Learning Best Paths in Quantum Networks

Xuchuang Wang[1], Maoli Liu[2], Xutong Liu[3],

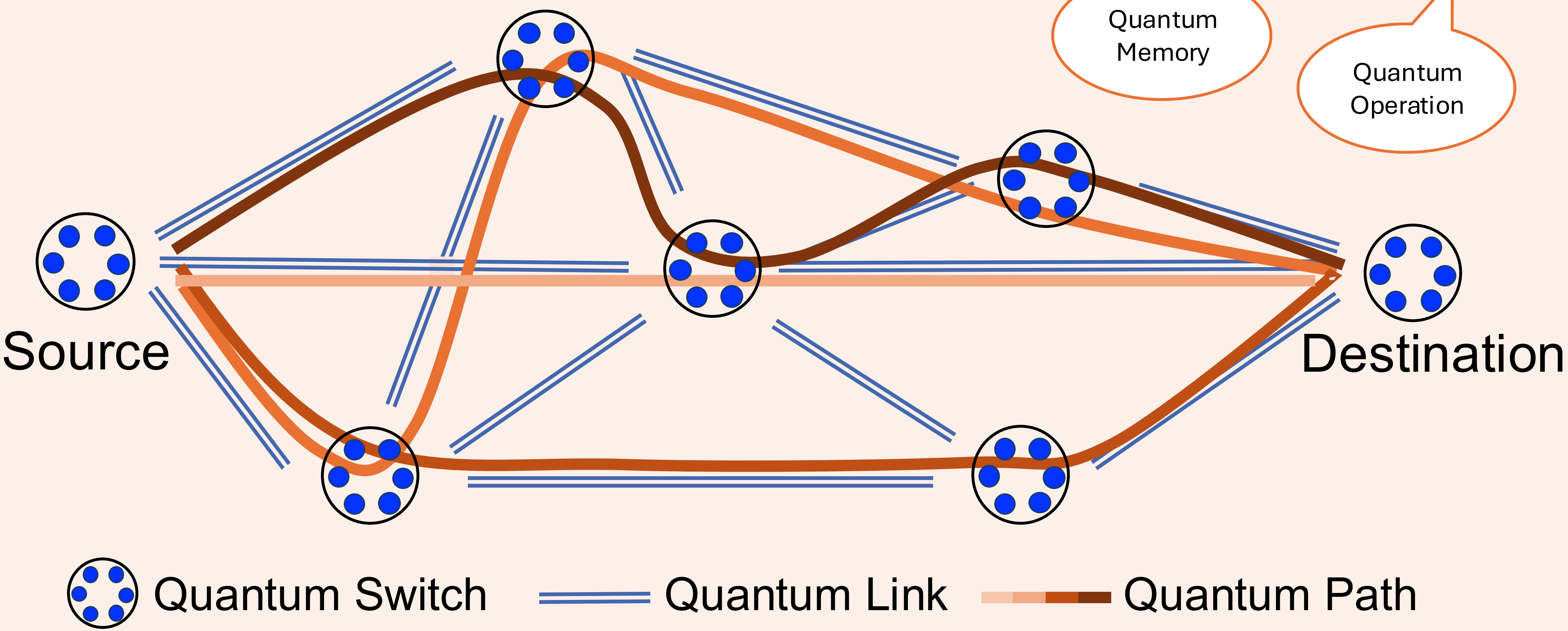
Mohammad Hajiesmaili[1], John C.S. Lui[2], Don Towsley[1]

[1]University of Massachusetts Amherst, [2]Chinese University of Hong Kong,
[3]Carnegie Mellon University

May 2025

Quantum Network

Quantum Network Category		Operation Error (Switch)	
		HEG	QEC
Loss Error (Link)	HEG	**Two-Way (1G)**	2G
	QEC		**One-Way (3G)**



Muralidharan, Sreraman, et al. "Optimal architectures for long distance quantum communication." *Scientific reports* 6.1 (2016): 20463.

