Statistical Model Checking for Entanglement Swapping in Quantum Networks

Anubhav Srivastava and M. V. Panduranga Rao

Indian Institute of Technology Hyderabad
India
{cs21mtech02001,mvp}@iith.ac.in

Abstract. Given the fragile, stochastic and time critical nature of quantum communication systems, it is useful to analyse them with the rigour of formal methods. However, computationally expensive methods like exact probabilistic model checking do not scale with the size of the quantum network. In this work, we analyse entanglement swapping, an important primitive in quantum networks, using statistical model checking. We investigate the robustness of entanglement swapping against important parameters like longevity of quantum memory, success probability of entanglement generation and Bell State Measurements, and heterogeneity of the quantum network nodes. We demonstrate the usefulness of the approach using the MultiVeStA statistical model checker and the SeQUeNCe quantum network simulator.

Keywords: Quantum Networks · Statistical Model Checking · Discrete Event Simulators for Quantum Networks.

1 Introduction

While quantum communications promise to be (unconditionally) secure, they involve transfer of fragile qubits across long distances. While transmission of classical bits over large distances is relatively easy, some peculiarities of quantum mechanics like the *no-cloning* theorem precludes the use of conventional classical repeater and signal boosting approaches.

Fortunately, it has been shown that quantum repeaters are possible to envisage, building upon the so-called *entanglement swapping* protocol [32, 7, 14]. Indeed, this forms the basis of future quantum *networks* [30]. This protocol allows (geographically distant) distant nodes to share maximally entangled pairs (also called EPR pairs: see Section 2 for a brief introduction) via intermediate quantum repeaters. Once this is achieved, several protocols like teleportation of quantum information are possible between the quantum nodes [8].

At an abstract level, a quantum network can be represented by an undirected graph with the nodes called quantum nodes–we do not distinguish between "end nodes" and intervening repeaters. The edges represent a quantum channel, say a fibre optic channel, between two quantum nodes. A physical qubit can be

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

transmitted from one node to another directly if and only if they are adjacent quantum nodes.

Let us consider the simple case of three nodes a, b and c where a is adjacent to b and b is adjacent to c, but not a. To distribute an EPR pair between a and c, the following two steps need to be performed. First, an EPR pair each is generated between a and b, and b and c through the physical quantum channel. Then, a *Bell State Measurement* (BSM) performed at b, to convert the $a-b$ and $b-c$ EPR pairs to an $a-c$ EPR pair. This sequence of operations can be extended in principle to arbitrary path lengths. Once a route (cf [21, 25, 9, 10]), that is, a path from source to destination quantum nodes, is determined on the quantum network, the main challenge is to establish end-to-end entanglement through entanglement swapping. However, as we will see, entanglement swapping over long paths itself presents several design and implementation challenges. Therefore, we focus on issues pertaining to entanglement swaps along line graphs in this work.

Components of quantum communication systems are difficult to build as of today. For example, for the system to be of use practically, quantum memory (that is, registers of qubits) of reasonable longevity is needed–at least of the order of a few minutes. Secondly, quantum operations like unitary gates and measurements need to be performed on the qubits. These operations are currently very error-prone and in general stochastic in nature. Finally, these protocols involve classical communications as well. This also brings into play synchronization issues. As such, design and implementation of these systems is not straightforward. Specifically, it is not easy to decide whether a given configuration and figure of merit of the system can yield desired performance, in terms of parameters of interest. Examples of such parameters include success probability of sharing an entangled pair between two nodes of the quantum communication network within a stipulated time. More importantly, synchronization properties and scheduling in the context of heterogeneous nodes need to be investigated.

Formal methods, in particular (probabilistic) model checking techniques, offer an approach for analyzing such (stochastic) systems [5]. Indeed, the use of formal methods to study correctness of quantum programs is catching on [17, 20, 18]. Even quantum cryptography protocols have been subjected to analysis through formal methods [6, 15]. Desu et al. showed an approach to study quantum communications for timeliness through Probabilistic Timed Automata (PTA) and Probabilistic Timed Computational Tree Logic (PTCTL) [12]. Unfortunately, state space exploration based exact model checking approaches are computationally expensive and do not scale well [16, 24]. Moreover, the modeling effort is non-trivial.

Statistical Model Checking (SMC) offers an inexpensive alternative, with the facility of using a (discrete event) simulator for a model of the system being analyzed [2, 28, 27, 23]. Assuming that the simulator is faithful to the actual stochastic system in the relevant parameters, and a probability measure is well defined on its runs, it is a good substitute for the actual system. In these circumstances, it can be subjected to analysis through Statistical Model Checking and the results can be thought of holding for the actual system.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

Fortunately, while development of quantum communication hardware has been somewhat slow, there has been significant progress in the design and implementation of sophisticated quantum networks simulation software [11, 13, 31].

