Entanglement Generation Protocol: Notes on Link+Physical Layer

Stephanie, Axel, Matthew, Erwin, Ronald

May 23, 2018

The objective of this document is to define the link layer in quantum networks connecting quantum processing nodes, and to propose a concrete link layer protocol based on an existing implementation of the physical layer with certain properties. In analogy to classical networks, the objective of the link layer will be to enable communication between two nodes A and B connected by a *link* on the same network. Here, enabling communication corresponds to producing entanglement between A and B, and we will hence refer to such protocols as Entanglement Generation Protocols (EGP). We propose the desired service, interface to the higher layer, as well as a concrete EGP. We first discuss an EGP between two nodes A and B, and discuss extensions to a proposed architecture connecting many nodes at the end.

To fit the link layer EGP into the future envisioned network stack we briefly sketch the stack framework here, going from higher to lower layer:

QTP - Qubit transport protocol (Transport Layer) Responsible for the end to end transmission of qubits.

EMP - Entanglement Management Protocol (Network Layer) Responsible for the generation of entanglement between two nodes that are not directly connected by a link, i.e. not on the same local network.

EGP - Entanglement Generation Protocol (Link Layer) Resonsible for the generation of entanglement between to nodes connect by a direct link.

1 Entanglement Generation Protocols

Let us first describe the interface, service, as well as performance criteria of entanglement generation protocols.

1.1 Higher layer to EGP

An EGP supports a single command from the higher layer, namely a request to produce entanglement, which we call a CREATE command. This command includes some desired properties of the entanglement such as for example a minimum fidelity, and a maximum waiting time. In an actual physical implementation, there is a tradeoff between these parameters. More time, for example, may allow the underlying implementation to use entanglement distillation to produce higher quality pairs.

CREATE Produce entanglement with a node on the same network (i.e. connected by a link). Arguments supplied are:

Partner ID	ID of the node to generate entanglement with.
Number k	Number of pairs we want to create.
$F_{\min}$	Minimum acceptable fidelity (with high confidence).
$t_{\max}$	Maximum acceptable waiting time before request is completed.
Purpose ID	Identifying the purpose or application at this node (optional, default 0).
Priority	Manual setting of a priority for entanglement production (optional).
create ID	Sequence number identifying this CREATE command.

1.2 EGP to higher layer

Following the reception of the CREATE command, several actions of the EGP are possible. Let us start with the positive outcome, and then consider possible errors.

OK Entangled pair has successfully been produced. One message per pair created, delivered immediately (best effort) following pair creation. With high confidence, the minimum acceptable fidelity $F_{\min}$ has been met, and the entanglement has been generated within the specified time frame $t_{\max}$. Information about the entanglement generation is provided, including an entanglement identifier. This identifier is required to be globally unique, and agreed upon by A and B. That is, A and B can locally use this entanglement identifier to determine which of their qubits is entangled with the remote node, and also which qubit belongs to which entangled pair. Entanglement identifiers are meant to be shared in the network by higher layer protocols and carry meaning beyond the nodes A and B. An entanglement identifier (Ent_{ID}) consists of:

(Node A ID, Node B ID)	IDs of the two nodes between which this entanglement is shared.
seqID	Sequence number. Unique (up to wrap around) between A and B, and globally unique when combined with the node IDs.
Goodness	Heuristic estimate for the fidelity of the generated pair.
$t_{Goodness}$	Time when this goodness was established (in EGP, usually the same as generation time).
t_{Create}	Time the pair was produced.

In addition the OK message also includes the following local information. We remark that Qubit IDs are exclusively local information (akin to the memory address in a computer) and not in general shared between network nodes.

Qubit ID	Logical Qubit ID of the entangled pair can locally be found.

Entanglement generation may fail for wide number of reasons, some of which form an immediate error. It may also be that the entanglement later expires, or is discarded of which the EGP will inform the higher layer. Let us start by listing the immediate failure modes, where in all such cases the create ID will be included allowing the higher layer to identify which request has failed.

ERR_UNSUPP Operation not supported. For example, creation of entanglement with the specified minimum fidelity is unattainable, or unattainable within the given time frame, even if the node is not loaded.

ERR_NOTIME Cannot meet the desired fidelity demand within the given time frame due to high load.

ERR_TIMEOUT Failure to produce entanglement within the specified time frame.

ERR_OTHER Failure for unspecified reasons, such as hardware failures.

In addition, the following failure mode can occur later when an entangled pair is expired. The primary use case of this will be to deal with extremely improbable failures in which recognition of the failure only becomes available after the higher layer has already received an OK message. This allows for a tradeoff between speed and certainty in recognizing failure modes. Since entanglement is very short lived, increased certainty can if desired be sacrificed for speed.

EXPIRE Expire Qubit ID. Any entanglement associated with Qubit ID has become unavailable.

