Distributed Quantum Walk Control Plane Implementation

André Ribeiro[1], Matheus Guedes de Andrade[2], Fabio Kon[1], and Don Towsley[2]

[1]Departamento de Ciência da Computação
Universidade de São Paulo (USP)

[2]Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract. *The Quantum Walk Control Protocol (QWCP) enables universal distributed quantum computing in quantum networks. In its debut, QWCP was specified in terms of logical quantum operations required by a quantum network to implement the protocol. In this paper, we propose two possible implementations of the QWCP based on different ways to encode quantum walks in physical qubits. The proposed encodings require different numbers of qubits and remote entangled states to perform control operations. Our approaches help to bridge the gap between the logical description of the QCWP and possible physical realizations of the protocol.* [1]

1. Introduction

Quantum networks are on the frontier of quantum research. The field combines various disciplines with the promise of enabling quantum communication among quantum processors, leading to applications that cannot be achieved with classical communication alone [Kimble 2008]. Similar to traditional computer networks, quantum networks will require efficient communication protocols to enable scalable quantum communication. In addition to orchestrating network control operations, quantum network protocols must be tailored to overcome the inherent fragility of quantum information. Qubits are prone to errors from a variety of sources, such as memory decoherence in storage and imperfect state transduction in photonic transmission.

In this context, distributed quantum computing (DQC) is a fundamental application enabled by quantum networks [Buhrman and Röhrig 2003]. It requires control protocols to orchestrate distributed quantum operations despite the choice of quantum hardware used in the network fabric. Previous work proposed the Quantum Walk Control Protocol (QWCP) [de Andrade et al. 2023] to enable universal DQC in quantum networks. The QWCP was described in terms of logical operators and can be used to control arbitrary quantum network operations, e.g., entanglement distribution [Pant et al. 2019]. The logical description of QWCP is general, although it does not directly address how the protocol is implemented in terms of qubits.

In this paper, we address the problem of implementing the QWCP in quantum processors. We provide two ways one can map the logical description of the QWCP to

[1]This research was supported in part by the NSF grant CNS-1955744, NSF- ERC Center for Quantum Networks grant EEC-1941583, INCT of the Future Internet for Smart Cities CNPq proc. 465446/2014-0, and FAPESP procs. 14/50937-1 and 2020/04031-1.

qubits in the network nodes: the linear encoding and the logarithmic encoding implementation. We describe the number of qubits required at each node in order to implement the QWCP, together with a description of the control operators for both proposed methods. Finally, we present an initial theoretical analysis on the performance of the proposed implementations, which shows a trade-off between network resources used for control and the time required to complete control operations.

2. System Model

We follow the system model described in [de Andrade et al. 2023] for a quantum network. Consider a symmetric directed graph $G = (V, A)$, with V and A representing the nodes and arcs of the graph, respectively. Since G is symmetric, $(v, u) \in A$ if, and only if, $(u, v) \in A$. We use $\delta(v)$ to denote the set of neighbors of vertex $v \in V$ and $d(v) = |\delta(v)|$ to denote v's degree.

Edge colorings are of particular interest to the description and analysis of the methods we propose in this article. Let $G' = (V, E)$ denote the undirected version of the network graph G where arcs (u, v) and (v, u) of G are represented by the same edge in G'. Let $\Gamma'_{G,k} : E \to \{1, \ldots, |V|\}^{|E|}$ denote an edge coloring of G' where at most k edges incident to the same node have the same color. Let $\Gamma_{G,k} : A \to \{1, \ldots, |V|\}$ be a coloring of the directed graph G obtained from $\Gamma'_{G,k}$ as $\Gamma_{G,k}(v, u) = \Gamma_{G,k}(u, v) = \Gamma'_{G,k}(v, u)$. Let $\Delta_{\Gamma_{G,k}}$ denote the number of colors in $\Gamma_{G,k}$. When $k = 1$, coloring $\Gamma_{G,k}$ reduces to the usual coloring where edges incident to same node have different colors. Finding the minimum number of colors for G when $k = 1$, i.e., $\min_{\Gamma_{G,1}} \Delta_{\Gamma_{G,1}}$, is an NP-hard problem, although Vizings's theorem states that $d_{\max} \leq \min_{\Gamma_{G,1}} \Delta_{\Gamma_{G,1}} \leq d_{\max} + 1$ [Stiebitz et al. 2012]. For simplicity, we omit the dependency of a coloring with G from now on, and use Γ to denote $\Gamma_{G,1}$.