In this work, we explore SMC for studying quantum communication systems. We use the MultiVeStA model checker [26] in conjunction with the quantum network simulator SeQUeNCe [31]. Our contributions are two-fold:

- We integrate MultiVeStA with SeQUeNCe to perform statistical model checking on quantum network protocols–we discuss some important aspects of this integration. We make the integration software available at [1] for further use by Quantum Networks researchers and developers.
- We perform an extensive study of some queries that we think are important for understanding entanglement swap protocols in quantum networks. The study yields interesting insights. The queries that we discuss have the following (not mutually exclusive) objectives:
  - comparison with Probabilistic Timed Automata model based results of Desu et al [12].
  - some additional queries on swap scheduling that are of interest.
  - a query on entanglement swap synchronization that is not only of extreme importance but also shows the usefulness of the tool-chain in performing complex analysis through nested queries.

We hope that this work and the tool-chain reported will kick-start a greater effort in analyzing quantum communication protocols and hardware performance through model checking.

The rest of the paper is arranged as follows. We discuss briefly the necessary preliminaries in the next section. In Section 3, we describe the integration of the statistical model checker and the quantum network simulator. Section 4 discusses the results in detail and the insights that they provide. We conclude the paper in Section 5 with a discussion of some future directions.

2 Relevant Preliminaries and Previous Work

We assume a basic familiarity with quantum mechanics, like the state, evolution and measurement postulates [22]. In this work, we will be particularly interested in the "maximally entangled" EPR pair, or simply EPR pair defined by the quantum state vector $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$.

As mentioned in the previous section, the central objective in quantum networks is to distribute an EPR pair between two quantum nodes. While this is straightforward for adjacent nodes, *entanglement swapping* helps in achieving this objective between non-adjacent nodes. The protocol proceeds as follows. Consider a line graph quantum network with three nodes a, b and c. The nodes a and b are adjacent to each other in the sense that they share a link over which qubits can be transported without any loss. They generate and share the (maximally entangled) EPR pair $\frac{|0_a 0_b\rangle+|1_a 1_b\rangle}{\sqrt{2}}$. Similarly, b and c are adjacent to each

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

other and share $\frac{|0_b 0_c\rangle + |1_b 1_c\rangle}{\sqrt{2}}$. Thus, the node b has two qubits, one from the EPR shared with a and one with b. As per the entanglement swapping protocol b performs a Bell State Measurement [7] on these two qubits, yielding $\frac{|0_a 0_c\rangle + |1_a 1_c\rangle}{\sqrt{2}}$– an EPR pair shared between a and c.

If the quantum memory of even one of two nodes that share an EPR pair is not long-lived, the EPR pair itself is not long-lived. Thus, we can associate the notion of *quality* with such an "EPR edge"–the ability to sustain an EPR pair for long duration. The entanglement swap protocol is error-prone due to several reasons. For the simplest case discussed above, it depends on the quality of $a-b$ and $b-c$ EPR edges. Similarly, it involves unitary gates like the single qubit Hadamard gate and the two qubit CNOT gate, and a measurement operation, the implementation of which is also error-prone.

Considering this, a rigorous analysis of a system model has the potential to yield deeper insights into design and implementation, and fine tuning of system parameters. Example analyses possible for entanglement swapping are as follows:

- External behaviour: The most important metric for users and applications on quantum networks is the time-limit μ within which the EPR pairs are distributed: Does the hardware and software configurations have the capability of sharing an EPR pair between two nodes of a quantum network within μ time-steps?
- Internal behaviour: (a) How does the success probability improve with quantum memories of increasing longevity? (b) A network designers' primary task is to find optimal hardware and software parameter values for components according to their use case. For example, how many EPR pair generation attempts have to be made between nodes of low coherence (informally, fidelity or closeness to the original quantum state) time quantum memories, when compared to nodes of high coherence time quantum memories, to obtain a similar end-to-end EPR distribution probability? This provides network designers with insights about the trade off between the *lifetime* of quantum memories and *retrial_ attempts* and therefore time-limit to be allocated to different parts of the network. (c) Entanglement swap along a path entails scheduling the swaps at different nodes. What sequence of swaps gives the best success probability of sharing end-to-end entanglement? Given that all quantum nodes may not be equal in their properties like availability at a particular instant and memory lifetimes, is it possible to perturb the schedules with reducing success probabilities by too much? (d) Given the time criticality of the quantum operations, parallel sub-operations can cause synchronization errors. For example, for a Bell State Measurement to be performed at quantum node b, both $a-b$ and $b-c$ EPR pairs need to have been created within a short time-gap of each other. What is the probability that this indeed happens?

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

2.1 (Statistical) Model Checking

Model checking offers a powerful tool for performing this kind of analysis. Given a mathematically precise model of the system, the problem is to automatically check if it satisfies a property that is also specified formally as a statement in an appropriate logic [5]. Indeed, Desu et al [12] employ such a technique to study quantum networks. Modeling the quantum network as a network of Probabilistic Timed Automata and specifying properties in Probabilistic Timed Computational Tree Logic, they report a simple timing analysis of the system using PRISM [19] and validate it using the quantum network simulator SeQUeNCe [31]. However, in general, exact model checking for Probabilistic Timed Automata is notorious for being computationally expensive –EXPTIME-complete even for two clocks [16]. Moreover, as the system grows larger and more complex involving more features, modeling it as Probabilistic Timed Automata also becomes difficult.

