Mathematical Modeling of Quantum Repeaters Chains

Lorenzo La Corte

July 5, 2024

Mathematical Model for Waiting Time and Fidelity

We derive expressions for the waiting time and fidelity of the first generated end-to-end link in the repeater chain protocol.

First, We derive a recursive definition for the random variable T_n, which **represents the waiting time in a $2n$-segment repeater chain**.

Then, we extend this definition to the Werner parameter W_n of the pair, which stands in one-to-one correspondence to its fidelity F_n using the equation:

$$F_n = \frac{1 + 3W_n}{4}. \tag{1}$$

The operations we study in the repeater chain protocols, denoted later as **protocol units**, are

- entanglement generation over a single hop,
- entanglement distillation,
- entanglement swapping.

And they all take a duration that is a multiple of L_0/c, the time to send information over a single segment.

For this reason, it is common to denote the waiting time in **discrete units** of L_0/c, which is a convention we comply with for T_n.

Cutoffs, introduced to optimize quantum repeater protocols, are also expressed in these discrete units.

Heraldeld Entanglement Generation

The first step in the repeater chain protocol is the generation of elementary entanglement between two nodes. In the following, we derive the waiting time and Werner parameter for the elementary entanglement generation protocol.

Waiting Time for Elementary Entanglement

In modeling the random variable T_n, which represents the waiting time in a 2^n segment repeater chain, we can reason by induction.

The base case T_0 **is the waiting time for the generation of elementary entanglement**.

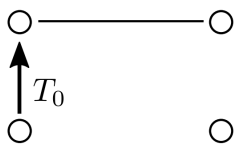

Figure 1: For two segments, T_0 represents the waiting time for the generation of a single link between two nodes without any intermediate repeater nodes.

Since we model the generation of single-hop entanglement by attempts which succeed with a fixed probability p_{gen}, the waiting time T_0 is a discrete random variable (in units of L_0/c) which follows a geometric distribution with probability distribution given by

$$\Pr(T_0 = t) = p_{\text{gen}}(1 - p_{\text{gen}})^{t-1} \quad \text{for} \quad t \in \{1, 2, 3, \ldots\} \tag{2}$$

where p_{gen} is the probability of success of the entanglement generation protocol.

Werner Parameter for Elementary Entanglement

The output state of the entanglement generation protocol is a Werner state with Werner parameter w_0.

Numerical Examples

To plot the CDF and PDF of the waiting time for the entanglement generation protocol, we can use the following code:

```
def entanglement_generation(p_gen=0.5, t_trunc=20):
    parameters = {
        # a protocol is represented by a tuple of 0 and 1,
        # where 0 stands for swap and 1 stands for distillation.
        # this example is a 0-level swap protocol,
        # as we consider only the entanglement generation.
        "protocol": (),
        # success probability of entanglement generation
        "p_gen": p_gen,
        # truncation time for the repeater scheme.
        # It should be increased to cover more time step
        # if the success proability decreases.
        # Commercial hardware can easily handle up to t_trunc=1e5
        "t_trunc": t_trunc
    }
    pmf, _ = repeater_sim(parameters)
    return pmf
```

Where the function `repeater_sim` is the algorithm introduced in [], which simulates the repeater chain protocol and returns the probability mass function (PMF) of the waiting time.

The following figures show the CDF and PDF of the waiting time for the entanglement generation protocol, with $p_{\text{gen}} = 0.2$ and $p_{\text{gen}} = 0.5$. We can see that the waiting time is **distributed according to a geometric distribution**, as expected.

In order for the algorithm to be computationally feasible, **the waiting time is truncated**, in this case at $t = 20$.

The parameter t_{trunc} in the algorithm should chosen to be large enough to capture the bulk of the distribution, but small enough to keep the computation time reasonable.

The truncation brings to an approximation of the distribution, as the probability of waiting for more than 20 units of time is not considered: this is more visible in the top plot, where the CDF is less close to 1 at $t = 20$.

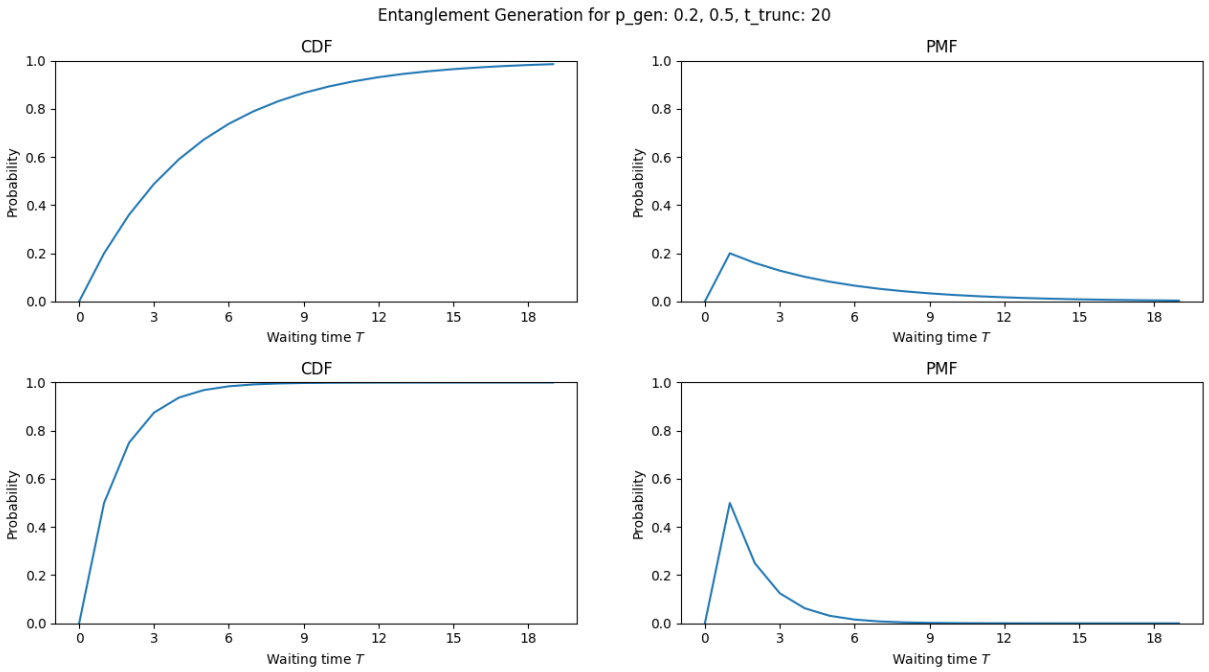

Figure 2: CDF and PDF of the waiting time for the entanglement generation protocol, with $p_{gen} = 0.2$ (top) and $p_{gen} = 0.5$ (bottom).

Entanglement Swapping

Once we have generated elementary entanglement, we can use it to **create entanglement over longer distances by entanglement swapping**.

Waiting Time for Entanglement Swapping

We defined T_0 as the waiting time for the generation of elementary entanglement, and our base for the induction. We now define our inductive step assuming that we have found an expression for T_n and we want to construct T_{n+1}.

In order to perform the entanglement swap to produce a single $(2^n + 1)$-hop link, a node needs to wait for the production of two (2^n)-hop links, one on each side.

Denote the waiting time for the pairs by $T_n^{(A)}$ and $T_n^{(B)}$, both of which are i.i.d. with T_n, as they are copies of it (see copies of a random variable).