1.2.1 Questions

- The term ”High confidence” is not defined and we need to decide what we mean by that, and also if this is some parameter where/by whom it is determined.

1.3 Performance metrics

Apart from correctly fullfilling requests, a variety of performance metrics can be considered for EGPs. Not all of these can be simultaneously optimized, but occasionally impose tradeoffs. We hereby also draw a distinction between performance metrics of interest to a specific “user” requesting entanglement from the EGP, and the overall performance of the network. Evidently, for all metrics below adverage, variance, and worst case behaviour is of interest. Once more data is available on how quantum networks are used in practise, one may also consider “typical” values for these metrics.

Let us first consider “user” centric metrics, measuring the experience of one invidual user rather than a behaviour of the network as a whole. We remark that nevertheless these metrics are a consequence of the total usage of the network.

Fidelity Quality of the entanglement produces. By design the fidelity has to exceed the minium requested fidelity $F_{\min}$.

Latency Time between submission of a CREATE request, and an OK response when successful. By design this time may not exceed $t_{\max}$.

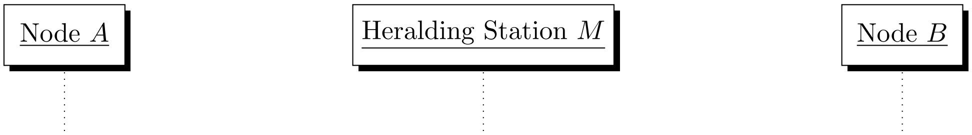
In addition, we can consider measures defined by the behaviour of the network when dealing with a large number of requests.

Throughput Number of pairs/s. Refined variants of throughput to be measured include: instantaneous throughput and sustained throughput.

Fairness Difference in performance metrics between requests originating at A and B.

Availability Availability is a concern here if a network node requires resetting and two nodes require resynchronization at certain time intervals.

This document will describe the desired service, interface as well as logical protocol of the EGP, which makes use of classical message delivery as well as entanglement generation. The protocol is geared towards producing entanglement between two nodes A and B, by means of an heralding station M between them. An extention to many nodes connected by means of switches to the heralding station will be discussed at the end.



As a high level summary (see details in section **??**), the desired services provided by the combined link+physical layer protocol is as follows:

- Confirmed (heralded) generation of entanglement with minimum fidelity F between A and B upon request.
- Reliability: “reasonable“ assurances that fidelity F is indeed supplied.
- Performance: If the number of generation requests is below a maximum threshold, then requests are honored w.h.p. within some time $t_{\max}$. Many other demands can be made, still TBD (eg on latency or throughput, depending on priorities, see Section **??**)
- Total order: A and B will agree on a total ordering of the entangled pairs without requiring further communication.
- Authenticity: If desired, control communication is authenticated.

This document is structured as follows: In Section 2, assumptions about the delivery of classical messages is described, as well as a short summary of well known classical methods that can turn these relatively weak assumptions into a higher quality assumptions that will be used in the final protocol at the end. In Section 3 an abstract version of a heralded entanglement generation protocol is described, along with the assumptions made about this protocol. This protocol forms the basis of the quantum part of the physical layer. As in the case of sending classical messages, we will lift this basic protocol using standard methods into one that satisfies slightly stronger assumptions which will be used in our overall protocol at the end. In Section **??** an elementary distributed task queue is described that will be used in the final protocol. Finally, Section **??** describes the overall protocol using the components developed in the earlier sections.

2 Sending classical messages

2.1 Starting situation

It will be assumed that there exists a means to transmit classical data between A, B and M. How this is realized is not the objective of this document, and it could be achieved both by a dedicated fiber (possibly using two wavelength for bidirectional communication), or interspersed with quantum signals on the same fiber. Of interest are merely standard numbers:

- Classical channels are bidirectional, meaning data could in principle be sent in both direction at the same time (and, as a relevant consequence, messages can cross and are not ordered in both directions)
- Likelihood of losses: p_{loss} probability of loss (e.g. following standard fiber loss plus electronics if applicable).
- Likelihood of errors: p_{err} probability of error - where we remark that as in other classical communication burst errors are probably dominant.
- Standard delays of interest: propagation delay (over the fiber), transmission delay (incl. delays of the electronics in putting the packets on the fiber), and processing delay, if known. We will assume that given the highly sophisticated electronics and the fact that the rate of classical communication is low due the relatively low repetition rate of entanglement generation attempts , the transmission and processing delay are essentially negligible.

