

# Resource estimation for delayed choice quantum entanglement based sneakernet networks using neutral atom qLDPC memories

S. Srikara<sup>1</sup>, Andrew D. Greentree<sup>2</sup>, and Simon J. Devitt<sup>1,3</sup>

<sup>1</sup>*Center for Quantum Software and Information, University of Technology Sydney, Sydney, New South Wales 2007, Australia*

<sup>2</sup>*Australian Research Council Centre of Excellence for Nanoscale BioPhotonics, School of Science, RMIT University, Melbourne, Victoria 3001, Australia*

<sup>3</sup>*InstituteQ, Aalto University, 02150 Espoo, Finland*



(Received 13 October 2024; accepted 28 March 2025; published 19 May 2025)

Quantum entanglement is a vital phenomenon required for realizing secure quantum networks, so much that distributed entanglement can be reimagined as a commodity which can be traded to enable and maintain these networks. We explore the idea of commercializing entanglement-based cryptography and future applications where advanced quantum memory systems support less advanced users. We design a sneakernet-based quantum communication network with a central party connecting the users through delayed choice quantum entanglement swapping, using quantum low-density-parity-check (qLDPC) encoded qubits on neutral atoms. Our analysis compares this approach with traditional surface codes, demonstrating that qLDPC codes offer superior scaling in terms of resource efficiency and logical qubit count. We show that with near-term attainable patch sizes, one can attain medium- to high-fidelity correlations, motivating further research towards the long-term realization of large-scale commercial quantum networks.

DOI: [10.1103/PhysRevResearch.7.023160](https://doi.org/10.1103/PhysRevResearch.7.023160)

## I. INTRODUCTION

The second quantum revolution refers to a future technological period when all the features of quantum mechanics are harnessed for information processing devices, expanding beyond the limited application of quantum mechanics in current computers, communication systems, and sensors [1–5]. The ultimate achievement of this revolution is embodied by quantum computers that are capable of operating on a vast scale, with the ability to tolerate faults and fix errors. These computers would need more than  $1 \times 10^6$  logical qubits [6] to tackle complex cryptographic tasks. While the creation of such devices remains in its early stages, there are compelling grounds to anticipate that scalable quantum computers will become tangible at some point within this century [7].

While there is reason to be hopeful about the progress of large-scale quantum computers, it remains uncertain if quantum computers for everyday consumers, such as a quantum smartphone, will ever become feasible. This refers to a compact and powerful computational device that is connected to a quantum network, possibly on a global scale. Possible constraints that could prevent the development of quantum smartphones include the requirement for extremely low temperatures to cool down qubits [8], or the necessity for optical communication channels [9] instead of microwave ones to connect to the quantum Internet, and the

vulnerability of qubits to mechanical vibrations and the impact of the Earth's magnetic field [10]. Suppose we accept the possibility of large-scale quantum computers and a quantum Internet. In that case, it is logical to enquire how a customer, who has no access to a quantum memory, but only has access to conventional gear such as classical detectors, classical memory, and classical computing, can participate in and perform complex quantum protocols.

One way is to do this by the use of distributed quantum entanglement [11]. Quantum entanglement is a pivotal phenomenon that forms the basis of various protocols in quantum communications and quantum computing [12–17]. Due to it being a major primitive for most quantum tasks, it would be natural to consider quantum entanglement as a tradeable commodity, such that users can purchase it to establish connections with other users in the network to perform quantum tasks such as quantum key distribution, distributed quantum computing, quantum authentication, etc. Commoditizing entanglement is an essential step toward large-scale commercial quantum networks.

To better demonstrate the above-mentioned ideas of entanglement commercialization and quantum access to common users with classical hardware, we design and analyze an entanglement one-time-pad QKD [18] based quantum network model networked using delayed choice entanglement swapping [19]. In this paper, we demonstrate that delayed choice entanglement swapping enables customers, who only own classical hardware, to use the offerings of a hypothetical future quantum memory company for the purpose of generating secure quantum keys. These keys are based on the E91 protocol [20]. Our design employs a central party Charlie, who generates Bell pairs, loads the halves separately

*Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.*

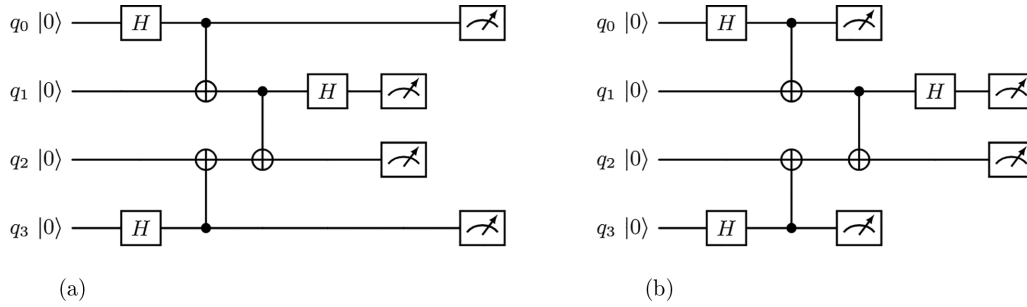


FIG. 1. Circuit diagram of quantum entanglement swapping. In (a), the conventional entanglement swapping [37] occurs, with Charlie performing the Bell measurement before Alice and Bob do. In (b), the order of measurement is reversed, with Alice and Bob measuring their qubits before Charlie. In both the circuits, all the qubits are starting at the  $|0\rangle$  state. These two circuits are equivalent as the two sets of measurements are parallel, nonoverlapping, and independent of each other.

onto different low-density-parity-check (qLDPC) codes, and distributes one set of the halves via sneakernet [21,22] to multiple end points where it is immediately measured upon arrival while keeping the other set of halves to himself on which he later performs Bell measurements to generate correlations between the end-point measurements of any pair of users. This enables the users to execute this task in a versatile manner by prepurchasing their key bits prior to selecting their communication partners. This protocol is beneficial for creating secure connections between networks of devices that have minimal hardware needs but can interact with a single, advanced quantum node or several nodes.

We design our protocol on quantum hypergraph product (HGP) codes [23], which are a class of qLDPC codes [24] for quantum error correction (QEC) [25], implemented on a neutral atom architecture [26,27]. These codes offer

significant advantages over traditional surface codes [28] in terms of encoding rate and resource efficiency [27,29]. Our analysis provides a detailed comparison between qLDPC and surface codes, demonstrating that qLDPC codes allow us to achieve higher logical qubit counts with fewer physical qubits as they have constant encoding rates regardless of patch size. This property potentially reduces the hardware requirements and improves the scalability of our proposed system. We have selected neutral atoms as our physical platform, as their ability to shuttle and rearrange within the lattice enables selective qubit unloading and efficient syndrome extraction [27]. This capability also allows us to use only one set of ancilla qubits and a single surface code patch, positioned adjacent to the qLDPC code block, for both loading and unloading operations via teleportation [27].

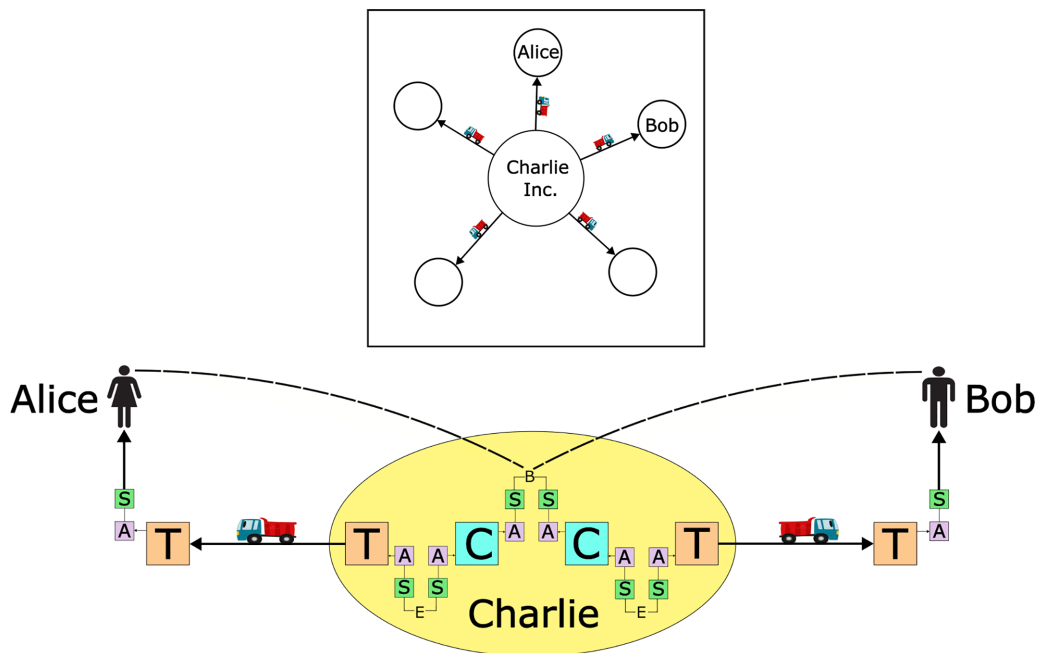


FIG. 2. Top: A schematic of a five-node delayed choice quantum network with Charlie Inc. in the center. Bottom: Delayed choice entanglement swapping protocol between two nodes Alice and Bob. S represents the surface code patches, A refers to the ancilla patches encoded in HGP code, C denotes Charlie's qLDPC memory, and T represents the qLDPC memory transported to the end nodes. Here, E represents the initial entangled Bell pair and B denotes Bell measurement.

