QuComm: Optimizing Collective Communication for DQC

Wu, Ding (UCSB), Li (PNNL)

April 9, 2025. based on the MICRO 2023 paper

Background: DQC Architecture and Communication Challenges

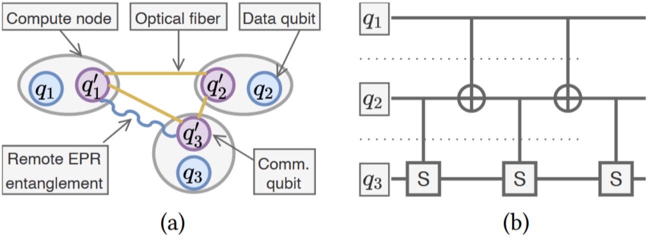


Figure: Common DQC setup. Data qubits handle computation; comm qubits generate EPR pairs.

- Distributed computing uses remote EPR entanglement.
- Inter-node communication is expensive and error-prone.

Background: Communication Protocols

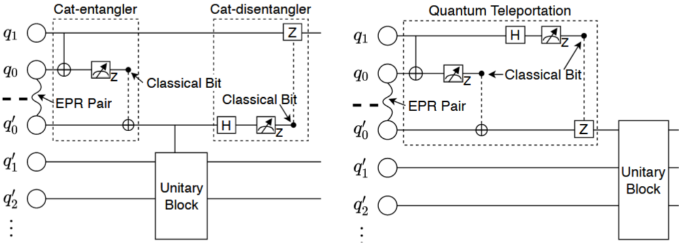


Figure: (a) Cat-Comm shares a qubit state; (b) TP-Comm moves it entirely.

- Cat-Comm: Efficient for read-only use of shared qubits.
- TP-Comm: Required if qubit must be modified.
- Each consumes 1 EPR pair.

QuComm: Overview

- Goal: Reduce costly inter-node communication in Distributed Quantum Computing (DQC).
- Key Idea: Identify and optimize *collective communication* patterns involving multiple nodes.
- Authors: Wu, Ding (UCSB), Li (PNNL)

What is Collective Communication?

- **Definition:** A group of inter-node gates involving multiple nodes whose qubit interaction graph is connected.
- **Goal:** Execute all gates in the group together to reduce total inter-node communications.
- **Benefit:** Significant savings — e.g., 5 CNOTs across 3 nodes can use just 2 EPRs if optimized collectively.
- **Challenges:**
  - Hidden in low-level gates.
  - Nontrivial to route and support on limited hardware.

Motivation: Limitations of Current DQC Compilers

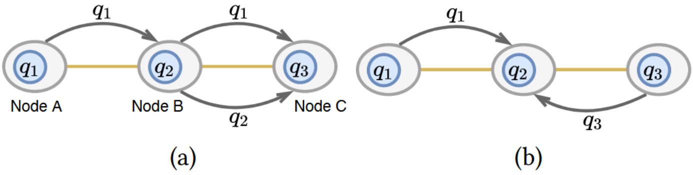


Figure: Two examples of routing the collective communication block.

- Existing compilers treat inter-node gates independently.
- Miss opportunities to group multi-node gates.
- QuComm reduces comms by executing groups collectively.

Method: QuComm Compilation Pipeline

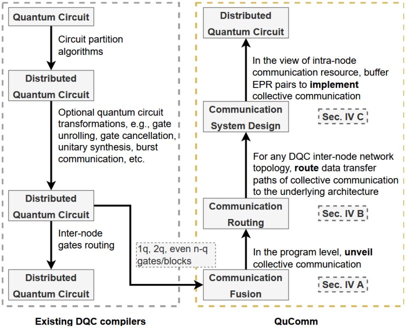


Figure: Compilation flow.

- Stage 1: Communication Fusion
- Stage 2: Communication Routing
- Stage 3: Communication Buffering

Stage 1: Communication Fusion

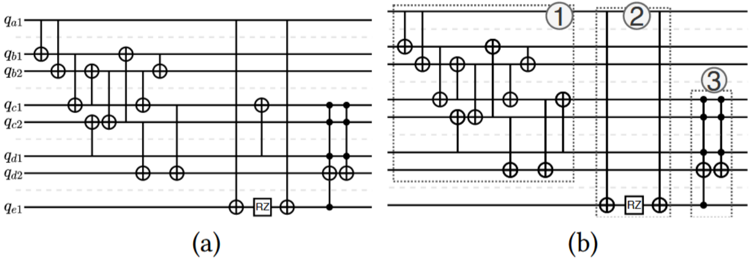
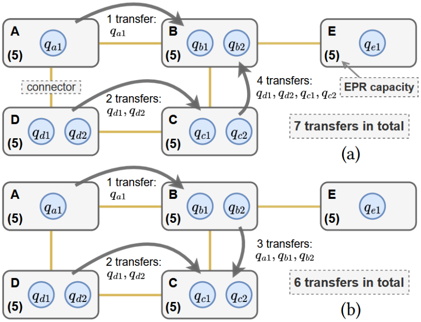


Figure: Scattered inter-node gates merged into one collective block, where each node's EPR capacity is 5.