A quantum network is a set of quantum processors (nodes) that can communicate with each other via quantum channels (arcs). Each node has a fixed number of qubits and can perform local quantum operations on them. We divide those qubits into two disjoint sets, N_v and M_v, which we refer to as network qubits and data qubits, respectively. In this work, we mainly focus on network qubits—which are associated with control operations—and omit data qubits to simplify exposition when possible.

The discrete-time coined quantum walk model is a unitary evolution process defined in the Hilbert space $\mathcal{H}_G = \mathcal{H}_V \otimes \mathcal{H}_C$ that encodes the graph's arcs [Portugal 2018]. In particular, $\mathcal{H}_V$ encodes the vertices and $\mathcal{H}_C$ encodes the coin space of the walker. Specifically, for each $(v, u) \in A$, a basis vector $|v, c_{vu}\rangle$ is defined within $\mathcal{H}_G$, with c_{vu} symbolizing the degree of freedom corresponding to the arc that connects v with u. The evolution of the quantum walk is given by $|\Psi(t + 1)\rangle = S(t)C(t) |\Psi(t)\rangle$, where C and S are the coin and shift operators, respectively. A generic coin operation on node v maps $\sum_{u\in\delta(v)} \alpha_{vu} |v, c_{vu}\rangle \to \sum_{u\in\delta(v)} \beta_{vu} |v, c_{vu}\rangle$, where $\alpha_{vu} \in \mathbb{C}$ and $\beta_{vu} \in \mathbb{C}$ respect unitary evolution. C mixes the amplitudes associated with the states representing the outward arcs of a node and shapes the probability that the walk moves from one node to another. The shift operator S maps $|v, c_{vu}\rangle \to |u, c_{uv}\rangle$. It changes amplitudes among different nodes and is related to the propagation of the quantum walk itself, i.e., S drives the movement of the quantum walk among network neighbors.

The QWCP utilizes a quantum walk system to propagate entanglement between network nodes enabling remotely controlled quantum operations. A diverse set of operations can be performed with the QWCP and we focus on the case where the protocol is used to route quantum control information through a network path $P = \{v_0, v_1, \ldots, v_n\}$. In a nutshell, the QWCP starts in the state $|v_0, c_{v_0}\rangle \otimes (\alpha|0\rangle + \beta|1\rangle)$, where the second term denotes the state of a data qubit in B. The first coin operator coin operator $C(0)$ prepares the state $(\alpha|v_0, c_{v_0}\rangle|0\rangle + \beta|A, c_{v_0v_1}\rangle|1\rangle)$, which is an entangled state between control and data qubits in v_0. It progresses by evolving the control state through successive applications of coin and shift operators until the quantum walk state has the form $\alpha|v_0, c_{v_0}\rangle + \beta|v_n, c_{v_n}\rangle$. The coin operators applied in the quantum walk evolution after initialization are permutations of the degrees of freedom such that, for $t > 0$, $C(t) : |v_t, c_{v_tv_{t-1}}\rangle \to |v_t, c_{v_tv_{t+1}}\rangle$. Note that the coin operators that drive the evolution of the quantum walk are local operations while shifts are non-local. For simplicity, we focus on the description of $C(t)$ for $t > 0$, although one can extend the results described in this paper for the data-control operations described in [de Andrade et al. 2023].

3. Quantum Walk Implementation

We present two ways one can implement the quantum walk protocol with networking qubits. Our models add to previous literature by mapping the mathematical definition of the protocol to operations that can be performed on qubits within a quantum processor. In essence, one can think that the description presented in [de Andrade et al. 2023] is a logical description of a quantum algorithm and our models provide two implementations of the algorithm.

