Distributing Graph States Among Quantum Networks

Alex Fischer, Don Towsley

University of Massachusetts, Amherst

Background: Graph States

- Class of quantum multipartite entangled states.
- Given graph (V, E), corresponding graph state has qubits corresponding to vertices; all qubits initially in $|+\rangle$ state; controlled-Z gate applied to every pair of qubits in E.
- Graph state is

$$\left(\prod_{(u,v)\in E} CZ_{u,v}\right) |+\rangle^{\otimes |V|},$$

  $CZ_{u,v}$—controlled-Z operation between qubits corresponding to vertices u and v.
- CZ operations commute—can be done in any order (or simultaneously).

Background: Quantum Network

- Collection of *nodes*: individual quantum computers with unlimited qubits and capacity to perform local quantum operations.
- Certain node pairs can generate EPR pairs between qubits in nodes.
- Nodes analogous to routers or repeaters: communicate with nearby nodes in their neighborhood in order to effect long-range communication.
- EPR pair generation expensive: want to minimize.
- Exactly which node pairs can generate EPR pairs determines topology of network.

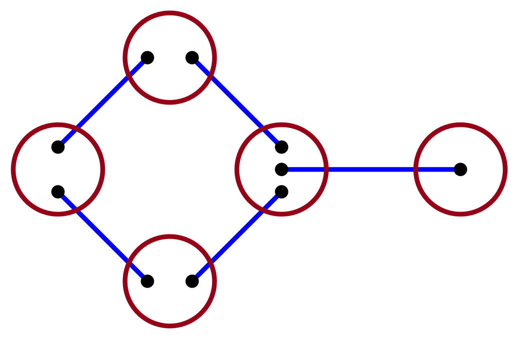


Figure 1: Red circles represent nodes; blue edges represent connections between nodes, which can be regenerated after being consumed by quantum operations within nodes.

Problem Statement

- Task: distribute any graph state among (possibly subset of) nodes of a quantum network.
- Minimize resource usage: time, EPR pair consumption, classical communication.
- Approach: start with a local copy of the graph state in a node, then distribute each vertex to relevant node (see Figure 2).

