

現実的な量子特性下における量子 ネットワーク経路選択の性能特性

立命館大学院 情報理工学研究科
先進ネットワーク研究室
Kim Minkyu 上山憲昭

研究背景

■ 量子ネットワーク

- 遠く離れた2地点間で量子通信を可能にする「**次世代の通信網**」

■ なぜ必要か

- 古典技術では到達できない「**安全性**」や「**計算能力**」を実現するため

■ 何ができるか(応用)

- 量子鍵配送 (QKD: Quantum Key Distribution)
 - 盗聴を検知でき、安全に鍵を共有
- 分散量子コンピューティング (DQC : Distributed Quantum Computing)
 - 単一の量子コンピュータは量子ビット数に物理的限界
 - 多数をネットワークで結び規模を拡張

研究背景

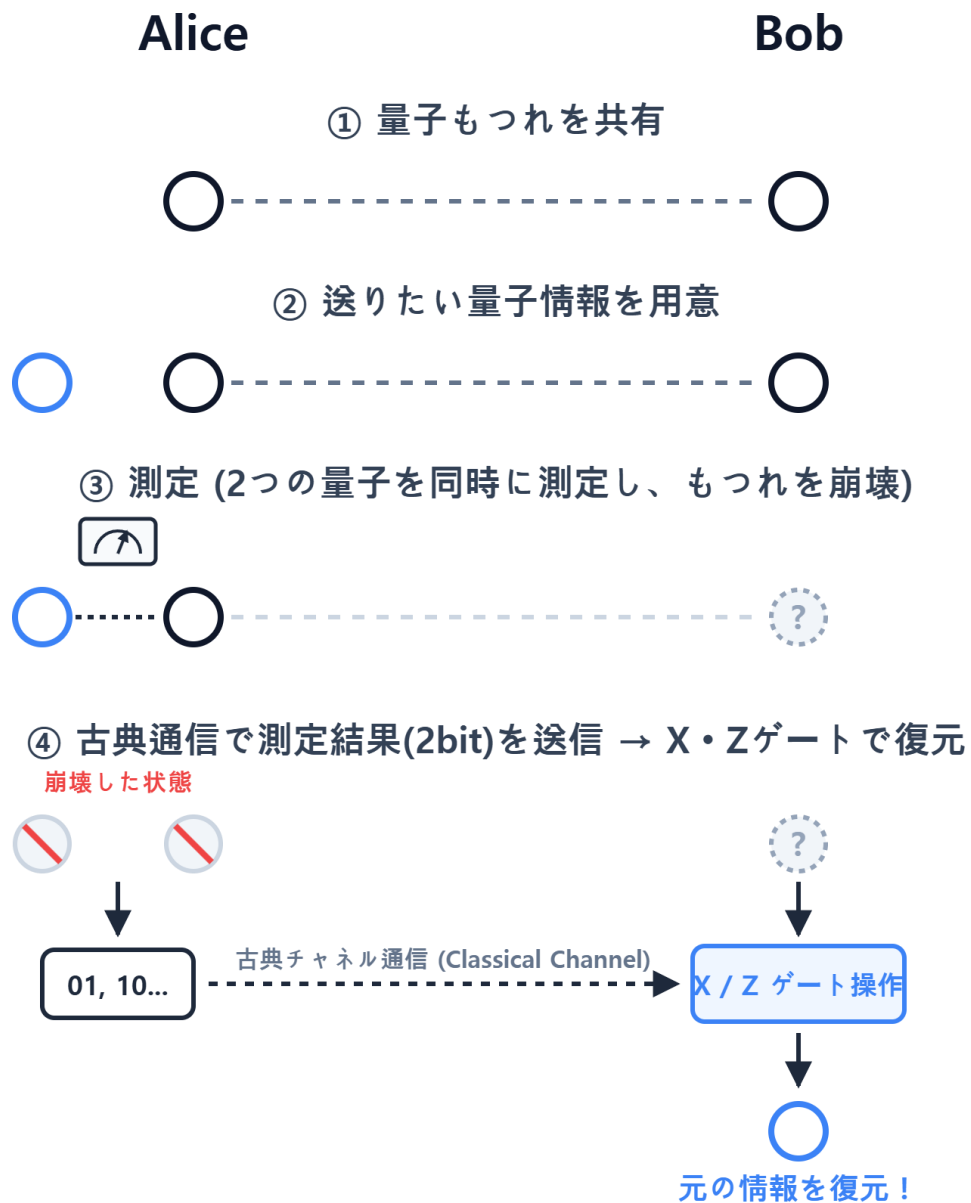
■ 量子もつれ(Entanglement)