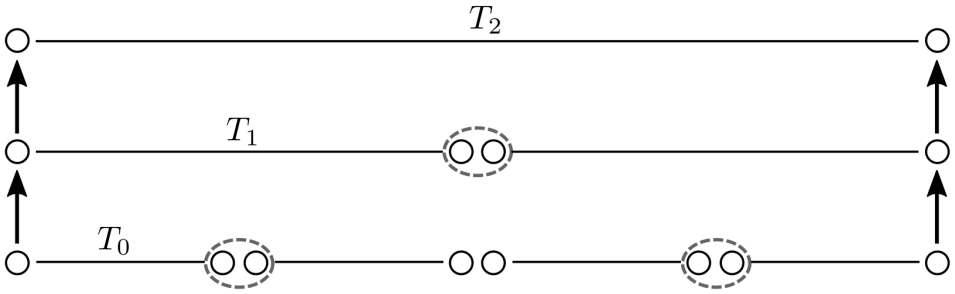

Figure 3: A link that spans 2^n segments is produced in a nested fashion, where at each nesting level two links are produced and subsequently swapped. []

Time until both pairs are available

We introduce a new random variable M_n modeling the time until both pairs are available

$$M_n := g_T(T_n^{(A)}, T_n^{(B)}). \tag{3}$$

The function g_T is generally defined as the maximum of the two waiting times

$$g_T(t_A, t_B) := \max(t_A, t_B). \tag{4}$$

which is also **the time at which the entanglement swap ends**.

This is distributed according to the CDF

$$\Pr(M_n \leq t) = \Pr(T_n^{(A)} \leq t \text{ and } T_n^{(B)} \leq t) = \Pr(T_n \leq t)^2. \tag{5}$$

From it, we can derive the PDF

$$\Pr(M_n = t) = \sum_{\substack{t_A, t_B: \\ t = \max(t_A, t_B)}} \Pr(T_n^{(A)} = t_A \text{ and } T_n^{(B)} = t_B). \tag{6}$$

Number of steps required

We introduce now K_n, a random variable following a geometric distribution

$$\Pr(K_n = k) = p_{swap}(1 - p_{swap})^{k-1} \tag{7}$$

modeling **the number of steps k until the first successful swap** at level n.

The fact that it follows a geometric distribution is a direct consequence of our choice to model the success probability p_{swap} to be independent of the state of the two input links.

Derivation of T_{n+1}

The derivation of T_{n+1} requires us to combine the random variable for the number of steps required K_n and the random variable for the waiting time for one attempt M_n.

In order to find the relation between M_n and T_{n+1}, note that when entanglement swap fails, the two input are lost and need to be regenerated. The regeneration of fresh entanglement after each failing entanglement swap **adds to the waiting time**.

Thus, T_{n+1} is a *compound random variable*: it is the sum of K_n copies of M_n.

Since the number of entanglement swaps K_n is geometrically distributed, we say that T_{n+1} is a *geometric compound sum* of K_n copies of M_n, denoted as

$$T_{n+1} = \sum_{k=1}^{K_n} M_n^{(k)} \tag{8}$$

Derivation of the PDF of T_{n+1}

The probability distribution of the waiting time T_{n+1}

$$\Pr(T_{n+1} = t) = \Pr\left[\left(\sum_{k=1}^{K_n} M_n^{(k)}\right) = t\right]$$

is computed as the marginal of the waiting time conditioned on a fixed number of swaps

$$\Pr(T_{n+1} = t) = \sum_{k=1}^{\infty} \Pr(K_n = k) \cdot \Pr\left[\left(\sum_{j=1}^{k} M_n^{(j)}\right) = t\right]. \tag{9}$$

Plugging in the expressions for $\Pr(K_n = k)$ (Equation 7) and recalling that the sum of k copies of M_n can be expressed as a convolution, we get

$$\Pr(T_{n+1} = t) = \sum_{k=1}^{\infty} p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \left(*_{j=1}^{k} m\right) \tag{10}$$

where, from Equation 6

$$m(t) := \Pr(M = t) = \sum_{\substack{t_A, t_B: \\ t = \max(t_A, t_B)}} \Pr(T_n^{(A)} = t_A \text{ and } T_n^{(B)} = t_B). \tag{11}$$

Werner Parameter for Entanglement Swapping

Considering two entangled pairs, respectively with Werner parameters w_A and w_B, the output Werner parameter w_{out}, if we do not consider decoherence, will be

$$w_{out} = w_A \cdot w_B. \tag{12}$$

However, when the first of the two pairs is generated, it has to wait for the elementary generation of the other; **during this time the first generated pair decoheres**. In particular, a Werner state $\rho(w)$ residing in memory for a time $\Delta(t)$ will transform into the Werner state $\rho(w_{decayed})$ with

$$w_{\text{decayed}} = w \cdot e^{-\Delta t/T_{coh}} \tag{13}$$

where T_{coh} is the joint coherence time of the two quantum memories holding the qubits.

Denote by A and B the input links to the entanglement swap and denote by (t_A, w_A) and (t_B, w_B) their respective delivery times and Werner parameters. Without loss of generality, suppose that the link A is produced after link B, i.e. $t_A \geq t_B$.

Link A is produced last, so the entanglement swap will be performed directly after its generation and hence link A will enter the entanglement swap with Werner parameter w_A. Link B is produced earliest and will therefore decohere until production of link A.

It follows that B's Werner parameter decoheres accordingly to (13), and therefore is, immediately before the swap, equal to

$$w'_B = w_B \cdot e^{-|t_A - t_B|/T_{coh}}.$$

Thus, the entanglement swap would produce the 2^{n+1}-hop state with Werner parameter

$$\begin{aligned} w_{out} &= w_A \cdot w'_B \\ &= w_A \cdot w_B \cdot e^{-|t_A - t_B|/T_{coh}}. \end{aligned} \tag{14}$$

Notice that choosing $t_A > t_B$ would have lead to the same result. In order to model at the same time the Werner Parameter and the Waiting Time we use a joint random variable (T_n, W_n).

In the **base case**, for a single segment ($n = 0$), we are in the (already discussed) entanglement generation phase. Here, the **waiting time and Werner parameter are uncorrelated** because we model the attempts at generating single-hop entanglement to be independent and to each take equally long.

At the **recursive step** n, we model the waiting time and Werner parameter as a joint random variable (T_n, W_n).

Forgetting Sum To define it properly, we introduce the concept of *forgetting sum* $\widehat{\sum}$, defined on sequences of tuples $\{(x_j, y_j) \mid 1 \leq j \leq m\}$ for some $m \in \{1, 2, \ldots\}$ as

$$\widehat{\sum}_{j=1}^{m} (x_j, y_j) := \left(\sum_{j=1}^{m} x_j, y_m \right) \tag{15}$$

This notation is key to characterize the joint random variable (T_{n+1}, W_{n+1}), as the Werner parameter of the output link is only influenced by the links that the successful entanglement swap acted upon, as we will see in the following.

Using this concept, we define the joint random variable (T_{n+1}, W_{n+1}) as

$$(T_{n+1}, W_{n+1}) := \widehat{\sum}_{k=1}^{K_n} (M_n, V_n)^{(k)}. \tag{16}$$

Starting from the auxiliary variable for the output waiting time of a pair M_n (Eq. 3), we introduce another auxiliary random variable for the Werner parameter of the output link V_n, which will depend on the Werner parameters of the input links.

We define the auxiliary joint random variable (M_n, V_n) as

$$(M_n, V_n) := g\left((T_n, W_n)^{(A)}, (T_n, W_n)^{(B)}\right). \tag{17}$$

The function g is given by

$$g\left((t_A, w_A), (t_B, w_B)\right) := \left(g_T(t_A, t_B), g_W\left((t_A, w_A), (t_B, w_B)\right)\right) \tag{18}$$

where g_T is defined in Eq. 4 and

$$g_W\left((t_A, w_A), (t_B, w_B)\right) := w_A \cdot w_B \cdot e^{-|t_A - t_B|/T_{coh}}. \tag{19}$$

with T_{coh} the quantum memory coherence time.