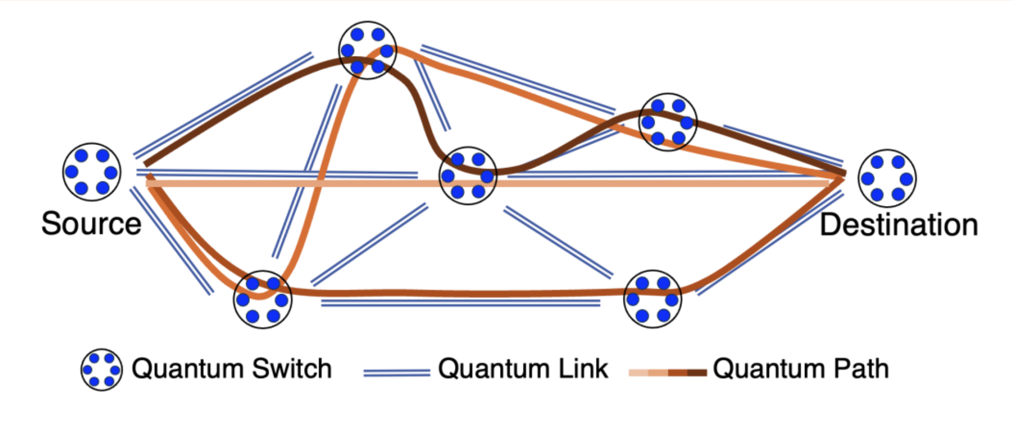
Outline

- Background and `BeQuP` Modeling
  - Network Benchmarking
- Online Learning Algorithms
  - Link-Level Benchmarking: `BeQuP-Link`
    - `BestPath` Subroutine
  - Path-Level Benchmarking: `BeQuP-Path`
    - `LinkEst` Subroutine
  - Analysis
- NetSquid Experiment

Outline

- **Background and `BeQuP` Modeling**
  - Network Benchmarking
- Online Learning Algorithms
  - Link-Level Benchmarking: `BeQuP-Link`
    - `BestPath` Subroutine
  - Path-Level Benchmarking: `BeQuP-Path`
    - `LinkEst` Subroutine
  - Analysis
- NetSquid Experiment

Learning Best Quantum Path



- Quantum Network: L links and K paths
  - Each link with **fidelity** f_ℓ, **depolarization** parameter p_ℓ **Unknown**
  - Path $\mathcal{L}(k)$'s depolarization parameter $p^{\text{path}}(k) = \prod_{\ell \in \mathcal{L}(k)} p_\ell$
  - Path $\mathcal{L}(k)$'s fidelity $f^{\text{path}}(k) = \frac{1+p^{\text{path}}(k)}{2}$

$K \gg L$, e.g., $K = O(L^{L_{\max}})$

- Objective: Find the path with the highest channel fidelity $f^{\text{path}}(k)$ with as few quantum resources as possible (resource complexity)
  - $k^* = \text{argmax}_k f^{\text{path}}(k)$

$$f^{\text{path}}(k) \longleftrightarrow p^{\text{path}}(k) \longleftrightarrow \sum_{\ell \in \mathcal{L}(k)} \log p_\ell \triangleq F(k)$$

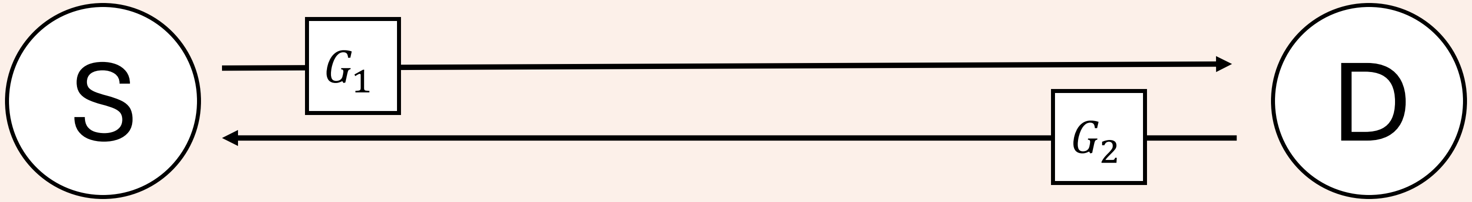
Network Benchmarki

Quantum Network Category		Operation Error	
		HEG	QEC
Loss Error	HEG	**Two-Way (1G)**	2G
	QEC		**One-Way (3G)**