Statistical model checking provides an alternative. Given an executable model of a stochastic system, Statistical Model Checking involves generating a Monte-Carlo sample of the runs of the model and evaluating a property of interest stated in an appropriate system of logic on each run. The number of runs is decided so as to allow the size of the $(1-a)*100\%$ confidence interval to be within $d/2$ distance of the estimated mean, for some chosen a and d [28, 27, 2].

MultiVeStA is a statistical model checking tool, which builds on a series of such tools. VeSta [29] supported a variety of modeling formalisms like (discrete and continuous time) Markov chains and the executable specification language PMaude for probabilistic read-write theories [3]. Further, not only were specification languages like PCTL supported, but also the QUAntitative Temporal EXpressions language (QuaTEx). The only requirements are that discrete event simulations can be performed on the models and the probability measures are well defined on the paths of the model. PVesTa provides supports for parallelism [4]. MultiVeStA provides for direct integration with discrete event simulators and supports counter-factual analysis [26]. The specification language MultiQuaTEx is a minor variant of QuaTEx.

2.2 SeQUeNCe

SeQUeNCe [31] is an open source discrete event quantum network simulator that aims at realistic simulation of quantum hardware and provides a suite of pre-built quantum networking protocols. It follows a layered architecture where each layer is implemented as a separate module.

It has a dedicated hardware layer that provides simulation components such as light sources, quantum memory and photon detectors. On top of the hardware layer, it provides an entanglement layer which includes protocols for entanglement generation, purification and swapping. SeQUeNCe also provides a network management module that performs *routing* to obtain an entanglement swap path between a source and destination and *reservation* to reserve the resources along the path for a request. SeQUeNCe provides other modules for quantum state

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

management, circuit execution, and event creation and execution during the simulation.

3 Integrating MultiVeStA with SeQUeNCe

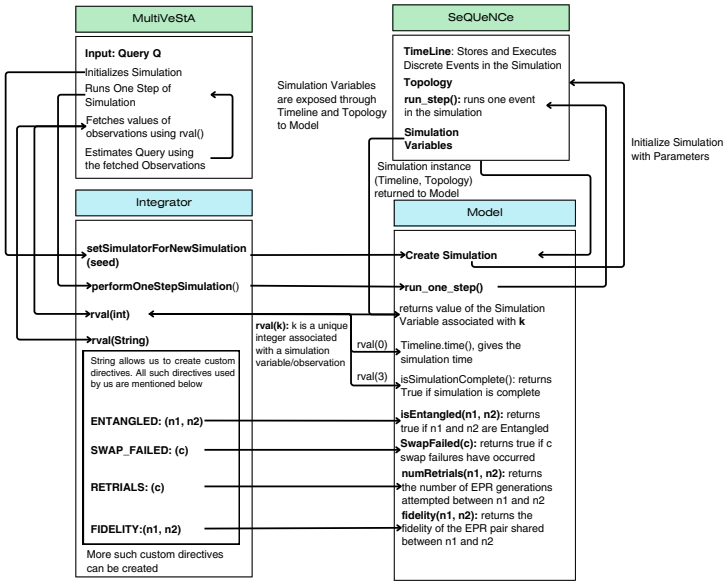


Fig. 1: Integration of MultiVeStA with SeQUeNCe

Fig 1 shows a high-level block diagram of the integration of MultiVeStA with SeQUeNCe. Briefly, we need to write two modules: the *Integrator* and the *Model*. MultiVeStA accesses SeQUeNCe through the *Integrator* to initialize, run the simulations and access the values of simulation variables. The *Integrator* implements a MultiVeStA specified function called $rval()$ that fetches the values of the simulation variables that are required to evaluate the queries, through the interface called *Model*. We also introduce custom *Directives* of the form of "$ACTION : (arg1, arg2, ..., argN)$" through the *String* arguments of $rval()$. These are extensible and new directives can be added depending on the requirements of the user. Please see the github repository for details [1].

4 Queries and Results

4.1 Experimental Setup

To demonstrate and exposit on different queries, we report experiments on linear topologies of three and five nodes and demonstrate how MultiQuaTEx queries can be used to obtain state information about the quantum network at any given time. To illustrate scalability, we use longer line graphs of up to hundred nodes.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

The metric of primary importance is the success probability of end-to-end entanglement. We vary various parameters and study their impact on this overall probability. We choose the same quantum channel attributes as in [12]:

```
quantum channel between adjacent nodes: 50km
attenuation of quantum channel: 0.2 dB/km
classical channel delay for adjacent nodes: 1ms
classical channel delay for path length 2: 2ms
quantum channel frequency =  100 GHz.
```

We focus on the following parameters in our *model*:

- τ is the lifetime of quantum memory
- μ is the time-limit within which an EPR pair should be shared between the end nodes
- p_{gen} is the success probability of EPR pair generation between neighbouring nodes
- p_{bsm} is the success probability of bell state measurement
- $retrial\_limit$ is the maximum number of times an operation can be attempted