3.1. Linear Encoding

We start by assuming that one can use $O(d_{\text{max}})$ qubits in a node to encode the quantum walk, where d_{max} is the maximum node degree. This assumption allows each arc of the network graph G to be represented by one physical qubit in the network. The one-to-one mapping between arcs and qubits leads to a one-hot encoding of the quantum walk: each basis vector $|v, c_{vu}\rangle$ is represented by the quantum state where the qubit associated with arc (v, u) is in state $|1\rangle$ and all other qubits are in state $|0\rangle$. Each superposition of a set of arcs $A' \subset A$ in the one-hot encoding is an entangled state of the qubits representing the arcs in A' resembling a multi-qubit W-state [Dür et al. 2000]. We refer to the state of the qubit associated with arc (v, u) as $|\psi\rangle_{vu}$. Moreover, we refer to this one-hot encoding scheme as the *linear encoding*.

We now describe the action of the coin and shift operators in the linear encoding case. Let q_{vu} denote the qubit in node v that encodes arc (v, u). Let $|\psi\rangle = |1\rangle_{vw} \bigotimes_{a\in A\setminus\{vw\}} |0\rangle_a$ denote the linear encoding of an arbitrary basis state. For a generic coin operator, $C|\psi\rangle = \sum_{u\in\delta(v)} \alpha_{vu} |1\rangle_{vu} \bigotimes_{a\in A\setminus\{vu\}} |0\rangle_a$, where $\alpha_{vu} \in \mathbb{C}$ denotes the amplitude of the basis state associated with (v, u) in $|\psi\rangle$. When $t > 0$, $C(t)$ is a permutation of the states encoding the outgoing arcs of a node, implemented in the linear encoding as $C(t) = \text{SWAP}(q_{v(v-1)}, q_{v(v+1)})$.[2] The shift operator is also a swap gate in the linear encoding, although it involves qubits in different nodes. We implement S using the remote swap gate depicted in Figure 1, which requires the expenditure of two Bell pairs. Using

[2]Note that coins performing arbitrary permutations of the outgoing arcs of a node can be implemented in the linear encoding with a sequence of swap gates.

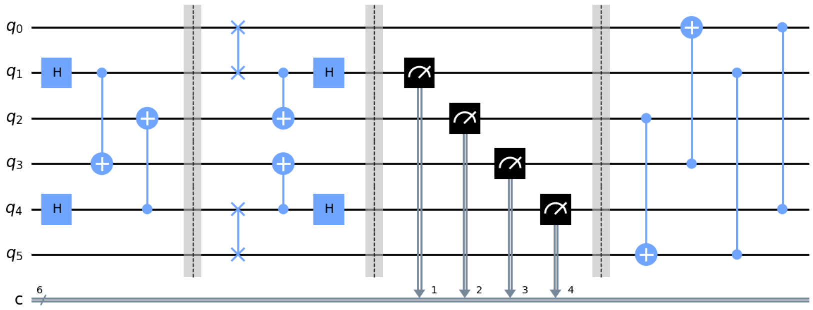
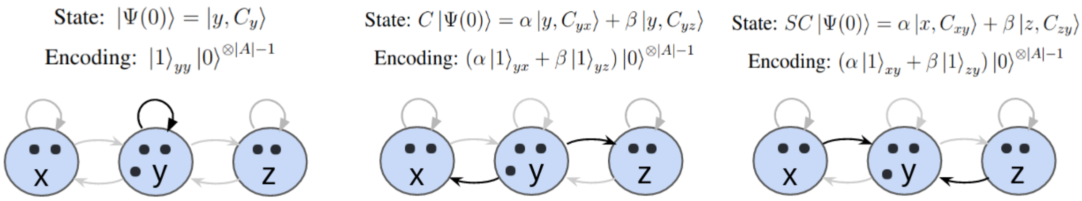


Figure 1. Swap circuit between q_0 and q_5, using two Bell pairs (q_1,q_3) and (q_2,q_4).



(a) Initial state. **(b) Coin operation.** **(c) Shift operation.**

Figure 2. Linear encoding system evolution with logical and code states.

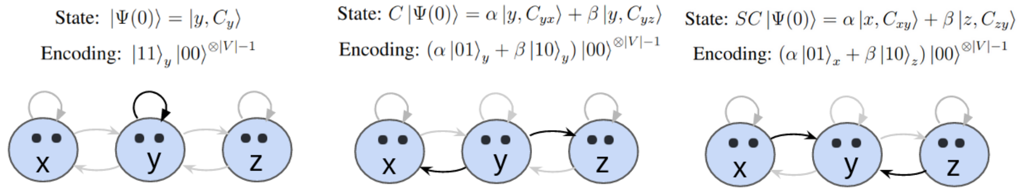
two Bell pairs to execute the remote swap gate is optimal, since this operation cannot be performed using a single Bell pair [Collins et al. 2001].

