Routing Costs for Distributed Inverse QFT Architectures

Jonathan R. Landers

May 12, 2026

A short extension note inspired by Cardama, Vázquez-Pérez, Pena, and Gómez [1].

Abstract

The inverse Quantum Fourier Transform (iQFT) is usually presented as a circuit-level object: a sequence of Hadamards and controlled phase rotations acting on a single logical register. In a distributed architecture, however, the same circuit acquires geometry. Some phase rotations are local inside a quantum processing unit (QPU); others must be carried through a network. Cardama et al. [1] exploit the exponentially decreasing phase angles in the iQFT to prune weak long-range interactions and reduce inter-node communication. This note adds a complementary view: after pruning determines *which* QPUs must communicate, the hardware graph determines *how far* those communications must travel. Modeling remote cost as a routing functional on the QPU interconnect yields a simple architecture-level proposition: a hypercube QPU network, with logical blocks placed by Gray code, gives worst-case inter-node distance $O(\log P)$, and retained horizon-D iQFT interactions have distance at most $\min\{D, \log_2 P\}$.

1 From a circuit to a geometry

On a single quantum processor, geometry is mostly hidden. If the machine holds all n qubits, then the inverse QFT is simply the unitary

$$\mathrm{iQFT}_n = \mathrm{QFT}_n^\dagger,$$

implemented by local basis changes and controlled phase rotations. Up to convention-dependent constants, the controlled phase angles decay exponentially with qubit separation:

$$|\theta_k| \asymp 2^{-k}.$$

Thus the farthest phase corrections are also the faintest ones. This is the familiar source of approximate QFT constructions, including the parallel-depth results of Cleve and Watrous [2] and the approximate Fourier transform perspective of Coppersmith [3].

In a distributed architecture, the same observation becomes physical. A small phase rotation between qubits on the same QPU is a gate. A small phase rotation between qubits on different QPUs is a network event. The question is no longer only: how many gates does the iQFT require? It is also: through what geometry must the surviving gates be routed?

Intuitive Interpretation

On one processor, the iQFT is mainly a circuit-depth and gate-count object. Across many processors, the same mathematical circuit becomes spatial: a controlled phase rotation now has an address, a route, and a communication cost. The algorithm has not changed, but the architecture has made its hidden geometry visible.

2 The distributed iQFT skeleton

Let P be the number of QPUs and let each QPU hold Q logical qubits, so

$$n = PQ.$$

Node $p \in \{0, \dots, P-1\}$ holds the block

$$B_p = \{pQ, pQ+1, \dots, (p+1)Q-1\}.$$

Cardama et al. [1] formulate the iQFT over this distributed register and introduce a communication horizon: phase rotations whose qubit-level distance is too large are omitted because their angles are below a prescribed threshold. At the node level, this yields a banded communication pattern rather than an all-to-all one.

For a short architecture note, the following idealized abstraction captures the effect.

Model: horizon-D distributed iQFT graph

Let

$$G_D = (V, E_D), \qquad V = \{0, \dots, P-1\},$$

where

$$(p, q) \in E_D \iff 0 < |p - q| \le D.$$

The parameter D is the node-level communication horizon. Exact distributed iQFT corresponds to the dense case $D = P - 1$. Approximate distributed iQFT corresponds to a finite horizon $D \ll P$.

Intuitive Interpretation

The horizon D is not a new quantum operation. It is a bookkeeping line: below the line, phase corrections are kept; beyond it, they are judged too small to justify remote communication. The result is a banded logical interaction graph instead of a dense one.

This abstraction deliberately separates two issues. The iQFT approximation controls the edge set E_D. The hardware architecture controls the distance cost of using those edges.