In some experiments described in the next section we compare the results of MultiVeStA queries with earlier results of [12]. In such cases, we use the time t_{gen} taken for EPR generation between neighboring nodes to translate the time units between the MultiVeStA-SeQUeNCe implementation and the PTA model. In our MultiVeStA-SeQUeNCe experiments t_{gen} is 0.002 seconds and in the PTA model it is 5 time units. Therefore 0.002 seconds of the MultiVeStA-SeQUeNCe experiments are equivalent to 5 time units of the PTA model. We run MultiVeStA with the following parameters for all the experiments:

```
a (of the confidence interval) = 0.05
d (of the confidence interval)= 0.1
batch size = 100
```

4.2 Impact of time-limit (μ) on the success probability

Given a quantum network, a user would wish to ascertain if the hardware, software and the configurations are such that it is possible to distribute an EPR pair within a stipulated time. The MultiQuaTEx query that corresponds to this, is shown in the listing below. We consider a simple network of three nodes a, b and c and seek to distribute an EPR pair between nodes a and c. While this can be easily extended to longer paths, we stick to three nodes because of ease of explanation and comparison with a previous work that report similar experiments [12]. It shows a *parametric* query that estimates the probability of distributing an EPR pair between the nodes a and c by varying the time-limit μ. We fix τ to 0.006 and vary μ from 1.003 to 1.040 seconds. We choose these values for the following reason: in the PTA Model of [12], τ was set to 15 time-units which is 3×5 time-units and using the conversion factor that gives $\tau = 3 \times 0.002 = 0.006$ seconds for the MultiVeStA implementation. Similarly μ ranges from 10 to 100 for the PTA Model, which is equivalent to μ ranging from $1 + 0.002 \times 2$ to $1 + 0.002 \times 20 = 1.004$ to 1.040 seconds for the MultiVeStA implementation. From Fig. 2a we observe the success probability increases when μ increases. This is because when time-limit μ increases, more EPR generations can be attempted and one of them finally leads to a successful entanglement.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

This leads to EPR pairs becoming available for the Bell State Measurement to be attempted. The expected success probability however only reaches 0.5 because p_{gen} and p_{bsm} are set to 0.5. This setting limits the overall success probability. With better hardware, the probabilities of successful generation of an EPR pair and Bell State Measurement increases, and will result in higher probability of successfully sharing an EPR pair between a and c.

```
P_ac(mu) = if (s.rval(0) > mu*0.001
                || s.rval("Entangled: (a, c)") == 1)
                then s.rval("Entangled: (a, c)")
          else if (s.rval("SWAP_FAILED: 1") == 1)
                then 0
              else
                #P_ac(mu)
              fi
        fi;
    eval parametric(E[ P_ac(mu) ], mu, 1004, 4, 1041);
```

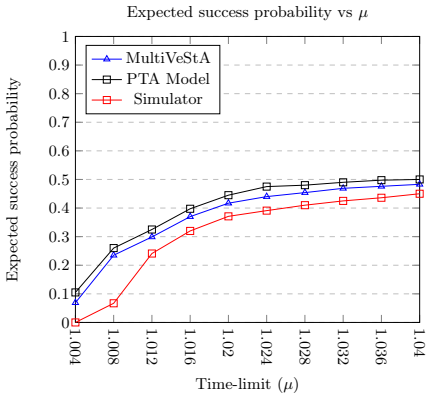
4.3 Impact of quantum memory lifetime (τ) on the success probability

The query that we discuss in this subsection aims to explore the impact of increasing the lifetime of quantum memory τ on the success probability of distributing an EPR pair.

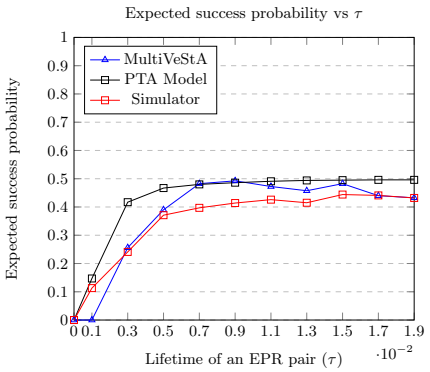
To perform this experiment, we set p_{gen} and p_{bsm} to 0.5 and the time-limit $\mu = 1.020$. These values are chosen to allow comparison with previous work [12]. The probabilities p_{gen} and p_{bsm} are kept the same as the PTA model whereas the time-limit μ has to translated in accordance with the MultiVeStA implementation. In the PTA model, μ was chosen as $10 \times \tau_{init} = 10 \times 5 = 50$ time-units. Following the same logic, we get $\mu = 1 + 10 \times \tau_{init} = 1 + 10 \times 0.002 = 1.020$ seconds. We add an offset of 1 second because the EPR distribution process starts at 1 second in current implementation. The query estimates the probability of an entanglement being distributed between a and c within the time-limit μ. If the time is less than 1.020 and a and c do not share an EPR pair, MultiVeStA moves the simulation to the next state using the next operator #. From Fig. 2b, we observe that the success probability increases with an increase in τ. This is because, by increasing τ, we allow EPR pairs to maintain fidelity for a longer time, thereby ensuring BSM is attempted. If successful, it yields an EPR pair between a and c. Similar to the last query, the success probability is limited to 0.5 because p_{gen} and p_{bsm} are 0.5. We see that the results from current implementation follow the trends previously discovered through the PTA Model.

```
P_ac()= if (s.rval(0) > 1.02 || s.rval("Entangled: (a, c)") == 1)
            then s.rval("Entangled: (a, c)")
        else if (s.rval("SWAP_FAILED: 1") == 1)
                then 0
             else
                #P_ac()
             fi
        fi;
eval E[ P_ac() ];
```

