Quantum Internet Research Group AD. Dahlberg
Internet-Draft MS. Skrzypczyk
Intended status: Experimental SW. Wehner, Ed.
Expires: April 12, 2020 QuTech, Delft University of Technology
October 10, 2019

The Link Layer service in a Quantum Internet
draft-dahlberg-ll-quantum-03

Abstract

In a classical network the link layer is responsible for transferring a datagram between two nodes that are connected by a single link, possibly including switches. In a quantum network however, the link layer will need to provide a robust entanglement generation service between two quantum nodes which are connected by a quantum link. This service can be used by higher layers to produce entanglement between distant nodes or to perform other operations such as qubit transmission, without full knowledge of the underlying hardware and its parameters. This draft defines what can be expected from the service provided by a link layer for a Quantum Network and defines an interface between higher layers and the link layer.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The most important fundamental operation in a quantum network is the generation of entanglement between nodes. Short-distance entanglement can be used to generate long-distance entanglement with the use of an operation called entanglement swap [1] (also formalised in [2]). If nodes A and B share an entangled pair and similarly for B and C, B can perform a so called Bell measurement [3] and send the measurement outcome (2 bits) over a classical channel to A or C such that in the end A and C share an entangled pair. Furthermore, long-distance entanglement does in turn enable long-distance qubit transmission by the use of quantum teleportation [3] (also formalised in [2]). Node A can teleport an unknown qubit state to B by consuming an entangled pair between A and B and sending two classical bits to B. For an overview of quantum networking and its applications we refer to [5].

Long lived entanglement between distant nodes capable of storing such entanglement has been demonstrated over a distance of up to 1.3 km [4], in a proof-of-principle experiment. This entanglement was also heralded, that is, there exits a so-called heralding signal that indicates success in entanglement production without consuming such entanglement. Short lived and non-heralded entanglement has been observed from a satellite over a distance of 1200 km [6] in a proof of principle experiment. The next step towards a quantum network is to turn ad-hoc experiments that produce entanglement into a reliable service. This is the role of the link layer, which turns an ad-hoc physical setup to a reliable entanglement generation service. Reliable here means that the higher layers can (unless a timeout or other critical failures occur) rely in deterministic entanglement production. In particular, this means that since the underlying physical process is often probabilistic but entanglement generation can be confirmed using the heralding signal, one of the main tasks of the link layer is to manage re-tries in producing entanglement at the the physical layer. Once an entangled pair has been generated, the nodes need to be able to agree on which qubits are involved in which entangled pair in order to use it, thus another main task of the link layer is to provide an entanglement identifier.

2. Scope

This draft is meant to define the service and interface of an link layer of a quantum network. Further considerations that motivate this definition can be found in [7]. It does not present a protocol realising this service. However a protocol that indeed does this have been proposed in [7], together with more details on use cases and design decisions in forming a quantum network stack.

3. Desired service

This section definces the service that a link layer provides in a quantum network. The interface and header specification is defined in the next section.

A link layer between two nodes A and B of a quantum network must provide the following minimal features (see [7] for an extended feature set):

o Allow both node A and B to initialize entanglement generation.

o Allow the initializing node to specify a desired minimum fidelity[3] and maximum waiting time.

o Notify both nodes of success or failure of entanglement generation before the requested maximum waiting time has passed since the request was initialized.

o If success is notified, the generated entangled pair has with high confidence higher (or equal) fidelity than the desired minimum fidelity.

- For a successful request, provide an entanglement identifier to allow higher layers to use identify the entangled pair in the network without the need for further communication.

4. Interface

This section describes the interface between higher layers and the link layer in a quantum network, along with header specifications for the type of messages. The interface consists of a single type of message from the higher layers to the link layer, which is the CREATE message for requesting entanglement generation. Response messages from the link layer to the higher layers take either the form of an ACK, an OK message or one of many error messages. The ACK is sent back directly upon receiving a CREATE if the link layer supports the request and contains a CREATE ID such that the higher layer can associated the subsequent OK messages to the correct request. It is assumed that the nodes in the network are assigned a unique ID in the network, which is used in the Remote Node ID parameters of the messages below.

4.1. Higher layers to link layer

The higher layers can send a CREATE message to the link layer to request the generation of entanglement. Along with other parameters, as specified below the higher layers can specify a minimum fidelity, a maximum waiting time and the number of entangled pairs to be produced.

4.1.1. Header specification

The CREATE message contains the following parameters:

- Remote Node ID (32 bits): Used if the node is directly connected to multiple nodes. Indicates which node to generate entanglement with.

- Minimum fidelity (16 bits): The desired minimum fidelity, between 0 and 1, of the generated entangled pair. A binary value encoding the integer 'n' represents the fidelity 'n' divided by (2^16-1).

- Time Unit (TU) (2 bits): The time units used for specifying Max Time, where (00, 01, 10) each indicate (micro-seconds, milliseconds, seconds) respectively and 11 is unused.

- Max Time (14 bits): The maximum time in the time units specified above that the higher layer is willing to wait for the request to be fulfilled. A binary value encoding the integer 'n' representing the time in the specified time units.

- o Purpose ID (16 bits): Allows the higher layer to tag the request for a specific purpose. If the request is from an application this can be thought of as a port number. The purpose ID can also be used by a network layer to specify that this entanglement request is part of long-distance entanglement generation over a specific path.

- o Number (16 bits): The number of entangled pairs to generate.

- o Priority (3 bits): Can be used to indicate if this request is of high priority and should ideally be fulfilled early. Higher means faster service.

- o Type of request (TPE) (1 bit): Either create and keep (K) or measure directly (M), where K stores the generated entanglement in memory and M measures the entanglement directly.

- o Atomic (ATO) (1 bit): A flag that indicates that the request should be satisfied as a whole without interuption by other requests.

- o Consecutive (CON) (1 bit): A flag indicating an OK is returned for each pair made for a request. Otherwise, an OK is sent only when the entire request is completed (more common in application use cases). For K type requests, this means all pair should be in memory at the same time.

- o Random basis choice for measure directly

  * (RL) (2 bits): Choose to measure the local qubit randomly in either

  * (RR) (2 bits): Choose to measure the remote qubit randomly in either

  Using the following encoding:

  * 00: No random choice

  * 01: X or Z basis (BB84)

  * 10: X, Y or Z basis (six state)

  * 11: CHSH rotated bases, Z basis rotated by angles +/- pi/4 around Y axis.

- o Probability distributions used to sample random basis for measure directly:

* (PL1) (8 bits): Parameter for local probability distribution used to sample basis if RL is not 00

* (PL2) (8 bits): Parameter for local probability distribution used to sample basis if RL is not 00

* (PR1) (8 bits): Parameter for remote probability distribution used to sample basis if RR is not 00

* (PR2) (8 bits): Parameter for remote probability distribution used to sample basis if RR is not 00

Each value is seen as the integer representing of the binary value. Probability distributions are used as follows

* If the specified random basis has 2 elements then the distribution obeys the probabilities (PL(R)1 / 255, 1 - PL(R)1 / 255)

* If the specified random basis has 3 elements then the distribution obeys the probabilities (PL(R)1 / 255, PL(R)2 / 255, 1 - PL(R)1 / 255 - PL(R)2 / 255)

o Rotation of measurement basis in the case of M types of requests for both the local and remote measurement. Three rotations from the defaults Z basis are performed, first a rotation around the X-axis (ROTX1L(R)), then a rotation around the Y-axis (ROTYL(R)) and finally a rotation again around the X-axis. Note that arbitrary rotations can be composed as these three rotations, see <https://en.wikipedia.org/wiki/Euler_angles>. If all three fields are 00000000, the qubits are measured in the Z basis. If RL(R) is not 00, these three fields (ROTX1L(R), ROTYL(R) and ROTX2L(R)) are ignored.

  * Measurement rotation around X for local (remote) node (ROTX1L(R)) (8 bits): Measurement to be performed in the case of M types of request. Default is Z measurement. Specified measurement to be rotated around the X axis by angle of 2 pi/256 * ROTX1

  * Measurement rotation around Y for local (remote) node (ROTYL(R)) (8 bits): Measurement to be performed in the case of M types of request. Default is Z measurement. Specified measurement to be rotated around the Y axis by an angle of 2 pi/256 * ROTY

  * Measurement rotation around X for local (remote) node (ROTX2L(R)) (8 bits): Measurement to be performed in the case

of M types of request. Default is Z measurement. Specified measurement to be rotated around the X axis by an angle of 2 pi/256 * ROTX2

The complete header specification of the CREATE message is given in Figure 1.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Remote Node ID                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       Minimum Fidelity        |TU |       Max Time            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Purpose ID           |            Number             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Prio |T|A|C|   |   |           |               |               |
   |rity |P|T|O|RL |RR | reserved  |      PL1      |      PL2      |
   |     |E|O|N|   |   |           |               |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               |               |               |               |
   |      PR1      |      PR2      |     ROTX1L    |     ROTXYL    |
   |               |               |               |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               |               |               |               |
   |     ROTX2L    |     ROTX1R    |     ROTYR     |     ROTX2R    |
   |               |               |               |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: CREATE message header format

4.2. Link layer to higher layers

When receiving a CREATE message from higher layers the link layer will directly respond and notify the higher layer whether requests will be scheduled for generation. If so the link layer responds with an ACK containing a CREATE ID. The higher layer may choose to use this CREATE ID together with the ID of the requesting node to associate OK messages it receives from the link layer to the correct request. Note that the ID of the requesting node is needed since the ACK is returned directly and the CREATE ID is thus not unique for requests from different nodes. If the link layer does not support the given request an error message is instead returned.

When a request is satisfied an OK message is sent to the higher layer. The OK message contains different fields depending on whether the request was of type K (keep) or M (measure directly). For K the OK contains a logical qubit identifier (LQID) such that the higher layer can know which logical qubit holds the generated entanglement. For M the OK contains the basis which the qubit was measured and the measurement outcome.

Both during and after entanglement generation, the link layer can return error messages to the higher layers, as further described below. For example if something happens to the qubit or another error occurs such that the entanglement is not valid anymore, the link layer can issue an ERR_EXPIRE message.

4.2.1. Header specification

To distinguish the different types of messages that the link layer can return to the higher layer, the first part of the header is a 4 bit field which specifies the type of message using the following mapping:

- o 0001: ACK
- o 0010: Type K OK
- o 0011: Type M OK
- o 0100: ERR

The complete header specification for these four types of messages are shown below in Figure 2 to Figure 5.

The ACK message contains the following parameters:

- o Create ID (16 bits): A Create ID that the higher layer can use to associate subsequent OK messages to the request.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type  |           Create ID           |         Unused        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ACK message header format

The type K OK message contains the following parameters:

- o Create ID (16 bits): Must be the same Create ID that was given in the ACK of the corresponding request.
- o Logical Qubit ID (LQID) (4 bits): A logical ID of the qubit which is part of the entangled pair.

o Directionality flag (D) (1 bit): Specifies if the request came from this node (D=0) or from the remote node (D=1).

o Sequence number (16 bits): A sequence number for identifying the entangled pair. It is assumed to be unique for entangled pairs between the given nodes. Thus together with the IDs of the nodes between which the entanglement is produced, one can create an entanglement identifier which is unique in the network.

o Purpose ID (16 bits): The purpose ID of the request (only used by the node which did not initiate the request)

o Remote Node ID (32 bits): Used if the node is directly connected to multiple nodes.

o Goodness (16 bits): An estimate of the fidelity of the generated entangled pair. Should not be seen as a guarantee.

o Time of Goodness (ToG) (16 bits): The time of the goodness estimate. Not necessarily the time when the estimate is performed but rather the time for which the estimate is for. Can be used to make an updated estimate based on decoherence times of the qubits.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type  |            Create ID            | LQID  |D|   Unused    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sequence Number        |          Purpose ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote Node ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Goodness            |       Time of Goodness        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Type K OK message header format

The type M OK message contains the following parameters:

o Create ID (16 bits): The same Create ID that was given in the ACK of the corresponding request.

o Measurement outcome (M) (1 bit): The outcome of the measurement performed on the entangled pair.

o Basis (3 bits): Which basis the entangled pair was measured in, used if the basis is random, i.e. if RBC is not 00 in the CREATE. The following representation is used:

  * 000: Z-basis

  * 001: X-basis

  * 010: Y-basis

  * 011: Z-basis rotated by angle pi/4 around Y-axis

  * 100: Z-basis rotated by angle -pi/4 around Y-axis

  * 101: Unused

  * 110: Unused

  * 111: Unused

- Directionality flag (D) (1 bit): Specifies if the request came from this node (D=0) or from the remote node (D=1).

- Sequence number (16 bits): A sequence number for identifying the entangled pair. It is assumed to be unique for entangled pairs between the given nodes. Thus together with the IDs of the nodes, one can create an entanglement identifier which is unique in the network.

- Purpose ID (16 bits): The purpose ID of the request (only used by the node which did not initiate the request)

- Remote Node ID (32 bits): Used if the node is directly connected to multiple nodes.

- Goodness (16 bits): An estimate of the fidelity of the generated entangled pair. Should not be seen as a guarantee.

Note: Time of Goodness is not needed here since there is no decoherence on the measurement outcomes.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type  |             Create ID             |M|D|Basis|   Unused  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sequence Number        |           Purpose ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Remote Node ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Goodness           |             Unused            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Type M OK message header format

The ERR message contains the following parameters:

- Create ID (16 bits): The same Create ID that was given in the ACK of the corresponding request.

- Error code (ERR) (4 bits): Specifies what error occurred. See below what the error codes mean.

- Expire by sequence numbers (S) (1 bit): Used by ERR_EXPIRE, to specify whether a range of sequence numbers should be expired (S=1) or all sequence numbers associated with the given Create ID and Origin Node (S=0).

- Sequence number low (16 bits): Used by error code ERR_EXPIRE to identify a range of sequence numbers that needs to be expired. Numbers above Sequence number low (inclusive) and below Sequence number high (exclusive) should be expired.

- Sequence number high (16 bits): Used by error code ERR_EXPIRE to identify a range of sequence numbers that needs to be expired. Numbers above Sequence number low (inclusive) and below Sequence number high (exclusive) should be expired.

- Origin Node (32 bits): Used if the node is directly connected to multiple nodes. Needed here since Create IDs are not unique for request from different nodes.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type  |             Create ID           |  ERR  |S|   Unused    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Sequence number low        |     Sequence number high      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Origin Node                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Error message header format

The different error codes using in an error message are the following:

- Error returned directly when a CREATE message is received:

  * ERR_UNSUPP (0001): The given request is not supported. For example if the minimum fidelity is not achievable or if the request is of type K and the hardware cannot store entanglement.

  * ERR_CREATE (0010): The create message could not be parsed.

  * ERR_REJECTED (0011): The request was rejected by this node based on for example the Purpose ID.

  * ERR_OTHER (0100): An unknown error occurred.

- Error returned after a CREATE message is received, before or after an OK is returned:

  * ERR_EXPIRE (0101): One or more already sent OK messages have expired and the entangled pair is not available anymore. Can either be specified as a range of sequence numbers or by a create ID by using the S flag.

  * ERR_REJECTED (0011): The request was rejected by the other node based on for example the Purpose ID.

  * ERR_TIMEOUT (0110): The request was not satisfied within the requested max waiting time.

5. IANA Considerations

This memo includes no request to IANA.

6. Acknowledgements

The authors would like to acknowledge funding received from the EU Flagship on Quantum Technologies, Quantum Internet Alliance.

The authors would further like to acknowledge Tim Coopmans, Leon Wubben, Filip Rozpedek, Matteo Pompili, Arian Stolk, Przemyslaw Pawelczak, Robert Knegjens, Julio de Oliveria Filho, Sidney Cadot, Joris van Rantwijk and Ronald Hanson for inputs and discusssion and Wojciech Kozlowski for useful feedback on this draft.

7. Informative References

[1] Briegel, H., Dur, W., Cirac, J., and P. Zoller, "Quantum repeates: The Role of Imperfect Local Operations in Quantum Communication", Physical Review Letters 81, 26, 1998, <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.81.5932>.

[2] Kompella, K., Aelmans, M., Wehner, S., Sirbu, C., and A. Dahlberg, "Advertising Entanglement Capabilities in Quantum Networks", QIRG Internet-Draft, 2018, <https://datatracker.ietf.org/doc/draft-kaws-qirg-advent/>.

[3] Nielsen, M. and I. Chuang, "Quantum Computation and Quantum Information", Book Cambridge University Press, 2010, <https://doi.org/10.1017/CBO9780511976667>.