We already studied the expression for the waiting time T_{n+1}. The other random variable W_{n+1} directly derives from V_n, which expresses the Werner parameter of the produced 2^{n+1}-hop link in case the swap is successful.

To prove that V_n is formulated properly, we argue that g_W correctly computes the Werner parameter of the output link after an entanglement swap.

The key idea that leads to the forgetting sum in Equation 16 is the following: if the entanglement swap fails, then the 2^{n+1}-hop link with its Werner parameter will never be produced since both initial 2^n-hop entangled pairs are lost. Thus, **for the waiting time**

we should consider the sum of the waiting times of all the attempts, but for the Werner parameter we should only consider the last successful attempt.

Let's explain it further.

In order to find how the Werner parameter on level $n+1$ is expressed as a function of the waiting times and Werner parameters at level n, consider a sequence (m_j, v_j) of waiting times m_j and Werner parameters v_j, where j runs from 1 to the first successful swap k.

- m_j correspond to the waiting time until the end of the entanglement swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link
- v_j to the output link's Werner parameter if the swap were successful.

From previus results, we found that the total waiting time is given by $\sum_{j=1}^{k} m_j$, the sum of the duration of the production of the lost pairs (see Equation 10).

Note, however, that the Werner parameter of the 2^{n+1} hop link **is only influenced by the links that the successful entanglement swap acted upon**. Since the entanglement swaps are performed until the first successful one, the output link is the last produced link and therefore its Werner parameter equals v_k.

We thus find that the waiting time t_{final} of the first 2^{n+1}-hop link and its Werner parameter w_{final} are given by:

$$t_{final} = \sum_{j=1}^{k} m_j \tag{20}$$

$$w_{final} = v_k \tag{21}$$

or, in a more compact form

$$(t_{final}, w_{final}) = \left(\sum_{j=1}^{k} m_j, v_k \right). \tag{22}$$

Taking into account that the number of swaps k that need to be performed until the first successful one is an instance of the random variable K_n, we arrive at the full recursive expression for the waiting time and Werner parameter at level $n+1$ as given in Equation 16.

Derivation of W_{out}

Let's now derive the expression for the Werner parameter of the output link W_{out}.

Let $W_s(t)$ be the **average Werner parameter** of the output link of **one attempt**, given that it succeeds and finishes at time t

$$W_s(t) = \sum_{\substack{t_A,t_B: \\ t=\max(t_A,t_B)}} \Pr(T_A = t_A, T_B = t_B) \cdot [p_{swap} \cdot w_{\text{out}}](t_A, t_B) \tag{23}$$

By taking Eq. 16 and replacing the iteration over all pair of possible input Werner parameters for each k by convolution we get

$$W_{out} = \sum_{k=1}^{\infty} p_{swap}(1-p_{swap})^{k-1} \left[\left(*_{j=1}^{k-1} m \right) * (m \cdot W_s) \right]. \tag{24}$$

Where m is defined in Eq. 11.

W_{out} represents the weighted average of the Werner parameter of the output link, over all possible sequences of failed attempts, followed by a successful one:

- all possible sequences of failed attempts are represented by the $k-1$ of m,
- the successful attempt is represented by the convolution of m and W_s.

Numerical Examples

To plot CDF, PDF and Werner parameter of the waiting time for a 3-level entanglement swapping protocol, we can use the following code snippet:

```
def entanglement_swap(p_gen=0.5, p_swap=0.5, w_0=0.99, t_trunc=20):
    parameters = {
        # 3-level swap protocol, without distillation
        # spanning over 9 nodes (8 segments)
        "protocol": (0, 0, 0),
        "p_gen": p_gen,
        "p_swap": p_swap,
        "t_trunc": t_trunc,
        "w0": w_0,
        # the memory coherence time
        # in the unit of one attempt of elementary link generation
        "t_coh": 400,
    }
    pmf, w_func = repeater_sim(parameters)
```

```
        # remove unstable Werner parameter
        t = 0
        while(pmf[t] < 1.0e-17):
            w_func[t] = np.nan
            t += 1

        return pmf, w_func
```

The Figure 4 shows the CDF and PDF of the waiting time for the entanglement swapping protocol, with $p_{gen} = 0.2$, $p_{swap} = 0.2$, and $t_{trunc} = 8000$.

The truncation approximated the distribution: the probability of waiting for more than 200 units of time is not considered. Thus, the plot only covers 98.57% of the distribution. However, the initial bit of the Werner parameter curve is more visible using this truncation.

We can see that the Werner parameter, which measures the quality of the entanglement

- starts from a value around 0.43,
- immediately, at $t = 8$, reaches its peak value of 0.87,
- rapidly decreases, reaching a value similar to the starting one.

This is due to the fact that the entanglement is lost over time, and the longer the waiting time, the lower the quality of the entanglement.

We can also compare the undament of the three curves using two different input Werner parameters, as showed in Figure 5.

The waiting time distribution is not affected by the initial Werner parameter w_0, as expected.

The Werner parameter drops in both the experiments, as already mentioned before. Notice that the difference between initial qualities of entanglement $w_0 = 0.99$ and $w_0 = 0.9$ impacts heavily on the undament for the Werner parameter.

Entanglement Distillation

Entanglement distillation takes in input two low-quality entangled pairs and produces a single high-quality one. For entanglement distillation, the success probability p_{dist} depends on the Werner parameters of the input links A and B; in particular

$$p_{dist} = \frac{1 + w'_A w'_B}{2} \tag{25}$$

where the primed notation denotes the Werner parameter with decay applied only to the link that waits in memory.

To compute T_{out} and W_{out} for this protocol unit we **separate the expressions** for the waiting time probability distribution of a successful and failed attempt, respectively $P_s(t)$ and $P_f(t)$.

These are computed by iterating over all possible combinations of the input links' generation time t_A, t_B that lead to a waiting time t for this attempt.

We then express the total waiting time distribution T_{out} and the Werner parameter W_{out} as those of the successful attempt averaged by the occurrence probability of all possible sequences of failed attempts, where the weighted average is efficiently computed using convolution (see Equation 40).

Waiting Time for Entanglement Distillation

To compute he waiting time distribution we consider the generation time of a successful or failed attempt separately and use the joint distribution of M and Y, where again M is the waiting time and Y is a binary variable that indicates the success of the attempt.

We define the joint distribution that **one attempt** succeeds or fails and takes time t as

$$P_s(t) := \Pr(M = t, Y = 1) = \sum_{t_A, t_B : t = \max(t_A, t_B) + t_c} \Pr(T_A = t_A, T_B = t_B) \cdot p(t_A, t_B) \tag{26}$$

$$P_f(t) := \Pr(M = t, Y = 0) = \sum_{t_A, t_B : t = \max(t_A, t_B) + t_c} \Pr(T_A = t_A, T_B = t_B) \cdot [1 - p(t_A, t_B)] \tag{27}$$

where t_c is the time used for classical communication and local operation. Here, we iterated over all possible combinations of the input links' generation time t_A, t_B that leads to a waiting time t for this attempt.

The sum of the waiting time for **all attempts** can be obtained by

$$\Pr(T_{out} = t) = \sum_{k=1}^{\infty} \left[\left(*_{j=1}^{k-1} P_f \right) * P_s \right] \tag{28}$$

where the sum over k considers all the possible numbers of attempts, convoluting, for each case, $k-1$ unsuccessful attempts and one successful one.

Neither P_f or P_s characterizes a random variable since they are joint distributions including Y, that is to say, Ps and Pf do not sum up to 1. Instead, we have

$$\sum_t P_s(t) + \sum_t P_f(t) = 1. \tag{29}$$

The convolution here cannot be simply interpreted as a sum of two random variables. Instead, it is the summed waiting time conditioned on the success/failure of each attempt.