Quantum Memory

Quantum Operation

- Goal: estimate a channel fidelity $f = \frac{1+p}{2}$ via estimating depolarization parameter p
- Approach: bounce



S and D are two nodes
G is a random Clifford gate

- **Requires Clifford gate operation at quantum switches!**

Link estimate $\hat{p}_\ell$ guarantees $p_\ell \in \left(\hat{p}_{\ell,t} - \mathrm{rad}_{\ell,t}, \hat{p}_{\ell,t} + \mathrm{rad}_{\ell,t}\right)$ w.h.p.

Helsen, Jonas, and Stephanie Wehner. "A benchmarking procedure for quantum networks." *npj Quantum Information* 9.1 (2023): 17.

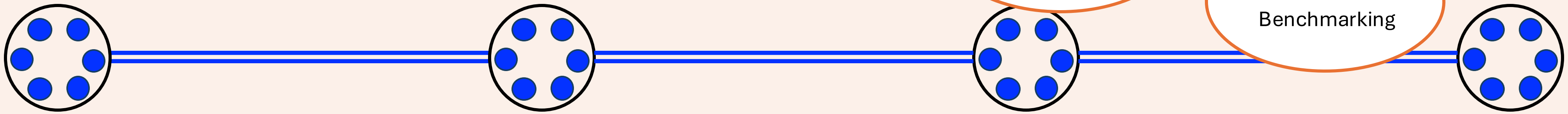
Benchmarking Levels

Quantum Network Category		Operation Error	
		HEG	QEC
Loss Error	HEG	**Two-Way (1G)**	2G
	QEC		**One-Way (3G)**

Path-level
Benchmarking

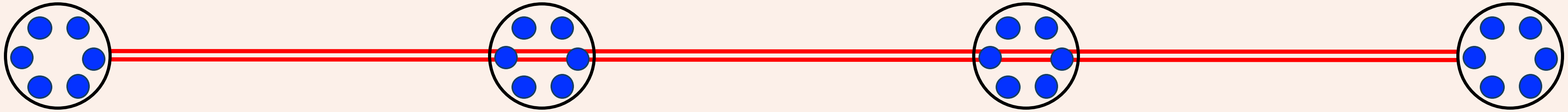
Link-level
Benchmarking

Link-level benchmarking: estimate p_ℓ of each link ℓ



- All quantum switches (all nodes) with required quantum operation ability
- Cost one unit of quantum resources

Path-level benchmarking: estimate $p^{\text{path}}(k)$ of each path k



- Only source and destination switches (two nodes) need the quantum operation ability
- Cost $L(k)$ (the number of links in the path) units of quantum resources

BeQuP: Online Learning for Best Quantum Path

Procedure 1 `BeQuP`: Learning Best Quantum Path

Input: Path set $\mathcal{K}$, link set $\mathcal{L}$, confidence parameter δ

repeat

Decision Making

Select a link ℓ_t / path k_t to benchmark

Observe link-level feedback $X_{\ell_t,t}$ from subroutine $\texttt{Bench}(\ell_t)$ / path-level $Y_t(k_t)$ from $\texttt{Bench}(\mathcal{L}(k_t))$

Stopping Condition

until Identify the best path k^* with confidence $1 - \delta$

Output: Best path k^*

Outline

- Background and `BeQuP` Modeling
  - Network Benchmarking
- **Online Learning Algorithms**
  - **Link-Level Benchmarking: `BeQuP-Link`**
    - `BestPath` Subroutine
  - Path-Level Benchmarking: `BeQuP-Path`
    - `LinkEst` Subroutine
  - Analysis
- NetSquid Experiment