[4] Hensen, B., Bernien, H., Dreau, A., Reiserer, A., Kalb, N., Blok, M., Ruitenberg, J., Vermeulen, R., Schouten, R., Abellan, C., Amaya, W., Pruneri, V., Mitchell, M., Markham, M., Twitchen, D., Elkouss, D., Wehner, S., Taminiau, T., and R. Hanson, "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres", Nature 526, 682-686, 2015, <https://arxiv.org/abs/1508.05949>.

[5] Wehner, S., Elkouss, D., and R. Hanson, "Quantum internet: A vision for the road ahead", Science 362, 6412, 2018, <http://science.sciencemag.org/content/362/6412/eaam9288?intcmp=trendmd-sci>.

[6] Yin, J., Cao, Y., Li, Y., Liao, S., Zhang, L., Ren, J., Cai, W., Liu, W., Li, B., Dai, H., Li, G., Lu, Q., Gong, Y., Xu, Y., Li, S., Li, F., Yin, Y., Jiang, Z., Li, M., Jia, J., Ren, G., He, D., Zhou, Y., Zhang, X., Wang, N., Chang, X., Zhu, Z., Liu, N., Chen, Y., Lu, C., Shu, R., Peng, C., Wang, J., and J. Pan, "Satellite-based entanglement distribution over 1200 kilometers", Science 356, 6343, 2017, <https://arxiv.org/abs/1707.01339>.

[7] Dahlberg, A., Skrzypczyk, M., Coopmans, T., Wubben, L., Rozpedek, F., Pompili, M., Stolk, A., Pawelczak, P., Knegjens, R., de Oliveira Filho, J., Hanson, R., and S. Wehner, "A Link Layer Protocol for Quantum Networks", arXiv pre-print arXiv:1903.09778, 2019, <https://arxiv.org/abs/1903.09778>.

Authors' Addresses

Axel Dahlberg
QuTech, Delft University of Technology
Lorentzweg 1
Delft 2628 CJ
Netherlands

Phone: +31 (0)65 8966821
Email: e.a.dahlberg@tudelft.nl

Matthew Skrzypczyk
QuTech, Delft University of Technology
Lorentzweg 1
Delft 2628 CJ
Netherlands

Email: m.d.skrzypczyk@student.tudelft.nl

Stephanie Wehner (editor)
QuTech, Delft University of Technology
Lorentzweg 1
Delft 2628 CJ
Netherlands

Email: s.d.c.wehner@tudelft.nl

Quantum Internet Research Group W. Kozlowski
Internet-Draft S. Wehner
Intended status: Informational QuTech
Expires: 1 March 2023 R. Van Meter
Keio University
B. Rijsman
Individual
A. S. Cacciapuoti
M. Caleffi
University of Naples Federico II
S. Nagayama
Mercari, Inc.
28 August 2022

Architectural Principles for a Quantum Internet
draft-irtf-qirg-principles-11

Abstract

The vision of a quantum internet is to enhance existing Internet technology by enabling quantum communication between any two points on Earth. To achieve this goal, a quantum network stack should be built from the ground up to account for the fundamentally new properties of quantum entanglement. The first quantum entanglement networks have been realised [Pompili21.1], but there is no practical proposal for how to organise, utilise, and manage such networks. In this draft, we attempt to lay down the framework and introduce some basic architectural principles for a quantum internet. This is intended for general guidance and general interest, but also to provide a foundation for discussion between physicists and network specialists. This document is a product of the Quantum Internet Research Group (QIRG).

Status of This Memo

This Internet-Draft will expire on 1 March 2023.

Copyright Notice

Table of Contents

1. Introduction

Quantum networks are distributed systems of quantum devices that utilise fundamental quantum mechanical phenomena such as superposition, entanglement, and quantum measurement to achieve capabilities beyond what is possible with non-quantum (classical) networks [Kimble08]. Depending on the stage of a quantum network [Wehner18] such devices may range from simple photonic devices capable of preparing and measuring only one quantum bit (qubit) at a time all the way to large-scale quantum computers of the future. A quantum network is not meant to replace classical networks, but rather form an overall hybrid classical-quantum network supporting new capabilities which are otherwise impossible to realise [VanMeterBook]. For example, the most well-known application of quantum communication, quantum key distribution (QKD), can create and distribute a pair of symmetric encryption keys in such a way that the security of the entire process relies on the laws of physics (and thus can be mathematically proven to be unbreakable) rather than the intractability of certain mathematical problems [Bennett14] [Ekert91]. Small networks capable of QKD have even already been deployed at short (roughly 100km) distances [Elliott03] [Peev09] [Aguado19] [Joshi20].

The quantum networking paradigm also offers promise for a range of new applications beyond quantum cryptography, such as distributed quantum computation [Cirac99] [Crepeau02], secure quantum computing in the cloud [Fitzsimons17], quantum-enhanced measurement networks [Giovanetti04], or higher-precision, long-baseline telescopes [Gottesman12]. These applications are much more demanding than QKD and networks capable of executing them are in their infancy. The first fully quantum, multinode network capable of sending, receiving, and manipulating distributed quantum information has only recently been realized [Pompili21.1]

Whilst a lot of effort has gone into physically realising and connecting such devices, and making improvements to their speed and error tolerance, there are no worked out proposals for how to run these networks. To draw an analogy with a classical network, we are at a stage where we can start to physically connect our devices and send data, but all sending, receiving, buffer management, connection synchronisation, and so on, must be managed by the application directly by using low-level, custom-built, and hardware-specific interfaces, rather than being managed by a network stack that exposes a convenient high-level interface, such as sockets. Only recently, was the first ever attempt at such a network stack experimentally demonstrated in a laboratory setting [Pompili21.2]. Furthermore, whilst physical mechanisms for transmitting quantum information exist, there are no robust protocols for managing such transmissions.

This document, produced by the Quantum Internet Research Group (QIRG), introduces quantum networks and presents general guidelines for the design and construction of such networks. Overall, it is intended as an introduction to the subject for network engineers and researchers. It should not be considered as a conclusive statement on how quantum network should or will be implemented. This document was discussed on the QIRG mailing list and several IETF meetings and represents the consensus of the QIRG members, both of experts in the subject matter (from the quantum as well networking domain) as well as newcomers who are the target audience.

2. Quantum information

In order to understand the framework for quantum networking, a basic understanding of quantum information theory is necessary. The following sections aim to introduce the minimum amount of knowledge necessary to understand the principles of operation of a quantum network. This exposition was written with a classical networking audience in mind. It is assumed that the reader has never before been exposed to any quantum physics. We refer the reader to [SutorBook] and [NielsenChuang] for an in-depth introduction to quantum information systems.

2.1. Quantum state

A quantum mechanical system is described by its quantum state. A quantum state is an abstract object that provides a complete description of the system at that particular moment. When combined with the rules of the system's evolution in time, such as a quantum circuit, it also then provides a complete description of the system at all times. For the purposes of computing and networking, the classical equivalent of a quantum state would be a string or stream of logical bit values. These bits provide a complete description of what values we can read out from that string at that particular moment and when combined with its rules for evolution in time, such as a logical circuit, we will also know its value at any other time.

Just like a single classical bit, a quantum mechanical system can be simple and consist of a single particle, e.g. an atom or a photon of light. In this case, the quantum state provides the complete description of that one particle. Similarly, just like a string of bits consists of multiple bits, a single quantum state can be used to also describe an ensemble of many particles. However, because quantum states are governed by the laws of quantum mechanics their behaviour is significantly different to that of a string of bits. In this section we will summarise the key concepts to understand these differences and the we will explain their consequences for networking in the rest of the draft.

2.2. Qubit

The differences between quantum computation and classical computation begin at the bit-level. A classical computer operates on the binary alphabet { 0, 1 }. A quantum bit, called a qubit, exists over the same binary space, but unlike the classical bit, its state can exist in a superposition of the two possibilities:

|qubit> = a |0> + b |1>,

where |X> is Dirac's ket notation for a quantum state (the value that a qubit holds), here the binary 0 and 1, and the coefficients a and b are complex numbers called probability amplitudes. Physically, such a state can be realised using a variety of different technologies such as electron spin, photon polarisation, atomic energy levels, and so on.

Upon measurement, the qubit loses its superposition and irreversibly collapses into one of the two basis states, either |0> or |1>. Which of the two states it ends up in may not be deterministic, but can be determined from the readout of the measurement. The measurement result is a classical bit, 0 or 1, corresponding to |0> and |1> respectively. The probability of measuring the state in the |0> state is |a|^2 and similarly the probability of measuring the state in the |1> state is |b|^2, where |a|^2 + |b|^2 = 1. This randomness is not due to our ignorance of the underlying mechanisms, but rather is a fundamental feature of a quantum mechanical system [Aspect81].

The superposition property plays an important role in fundamental gate operations on qubits. Since a qubit can exist in a superposition of its basis states, the elementary quantum gates are able to act on all states of the superposition at the same time. For example, consider the NOT gate:

NOT (a |0> + b |1>) -> a |1> + b |0>.

It is important to note that "qubit" can have two meanings. In the first meaning, "qubit" refers to a physical quantum *system* whose quantum state can be expressed as a superposition of two basis states, which we often label |0> and |1>. Here, "qubit" refers to a physical implementation akin to what a flip-flop, switch, voltage, or current would be for a classical bit. In the second meaning, "qubit" refers to the abstract quantum *state* of a quantum system with such two basis states. In this case, the meaning of "qubit" is akin to the logical value of a bit, from classical computing, i.e. "logical 0" or "logical 1". The two concepts are related, because a physical "qubit" (first meaning) can be used to store the abstract "qubit" (second meaning). Both meanings are used interchangeably in literature and the meaning is generally clear from the context.

2.3. Multiple qubits

When multiple qubits are combined in a single quantum state the space of possible states grows exponentially and all these states can coexist in a superposition. For example, the general form of a two-qubit register is

a |00> + b |01> + c |10> + d |11>

where the coefficients have the same probability amplitude interpretation as for the single qubit state. Each state represents a possible outcome of a measurement of the two-qubit register. For example, |01> denotes a state in which the first qubit is in the state |0> and the second is in the state |1>.

Performing single qubit gates affects the relevant qubit in each of the superposition states. Similarly, two-qubit gates also act on all the relevant superposition states, but their outcome is far more interesting.

Consider a two-qubit register where the first qubit is in the superposed state (|0> + |1>)/sqrt(2) and the other is in the state |0>. This combined state can be written as:

(|0> + |1>)/sqrt(2) x |0> = (|00> + |10>)/sqrt(2),

where x denotes a tensor product (the mathematical mechanism for combining quantum states together).

The constant 1/sqrt(2) is called the normalisation factor and reflects the fact that the probabilities of measuring either a |0> or a |1> for the first qubit add up to one.

Let us now consider the two-qubit controlled-NOT, or CNOT, gate. The CNOT gate takes as input two qubits, a control and target, and applies the NOT gate to the target if the control qubit is set. The truth table looks like

IN	OUT
00	00
01	01
10	11
11	10

Table 1

Now, consider performing a CNOT gate on the state with the first qubit being the control. We apply a two-qubit gate on all the superposition states:

CNOT (|00> + |10>)/sqrt(2) -> (|00> + |11>)/sqrt(2).

What is so interesting about this two-qubit gate operation? The final state is *entangled*. There is no possible way of representing that quantum state as a product of two individual qubits; they are no longer independent. That is, it is not possible to describe the quantum state of either of the individual qubits in a way that is independent of the other qubit. Only the quantum state of the system that consists of both qubits provides a physically complete description of the two-qubit system. The states of the two individual qubits are now correlated beyond what is possible to achieve classically. Neither qubit is in a definite |0> or |1> state, but if we perform a measurement on either one, the outcome of the partner qubit will *always* yield the exact same outcome. The final state, whether it's |00> or |11>, is fundamentally random as before, but the states of the two qubits following a measurement will always be identical. One can think of this as flipping two coins, but the coins always both land on "heads" or both land on "tails" together. Something that we know is impossible classically.

Once a measurement is performed, the two qubits are once again independent. The final state is either |00> or |11> and both of these states can be trivially decomposed into a product of two individual qubits. The entanglement has been consumed and the entangled state must be prepared again.

3. Entanglement as the fundamental resource

Entanglement is the fundamental building block of quantum networks. Consider the state from the previous section:

(|00> + |11>)/sqrt(2).

Neither of the two qubits is in a definite |0> or |1> state and we need to know the state of the entire register to be able to fully describe the behaviour of the two qubits.

Entangled qubits have interesting non-local properties. Consider sending one of the qubits to another device. This device could in principle be anywhere: on the other side of the room, in a different country, or even on a different planet. Provided negligible noise has been introduced, the two qubits will forever remain in the entangled state until a measurement is performed. The physical distance does not matter at all for entanglement.

This lies at the heart of quantum networking, because it is possible to leverage the non-classical correlations provided by entanglement in order to design completely new types of application protocols that are not possible to achieve with just classical communication. Examples of such applications are quantum cryptography [Bennett14] [Ekert91], blind quantum computation [Fitzsimons17], or distributed quantum computation [Crepeau02].

Entanglement has two very special features from which one can derive some intuition about the types of applications enabled by a quantum network.

The first stems from the fact that entanglement enables stronger than classical correlations, leading to opportunities for tasks that require coordination. As a trivial example, consider the problem of consensus between two nodes who want to agree on the value of a single bit. They can use the quantum network to prepare the state (|00> + |11>)/sqrt(2) with each node holding one of the two qubits. Once either of the two nodes performs a measurement, the state of the two qubits collapses to either |00> or |11>, so whilst the outcome is random and does not exist before measurement, the two nodes will always measure the same value. We can also build the more general multi-qubit state (|00...> + |11...>)/sqrt(2) and perform the same algorithm between an arbitrary number of nodes. These stronger than classical correlations generalise to more complicated measurement schemes as well.

The second feature of entanglement is that it cannot be shared, in the sense that if two qubits are maximally entangled with each other, then it is physically impossible for these two qubits to also be entangled with a third qubit [Terhal04]. Hence, entanglement forms a sort of private and inherently untappable connection between two nodes once established.

Entanglement is created through local interactions between two qubits or as a product of the way the qubits were created (e.g. entangled photon pairs). To create a distributed entangled state, one can then physically send one of the qubits to a remote node. It is also possible to directly entangle qubits that are physically separated, but this still requires local interactions between some other qubits that the separated qubits are initially entangled with. Therefore, it is the transmission of qubits that draws the line between a genuine quantum network and a collection of quantum computers connected over a classical network.

A quantum network is defined as a collection of nodes that is able to exchange qubits and distribute entangled states amongst themselves. A quantum node that is able only to communicate classically with another quantum node is not a member of a quantum network.

More complex services and applications can be built on top of entangled states distributed by the network, see e.g. [ZOO]

4. Achieving quantum connectivity

This section explains the meaning of quantum connectivity and the necessary physical processes at an abstract level.

4.1. Challenges

A quantum network cannot be built by simply extrapolating all the classical models to their quantum analogues. Sending qubits over a wire like we send classical bits is simply not as easy to do. There are several technological as well as fundamental challenges that make classical approaches unsuitable in a quantum context.

4.1.1. The measurement problem

In classical computers and networks we can read out the bits stored in memory at any time. This is helpful for a variety of purposes such as copying, error detection and correction, and so on. This is not possible with qubits.

A measurement of a qubit's state will destroy its superposition and with it any entanglement it may have been part of. Once a qubit is being processed, it cannot be read out until a suitable point in the computation, determined by the protocol handling the qubit, has been reached. Therefore, we cannot use the same methods known from classical computing for the purposes of error detection and correction. Nevertheless, quantum error detection and correction schemes exist that take this problem into account and how a network chooses to manage errors will have an impact on its architecture.

4.1.2. No-cloning theorem

Since directly reading the state of a qubit is not possible, one could ask if we can simply copy a qubit without looking at it. Unfortunately, this is fundamentally not possible in quantum mechanics [Park70] [Wootters82].

The no-cloning theorem states that it is impossible to create an identical copy of an arbitrary, unknown quantum state. Therefore, it is also impossible to use the same mechanisms that worked for classical networks for signal amplification, retransmission, and so on as they all rely on the ability to copy the underlying data. Since any physical channel will always be lossy, connecting nodes within a quantum network is a challenging endeavour and its architecture must at its core address this very issue.

4.1.3. Fidelity

In general, it is expected that a classical packet arrives at its destination without any errors introduced by hardware noise along the way. This is verified at various levels through a variety of error detection and correction mechanisms. Since we cannot read or copy a quantum state, error detection and correction is more involved.

To describe the quality of a quantum state, a physical quantity called fidelity is used [NielsenChuang]. Fidelity takes a value between 0 and 1 -- higher is better, and less than 0.5 means the state is unusable. It measures how close a quantum state is to the state we have tried to create. It expresses the probability that the state will behave exactly the same as our desired state. Fidelity is an important property of a quantum system that allows us to quantify how much a particular state has been affected by noise from various sources (gate errors, channel losses, environment noise).

