A Link Layer Protocol for Quantum Networks

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

ABSTRACT

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

CCS CONCEPTS

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

KEYWORDS

Quantum Internet, Quantum Networks, Link Layer

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

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

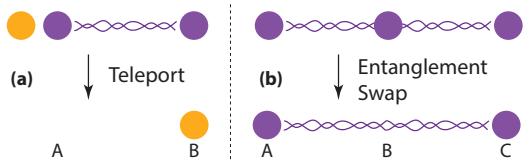

Figure 1: Entanglement enables long-distance quantum communication: (a) once two qubits (purple/dark) are confirmed to be entangled (threaded links between qubits), a data qubit (yellow/light) can be sent deterministically using teleportation [11], consuming the entangled pair; (b) long-distance entanglement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

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

1 INTRODUCTION

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

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

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

*The first two authors contributed equally to this work.

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

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

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

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

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

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

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

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

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

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

2 RELATED WORK

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

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

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

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

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

3 DESIGN CONSIDERATIONS FOR QUANTUM NETWORK ARCHITECTURES

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

3.1 Qubits and Entanglement

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

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

3.2 Quantum Network Devices

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

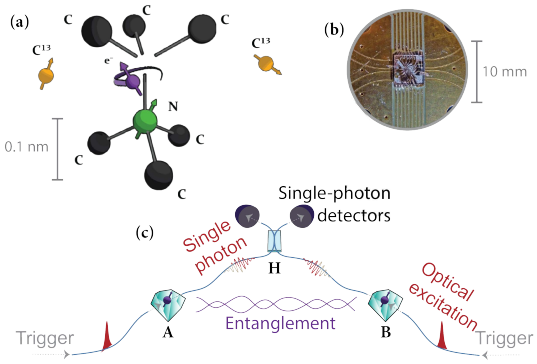

Figure 3: Heralded entanglement generation on the NV platform. (a) NV centers are point defects in diamond with an electronic spin as a communication qubit (purple) and carbon-13 nuclear spins as memory qubits (yellow), realized in custom chips (b). (c) A trigger produces entanglement between the communication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

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

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

3.3 Use Cases

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

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

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

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

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

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

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

3.4 Network Stack

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

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

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

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

4 DESIGN CONSIDERATIONS FOR QUANTUM LINK LAYER

4.1 Desired Service

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

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

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

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

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

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

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

4.2 Performance Metrics

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

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

4.3 Error Detection

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

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

4.4 Physical Entanglement Generation

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

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

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

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

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

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

4.5 Hardware Considerations

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

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

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

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

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

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

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

5 PROTOCOLS

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

5.1 Physical Layer MHP

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

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

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

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

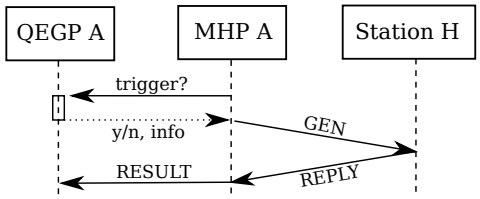

Figure 4: Timeline of the MHP polling higher layers to see if entanglement should be produced.

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

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

5.2 Link Layer QEGP

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

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

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

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

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

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

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

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

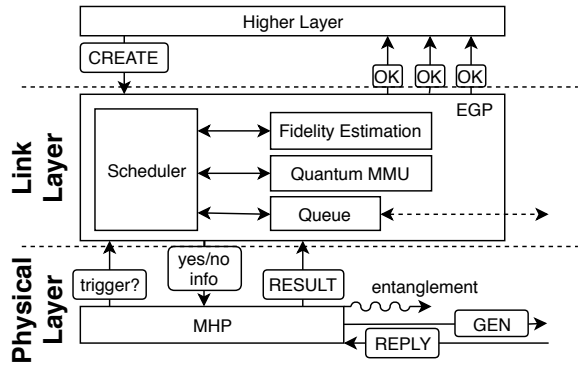

Figure 5: Flow diagram of the MHP and QEGP operation. The QEGP handles CREATE requests and schedules entanglement generation attempts are issued to the MHP. Replies from the midpoint are parsed and forwarded to the QEGP from request management.