## ALGORITHM 1. Delayed choice quantum network protocol.

---



---

```

1: for all end users do
2:   Charlie prepares Bell pairs encoded in surface codes
3:   Charlie loads each half of the Bell pairs into separate sets of qLDPC code blocks via teleportation
4:   Charlie stores one set of qLDPC blocks and transports the other set to destination
5:   End user puts in a purchase order for  $n_{\text{bits}}$  number of bits as per their requirement
6:   for  $i$  from 1 to  $n_{\text{bits}}$  do
7:     qATM at the destination unloads a logical qubit from a qLDPC block onto a surface code
8:     qATM measures the logical surface code qubit to get a bit
9:     qATM transfers the bit onto the user's smartphone
10:    Bit stored in the end user's smartphone for later use.
11:   end for
12: end for
13: Publish measurement bases
14: for users Alice and Bob wishing to connect do
15:   Alice and Bob decide to correlate  $n_{\text{key}}$  number of bits
16:   for  $i$  from 1 to  $n_{\text{key}}$  do
17:     Charlie unloads two corresponding qLDPC qubits onto two surface code patches
18:     Charlie performs logical Bell measurement on surface code patches
19:     Charlie broadcasts the measurement result to Alice and Bob, resulting in a correlated key bit
20:   end for
21:   Alice and Bob perform standard reconciliation techniques and get the final secure key
22: end for

```

---



---

The foundation of our paper is based on the research conducted on consumer-level quantum key distribution [30–32], as well as the research on the quantum sneakernet [21], which explores the spread of entanglement across long distances with high bandwidth and low latency. While our primary focus is on secure communication protocols, it is feasible to adapt our methodology for many applications such as quantum voting [33], clock synchronization [34], and reference frame alignment [35].

This paper is organized as follows: In Sec. II, we briefly describe the phenomenon of delayed choice entanglement swapping. In Sec. III, we describe our design. Section IV derives the closed-form mathematical expressions necessary for our resource estimations. In Sec. V we discuss our findings, and finally conclude our paper in Sec. VI. A link to the PYTHON code of our analysis is given in [36].

## II. DELAYED CHOICE ENTANGLEMENT SWAPPING

To describe our protocol, we start by providing a concise overview of the initial delayed choice entanglement-swapping technique, as explained by Peres [19]. In accordance with Peres, we examine the most basic iteration of this protocol. This protocol aims to establish entanglement and eventually generate correlated bits between two parties, Alice and Bob, who choose a third party, Charlie, to mediate the entanglement generation. Consider three individuals, namely Alice, Bob, and Charlie, as seen in the circuit diagram in Fig. 1. Two Bell pairs  $q_0q_1$  and  $q_2q_3$  are present such that  $q_0$  is in the custody of Alice,  $q_3$  with Bob, and  $q_1$  and  $q_2$  with Charlie. Qubits  $q_0$  of Alice and  $q_3$  of Bob get entangled when Charlie performs a joint Bell measurement on his qubits  $q_1$  and  $q_2$ . However, the specific relationship between Alice's and Bob's qubits cannot be known until Charlie reveals the outcome of his Bell state measurement. Alice's and Bob's photons are

perfectly entangled, disregarding any errors or decoherence. This exact entanglement allows them to be used for various entanglement-based protocols, such as cryptography using the E91 protocol [20].

Alternatively, Charlie has the option to measure his photons separately. By adopting this approach, there would be a complete absence of entanglement between the photons belonging to Alice and Bob. Consequently, any correlations seen would be purely coincidental and not indicative of any underlying connection. If Charlie attempted to conceal his independent measurements, Alice and Bob might easily expose the deceit by using the conventional method of conducting measurements and announcing certain findings subsequent to receiving Charlie's "correction" signals.

Peres's idea goes beyond entanglement swapping by recognizing that Alice, Bob, and Charlie might potentially be spacelike separated. This suggests that one can discover a frame of reference when the sequence of qubit measurement is modified, as shown by Zbinden *et al.* [38]. In the preceding discourse, we used the assumption that Charlie conducted his measurement before Alice and Bob did. Alternatively, it is possible for Alice and Bob to conduct their measurements prior to Charlie [Fig. 1(b)]. In this scenario, the outcome of their measurements itself is not affected by whether or not Charlie performed either the Bell state or independent measurement, but the degree of correlation between Alice's and Bob's measured bits is dependent upon the kind of measurement performed by Charlie. This serves as the foundation for the delayed choice entanglement-swapping protocol and is crucial for the advancement of our ideas. In the delayed choice entanglement swapping technique, Alice and Bob perform measurements on their qubits and record the outcomes using classical storage. At a later point, Charlie has a decision to either measure his qubits separately or do a Bell state measurement. The measurement outputs are either associated in

a known manner or wholly uncorrelated, depending on the measurement option chosen by Charlie and the random basis choice of Alice and Bob.

### III. SETUP

We now assume that Charlie is replaced by a company Charlie Inc. whose business is to store and distribute entangled states. This requires Charlie Inc. to have large-scale long-lived quantum memory and a means of distributing entangled states. Our protocol is detailed in Fig. 2 and is also described concisely in Algorithm 1. Charlie Inc., acting as a central node, first prepares Bell pairs in surface codes and teleports each half into separate qLDPC code blocks via a qLDPC (HGP) ancilla. This is done because direct computations and measurements for qLDPC codes are not yet well defined at the time of this writing. Charlie Inc. then keeps one set of neutral-atom encoded qLDPC blocks and sends the other set to an end user via physical transport, like a train or a truck. This process is done for several such end users at different destinations. This is called the “sneakernet” method. At the user’s end, the user purchases a certain number of bits based on their needs. At the user’s end, a quantum ATM (qATM) “unloads” the incoming qLDPC blocks by sequentially teleporting the constituent qubits onto a surface code, followed by measuring the surface code randomly in  $X$  or  $Z$  bases, and commercially selling the resulting classical bits to users, which is in turn stored in a device belonging to the user, such as a smartphone. For our analysis, we assume that the process of transferring classical bits onto the end users’ classical devices is secure. The measurement bases are then made public.

Among several end users in the network, let us say two of them, namely Alice and Bob, want to connect at a later time. They decide upon how many bits they need to correlate for their requirement. Charlie Inc. then converts the corresponding logical qubits in the qLDPC blocks on his end into surface code patches and performs Bell measurements on those patches, thereby correlating Alice’s and Bob’s bits. Finally, Alice and Bob use their correlated bits to generate a draft encryption key. Depending on their security requirement, they can enhance the key’s security using Clauser-Horne-Shimony-Holt tests [39] and privacy amplification, sacrificing some bits in the process. The remaining bits form a secure encryption key for end-to-end message encryption. This approach turns entanglement into a tradeable commodity and allows users with only classical hardware to participate in quantum cryptography protocols. For example, Alice measures a number of qubits, labeled  $A_1, A_2, \dots, A_n$ , and informs Charlie Inc. which qubit measurement results she now owns. Charlie Inc. then records the owner of the qubits and stores the other half of the Bell pairs in long-term quantum storage. Similarly, Bob buys the rights to measure qubits  $B_1, B_2, \dots, B_n$  and Dave buys the rights to measure qubits  $D_1, D_2, \dots, D_n$ . At this stage, neither participant has decided who they wish to exchange correlations with; they have simply “topped up” their future quantum potential, i.e., the bits with which they can perform an unconditionally secure one-time pad with another user of their choice in the future. If Alice and Dave wish to generate entangled data for some protocol, then they need to publicly declare to Charlie Inc. how many topped up bits

they want to have correlated, then Charlie Inc. performs the entanglement swap between the appropriate (stored) Bell pairs and broadcasts the measurement results. If Alice and Dave wish to check Charlie Inc.’s fidelity, they may sacrifice some of their bits, and assuming they are satisfied, can proceed with standard key distillation and privacy amplification [40]. If Alice then wishes to share secrets with Bob, then again, she need only declare to Charlie Inc. how many bits she wants to correlate (of course, she cannot use measurement results that have previously been used with Dave), and proceed as before.