Interestingly, quantum applications do not need perfect fidelity to be able to execute -- as long as the fidelity is above some application-specific threshold, they will simply operate at lower rates. Therefore, rather than trying to ensure that we always deliver perfect states (a technologically challenging task) applications will specify a minimum threshold for the fidelity and the network will try its best to deliver it. A higher fidelity can be achieved by either having hardware produce states of better fidelity (sometimes one can sacrifice rate for higher fidelity) or by employing quantum error detection and correction mechanisms (see [Mural16] and [VanMeterBook] chapter 11).

4.1.4. Inadequacy of direct transmission

Conceptually, the most straightforward way to distribute an entangled state is to simply transmit one of the qubits directly to the other end across a series of nodes while performing sufficient forward quantum error correction (Section 4.4.3.2) to bring losses down to an acceptable level. Despite the no-cloning theorem and the inability to directly measure a quantum state, error-correcting mechanisms for quantum communication exist [Jiang09] [Fowler10] [Devitt13] [Mural16]. However, quantum error correction makes very high demands on both resources (physical qubits needed) and their initial fidelity. Implementation is very challenging and quantum error correction is not expected to be used until later generations of quantum networks are possible (see [Mural16] figure 2 and Section 4.4.3.3). Until then, quantum networks rely on entanglement swapping (Section 4.4.2) and teleportation (Section 4.3). This alternative relies on the observation that we do not need to be able to distribute any arbitrary entangled quantum state. We only need to be able to distribute any one of what are known as the Bell pair states [Briegel98].

4.2. Bell pairs

Bell pair states are the entangled two-qubit states:

|00> + |11>, |00> - |11>, |01> + |10>, |01> - |10>,

where the constant 1/sqrt(2) normalisation factor has been ignored for clarity. Any of the four Bell pair states above will do, as it is possible to transform any Bell pair into another Bell pair with local operations performed on only one of the qubits. When each qubit in a Bell pair is held by a separate node, either node can apply a series of single qubit gates to their qubit alone in order to transform the state between the different variants.

Distributing a Bell pair between two nodes is much easier than transmitting an arbitrary quantum state over a network. Since the state is known, handling errors becomes easier and small-scale error-correction (such as entanglement distillation discussed in a later section) combined with reattempts becomes a valid strategy.

The reason for using Bell pairs specifically as opposed to any other two-qubit state is that they are the maximally entangled two-qubit set of basis states. Maximal entanglement means that these states have the strongest non-classical correlations of all possible two-qubit states. Furthermore, since single-qubit local operations can never increase entanglement, less entangled states would impose some constraints on distributed quantum algorithms. This makes Bell pairs particularly useful as a generic building block for distributed quantum applications.

4.3. Teleportation

The observation that we only need to be able to distribute Bell pairs relies on the fact that this enables the distribution of any other arbitrary entangled state. This can be achieved via quantum state teleportation [Bennett93]. Quantum state teleportation consumes an unknown qubit state that we want to transmit and recreates it at the desired destination. This does not violate the no-cloning theorem as the original state is destroyed in the process.

To achieve this, an entangled pair needs to be distributed between the source and destination before teleportation commences. The source then entangles the transmission qubit with its end of the pair and performs a read out of the two qubits (the sum of these operations is called a Bell state measurement). This consumes the Bell pair's entanglement, turning the source and destination qubits into independent states. The measurements yields two classical bits which the source sends to the destination over a classical channel. Based on the value of the received two classical bits, the destination performs one of four possible corrections (called the Pauli corrections) on its end of the pair, which turns it into the unknown qubit state that we wanted to transmit. This requirement to communicate the measurement read out over a classical channel unfortunately means that entanglement cannot be used to transmit information faster than the speed of light.

The unknown quantum state that was transmitted was never fed into the network itself. Therefore, the network needs to only be able to reliably produce Bell pairs between any two nodes in the network. Thus, a key difference between a classical and quantum data planes is that a classical one carries user data, but a quantum data plane provides the resources for the user to transmit user data themselves without further involvement of the network.

4.4. The life cycle of entanglement

Reducing the problem of quantum connectivity to one of generating a Bell pair has facilitated the problem, but it has not solved it. In this section, we discuss how these entangled pairs are generated in the first place, and how their two qubits are delivered to the end-points.

4.4.1. Elementary link generation

In a quantum network, entanglement is always first generated locally (at a node or an auxiliary element) followed by a movement of one or both of the entangled qubits across the link through quantum channels. In this context, photons (particles of light) are the natural candidate for entanglement carriers, called flying qubits. The rationale for this choice is related to the advantages provided by photons such as moderate interaction with the environment leading to moderate decoherence, convenient control with standard optical components, and high-speed, low-loss transmissions. However, since photons are hard to store, a transducer must transfer the flying qubit's state to a qubit suitable for information processing and/or storage (often referred to as a matter qubit).

Since this process may fail, in order to generate and store entanglement efficiently, we must be able to distinguish successful attempts from failures. Entanglement generation schemes that are able to announce successful generation are called heralded entanglement generation schemes.

There exist three basic schemes for heralded entanglement generation on a link through coordinated action of the two nodes at the two ends of the link [Cacciapuoti19]:

* "At mid-point": in this scheme an entangled photon pair source sitting midway between the two nodes with matter qubits sends an entangled photon through a quantum channel to each of the nodes. There, transducers are invoked to transfer the entanglement from the flying qubits to the matter qubits. In this scheme, the transducers know if the transfers succeeded and are able to herald successful entanglement generation via a message exchange over the classical channel.

* "At source": in this scheme one of the two nodes sends a flying qubit that is entangled with one of its matter qubits. A transducer at the other end of the link will transfer the entanglement from the flying qubit to one of its matter qubits. Just like in the previous scheme, the transducer knows if its transfer succeeded and is able to herald successful entanglement generation with a classical message sent to the other node.

* "At both end-points": in this scheme both nodes send a flying qubit that is entangled with one of their matter qubits. A detector somewhere in between the nodes performs a joint measurement on the two qubits, which stochastically projects the remote matter qubits into an entangled quantum state. The detector knows if the entanglement succeeded and is able to herald successful entanglement generation by sending a message to each node over the classical channel.

The "mid-point source" scheme is more robust to photon loss, but in the other schemes the nodes retain greater control over the entangled pair generation.

Note that whilst photons travel in a particular direction through the quantum channel the resulting entangled pair of qubits does not have a direction associated with it. Physically, there is no upstream or downstream end of the pair.

4.4.2. Entanglement swapping

The problem with generating entangled pairs directly across a link is that efficiency decreases with channel length. Beyond a few 10s of kilometres in optical fibre or 1000 kilometres in free space (via satellite) the rate is effectively zero and due to the no-cloning theorem we cannot simply amplify the signal. The solution is entanglement swapping [Briegel98].

A Bell pair between any two nodes in the network can be constructed by combining the pairs generated along each individual link on a path between the two end-points. Each node along the path can consume the two pairs on the two links that it is connected to in order to produce a new entangled pair between the two remote ends. This process is known as entanglement swapping. Pictorially it can be represented as follows:

```
+---------+      +---------+      +---------+
|    A    |      |    B    |      |    C    |
|         |------|         |------|         |
|      X1~~~~~~~~~~X2   Y1~~~~~~~~~~Y2      |
+---------+      +---------+      +---------+
```

where X1 and X2 are the qubits of the entangled pair X and Y1 and Y2 are the qubits of entangled pair Y. The entanglement is denoted with ~~. In the diagram above, nodes A and B share the pair X and nodes B and C share the pair Y, but we want entanglement between A and C.

To achieve this goal, we simply teleport the qubit X2 using the pair Y. This requires node B to perform a Bell state measurement on the qubits X2 and Y1 which result in the destruction of the entanglement between Y1 and Y2. However, X2 is recreated in Y2's place, carrying with it its entanglement with X1. The end-result is shown below:

```
+---------+      +---------+      +---------+
|    A    |      |    B    |      |    C    |
|         |------|         |------|         |
|      X1~~~~~~~~~~~~~~~~~~~~~~~~~~~X2      |
+---------+      +---------+      +---------+
```

Depending on the needs of the network and/or application, a final Pauli correction at the recipient node may not be necessary since the result of this operation is also a Bell pair. However, the two classical bits that form the read out from the measurement at node B must still be communicated, because they carry information about which of the four Bell pairs was actually produced. If a correction is not performed, the recipient must be informed which Bell pair was received.

This process of teleporting Bell pairs using other entangled pairs is called entanglement swapping. Quantum nodes that create long-distance entangled pairs via entanglement swapping are called quantum repeaters in academic literature [Briegel98] and we will use the same terminology in this draft.

4.4.3. Error Management

4.4.3.1. Distillation

Neither the generation of Bell pairs nor the swapping operations are noiseless operations. Therefore, with each link and each swap the fidelity of the state degrades. However, it is possible to create higher fidelity Bell pair states from two or more lower fidelity pairs through a process called distillation (sometimes also referred to as purification) [Dur07].

To distil a quantum state, a second (and sometimes third) quantum state is used as a "test tool" to test a proposition about the first state, e.g., "the parity of the two qubits in the first state is even." When the test succeeds, confidence in the state is improved, and thus the fidelity is improved. The test tool states are destroyed in the process, so resource demands increase substantially when distillation is used. When the test fails, the tested state must also be discarded. Distillation makes low demands on fidelity and resources compared to quantum error correction, but distributed protocols incur round-trip delays due to classical communication [Bennett96].

4.4.3.2. Quantum Error Correction

Just like classical error correction, quantum error correction (QEC) encodes logical qubits using several physical (raw) qubits to protect them from errors described in Section 4.1.3 [Jiang09] [Fowler10] [Devitt13] [Mural16]. Furthermore, similarly to its classical counterpart, QEC can not only correct state errors but also account for lost qubits. Additionally, if all physical qubits which encode a logical qubit are located at the same node, the correction procedure can be executed locally, even if the logical qubit is entangled with remote qubits.

Although QEC was originally a scheme proposed to protect a qubit from noise, QEC can also be applied to entanglement distillation. Such QEC-applied distillation is cost-effective but requires a higher base fidelity.

4.4.3.3. Error management schemes

Quantum networks have been categorized into three "generations" based on the error management scheme they employ [Mural16]. Note that these "generations" are more like categories; they do not necessarily imply a time progression and do not obsolete each other, though the later generations do require more advanced technologies. Which generation is used depends on the hardware platform and network design choices.

Table 2 summarises the generations.

	First generation	Second generation	Third generation
Loss tolerance	Heralded entanglement generation (bi-directional classical signaling)	Heralded entanglement generation (bi-directional classical signaling)	Quantum Error Correction (no classical signaling)
Error tolerance	Entanglement distillation (bi-directional classical signaling)	Entanglement distillation (uni-directional classical signaling) or Quantum Error Correction (no classical signaling)	Quantum Error Correction (no classical signaling)

Table 2: Classical signaling and generations

Generations are defined by the directions of classical signalling required in their distributed protocols for loss tolerance and error tolerance. Classical signalling carries the classical bits and incurs round-trip delays described in Section 4.4.3.1, hence they affect the performance of quantum networks, especially as the distance between the communicating nodes increases.

Loss tolerance is about tolerating qubit transmission losses between nodes. Heralded entanglement generation, as described in Section 4.4.1, confirms the receipt of an entangled qubit using a heralding signal. A pair of directly connected quantum nodes repeatedly attempt to generate an entangled pair until the a heralding signal is received. As described in Section 4.4.3.2, QEC can be applied to complement lost qubits eliminating the need for re-attempts. Furthermore, since the correction procedure is composed of local operations, it does not require a heralding signal. However, it is possible only when the photon loss rate from transmission to measurement is less than 50%.

Error tolerance is about tolerating quantum state errors. Entanglement distillation is the easiest mechanism for improved error tolerance to implement, but it incurs round-trip delays due the requirement for bi-directional classical signalling. The alternative, QEC, is able to correct state errors locally so that it does not need any classical signalling between the quantum nodes. In between these two extremes, there is also QEC-applied distillation, which requires uni-directional classical signalling.

The three "generations" summarised:

1. First generation quantum networks use heralding for loss tolerance and entanglement distillation for error tolerance. These networks can be implemented even with a limited set of available quantum gates.

2. Second generation quantum networks improve upon the first generation with QEC codes for error tolerance (but not loss tolerance). At first, QEC will be applied to entanglement distillation only which requires uni-directional classical signalling. Later, QEC codes will be used to create logical Bell pairs which no longer require any classical signalling for the purposes of error tolerance. Heralding is still used to compensate for transmission losses.

3. Third generation quantum networks directly transmit QEC encoded qubits to adjacent nodes, as discussed in Section 4.1.4. Elementary link Bell pairs can now be created without heralding or any other classical signalling. Furthermore, this also enables direct transmission architectures in which qubits are forwarded end-to-end like classical packets rather than relying on Bell pairs and entanglement swapping.

Despite the fact that there are important distinctions in how errors will be managed in the different generations it is unlikely that all quantum networks will consistently use the same method. This is due to different hardware requirements of the different generations and the practical reality of network upgrades. Therefore, it is unavoidable that eventually boundaries between different error management schemes start forming. This will affect the content and semantics of messages that must cross those boundaries -- both for connection setup and real-time operation [Nagayama16].

4.4.4. Delivery

Eventually, the Bell pairs must be delivered to an application (or higher layer protocol) at the two end-nodes. A detailed list of such requirements is beyond the scope of this draft. At minimum, the end-nodes require information to map a particular Bell pair to the qubit in their local memory that is part of this entangled pair.

5. Architecture of a quantum internet

It is evident from the previous sections that the fundamental service provided by a quantum network significantly differs from that of a classical network. Therefore, it is not surprising that the architecture of a quantum internet will itself be very different from that of the classical Internet.

5.1. Challenges

This subsection covers the major fundamental challenges building quantum networks. Here, we only describe the fundamental differences. Technological limitations are described later.

1. Bell pairs are not equivalent to payload carrying packets.

   In most classical networks, including Ethernet, Internet Protocol (IP), and Multi-Protocol Label Switching (MPLS) networks, user data is grouped into packets. In addition to the user data, each packet also contains a series of headers which contain the control information that lets routers and switches forward it towards its destination. Packets are the fundamental unit in a classical network.

In a quantum network, the entangled pairs of qubits are the basic unit of networking. These qubits themselves do not carry any headers. Therefore, quantum networks will have to send all control information via separate classical channels which the repeaters will have to correlate with the qubits stored in their memory. Furthermore, a Bell pair consists of two qubits distributed across two nodes which is unlike a classical packet which is located at a single node. This has a fundamental impact on how quantum networks will be managed and how protocols need to be designed. To make long-distance Bell pairs, the nodes may have to keep their qubits in their quantum memories and wait until control information is exchanged before proceeding with the next operation. This signalling will result in additional latency which will depend on the distance between the nodes holding the two ends of the Bell pair. Error management, such as entanglement distillation, is a typical example of such control information exchange [Nagayama21] (see also Section 4.4.3.3).

2. "Store and forward" vs "store and swap" quantum networks.

As described in Section 4.4.1, quantum links provide Bell pairs that are undirected network resources, in contrast to directed frames of classical networks. This phenomenological distinction leads to architectural differences between quantum networks and classical networks. Quantum networks combine multiple elementary link Bell pairs together to create one end-to-end Bell pair, whereas classical networks deliver messages from one end to the other end hop by hop.

Classical networks receive data on one interface, store it in local buffers, then forward the data to another appropriate interface. Quantum networks store Bell pairs and then execute entanglement swapping instead of forwarding in the data plane. Such quantum networks are "store and swap" networks. In "store and swap" networks, we do not need to care about the order in which the Bell pairs were generated since they are undirected. However, whilst the ordering does not matter, it is very important that the right entangled pairs get swapped, and that the intermediate measurement outcomes (see Section 4.4.2) are signalled to and correlated with the correct qubits at the other nodes. Otherwise, the final end-to-end entangled pair will not be created between the expected end-points or will be in a different quantum state than expected. For example, rather than Alice receiving a qubit that is entangled with Bob’s qubit, her qubit is entangled with Charlie’s qubit. This distinction makes control algorithms and optimisation of quantum networks different from classical ones, in the sense that swapping is stateful in contrast to stateless packet-by-packet forwarding. Note that third generation quantum networks, as described in Section 4.4.1, will be able to support a "store and forward" architecture in addition to "store and swap".

3. An entangled pair is only useful if the locations of both qubits are known.

   A classical network packet logically exists only at one location at any point in time. If a packet is modified in some way, whether headers or payload, this information does not need to be conveyed to anybody else in the network. The packet can be simply forwarded as before.