To summarize, the full computation goes as follows:

1. We compute P_f and P_s: the joint distribution that one attempt succeeds or fails and takes time t.
2. Once these are computed, we can compute T_{out} and W_{out} as convolutions of $k-1$ unsuccessful attempts and one successful one, summing over all possible numbers of attempts k.

Werner Parameter for Entanglement Distillation

Also for the Werner parameter, we use the separate expressions for the successful and failed attempts.

Moreover we use $W(t)$, the average Werner parameter, to reduce the dependence on Werner parameters to the dependence on the waiting time.

First, we compute the average Werner parameter of the output link of **one attempt**, given that it succeeds and finishes at time t

$$W_s(t) = \frac{\sum_{\substack{t_A, t_B: \\ t = \max(t_A, t_B) + t_c}} \Pr(T_A = t_A, T_B = t_B) \cdot [p_{dist} \cdot w_{\text{out}}](t_A, t_B)}{P_s(t)}. \tag{30}$$

where w_{out} is the Werner parameter of the output link if the distillation attempt **was successful**.

Next, we take a weighted average of W_s over all possible sequences of failed attempts, followed by a single successful attempt:

$$W_{out} = \frac{\sum_{k=1}^{\infty} \left[\left(*_{i=1}^{k-1} P_f^{(j)} \right) * (P_s \cdot W_s) \right](t)}{\Pr(T_{out} = t)}. \tag{31}$$

This give us the Werner parameter of the output link.

Numerical Examples

Here, we present the execution of a protocol involving entanglement generation, distillation, and swapping.

This example does not follow the standard number of nodes. As a matter of fact, we have 3 segments (not a power of two) and 4 nodes, including the end parties A and B.

The protocol is visualized in Figure 6, and results from the different steps are shown in Figure 7.

We can get some general insights from the plots:

- As the protocol units are applied, the probability that the link is produced at time t shifts to a larger t, indicating longer waiting time.
- When protocol units different from distillation are applied, the Werner parameter decreases for most times t.

One important aspect to remark is that the Werner parameter curves are truncated if the corresponding PMF value goes under a certain threshold. This is because for these values of the PMF, the corresponding Werner parameter is unstable.

It is also interesting to notice that not always the distillation brings to higher values for the Werner parameter, like in this case.

In particular, the outcome of the Werner parameter depends on the quality of the input links, the times of their generation, and the time of decoherence t_{coh}. In the Figure 8 we only consider the steps before and after distillation, presented as the second and third row in Figure 7. We fix all the parameters except for the time of decoherence, that takes values $t_{coh} = 400, 1600$.

We notice that in one case the produced link has higher Werner parameter far all t, while in the other this is not true.

Cutoffs

The *cutoff* operation selects the input links and accepts them if the cutoff condition described in Section II-B is fulfilled. We consider only the case where *cutoff* is followed by *swap* or *dist*, so that the two blocks together output a single entangled link.

Waiting Time Distribution for Cutoffs

We define a new binary variable Y_{cut} representing whether the cutoff condition is fulfilled. The corresponding success probability is described by p_{cut} in Table 1. In addition, we also define the waiting time of one cutoff attempt as Z, in contrast to M for a swap or distillation attempt. For *cutoff*, we need to distinguish the waiting time of a successful and a failed attempt. In the case of success, we always have $Z_s = \max(T_A, T_B)$, i.e., we wait until two links are produced. However, in the case of failure, the waiting time is different for different cutoff strategies. With the notation $Z_f = t_{\text{fail}}(T_A, T_B)$, we have the following: for *dif-time-cutoff*, $t_{\text{fail}}(T_A, T_B) = \min(T_A, T_B) + \tau$, because there is no need to wait for the second link longer than the cutoff threshold. For *max-time-cutoff*, $t_{\text{fail}}(T_A, T_B)$ is the constant τ, i.e., the maximal allowed waiting time. For *fidelity-cutoff*, it is $t_{\text{fail}}(T_A, T_B) = \max(T_A, T_B)$.

Similar as the nested structure shown in Fig. 2, a swap or distillation attempt is now composed of several cutoff attempts.

We can write its waiting time M as

$$M = \sum_k \left\{ \left[Y_{\text{cut}}^{(k)} \prod_{j=1}^{k-1} \left(1 - Y_{\text{cut}}^{(j)}\right) \right] \cdot \left[Z_{\text{s}}^{(k)} + \sum_{i=1}^{k-1} \left(Z_{\text{f}}^{(i)}\right) \right] \right\}.$$

This expression will replace $M = \max(T_{\text{A}}, T_{\text{B}})$ used in (4). For $\tau = \infty$ or $w_{\text{cut}} = 0$, i.e., no cutoff, Y_{cut} is always 1 . Therefore, $k = 1$ is the only surviving term and the two expressions coincide.

To calculate the waiting time distribution, we need three joint distributions- P'_{f} for unsuccessful input link preparation because of the cutoff, $P'_{\text{s,f}}$ for successful preparation but unsuccessful swap/distillation, and $P'_{\text{s,S}}$ for both successful

$$
\begin{aligned}
P'_{\mathrm{f}}(t) &= \Pr\left(M = t, Y_{\mathrm{cut}} = 0\right) \\
&= \sum_{t_{\mathrm{A}}, t_{\mathrm{B}}: t_{\mathrm{fail}}(t_{\mathrm{A}}, t_{\mathrm{B}}) + t_{\mathrm{c}} = t} \Pr\left(T_{\mathrm{A}} = t_{\mathrm{A}}, T_{\mathrm{B}} = t_{\mathrm{B}}\right) \\
&\quad \cdot \left[1 - p_{\mathrm{cut}}\right](T_{\mathrm{A}}, T_{\mathrm{B}}) \\
P'_{\mathrm{s,f}}(t) &= \Pr\left(M = t, Y_{\mathrm{cut}} = 1, Y = 0\right) \\
&= \sum_{t_{\mathrm{A}}, t_{\mathrm{B}}: \max(t_{\mathrm{A}}, t_{\mathrm{B}}) + t_{\mathrm{c}} = t} \Pr\left(T_{\mathrm{A}} = t_{\mathrm{A}}, T_{\mathrm{B}} = t_{\mathrm{B}}\right) \\
&\quad \cdot \left[p_{\mathrm{cut}} \cdot (1 - p)\right](t_{\mathrm{A}}, t_{\mathrm{B}}) \\
P'_{\mathrm{s,S}}(t) &= \Pr\left(M = t, Y_{\mathrm{cut}} = 1, Y = 1\right) \\
&= \sum_{t_{\mathrm{A}}, t_{\mathrm{B}}: \max(t_{\mathrm{A}}, t_{\mathrm{B}}) + t_{\mathrm{c}} = t} \Pr\left(T_{\mathrm{A}} = t_{\mathrm{A}}, T_{\mathrm{B}} = t_{\mathrm{B}}\right) \\
&\quad \cdot \left[p_{\mathrm{cut}} \cdot p\right](t_{\mathrm{A}}, t_{\mathrm{B}})
\end{aligned}
$$

The prime notation indicates that they describe the waiting time of one attempt in CUT-OFF, in contrast to one attempt in swap or distillation.