- 隣接する2ノードを結ぶ「**使い捨て(一度きり)のリンク**」
- 普通の通信回線と違い、一度使うとなくなる
 - 通信のたびに作り直す必要がある
- このリンクがあって初めて、2ノードが量子的につながる

■ 忠実度(Fidelity)

- そのリンクの「**品質**」を表す指標($F=1$ が完全)
- 品質が低いと、情報が正しく伝わらない

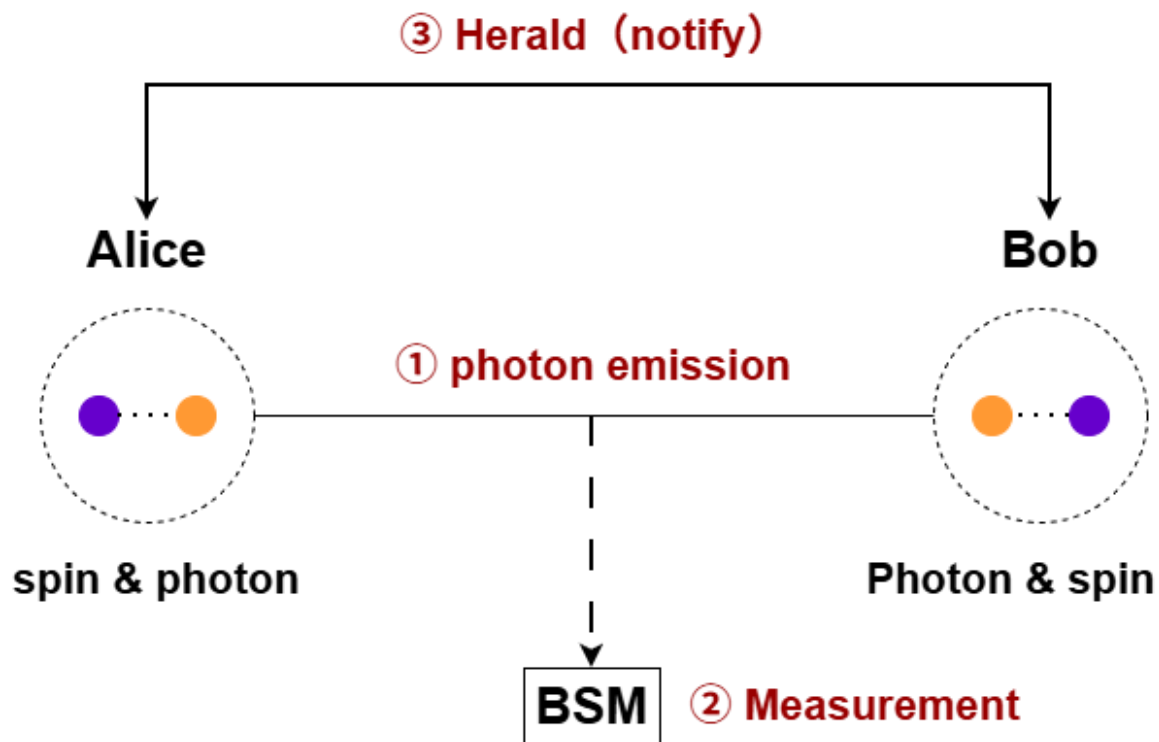
量子通信の原理



使い捨てリンクの問題

■ リンクの生成

- 中間地点でリンク生成を試み、成功を両ノードに通知
- ただし確率的 – いつも成功するとは限らない



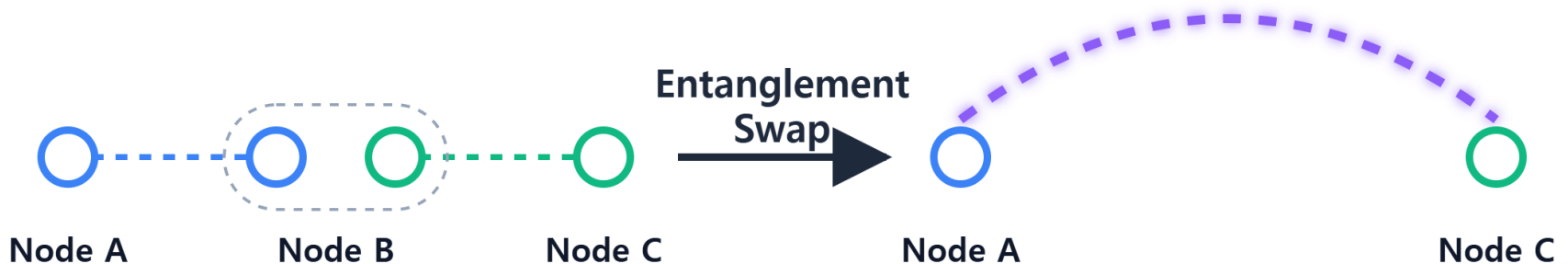
■ デコヒーレンス(Decoherence) ← **本研究の核心**

- 時間が経つとリンクが劣化し、使えなくなる

Entanglement Swapping & Purification

■ エンタングルメントルーティング(Entanglement Swapping)

- 中継ノードで、隣り合う2つのリンクを「**連結**」する
- End-to-Endまで届かせる技術
- ただし、つなぐほど品質は下がる



■ 精製(Purification)

- 品質の低いリンク2つを使い、良いリンク1つを作り直す
- ただし、失敗すると使った2つのリンクを失う

量子ネットワークにおけるルーティング課題