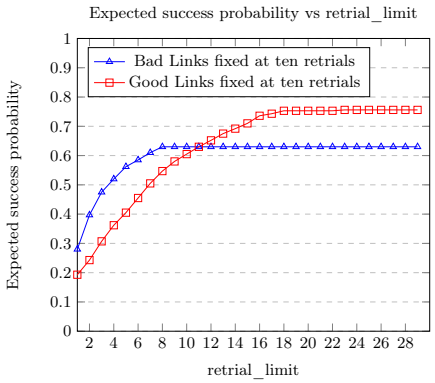
ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25



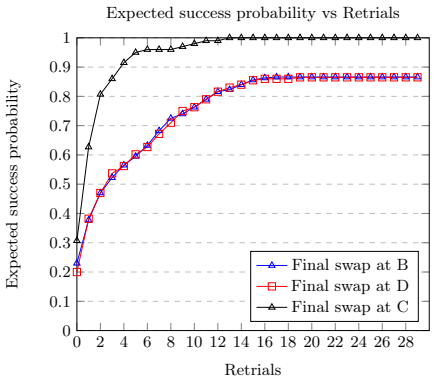
(a) $\tau = 0.006s$, $p_{gen} = 0.5$, $p_{bsm} = 0.5$.



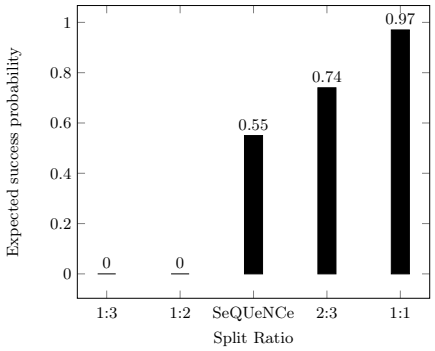
(b) $\mu = 1.02s$, $p_{gen} = 0.5$, $p_{bsm} = 0.5$.



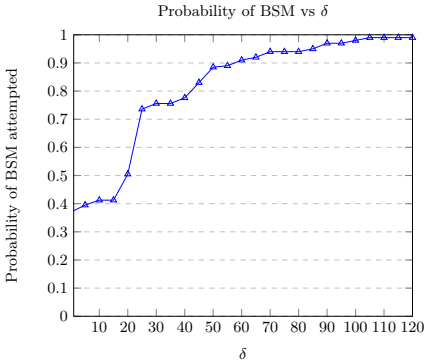
(c) Impact of retrial_limit and quality of links on overall success probability. $p_{gen} = 0.8$, $p_{bsm} = 1$, and $\mu = 1.1s$.



(d) Impact of schedule on final success probability



(e) Impact of Swap Schedule on Success Probability for longer paths



(f) Impact of δ on probability of swap being attempted

Fig. 2: Results of Queries

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

4.4 Impact of *retrial_limit* and *link_quality* on success probability

While in the previous experiments, we assumed that all quantum nodes are equal, we now drop that assumption. In this subsection, we wish to evaluate the impact of the quality of quantum memory and the number of retrials of EPR generation allowed on the overall success probability.

We define *retrial_limit* as the maximum number of EPR generation attempts allowed between two *adjacent* nodes. An EPR pair between two nodes is called *Good* if and only if the longevity of quantum memory at *both* nodes is high. Otherwise, it is called as *Bad*. The query in the following listing takes *retrial_limit* of each link as an argument. If at any time-step, any of the retrial limit is exceeded, the query returns 0. Otherwise, it runs until the time-limit μ. We report this experiment on a linear quantum network of five nodes labeled a to e. At every time-step, the query checks if nodes a and e are entangled, returning 1 if they are. At the end of the time-limit, the query returns 1 if a and e are entangled or 0 otherwise. The lifetime of quantum memories at nodes a and b is 0.010 seconds, at nodes d and e it is 0.006 seconds and at node c it is 0.008 seconds. As per definition, links ab and bc are *good*, whereas links cd and de are considered *bad*. Our objective through the query below is to show that *bad* links require greater number of retrials than *good* links to provide same success probability of entanglement distribution between nodes a and e.

```
P_ae(ab, bc, cd, de) =
      if (s.rval(0) > 1.1
        || s.rval("Entangled: (a, e)") == 1)
        then s.rval("Entangled: (a, e)")
      else if (s.rval("RETRIALS: (a, b)") <= ab &&
                s.rval("RETRIALS: (b, c)") <= bc &&
                s.rval("RETRIALS: (c, d)") <= cd &&
                s.rval("RETRIALS: (d, e)") <= de)
              then #P_ae(ab, bc, cd, de)
            else
              0
            fi
      fi;
```