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

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

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

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

6 EVALUATION

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

We conduct the following simulation runs:

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

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

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

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

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

6.1 Robustness

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

6.2 Performance Metrics

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

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

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

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

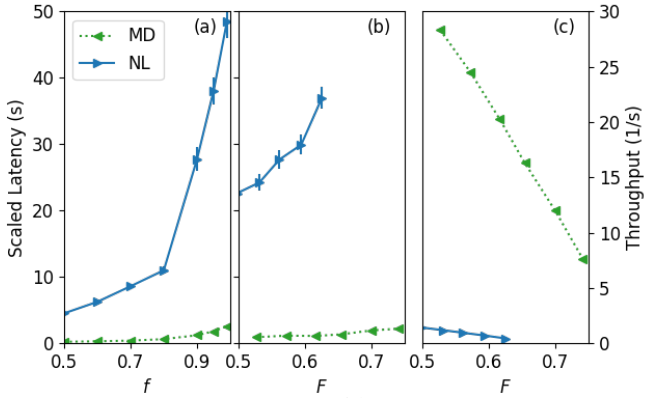

Figure 6: Performance trade-offs. (a) Scaled latency vs. f_P determining fraction of throughput (b) Scaled latency vs. fidelity F_{min}. Demanding a higher F_{min} lowers the probability of success (Section 4.4), meaning (c) throughput directly scales with F_{min} (each point averaged over 40 short runs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

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

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

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

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

6.3 Scheduling

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

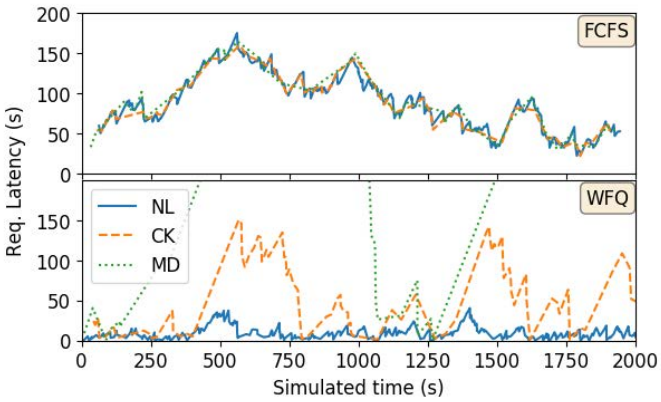

Figure 7: Request latency vs. time for two scheduling scenarios (long runs simulated 120 h wall time). As expected the max. latency for *NL* is decreased due to strict priority. In this scenario, there are more incoming *NL* requests ($f_{NL} = 0.99 \cdot 4/5$, $f_{CK} = 0.99 \cdot 1/5$ and $f_{MD} =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

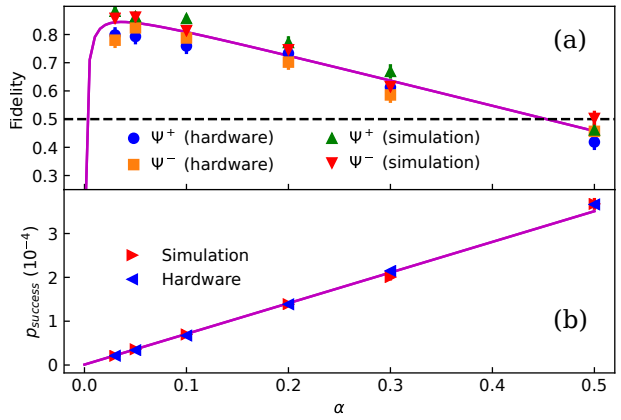

Figure 8: Validation against data from NV hardware (Lab scenario). Fidelity (a) and probability an attempt succeeds (b) in terms of α (Section 4.4) shows good agreement between hardware and simulation points (each at least 300 pairs averaged, 5s–117s simulated time, 500k–10.000k attempts, 122 hours wall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

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

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

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

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

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

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

7 CONCLUSION

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

ACKNOWLEDGEMENTS

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

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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