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 defînces 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.

- 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.

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

- 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.

- 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.

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

- 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.

- 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.

- 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:

- 0001: ACK

- 0010: Type K OK

- 0011: Type M OK

- 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:

- 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:

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

- 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

- 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, 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.

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:

- o 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.

- o 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