We run this query with two scenarios, the results for which are shown in Fig. 2c. The blue plot shows the variation in overall success probability when *retrial_limit* for *bad* links is fixed at 10 and *retrial_limit* for *good* links is varied from 1 to 30. Similarly the red plot shows the overall success probability when *retrial_limit* for *good* links is fixed at 10 and *retrial_limit* for *bad* links is varied from 1 to 30.

We observe that for values of *retrial_limit* less than ten, fixing retrials for bad links seems to give better results. This is because *bad* links are fixed at a higher *retrial_limit* (=10) than the good links. Any EPR pairs that decohere along the *bad* links can be retried up to 10, yielding higher success probability. However, fixing *retrial_limit* for *good* links does not help much, as can be seen in the red plot. The *retrial_limit* for *bad* links is less than ten, and as such the decohering EPR pairs do not get many retrial opportunities for regeneration. If we look at *retrial_limit* greater than 10, we observe that red plot increases steadily with an increase in *retrial_limit* whereas blue plot plateaus out.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

The reason for the plateauing is as follows. For the blue plot, any further increase in *retrial_limit* for *good* links does not help because *good* links already hold entanglements for a long time and the failures are occurring due to entanglements decohering on *bad* links that do not have more attempts available.

On the other hand, the red plot increases steadily with an increase in *retrial_limit* as *bad* links are getting more attempts to generate the expired entanglements. The *good* links, fixed at 10 retrials, already have sufficient time to hold the entanglement until all the required entanglements become available for a Bell State Measurement. The two plots intersect at *retrial_limit*=10 where the *retrial_limit* of both the *good* and the *bad* links is 10.

4.5 Impact of *schedule* on success probability

Schedule refers to the order in which nodes attempt BSM operations along the path to distribute EPR pairs between the end nodes. Through this experiment we want to study variations in success probability for different values of retrials for two schedules.

For this experiment, we select a linear topology $a-b-c-d-e$ of five good nodes, all having a memory lifetime of 0.010 seconds. We set the p_{gen} to 0.8 as we want to study the impact of EPR generation failures and therefore retrials on the final success probability. We set the p_{bsm} to 1 because swap failures are independent of retrials along the edges or the coherence time of quantum memories. From Fig. 2d, we can see the success probability is higher when the final swap happens at c compared to when final swap happens at b or d.

This is because when final swap happens at c, the a-c and c-e entanglement operations can be attempted simultaneously in parallel. This allows a-c and c-e EPR pairs to be available for a BSM to be attempted to provide a-e. In case of a final swap at b, the first swap happens at d to give c-e and second swap happens at c to give b-e. This takes significant time within which the entanglement generated between a-b decoheres. This makes it difficult for EPR pairs a-b and b-e to be available at the same time for the final Bell State Measurement to be attempted at b. Symmetrically, the schedule with the final swap at d also faces the same problem.

Swap Schedules for Longer Paths. It turns out that the choice of schedule plays a critical role in general in the end-to-end entanglement distribution probability. We scale to line graphs of fifty nodes where each node has a memory lifetime of $0.1s$ and we set p_{gen} and p_{bsm} to 1. We compare five different schedules; the results of this comparison are shown in Fig. 2e. We refer to the node at which a Bell State Measurement takes place as the *swap-node*. The split is obtained recursively around this node. We define *split-ratio*, as the ratio of number of nodes to the left and right of the swap node.

We begin with a schedule called split-in-the-middle where the swap-node has equal number of nodes to the left and right (thus split ratio of 1:1). We see in Fig. 2e that this performs best (success probability of 0.97), this is because

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

independent swap operations can be attempted in parallel to the highest degree in this split.

We skew the split by decreasing the *split-ratio* to 2:3. In this case we see that the success probability decreases to 0.74. This is expected, since we have moved onto an asymmetric scenario where there is a possibility that one of the sides of the *swap-node* is not available for the swap to be performed. As we keep on skewing the split (by decreasing the split-ratio to 1:2 and then 1:3), we see that the probability drops to 0. This is because the entanglements on the final swap-node are not becoming available simultaneously due to different number of nodes on both sides. The default schedule created by SeQUeNCe results in the success probability of 0.55. These results show that a trade-off between a schedule perturbed away from 1:1 and successful probability can be struck, taking into account potential temporary unavailability of the nodes involved.

4.6 Impact of δ on probability of BSM being attempted

A related challenge while scheduling swaps, is to understand the impact of the time interval between parallel operations. Consider a three node quantum network as before. Although $a-b$ and $b-c$ EPR pairs can be generated in parallel, a significant lag in their completion times could lead to problems. Without loss of generality, let us say $a-b$ is completed, and waiting for $b-c$. It is possible that by the time $b-c$ is completed, $a-b$ decoheres and the subsequent Bell State Measurement (BSM) at b cannot be performed.