### IV. RESOURCE ESTIMATION

In this section, we derive closed-form expressions for logical failure rates (LFRs), time scales, and logistics concerning the protocol. For our analysis, we consider citywide and districtwide distances, hence we will be considering trucks or train cars for transport of quantum memories. We consider both trucks and train cars to be equivalent in carrying capacity, transport speeds, etc., and may tend to use the terms interchangeably. Throughout this section, we consider the processes and operations occurring in the quantum error corrected qLDPC memories as they go through all the stages (loading, transport, and unloading) of the protocol.

For our protocol, we select the qLDPC codes constructed from the hypergraph product of classical codes based on  $(3, 4)$ -regular bipartite graphs with strong expansion properties [27]. These graphs are selected randomly, and the resulting quantum codes exhibit a minimum encoding rate of  $1/25$ . In [27], the authors discuss the implementation of computation with qLDPC codes using neutral atoms, which provide an order of magnitude improvement in the number of physical qubits used compared to surface codes. They introduce a technique by which logical qubits can be teleported between the surface codes and qLDPC code blocks, thus aiding in the possibility of computation and individual logical qubit measurements, which becomes essential for the implementation for our protocol. The authors also provide closed-form expressions for various parameters such as logical failure rates and cycle times, which we incorporate in our resource analysis.

#### A. Rearrangement of atoms

For a  $[n, k, d]$  HGP code with a coding rate  $r = k/n$  where  $n$  is the number of physical qubits,  $k$  is the number of logical qubits, and  $d$  is the code distance, each syndrome extraction cycle takes time  $8t_r$  [27], where the factor 8 represents the number of atomic rearrangements needed to complete one error correction cycle, and  $t_r$  is the time required to move the atomic qubits around for one rearrangement [27]. By overlapping the syndrome measurements with the atom repositioning of the next iteration, we can avoid extending the total execution time. We express  $t_r$  as a function of  $n$ , given by

$$t_r(n) \approx 2\tau_t \log L + (3 + 2\sqrt{2}) \sqrt{\frac{6Ld_p}{a_p}} \quad (1)$$

where  $\tau_t = 50 \mu\text{s}$  is the transfer time between the atomic traps,  $a_p = 0.02 \mu\text{m} \mu\text{s}^{-2}$  is the maximum acceleration rate of the atom,  $d_p = 5 \mu\text{m}$  is the spacing between the atoms in the qLDPC grid, and  $L \approx \sqrt{n}$  is the number of atoms in a line



to be rearranged. The gate times of neutral atom architectures are on the order of microseconds [41] while  $t_r$  is on the order of several milliseconds. Therefore  $t_r$  becomes the dominant rate-limiting timescale and the gate times can be ignored while computing neutral atom qLDPC cycle times.

### B. LFRs

The LFR  $R_L$  per error correction cycle for a  $[n, k = n/25, d = \sqrt{k}]$  HGP code is given by [27]

$$R_L(n) = 0.07(p_g/0.006)^{0.47n^{0.27}} \quad (2)$$

where  $p_g$  is the physical gate error and  $n$  is the number of physical qubits,  $k = n/25$  is the number of logical qubits, and  $d = \sqrt{k}$  is the code distance. Reference [27] considers HGP codes such that  $k = n/25$  and  $d = \sqrt{k} = \sqrt{n}/5$ . We consider this configuration as well in our analysis. Similarly, the LFR  $R_S$  per error correction cycle for a surface code of distance  $d_{SC}$  is given by [21]

$$R_S = 0.3(70p_g)^{\frac{d_{SC}+1}{2}}. \quad (3)$$

Let  $n_a$  be the number of physical qubits in the ancilla HGP code. Let  $n_t$  be the number of physical qubits in the qLDPC memory to be transported. Let  $n_c$  be the number of physical qubits in the qLDPC memory of Charlie Inc. We make two key assumptions in our analysis. First, we consider the number of physical qubits in Charlie Inc.'s qLDPC memory ( $n_c$ ) and the transported qLDPC memory ( $n_t$ ) to be equal, denoting both as  $n_m$ , where  $m$  stands for memory. Secondly, we consider the distance of the computational surface code ( $d_{SC}$ ) as a function of the qLDPC memory size, specifically  $d_{SC} = \sqrt{n_m}/5$ , as used by the analysis in [27].

Throughout our derivations, we use the following binomial approximation:

$$\begin{aligned} (1-p) &= (1-p_1)^{n_1}(1-p_2)^{n_2} \dots, \\ \Rightarrow p &\approx n_1 p_1 + n_2 p_2 + \dots, \\ \forall 0 < p_i < 1. \end{aligned} \quad (4)$$

This approximation is valid due to the logical failure rates being much lower than 1, due to which logical errors can be propagated via simple addition.

### C. Initialization of qLDPC code patches

First, Charlie Inc. needs to initialize two qLDPC patches, one (C) for his memory and the other (T) for transporting to the destination node. The LFR  $R_0$  for this is given by

$$R_0 = 2 \frac{\sqrt{n_m}}{5} R_L(n_m). \quad (5)$$

Here  $d = \sqrt{n_m}/5$  is the code distance, which is also the number of error correction cycles, and  $R_L(n_m)$  is the LFR per error correction cycle. We assume that these memories are initialized in parallel with the previous memory batch being loaded onto the truck. Therefore, we do not count it into our timescales.

### D. Bell pair creation

Next step, we need to create a Bell pair. Now, for creating a single Bell pair, we need two surface codes and  $2\frac{\sqrt{n_m}}{5}$  error correction cycles. The LFR  $R_1$  for this is given by [21,42]

$$R_1 = 2 \frac{\sqrt{n_m}}{5} R_S. \quad (6)$$

Since loading the current bell pair and creating the next bell pair are done simultaneously, we do not count this timescale.

### E. Loading of the Bell pair

For the loading of both halves of the surface coded Bell pair onto their respective qLDPC patches C and T, we need to teleport it from the surface code to the HGP patch via an ancilla HGP. The ancilla HGP is of the configuration  $[n_a = n_m/25, k = 1, d = \sqrt{n_a}]$ . For the teleportation, we need first to initialize the ancillas. Then we need to do two lattice surgery procedures, one merge-and-split between the surface code and the ancilla HGP patch, and another between the ancilla HGP and the qLDPC patches. The LFR for loading onto the transport qLDPC is given by

$$\begin{aligned} R_2 &= 2 \left[ \sqrt{n_a} R_L(n_a) + \frac{\sqrt{n_m}}{5} [R_S + R_L(n_a)] \right. \\ &\quad \left. + \frac{\sqrt{n_m}}{5} \left( R_L(n_a) + \frac{n_m}{25} R_L(n_m) \right) \right] \end{aligned} \quad (7)$$

where the first term is for the ancilla initialization involving  $d = \sqrt{n_a}$  rounds of error correction; the second term is for lattice surgery between surface code and ancilla involving  $d = \frac{\sqrt{n_m}}{5}$  error correction cycles, with LFR per cycle for the surface code being  $R_S$  and that of the ancilla HGP being  $R_L(n_a)$ ; and the third term is for that of ancilla and memory patch. The patch sizes are chosen in such a way that all three patches (memory, ancilla, and surface code) have the same code distance, and hence require the same number of error correction cycles. Here the  $\frac{n_m}{25}$  factor appears since the first loaded qubit will undergo these sets of loading QEC cycles  $\frac{n_m}{25}$  times, which propagates its error. The term 2 arises because we need to teleport two halves of the Bell pair, one to the transport memory T and the other one to Charlie Inc.'s memory C. The time  $T_2$  required for this is given by

$$T_2 = \frac{n_m}{25} \left( \sqrt{n_a} t_{cyc}(n_a) + \frac{\sqrt{n_m}}{5} t_{cyc}(n_a) + \frac{\sqrt{n_m}}{5} t_{cyc}(n_m) \right) \quad (8)$$

where  $t_{cyc} = 8t_r$  is the time taken to complete one error correction cycle. Here we combine parallel cycles, such as that happening simultaneously on surface code and ancilla, and on the ancilla and the qLDPC memory.

### F. Storage and transport

Let us say the destination node is  $T_3$  units of time drive from Charlie Inc. Throughout the transit,  $M_t = \frac{T_3}{T_{cyc}(n_m)}$  number of error correction cycles happen on the transport memory T. The LFR  $R_3$  occurring during the transport is given by

$$R_3 = \frac{T_3}{T_{cyc}(n_m)} R_L(n_m). \quad (9)$$

Here, the idling errors, which emerge as an adjustment to the value of  $p_g$ , have been incorporated into  $R_L$ , and a scenario has been considered where it is beneficial to perform QEC cycles as often as possible. This is explained in Sec. V under the topic of idling errors [Eq. (20)].