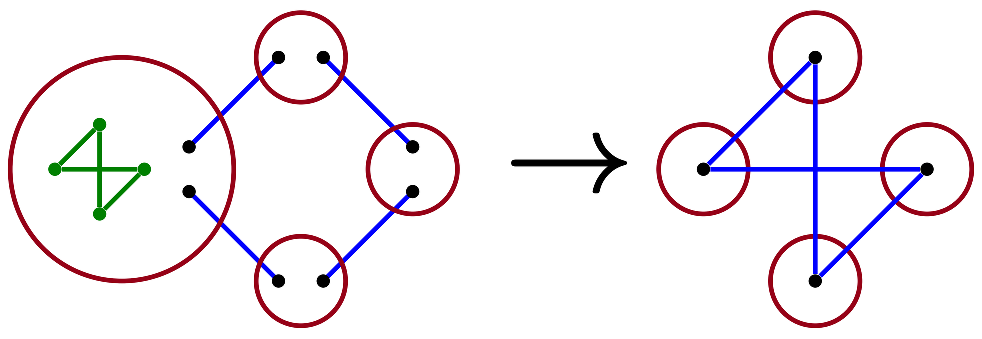
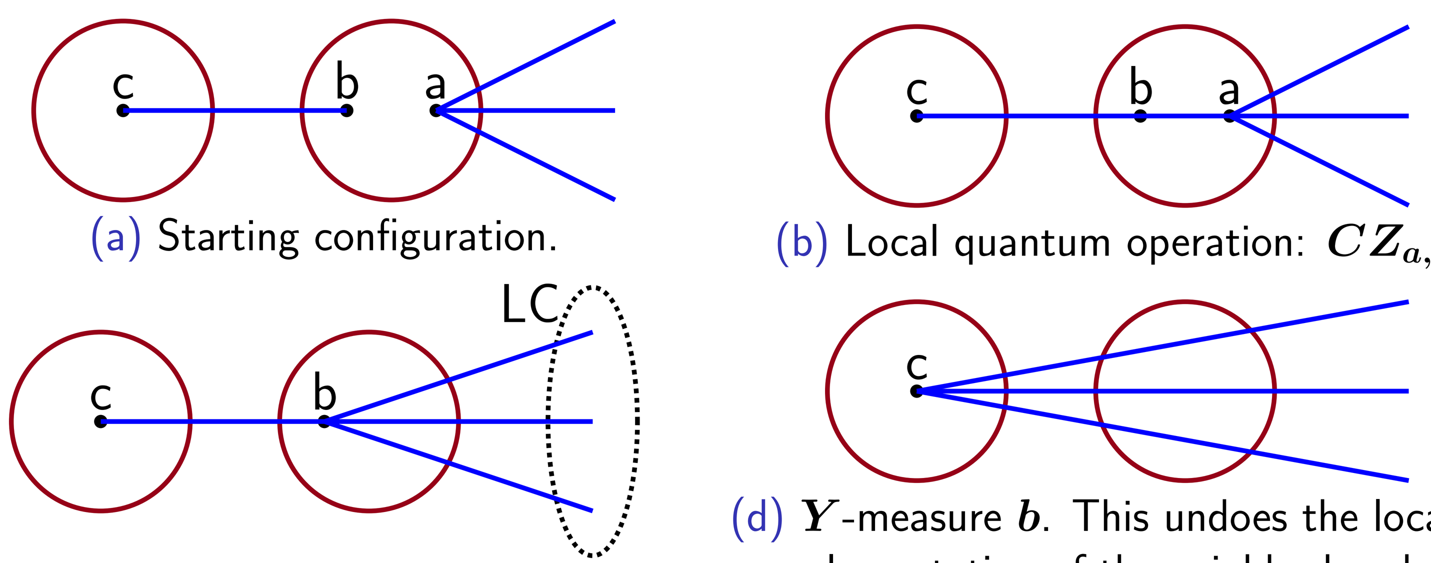


Figure 2: Setup and end result of our graph state distribution approach. Local copy of the final graph state (green) is prepared within a node and distributed throughout network.

Connection Transfer

- Building block task: change network graph state so that edges connected to qubit a are instead connected to qubit c in another node; consumes an EPR pair between c's node and a's node.
- Two approaches: via graphical operations (Figure 3); via teleportation.



(a) Starting configuration.

(b) Local quantum operation: $CZ_{a,b}$.

(c) Y-measure a. Results in locally complementing a's (former) neighborhood.

(d) Y-measure b. This undoes the local complementation of the neighborhood of a.

Figure 3: The connection transfer process. We wish to transfer the edges connected to a to qubit c, by consuming EPR pair between b and c.

- Both methods have commutation properties which allow us to transfer connections along a path in the network in $O(1)$ time by doing all CZ and measurement operations at once, then communicating all measurement results to destination node, then doing one local correction operation.

Graph State Distribution Algorithm

- Based on a connection transfer algorithm:
  1. Create local copy of graph state in some node, referred to as the root.
  2. For each qubit in graph state, perform connection transfer of that qubit's connections along path of edges in network to relevant node in the network, regenerating EPR pairs as necessary.
- Performance of algorithm depends on paths chosen—$(n-1)$ paths (n is number of network nodes), each going from the root to one of $n-1$ other nodes in the network that will share graph state.