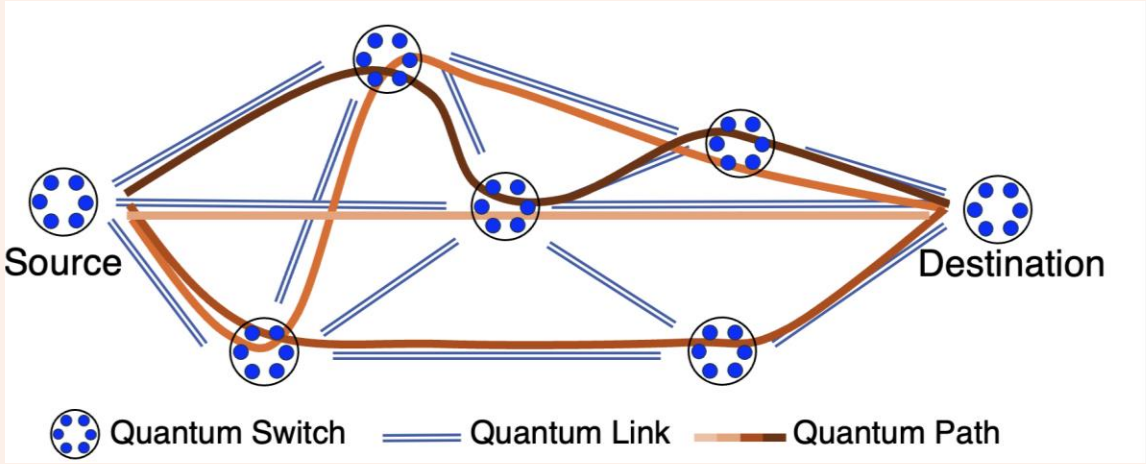
Link-level Benchmarking: `BestPath` Subroutine

OUTPUT: Best path argmax $F(k)$

INPUT: All link depolarization parameters p_ℓ & Topology $\mathcal{K}$

$$k \leftarrow \texttt{BestPath}(\boldsymbol{p}, \mathcal{K})$$

- e.g., Shortest path (**Dijkstra's algorithm**) with edge weight $-\log p_\ell$



One natural idea is first to estimate all p_ℓ and apply `BestPath`.
Can we do better?

Link estimate $\hat{p}_\ell$ guarantees $p_\ell \in \left(\hat{p}_{\ell,t} - \mathrm{rad}_{\ell,t}, \hat{p}_{\ell,t} + \mathrm{rad}_{\ell,t}\right)$ w.h.p.

Link-level Benchmarking: **BeQuP-Link** Algo.

- Find the empirical best path $\hat{k}_t \leftarrow \texttt{BestPath}(\hat{\boldsymbol{p}}_t, \mathcal{K})$
- Pess/Optimistic estimate $\tilde{p}_{\ell,t} \leftarrow \begin{cases} \hat{p}_{\ell,t} - \mathrm{rad}_{\ell,t} & \text{if } \ell \in \mathcal{L}(\hat{k}_t) \\ \hat{p}_{\ell,t} + \mathrm{rad}_{\ell,t} & \text{otherwise} \end{cases}$
- Find the **disturbed** empirical best path $\tilde{k}_t \leftarrow \texttt{BestPath}(\tilde{\boldsymbol{p}}_t, \mathcal{K})$

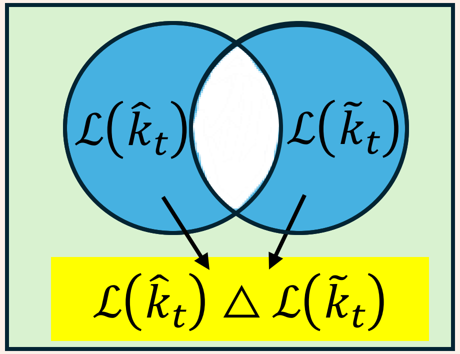
Stopping Condition

- $\mathcal{L}(\hat{k}_t) = \mathcal{L}(\tilde{k}_t)$, then output the best path $\hat{k}_t$
- Pick a link to explore [the most under-explored link]

Decision Making

$$\ell_t \leftarrow \underset{\ell \in \mathcal{L}(\hat{k}_t) \triangle \mathcal{L}(\tilde{k}_t)}{\operatorname{argmax}} \mathrm{rad}_{\ell,t}$$