### G. Unload and measure at qATM

Once the transport memory arrives at a qATM in the destination node, we need to initialize the surface code and the ancilla, and then sequentially teleport the logical qubits from the qLDPC to the surface code via the ancilla, and measure the surface code qubit. The LFR for this is given by

$$R_4 = \frac{\sqrt{n_m}}{5} R_S + \sqrt{n_a} R_L(n_a) + \frac{\sqrt{n_m}}{5} [R_S + R_L(n_a)] + \frac{\sqrt{n_m}}{5} \left( R_L(n_a) + \frac{n_m}{25} R_L(n_m) \right) + R_S. \quad (10)$$

This is similar to Eq. (7), with an additional term  $R_S$  in the end which is the one-round LFR that occurs due to measurement. The time taken for this is

$$T_4 = \frac{n_m}{25} \left( 6 \frac{\sqrt{n_m}}{5} t_g + \sqrt{n_a} t_{\text{cyc}}(n_a) + \frac{\sqrt{n_m}}{5} t_{\text{cyc}}(n_a) + \frac{\sqrt{n_m}}{5} t_{\text{cyc}}(n_m) + 6 t_g \right). \quad (11)$$

Measuring a surface code logical qubit involves one round of error correction. The depth of a surface code syndrome extraction is 6 and the physical gate time of the architecture is given by  $t_g$ . Hence the last term.

Now, the total network LFR  $R_{\text{net}}$  of the processes so far and the total time  $T_{\text{tot}}$  taken for generating this correlation are

$$R_{\text{net}} = \sum_{j=0}^4 R_j, \quad T_{\text{tot}} = \sum_{j=2}^4 T_j. \quad (12)$$

### H. Unload and Bell measure at Charlie Inc.'s end

Now we have two such half-measured Bell pairs distributed between the network pairs Charlie Inc.–Alice and Charlie Inc.–Bob. So we have two memory drives, one each for each network pair. Let us say that at a later time  $T_{\text{later}}$ , Alice and Bob decide to correlate their bits. For calculation purposes, we assume that this decision has been made by the time  $T_{\text{later}} = T_{\text{tot}}$ . We assume that by this time, the end users Alice and Bob have decided to communicate with Charlie Inc. and have informed him accordingly. Let the total number of error correction cycles  $M_{\text{tot}_c}$  be done by Charlie Inc.'s memory drives during this time. This is given by  $M_{\text{tot}_c} = T_{\text{later}}/t_{\text{cyc}}(n_m)$ . Since we have two drives, the LFR  $R_{\text{store}}$  is given by

$$R_{\text{store}} = 2 M_{\text{tot}_c} R_L(n_m) = 2 \frac{T_{\text{tot}}}{t_{\text{cyc}}(n_m)} R_L(n_m). \quad (13)$$

The factor of 2 occurs because there are two memory drives. Now the qubits need to be teleported onto surface code via an

ancilla. The LFR  $R_{\text{unload}}$  of that is given by

$$R_{\text{unload}} = 2 \left[ \frac{\sqrt{n_m}}{5} R_S + \sqrt{n_a} R_L(n_a) + \frac{\sqrt{n_m}}{5} \left( R_L(n_a) + \frac{n_m}{25} R_L(n_m) \right) + \frac{\sqrt{n_m}}{5} [R_L(n_a) + R_S] \right]. \quad (14)$$

Now for Bell measurement, we require  $4 \frac{\sqrt{n_m}}{5}$  QEC cycles [43]. The LFR  $R_{\text{BM}}$  for this is given by

$$R_{\text{BM}} = 4 \frac{\sqrt{n_m}}{5} R_S. \quad (15)$$

This generates a correlation between Alice and Bob.

### I. Total LFR

So, therefore, the LFR  $R_{\text{tot}}$  of the overall final correlation between end users Alice and Bob is given by

$$R_{\text{tot}} = R_{\text{BM}} + R_{\text{unload}} + R_{\text{store}} + 2R_{\text{net}}. \quad (16)$$

And the success probability  $F_{\text{final}}$  of this correlation is given by

$$F_{\text{final}} = 1 - R_{\text{tot}}. \quad (17)$$

### J. Logistics

Logical bit rate  $r_L$  per qLDPC block is given by

$$r_L = \frac{n_m}{25 T_4}. \quad (18)$$

This is the number of logical qubits from the qLDPC block unloaded and measured per unit time. It takes  $T_4$  amount of time to unload one entire qLDPC patch consisting of  $\frac{n_m}{25}$  logical qubits.

Let  $r_e$  be the desired target entangled bit (E-bit) rate. Then, the number  $N$  of parallel qLDPC blocks required to achieve  $r_e$  is given by  $N = r_e/r_L$ . The blocks get depleted every  $T_4$  units of time. Therefore, we need to replace  $N$  blocks every  $T_4$  units of time. Let us say a truck or a train car has the carrying capacity of  $B$  blocks, where  $B = n_t/n_m$  where  $n_t$  is the total number of physical qubits the truck or train car can carry. We need  $N/B$  trucks to be released by Charlie Inc. every  $T_4$  units of time, for one qATM destination. If we are assuming  $S$  number of qATM destinations, then the number of trucks  $n_{\text{truck}}$  released by Charlie Inc. every  $T_4$  units of time is given by  $n_{\text{truck}} = SN/B$ .

The life cycle of a truck involves  $T_2$  units of time to load the qLDPC patches,  $T_3$  time to transit,  $T_4$  time to unload and get the logical qubits measured, and  $T_3$  again to move back to Charlie Inc.'s hub for the next round. So, the total number of trucks  $N_{\text{truck}_{\text{tot}}}$  required for the whole network to

run continuously is given by

$$N_{\text{truck}_{\text{tot}}} = \frac{SN}{(n_t/n_m)} \frac{T_2 + 2T_3 + T_4}{T_4}. \quad (19)$$

## V. RESULTS AND DISCUSSION

The required E-bit rate ( $r_e$ ), or bandwidth, is a critical parameter in our system design. Contemporary communication standards provide context for our bandwidth requirements. For instance, voice communications necessitate 87 kbps, standard definition video conferencing requires 128 kbps, and high-definition video conferencing demands up to 4 Mbps [44]. In the realm of text messaging, an average SMS of 120–160 characters consumes approximately 16 000 bits [45], which we consider for our analysis. Note that this is somewhat contrived, but it gives us an anchor to do our analysis. To determine  $r_e$ , we consider several factors: the frequency of one-time pad recharging, the practical time a user might wait at a qATM while recharging ( $t_{\text{wait}}$ ), and the average number of bits needed between refills ( $b_{\text{req}}$ ). The number of end users at a destination ( $n_{\text{users}}$ ) is constrained by the number of available time slots in a day, calculated as 86 400 s divided by  $t_{\text{wait}}$ . Consequently, we express  $r_e$  as  $b_{\text{req}}/t_{\text{wait}}$ .

We estimate  $r_e$  for SMS usage as an illustrative example. Assuming an average person sends 85 texts daily [46], each requiring 16 000 bits, we calculate  $b_{\text{req}}$  as  $85 \times 16\,000 = 1\,360\,000$  bits. With a  $t_{\text{wait}}$  of 10 min (600 s) every day,  $r_e$  equals  $1\,360\,000/600 = 2.3$  kbps or 2.3 kHz.

The gate error rate  $p_g$  for neutral atom architectures varies from 0.001 to 0.005. Quantinuum, a quantum computing company, has demonstrated a one-qubit gate fidelity of 0.999 979 and a two-qubit gate fidelity of 0.999 14 [47]. Therefore it would be reasonable to consider  $p_g = 1 - 0.9992 = 0.0008$  as a near-term realizable optimistic estimate.

### A. Idling errors

Idling errors can be modeled by adjusting the two-qubit gate error rate  $p_g$  as follows [27]:

$$p_g \rightarrow p_g \left( 1 + \frac{3t_r(n)}{0.005T_c} \right) \quad (20)$$

where  $T_c = 10$  s is the coherence time of neutral atom qubits. If we plug in a worst-case scenario of a patch size of 100 000, we see that  $p_g$  gets scaled up by a factor of 1.2754, i.e., by 27.54%. Considering our original estimate of  $p_g = 0.0008$ , the new  $p_g$  becomes  $p_g = 0.0008 \times 1.2754 = 0.001$ , which we use for our analysis.