   In contrast, entanglement is a phenomenon in which two or more qubits exist in a physically distributed state. Operations on one of the qubits change the mutual state of the pair. Since the owner of a particular qubit cannot just read out its state, it must coordinate all its actions with the owner of the pair's other qubit. Therefore, the owner of any qubit that is part of an entangled pair must know the location of its counterpart. Location, in this context, need not be the explicit spatial location. A relevant pair identifier, a means of communication between the pair owners, and an association between the pair ID and the individual qubits is sufficient.

4. Generating entanglement requires temporary state.

   Packet forwarding in a classical network is largely a stateless operation. When a packet is received, the router does a lookup in its forwarding table and sends the packet out of the appropriate output. There is no need to keep any memory of the packet any more.

   A quantum node must be able to make decisions about qubits that it receives and is holding in its memory. Since qubits do not carry headers, the receipt of an entangled pair conveys no control information based on which the repeater can make a decision. The relevant control information will arrive separately over a classical channel. This implies that a repeater must store temporary state as the control information and the qubit it pertains to will, in general, not arrive at the same time.

5.2. Classical communication

In this draft we have already covered two different roles that classical communication must perform:

* communicate classical bits of information as part of distributed protocols such as entanglement swapping and teleportation,

* communicate control information within a network, including both background protocols such as routing as well as signalling protocols to set up end-to-end entanglement generation.

Classical communication is a crucial building block of any quantum network. All nodes in a quantum network are assumed to have classical connectivity with each other (within typical administrative domain limits). Therefore, quantum nodes will need to manage two data planes in parallel, a classical one and a quantum one. Additionally, a node must be able to correlate information between the two planes so that the control information received on a classical channel can be applied to the qubits managed by the quantum data plane.

5.3. Abstract model of the network

5.3.1. The control and data planes

Control plane protocols for quantum networks will have many responsibilities similar to their classical counterparts, namely discovering the network topology, resource management, populating data plane tables, etc. Most of these protocols do not require the manipulation of quantum data and can operate simply by exchanging classical messages only. There may also be some control plane functionality that does require the handling of quantum data, e.g. a quantum ping [I-D.irtf-qirg-quantum-internet-use-cases]. As it is not clear if there is much benefit in defining a separate quantum control plane given the significant overlap in responsibilities with its classical counterpart, the question of whether there should be a separate quantum control plane is beyond the scope of this document.

However, the data plane separation is much more distinct and there will be two data planes: a classical data plane and a quantum data plane. The classical data plane processes and forwards classical packets. The quantum data plane processes and swaps entangled pairs. Third generation quantum networks may also forward qubits in addition to swapping Bell pairs.

In addition to control plane messages, there will also be control information messages that operate at the granularity of individual entangled pairs, such as heralding messages used for elementary link generation (Section 4.4.1). In terms of functionality, these messages are closer to classical packet headers than control plane messages and thus we consider them to be part of the quantum data plane. Therefore, a quantum data plane also includes the exchange of classical control information at the granularity of individual qubits and entangled pairs.

5.3.2. Elements of a quantum network

We have identified quantum repeaters as the core building block of a quantum network. However, a quantum repeater will have to do more than just entanglement swapping in a functional quantum network. Its key responsibilities will include:

1. Creating link-local entanglement between neighbouring nodes.

2. Extending entanglement from link-local pairs to long-range pairs through entanglement swapping.

3. Performing distillation to manage the fidelity of the produced pairs.

4. Participating in the management of the network (routing, etc.).

Not all quantum repeaters in the network will be the same; here we break them down further:

* Quantum routers (controllable quantum nodes) - A quantum router is a quantum repeater with a control plane that participates in the management of the network and will make decisions about which qubits to swap to generate the requested end-to-end pairs.

* Automated quantum nodes - An automated quantum node is a data plane only quantum repeater that does not participate in the network control plane. Since the no-cloning theorem precludes the use of amplification, long-range links will be established by chaining multiple such automated nodes together.

* End-nodes - End-nodes in a quantum network must be able to receive and handle an entangled pair, but they do not need to be able to perform an entanglement swap (and thus are not necessarily quantum repeaters). End-nodes are also not required to have any quantum memory as certain quantum applications can be realised by having the end-node measure its qubit as soon as it is received.

* Non-quantum nodes - Not all nodes in a quantum network need to have a quantum data plane. A non-quantum node is any device that can handle classical network traffic.

Additionally, we need to identify two kinds of links that will be used in a quantum network:

* Quantum links - A quantum link is a link which can be used to generate an entangled pair between two directly connected quantum repeaters. This may include additional mid-point elements described in Section 4.4.1. It may also include a dedicated classical channel that is to be used solely for the purpose of coordinating the entanglement generation on this quantum link.

* Classical links - A classical link is a link between any node in the network that is capable of carrying classical network traffic.

Note that passive elements, such as optical switches, do not destroy the quantum state. Therefore, it is possible to connect multiple quantum nodes with each other over an optical network and perform optical switching rather than routing via entanglement swapping at quantum routers. This does require coordination with the elementary link entanglement generation process and it still requires repeaters to overcome the short-distance limitations. However, this is a potentially feasible architecture for local area networks.

5.3.3. Putting it all together

A two-hop path in a generic quantum network can be represented as:

```
+-----+                                         +-----+
| App |- - - - - - - - - -CC- - - - - - - - - -| App |
+-----+                 +------+                +-----+
| EN  |------ CL ------|  QR  |------ CL ------| EN  |
|     |------ QL ------|      |------ QL ------|     |
+-----+                 +------+                +-----+
```

App - user-level application
EN - end-node
QL - quantum link
CL - classical link
CC - classical channel (traverses one or more CLs)
QR - quantum repeater

An application (App) running on two end-nodes (ENs) attached to a network will at some point need the network to generate entangled pairs for its use. This may require negotiation between the end-nodes (possibly ahead of time), because they must both open a communication end-point which the network can use to identify the two ends of the connection. The two end-nodes use a classical channel (CC) available in the network to achieve this goal.

When the network receives a request to generate end-to-end entangled pairs it uses the classical communication links (CLs) to coordinate and claim the resources necessary to fulfill this request. This may be some combination of prior control information (e.g. routing tables) and signalling protocols, but the details of how this is achieved are an active research question. A thought experiment on what this might look like be can be found later in this draft in Section 7.

During or after the distribution of control information, the network performs the necessary quantum operations such as generating entanglement over individual quantum links (QLs), performing entanglement swaps at quantum repeaters (QRs), and further signalling to transmit the swap outcomes and other control information. Since Bell pairs do not carry any user data, some of these operations can be performed before the request is received in anticipation of the demand.

Note that here, "signalling" is used in a very broad sense and covers many different types of messaging necessary for entanglement generation control. For example, heralded entanglement generation requires very precise timing synchronisation between the neighbouring nodes and thus the triggering of entanglement generation and heralding may happen over its own, perhaps physically separate CL, as was the case in network stack demonstration in [Pompili21.2]. Higher level signalling with less stringent timing requirements (e.g. control plane signalling) may then happen over its own CL.

The entangled pair is delivered to the application once it is ready, together with the relevant pair identifier. However, being ready does not necessarily mean that all link pairs and entanglement swaps are complete, as some applications can start executing on an incomplete pair. In this case the remaining entanglement swaps will propagate the actions across the network to the other end, sometimes necessitating fixup operations at the end node.

5.4. Physical constraints

The model above has effectively abstracted away the particulars of the hardware implementation. However, certain physical constraints need to be considered in order to build a practical network. Some of these are fundamental constraints and no matter how much the technology improves, they will always need to be addressed. Others are artifacts of the early stages of a new technology. Here, we consider a highly abstract scenario and refer to [Wehner18] for pointers to the physics literature.

5.4.1. Memory lifetimes

In addition to discrete operations being imperfect, storing a qubit in memory is also highly non-trivial. The main difficulty in achieving persistent storage is that it is extremely challenging to isolate a quantum system from the environment. The environment introduces an uncontrollable source of noise into the system which affects the fidelity of the state. This process is known as decoherence. Eventually, the state has to be discarded once its fidelity degrades too much.

The memory lifetime depends on the particular physical setup, but the highest achievable values in quantum network hardware currently are on the order of seconds [Abobeih18] although a lifetime of a minute has also been demonstrated for qubits not connected to a quantum network [Bradley19] (as of 2020). These values have increased tremendously over the lifetime of the different technologies and are bound to keep increasing. However, if quantum networks are to be realised in the near future, they need to be able to handle short memory lifetimes, for example by reducing latency on critical paths.

5.4.2. Rates

Entanglement generation on a link between two connected nodes is not a very efficient process and it requires many attempts to succeed [Hensen15] [Dahlberg19]. For example, the highest achievable rates of success between nitrogen-vacancy center nodes, which in addition to entanglement generation are also capable of storing and processing the resulting qubits, are on the order of 10 Hz. Combined with short memory lifetimes this leads to very tight timing windows to build up network-wide connectivity.

Other platforms have shown higher entanglement rates, but this usually comes at the cost of other hardware capabilities, such as no quantum memory and/or limited processing capabilities [Wei22]. Nevertheless, the current rates are not sufficient for practical applications beyond simple experimental proofs of concept. However, they are expected to improve over time as quantum network technology evolves [Wei22].

5.4.3. Communication qubits

Most physical architectures capable of storing qubits are only able to generate entanglement using only a subset of available qubits called communication qubits [Dahlberg19]. Once a Bell pair has been generated using a communication qubit, its state can be transferred into memory. This may impose additional limitations on the network. In particular, if a given node has only one communication qubit it cannot simultaneously generate Bell pairs over two links. It must generate entanglement over the links one at a time.

5.4.4. Homogeneity

Currently all existing quantum network implementations are homogeneous and they do not interface with each other. In general, it is very challenging to combine different quantum information processing technologies.

There are many different physical hardware platforms for implementing quantum networking hardware. The different technologies differ in how they store and manipulate qubits in memory and how they generate entanglement across a link with their neighbours. For example, hardware based on optical elements and atomic ensembles [Sangouard11] is very efficient at generating entanglement at high rates, but provides limited processing capabilities once the entanglement is generated. On the other hand, nitrogen-vacancy based [Hensen15] or trapped ion [Moehring07] platforms offer a much greater degree of control over the qubits, but have a harder time generating entanglement at high rates.

In order to overcome the weaknesses of the different platforms, coupling the different technologies will help to build fully functional networks. For example, end-nodes may be implemented using technology with good qubit processing capabilities to enable complex applications, but automated quantum nodes that that serve only to "repeat" along a linear chain, where the processing logic is much simpler, can be implemented with technologies that sacrifice processing capabilities for higher entanglement rates at long distances [Askarani21].

This point is further exacerbated by the fact that quantum computers (i.e. end-nodes in a quantum network) are often based on different hardware platforms than quantum repeaters thus requiring a coupling (transduction) between the two. This is especially true for quantum computers based on superconducting technology which are challenging to connect to optical networks. However, even trapped ion quantum computers, which is a platform that has shown promise for quantum networking, will still need to connect to other platforms that are better at creating entanglement at high rates over long distances (hundreds of kms).

6. Architectural principles

Given that the most practical way of realising quantum network connectivity is using Bell pair and entanglement swapping repeater technology, what sort of principles should guide us in assembling such networks such that they are functional, robust, efficient, and most importantly, do they work? Furthermore, how do we design networks so that they work under the constraints imposed by the hardware available today, but do not impose unnecessary burdens on future technology?

As quantum networking is a completely new technology that is likely to see many iterations over its lifetime, this draft must not serve as a definitive set of rules, but merely as a general set of recommended guidelines for the first generations of quantum networks based on principles and observations made by the community. The benefit of having a community built document at this early stage is that expertise in both quantum information and network architecture is needed in order to successfully build a quantum internet.

6.1. Goals of a quantum internet

When outlining any set of principles we must ask ourselves what goals do we want to achieve as inevitably trade-offs must be made. So what sort of goals should drive a quantum network architecture? The following list has been inspired by the history of computer networking and thus it is inevitably very similar to one that could be produced for the classical Internet [Clark88]. However, whilst the goals may be similar the challenges involved are often fundamentally different. The list will also most likely evolve with time and the needs of its users.

1. Support distributed quantum applications

   This goal seems trivially obvious, but makes a subtle, but important point which highlights a key difference between quantum and classical networks. Ultimately, quantum data transmission is not the goal of a quantum network - it is only one possible component of more advanced quantum application protocols [Wehner18]. Whilst transmission certainly could be used as a building block for all quantum applications, it is not the most basic one possible. For example, entanglement-based QKD, the most well known quantum application protocol, only relies on the stronger-than-classical correlations and inherent secrecy of entangled Bell pairs and does not have to transmit arbitrary quantum states [Ekert91].

The primary purpose of a quantum internet is to support distributed quantum application protocols and it is of utmost importance that they can run well and efficiently. Thus, it is important to develop performance metrics meaningful to application to drive the development of quantum network protocols. For example, the Bell pair generation rate is meaningless if one does not also consider their fidelity. It is generally much easier to generate pairs of lower fidelity, but quantum applications may have to make multiple re-attempts or even abort if the fidelity is too low. A review of the requirements for different known quantum applications can be found in [Wehner18] and an overview of use-cases can be found in [I-D.irtf-qirg-quantum-internet-use-cases].

2. Support tomorrow's distributed quantum applications

The only principle of the Internet that should survive indefinitely is the principle of constant change [RFC1958]. Technical change is continuous and the size and capabilities of the quantum internet will change by orders of magnitude. Therefore, it is an explicit goal that a quantum internet architecture be able to embrace this change. We have the benefit of having been witness to the evolution of the classical Internet over several decades and seen what worked and what did not. It is vital for a quantum internet to avoid the need for flag days (e.g. NCP to TCP/IP) or upgrades that take decades to roll out (e.g. IPv4 to IPv6).

Therefore, it is important that any proposed architecture for general purpose quantum repeater networks can integrate new devices and solutions as they become available. The architecture should not be constrained due to considerations for early-stage hardware and applications. For example, it is already possible to run QKD efficiently on metropolitan scales and such networks are already commercially available. However, they are not based on quantum repeaters and thus will not be able to easily transition to more sophisticated applications.

3. Support heterogeneity

   There are multiple proposals for realising practical quantum repeater hardware and they all have their advantages and disadvantages. Some may offer higher Bell pair generation rates on individual links at the cost of more difficult entanglement swap operations. Other platforms may be good all around, but are more difficult to build.

   In addition to physical boundaries, there may be distinctions in how errors are managed (Section 4.4.3.3). These difference will affect the content and semantics of messages that cross these boundaries -- both for connection setup and real-time operation.

   The optimal network configuration will likely leverage the advantages of multiple platforms to optimise the provided service. Therefore, it is an explicit goal to incorporate varied hardware and technology support from the beginning.

4. Ensure security at the network level

   The question of security in quantum networks is just as critical as it is in the classical Internet, especially since enhanced security offered by quantum entanglement is one of the key driving factors.

   Fortunately, from an application's point of view, as long as the underlying implementation corresponds to (or sufficiently approximates) theoretical models of quantum cryptography, quantum cryptographic protocols do not need the network to provide any guarantees about the confidentiality or integrity of the transmitted qubits or the generated entanglement (though they may impose requirements on the classical channel, e.g to be authenticated [Wang21]). Instead, applications will leverage the classical networks to establish the end-to-end security of the results obtained from the processing of entangled qubits. However, it is important to note that whilst classical networks are necessary to establish these end-to-end guarantees, the security relies on the properties of quantum entanglement. For example, QKD uses classical information reconciliation [Tang19] for error correction and privacy amplification [Elkouss11] for generating the final secure key, but the raw bits that are fed into these protocols must come from measuring entangled qubits [Ekert91]. In another application, secure delegated quantum computing, the client hides its computation from the server by sending qubits to the server and then requesting it (in a classical message) to measure them in an encoded basis. The client then decodes the results it receives from the server to obtain the result of the computation [Broadbent10]. Once again, whilst a classical network is used to achieve the goal of secure computation, the remote computation is strictly quantum.

Nevertheless, whilst applications can ensure their own end-to-end security, network protocols themselves should be security aware in order to protect the network itself and limit disruption. Whilst the applications remain secure they are not necessarily operational or as efficient in the presence of an attacker. For example, if an attacker can measure every qubit between two parties trying to establish a key using QKD, no secret key can be generated. Security concerns in quantum networks are described in more detail in [Satoh17] [Satoh20].

5. Make them easy to monitor

   In order to manage, evaluate the performance of, or debug a network it is necessary to have the ability to monitor the network while ensuring there will be mechanisms in place to protect the confidentiality and integrity of the devices connected to it. Quantum networks bring new challenges in this area so it should be a goal of a quantum network architecture to make this task easy.