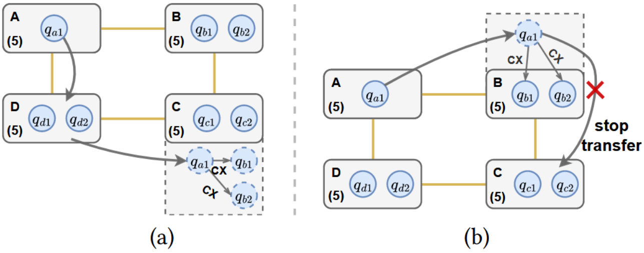
- Identify and group overlapping multi-node gates.
- Merge if group execution reduces communication cost.
- Result: Fewer, more efficient collective blocks.

Stage 2: Routing



- Select optimal aggregation node.
- Design shortest paths with early gate execution.

Stage 2: Routing



- Select optimal aggregation node.
- Design shortest paths with early gate execution.

Stage 3: Communication Buffering

- **Problem:** Not enough comm qubits for large blocks.
- **Solution:** Use spare data qubits to buffer EPR pairs.
- **Result:** Enables large block execution with limited resources.
- Only add buffer when it reduces overall cost.

Evaluation Overview

- Benchmarks: XOR, RCA, QFT, Grover, etc.
- Baseline: AutoComm, GP-CAT, GP-SWAP
  - **AutoComm:** state-of-the-art DQC compiler prior to QuComm
  - GP-CAT: Uses only Cat-Comm
  - GP-SWAP: Swaps qubits into place
- Metric: Number of inter-node communication ops

Comparison to AutoComm

Program	Comm reduction by QuComm L1			Comm reduction by QuComm L1+L2			Comm reduction by QuComm L1+L2+L3		
	1 comm lqb/node	3 comm lqb/node	5 comm lqb/node	1 comm lqb/node	3 comm lqb/node	5 comm lqb/node	1 comm lqb/node	3 comm lqb/node	5 comm lqb/node
XOR-100	46.2%	66.7%	66.7%	46.2%	75.0%	75.0%	76.9%	75.0%	75.0%
XOR-200	26.2%	58.8%	70.6%	26.2%	58.8%	70.6%	33.3%	58.8%	70.6%
XOR-300	3.7%	41.8%	54.1%	3.7%	43.9%	56.1%	69.2%	68.4%	68.4%
RCA-100	0.0%	0.0%	0.0%	0.0%	50.0%	50.0%	80.0%	50.0%	50.0%
RCA-200	0.0%	0.0%	0.0%	0.0%	50.0%	50.0%	75.0%	50.0%	50.0%
RCA-300	0.0%	0.0%	0.0%	0.0%	50.0%	50.0%	80.0%	50.0%	50.0%
XORR-100	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	60.0%	0.0%	0.0%
XORR-200	0.0%	0.0%	0.0%	0.0%	50.0%	20.0%	28.0%	50.0%	20.0%
XORR-300	0.0%	0.0%	0.0%	0.0%	74.1%	61.1%	84.4%	74.1%	61.1%
QFT-100	0.0%	0.0%	0.0%	0.0%	20.0%	20.0%	42.1%	20.0%	20.0%
QFT-200	0.0%	0.0%	0.0%	0.0%	29.4%	29.4%	43.8%	29.4%	29.4%
QFT-300	0.0%	0.0%	0.0%	0.0%	61.0%	61.0%	59.6%	61.0%	61.0%
Grover-100	46.2%	66.7%	66.7%	46.2%	75.0%	75.0%	76.9%	75.0%	75.0%
Grover-200	26.2%	58.8%	70.6%	26.2%	58.8%	70.6%	33.3%	58.8%	70.6%
Grover-300	3.7%	41.8%	54.1%	3.7%	43.9%	56.1%	69.2%	68.4%	68.4%

Figure: Detailed benchmark results showing QuComm reduces communication by 40–70% over AutoComm. Each compiler stage (Fusion, Routing, Buffering) contributes to consistent improvements across programs like QFT, Grover, and RCA.

Comparison to GP-CAT, GP-SWAP

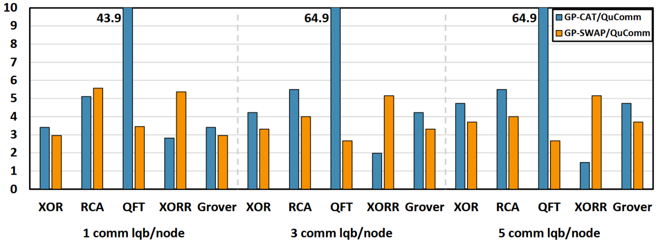
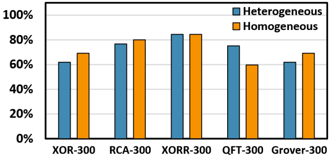


Figure: QuComm achieves up to 5× fewer inter-node communications compared to GP-CAT and GP-SWAP. Collective block optimization enables large communication savings that simpler burst-based or SWAP-based methods cannot match.

Architecture Adaptability



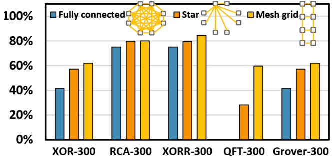


Figure: QuComm remains effective across topologies and node sizes.

- Works on mesh, star, and full networks.
- Robust to node heterogeneity.
- Collective optimization generalizes well.

Conclusion

- QuComm uncovers hidden multi-node patterns.
- Substantially reduces inter-node gate count.
- Enables scalable DQC with limited hardware.