Different links between the two paths



- Update Parameters

Outline

- Background and `BeQuP` Modeling
  - Network Benchmarking
- **Online Learning Algorithms**
  - Link-Level Benchmarking: `BeQuP-Link`
    - `BestPath` Subroutine
  - **Path-Level Benchmarking: `BeQuP-Path`**
    - `LinkEst` Subroutine
  - Analysis
- NetSquid Experiment

#Path $K \gg$ #Link L, e.g., $K = O(L^{L_{\max}})$

Estimate Link Parameters from Path Feedback

Path-level `BeQuP-Path`: **`LinkEst`** Subroutine

- **Input:** a set of paths $\mathcal{S}$ and a number of samples N
- [1: G/D-Optimal Design] $\boldsymbol{\lambda}^{\mathcal{S}} \leftarrow \max_{\boldsymbol{\lambda}} \mathbb{E}_{k \in \mathcal{D}(\boldsymbol{\lambda}, \mathcal{S})}[\boldsymbol{x}(k)\boldsymbol{x}^T(k)]$
- [2: Random Sampling] For $n = 1, \ldots, N$ do
  - Sample path k_n from $\mathcal{S}$ with probability $\mathcal{D}(\boldsymbol{\lambda}^{\mathcal{S}}, \mathcal{S})$
  - $Y_n \leftarrow$Bench$(\mathcal{L}(k_n))$ [Path Feedback]
- $\boldsymbol{A} \leftarrow N \sum_{k \in \mathcal{S}} \lambda^{\mathcal{S}}(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)$
- $\boldsymbol{b} \leftarrow \sum_{n=1}^{N} \log(Y_n)\, \boldsymbol{x}(k_n)$
- [3: Solve Linear Equation] **Output:** $\log \widehat{\boldsymbol{p}} \leftarrow \boldsymbol{A}^{-1}\boldsymbol{b}$

[Link Estimates]

$$p^{\text{path}}(k) = \prod_{\ell \in \mathcal{L}(k)} p_\ell$$

$$x_\ell(k) = \begin{cases} 1 & \text{if } \ell \in \mathcal{L}(k) \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{cases} \log p^{\text{path}}(1) = \sum_{\ell \in \mathcal{L}(1)} \log p_\ell \\ \vdots \\ \log p^{\text{path}}(K) = \sum_{\ell \in \mathcal{L}(K)} \log p_\ell \end{cases}$$

Quantum Resource Complexity & Comparison

- BeQuP-Link:

$$Q^{\text{link}} = O\left(L_{\max}^2 \sum_{\ell \in \mathcal{L}} \frac{1}{\Delta_\ell^2} \log \frac{L}{\delta}\right)$$

- $\Delta_\ell = \begin{cases} F(k^*) - \max_{k \in \mathcal{K}:\ell \in \mathcal{L}(k)} F(k) & \text{if } \ell \notin \mathcal{L}(k^*) \\ F(k^*) - \max_{k \in \mathcal{K}:\ell \notin \mathcal{L}(k)} F(k) & \text{if } \ell \in \mathcal{L}(k^*) \end{cases}$
- $L_{\max} = \max_{k \in \mathcal{K}} L(k)$: length of the longest path

- BeQuP-Path:

$$Q^{\text{path}} = O\left(L_{\max} \sum_{\ell=2}^{L} \frac{1}{\left(\Delta^{\text{path}}([\ell])\right)^2} \log \frac{K}{\delta}\right)$$

$K \gg L$, e.g., $K = O(L^{L_{\max}})$

- $\Delta^{\text{path}}(k) = F(k^*) - F(k)$
- $\Delta^{\text{path}}([\ell])$ is the ℓ-th smallest path gap

Compare with prior baselines

$Q^{\text{uniform}} = O\left(\frac{K L_{\max}}{\left(\Delta^{\text{path}}([1])\right)^2} \log \frac{K}{\delta}\right)$
$Q^{\text{LinkSelfiE}} = O\left(L_{\max} \sum_{k \in \mathcal{K}} \frac{1}{\left(\Delta^{\text{path}}(k)\right)^2} \log \frac{K}{\delta}\right)$ [Our INFOCOM'24]