   The fundamental unit of quantum information, the qubit, cannot be actively monitored as any readout irreversibly destroys its contents. One of the implications of this fact is that measuring an individual pair's fidelity is impossible. Fidelity is meaningful only as a statistical quantity which requires the constant monitoring and the sacrifice of generated Bell pairs for tomography or other methods.

   Furthermore, given one end of an entangled pair, it is impossible to tell where the other qubit is without any additional classical metadata. It is impossible to extract this information from the qubits themselves. This implies that tracking entangled pairs necessitates some exchange of classical information. This information might include (i) a reference to the entangled pair that allows distributed applications to coordinate actions on qubits of the same pair, and (ii) the two bits from each entanglement swap necessary to identify the final state of the Bell pair (Section 4.4.2).

6. Ensure availability and resilience

   Any practical and usable network, classical or quantum, must be able to continue to operate despite losses and failures, and be robust to malicious actors trying to disable connectivity. What differs in quantum networks as compared to classical networks in this regard is that we now have two data planes and two types of channels to worry about: a quantum and a classical one. Therefore, availability and resilience will most likely require a more advanced treatment than they do in classical networks.

Note that privacy, whilst related to security, is not listed as an explicit goal, because the privacy benefits will depend on the use case. For example, QKD only provides increased security for the distribution of symmetric keys [Bennett14] [Ekert91]. The handling, manipulation, sharing, encryption, and decryption of data will remain entirely classical limiting the benefits to privacy that can be gained from using a quantum network. On the other hand, there are applications like blind quantum computation which provides the user with the ability to execute a quantum computation on a remote server without the server knowing what the computation was or its input and output [Fitzsimons17]. Therefore, privacy must be considered on a per-application basis. An overview of quantum network use cases can be found in [I-D.irtf-qirg-quantum-internet-use-cases].

6.2. The principles of a quantum internet

The principles support the goals, but are not goals themselves. The goals define what we want to build and the principles provide a guideline in how we might achieve this. The goals will also be the foundation for defining any metric of success for a network architecture, whereas the principles in themselves do not distinguish between success and failure. For more information about design considerations for quantum networks see [VanMeter13.1] [Dahlberg19].

1. Entanglement is the fundamental service

   The key service that a quantum network provides is the distribution of entanglement between the nodes in a network. All distributed quantum applications are built on top of this key resource. Applications such as clustered quantum computing, distributed quantum computing, distributed quantum sensing networks, and certain kinds of quantum secure networks all consume quantum entanglement as a resource. Some applications (e.g. quantum key distribution) simply measure the entangled qubits to obtain a shared secret key [QKD]. Other applications (e.g. distributed quantum computing) build more complex abstractions and operations on the entangled qubits, e.g., distributed CNOT gates [DistCNOT] or teleportation of arbitrary qubit states [Teleportation].

A quantum network may also distribute multipartite entangled states (entangled states of three or more qubits) [Meignant19] which are useful for applications such as conference key agreement [Murta20], distributed quantum computing [Cirac99], secret sharing [Qin17], and clock synchronisation [Komar14]. Though it was worth noting that multipartite entangled states can also be constructed from multiple entangled pairs distributed between the end-nodes.

2. Bell Pairs are indistinguishable

   Any two Bell Pairs between the same two nodes are indistinguishable for the purposes of an application provided they both satisfy its required fidelity threshold. This observation is likely to be key in enabling a more optimal allocation of resources in a network, e.g. for the purposes of provisioning resources to meet application demand. However, the qubits that make up the pair themselves are not indistinguishable and the two nodes operating on a pair must coordinate to make sure they are operating on qubits that belong to the same Bell pair.

3. Fidelity is part of the service

   In addition to being able to deliver Bell pairs to the communication end-points, the Bell Pairs must be of sufficient fidelity. Unlike in classical networks where most errors are effectively eliminated before reaching the application, many quantum applications only need imperfect entanglement to function. However, quantum applications will generally have a threshold for Bell pair fidelity below which they are no longer able to operate. Different applications will have different requirements for what fidelity they can work with. It is the network's responsibility to balance the resource usage with respect to the applications' requirements. It may be that it is cheaper for the network to provide lower fidelity pairs that are just above the threshold required by the application than it is to guarantee high fidelity pairs to all applications regardless of their requirements.

4. Time is an expensive resource

   Time is not the only resource that is in short supply (memory, and communication qubits are as well), but ultimately it is the lifetime of quantum memories that imposes some of the most difficult conditions for operating an extended network of quantum nodes. Current hardware has low rates of Bell pair generation, short memory lifetimes, and access to a limited number of communication qubits. All these factors combined mean that even a short waiting queue at some node could be enough for a Bell pair to decohere or result in an end-to-end pair below an application's fidelity threshold. Therefore, managing the idle time of qubits holding live quantum states should be done carefully. Ideally by minimising the idle time, but potentially also by moving the quantum state for temporary storage to a quantum memory with a longer lifetime.

5. Be flexible with regards to capabilities and limitations

   This goal encompasses two important points. First, the architecture should be able to function under the physical constraints imposed by the current generation hardware. Near-future hardware will have low entanglement generation rates, quantum memories able to hold a handful of qubits at best, and decoherence rates that will render many generated pairs unusable.

   Second, the architecture should not make it difficult to run the network over any hardware that may come along in the future. The physical capabilities of repeaters will improve and redeploying a technology is extremely challenging.

7. A thought experiment inspired by classical networks

To conclude, we discuss a plausible quantum network architecture inspired by MPLS. This is not an architecture proposal, but rather a thought experiment to give the reader an idea of what components are necessary for a functional quantum network. We use classical MPLS as a basis as it is well known and understood in the networking community.

Creating end-to-end Bell pairs between remote end-points is a stateful distributed task that requires a lot of a-priori coordination. Therefore, a connection-oriented approach seems the most natural for quantum networks. In connection-oriented quantum networks, when two quantum application end-points wish to start creating end-to-end Bell pairs, they must first create a quantum virtual circuit (QVC). As an analogy, in MPLS networks end-points must establish a label switched path (LSP) before exchanging traffic. Connection-oriented quantum networks may also support virtual circuits with multiple end-points for creating multipartite entanglement. As an analogy, MPLS networks have the concept of multi-point LSPs for multicast.

When a quantum application creates a quantum virtual circuit, it can indicate quality of service (QoS) parameters such as the required capacity in end-to-end Bell pairs per second (BPPS) and the required fidelity of the Bell pairs. As an analogy, in MPLS networks applications specify the required bandwidth in bits per second (BPS) and other constraints when they create a new LSP.

Different applications will have different QoS requirements. For example, applications such as QKD, that don't need to process the entangled qubits and only need measure them and store the resulting outcome, may require a large volume of entanglement, but will be tolerant of delay and jitter for individual pairs. On the other hand, distributed/cloud quantum computing applications may need fewer entangled pairs, but instead, may need all of them to be generated in one go so that they can be processed all together before any of them decohere.

Quantum networks need a routing function to compute the optimal path (i.e. the best sequence of routers and links) for each new quantum virtual circuit. The routing function may be centralized or distributed. In the latter case, the quantum network needs a distributed routing protocol. As an analogy, classical networks use routing protocols such as open shortest path first (OSPF) and intermediate-system to intermediate system (IS-IS). However, note that the definition of "shortest-path"/"least-cost" may be different in a quantum network to account for its non-classical features, such as fidelity [VanMeter13.2].

Given the very scarce availability of resources in early quantum networks, a traffic engineering function is likely to be beneficial. Without traffic engineering, quantum virtual circuits always use the shortest path. In this case, the quantum network cannot guarantee that each quantum end-point will get its Bell pairs at the required rate or fidelity. This is analogous to "best effort" service in classical networks.

With traffic engineering, quantum virtual circuits choose a path that is guaranteed to have the requested resources (e.g. bandwidth in BPPS) available, taking into account the capacity of the routers and links and taking into account the resources already consumed by other virtual circuits. As an analogy, both OSPF and IS-IS have traffic engineering (TE) extensions to keep track of used and available resources, and can use constrained shortest path first (CSPF) to take resource availability and other constraints into account when computing the optimal path.

The use of traffic engineering implies the use of call admission control (CAC): the network denies any virtual circuits for which it cannot guarantee the requested quality of service a-priori. Or alternatively, the network pre-empts lower priority circuits to make room for the new one.

Quantum networks need a signaling function: once the path for a quantum virtual circuit has been computed, signaling is used to install the "forwarding rules" into the data plane of each quantum router on the path. The signaling may be distributed, analogous to the resource reservation protocol (RSVP) in MPLS. Or the signaling may be centralized, similar to OpenFlow.

Quantum networks need an abstraction of the hardware for specifying the forwarding rules. This allows us to de-couple the control plane (routing and signaling) from the data plane (actual creation of Bell pairs). The forwarding rules are specified using abstract building blocks such as "creating local Bell pairs", "swapping Bell pairs", "distillation of Bell pairs". As an analogy, classical networks use abstractions that are based on match conditions (e.g. looking up header fields in tables) and actions (e.g. modifying fields or forwarding a packet to a specific interface). The data-plane abstractions in quantum networks will be very different from those in classical networks due to the fundamental differences in technology and the stateful nature of quantum networks. In fact, choosing the right abstractions will be one of the biggest challenges when designing interoperable quantum network protocols.

In quantum networks, control plane traffic (routing and signaling messages) is exchanged over a classical channel, whereas data plane traffic (the actual Bell pair qubits) is exchanged over a separate quantum channel. This is in contrast to most classical networks, where control plane traffic and data plane traffic share the same channel and where a single packet contains both user fields and header fields. There is, however, a classical analogy to the way quantum networks work. Generalized MPLS (GMPLS) networks use separate channels for control plane traffic and data plane traffic. Furthermore, GMPLS networks support data planes where there is no such thing as data plane headers (e.g. DWDM or TDM networks).

8. Security Considerations

Security is listed as an explicit goal for the architecture and this issue is addressed in the section on goals. However, as this is an informational draft it does not propose any concrete mechanisms to achieve these goals.

9. IANA Considerations

This draft includes no request to IANA.

10. Acknowledgements

The authors want to thank Carlo Delle Donne, Matthew Skrzypczyk, Axel Dahlberg, Mathias van den Bossche, Patrick Gelard, Chonggang Wang, Scott Fluhrer, Joey Salazar, Joseph Touch, and the rest of the QIRG community as a whole for their very useful reviews and comments to the document.

11. Informative References

[Abobeih18] Abobeih, M.H., Cramer, J., Bakker, M.A., Kalb, N., Markham, M., Twitchen, D.J., and T.H. Taminiau, "One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment", Nature communications Vol. 9, Iss. 1, pp. 1-8, 2018, <https://arxiv.org/abs/1801.01196>.

[Aguado19] Aguado, A., Lopez, V., Diego, D., Peev, M., Poppe, A., Pastor, A., Folgueira, J., and M. Vicente, "The engineering of software-defined quantum key distribution networks", IEEE Communications Magazine Vol. 57, Iss. 7, pp. 20-26, 2019, <http://arxiv.org/abs/1907.00174>.

[Askarani21] Askarani, M.F., Chakraborty, K., and G.C. do Amaral, "Entanglement Distribution in Multi-Platform Buffered-Router-Assisted Frequency-Multiplexed Automated Repeater Chains", arXiv 2106.04671, 2021, <https://arxiv.org/abs/2106.04671>.

[Aspect81] Aspect, A., Grangier, P., and G. Roger, "Experimental tests of realistic local theories via Bell’s theorem", Physical Review Letters Vol. 47, Iss. 7, pp. 460-463, 1981, <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.47.460>.

[Bennett14] Bennett, C.H. and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing", Theoretical Computer Science Vol. 560 (Part 1), pp. 7-11, 2014, <https://arxiv.org/abs/2003.06557>.

[Bennett93] Bennett, C.H., Brassard, G., Crepeau, C., Jozsa, R., Peres, A., and W.K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels", Physical Review Letters Vol. 70, Iss. 13, pp. 1895-1899, 1993, <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.70.1895>.

[Bennett96] Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., and W.K. Wootters, "Mixed state entanglement and quantum error correction", Physical Review A Vol. 54, Iss. 5, pp. 3824-3851, 1996, <https://arxiv.org/abs/quant-ph/9604024>.

[Bradley19] Bradley, C.E., Randall, J., Abobeih, M.H., Berrevoets, R.C., Degen, M.J., Bakker, M.A., Markham, M., Twitchen, D.J., and T.H. Taminiau, "A 10-qubit solid-state spin register with quantum memory up to one minute", Physical Review X Vol. 9, Iss. 3, pp. 031045, 2019, <https://arxiv.org/abs/1905.02094>.

[Briegel98] Briegel, H.-J., Dur, W., Cirac, J.I., and P. Zoller, "Quantum repeaters: The role of imperfect local operations in quantum communication", Physical Review Letters Vol. 81, Iss. 26, pp. 5932-5935, 1998, <https://arxiv.org/abs/quant-ph/9803056>.

[Broadbent10] Broadbent, A., Fitzsimons, J., and E. Kashefi, "Measurement-Based and Universal Blind Quantum Computation", Springer-Verlag 978-3-642-13678-8, 2010, <https://link.springer.com/chapter/10.1007/978-3-642-13678-8_2>.

[Cacciapuoti19] Cacciapuoti, A.S., Caleffi, M., Van Meter, R., and L. Hanzo, "When Entanglement meets Classical Communications: Quantum Teleportation for the Quantum Internet", IEEE Transactions on Communications Vol. 68, Iss. 6, pp. 3808-3833, 2019, <https://arxiv.org/abs/1907.06197>.

[Cirac99] Cirac, J.I., Ekert, A.K., Huelga, S.F., and C. Macchiavello, "Distributed quantum computation over noisy channels", Physical Review A Vol. 59, Iss. 6, pp. 4249, <https://arxiv.org/abs/quant-ph/9803017>.

[Clark88] Clark, D., "The design philosophy of the DARPA internet protocols", Symposium proceedings on Communications architectures and protocols pp. 106-114, 1988, <https://dl.acm.org/doi/abs/10.1145/52324.52336>.

[Crepeau02]
Crepeau, C., Gottesman, D., and A. Smith, "Secure multi-party quantum computation", Proceedings of the thiry-fourth annual ACM symposium on Theory of computing pp. 643-652, 2002, <https://arxiv.org/abs/quant-ph/0206138>.

[Dahlberg19]
Dahlberg, A., Skrzypczyk, M., Coopmans, T., Wubben, L., Rozpedek, F., Pompili, M., Stolk, A., Pawelczak, P., Knegjens, R., de Oliveira Filho, J., Hanson, R., and S. Wehner, "A link layer protocol for quantum networks", Proceedings of the ACM Special Interest Group on Data Communication pp. 159-173, 2019, <https://arxiv.org/abs/1903.09778>.

[Devitt13] Devitt, S.J., Nemoto, K., and W.J. Munro, "Quantum error correction for beginners", Reports on Progress in Physics Vol. 76, Iss. 7, pp. 076001, 2013, <https://arxiv.org/abs/0905.2794>.

[DistCNOT] Quantum Network Explorer by QuTech, "Distributed CNOT", 2021, <https://www.quantum-network.com/applications/distributed-cnot/>.

[Dur07] Duer, W. and H.J. Briegel, "Entanglement purification and quantum error correction", Reports on Progress in Physics Vol. 70, Iss. 8, pp. 1381-1424, 2007, <https://arxiv.org/abs/0705.4165>.

[Ekert91] Ekert, A.K., "Quantum cryptography based on Bell's theorem", Physical Review Letters Vol. 67, Iss. 6, pp. 661-663, 1991, <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.67.661>.

[Elkouss11]
Elkouss, D., Martinez-Mateo, J., and V. Martin, "Information Reconciliation for Quantum Key Distribution", Quantum Information and Computation Vol. 11, No. 3 and 4, pp. 0226-0238, 2011, <https://arxiv.org/abs/1007.1616>.

[Elliott03]
Elliott, C., Pearson, D., and G. Troxel, "Quantum cryptography in practice", Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications pp. 227-238, 2003, <https://arxiv.org/abs/quant-ph/0307049>.

[Fitzsimons17]
Fitzsimons, J.F. and E. Kashefi, "Unconditionally verifiable blind quantum computation", Physical Review A Vol. 96, Iss. 1, pp. 012303, 2017, <https://arxiv.org/abs/1203.5217>.

[Fowler10] Fowler, A.G., Wang, D.S., Hill, C.D., Ladd, T.D., Van Meter, R., and L.C.L. Hollenberg, "Surface code quantum communication", Physical Review Letters Vol. 104, Iss. 18, pp. 180503, 2010, <https://arxiv.org/abs/0910.4074>.