We illustrate the linear encoding through a three-node network example depicted in Figure 2. In this setup, three qubits in y encode the walker's state, while two encoding qubits in both x and z are utilized. Figure 2a shows the initial logical state when the quantum walk is initialized on the self-loop (y, y), together with the corresponding linear encoding. A coin operator that maps $|y, C_y\rangle$ into the generic superposition $\alpha |y, C_{yx}\rangle + \beta |x, C_{yz}\rangle$, with $\alpha, \beta \in \mathbb{C}$, is applied, generating the states depicted in Figure 2b. Finally, the shift operator propagates the quantum walk to nodes x and z, yielding the states depicted in Figure 2c.

3.2. Logarithmic Encoding

The linear encoding does not utilize the entire power of qubits to encode the quantum walk. We now present an encoding scheme that only requires $\lceil \log_2(\Delta_\Gamma + 1) \rceil$ qubits in each network node. unlike the linear encoding scheme, we do not provide a complete description of the operations for this encoding in terms of quantum gates. Instead, we describe them in terms of multi-qubit unitaries that can be implemented with any universal gate set.

Let $K = \lceil \log_2(\Delta_\Gamma + 1) \rceil$ and $|\psi\rangle_v$ denote the state of all K qubits in node v. The absence of the quantum walker in v is codified by the state $|\psi\rangle_v = |0 \ldots 0\rangle$. States are encoded following the coloring Γ such that $|v, c_{vu}\rangle$ is represented by the state $|\Psi\rangle_v = |\Gamma(v, u)\rangle_v$. Since arcs (u, v) and (v, u) have the same color, the state of the qubits in u encoding (u, v) is the same as the state of qubits in v encoding (v, u). Thus, a generic superposition $(\alpha |v, C_{vu}\rangle + \beta |w, c_{wl}\rangle)$ is encoded as the entangled state $(\alpha |\Gamma(v, u)\rangle_v |0 \ldots 0\rangle_w + \beta |0 \ldots 0\rangle_v |\Gamma(w, l)\rangle_w)$, for $(v, u), (w, l) \in A$.



(a) Initial state. **(b) Coin operation.** **(c) Shift operation.**

Figure 3. Logarithmic encoding system evolution with logical and code states.

A generic coin operator in v drives an evolution of the form as $\sum_{j=0}^{K-1} \alpha_j |j\rangle_v = \sum_{j=0}^{K-1} \beta_j |j\rangle_v$, where $\{\alpha_k\}$ and $\{\beta_k\}$ respect unitarity conditions. The coin operator driving the propagation of the quantum walk over path P is such that $C(t) : |\Gamma(v_t, v_{t-1})\rangle \to |\Gamma(v_t, v_{t+1})\rangle$. This operation can be performed through a sequence of swaps, where qubits q_k and q_j in v_t are swapped if the k-th bit of $\Gamma(v_t, v_{t-1})$ and j-th bit of $\Gamma(v_t, v_{t+1})$ differ. The shift operator maps state $|\Gamma(v, u)\rangle_v |0\dots0\rangle_u$ to state $|0\dots0\rangle_u |\Gamma(v, u)\rangle_v$. This encoding makes the shift operation considerably more complex than the linear encoding case. In the general case, the unitary that implements the shift between two nodes u and v acts on all qubits in both u and v. Furthermore, the most efficient way to implement S in terms of number of Bell pairs consumed that performs the shift operation in the logarithmic approach is not known. Nonetheless, a feasible way to implement the operator is to teleport all qubits from v to u, perform the unitary locally, and teleport the qubits back to v. This approach requires $2\lceil \log(\Delta_\Gamma) + 1 \rceil$ Bell pairs. As stated in Section 2, finding an optimal coloring for the encoding is an NP-Hard problem, although any edge coloring can be used to generate an encoding for the quantum walk.