Unlike most other architectures, neutral atoms have significantly slower one-qubit gates than two-qubit gates. Hence we take the one-qubit gate time  $t_g = 2 \mu\text{s}$  [41]. Considering the form factor of Pasqal's processor [48] and their claim of producing a  $10^4$  qubit unit by 2026, we take a rough estimate for our calculations that any truck or train car in the near future can carry a total of  $n_t = 1\,000\,000$  physical qubits. We also set the number of qATM destinations  $S = 5$ . In this paper, we assume intrametropolitan distances, with typical transport times ranging from 1 to 3 h. We take the teleportation ancilla HGP patch size  $n_a = \frac{n_m}{25}$  because it maintains the same distance as

the qLDPC code and the surface code, and its edge is as wide as the size of the logical operators in the qLDPC code [27].

From Fig. 3, we can see that to achieve a lower total logical error rate  $R_{\text{tot}}$ , we need larger qLDPC patches. Larger patches have greater code distance, hence higher tolerance to errors. To achieve greater than 80% fidelity (i.e.,  $R_{\text{tot}} < 0.2$ ), we need patch sizes of at least about 35 000 physical qubits (1400 logical qubits). Patch sizes over 53 000 physical qubits (2120 logical qubits) can achieve fidelities of over 90% while 80 000 qubit-sized patches (3200 logical qubits) can give greater than 95% fidelity. The higher the fidelity, the lower the number of bits wasted for standard reconciliation and privacy amplification, and with fidelities over 95%, this number becomes negligible [49]. Pasqal, an enterprise dealing with neutral atom quantum architectures, claims that devices with a capacity of 10 000 physical qubits will be a reality in the next couple of years [48], so it would be reasonable to assume larger patch sizes of the order of high  $10^4$  physical qubits to be achievable in the next few years.

From Fig. 4, we can observe that the protocol can tolerate longer transport times  $T_3$  with larger patch sizes. This is because larger patch sizes have greater logical error tolerance, and hence can be transported to longer distances. It can also be observed that the plot truncates below a certain patch size when the value of  $T_3$  comes down to zero. This is because, for any  $n_m$  below that patch size, all terms in  $R_{\text{tot}}$  except the  $R_3$  term push the  $R_{\text{tot}}$  greater than the target error rate. Due to this, the solver function in our plot code plugs in a negative  $T_3$  to pull down the  $R_{\text{tot}}$  to match the target. This means that the target  $R_{\text{tot}}$  is not achievable for patch sizes below this cutoff size. From both protocols, we see that the error rate  $R_{\text{tot}}$  increases with an increase in transport time due to a greater number of error correction cycles.

From Fig. 5, we can see that for a given transport time (90 min here), the number of trucks or train cars required reduces with an increase in the tolerable error rate  $R_{\text{tot}}$  of the network. This is because the greater the tolerable error rate  $R_{\text{tot}}$ , the smaller the patch size. Smaller patch sizes are of lower code distances and hence require fewer rounds of error correction. This translates to faster unloading times, requiring a lesser number of patches in parallel, and ultimately requiring a lower number of trucks. For fidelities below 95% (i.e.,  $R_{\text{tot}} > 0.05$ ), we can run this network with under 10 000 trucks for 2300-Hz bandwidth. This is comparable to the total number of buses or train cars in an average city public transport.

Figure 6 compares the qLDPC network with the corresponding surface code network, in terms of number of trucks, and additionally, Table I also lists the operational unit economics  $C_o$  in terms of the cost per bit, which is a combination of transportation cost  $C_t$  and the cost  $C_q$  of maintaining the quantum infrastructure, i.e.,

$$C_o = C_t + C_q. \quad (21)$$

The transport cost  $C_t$  is calculated as follows:

$$C_t = \frac{R_h N_{\text{truck}_{\text{tot}}}}{3600 r_e S} \quad (22)$$

where  $R_h$  is the hourly rent of one truck, which we assume to be USD \$150;  $N_{\text{truck}_{\text{tot}}}$  is as described in Eq. (19);  $r_e$  is the E-bit rate or the bandwidth of the network; and  $S$  is the number

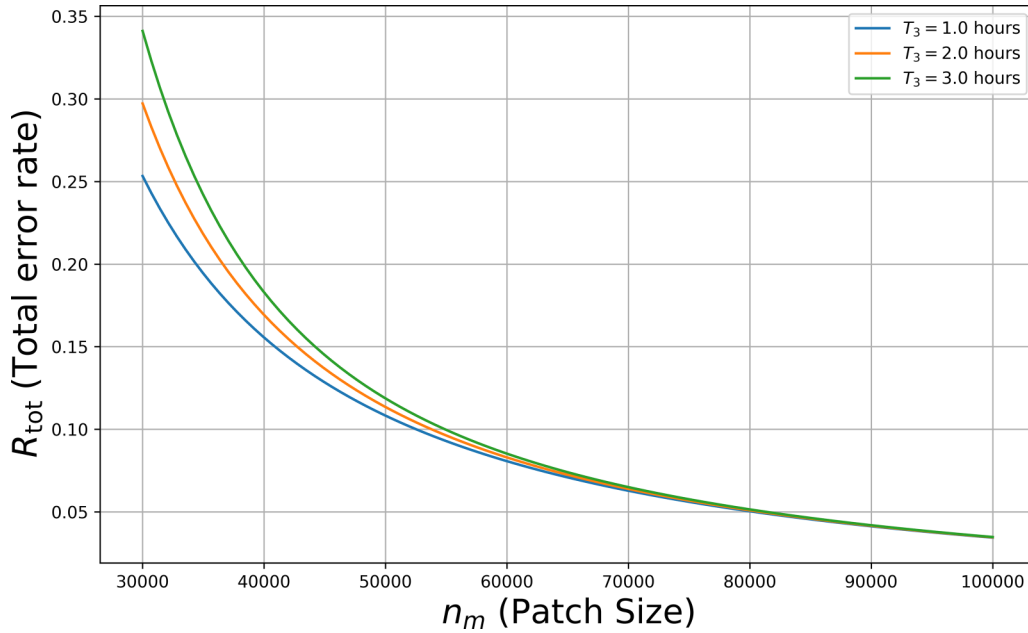


FIG. 3. The overall logical error rate  $R_{\text{tot}}$  plotted against qLDPC patch size  $n_m$  for various transport times  $T_3$ , showing an inverse relationship between the total logical error rate  $R_{\text{tot}}$  and the size of qLDPC patches. To obtain fidelity levels exceeding 80% (corresponding to  $R_{\text{tot}} < 0.2$ ), patch sizes of approximately 35 000 qubits or more are necessary. Further improvements in fidelity can be achieved with larger patches: those containing over 53 000 qubits can reach fidelities surpassing 90%, while patches comprising 80 000 qubits or more are capable of attaining fidelity levels above 95%.

of network nodes or end stations. Next, the total cost  $C_q$  of maintaining the quantum infrastructure is given by

$$C_q = \frac{2 \frac{n_t}{n_m} N_{\text{truck}_{\text{tot}}} C_m}{r_e S \times 86\,400 \times 365} \quad (23)$$

where  $C_m$  is the yearly cost of maintaining a single qLDPC memory patch containing  $n_m$  physical qubits. The numerator represents the yearly cost of maintaining all the qLDPC memory patches in the network, where the factor 2 arises from the fact that we are dealing with Bell pairs, and  $n_t/n_m$

