REGARDING OPTIMAL RESOURCE ALLOCATION IN QUANTUM NETWORKS

ABHAY SHANKAR K, RISHIT D, SAKSHI TAKALE

Abstract. We wish to present a well-informed mathematical approach to the economics of quantum networks in the future. To do so, we discuss theoretically optimal link-bandwidth allocation and assign costs proportionate to the 'importance' of a link or its ends, and also present ways to assess said importance.

1. Recap

Find a function $f : T \times \mathcal{E} \to \mathcal{P}(\mathcal{D} \times \mathbb{N})$ that minimises the cost

$$\mathcal{C} = \sum_{t \in T} \left(\sum_{e \in E'(t)} c\,(e) \right)$$

where

- $E'(t) = \{e \in \mathcal{E} \mid f(t, e) \neq \{\}\}$ is the set of all edges that are active at time t
- $T \subset \mathbb{N}$ is the set of all time slots during which the network operates.
- The output of the allocation function is a set of (α, n) pairs, which specify the number of qubits (n) of which demand (α) the edge should transmit. This corresponds to $(\alpha_1, \gamma_1)..((\alpha_k, \gamma_k))$.
  - Here $\forall k\ \gamma_k < \alpha_k$`.demand`, i.e. the edge cannot transmit more qubits than the demand,
  - $\Sigma_k \gamma_k \leq \Gamma$ for a given edge,
  - Flow conservation: $\forall t \in T\ \forall u \in V\ \forall g \in \mathcal{E}_u\ \Sigma_g s(g) f(t, g) = 0$ where $\mathcal{E}_u$ is the set of edges incident on u and $s(g)$ is 1 for outgoing edges and -1 for incoming edges.
  - α_k is a valid demand.

2. Algorithm

I have no clue. Big graphy stuff. Probably finding the minimum demand, find the maxflow path between the nodes, assign the demand, and ad infinitum. Also probably have to do some BFS timeslot checks to make sure the link is not full. Maybe do buffering if link at low capacity.

We derive inspiration from the SJF algorithm in queueing theory. We find the demand corresponding to the shortest transmission time, and add that demand to the current timeslot. We iteratively find the shortest transmission time and add

3. Heuristics

Too many ideas. We won't know which subset of this is practical until we math it.

- Total link contributions from links in the path (capacity, cooldown).
- Buffer size at nodes involved - that way greater chance of not being dropped.
- Bottleneck bandwidth.
- Node costs?
- Betweenness-centrality, except for maxflow paths? Do we need a new link attribute for this? Maybe something in the topology that creates a certain distribution (think pmf) of this shiny new attribute among all the edges?
- GHZ extra cost?

- Link-layer QoS guarantees? - i.e. number of times a link tries to reconstruct a qubit if measurement keeps failing before giving up. We could incorporate a QoS guarantee into the link-layer protocol.

4. Sample

So we put up a few different networks of a few different topologies (not too many) and an arbitrary demand vector, then compute the cost of each one.