EPR Pair Consumption Bounds

- Minimizing EPR pair consumption equivalent to following graph problem: minimize sum of lengths of paths each going from a root node to every other node.
- For simplicity, consider a spanning tree of the network.
- Fact 1: at most $n-i$ nodes can be distance i away from root in this tree; max sum of path lengths is

$$(n-1)+(n-2)+(n-3)+\cdots+1=\frac{n(n-1)}{2}.$$

- Fact 2: can select root that is distance at most $\lceil\frac{n-1}{2}\rceil \leq \frac{n}{2}$ from each node. Hence max sum of path lengths is

$$\frac{n}{2}\cdot\frac{n}{2}+\left(\frac{n}{2}-1\right)+\left(\frac{n}{2}-2\right)+\cdots+1$$
$$=\frac{n^2}{4}+\frac{n/2(n/2-1)}{2}=\frac{3n^2-2n}{8}.$$

Timestep Requirements

- Connection transfers for each qubit in graph state can all be done in one timestep.
- $n-1$ timesteps required for connection transfers for all qubits in graph state.
- For paths in network do not share any edges, connection transfers along those paths can be done simultaneously in one timestep.
- Minimizing time is equivalent to graph problem: Find $n-1$ paths from root to every other node that minimizes maximum number of times any edge is used.

Timestep Minimization via Network Flow

- Question "can we construct $n-1$ paths from the root node to every other node such that no edge is used more than k times?" is equivalent to network flow problem in Figure 4.
- We can minimize time taken by binary search on k.

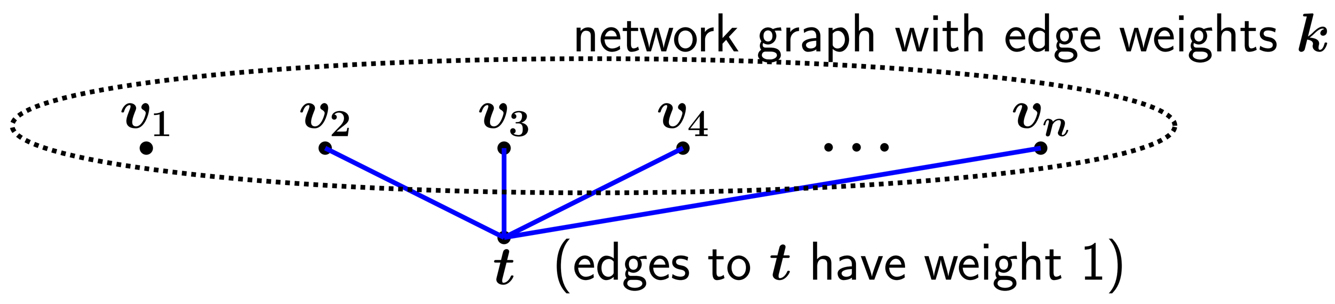


Figure 4: Let every edge in this graph (which is the network graph, plus $n-1$ edges from every non-root node to an extra vertex t) have weight k, except for the edges to t which have weight 1. Then max flow from v_1 (the root) to t is $n-1$ iff there is set of $n-1$ paths from root to every other node such that every edge is used at most k times.

Resource Graph State

- Assume we know the set of nodes that will request to share a graph state, but we don't know what the graph state is.
- We can distribute a *resource graph state* ahead of time among those nodes—graph state that allows us to distribute any other graph state in one timestep once we know desired graph state.
- This approach suggests following resource graph state: EPR pairs shared between one root node and all other nodes in the set (Figure 5).
- Requires maintaining $2(n-1)$ qubits.

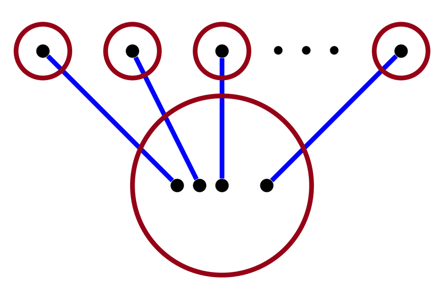


Figure 5: A resource graph state that requires maintaining $2(n-1)$ qubits, where n is number of nodes that will share final graph state.

Prior Work Comparison

	EPR pairs	Time	Res. Graph State Qubits	Classical Comm.
Our approach	$\frac{3n^2-2n}{8}$	$n-1$	$2(n-1)$	$O(n^2)$
Prior work	$\frac{n(n-1)}{2}$	$n-1$	$\frac{n(n+1)}{2}$	$O(n^2)$

Table 1: Comparison of various performance metrics to prior work on graph state distribution.