- エンタングルメントルーティング(Entanglement Routing)
 - 「もろいリンク」をつないで、End-to-Endを作る経路を選ぶ問題
 - 複数の問題が絡み合う：
 - **リンク生成**：確率的な生成、貯められる数にも限り
 - **経路探索**：何を基準に?「最短」か「品質」か
 - **E2E連結・品質維持**：どう連結し、品質を保つか(Swap・精製)
 - **デコヒーレンス(Decoherence)**：
時間で劣化し、ノードごとに状態が違う

関連研究

- エンタングルメントルーティングの既存研究
(Q-CAST[9], EFiRAP[10], DER[11] など)
 - 経路選択・End-to-Endの効率・品質の向上を扱う
- しかし、量子条件を理想化している
 - リンクの生成：確率に作られることを十分に扱わない
 - デコヒーレンス：時間的な劣化を反映しない
 - ※Chaudhary[2], Taherpour[3]はデコヒーレンスを扱うが、リンクの物理的な作り方は抽象化している
- **物理的な作り方まで含めて、複数の手法を比較した現実的な評価は乏しい**

研究目的

■ 動機

- 既存のルーティング研究では、デコヒーレンスの影響を十分に考慮した定量的な分析が乏しかった

■ 研究目的

- 先行研究[4]の現実的なプラットフォームを継承
- **デコヒーレンス On/Off条件**で、複数の経路選択手法の挙動を分析
- Dijkstraをbaselineに追加して比較