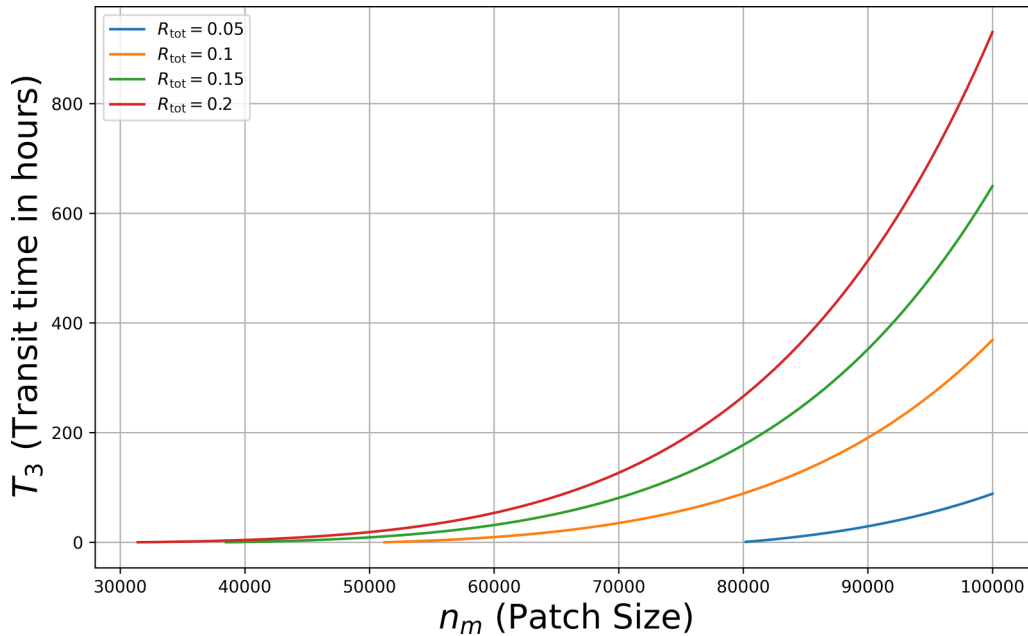


FIG. 4. Transport time  $T_3$  plotted against qLDPC patch size  $n_m$ , for various tolerable LFRs. The analysis demonstrates a positive correlation between patch size and the maximum tolerable transport time  $T_3$ . This relationship stems from the enhanced logical error tolerance of larger patches, which enables longer-distance transport. The plots exhibit a lower limit on patch size, at which the  $T_3$  value becomes zero. This occurs when the sum of all error terms except  $R_3$  exceeds the target error rate, regardless of  $T_3$ . In such cases, the numerical solver attempts to compensate by assigning negative  $T_3$  values, indicating that the target  $R_{\text{tot}}$  is unattainable for patches smaller than this threshold.



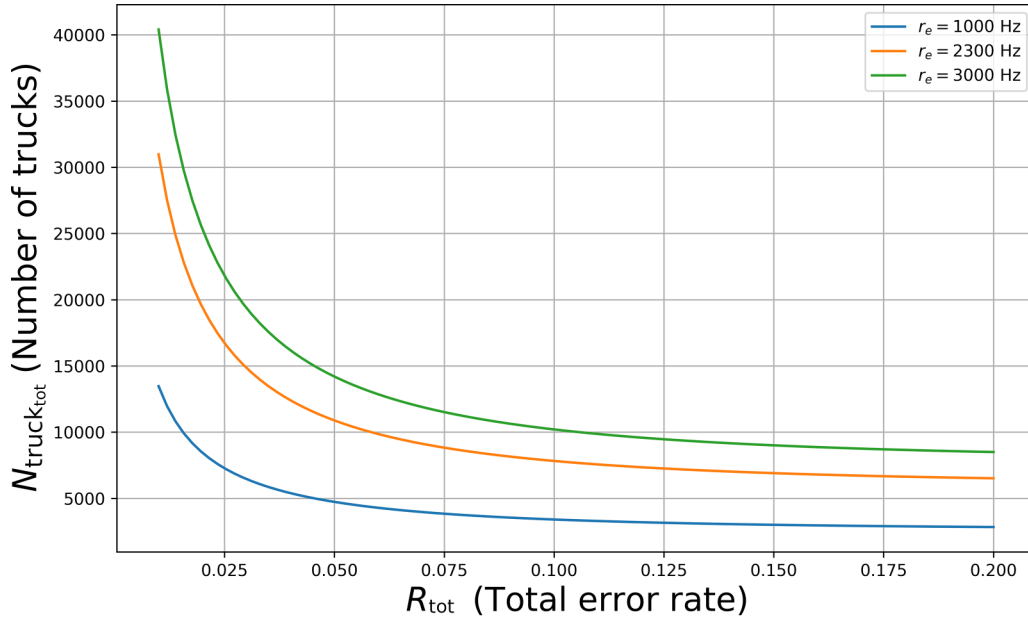


FIG. 5. Number  $N_{\text{truck}_{\text{tot}}}$  of trucks or train cars required for the network, as a function of overall LFR  $R_{\text{tot}}$ , showing an inverse relationship between the network's tolerable error rate  $R_{\text{tot}}$  and the required number of transport vehicles (trucks or train cars) for a fixed transit duration of 90 min. As the acceptable  $R_{\text{tot}}$  increases, the fidelity requirement decreases, and smaller patch sizes become viable. These reduced patch sizes correspond to lower code distances, necessitating fewer error correction rounds. Consequently, the unloading process becomes faster, reducing the number of parallel qLDPC patches needed. This cascade effect ultimately results in a decreased demand for transport vehicles to maintain the desired bandwidth.

is the number of quantum computer units on a truck. The denominator represents the number of bits emanated per year by the network. We consider an estimate of  $C_m = 2\,000\,000$  per year per quantum device unit [50]. For surface codes, the quantity  $n_m$  would represent the number of physical qubits

in a single quantum device, which would contain multiple surface code patches. We keep  $n_m$  the same for both qLDPC and surface codes.

The number of transport vehicles required by the qLDPC network is one order of magnitude lower than the number

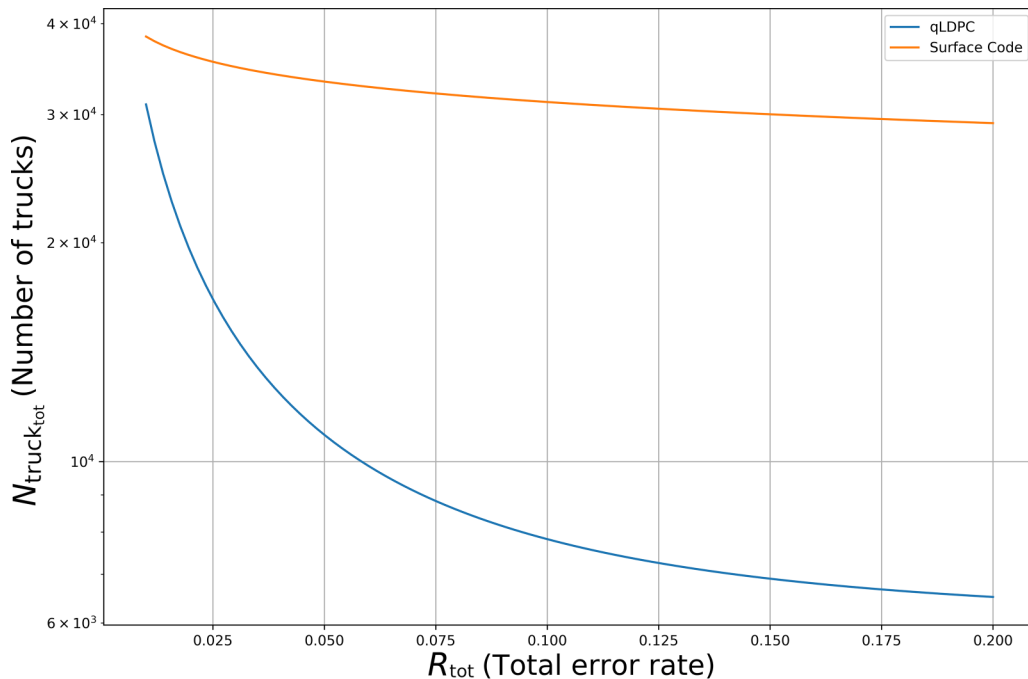


FIG. 6. Resource requirements (number of vehicles) for surface code vs qLDPC. The qLDPC network protocol demonstrates significantly higher efficiency in transport resource utilization compared to the surface code protocol, nearly by an order of magnitude.

TABLE I. Table showing the number of trucks required and the cost of each bit sold rounded to the nearest cent, for qLDPC-based protocol and a pure surface code based protocol. Here, we have taken the transit time  $T_3$  to be 90 min, network bandwidth to be 2.3 kHz, hourly rent per truck as USD \$150, and the total number of end nodes to be 5. The cost is calculated as per Eq. (21).

QEC code	Number of trucks	Cost per bit
qLDPC	7444	USD \$1.40
Surface code	31920	USD \$5.99

of vehicles required by the surface code protocol, and so is the cost. For example, for a minimum fidelity of 0.92, patch size  $n_m = 60\,000$ , and a transit time of 90 min, the qLDPC network requires 7444 vehicles to run and costs  $C_o = \text{USD } \$1.40$  per bit, whereas an equivalent surface code requires 31 920 vehicles and costs  $C_o = \text{USD } \$5.99$  per bit, about four times more. This is due to the constant encoding rates of qLDPC codes regardless of the tolerable logical error rate, unlike surface codes whose encoding rate worsens with the required tolerable logical error rate.

Nevertheless, even with qLDPC codes, the cost of \$1.40 per qubit would mean an average of \$22 400 per text message at 16 000 bits per message, which is extremely expensive, thus making the protocol in its current form commercially infeasible. High number of trucks and consequently high costs per qubit are both mainly attributable to the fact that neutral atom architectures have their physical qubits sparsely spaced in the lattice, and their gate execution times are relatively slow. If we can implement qLDPC codes on faster, more condensed architectures in the future, such as the silicon quantum dot qubits for example [51], we can potentially reduce the number of trucks and consequently the cost per bit by about three orders of magnitude.