Therefore, we wish that both EPR formations happen within δ time of each other, for small δ. Therefore δ is an important parameter because EPR pairs are short-lived and we want the BSM to proceed as soon as they are generated. For example in the three node linear topology, if $a-b$ is generated first at time-step t, we would want $b-c$ to be generated within $t+\delta$. The query in the following listing returns 1 if both EPR pairs are available within $t+\delta$ time of each other and 0 otherwise. The query does this for different values of δ from 0 to 0.012 seconds, where 0 enforces both the entanglements should be available at the same time-step.

```
P_swap(T, delta) =
            if(s.rval(0)>T)
                then 0
            else if((s.rval("Entangled: (a, b)") == 1)
                    && (s.rval("Entangled: (b, c)") == 1))
                    then 1
                else if(((s.rval("Entangled: (a, b)") == 1)
                    || (s.rval("Entangled: (b, c)") == 1))
                    && (T == 1000))
                    then
                        #P_swap(s.rval(0)+delta*0.0001, delta)
                    else
                        #P_swap(T, delta)
                    fi
                fi
            fi;
eval parametric(E[ P_swap(1000, delta) ], delta, 0, 5, 120);
```

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

We can see from Fig. 2f that the probability of swap being attempted increases steadily with increase in δ. This is because an increase in δ increases the time window within which both EPR pairs have to become available for the Bell State Measurement to be attempted. We also observe that at $\delta = 0$, 20% of the time the EPR pairs a-b and b-c are available at the same time-step. When δ is 0.012, the probability of swap being attempted is almost 1, that is the EPR pairs $a - b$ and $b - c$ are almost always available for the Bell State Measurement to proceed.

4.7 Running Time

Fig. 3 shows the running time of a simple query regarding success probability of end-to-end EPR pair establishment with increasing path length, in increments of 10. We discuss the results in three regimes. The first two regimes are from path length 10 to 30, and 70 to 100. In these regimes, the probability of success is 1 and 0 respectively. Further, the CI conditions are satisfied with a small number of simulations. Therefore, the rise in running time is linear in path length in both the regimes. The second regime is from path length 40 to 60, where the probability of success lies strictly more than 0 and strictly less than 1. For these cases, to satisfy the CI conditions, more simulations need to be sampled. Therefore, there is a rise in running time in this regime. Nevertheless, an exact model checking approach would show a steep increase due to state space explosion for a model complex enough to capture such a system. These simulations were run on an AWS instance with 16GB of RAM and four virtual CPUs each having a clock speed of 2.3GHz.

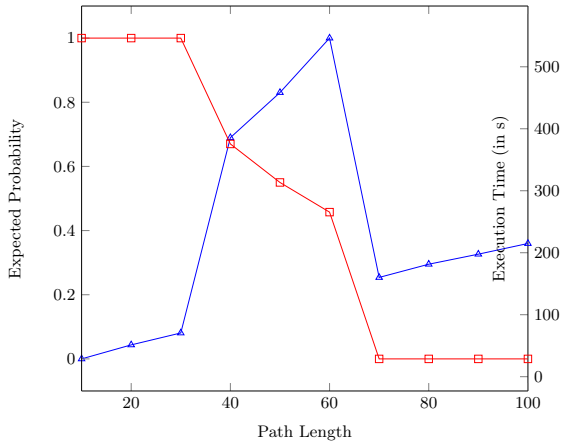


Fig. 3: Execution Time vs Path Length (Number of nodes). The blue plot shows the execution time (in seconds) and the red plot shows the corresponding probabilities of successfully establishing entangled pairs between the end nodes.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

5 Conclusions and Future Work

In this work, through an integration of a statistical model checker and quantum network simulator, we show how formal methods can be used for timing and performance analysis of quantum networks. A more detailed analysis need to be carried out that helps in identifying which hardware parameters make the most impact on price and performance. We believe this will help in planning for trade-offs in design of future quantum networks. An investigation through more expressive logics and more powerful model checkers would also yield rich dividends. Finally, can we borrow synthesis techniques from formal methods for developing high throughput, secure quantum networks?

Acknowledgments. The authors thank QCAL MeitY, Govt of India, for providing AWS quantum computing credits.