評価環境：Proactive Link Preparation

■ 量子資源の事前生成・準備

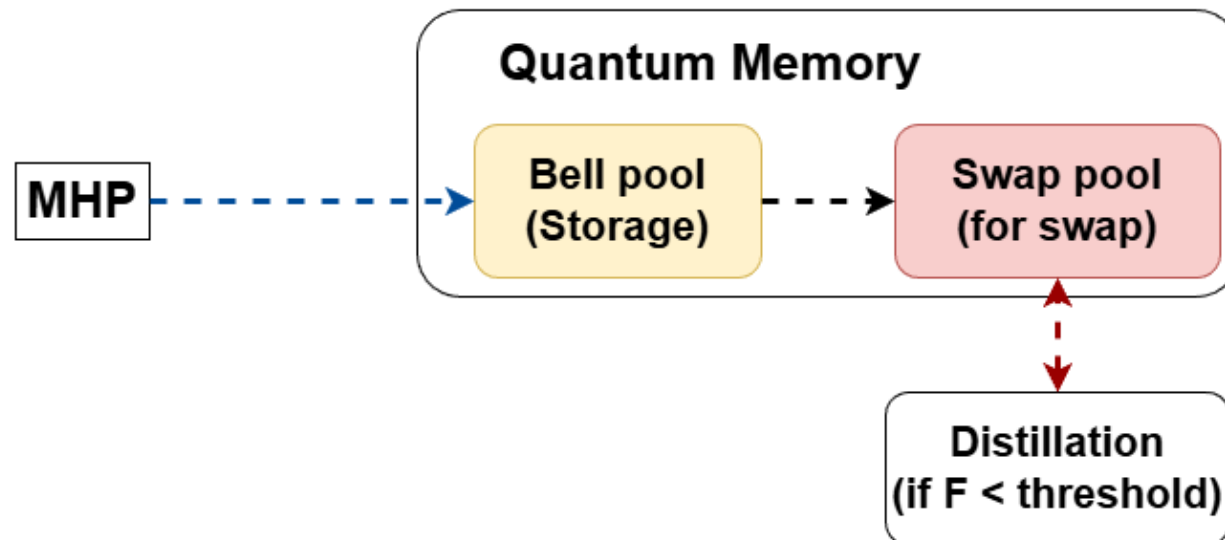
■ 手順

- MHPによってBell pair **自動生成**
- リンク単位で**独立・継続的**に実行

■ 量子メモリ

- 2poolに論理的に分離
 - Bell pair pool : pair管理用
 - swapping pool : スワッピング・精製用

Proactive Preparation



評価シナリオ

■ シミュレーション条件

- シミュレータ
 - NetSquid (QuTechの量子ネットワークシミュレータ)
- トポロジ
 - Waxmanモデル (20ノード, 35リンク)
- トラフィック到着率
 - 前半(0-30秒): 10 req/s → 後半(30-60秒): 100 req/s
 - **中間時点で1回変更**
- シミュレーション時間
 - 60秒
- 候補経路数
 - $k=3$
- E2E試行回数
 - 6回/要求 (3候補 × 2回)

評価対象：経路選択手法

■ 経路選択手法の比較(5手法)

■ 経路候補を用いない(最短経路固定)

比較手法	特徴
Dijkstra	最短ホップ経路を選択(baseline)

■ 経路候補から選択(k=3)

比較手法(非適応)	特徴
Round-Robin	候補経路を順番に選択
Random	候補経路をランダムに選択

比較手法(適応:MAB)	特徴
Thompson Sampling	Beta分布による確率的選択
UCB1	平均報酬＋探索項による決定的選択

評価指標

■ 評価指標

■ 基本指標(構成要素)

- $1_{success}$: E2E接続の成否(成功=1, 失敗=0)
- F_{E2E} : End-to-Endリンクの忠実度(Fidelity)

■ MAB報酬

- $Average\ Reward = 0.5 \times 1_{success} + 0.5 \times F_{E2E}$

■ 性能・挙動の分析(グラフで比較)