## VI. CONCLUSION

Our analysis shows that in the near term, it is possible to build a multinode delayed choice network, all separated by intrametropolitan distances (with transport times of 1–3 h), with a central Charlie Inc. who uses  $O(10^3)$  vehicles to transport and sell entanglement-based correlations to end users with classical hardware (such as a smartphone) at the end nodes. Assuming that Charlie Inc. has the requisite quantum memory and storage, and it is possible to distribute entanglement at a rate faster than the entanglement consumption, then many end users can top up their results on classical storage devices, and use them at their convenience, topping up as required. This commodifies entanglement, paving the way to large-scale commercial quantum networks catering to users with non-quantum hardware. The qLDPC codes, with their compact constant encoding rates, play a major role in establishing this feasibility.

In our analysis, we adopt certain conventions and assumptions based on recent research in quantum error correction codes. The relationship between the number of physical qubits in qLDPC memory and surface code is a key consideration. We assume that a qLDPC memory with  $n_m$  physical qubits is connected to a surface code with  $n_m/25$  physical qubits

for teleportation. This ratio is consistent with the simulations presented in Ref. [27], although it is not explicitly stated as a requirement. While it is theoretically possible to use surface codes of various sizes, we adhere to this ratio for consistency with existing literature.

Regarding the teleportation scheme for transferring logical qubits between qLDPC and surface codes, our analysis focuses on HGP codes. Reference [27] specifically defines this teleportation scheme for HGP codes, but not for lifted product (LP) codes. While it may be possible to extend this scheme to LP codes, such an extension would require additional assumptions about the ability to generate logical  $X$  operators at the edge of the patch. To maintain rigor and avoid unsubstantiated claims, we restrict our analysis to HGP codes, for which the teleportation scheme is well defined. This decision ensures that our results are based on established protocols and avoids potential inaccuracies that could arise from extrapolating beyond the current state of knowledge in the field. Designing a teleportation protocol for these other non-HGP qLDPC code families can reduce patch sizes by a factor of 2 to 4, reducing our resource requirements.

Another caveat is that of classically transferring measurement results onto the users' smartphones. In future research, if a protocol can be devised to teleport the logical qubits of a quantum error correcting code patch onto a single photon, then photon measurement devices can be built into smartphones, which makes the transfer quantum secure. An alternate way, as outlined in the Algorithm 2, is to perform a heralded qubit transfer from each atomic physical qubit on the surface code to a photon [52,53] and measure those photons on the end user's smartphone [31] in a predefined sequence and decode to extract the logical surface code measurement, but this method has only 0.88 fidelity and 0.69 efficiency [52], which when propagated over all the atoms of a single surface code would lead to high failure rates and high resource overheads. Therefore, development of a more efficient method of performing atom-to-photon qubit transfer would be beneficial towards a highly secure network with minimal resource overhead.

With this kind of network, one can easily imagine creating new symmetric keys between multiple users, for example, a consumer who wishes to secretly share her banking details with a large number of different vendors, all via guaranteed entanglement-generated quantum keys. We have so far considered a single Charlie Inc., with one long-term mass storage and qATMs distributed in the field. However, as long as Charlie Inc. has set up preexisting and refreshed entanglement, there is no reason why it is not possible for Charlie Inc. to have multiple long-term quantum memory sites in geographically distinct locations, which we term quantum hubs. With a constant refresh of entanglement between Charlie Inc.'s hubs, in principle, global scale links could be achieved with additional entanglement-swapping operations.

## APPENDIX: SURFACE CODE BASED DELAYED CHOICE NETWORK

For comparison, we design a surface code based delayed choice quantum network. Any variable not explained here has been adopted from the main paper.

## ALGORITHM 2. Secure delayed choice quantum network protocol with photonic measurements.

---

```

1: for all end users do
2:   Charlie Inc. prepares Bell pairs encoded in surface codes
3:   Charlie Inc. loads each half of the Bell pairs into separate sets of qLDPC code blocks via teleportation
4:   Charlie Inc. stores one set of qLDPC blocks and transports the other set to destination
5:   End user puts in a purchase order for  $n_{\text{bits}}$  number of bits as per their requirement
6:   for  $i$  from 1 to  $n_{\text{bits}}$  do
7:     qATM at the destination unloads a logical qubit from a qLDPC block onto a surface code
8:     qATM converts surface code physical qubits to photons in a pre-defined order via pulsed excitation and de-excitation
9:     Photons sequentially sent to end user's smartphone and measured
10:    Photonic measurements determine the logical surface code measurement, leading to a resultant measured bit
11:    Bit stored in the end user's smartphone for later use.
12:  end for
13: end for
14: Publish measurement bases
15: for users Alice and Bob wishing to connect do
16:   Alice and Bob decide to correlate  $n_{\text{key}}$  number of bits
17:   for  $i$  from 1 to  $n_{\text{key}}$  do
18:     Charlie Inc. unloads two corresponding qLDPC qubits onto two surface code patches
19:     Charlie Inc. performs Bell measurement on surface code patches
20:     Charlie Inc. broadcasts the measurement result to Alice and Bob, resulting in a correlated key bit
21:   end for
22:   Alice and Bob perform standard reconciliation techniques and get the final secure key
23: end for

```

---

In this network, there is no teleportation involved between a separate storage code and an input code since we are dealing with surface codes directly. Hence, the storage-and-transport code is the same as the input/output code.

Let  $n_{ms}$  be the number of qubits in the surface code patch. Its distance will be  $\sqrt{n_{ms}}$ . Surface code LFR  $R_{ss}$  per cycle is given by

$$R_{ss} = 0.3(70p_g)^{\frac{\sqrt{n_{ms}}+1}{2}}. \quad (\text{A1})$$

### 1. Bell pair creation

First, Charlie Inc. creates Bell pairs. This involves  $2\sqrt{n_{ms}}$  error correction cycles. Its LFR is given by

$$R_{1s} = 2\sqrt{n_{ms}}R_{ss}. \quad (\text{A2})$$

The time taken for this is computed as

$$T_{1s} = 6t_g\sqrt{n_{ms}}. \quad (\text{A3})$$

For every Bell pair Charlie Inc. creates, he keeps the surface code patch containing one half of the Bell pair to himself, while he transports the other half to an end node.

### 2. Transport

During transport, the surface code undergoes  $\frac{T_{3s}}{6t_g}$  cycles, where  $T_{3s}$  is the transport time. The LFR during this time is given by

$$R_{3s} = \frac{T_{3s}}{6t_g}R_{ss}. \quad (\text{A4})$$

### 3. Measurement

The surface code patches are measured as soon as the truck arrives at the end-node qATM. The time taken to measure a

surface code patch is  $6t_g$ , and the measurement error is  $R_{ss}$ . Since each surface code has just one qubit, we can measure each of these surface codes together in parallel. Since our required bit rate is  $r_e$ , we can measure  $r_e$  number of surface codes every second in parallel to obtain the required bandwidth.

Now, we have the current net LFR after distribution:

$$R_{\text{net}} = R_{1s} + R_{3s} + R_{ss}. \quad (\text{A5})$$

And the total time taken for this is

$$T_{\text{tot}} = T_{1s} + T_{3s} + 6t_g. \quad (\text{A6})$$

### 4. Storage and measurement

Meanwhile, Charlie Inc., on his side, needs to store his halves of the Bell pairs, at least until the other halves are unloaded, i.e., until  $T_{\text{tot}}$  amount of time has passed. The LFR for this is given by

$$R_{\text{store}} = 2\frac{T_{\text{tot}}}{6t_g}R_{ss}. \quad (\text{A7})$$

The factor 2 comes from the fact that Charlie Inc. is storing two halves, each corresponding to that of Alice and Bob respectively.