3 A graph-distance communication cost

Let the physical QPU interconnect be a graph

$$G_{\text{phys}} = (V_{\text{phys}}, E_{\text{phys}}), \qquad |V_{\text{phys}}| = P.$$

A placement is a bijection

$$\sigma : V \to V_{\text{phys}},$$

assigning logical iQFT blocks to physical QPU locations. Define physical communication distance by graph geodesic distance:

$$d_{\text{phys}}(u, v) = \text{dist}_{G_{\text{phys}}}(u, v).$$

Now attach a nonnegative weight w_{pq} to each retained logical communication edge. Depending on the modeling choice, w_{pq} may count communication blocks, EPR-pair demand, retained remote gates, or a phase-weighted sum of the corresponding controlled rotations.

Definition: graph-distance iQFT communication cost

For a placement σ, define

$$C_D(\sigma) = \sum_{(p,q)\in E_D} w_{pq}\, d_{\mathrm{phys}}(\sigma(p), \sigma(q)).$$

This functional measures not only how many remote iQFT interactions survive pruning, but how far they must travel through the physical QPU network.

Intuitive Interpretation

The cost $C_D(\sigma)$ has two moving parts. The set E_D says which processors must talk; the distance term says how expensive each conversation is once the processors have been placed on hardware. Pruning reduces the number of calls. Routing makes the remaining calls shorter.

In this language, the pruning argument and the architecture argument are complementary:

$$\text{pruning reduces } |E_D|, \qquad \text{geometry reduces } d_{\mathrm{phys}}.$$

The original distributed iQFT result controls the first term. The next natural question is whether a useful physical graph can control the second.

4 A hypercube special case

The simplest useful special case is a hypercube. Assume

$$P = 2^m.$$

Let the physical QPU network be the m-dimensional hypercube

$$H_m = \{0,1\}^m,$$

where two vertices are adjacent when their binary labels differ in exactly one coordinate. The graph distance in H_m is Hamming distance, so

$$\mathrm{diam}(H_m) = m = \log_2 P.$$

To preserve locality of consecutive logical blocks, place the blocks by Gray code:

$$\sigma(p) = \mathrm{Gray}(p) = p \oplus (p \gg 1),$$

where $\oplus$ denotes bitwise XOR and $p \gg 1$ denotes a right shift.

Proposition: Gray-code hypercube placement

Let $P = 2^m$ and let $G_{\text{phys}} = H_m$. Place logical node p at physical vertex

$$\sigma(p) = \text{Gray}(p).$$

Then for every retained horizon-D iQFT edge $(p, q) \in E_D$,

$$d_{H_m}(\sigma(p), \sigma(q)) \leq \min\{|p - q|, \log_2 P\} \leq \min\{D, \log_2 P\}.$$

Consequently,

$$C_D(\sigma) \leq \min\{D, \log_2 P\} \sum_{(p,q)\in E_D} w_{pq}.$$

Proof. The distance in an m-dimensional hypercube is Hamming distance. Therefore every pair of physical vertices is separated by at most the hypercube diameter:

$$d_{H_m}(u, v) \leq m = \log_2 P.$$

This gives

$$d_{H_m}(\sigma(p), \sigma(q)) \leq \log_2 P$$

for all p, q.

Now use the Gray-code placement. Consecutive Gray-code words differ in exactly one bit, hence

$$d_{H_m}(\sigma(r), \sigma(r + 1)) = 1$$

for every consecutive logical pair. If $q > p$, then

$$\sigma(p), \sigma(p + 1), \ldots, \sigma(q)$$

is a path in the hypercube of length $q - p$. Hence the shortest-path distance satisfies

$$d_{H_m}(\sigma(p), \sigma(q)) \leq q - p = |p - q|.$$

Combining the diameter bound with the Gray-code path bound gives

$$d_{H_m}(\sigma(p), \sigma(q)) \leq \min\{|p - q|, \log_2 P\}.$$

For $(p, q) \in E_D$, we have $|p - q| \leq D$, so

$$d_{H_m}(\sigma(p), \sigma(q)) \leq \min\{D, \log_2 P\}.$$

Multiplying by w_{pq} and summing over retained edges proves the cost bound. □

Corollary: exact and horizon-limited regimes

For exact all-to-all distributed iQFT, $D = P - 1$, every required pair of QPUs is at most $\log_2 P$ hops apart on the hypercube. For approximate horizon-D iQFT, every retained edge is at most $\min\{D, \log_2 P\}$ hops apart. In particular, if $D = O(1)$ and the per-edge weights are uniformly bounded, then the geodesic communication cost is linear in P.