Link vs. Path

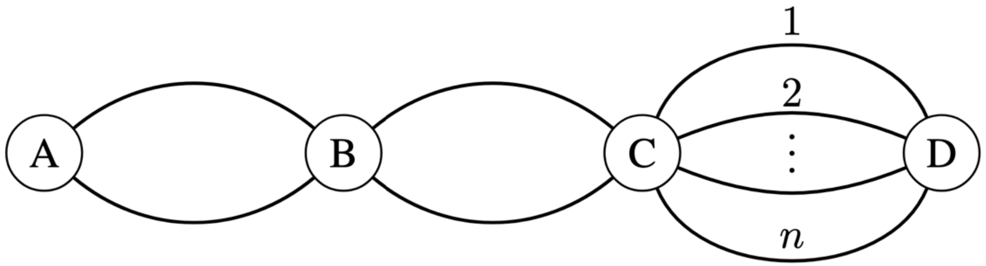
$Q^{\text{path}} = O\left(L_{\max}^2 \sum_{\ell \in \mathcal{L}} \frac{1}{\left(\Delta^{\text{path}}([\ell])\right)^2} \log \frac{L}{\delta}\right)$
$Q^{\text{link}} = O\left(L_{\max}^2 \sum_{\ell \in \mathcal{L}} \frac{1}{\Delta_\ell^2} \log \frac{L}{\delta}\right)$

Liu, Maoli, et al. "LinkSelfiE: Link selection and fidelity estimation in quantum networks." *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2024.

Outline

- Background and `BeQuP` Modeling
  - Network Benchmarking
- Online Learning Algorithms
  - Link-Level Benchmarking: `BeQuP-Link`
    - `BestPath` Subroutine
  - Path-Level Benchmarking: `BeQuP-Path`
    - `LinkEst` Subroutine
  - Analysis
- **NetSquid Experiment**

