations that consume the qubits. This ensures that the entangled links in the qubits are correctly paired with one another for entanglement swapping and entanglement distillation.

To determine M, we produce a "mini" schedule for $P(A, I)$. We track the vacancy/occupation of communication qubits and storage qubits in time as they are used by quantum repeater protocol operations. Each operation $a \in A$ for a quantum repeater protocol $P(A, I)$ is associated with an amount of time a_R to spend performing the operation. For link establishment, this time corresponds to the expected latency of entanglement establishment between the two connected network nodes $u, v \in a_V$ for the given fidelity a_F. Our quantum repeater protocols aim to produce a single entangled link per operation, thus operations for link establishment are allocated enough slots to permit executing for a_R. For entanglement swapping and entanglement distillation operations, a_R corresponds to the latency of performing the operation. Operations for entanglement swapping and entanglement distillation need only be allocated enough slots in the schedule to execute the operation once.

To reduce protocol latency and preserve link fidelity, operations are scheduled in two passes. The first pass determines an as-soon-as-possible (ASAP) scheduling of the protocol operations. This determines the latency of the protocol and finds an initial scheduling. The schedule is then processed again to push link establishment as-late-as-possible (ALAP) while keeping all entanglement swap and entanglement distillation operations in place. Doing so reduces that amount of time a link may need to be stored before being consumed by one of these operations.

A.1.2 Periodic Task Scheduling. Here, we briefly describe our scheduling approach based on non-preemptive periodic task scheduling as relevant to our RCPSP method. For more information about the periodic task scheduling problem, see [15, 42]. Our notation for periodic task scheduling is as follows:

- $\tau_i = (\Phi_i, C_i, T_i)$: The ith task, defined by three parameters: The release offset Φ_i (set to 0), the worst-case execution time C_i, and the period T_i.
- $\tau_{i,j}$: The jth instance of task τ_i.
- C_i: The worst-case execution time of ith task. Each instance $\tau_{i,j}$ of τ_i has this worst-case execution time.
- T_i: The period of task τ_i, specifies the amount of time between subsequent releases of instances of τ_i.
- Γ: A set of tasks.

- H: the hyperperiod of the schedule, computed by the least common multiple (LCM) of the task periods $LCM(T_1, ..., T_n)$.

In this method, each protocol is converted into a periodic task capturing QoS requirements of the corresponding demand. Producing a schedule $\mathcal{S}$ for the periodic task set provides a set of start times for each concrete repeater protocol $P_i(A_i, I_i, M_i, Q_i)$. The network schedule may be produced by allocating slots to activities based on the starting slot of a repeater protocol in the periodic task schedule and the relative time mapping M_i.

A.1.3 Resource-Constrained Project Scheduling. Here we provide a description of our resource-constrained project scheduling (RCPSP) method for the problem of constructing schedules of quantum repeater protocols. For additional background on the RCPSP, see [11]. We first present a definition of the notation we use here for the RCPSP:

- j_i: An activity in the project.
- j_s, j_e: Dummy start and end activities in an activity-on-node network. Has zero processing time and no resource requirements.
- p_i: Processing time of activity j_i.
- K: The set of all resources.
- h_{ik}: Required number of resource k by activity j_i to execute. j_i may not begin unless h_{ik} units of resource k are available.

The RCPSP method first constructs an instance of the activity-on-node network for each concrete repeater protocol $P_i(A_i, I_i, M_i, Q_i)$ in $\mathcal{P}$. At this level, minimal/maximal time lags between activities are chosen to respect the relative time mapping M_i while the project activities themselves are non-preemptable to respect the estimated end-to-end fidelity of the repeater protocol. The set of resources K is the set of communication and storage qubits present at the nodes in the network. First, we show how we construct the activity-on-node network for a single concrete repeater protocol and then show how the full activity-on-node network is constructed for the scheduling problem.

Producing an instance of the activity-on-node network for a concrete repeater protocol $P(A, I, M, Q)$ begins by constructing dummy start and end activities j_s and j_e with processing times $p_s = p_e = 0$ and no resource requirements. A project activity j_i is constructed for each repeater protocol activity $a \in A$ with processing time $p_i = \lceil \frac{a_R}{t_{slot}} \rceil$ and resource requirements $h_{ik} = 1$ for all $k \in Q(a)$. Qubits that store entangled links remain occupied until the link is used in an entanglement swap or entanglement distillation, thus we add activities to the activity-on-node network reflecting the occupation of network resources between activities. Timing constraints are added between nodes in the activity on node network based on the relative time map M.

The activity-on-node network for the full scheduling problem must be constructed to reflect the rate requirements of the network demand. To determine the number of each demand's activity-on-node network, we reuse the notion of the hyperperiod H from the periodic task scheduling method. We first compute the period for each protocol as $T = \frac{1}{t_{slot} R_{min,i}}$ and compute the hyperperiod H as before. The number of instances N of protocol $P(A, I, M, Q)$ needed to obtain an average rate of R_{min} is then $\frac{H}{T_i}$. Constructing the full activity-on-node network is done by instantiating N instances of each protocol's activity-on-node network and connecting each instance's dummy start and dummy end to a common dummy start and end activity for the full project. Minimal and maximal time lags are added between the dummy start and end nodes that reflect release and deadline of the activity-on-node network every T time units. While we have not done so in this work, the minimal and maximal time lags may be set in order to control the jitter experienced by users.

The number of activities in the constructed activity-on-node network scales as $O(NS)$ where N is the total number of repeater protocol instances encoded in the final network and S is the maximum number of actions in an instance of a repeater protocol. This results from each activity in the activity-on-node network corresponding to an operation for an instance of a repeater protocol in the network. Since we construct multiple instances of each repeater protocol based on the rate it is demanded to execute, a repeater protocol's operations may appear multiple times in the final network.

To reduce the run-time complexity of producing the schedule, we consider a modification to our method for constructing an activity-on-node network for an instance of a concrete repeater protocol. This heuristic, RCPSP-NP-FPR (**F**ull **P**rotocol **R**eservation), produces an activity-on-node network that attempts to reserve all resources used by a concrete repeater protocol for the full duration of a protocol. To achieve this, one can construct an activity-on-node network for an instance of a concrete repeater protocol using three activities. First, an activity is made with execution time p equal to the repeater protocol latency (in slots) and resource requirements $h_l = 1$ for each qubit k_l used by the protocol. That is, $h_l = 1$ if $l \in \cup_{a \in A} Q(a)$. We then attach a dummy start and dummy end activity to this activity with minimal and maximal time lags of 0. Constructing the full activity-on-node network for the scheduling problem then uses these three-node activity-on-node networks for each repeater protocol instance in the final network. As a result, the size of the full activity-on-node network scales as $O(N)$ given each individual activity-on-node network has constant size which reduces the overall run-time complexity to $O(N^2|K| \log(N)$.