- **Success Rate** ← $1_{success}$ から導かれたE2E接続の成功率
- **Path Length** ← 経路選択の傾向

性能評価(全体)

■ Decoherence OFF

- 適応的手法(**UCB1・Thompson**)が優位
- ただし、baselineの**Dijkstraとの差は小さい**

表 3: 性能比較 (デコヒーレンス OFF)

手法	平均報酬	E2E忠実度	成功率 (%)	平均経路長
UCB1	0.3216	0.975	32.6	3.05
Thompson	0.3235	0.975	32.8	3.04
Round-Robin	0.2487	0.974	25.2	2.38
Random	0.2402	0.971	24.4	3.25
Dijkstra	0.3101	0.980	31.3	2.35

■ Decoherence ON

- 全体的に性能低下
- 手法間の差が**圧縮**され、ほぼ同等
- 非適応の**Round-Robin**が上位
- ThompsonとDijkstraの差が消失

表 4: 性能比較 (デコヒーレンス ON)

手法	平均報酬	E2E忠実度	成功率 (%)	平均経路長
UCB1	0.1042	0.789	11.7	2.22
Thompson	0.0978	0.786	10.9	2.34
Round-Robin	0.1056	0.788	11.8	2.09
Random	0.0876	0.786	9.8	2.33
Dijkstra	0.0974	0.795	10.8	1.93