NetSquid Experiments



Extend our algorithm for finding the path with the highest SKF for QKD

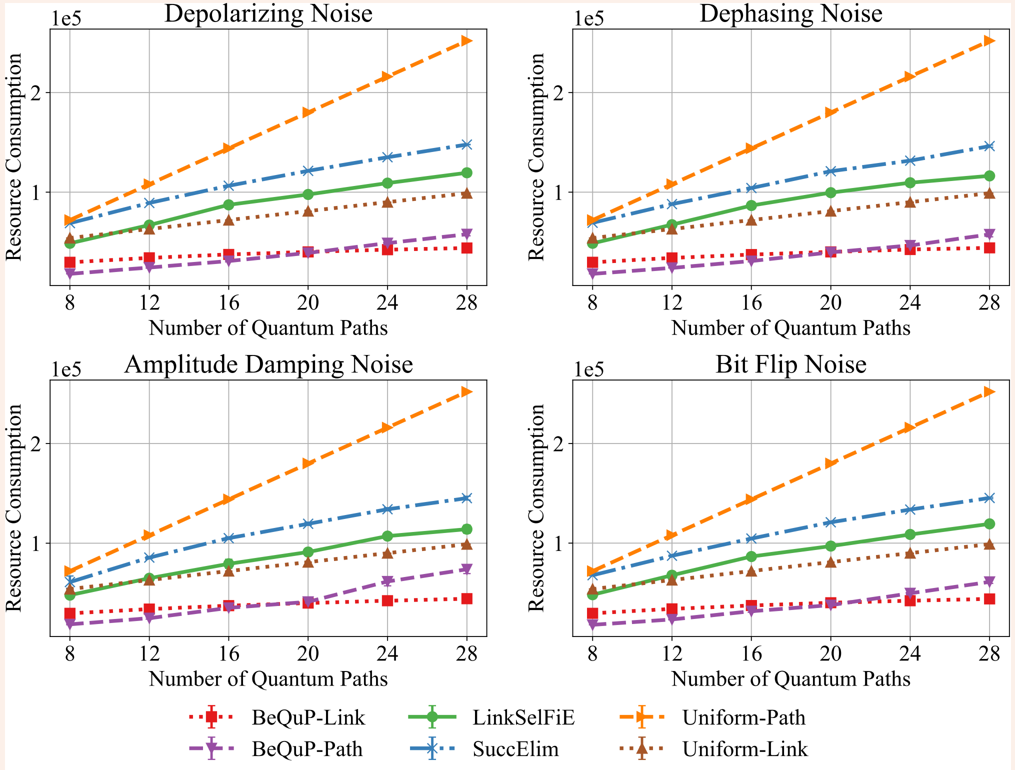


Figure 1: High-fidelity Path Identification

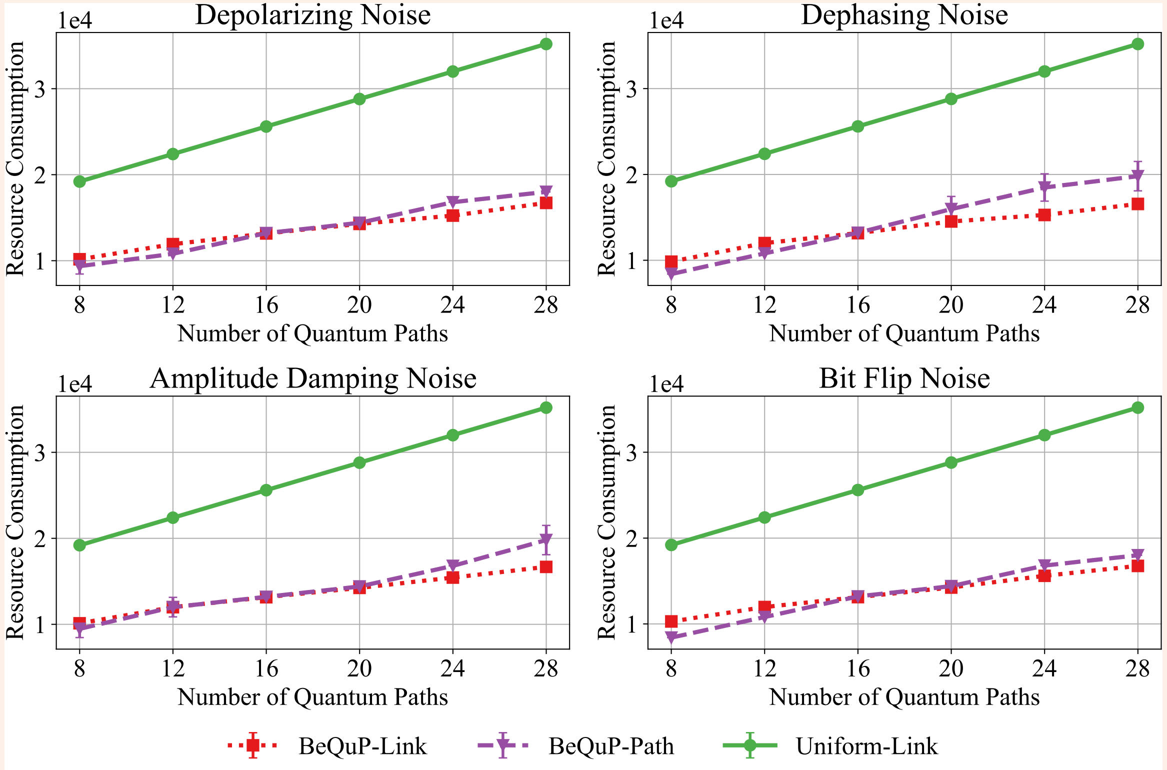


Figure 2: High-SKF Path Identification

Coopmans, Tim, et al. "Netsquid, a network simulator for quantum information using discrete events." *Communications Physics* 4.1 (2021): 164.

Learning Best Quantum Path Takeways

- Link-level: difference in **lower and upper confidence bounds**
- Path-level: **optimal experimental design** for link estimates
- Key idea: reduce from path [**coarse**-grained] **to** link [**fine-grained**]

Thank you!