For one attempt in swap/distillation with time-out, we then get similarly to (7)

$$
\begin{aligned}
P_{\mathrm{S}}(t) = \Pr(M = t, Y = 1) &= \sum_{k} \left[\begin{pmatrix} k-1 \\ j=1 \end{pmatrix} P_{\mathrm{f}}^{\prime(j)} \right) * P'_{\mathrm{s,s}} \Bigg] (t) \\
P_{\mathrm{f}}(t) = \Pr(M = t, Y = 0) &= \sum_{k} \left[\left(P_{\mathrm{f}}^{\prime(j)} \right) * P'_{\mathrm{s,f}} \right] (t)
\end{aligned}
$$

as well as the expressions in Fourier space analogous to

$$
\begin{aligned}
P_{\mathrm{s}}(t) = \Pr(M = t, Y = 1) &= \mathcal{F}^{-1} \left[\frac{\mathcal{F}\left[P'_{\mathrm{s,s}}\right]}{1 - \mathcal{F}\left[P'_{\mathrm{f}}\right]} \right] \\
P_{\mathrm{f}}(t) = \Pr(M = t, Y = 0) &= \mathcal{F}^{-1} \left[\frac{\mathcal{F}\left[P'_{\mathrm{s,f}}\right]}{1 - \mathcal{F}\left[P'_{\mathrm{f}}\right]} \right]
\end{aligned}
$$

The total waiting time then follows by substituting the expressions for P_{f} and P_{S} above in (7) or (11).

For entanglement swap, i.e., constant success probability $p_{\mathrm{swap},}$, simplification can be made for this calculation. In this special case, $P'_{\mathrm{s,f}}$ and $P'_{\mathrm{s,S}}$ differ only by a constant and the same holds for P_{s} and P_{f}.

Werner Parameter for Cutoffs

For the Werner parameter, we now need three steps.

We start from calculating the resulting Werner parameter of a swap or distillation for the very last preparation attempt where $Y_{\text{cut}} = Y = 1$. It is denoted by W'_{s} and we only need to replace P_{s} by $P'_{\text{s,S}}$ and $p \cdot w_{\text{out}}$ by $p_{\text{cut}} \cdot p \cdot w_{\text{out}}$ in (12).

Next, we compute the Werner parameter $W_{\text{s}}(t)$ as a function of time t that includes the failed cutoff attempts, in analog to the derivation of (13). $W_{\text{s}}(t)$ is the Werner parameter that the pair of output links of cutofF will produce, given that the swap or distillation operation following is successful:

$$W_{\text{s}}(t) = \frac{\sum_{k=1}^{\infty} \left[\begin{pmatrix} k-1 \\ * \\ j=1 \end{pmatrix} P'_{\text{f}} * \left(P'_{\text{s,s}} \cdot W'_{\text{s}} \right) \right](t)}{P_{\text{s}}(t)}$$

Finally, we consider the time consumed by failed attempts in SWAP or DIST and obtain

Using the Fourier transform, the two expressions above become

$$W_{\text{s}}(t) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}\left[P'_{\text{s,s}} \cdot W'_{\text{s}}\right]}{1 - \mathcal{F}\left[P'_{\text{f}}\right]} \right] \frac{1}{P_{\text{s}}}$$

$$W_{\text{out}}(t) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}\left[P_{\text{s}} \cdot W_{\text{s}}\right]}{1 - \mathcal{F}\left[P_{\text{f}}\right]} \right] \frac{1}{\Pr\left(T_{\text{out}} = t\right)}$$

Cutoffs are conditions imposed on the pair generated through entanglement generation. They take in input an entangled pair shared between A and B and give in ouput either:

- the pair itself, if the cutoff condition is satisfied,
- nothing, if the condition is not satisfied.

Thus, cutoffs preceed either entanglement swapping or distillation. Therefore, we model cutoffs together with these two protocols, by taking their described mathematical models and expanding them to include the cutoffs conditions.

Closed Expressions in the Fourier Domain

- explain why the ft acts on a finite sequence of numbers

- what are the fft and ifft constraints?

Waiting Time Distribution for Entanglement Distillation

Since the discrete Fourier transform acts on a finite sequence of numbers we first truncate the probability distribution at a fixed time L,

The Fourier transform is a linear map.

Moreover it converts convolutions into element-wise multiplication

As a consequence, taking the Fourier transform of both sides of Equation ?? yields

Full Derivation The probability distribution of the total waiting time T_{out} for all attempts is given by:

$$\Pr(T_{\text{out}} = t) = \sum_{k=1}^{\infty} \left[\left(*_{j=1}^{k-1} P_f^{(j)} \right) * P_s \right](t) \tag{32}$$

Applying the Fourier transform $\mathcal{F}$ to both sides, we get:

$$\mathcal{F}[\Pr(T_{\text{out}} = t)] = \sum_{k=1}^{\infty} \left[\left(\prod_{j=1}^{k-1} \mathcal{F}[P_f^{(j)}] \right) \cdot \mathcal{F}[P_s] \right](t) \tag{33}$$

For k attempts, the productory $\prod_{j=1}^{k-1} \mathcal{F}[P_f^{(j)}]$ represents the convolution of the failure distributions in the Fourier domain.

From the assumption made, these distributions $P_f^{(j)}$ are identical, so:

$$\prod_{j=1}^{k-1} \mathcal{F}[P_f^{(j)}] = (\mathcal{F}[P_f])^{k-1} \tag{34}$$

Using the identity for the sum of a geometric series $\sum_{k=1}^{\infty} x^{k-1} = \frac{1}{1-x}$, the equation simplifies to:

$$\mathcal{F}[\Pr(T_{\text{out}} = t)] = \sum_{k=1}^{\infty} \left((\mathcal{F}[P_f])^{k-1} \cdot \mathcal{F}[P_s] \right) \tag{35}$$

Recognizing the geometric series in the sum, we obtain:

$$\sum_{k=1}^{\infty} (\mathcal{F}[P_f])^{k-1} = \frac{1}{1 - \mathcal{F}[P_f]} \tag{36}$$

Thus, incorporating the success term $\mathcal{F}[P_s]$, we get:

$$\mathcal{F}[\Pr(T_{\text{out}} = t)] = \frac{\mathcal{F}[P_s]}{1 - \mathcal{F}[P_f]} \tag{37}$$

Finally, applying the inverse Fourier transform $\mathcal{F}^{-1}$, we obtain:

$$\Pr(T_{\text{out}} = t) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[P_s]}{1 - \mathcal{F}[P_f]}\right](t) \tag{38}$$

Werner Parameter for Entanglement Distillation

The transformation of Equation ?? to Fourier space involves the same concepts and yields

Again, the convolution has become a productory of identical copies of the same random variable; the resulting term, exponentiated to $k-1$, is then transformed through geometric series, yielding the final expression.

Entanglement Swapping in the Fourier Domain

Equation ?? and ?? can also be moved to the Fourier space as follows.

The derivation is simple: both expressions can be written in Fourier space by substituting $P_s = p_{swap} \cdot m(t)$ and $Pf = (1 - pswap) \cdot m(t)$ in ?? and ??.