References

1. https://github.com/anubhavsrivastavaa/multivesta-qnetworks
2. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans on Modeling and Comp Simul (TOMACS) **28**(1), 6 (2018)
3. Agha, G.A., Meseguer, J., Sen, K.: Pmaude: Rewrite-based specification language for probabilistic object systems. Electron. Notes Theor. Comput. Sci. **153**(2), 213–239 (2006)
4. AlTurki, M., Meseguer, J.: Pvesta: A parallel statistical model checking and quantitative analysis tool. In: Algebra and Coalgebra in Computer Science - 4th Intl Conf, CALCO 2011. LNCS, vol. 6859, pp. 386–392. Springer (2011)
5. Baier, C., Katoen, J.P.: Principles of Model Checking (Representation and Mind Series). The MIT Press (2008)
6. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proc of IEEE International Conf on Computers, Systems, and Signal Processing. p. 175. India (1984)
7. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. Phys. Rev. Lett. **70**, 1895–1899 (Mar 1993)
8. Bouwmeester, D., Pan, J.W., Mattle, K., Eibl, M., Weinfurter, H., Zeilinger, A.: Experimental quantum teleportation. Nature **390**(6660), 575–579 (1997)
9. Caleffi, M.: Optimal routing for quantum networks. IEEE Access **5**, 22299–22312 (2017). https://doi.org/10.1109/ACCESS.2017.2763325
10. Chakraborty, K., Rozpedek, F., Dahlberg, A., Wehner, S.: Distributed routing in a quantum internet. arXiv preprint arXiv:1907.11630 (2019)
11. Dahlberg, A., Wehner, S.: Simulaqron—a simulator for developing quantum internet software. Quantum Sci and Tech **4**(1), 015001 (Sep 2018)
12. Desu, S.S.T., Srivastava, A., Rao, M.V.P.: Model checking for entanglement swapping. In: Formal Modeling and Analysis of Timed Systems - 20th International Conference, FORMATS 2022. LNCS, vol. 13465, pp. 98–114. Springer (2022)
13. Diadamo, S., Nötzel, J., Zanger, B., Beşe, M.M.: Qunetsim: A software framework for quantum networks. IEEE Transactions on Quantum Engineering **2**, 1–12 (2021)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25

14. Goebel, A.M., Wagenknecht, C., Zhang, Q., Chen, Y.A., Chen, K., Schmiedmayer, J., Pan, J.W.: Multistage entanglement swapping. Phys. Rev. Lett. **101**, 080403 (Aug 2008)
15. Huang, B., Huang, Y., Kong, J., Huang, X.: Model checking quantum key distribution protocols. In: 2016 8th Intl Conf. on Information Technology in Medicine and Education (ITME). pp. 611–615 (2016)
16. Jurdziński, M., Laroussinie, F., Sproston, J.: Model checking probabilistic timed automata with one or two clocks. In: Tools and Algorithms for the Construction and Analysis of Systems. pp. 170–184. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
17. Kakutani, Y.: A logic for formal verification of quantum programs. In: Advances in Computer Science - ASIAN 2009. Information Security and Privacy. pp. 79–93. Springer (2009)
18. Khatri, S.: On the design and analysis of near-term quantum network protocols using markov decision processes (2022), https://arxiv.org/abs/2207.03403
19. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV'11). LNCS, vol. 6806, pp. 585–591. Springer (2011)
20. Liu, J., Zhan, B., Wang, S., Ying, S., Liu, T., Li, Y., Ying, M., Zhan, N.: Formal verification of quantum algorithms using quantum hoare logic. In: Computer Aided Verification. pp. 187–207. Springer (2019)
21. Meter, R.V.: Quantum Networking. John Wiley & Sons, London (2014)
22. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
23. Nimal, V.: Statistical approaches for probabilistic model checking. Ph.D. thesis, University of Oxford (2010)
24. Norman, G., Parker, D., Sproston, J.: Model checking for probabilistic timed automata. Formal Methods in System Design **43**(2), 164–190 (2013)
25. Schoute, E., Mancinska, L., Islam, T., Kerenidis, I., Wehner, S.: Shortcuts to quantum network routing. arXiv preprint arXiv:1610.05238 (2016)
26. Sebastio, S., Vandin, A.: Multivesta: statistical model checking for discrete event simulators. In: 7th Intl Conf on Performance Evaluation Methodologies and Tools, ValueTools '13. pp. 310–315. ICST/ACM (2013)
27. Sen, K., Viswanathan, M., Agha, G.: Statistical model checking of black-box probabilistic systems. In: International Conference on Computer Aided Verification. pp. 202–215. Springer (2004)
28. Sen, K., Viswanathan, M., Agha, G.: On statistical model checking of stochastic systems. In: International Conference on Computer Aided Verification. pp. 266–280. Springer (2005)
29. Sen, K., Viswanathan, M., Agha, G.A.: VESTA: A statistical model-checker and analyzer for probabilistic systems. In: Second Intl Conf. on the Quantitative Evaluation of Systems (QEST 2005). pp. 251–252. IEEE (2005)
30. Wehner, S., Elkouss, D., Hanson, R.: Quantum internet: A vision for the road ahead. Science **362**(6412) (2018)
31. Wu, X., Kolar, A., Chung, J., Jin, D., Zhong, T., Kettimuthu, R., Suchara, M.: Sequence: A customizable discrete-event simulator of quantum networks (2020)
32. Żukowski, M., Zeilinger, A., Horne, M.A., Ekert, A.K.: "Event-ready-detector" bell experiment via entanglement swapping. Phys. Rev. Lett. **71**, 4287–4290 (Dec 1993)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_25