Delft University of Technology

Designing a quantum network protocol

Kozlowski, W.; Dahlberg, E.A.; Wehner, S.D.C.

DOI
10.1145/3386367.3431293

Publication date
2020

Document Version
Final published version

Published in
Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies

Citation (APA)
Kozlowski, W., Dahlberg, E. A., & Wehner, S. D. C. (2020). Designing a quantum network protocol. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies* (pp. 1-16). (CoNEXT 2020 - Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies). ACM. https://doi.org/10.1145/3386367.3431293

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

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].

This work is licensed under a Creative Commons Attribution International 4.0 License.

CoNEXT '20, December 1–4, 2020, Barcelona, Spain
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7948-9/20/12.
https://doi.org/10.1145/3386367.3431293

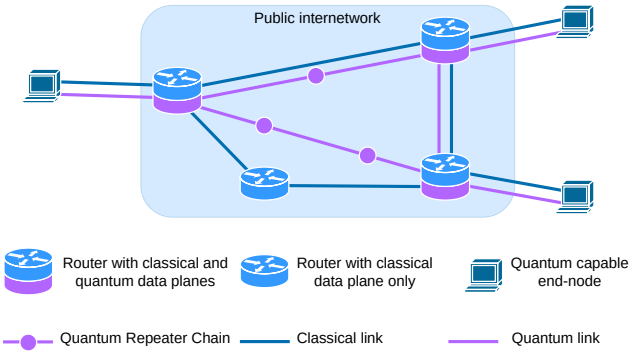

Figure 1: Quantum networks will use existing network infrastructure to exchange classical messages for the purposes of running quantum protocols as well as the control and management of the network itself. Long-distance links will be built using chains of automated quantum repeaters.

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 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:

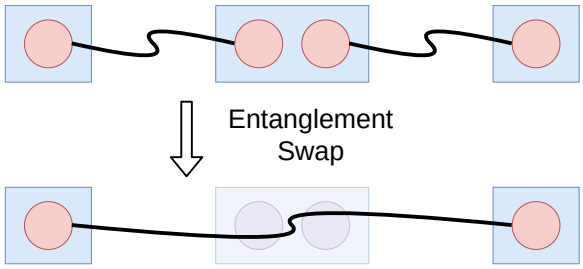

Figure 3: Quantum repeaters create long-distance entanglement by connecting short-distance entangled pairs. Initially two entangled pairs each have one qubit in the memory of the middle repeater. An entanglement swap is performed on these qubits which destroys the entanglement of the two pairs, but as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

(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.

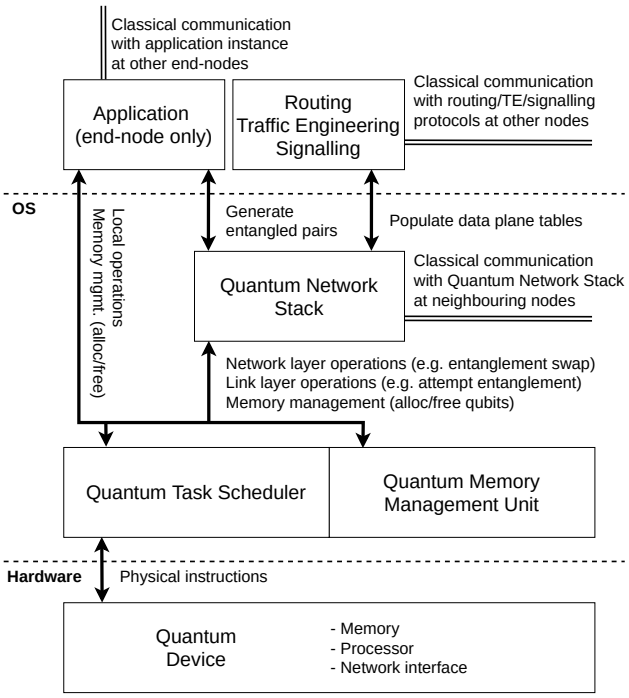

Figure 4: Local system components of a quantum node. The quantum memory, processor, and network interface are all one hardware component on current platforms. Local gate and network operations are performed on qubits in the main memory. Management and arbitration of local hardware resources belong to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

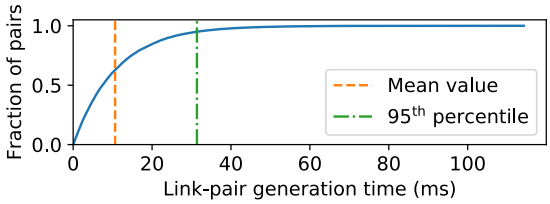

Figure 5: The cumulative distribution function for the time taken to generate a link-pair of fidelity 0.95 over a 2 m long fibre with the same hardware parameters as used in Sec. 5. The y-axis denotes the fraction of pairs generated in less than the time indicated on the x-axis. We see that on average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

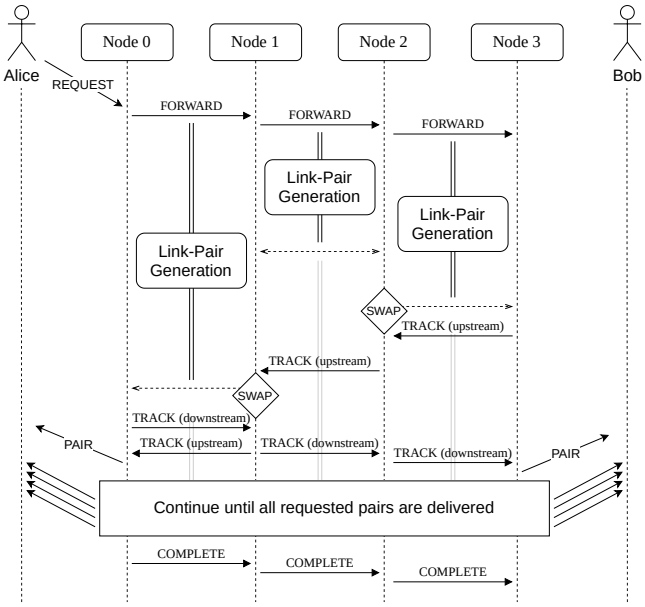

Figure 6: Example sequence of the QNP.