As a result, the waiting time distribution in the Fourier space is:

and the Werner parameter is

Cutoffs in the Fourier Domain

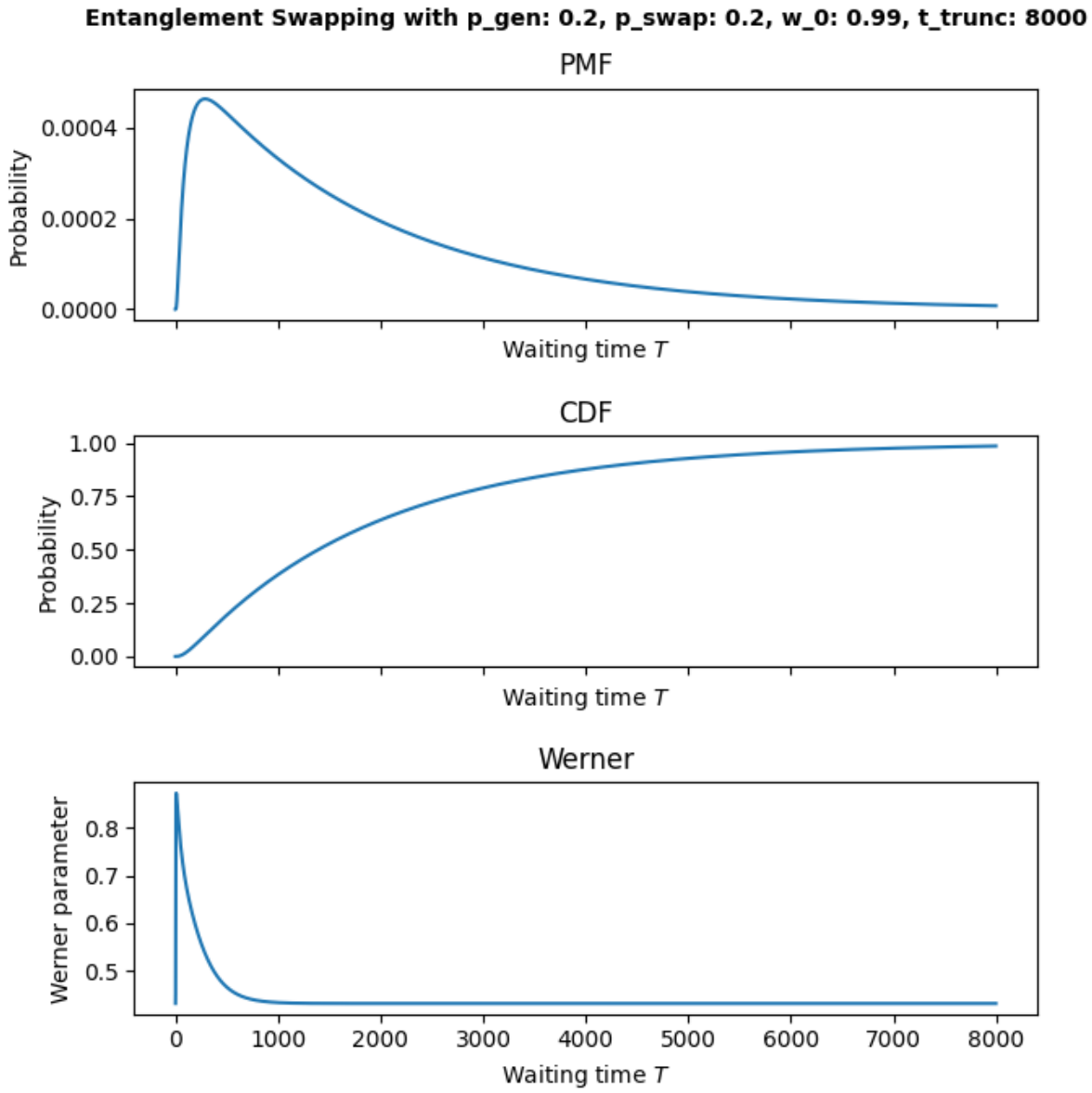

Figure 4: CDF and PDF of the waiting time for the entanglement swapping protocol, with $p_{gen} = 0.2$, $p_{swap} = 0.2$, $w_0 = 0.99$ and $t_{trunc} = 8000$.

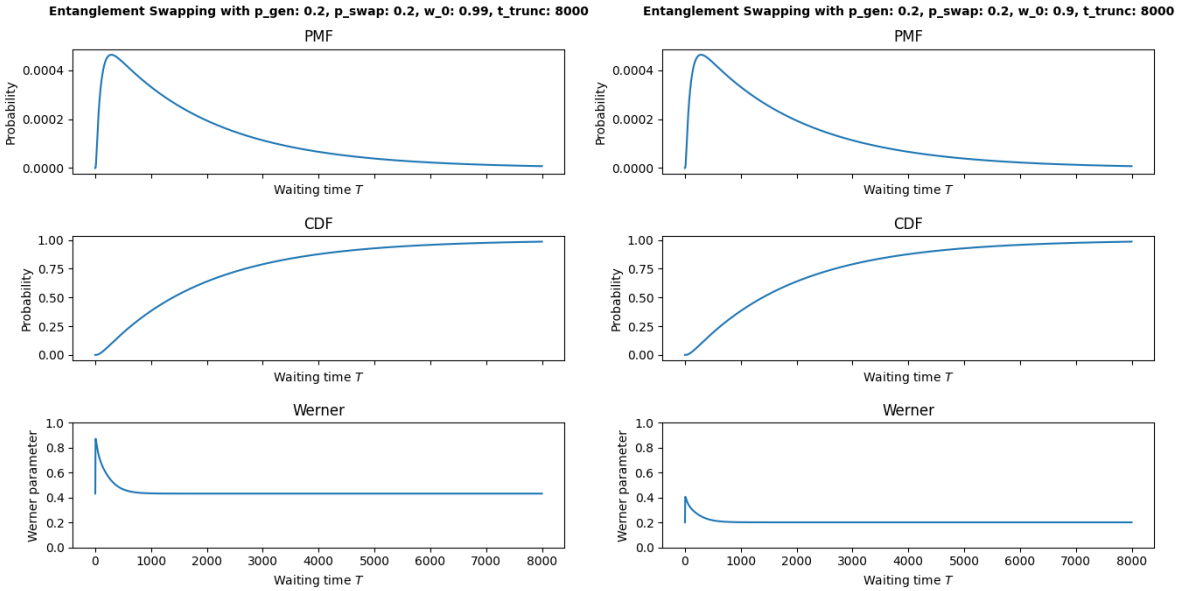

Figure 5: CDF and PDF of the waiting time for the entanglement swapping protocol, with $p_{gen} = 0.2$, $p_{swap} = 0.2$, $t_{trunc} = 8000$, and $w_0 = 0.99$ (left) and $w_0 = 0.9$ (right).