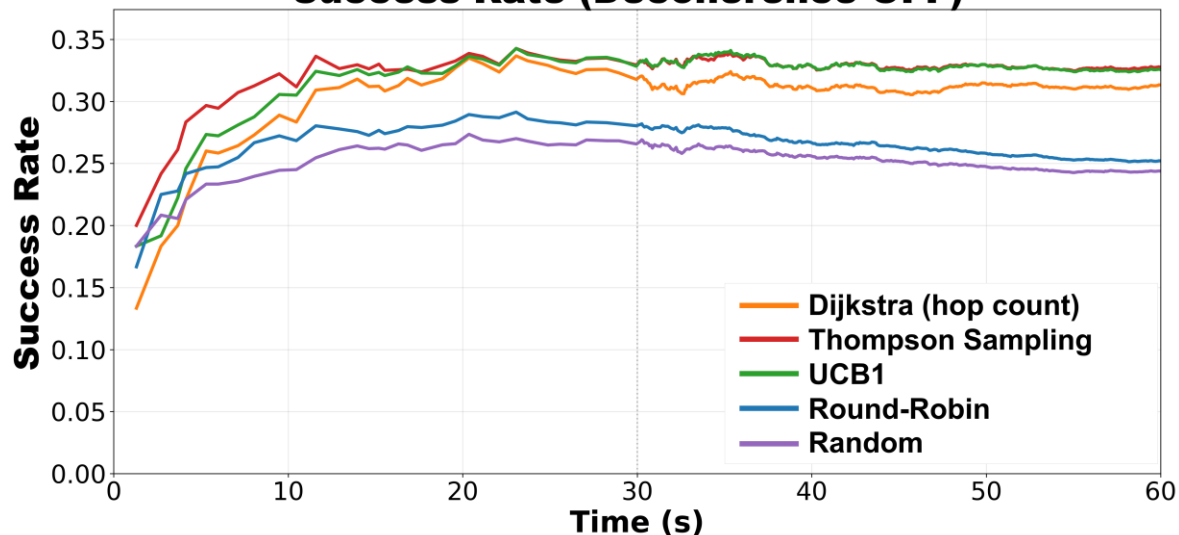
性能評価2

■ Success Rate

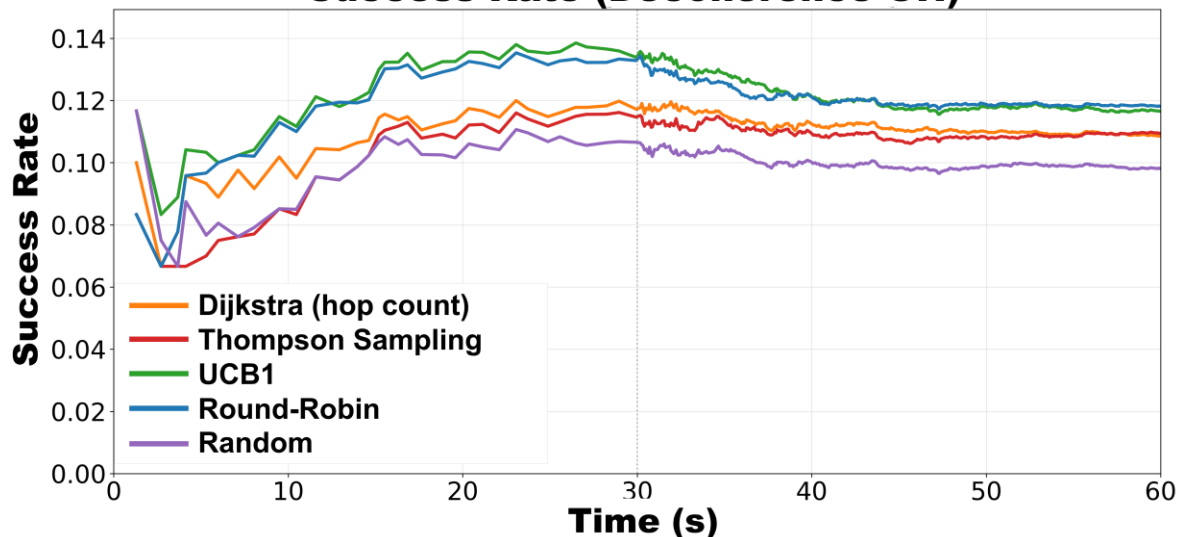
■ OFF :
手法間の差が明確
(UCB1・Thompson・
Dijkstraが上位、
RR・Randomが下位)

■ ON :
全体が低下し、
手法間の差が圧縮

Success Rate (Decoherence OFF)



Success Rate (Decoherence ON)



■ 適応的手法の優位が、Decoherenceで薄れる

性能評価3

■ 平均経路長

■ OFF :