The corollary does not replace the pruning result. It says something different. Pruning controls the number of conversations. Hypercube geometry controls how long the remaining conversations are allowed to be.

Intuitive Interpretation

The hypercube is useful here because it behaves like a compact routing fabric. It preserves immediate logical neighbors through the Gray-code path, yet no pair of processors is ever farther than $\log_2 P$ hops. Local dependencies stay local, and even nonlocal dependencies are logarithmically bounded.

5 Interpretation

The iQFT contains a small architectural gift: its controlled phase rotations are ordered by significance. Long-range phase corrections are exponentially weaker than short-range corrections. Cardama et al. [1] turn that decay into a communication horizon. The geodesic model says that, once such a horizon is chosen, the surviving communication should be viewed as a weighted graph that must be embedded into a physical network.

The hypercube special case is attractive because it gives two useful properties at once:

- **Global reach:** any pair of QPUs is separated by only $O(\log P)$ hops.
- **Local preservation:** Gray-code placement maps consecutive logical blocks to adjacent physical QPUs.

So the qualitative picture is:

iQFT phase decay
$\Downarrow$
banded logical communication
$\Downarrow$
short geodesic routing on a chosen hardware graph.

This is a modest extension, but it is a useful one. It makes explicit that distributed iQFT efficiency has two layers: the algorithmic layer determines which phase interactions are still worth keeping, and the architectural layer determines how expensively those interactions are realized.

6 Limitations and possible next steps

The model above intentionally ignores several effects that a full architecture paper would need to address. Hypercubes have logarithmic degree, which may be unrealistic for some modular quantum platforms. The proposition bounds path length but not edge congestion, routing conflicts, entanglement generation latency, purification overhead, or accumulated noise along multi-hop paths. Those effects could be incorporated by replacing graph distance with a weighted latency or fidelity cost.

Nevertheless, the formulation suggests a natural optimization problem:

$$\min_{\sigma} \sum_{(p,q)\in E_D} w_{pq}\, d_{\mathrm{phys}}(\sigma(p), \sigma(q)).$$

For the iQFT, the weights are not arbitrary. They inherit the ordered decay of the phase rotations. That structure may make the placement problem easier than a generic graph embedding problem: the most important conversations are already arranged close to the diagonal.

Summary in one sentence

The distributed iQFT paper reduces the number of remote phase interactions; the routing extension asks how to place the remaining interactions on a physical QPU graph so that the surviving phase information travels along short paths.

References

[1] F. Javier Cardama, Jorge Vázquez-Pérez, Tomás F. Pena, and Andrés Gómez. *Communication-Efficient Distributed Inverse Quantum Fourier Transform*. arXiv:2605.10710, 2026. https://arxiv.org/abs/2605.10710

[2] Richard Cleve and John Watrous. *Fast Parallel Circuits for the Quantum Fourier Transform*. arXiv:quant-ph/0006004, 2000. Later published in *SIAM Journal on Computing*, 45(4):1570–1595, 2016. https://arxiv.org/abs/quant-ph/0006004

[3] D. Coppersmith. *An Approximate Fourier Transform Useful in Quantum Factoring*. arXiv:quant-ph/0201067, 2002. Originally IBM Research Report RC19642, 1994. https://arxiv.org/abs/quant-ph/0201067

[4] Peter W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. arXiv:quant-ph/9508027, 1995. https://arxiv.org/abs/quant-ph/9508027