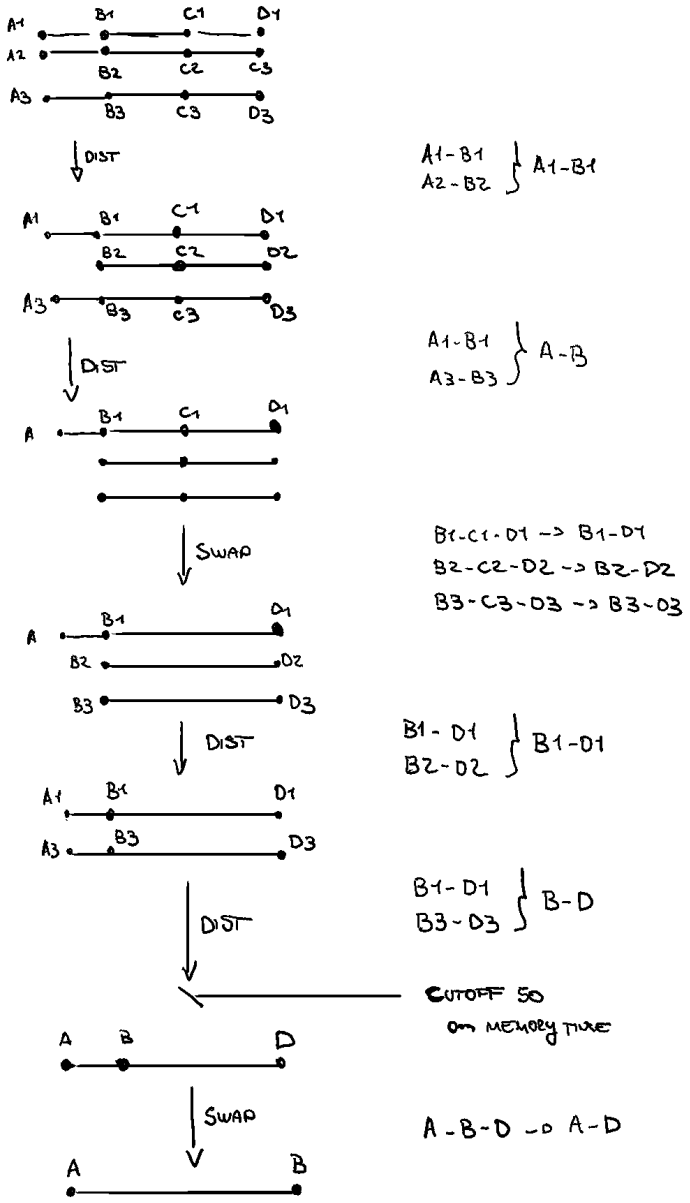

Figure 6: We show a mixed protocol, with the number of qubits and segments not being a power of two. Notice that it is only for illustrative purposes, and the protocol is not optimal. We have a four nodes repeater chain with three segments. The end nodes A and D have 3 qubits, while the intermediate nodes B and C have 6 qubits.

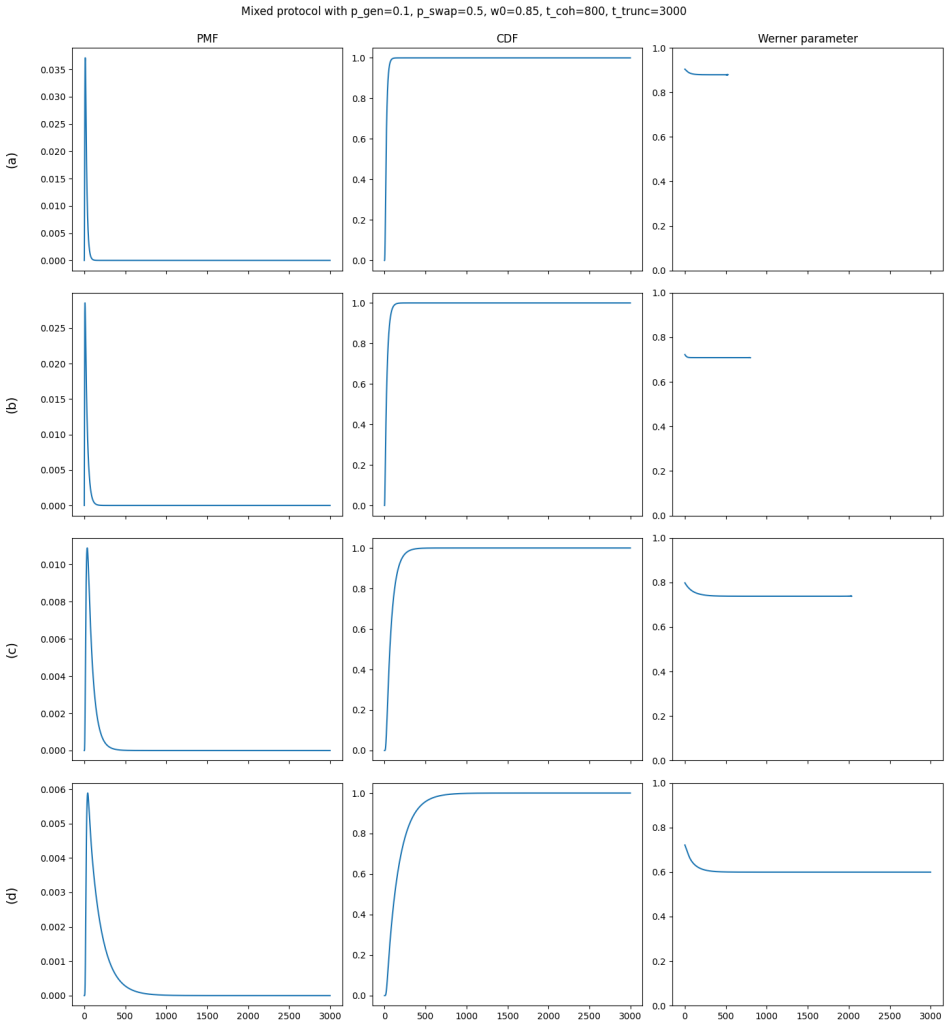

Figure 7: These plots show the waiting time distribution and the Werner parameter for the entanglement generation, distillation, and swapping protocol. (a) after the generation of the elementary links and the distillation of 3 low-quality links $A-B$ into a high-quality one. (b) Consider $i \in 1,2,3$. Three pair of links $(B_i - C_i, C_i - D_i)$, are swapped into the links $B_i - D_i$. (c) The three low-quality links $B_i - D_i$ are distilled into a single high-quality link $B - D$. (d) The end-link $A - D$ is produced starting from $A - B$ and $B - D$. The secret key rate at the end of the protocol is $3.1 * 10^{-3}$.

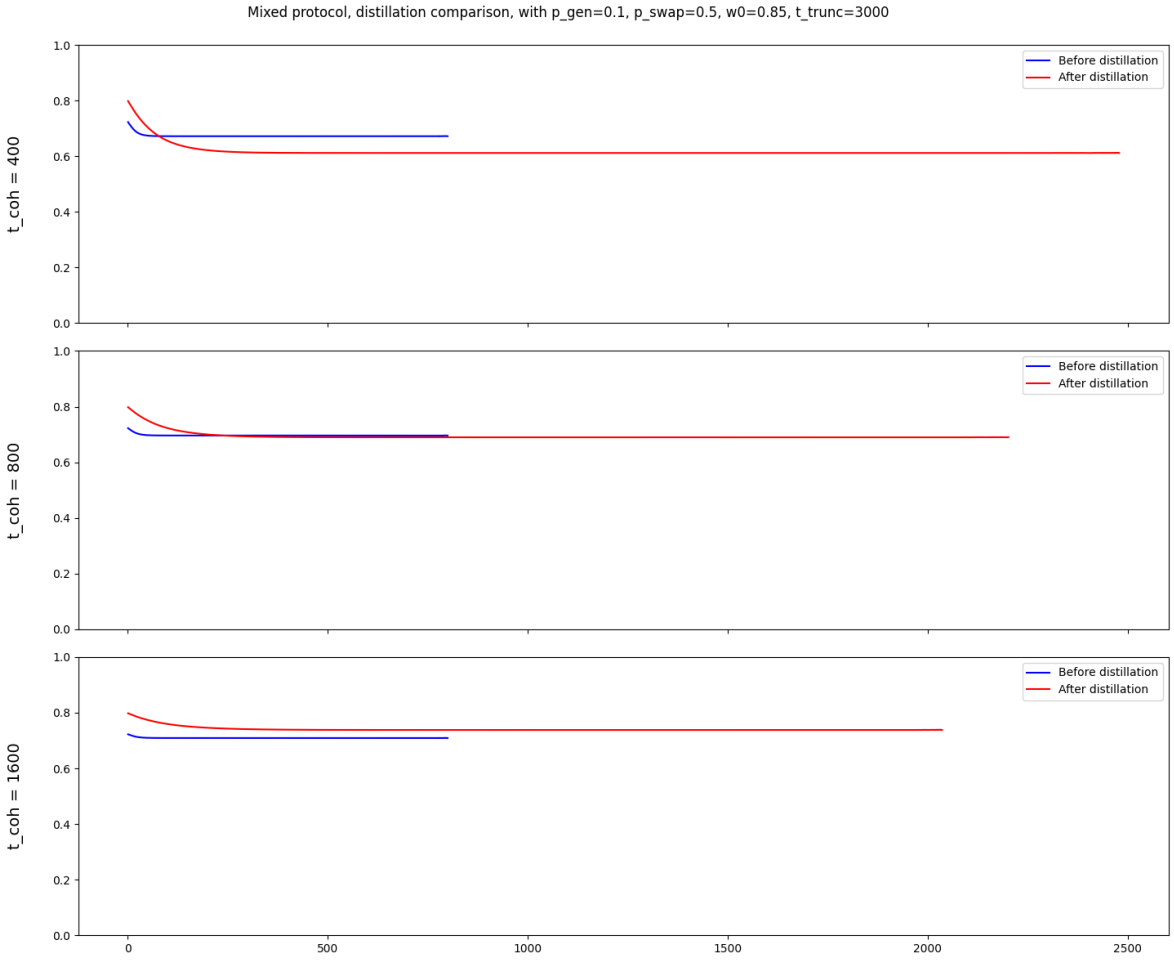
Mixed protocol, distillation comparison, with p_gen=0.1, p_swap=0.5, w0=0.85, t_trunc=3000
t_coh = 400
t_coh = 800
t_coh = 1600
Before distillation
After distillation
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500