Next, Charlie Inc. performs a Bell measurement between the two surface code patches. The LFR for this is given by

$$R_{\text{BM}} = 4\sqrt{n_{ms}}R_{ss}. \quad (\text{A8})$$

### 5. Overall

The total LFR for the whole protocol is

$$R_{\text{tot}} = R_{\text{BM}} + R_{\text{store}} + 2R_{\text{net}}. \quad (\text{A9})$$

## 6. Logistics

Since each logical qubit can be measured in parallel, the number of patches we need per second is equal to the desired bandwidth  $r_e$ . Considering  $S$  number of end nodes and trucks or train cars with capacity of  $n_r$  physical qubits, the number of trucks per second is given as  $N_{\text{truck}} = \frac{Sr_e}{(\frac{n_r}{n_{ms}})}$ . The life cycle

of a truck is the total time  $T_{\text{tot}}$  from loading to unloading plus the time  $T_{3s}$  needed to go back. Therefore, the total number of trucks required for the network is given by

$$N_{\text{truck}_{\text{tot}_s}} = \frac{Sr_e}{\left(\frac{n_r}{n_{ms}}\right)} (T_{\text{tot}} + T_{3s}). \quad (\text{A10})$$

- 
- [1] J. P. Dowling and G. J. Milburn, Quantum technology: The second quantum revolution, *Phil. Trans. R. Soc. A* **361**, 1655 (2003).
  - [2] I. H. Deutsch, Harnessing the power of the second quantum revolution, *PRX Quantum* **1**, 020101 (2020).
  - [3] L. Jaeger, The second quantum revolution, in *From Entanglement to Quantum Computing and Other Super-Technologies* (Copernicus, 2018).
  - [4] M. Atzori and R. Sessoli, The second quantum revolution: Role and challenges of molecular chemistry, *J. Am. Chem. Soc.* **141**, 11339 (2019).
  - [5] P. P. Rohde, *The Quantum Internet: The Second Quantum Revolution* (Cambridge University, New York, 2021).
  - [6] H. Riel, Quantum computing technology, in *2021 IEEE International Electron Devices Meeting (IEDM)* (IEEE, New York, 2021), pp. 1–3.
  - [7] J. Gambetta, IBM's roadmap for scaling quantum technology, IBM Research Blog (September 2020).
  - [8] C. Gardiner and P. Zoller, *The Quantum World of Ultra-Cold Atoms and Light Book II: The Physics of Quantum-Optical Devices* (World Scientific, Singapore, 2015), Vol. 4.
  - [9] J. L. O'Brien, Optical quantum computing, *Science* **318**, 1567 (2007).
  - [10] M. Körber, O. Morin, S. Langenfeld, A. Neuzner, S. Ritter, and G. Rempe, Decoherence-protected memory for a single-photon qubit, *Nat. Photon.* **12**, 18 (2018).
  - [11] S. Perseguers, G. Lapeyre Jr., D. Cavalcanti, M. Lewenstein, and A. Acín, Distribution of entanglement in large-scale quantum networks, *Rep. Prog. Phys.* **76**, 096001 (2013).
  - [12] P. P. Rohde, *The Quantum Internet: The Second Quantum Revolution* (Cambridge University, New York, 2021).
  - [13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Ed. (Cambridge University Press, Cambridge, 2020).
  - [14] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, Quantum entanglement, *Rev. Mod. Phys.* **81**, 865 (2009).
  - [15] S. Wehner, D. Elkouss, and R. Hanson, Quantum internet: A vision for the road ahead, *Science* **362**, eaam9288 (2018).
  - [16] H. J. Kimble, The quantum internet, *Nature (London)* **453**, 1023 (2008).
  - [17] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, Quantum internet: Networking challenges in distributed quantum computing, *IEEE Network* **34**, 137 (2019).
  - [18] A. Sharma and A. Kumar, A survey on quantum key distribution, in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)* (IEEE, New York, 2019), Vol. 1, pp. 1–4.
  - [19] A. Peres, Delayed choice for entanglement swapping, *J. Mod. Opt.* **47**, 139 (2000).
  - [20] A. K. Ekert, Quantum cryptography based on Bell's theorem, *Phys. Rev. Lett.* **67**, 661 (1991).
  - [21] S. J. Devitt, A. D. Greentree, A. M. Stephens, and R. Van Meter, High-speed quantum networking by ship, *Sci. Rep.* **6**, 36163 (2016).
  - [22] M. Zhong, M. P. Hedges, R. L. Ahlefeldt, J. G. Bartholomew, S. E. Beavan, S. M. Wittig, J. J. Longdell, and M. J. Sellars, Optically addressable nuclear spins in a solid with a six-hour coherence time, *Nature (London)* **517**, 177 (2015).
  - [23] J.-P. Tillich and G. Zémor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, *IEEE Trans. Inf. Theory* **60**, 1193 (2013).
  - [24] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
  - [25] D. A. Lidar and T. A. Brun, *Quantum Error Correction* (Cambridge University, New York, 2013).
  - [26] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, *Quantum* **4**, 327 (2020).
  - [27] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, *Nat. Phys.* **20**, 1084 (2024).
  - [28] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).
  - [29] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, *Nature (London)* **627**, 778 (2024).
  - [30] Q. Zhang, F. Xu, Y.-A. Chen, C.-Z. Peng, and J.-W. Pan, Large scale quantum key distribution: Challenges and solutions, *Opt. Express* **26**, 24260 (2018).
  - [31] D. Lowndes, S. Frick, A. Hart, and J. Rarity, A low cost, short range quantum key distribution system, *EPJ Quantum Technol.* **8**, 15 (2021).
  - [32] J. Duligall, M. Godfrey, K. Harrison, W. Munro, and J. Rarity, Low cost and compact quantum key distribution, *New J. Phys.* **8**, 249 (2006).
  - [33] J. A. Vaccaro, J. Spring, and A. Cheffles, Quantum protocols for anonymous voting and surveying, *Phys. Rev. A* **75**, 012333 (2007).
  - [34] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum-enhanced positioning and clock synchronization, *Nature (London)* **412**, 417 (2001).
  - [35] E. Bagan, M. Baig, and R. Muñoz-Tapia, Quantum reverse engineering and reference-frame alignment without nonlocal correlations, *Phys. Rev. A* **70**, 030301(R) (2004).

- [36] Link to the PYTHON code of our analysis, <https://shorturl.at/QZ9v7>.
- [37] B. Coecke, The logic of entanglement, in *Horizons of the Mind. A Tribute to Prakash Panangaden: Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday* (Springer, New York, 2014), pp. 250–267.
- [38] H. Zbinden, J. Brendel, W. Tittel, and N. Gisin, Experimental test of relativistic quantum state collapse with moving reference frames, *J. Phys. A* **34**, 7103 (2001).
- [39] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, Proposed experiment to test local hidden-variable theories, *Phys. Rev. Lett.* **23**, 880 (1969).
- [40] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, Generalized privacy amplification, *IEEE Trans. Inf. Theory* **41**, 1915 (1995).
- [41] K. Wintersperger, F. Dommert, T. Ehmer, A. Houshanov, J. Klepsch, W. Maurer, G. Reuber, T. Strohm, M. Yin, and S. Luber, Neutral atom quantum computing hardware: Performance and end-user perspective, *EPJ Quantum Technol.* **10**, 32 (2023).
- [42] H. Leone, S. Srikara, P. P. Rohde, and S. Devitt, Upper bounds for the clock speeds of fault-tolerant distributed quantum computation using satellites to supply entangled photon pairs, *Phys. Rev. Res.* **5**, 043302 (2023).
- [43] D. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, Surface code quantum computing by lattice surgery, *New J. Phys.* **14**, 123011 (2012).
- [44] Broadband communications options for public safety agencies, December 2012, [https://www.infrastructure.gov.au/sites/default/files/ibes\\_report\\_-\\_psa\\_broadband\\_options\\_final\\_0.pdf](https://www.infrastructure.gov.au/sites/default/files/ibes_report_-_psa_broadband_options_final_0.pdf), accessed August 14, 2024.
- [45] Bandwidth requirement, [https://ozekisms.com/p\\_2499-bandwidth-requirement.html](https://ozekisms.com/p_2499-bandwidth-requirement.html), accessed August 14, 2024.
- [46] Texting statistics, <https://99firms.com/blog/texting-statistics/>, accessed August 14, 2024.
- [47] C. Easttom, Trapped ion quantum computing, in *Hardware for Quantum Computing* (Springer, New York, 2024), pp. 1–16.
- [48] Pasqal issues roadmap to 10,000 qubits in 2026 and fault tolerance in 2028, <https://shorturl.at/VU4R1>, accessed August 13, 2024.
- [49] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, Experimental quantum cryptography, *J. Cryptology* **5**, 3 (1992).
- [50] Costs entailing quantum computers, <https://quantumzeitgeist.com/how-much-do-quantum-computers-cost/>, accessed August 13, 2024.
- [51] M. Eriksson, S. Coppersmith, and M. Lagally, Semiconductor quantum dot qubits, *MRS Bull.* **38**, 794 (2013).
- [52] N. Kalb, A. Reiserer, S. Ritter, and G. Rempe, Heralded storage of a photonic quantum bit in a single atom, *Phys. Rev. Lett.* **114**, 220501 (2015).
- [53] J. P. Covey, H. Weinfurter, and H. Bernien, Quantum networks with neutral atom processing nodes, *npj Quantum Inf.* **9**, 90 (2023).