We illustrate the logarithmic encoding in Figure 3 with the same three-node network exampled used for the linear encoding in Figure 2.

4. Performance analysis

Performing coin operations in both encodings described in Section 3 is simple from a networking perspective since they only require local operations in the nodes. Shift operations are more complex, since they require the generation of entanglement between neighboring nodes. We now analyse the performance of the two proposed encodings in terms of the network resources required to implement shift operations. We now assume that network nodes have additional network qubits that can be used to create Bell pairs with their neighbors in order to implement the remote gates required by shift operators. From now on, we divide the network qubits in the nodes into two groups: encoding and operation qubits.

Shift operations in both encodings are defined for a single link and performing the shift operation for the entire network—which we refer to as the network shift throughout the remainder of this paper—requires shifts on each network link. There is trade-off between the number of Bell pairs utilized to perform the network shift and the time required to complete the operation. This trade-off stems from the fact that the number of parallel shifts that can be executed at a given node depends on how many non-local Bell pairs that node can share with its neighbors at a given step.

Let a k-parallel shift denote shifting $k \geq 1$ links on a node simultaneously. Let a sequential shift denote shifting each link of a node one at a time, i.e., a 1-parallel shift. Since the time to execute a shift operation is negligible if a node has sufficient pre-shared Bell pairs, parallel node shifts are possible in both encodings. Executing a k-parallel shift in a node consumes kB_{enc} Bell pairs simultaneously, where B_{enc} is the number of Bell pairs consumed by a shift operation over a link in a given encoding. Note that $B_{\text{enc}} = 2$ in the linear encoding and $B_{\text{enc}} = 2\lceil \log(\Delta_\Gamma) + 1 \rceil$ in the logarithmic encoding. A sequential shift in node v incurs a time delay on the order of $O(d(v))$, while k-parallel shifts lead to a time delay of $O(d(v)/k)$. We discuss the time required to complete the network shift in terms of *shift rounds*, or simply *rounds*: a round is a maximal set of shift operations performed in different nodes such that no additional shift can be performed without additional Bell states. The number of rounds required to complete the network shift relates to edge coloring problems. In particular, the number of rounds is $\lceil \Delta_{\Gamma_k}/k \rceil$ when nodes perform k-parallel shifts. For instance, the network shift on an n-node star requires $n-1$ rounds if only sequential shifts are allowed.

5. Conclusion

In this paper, we described two different implementations of a mathematical quantum walk control plane [de Andrade et al. 2023] based on *(i)* a linear encoding and *(ii)* a logarithmic encoding. For both cases, we specified how to arrange the qubits in the network and how to perform the operations needed to implement the quantum walk control plane for universal distributed quantum computation. Finally, we analyzed and compared the proposed encoding schemes in terms of the trade-off between network resources and operation time. As future work, we plan to conduct simulations of the control plane. Our goal is to further investigate the performance of the two models proposed when effects such as classical communication latency and memory decoherence are taken into account.

References

Buhrman, H. and Röhrig, H. (2003). Distributed quantum computing. In *International Symposium on Mathematical Foundations of Computer Science*, pages 1–20. Springer.

Collins, D., Linden, N., and Popescu, S. (2001). Nonlocal content of quantum operations. *Phys. Rev. A*, 64:032302.

de Andrade, M. G., Panigrahy, N. K., Dai, W., Guha, S., and Towsley, D. (2023). Universal quantum walk control plane for quantum networks. *arXiv preprint arXiv:2307.06492*.

Dür, W., Vidal, G., and Cirac, J. I. (2000). Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(6):062314.

Kimble, H. J. (2008). The quantum internet. *Nature*, 453(7198):1023–1030.

Pant, M., Krovi, H., Towsley, D., Tassiulas, L., Jiang, L., Basu, P., Englund, D., and Guha, S. (2019). Routing entanglement in the quantum internet. *npj Quantum Information*, 5(1):25.

Portugal, R. (2018). *Quantum Walks, and Search Algorithms*. Springer Nature, Switzerland.

Stiebitz, M., Scheide, D., Toft, B., and Favrholdt, L. M. (2012). *Graph edge coloring: Vizing's theorem and Goldberg's conjecture*, volume 75. John Wiley & Sons.