Figure 8:

Mathematical Background

Convolution

Convolution is a fundamental mathematical operation used to combine two functions to produce a third function, which represents how the shape of one function is modified by the other.

For two discrete functions a and b, the convolution $a * b$ is defined as:

$$(a * b)(z) = \sum_{x=0}^{z} a(x) \cdot b(z - x) \tag{39}$$

Convolution of two Probability Distributions In the context of probability distributions, if X and Y are independent random variables with probability distribution functions p_X and p_Y, their sum $Z = X + Y$ has a probability distribution function p_Z given by the convolution of p_X and p_Y:

$$p_Z(z) = \sum_{x=0}^{z} p_X(x) \cdot p_Y(z - x) \tag{40}$$

Thus, convolution is used to determine the probability distribution of the sum of independent random variables by combining their individual probability distributions.

Example Consider this simple example. Let X and Y discrete independent random variables.

The probability distribution $\Pr(Z = z)$ of the random variable $Z = X + Y$ is computed as

follows:

x	$\Pr(X = x)$	y	$\Pr(Y = y)$	z	$\Pr(Z = z)$	Derivation of $\Pr(Z = z)$
0	0.2	0	0.3	0	0.06	$0.2 \cdot 0.3$
1	0.5	1	0.4	1	0.23	$0.2 \cdot 0.4 + 0.5 \cdot 0.3$
2	0.3	2	0.3	2	0.35	$0.2 \cdot 0.3 + 0.5 \cdot 0.4 + 0.3 \cdot 0.3$
				3	0.27	$0.5 \cdot 0.3 + 0.3 \cdot 0.4$
				4	0.09	$0.3 \cdot 0.3$

Note that $\Pr(Z = z)$ has five elements (all the possible sums), and it is valid as its probabilities sum to 1.

Associativity The convolution operator $*$ is associative, meaning that for any three functions a, b, and c:

$$(a * b) * c = a * (b * c) \tag{41}$$

Random Variables

In this section, we fix notation on random variables and operations on them.

Most random variables in the context of quantum repeaters

- are discrete,
- have as domain a subset of nonnegative integers.

PDF Let X be such a random variable, then its probability distribution function is a map

$$p_X : x \mapsto \Pr(X = x) \tag{42}$$

which describes the probability that its outcome will be $x \in \{0, 1, 2, \ldots\}$.

CDF Equivalently, X is described by its cumulative distribution function

$$\Pr(X \leq x) = \sum_{y=0}^{x} \Pr(X = y), \tag{43}$$

which is transformed to the probability distribution function as

$$\Pr(X = x) = \Pr(X \leq x) - \Pr(X \leq x - 1). \tag{44}$$

Independent Random Variables Two random variables X and Y are independent if

$$\Pr(X = x \text{ and } Y = y) = \Pr(X = x) \cdot \Pr(Y = y) \tag{45}$$

for all x and y in the domain.

Copies of a Random Variable By a *copy* of X, we mean a fresh random variable which is independent from X and identically distributed (i.i.d.). We will denote a copy by a superscript in parentheses. For example, $X^{(1)}$, $X^{(142)}$ and $X^{(A)}$ are all copies of X.

Expected Value The mean of X is denoted by

$$E[X] = \sum_{x=0}^{\infty} \Pr(X = x) \cdot x \tag{46}$$

and can equivalently be computed as

$$E[X] = \sum_{x=1}^{\infty} \Pr(X \geq x). \tag{47}$$

Function of Random Variables If f is a function which takes two nonnegative integers as input, then the random variable $f(X, Y)$ has probability distribution function defined as

$$\Pr(f(X, Y) = z) := \sum_{\substack{x=0,y=0 \\ f(x,y)=z}}^{\infty} \Pr(X = x \text{ and } Y = y). \tag{48}$$

Sum of Random Variables An example of such a function is addition.

Define $Z := X + Y$ where X and Y are independent, then the probability distribution p_Z of Z is given by

$$p_Z(z) = \Pr(Z = z) = \sum_{\substack{x=0,y=0 \\ x+y=z}}^{\infty} \Pr(X = x \text{ and } Y = y). \tag{49}$$

But since $y = z - x$, this is equivalent to

$$p_Z(z) = \Pr(Z = z) = \sum_{x=0}^{z} \Pr(X = x \text{ and } Y = z - x) \tag{50}$$

$$= \sum_{x=0}^{z} \Pr(X = x) \cdot Pr(Y = z - x) \tag{51}$$

$$= \sum_{x=0}^{z} p_X(x) \cdot p_Y(z - x) \tag{52}$$

which is the convolution of the distributions p_X and p_Y (see 40), denoted as $p_Z = p_X * p_Y$.

Notice that, since the convolution operator $*$ is associative (see 41), writing $a * b * c$ is well-defined, for functions a, b, c, from the nonnegative integers to the real numbers.

In general, **the probability distribution of sums of independent random variables equals the convolutions of their individual probability distribution functions**.

Geometric Distribution

The geometric distribution is a discrete probability distribution that models the number of trials needed to achieve the first success in a sequence of independent Bernoulli trials, each with the same success probability p.

Probability Distribution Function (PDF)

The probability distribution function (PDF) of a geometric distribution gives the probability that the first success occurs on the t-th trial. It is defined as:

$$\Pr(T = t) = p \cdot (1 - p)^{t-1} \quad \text{for} \quad t \in \{1, 2, 3, \ldots\}, \tag{53}$$

where:

- T is the random variable representing the trial number of the first success,
- p is the probability of success on each trial,
- $(1 - p)$ is the probability of failure on each trial.

This formula expresses that the first $t - 1$ trials must be failures, each occurring with probability $(1 - p)$, and the t-th trial must be a success (with probability p).

Cumulative Distribution Function (CDF)

The cumulative distribution function (CDF) of a geometric distribution gives the probability that the first success occurs on or before the t-th trial. It is defined as:

$$\Pr(T \leq t) = 1 - (1-p)^t. \tag{54}$$

Derivation of the CDF The CDF of a geometric distribution can be derived from its PDF as follows:

$$\begin{aligned}
\Pr(T \leq t) &= 1 - \Pr(T > t) \\
&= 1 - \sum_{k=t+1} \Pr(T = k) \\
&= 1 - \left\{ p(1-p)^t + p(1-p)^{t+1} + p(1-p)^{t+2} + \ldots \right\} \\
&= 1 - p(1-p)^t \sum_{k=0} (1-p)^k \\
&= 1 - (1-p)^t \sum_{k=0} p(1-p)^k \\
&= 1 - (1-p)^t.
\end{aligned}$$