[Giovanetti04]
Giovanetti, V., Lloyd, S., and L. Maccone, "Quantum-enhanced measurements: beating the standard quantum limit", Science Vol. 306, Iss. 5700, pp. 1330-1336, 2004, <https://arxiv.org/abs/quant-ph/0412078>.

[Gottesman12]
Gottesman, D., Jennewein, T., and S. Croke, "Longer-baseline telescopes using quantum repeaters", Physical Review Letters Vol. 109, Iss. 7, pp. 070503, 2012, <https://arxiv.org/abs/1107.2939>.

[Hensen15] Hensen, B., Bernien, H., Dreau, A.E., Reiserer, A., Kalb, N., Blok, M.S., Ruitenberg, J., Vermeulen, R.F.L., Schouten, R.N., Abellan, C., Amaya, W., Pruneri, V., Mitchell, M.W., Markham, M., Twitchen, D.J., Elkouss, D., Wehner, S., Taminiau, T.H., and R. Hanson, "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres", Nature Vol. 526, Iss. 7575, pp. 682-686, 2015, <https://arxiv.org/abs/1508.05949>.

[I-D.irtf-qirg-quantum-internet-use-cases]
Wang, C., Rahman, A., Li, R., Aelmans, M., and K. Chakraborty, "Application Scenarios for the Quantum Internet", Work in Progress, Internet-Draft, draft-irtf-qirg-quantum-internet-use-cases-13, 10 June 2022, <https://datatracker.ietf.org/api/v1/doc/document/draft-irtf-qirg-quantum-internet-use-cases/>.

[Jiang09] Jiang, L., Taylor, J.M., Nemoto, K., Munro, W.J., Van Meter, R., and M.D. Lukin, "Quantum repeater with encoding", Physical Review A Vol. 79, Iss. 3, pp. 032325, 2009, <https://arxiv.org/abs/0809.3629>.

[Joshi20] Joshi, S.K., Aktas, D., Wengerowsky, S., Loncaric, M., Neumann, S.P., Liu, B., Scheidl, T., Lorenzo, G.C., Samec, Z., Kling, L., Qiu, A., Razavi, M., Stipcevic, M., Rarity, J.G., and R. Ursin, "A trusted-node-free eight-user metropolitan quantum communication network", Science Advances Vol. 6, no.36, pp. eaba0959, 2020, <https://arxiv.org/abs/1907.08229>.

[Kimble08] Kimble, H.J., "The Quantum Internet", Nature Vol. 453, Iss. 7198, pp. 1023-1030, 2008, <http://arxiv.org/abs/0806.4195>.

[Komar14] Komar, P., Kessler, E.M., Bishof, M., Jiang, L., Sorensen, A.S., Ye, J., and M.D. Lukin, "A quantum network of clocks", Nature Physics Vol. 10, Iss. 8, pp. 582-587, 2014, <https://arxiv.org/abs/1310.6045>.

[Meignant19] Meignant, C., Markham, D., and F. Grosshans, "Distributing graph states over arbitrary quantum networks", Physical Review A Vol. 100, Iss. 5, pp. 052333, 2019, <https://arxiv.org/abs/1811.05445>.

[Moehring07] Moehring, D.L., Maunz, P., Olmschenk, S., Younge, K.C., Matsukevich, D.N., Duan, L.M., and C. Monroe, "Entanglement of single-atom quantum bits at a distance", Nature Iss. 449, pp. 68-71, 2007, <https://www.nature.com/articles/nature06118>.

[Mural16] Muralidharan, S., Li, L., Kim, J., Lutkenhaus, N., Lukin, M., and L. Jiang, "Optimal architectures for long distance quantum communication", Scientific Reports Vol. 6, Iss. 1, pp. 1-10, 2016, <https://www.nature.com/articles/srep20463>.

[Murta20] Murta, G., Grasselli, F., Kampermann, H., and D. Bruss, "Quantum conference key agreement: A review", Advanced Quantum Technologies Vol. 3, Iss. 11, pp. 2000025, 2020, <https://arxiv.org/abs/2003.10186>.

[Nagayama16] Nagayama, S., Choi, B.-S., Devitt, S., Suzuki, S., and R. Van Meter, "Interoperability in encoded quantum repeater networks", Physical Review A Vol. 93, Iss. 4, pp. 042338, 2016, <https://arxiv.org/abs/1508.04599>.

[Nagayama21] Nagayama, S., "Towards End-to-End Error Management for a Quantum Internet", arXiv 2112.07185, 2021, <https://arxiv.org/abs/2112.07185>.

[NielsenChuang]
Nielsen, M.A. and I.L. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press , 2011.

[Park70] Park, J.L., "The concept of transition in quantum mechanics", Foundations of Physics Vol. 1, Iss. 1, pp. 23-33, 1970, <https://link.springer.com/content/pdf/10.1007/BF00708652.pdf>.

[Peev09] Peev, M., Pacher, C., Alleaume, R., Barreiro, C., Bouda, J., Boxleitner, W., Debuisschert, T., Diamanti, E., Dianati, M., Dynes, J.F., Fasel, S., Fossier, S., Fuerst, M., Gautier, J.-D., Gay, O., Gisin, N., Grangier, P., Happe, A., Hasani, Y., Hentschel, M., Huebel, H., Humer, G., Laenger, T., Legre, M., Lieger, R., Lodewyck, J., Loruenser, T., Luetkenhaus, N., Marhold, A., Matyus, T., Maurhart, O., Monat, L., Nauerth, S., Page, J.-B., Poppe, A., Querasser, E., Ribordy, G., Robyr, S., Salvail, L., Sharpe, A.W., Shields, A.J., Stucki, D., Suda, M., Tamas, C., Themel, T., Thew, R.T., Thoma, Y., Treiber, A., Trinkler, P., Tualle-Brouri, R., Vannel, F., Walenta, N., Weier, H., Weinfurter, H., Wimberger, I., Yuan, Z.L., Zbinden, H., and A. Zeilinger, "The SECOQC quantum key distribution network in Vienna", New Journal of Physics Vol. 11, Iss. 7, pp. 075001, 2009, <http://stacks.iop.org/1367-2630/11/i=7/a=075001>.

[Pompili21.1]
Pompili, M., Hermans, S.L.N., Baier, S., Beukers, H.K.C., Humphreys, P.C., Schouten, R.N., Vermeulen, R.F.L., Tiggelman, M.J., dos Santos Martins, L., Dirkse, B., Wehner, S., and R. Hanson, "Realization of a multi-node quantum network of remote solid-state qubits", Science Vol. 372, Iss. 6539, pp. 259-264, 2021, <https://arxiv.org/abs/2102.04471>.

[Pompili21.2]
Pompili, M., Delle Donne, C., te Raa, I., van der Vecht, B., Skrzypczyk, M., Ferreira, G., de Kluijver, L., Stolk, A.J., Hermans, S.L.N., Pawelczak, P., Kozlowski, W., Hanson, R., and S. Wehner, "Experimental demonstration of entanglement delivery using a quantum network stack", arXiv 2111.11332, 2021, <https://arxiv.org/abs/2111.11332>.

[Qin17] Qin, H. and Y. Dai, "Dynamic quantum secret sharing by using d-dimensional GHZ state", Quantum information processing Vol. 16, Iss. 3, pp. 64, 2017, <https://link.springer.com/content/pdf/10.1007/s11128-017-1525-y.pdf>.

[QKD] Quantum Network Explorer by QuTech, "Quantum Key Distribution", 2021, <https://www.quantum-network.com/applications/qkd/>.

[RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <https://www.rfc-editor.org/info/rfc1958>.

[Sangouard11] Sangouard, N., Simon, C., de Riedmatten, H., and N. Gisin, "Quantum repeaters based on atomic ensembles and linear optics", Reviews of Modern Physics Vol. 83, Iss. 1, pp. 33-80, 2011, <https://arxiv.org/abs/0906.2699>.

[Satoh17] Satoh, T., Nagayama, S., and R. Van Meter, "The network impact of hijacking a quantum repeater", Quantum Science and Technology Vol. 3, Iss. 3, pp. 034008, 2017, <https://arxiv.org/abs/1701.04587>.

[Satoh20] Satoh, T., Nagayama, S., Suzuki, S., Matsuo, T., and R. Van Meter, "Attacking the quantum internet", arXiv 2005.04617, 2020, <https://arxiv.org/abs/2005.04617>.

[SutorBook] Sutor, R.S., "Dancing with Qubits", Packt Publishing , 2019.

[Tang19] Tang, B.-Y., Liu, B., Zhai, Y.-P., Wu, C.-Q., and W.-R. Yu, "High-speed and Large-scale Privacy Amplification Scheme for Quantum Key Distribution", Scientific Reports Vol. 9, Iss. 1, pp. 1-8, 2019, <https://www.nature.com/articles/s41598-019-50290-1>.

[Teleportation] Quantum Network Explorer by QuTech, "State teleportation", 2021, <https://www.quantum-network.com/applications/state-teleportation/>.

[Terhal04] Terhal, B.M., "Is entanglement monogamous?", IBM Journal of Research and Development Vol. 48, Iss. 1, pp. 71-78, 2004, <https://ieeexplore.ieee.org/document/5388928>.

[VanMeter13.1]
Van Meter, R. and J. Touch, "Designing quantum repeater networks", IEEE Communications Magazine Vol. 51, Iss. 8, pp. 64-71, 2013, <https://ieeexplore.ieee.org/document/6576340>.

[VanMeter13.2]
Van Meter, R., Satoh, T., Ladd, T.D., Munro, W.J., and K. Nemoto, "Path selection for quantum repeater networks", Networking Science Vol. 3, Iss. 1-4, pp. 82-95, 2013, <https://arxiv.org/abs/1206.5655>.

[VanMeterBook]
Van Meter, R., "Quantum Networking", ISTE Ltd/John Wiley and Sons Inc 978-1-84821-537-5, 2014.

[Wang21] Wang, L.-J., Zhang, K.-Y., Wang, J.-Y., Cheng, J., Yang, Y.-H., Tang, S.-B., Yan, D., Tang, Y.-L., Liu, Z., Yu, Y., Zhang, Q., and J.-W. Pan, "Experimental authentication of quantum key distribution with post-quantum cryptography", npj Quantum Information Vol. 7, no. 1, pp. 1-7, 2021, <https://www.nature.com/articles/s41534-021-00400-7>.

[Wehner18] Wehner, S., Elkouss, D., and R. Hanson, "Quantum internet: A vision for the road ahead", Science Vol. 362, Iss. 6412, 2018, <http://science.sciencemag.org/content/362/6412/eaam9288.full>.

[Wei22] Wei, S.-H., Jing, B., Zhang, X.-Y., Liao, J.-Y., Yuan, C.-Z., Fan, B.-Y., Lyu, C., Zhou, D.-L., Wang, Y., Deng, G.-W., Song, H.-Z., Oblak, D., Guo, G.-C., and Q. Zhou, "Towards real-world quantum networks: a review", arXiv 2201.04802, 2022, <https://arxiv.org/abs/2201.04802>.

[Wootters82]
Wootters, W.K. and W.H. Zurek, "A single quantum cannot be cloned", Nature Vol. 299, Iss. 5886, pp. 802-803, 1982, <https://www.nature.com/articles/299802a0>.

[ZOO] "The Quantum Protocol Zoo", <https://wiki.veriqloud.fr/>.

Authors' Addresses

Wojciech Kozlowski
QuTech
Building 22
Lorentzweg 1
2628 CJ Delft
Netherlands
Email: w.kozlowski@tudelft.nl

Stephanie Wehner
QuTech
Building 22
Lorentzweg 1
2628 CJ Delft
Netherlands
Email: s.d.c.wehner@tudelft.nl

Rodney Van Meter
Keio University
5322 Endo, Kanagawa
252-0882
Japan
Email: rdv@sfc.wide.ad.jp

Bruno Rijsman
Individual
Email: brunorijsman@gmail.com

Angela Sara Cacciapuoti
University of Naples Federico II
Department of Electrical Engineering and Information Technologies
Claudio 21
80125 Naples
Italy
Email: angelasara.cacciapuoti@unina.it

Marcello Caleffi
University of Naples Federico II
Department of Electrical Engineering and Information Technologies
Claudio 21
80125 Naples
Italy
Email: marcello.caleffi@unina.it

Shota Nagayama
Mercari, Inc.
Roppongi Hills Mori Tower 18F
6-10-1 Roppongi, Minato-ku,
106-6118
Japan
Email: shota.nagayama@mercari.com

QIRG K. Kompella
Internet-Draft M. Aelmans
Intended status: Standards Track Juniper Networks, Inc.
Expires: September 28, 2019 S. Wehner
QuTech
C. Sirbu
Redbit Networks
A. Dahlberg
QuTech
March 27, 2019

Advertising Entanglement Capabilities in Quantum Networks
draft-kaws-qirg-advent-03

Abstract

This document describes the use of link-state routing protocols on classical links in Quantum Networks. It contains proposals for additions to the IS-IS and OSPF protocols in order for them to transport relevant information for a Quantum Network, specifically, for the creation and manipulation of entangled pairs. The document will describe some of the necessary attributes and some suggestions of how this information may be used.

No Schrodinger's cats were harmed in the creation of this document.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [2].

Status of This Memo

This Internet-Draft will expire on September 28, 2019.

Table of Contents

1. Introduction

Quantum networking is an emerging field using the strange (even counterintuitive) properties of quantum mechanics to bring new, useful capabilities to computing and networking. One of these is "entanglement" [8], where the state of a group of particles must be described as a unit -- it cannot be decomposed to the state of each particle independently. Entangled pairs (often called EPR pairs, abbreviated here as EP) of particles can be used for quantum teleportation [10] and for quantum key distribution (QKD) [14].

A Quantum Network consists of quantum nodes and links. Here, we will be concerned with controllable quantum nodes (CQN) that allow control decisions. We posit a classical network parallel to the quantum network, with classical nodes (CN) and links. A classical node is colocated with a quantum node; a classical link may be a fiber or wavelength parallel to the corresponding quantum link. The existence of such a classical link is required by most quantum methods to create EPs deterministically or in a heralded fashion, where the creation of EPs is conditioned on a specific signal. To make useful decisions, it is desirable to augment this data to describe the capabilities and states of quantum nodes and links. At current time there is a need for classical links besides quantum links. In the future this might change into a situation where classical links will perhaps become obsolete.

This document proposes to carry entanglement capability data as Type Length Values (TLVs) over IS-IS or OSPF link-state advertisements over the corresponding classical network. A subset of the CQNs may run quantum applications such as QKD; these nodes may want to initiate multihop EPs.

Once an EP is created, the state of one particle ("quantum bit" or qubit) of an EP can be transferred to another qubit within the same QN by a process known as swapping or a SWAP gate ([12]). Also, several pairs of imperfectly entangled qubits can be "distilled" ([13]) to fewer but "better entangled" qubits.

Long distance entanglement can be produced from piecewise short distance entanglement: Given an EP between CQN A and CQN B, and another EP between CQN B and CQN C, one can create an EP between CQN A and CQN C by a process known as an "entanglement swap". These operations can be used to manipulate EPs to improve their lifetimes or their quality, or to create multihop EPs. Physically, qubits can be realized in many ways. For example, they can be represented by the energy levels of Nitrogen Vacancy (NV) Centers in diamond ([16], [17]). Logically, a qubit can be classified as a "communication qubit", a "traveling qubit" or a "storage qubit".

This document primarily discusses the exchange of quantum capabilities over a classical network. Some illustrative examples of how these capabilities can be used in a quantum network may be given, but this document should not be considered authoritative on these procedures.

1.1. Definitions and Notation

The following terms are used in this document:

Quantum link: A quantum link is a connection transporting traveling qubits, typically photons. This could be a physical link, or by means of teleportation over pre-established entanglement amongst distant network nodes. Such pre-shared entanglement effectively forms a shortcut - a virtual quantum link - which can be used exactly once. This document does not describe the usage of these links itself.

Classical link: A classical link is a connection transporting packets. This could be a physical or virtual link carried over a (MPLS) network. The proposed extensions in this document use these links to exchange capabilities.

Controllable Quantum Node (CQN): A controllable quantum node is a quantum device consisting of at least one qubit, capable of performing (a subset of) the following operations described in detail below: storing qubits for some amount of time, performing quantum operations such as entanglement distillation and entanglement swapping, and producing entanglement between the nodes and traveling qubits. The latter are generally realized using photons over fibers or through free space.

The term controllable refers to the fact that external control in software is capable of selecting the desired operations and qubits to use. Such nodes can be quantum repeaters that allow choices of operations to be made, as well as quantum end nodes capable of executing complex application protocols [14]. Quantum repeaters that merely allow timing control, such as automatic entanglement swapping whenever qubits arrive in a specific timing interval, will not be referred to as CQN. Such automated repeaters can be seen as lying at the quantum physical layer and do not enter routing or other decision making, apart from being switched on or off, and hence are not relevant to advertisement protocols like the ones considered here.