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.

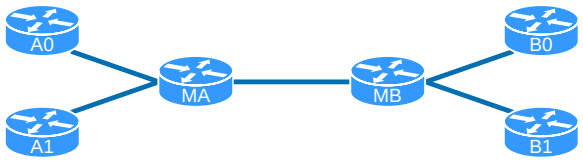

Figure 7: The evaluation topology. MA-MB is a bottleneck link between the A nodes and the B nodes. All links consist of a quantum and a classical channel.

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

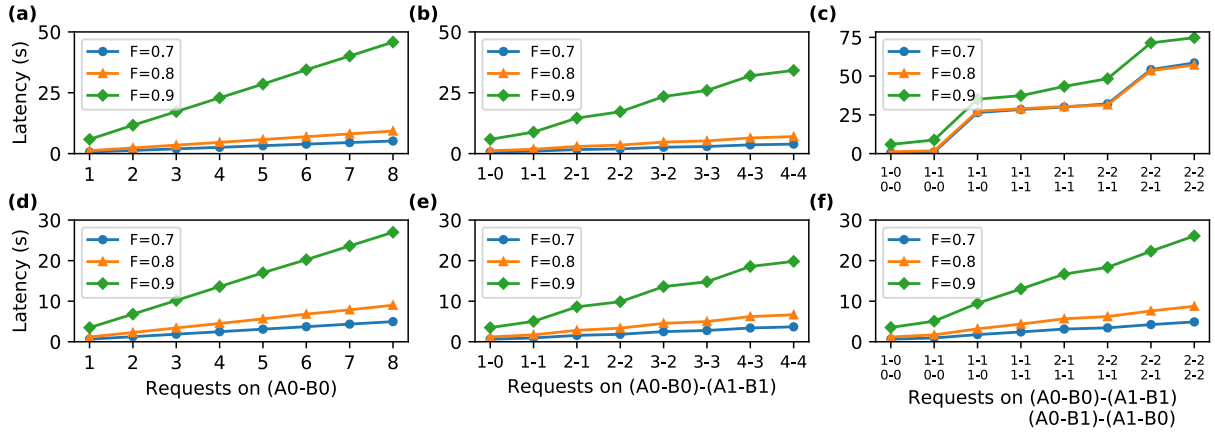

Figure 8: Average latency of requests on the A0-B0 circuit when 1–8 simultaneous requests, each for 100 pairs, are issued across (a,d) 1 circuit (A0-B0), (b,e) 2 circuits (A0-B0, A1-B1), and (c,f) 4 circuits (A0-B0, A1-B1, A0-B1, A1-B0). We consider a long (a-c) and short (d-f) cutoff time (see main 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 1st and 5th requests, circuit A1-B1: the 2nd and 6th, A0-B1: the 3rd, and A1-B0: the 4th. 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

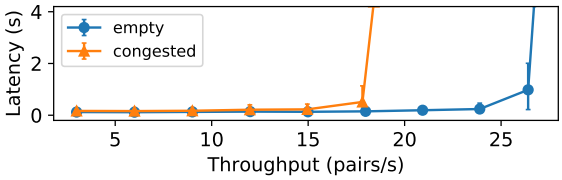

Figure 9: Average latency vs. throughput of A0-B0 circuit as we increase the rate of 3-pair requests over A0-B0. In the "empty" case, there is no other traffic in the network. In the "congested" case, there is a long running flow on A1-B1 at the same time competing for the bottleneck link. Error bars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

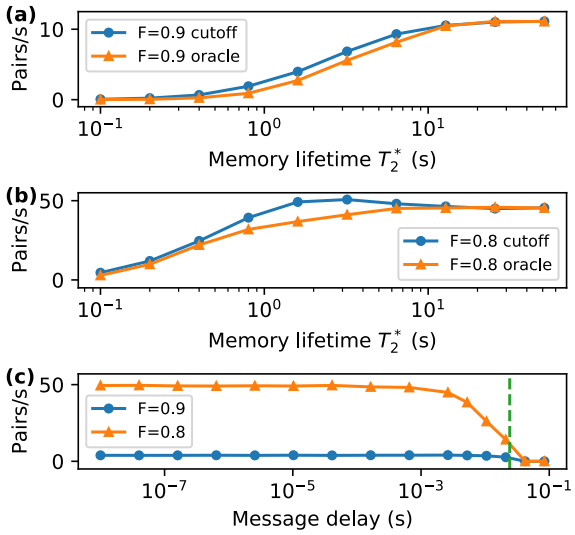

Figure 10: Robustness against decoherence. (a,b) show the effects of short memory lifetime on the throughput of the two competing circuits. Note that the F=0.9 with cutoff throughput becomes low, but not zero. (c) shows the effects of classical message delay (time from sending from one node to processing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

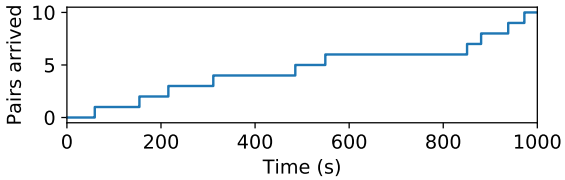

Figure 11: The number of pairs produced as a function of time on a near-future network. The protocol is able to deliver entanglement despite the limited resources.

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].