2.2 Enhanced situation

Two standard methods exist to enhance this situation to the following, whose exact form and choice depends on the parameters above:

- Error dectection: This can be achieved using standard methods, where probably a simple CRC depending on the length of the headers is most appropriate. This will add a number of bits to the messages headers below if employed. For example, for a standard CRC-32 as used in Ethernet, the CRC is computed over the message and stored in a 32 bit header field.
- Message authentication: In this case, this refers to the fact that A knows the messages originate with B (and vice versa). Similarly, M can authenticate messages if desired. Such authentication can be realized using a message authentication code (MAC) (see eg **?**). These can be realize with varying levels of security. If A and B share consumable key (such as for example generated by QKD), they can afford to use one-time MAC which - similar to a one-time pad - offers information-theoretic security. Such MACS, for example based on two-universal hashing, can in principle be fast (see e.g. **?** needing however various levels of key), although it is a question whether they are fast enough to be useful in this context. Steph: Note that this does not automatically imply the entanglement is authenticated: this would only be the case if the midpoint is trusted which is evidently against all principles here. Given the local nodes take actions, like initiating entanglement generation for example, based on classical messages

received - it strikes me as highly desirable to do this in order to ensure some form of robustness, given that this can even affect the quality of the qubits already stored etc

3 Entanglement Generation

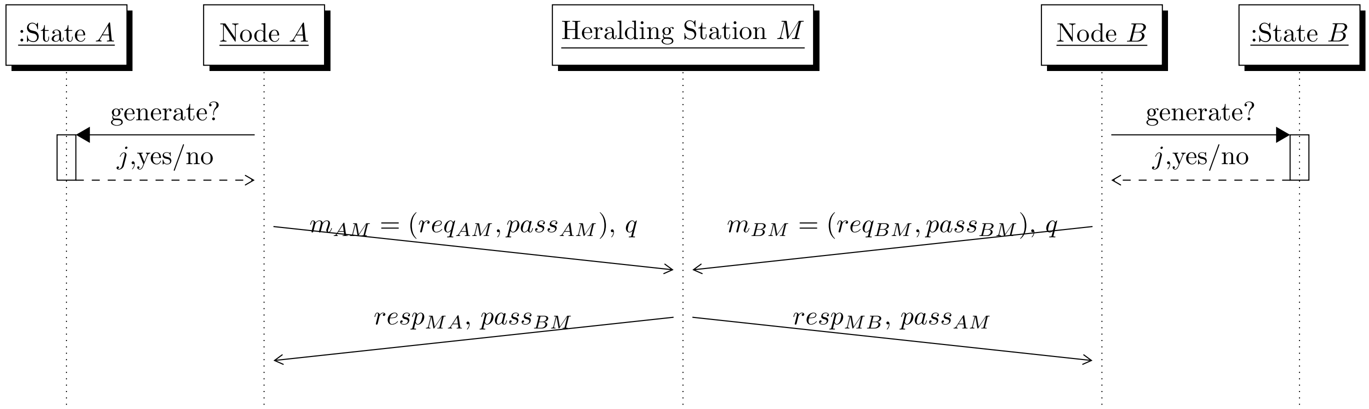
3.1 Starting situation

In the following, I will highly abstract away from the present entanglement generation protocols to focus only on the relevant elements for the final protocol, namely the exchange of classical messages, what information is available where and at what time and who can make decisions (such as choice of fidelity). The following model, also applies to the memory-assisted scheme based on entanglement distillation with minor modifications, but for simplicity I will assume that it is single click or BK.

General assumptions are:

- An association between the classical control messages m below, and the entanglement generation. For this reason, I will write classical message transmission as simply m, and q for arbitrary quantum signal q. To make it clear, how I will use this abstract description later, I will always take $m = (req, pass)$ where req is request information only for the mid point and later protocol specific, and $pass$ is something that will by default always be passed onto the other side (also protocol specific). Midpoint will provide a response $resp$, for example, success or failure.
- Variables $v_1, \ldots, v_k$, where v_j specifies the number of pairs yet to be created of quality choice j (for example, forming an agregate of a particular choice of bright state population α, or other parameters). It is assumed (see Section **??**) that A, and B agree on the values of these variables. For now simply take $k = 1$.
- Generation proceeds automatically in each time step, depending on the value of the variables above.

Very abstractly, a generation protocol thus takes the following form with respect to the classical states and message exchanges:



As a simple example, consider the single click protocol used in continuous mode as in **?** for the generation of one pair $v_1 = 1$ for an agreed upon quality, m_{AM}, m_{BM} are empty, $resp_{MA}, resp_{MB} \in \{OK, FAIL\}$. The protocol proceeds until the success is reached, in which case $v_1 = 0$ and generation stops. (Note that e.g. NV reset is not pictured here, as I'm only interested in classical message exchange)

4 Enhanced situation

Based on the general shape of such protocols above, one can now consider a slight "enhancement" of a protocol of this form - like single-click - that makes explicit some (probably obvious) failure modes, and produces a total ordering of pairs that A and B agree upon, even if some messages may go missing.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th><th></th></tr>
<tr><td colspan="32">Rest of header</td><td>} To be filled later</td></tr>
<tr><td colspan="32">Error detection CRC</td><td></td></tr>
<tr><td colspan="32">Message authentication (MAC)</td><td></td></tr>
</table>