Quantum end node (QEN): In this document, a quantum end node [14] is one of a pair of quantum nodes forming a entanglement via a sequence of zero or more CQNs. Quantum end nodes typically run a higher-layer quantum application such as QKD.

Entangled Pair (EP): An entangled pair is a special state of two qubits, known as an EPR pair [8]. An entangled pair of qubits c@A and c@B is denoted [[c@A, c@B]].

The process of entangling two particles c@A and c@B is denoted as follows:

ent(c@A, c@B) -> [[c@A, c@B]]

ent(c@A, c@B) may take time T and succeed with probability P, and yield an entangled pair [[c@A, c@B]] of fidelity F.

Fidelity: A measure of the quality of the entanglement of an EP QFid [11]). Fidelity lies in the interval [0, 1] where a higher value indicates better quality; usable fidelity values lie in the half-open interval (0.5, 1].

Communication qubit: A qubit is called a communication qubit if it is possible to produce entanglement between this qubit and a traveling photon. This can be done by emission from the quantum node, that is, entanglement is produced between the qubit and the photon which is emitted from the quantum node. This process has been demonstrated in a number of physical systems that can be used as quantum nodes such as NV in diamond ([16], [17]), Ion Traps ([18]) and Neutral Atoms ([19]). An example of a communication qubit is the electron spin of the NV in diamond system ([15]). Entanglement between a communication qubit and traveling photons can also be produced by absorption. Examples include atomic ensemble memories ([20]).

A communication qubit c at CQN A is denoted by c@A, or simply c (if the node A is understood).

Storage qubit: A qubit is called a storage qubit if the node has the capability to use this qubit as a (temporary) quantum memory, but the qubit cannot serve as a communication qubit. To make storage qubits useful a node is required to possess the ability to transfer the state of a communication qubit to a storage qubit. An example of a storage qubit is the nuclear spin in the NV in diamond system [16].

A storage qubit s at node B is denoted s@B.

Swap: Two qubits located in the same CQN can interchange states ([13]). For example, the states of a communication qubit and a storage qubit at A can be swapped as follows:

swap(c@A, s@A)

If c@A was entangled with c@B, the result is that s@A is now entangled with c@B.

Distillation: Distillation is the process of turning a large number of weakly entangled states into a smaller number of highly entangled states ([13]).

For example, EPs [[c1@A, c1@B]] and [[c2@A, c2@B]] of fidelities F1 and F2 respectively may be distilled as follows:

```
dist([[c1@A, c1@B]], [[c2@A, c2@B]]) -> [[c3@A, c3@B]]
```

If distillation is successful, the fidelity F3 of [[c3@A, c3@B]] will be higher than F1 and F2.

Entanglement Swap: Given two EPs [[c@A, c1@B]] and [[c2@B, c@C]], one can perform an entanglement swap:

```
entSwap([[c@A, c1@B]], [[c2@B, c@C]]) -> [[c@A, c@C]]
```

to create a new EP between c@A and c@C. This is how "multihop" EPs are created from a sequence of "single-hop" EPs.

The swap operation can also be used within a CQN. A possible use case is when there aren't enough communication qubits to create the needed EPs. If, in the above example, B doesn't have two communication qubits c1 and c2, the following can be done:

```
ent(c@A, c@B)  -> [[c@A, c@B]]  # entangle
swap(c@B, s@B) -> [[c@A, s@B]]  # swap EP to storage qubit
ent(c@B, c@C)  -> [[c@B, c@C]]  # use freed up qubit c@B
entSwap(c@A, c@B) -> [[s@B, c@C]]       # create multihop EP
```

2. Motivation

Consider the following (very simple) quantum network consisting of QENs A and B, and CQNs X, Y, Z, U, V. The goal is to create an EP between qubits at A and at B, perhaps for the high-level task of QKD between A and B.

```
                 X - Y - Z
               .           .
             A               B                 A, B: QEN
               .           .
                  U --- V             X, Y, Z, U, V: CQNs
```

From A's point of view, here are a number of questions:

1. Is B reachable from A via quantum links that allow EP creation?

2. If so, along what sequence(s) of quantum nodes?

3. Can each pair of adjacent CQNs in this sequence form EPs? If so, how long will it take, and what fidelity can be expected?

4. If each pair of adjacent CQNs successfully forms EPs of sufficient fidelity, can these be swapped to form a multihop EP between A and B?

5. If a multihop EP between A and B were to be formed, would it be of good enough fidelity, or should a second multihop EP be formed and the two EPs distilled into one high fidelity EP? How many times should this process be repeated?

6. If the overall answer is Yes, should A proceed via sequence A, X, Y, Z, B, or sequence A, U, V, B?

This document aims to provide all CQNs in a quantum network with the information they need to answer such questions, and to create EPs at their desired fidelity and speed.

3. Theory of Operation

A CQN contains one or more communication qubits and one or more storage qubits. Many proposals exist for producing EPs between remote quantum nodes (see for example [16], [17], [18], [20]). Abstractly, these result in the generation of EPs with fidelity F after an expected time t. To give an example, we describe the generation of EPs that has been implemented in NV in diamond ([16]), and Ion Traps ([18]). The largest distance for producing long-lived entanglement is presently 1.3kms ([17]). To entangle a pair of communication qubits, the QNs send carefully timed photons towards the HS. If the process is successful, HS sends an OK message to both QNs.

```
+----------+                                        +----------+
|          |  c-chan    +------------+   c-chan     |          |
| Control- | <-------> | Heralding  |  <-------->  | Control- |
| lable    |            |  Station   |              |   lable  |
| Quantum  | *~~~~~~~>  |            |  <~~~~~~~~*  | Quantum  |
| Node     |  q-chan    +------------+   q-chan     |     Node |
| A        |                                        |        B |
|          | <------------------------------------> |          |
+----------+    classical network control plane     +----------+
```

The classical network control plane is of particular interest here as it would be used by the proposed protocol to advertise and exchange information about the capabilities of the CQNs to generate entanglement. This classical channel exists between all CQNs and is shared with other application specific control and data plane traffic.

3.1. Multihop Entanglement

```
                    resulting entanglement
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
      +-+                            +-+                            +-+
      |A+----------------------------+B+----------------------------+C|
      +-+   A-B Link properties      +-+                            +-+
                [(F1,t1), (F2,t2)]    Node B properties:
                                      - Number of Communication Qubits
                                      - Number of Storage Qubits
                                      Node capabilities (operations):
                                      - Swap comm <-> storage
                                      - Entanglement swap
                                      - [(Distillation scheme, time)...]
```

In the figure above, an example request for an entangled pair between nodes A and B will be affected by the following properties:

o A chosen combination of F(idelity) and t(ime) duration to produce an entanglement at the respective Fidelity. These parameters roughly equate to the quality of the link, the accuracy with which the nodes can use the link, and the delay in classical networking.

o The actual capability of nodes A and B to make use of the communication qubits.

A new EP creation between CQNs B and C will similarly be affected by the same parameters as above.

```
                                           resulting entanglement
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~* *~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
      +-+                            +-+                            +-+
      |A+----------------------------+B+----------------------------+C|
      +-+                            +-+   B-C Link properties      +-+
                                               [(F1,t1), (F2,t2)]
```

And finally, with an entanglement swap operation at node B (which is a node specific capability and has a specific duration) we end up with an A-C EP:

```
        *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
      +-+                            +-+                            +-+
      |A+----------------------------+B+----------------------------+C|
      +-+                            +-+                            +-+
                       Node B entanglement swap operation
```

3.2. Distillation

If a pair of CQNs A and B share a number of EPs of insufficient quality, they may be combined into a single EP of higher quality by distillation. To do so, these CQNs need to agree on which distillation scheme to use before distillation can proceed. This does not necessarily need to be via communication between A and B, if one agrees upon a deterministic procedure of selecting one. This document suggests the following procedure:

1. A and B look at the distillation schemes that both advertise in common.

2. If there is none in common, stop. Distillation is not possible.

3. If there is a non-trivial subset in common, the first scheme in the node with the lower router ID is to be used by A and B.

Given a chosen distillation scheme (S,t,p), an additional time delay will be added for the actual operation: For a 2:1 distillation scheme between nodes A and B, 2 EPs need to be produced followed by an operation on A and B that produces 1 EP. This operation will take time some expected time t, and succeed with probability p.

```
     2:1 distillation (S,t,p)
  *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
  *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
+-+                            +-+                            +-+
|A+----------------------------+B+----------------------------+C|
+-+   A-B Link properties      +-+                            +-+
          [(F1,t1), (F2,t2)]
```

3.3. Node Properties

We are interested in exposing the properties of CQNs (including QENs) to allow sophisticated decision making, for example in the creation of entanglement. These properties include:

1. Number of communication qubits. The number of communication qubits determines the number of entangled pairs that the node can produce simultaneously.

2. Number of storage qubits

3. Possible operations, along with their execution time and probability of success:

    1. Swap between communication and storage qubits

2. Entanglement swap

3. List of supported distillation schemes (in order of preference).

Note that several other parameters can be advertised, such as the T1 and T2 times for a qubit's decoherence. These are omitted for now, instead just giving the decay of the fidelity of an EP. If deemed useful, T1 and T2 times can additionally be advertised.

3.4. Link Properties

A list of (Fn,tn) pairs describing the tradeoffs of a possible entanglement produced by two nodes (the ends of said link): tn is the time to produce an entangled pair with fidelity Fn.

4. The (Ab)use of Protocols

The routing protocols IS-IS and/or OSPF could be used in order to advertise entanglement capabilities. This section describes the additional data fields needed in order to facilitate the objective.

4.1. A Brief Primer on Link-state Protocols

This document suggests the use of a link-state protocol to distribute the capabilities of CQNs to create entanglement. This section offers a short introduction to link-state protocols for those not familiar with them.

Consider a directed graph G=(V, E) with vertices (nodes) V and edges (links) E. Consider also G'=(V', E'); there is a 1-1 mapping from V' to V and from E' to E such that e1' = (v1', v2') is in E' iff e1 = (v1, v2) is in E and v1' maps to v1 and v2' maps to v2. G' represents the quantum network; V' represents the set of CQNs, and E' the set of quantum links between pairs of CQNs; G represents a classical network parallel to G'; that is, each CQN v' has a corresponding classical node v. v plays a dual role: it is the control node for v', and proxies on behalf of v' in the link-state protocol.

The basic objective of a link-state protocol is to "flood" properties of nodes and (directed) links to all nodes in the network. This is accomplished by means of "link-state advertisements" (LSAs) that each node originates and sends to its immediate neighbors. The neighbors in turn send received LSAs to their own neighbors; this process repeats until every node receives every LSA (hence the term "flooding"). The focus of LSAs is the link properties (hence _link-state_ advertisements), although node properties are also advertised.

There are mechanisms to prevent looping of LSAs, and for reliable flooding. There is also a sequence number by which a more recent update of an LSA can be identified as such, and a mechanism for "aging out" LSAs belonging to nodes no longer in the network. In what follows, quantum node and link properties are added to the link-state advertisements of the corresponding classical node. Note that link properties need not be symmetric; that is, the link properties of (v, w) need not be the same as those of (w, v).

The net result of flooding is that every node has the same picture of the network (modulo LSAs in flight); in particular, each node knows the overall topology and connectivity of the network, and can use this information to make decisions. In a classical network, such a decision could be to compute a shortest path; for the quantum network, it could be choosing a feasible path (i.e., sequence of CQNs) for a multihop entanglement. Note that a node doesn’t really know when it has complete and up-to-date information about the network; LSA updates may be originated at any time. Usually, this is okay: for example, if a node v learns enough of the network to have a path to another node w, it can compute a multihop entanglement to w. Subsequent updates may provide a more optimal (or higher probability) entanglement path. There are heuristics one can apply to guess that the link-state database (LSDB) (i.e., the union of all LSAs) is complete-ish; however, as nodes (and links) can fail or disconnect, there really is no such thing as "the full LSDB".

Each node v is identified by a "router ID" (an IP address uniquely allocated to v), denoted by rid(v). A link L = (v, w) is identified by (rid(v), i) where i is an index allocated by v for L unique for each link emanating from v. (L may also be identified by IP addresses, but we’ll ignore that for now.) It is generally expected that a directed link (v, w) is matched by a link (w, v); if not, (v, w) is ignored from subsequent consideration; in particular, no link properties are advertised for this link by v. Note that a pair of nodes may have multiple links between them; for simplicity, the notation will not be extended to indicate this. We’ll assume rid(v’) = rid(v) and the index allocated to a quantum link e’ is the same as that of the corresponding classical link e.

Let v, w be a pair of neighboring nodes, and let L1 = (v, w) and L2 = (w, v) in E be directed links in opposite directions between v and w with identifiers (rid(v), i1) and (rid(w), i2) respectively (where i1 is the index allocated for L1 by v, and similarly for i2)). As a first step in running a link-state protocol, v runs a "hello protocol" all its links; in particular, over L1. Similarly, w will run the hello protocol over L2. The hello protocol serves to exchange the indices i1 and i2, and thus identify (rid(v), i1) as the reverse link of (rid(w), i2). This allows both v and w to correlate the link properties of L1 and L2. If the hello protocol fails between v and w, neither node includes link properties for the link in their LSAs.

Once the hello protocol has been run on all links, v starts the process of generating and sending its own LSA over all its links, and of receiving the current LSDB from its neighbors. Note that an LSA originated by v must propagate unchanged across the network; only v is allowed to change it (and such a change must be accompanied by updating the LSA's sequence number). Such an update is triggered by a new link coming up, an existing link going down, or a node or link property changing.

IS-IS and OSPF are in principle similar, although the details of the protocol mechanisms and encodings vary. In both protocols, a Type-Length-Value (TLV) is used to encode most node and link properties. In IS-IS, TLVs are used for all properties, and a single type of LSA is used; in OSPF, there are several types of LSAs, and many (but not all) properties are encoded as TLVs.

[1] has examples of "standard" LSAs for routing; [4] has the so-called Traffic Engineering LSAs.

4.2. Node Properties

Here, we give a protocol-independent description of quantum node properties; later documents will specify the encoding specifically for IS-IS and OSPF.

Note that the following list of node properties is a strawman; all details are subject to change, and other properties may be added as needed.

The following node properties are added to the appropriate LSA:

```
<Qubit-TLV><NCQ><NSQ>
<CS-Swap><Prob><ExecTime>
<Ent-Swap><Prob><ExecTime>
<Measure><Prob><ExecTime>
<NDistSch><DistScheme1><DistScheme2>
```

4.3. Link Properties

Only one link property is listed. It gives the time-fidelity tradeoffs of an entanglement operation as a list:

```
<N-Ent-TO><time1><fid1><time2><fid2>...
```

This is interpreted as follows: an entanglement operation may be initiated between nodes v and w over link (v, w). Depending on how fast one wants to complete (time-i), the list gives the corresponding fidelity of the resulting entanglement (fid-i). time-i is given in nanoseconds; fid-i as a number between 0 and 999999. THe denominator is 1000000.

Note that this link property is symmetric, as entanglement is initiated simultaneously at v and w.

5. Security Considerations

It is not anticipated that adding these extensions to IS-IS and OSPF will present new security hazards to those protocols. Since however a common application of entangled pairs is for security purposes (such as QKD), it is worth investigating whether this application places a higher burden of security on the underlying protocols.

6. Acknowledgments

The authors would like to thank the following people for their contributions and support to this document: Vesna Manojlovic, Marcello Caleffi and Rodney Van Meter. Kompella would also like to thank Bruno Rijsman for encouraging him to learn about Quantum Computing and Networking.

Also:

```
       _,'|             _.-''``-...___..--';)
       /_ \'.      __..-' ,      ,--...--'''
      <\    .`--'''       `     /'
       `-';'               ;   ; ;
 __...--''     ___...--_..'  .;.'
(,__....----'''       (,..--''
```

7. Editorial Comments

It could be worth investigating the use of a distance-vector routing protocol to limit flooding.

8. IANA Considerations

There are no requests as yet to IANA for this document.

9. References

9.1. Normative References

[1] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[3] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[4] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[5] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <https://www.rfc-editor.org/info/rfc5329>.

[6] Aggarwal, R. and K. Kompella, "Advertising a Router's Local Addresses in OSPF Traffic Engineering (TE) Extensions", RFC 5786, DOI 10.17487/RFC5786, March 2010, <https://www.rfc-editor.org/info/rfc5786>.

9.2. Informative References

[7] "Qubit", <https://en.wikipedia.org/wiki/Qubit>.

[8] "Quantum Entanglement", <https://en.wikipedia.org/wiki/Quantum_entanglement>.