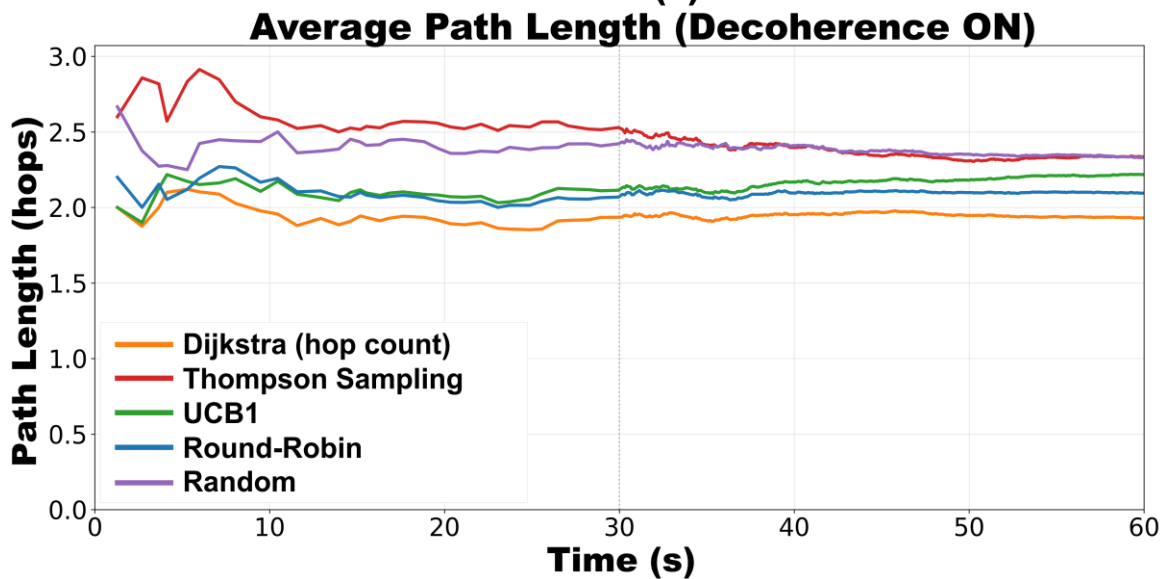
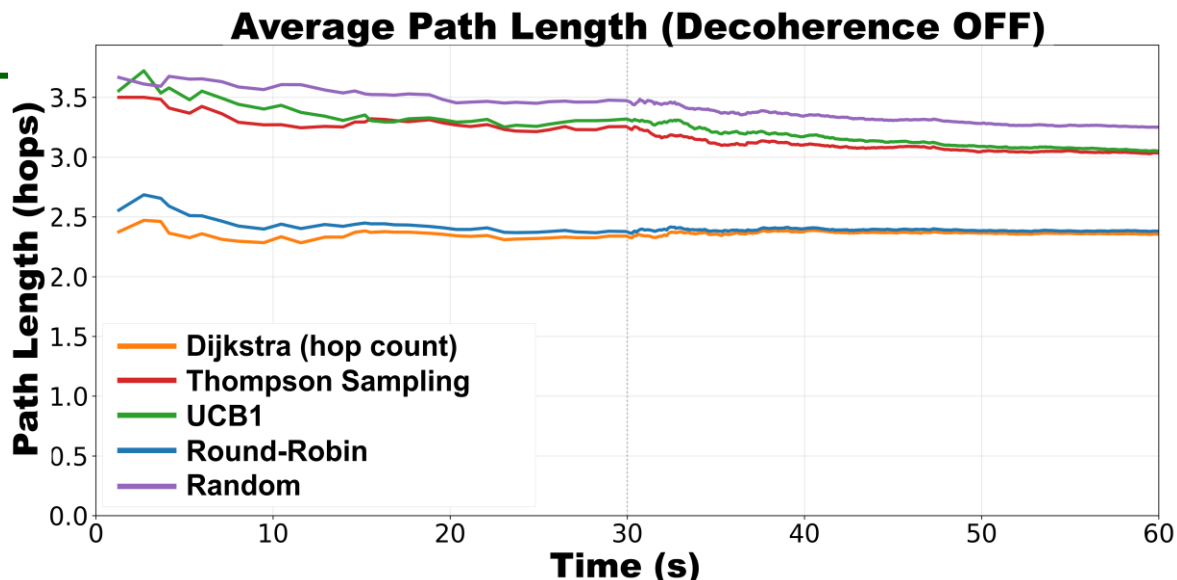
経路長が手法ごとに分散

(MAB・Randomは長め、Dijkstra・RRは短め)

■ ON :

全体が短くなり、差も縮小

■ Decoherenceで長い経路が使えなくなり、経路選択の自由度が失われる



まとめ

■ まとめ

- 先行研究[4]の**現実的なプラットフォーム**を継承し、**デコヒーレンスOn/Off**条件で経路選択手法を比較・分析
- Decoherenceは**性能を大きく低下**させ、手法間の差も圧縮する
- 経路制御の設計・評価には、**デコヒーレンスの考慮が不可欠**

今後の課題

■ 今後の課題

- 経路最適化の**効果は限定的**
 - 経路制御と「連結(Swap)」の統合的最適化へ
- 自律的な「もつれのセグメント」形成への拡張
 - 要求とは独立に、ノードが自律的に連結(Swap)を先行実行し、マルチホップの区間を事前に形成・維持する