[9] "Quantum Network", <https://en.wikipedia.org/wiki/Quantum_network>.

[10] "Quantum Teleportation", <https://en.wikipedia.org/wiki/Quantum_teleportation>.

[11] "Quantum Fidelity", <https://en.wikipedia.org/wiki/Fidelity_of_quantum_states>.

[12] Wehner, S., Elkouss, D., and R. Hanson, "Quantum internet: A vision for the road ahead", Science 362, 6412, 2018.

[13] Rozpedek, F., Schiet, T., Thinh, L., Elkouss, D., Doherty, A., and S. Wehner, "Optimizing practical entanglement distillation", Phys. Rev. A 97, 062333, 2018, <https://arxiv.org/abs/1803.10111>.

[14] "Quantum Key Distribution", <https://en.wikipedia.org/wiki/Quantum_key_distribution>.

[15] "Nitrogen-vacancy center", <https://en.wikipedia.org/wiki/Nitrogen-vacancy_center>.

[16] Humphreys, P., "Deterministic delivery of remote entanglement on a quantum network", Nature 558, 2018.

[17] Hensen, B. and , "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres", Nature 526, 2015.

[18] Hucul, D. and , "Modular entanglement of atomic qubits using photons and phonons", Nature Physics 11(1), 2015.

[19] Noelleke, C. and , "Efficient Teleportation Between Remote Single-Atom Quantum Memories", Physical Review Letters 110, 140403, 2013.

[20] Sangouard, N. and , "Quantum repeaters based on atomic ensembles and linear optics", Reviews of Modern Physics 83, 33, 2011.

Authors' Addresses

Kireeti Kompella
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: kireeti.kompella@gmail.com

Melchior Aelmans
Juniper Networks, Inc.
Boeing Avenue 240
Schipol-Rijk, PZ 1119
The Netherlands

Email: maelmans@juniper.net

Stephanie Wehner
QuTech
Van der Waalsweg 122
Delft, LC 2611
The Netherlands

Email: s.d.c.wehner@tudelft.nl

Cristian Sirbu
Redbit Networks
Dublin
Republic of Ireland

Email: cristian@redbit.network

Axel Dahlberg
QuTech
Van der Waalsweg 122
Delft, LC 2611
The Netherlands

Email: E.A.Dahlberg@tudelft.nl

Quantum Internet Research Group
Internet-Draft
Intended status: Informational
Expires: March 13, 2020

R. Van Meter
T. Matsuo
Keio University
September 10, 2019

Connection Setup in a Quantum Network
draft-van-meter-qirg-quantum-connection-setup-01

Abstract

Near-term quantum networks will grow to form a Noisy, Intermediate-Scale Quantum Internet (NISQI). Connection setup will require adapting behavior along the path to the noise levels of individual elements. In this proposal, path creation is triggered by an application at the Initiator, information is accumulated node-by-node on an outbound pass in a series of QCap (quantum capability) blocks, then the RuleSets are created at the Responder. RuleSets are installed at the individual nodes on the return pass. This document describes the architecture of connection setup in a network. Details of the RuleSets and QCaps, addressing architecture, link protocols, routing, resource allocation (multiplexing), extension of this setup procedure to an internetwork, and extension to multiparty communications are beyond the scope of this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Building a connection across a quantum network [theqi] is a classical task. Because of the low success probability of quantum communication due to photon loss and the extremely high error rates due to the fragile nature of quantum information, quantum communication between two nodes more closely resembles a coordinated computation distributed among the set of nodes forming the path between the two nodes than a store-and-forward network session [qnetworking].

Use of the quantum network is driven by applications running at two (or more) classical nodes. Overall behavior is similar to client-server computing. The connection is initiated from a node similar to client and responded to by a node similar to a server. The details of the sending and receiving of the classical messages are not specified in this document, but can be modeled as if being sent over a TCP socket. Messages are assumed to be reliable and delivered in order. These messages have no hard real time requirement, though the subsequent data phase of the operation may.

This connection setup process must collect information about the hardware (channels and buffer memories) to be used, because of the heterogeneity of the underlying hardware. Loss in optical channels naturally varies with channel length and other factors, and has a large impact on quantum communication performance. Individual quantum buffers holding quantum bits (qubits) will vary in quality, as well.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Concepts and Glossary

The following terms will be used:

Bell pair a common form of entangled quantum state useful in communications.

End node a quantum network node with a single interface. End nodes may have stationary quantum memories, or may be capable only of measuring photons; this distinction is beyond the scope of this document.

Entanglement the condition of a group of qubits (typically two qubits in this document) in a shared state that cannot be described using only real, non-negative, classical probabilities.

Entanglement Swapping executed at node B splices an entangled state shared with node A to an entangled state shared with node C, creating A-C entanglement and disentangling B from both nodes.

Fidelity a measure of the quality of a quantum state; roughly, the probability that the system holds the desired state.

Initiator the initiator of the classical process of establishing the connection by sending a message toward the Responder.

Purification an error detection mechanism on quantum states. Typically, one quantum state is used to test the condition of a second state; the first state is destroyed in the process. If the purification fails, it is unknown whether the first or second state was in error, and the second state is discarded as well. If purification succeeds, our confidence in the state is improved.

QCap an information block describing the quantum capabilities of a particular node and link.

Qubit a quantum system with two states that can be stored in memory or transmitted through a channel, manipulated in a constrained set of operations, entangled with other qubits, and measured.

Repeater a quantum network node with two interfaces, typically sitting in the middle of a chain. Repeaters do not require routing functionality, but otherwise have the same capabilities as routers. As spacing between nodes may be required to be as short as ten kilometers, depending on technology, what would be single fiber hops in a classical network will be a long chain of repeaters.

Responder is the classical endpoint of the connection setup process, where the message sent by the Initiator terminates. The Responder creates the RuleSets for all nodes in the path, and commonly will be the smarter node.

Router a quantum network node with a more than two interfaces, requiring routing capability.

RuleSet describes the actions that a nodes should take when certain conditions occur. The contents of RuleSets are beyond the scope of this document.

The terms "source" and "destination" are not appropriate at the connection level in a quantum network, because distributed quantum states are not necessarily used for the unidirectional transfer of information. Therefore, we use Initiator and Responder to designate roles in the connection setup process, but those roles do not not necessarily correspond to any asymmetry during the connection lifetime. Source and destination are not appropriate because:

1. There may not even be data transferred between nodes; the entanglement might be used for some shared operation that doesn't involve qubits moving back and forth via teleportation. Quantum key distribution (QKD) is an obvious example, where both ends measure the entangled state and destroy it in order to get a classical bit.

2. Temporally, operations may not even happen left-to-right along the chain of repeaters, again violating the notion that data is moving.

"Source" and "destination" may be used to describe the movement of an individual classical message.

Links are assumed to be point-to-point. Multidrop physical layers are possible, but quantum broadcast or multicast are not directly possible at the physical level, and would have to be emulated.

3. Connection Setup Phases

3.1. Short Description of Phases

The single-network, two-node connection setup procedure consists of three basic phases:

1. The outbound request is routed from Initiator to Responder using a standard NextHop-based forwarding table, accumulating information about the path along the way in a stack of QCaps.

2. When the request arrives at the Responder, the Responder uses that information to create a complete RuleSet for every node. The RuleSets are assembled into a stack with the nearest node at the top.

3. The RuleSets are sent back along the original path, with each node removing its RuleSet from the message (popping the stack), then forwarding the remaining QCaps on until it returns to the Initiator.

3.2. Rationale for this Architecture

The outbound pass collects information about the nodes and links, to be used by the Responder to formulate the RuleSets. Why is the information collected in this fashion rather than shared more broadly across the network, e.g. as part of a modified routing protocol such as OSPF [RFC2328]? Why does a single node create the RuleSets for all nodes, rather than allowing individual nodes to create their own RuleSets when they see the PathSetupRequest message?

1. Because Repeaters may be spaced as closely as every 10km, a full topology for a network listing every Repeater may be excessively large for routing purposes, but such information is needed for building RuleSets.

2. The information collected may be substantially larger in volume than simple link costs.

3. The information collected and used may be too dynamic for a routing protocol.

4. Sharing of this information can be unnecessary when routing is driven by policy decisions rather than technical capabilities.

5. Centralization of the RuleSet creation is necessary because all RuleSets must cooperate toward a single goal, and the correct breakdown of responsibility cannot be determined from partial information.

6. Centralization of RuleSet creation allows a Responder to upgrade its policies independently and to improve the process if its developers have found better tuning mechanisms. A distributed mechanism would require that all nodes in the path upgrade at the same time to avoid the creation of inconsistent policies, and limit the ability of Responders (often service providers of some sort) to innovate.

4. Message Contents and Elements

This section outlines the principal information to be carried in the messages. Detailed packet formats are beyond the scope of this document, and may vary from network to network.

4.1. PathSetupRequest

At minimum, the PathSetupRequest message must contain:

1. node addresses for the Initiator and Responder

2. the class of service requested [qiroadmap]

3. minimum performance parameters (fidelity and throughput)

4.2. Quantum Capabilities (QCap)

A QCap (quantum capabilities) block to be added to the stack in the PathSetupRequest message describes the functions, performance and quality of the node and link. This may include:

1. the fidelity of Bell pairs created by the quantum channel

2. the fidelity of local operations performed by the node for purification or entanglement swapping

3. the rate at which entanglement can be created (Bell pairs per second)

The details of the required information may differ between networks. A standardized form of this information for sharing between networks will be used for internetworking operation.

4.3. RuleSets

A RuleSet block in the stack in the PathSetupResponse message describes the rules to be executed at each node. A rule consists of a Condition clause and an Action clause. A Condition clause lists the existence of particular entangled states, or the reception of particular messages. The Action clause describes the actions of purification, entanglement swapping, or even discarding an entangled state, as appropriate. The details are beyond the scope of this document.

In order to implement multiplexing schemes (e.g. buffer-space multiplexing, time-division multiplexing, or statistical multiplexing) based on the RuleSet-based network architecture, a RuleSet may include descriptors that define the usable resources for each link involved in that specific connection.

If a link carries only a single connection, all resources available may be fully assigned to that single connection to maximize the throughput. However, a link may receive a second RuleSet generated for a new connection. In that case, the nodes must be able to correctly update and reassign the available resources. Further details of the resource reservation and reclamation process are beyond the scope of this document.

5. Processing the SetupRequest

5.1. Initiating a Connection Setup Request

An Initiator, driven by an application request for quantum network services between itself and the Responder, builds the PathSetupRequest, populates the first QCap block, selects the next hop, and sends the request. Note that there is no need for either the Initiator or the Responder to know the entire network topology, only be able to select a next hop appropriately. The details of the routing are beyond the scope of this document.

5.2. Outbound Processing

Creation of the RuleSets requires knowledge of the number of nodes involved. A quantum node adds its own address when receiving the request packet, before sending to the next node. The stack size indicates how many nodes are involved. Additionally, the RuleSet creator may require information regarding links between nodes along the path - e.g. to be used when optimizing the order of entanglement swapping.

The pseudocode below outlines the processing on receipt of the PathSetupRequest message.

```
procedure ProcessFlatPathSetupRequest(Msg)
    Msg.HopStack.Push(MyHopInfo)
    if (MyAddr != Msg.ConnSpec.Responder)
        // Process and forward
        NextQuantumHop = GetNextQuantumHop(Msg.ConnSpec.Responder)
        LinkInfo = GetLinkInfo(NextQuantumHop)
        Msg.HopStack.Push(LinkInfo)
        Forward(NextQuantumHop,Msg)
    else
        // have reached the far end, need to build RuleSets
        // for everybody, then return
        ReturnMsg = ProcessFlatPath(Msg)
        MyRuleSet = ReturnMsg.RuleSetStack.Pop()
        InstallRuleSet(MyRuleSet)
        NextQuantumHop = ReturnMsg.RuleSetStack.Top.Addr
        Forward(NextQuantumHop,Msg)
    endif
endprocedure
```

Note that although we use the term "NextQuantumHop" here, that refers to a neighboring quantum node, and does not imply that the classical node’s neighbor is necessarily the same; it could, in theory, pass through multiple nodes to get there.

5.3. Responder Processing

The Responder accepts the final PathSetupRequest message with the complete stack of information about node capabilities and links, and builds a corresponding stack of RuleSets, one per node in the path. The Responder's processing is outlined in the "then" clause of the pseudocode above. The details of this creation process are beyond the scope of this document, and may be kept secret from other nodes in the path.

5.4. Return Processing

The pseudocode below outlines the processing on receipt of the PathSetupReturn message.

```
procedure ProcessFlatPathSetupReturn(Msg)
    MyRuleSet = ReturnMsg.RuleSetStack.Pop()
    InstallRuleSet(MyRuleSet)
    If (ReturnMsg.RuleSetStack.Size != 0)
        NextQuantumHop = ReturnMsg.RuleSetStack.Top.Addr
        Forward(NextQuantumHop,Msg)
    endif
endprocedure
```

The RuleSetStack should only be empty after the Initiator node of the original request removes its RuleSet, so this should be followed by activating the connection.

6. Rejection and Robustness of the Setup Process

6.1. Rejection by a Repeater or Router

A repeater or router that receives a PathSetupRequest may reject the request if it has no quantum communication resources available. It should not reject the request simply because it believes the requirements of the request (fidelity or rate) to be difficult to fulfill; that responsibility lies with the Responder.

When a node rejects the PathSetupRequest, it shall inform the other nodes along the portion of the path that have already received the PathSetupRequest by creating a PathSetupResponse message with an error code that indicates failure and sending that message to the node on the top of the stack. As with a successful PathSetupResponse, the list of nodes to which the message must be sent is created as a stack. Other than the addresses and the error code, the message may be empty; no RuleSets are required. The message is then iteratively returned, with each node popping its own address and forwarding to the next.

6.2. Rejection by a Responder

A Responder may reject a PathSetupRequest for any reason:

1. As with any classical system, it may simply choose to reject the request for any service-related reason, such as security, licensing, etc.

2. It may determine that the request cannot be fulfilled with the resources offered by nodes in the path.

When a node rejects the PathSetupRequest, it shall inform the other nodes along the path by creating a PathSetupReturn message with an error code that indicates failure and sending that message to the node on the top of the stack. As with a successful PathSetupResponse, the list of nodes to which the message must be sent is created as a stack. Other than the addresses and the error code, the message may be empty; no RuleSets are required. The message is then iteratively returned, with each node popping its own address and forwarding to the next.

6.3. Robustness

As the rate of connection initiation increases, competition for resources will also increase. A soft reservation mechanism that temporarily allocates resources in the anticipation of reception of a RuleSet may be used, with the reservation timing out and resources being released if no RuleSet arrives within a certain period. Specification of this mechanism is beyond the scope of this document.

Deeper integration of routing with real-time availability of resources is beyond the scope of this document.

7. Contributors

Besides the authors, Luciano Aparicio, Clement Durand, Dominic Horsman, Shota Nagayama, Takahiko Satoh, Shigeya Suzuki, Amin Taherkhani, and Joe Touch have made substantial contributions to the network architecture and the concepts described here.

We also thank Chia-Hung Chien, Kaori Ishizaki, Bill Munro, Kae Nemoto, Takafumi Oka, Shinnosuke Ozawa, and Thaddeus Ladd.

Comments by Wojciech Kozlowski, Gyananjay Rai and Patrick Gelard are reflected in this draft.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

Security implications of this entire process are extensive.

To minimize the probability of tampering, each information block added to the request on the outbound leg should be signed by the node adding the block.

Each information block describes hardware configuration, and therefore inherently leaks information about the network topology and condition. This document addresses only connection setup within a single network. Internetwork connection setup will require mechanisms to limit the leaking of sensitive network information across organizational boundaries.

Likewise, each RuleSet should be signed to prevent tampering during the PathSetupResponse phase.

Both the Request and Response phase may be encrypted using appropriate public key mechanisms.

It is also known that quantum networks may be vulnerable to attacks not possible in classical networks. These concerns are beyond the scope of this document.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[qiroadmap]
Wehner, S., Elkouss, D., and R. Hanson, "Quantum internet: A vision for the road ahead", Science 362, 2018.

[qnetworking]
Van Meter, R., "Quantum Networking", Wiley-iSTE , 2014.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[theqi] Kimble, J., "The Quantum Internet", Nature 453, 1023-1030, 2008.

Authors' Addresses

Rodney Van Meter
Keio University
5322 Endo
Fujisawa, Kanagawa 252-0882
JP

Phone: +81-46-649-3529
Email: rdv@sfc.wide.ad.jp

Takaaki Matsuo
Keio University
5322 Endo
Fujisawa, Kanagawa 252-0882
JP

Phone: +81-46-649-3529
Email: kaaki@sfc.wide.ad.jp