Entanglement Routing for Quantum Networks

Chen Qian

Department of Computer Science and Engineering
University of California, Santa Cruz

My background

- BSc Nanjing U, 2006
- MPhil HKUST, 2008, with Lionel Ni
- PhD UT Austin, 2013, with Simon Lam
- 2013-2016, Assistant Professor, University of Kentucky
- 2016-present, Assistant Professor -> Associate Professor -> Professor, University of California Santa Cruz

With quantum

- Started from undergrad days
- Continued in 2018
- My PhD student Shouqian Shi

fcst.ceaj.org › abstract › abstract654 · Translate this page
量子程序设计语言NDQJava2处理系统−−汇编程序与解释程序
量子程序设计语言NDQJava2处理系统−−汇编程序与解释程序. 朱正文1,2，徐家福1,2+. 1. 南京大学软件新技术国家重点实验室，南京210046 2. 南京大学计算机 ...

www.thepaper.cn › newsDetail_forw... ▾ Translate this page
南大校媒悼念徐家福教授：80岁转行，拜师8位量子 ... - 澎湃新闻
Jan 17, 2018 — 徐家福教授，博士生导师，中国计算机科学和计算机软件学专家。江苏南京人。1948年毕业于中央大学（今南京大学）。1981年任南京大学 ...

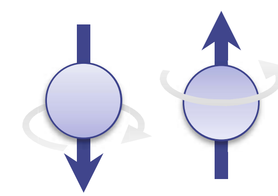
cst.qdu.edu.cn › info ▾ Translate this page
徐家福教授作“量子程序设计语言初探”学术报告-青岛大学计算机 ...
Dec 20, 2011 — 11月28日上午，南京大学计算机系教授徐家福先生应邀来我院作了题为“量子程序设计语言初探”的学术报告。 量子计算机是目前世界上极具挑战性 ...

k.sina.cn › ... · Translate this page
南大校媒悼念徐家福教授：80岁转行，拜师8位量子力学 ... - 新浪
Jan 17, 2020 — 2004年，徐家福先生以80岁高龄转行研究量子计算，从量子力学读起，拜8位量子力学专家为师。他于2006年和2007年开发出两种量子程序设计 ...

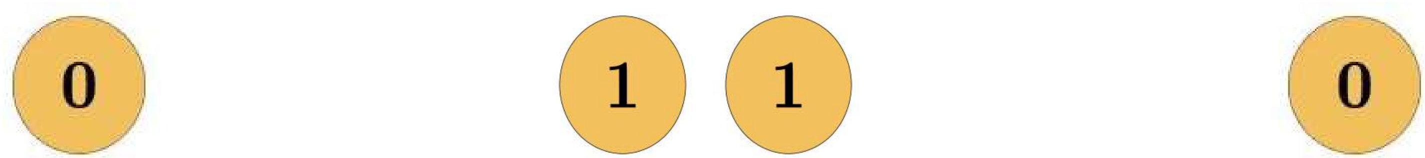
new.qq.com › omn ▾ Translate this page
“无冕院士”徐家福：中国软件事业奠基人，80岁高龄转行从事 ...
Mar 21, 2019 — 80岁高龄转行研究量子计算. 1946年，中央大学迁回南京，徐家福回到了阔别十年的故乡。也就是在这一年，世界上第一台计算机“埃尼阿克”在 ...

books.google.com › books › about · Translate this page
量子程序设计语言- 徐家福, 宋方敏- Google Books
本书共分10章,陈述了研习量子程序设计语言所需之基础知识,即数学基础,计算机科学基础,量子力学基础;简要介绍了量子伪码,Qgol,QCL,qGCL,QML五种量子程序 ...

Quantum entanglement



- State correlation of two particles (qubits)
  - Reveal *both* by revealing *either*
  - Even separated by a large distance



- Establish an entanglement
  - Send photon via optical fiber
  - May try multiple times

Applications

- Data bit transmission: ***perfectly secure***
  - Need classical communication: authenticity and integrity
    - Data bit to send   0: “the same”   1: “the opposite”
    - Perfect confidentiality is guaranteed

- Quantum teleportation, remote quantum key distribution (QKD), distributed consensus, *etc.*

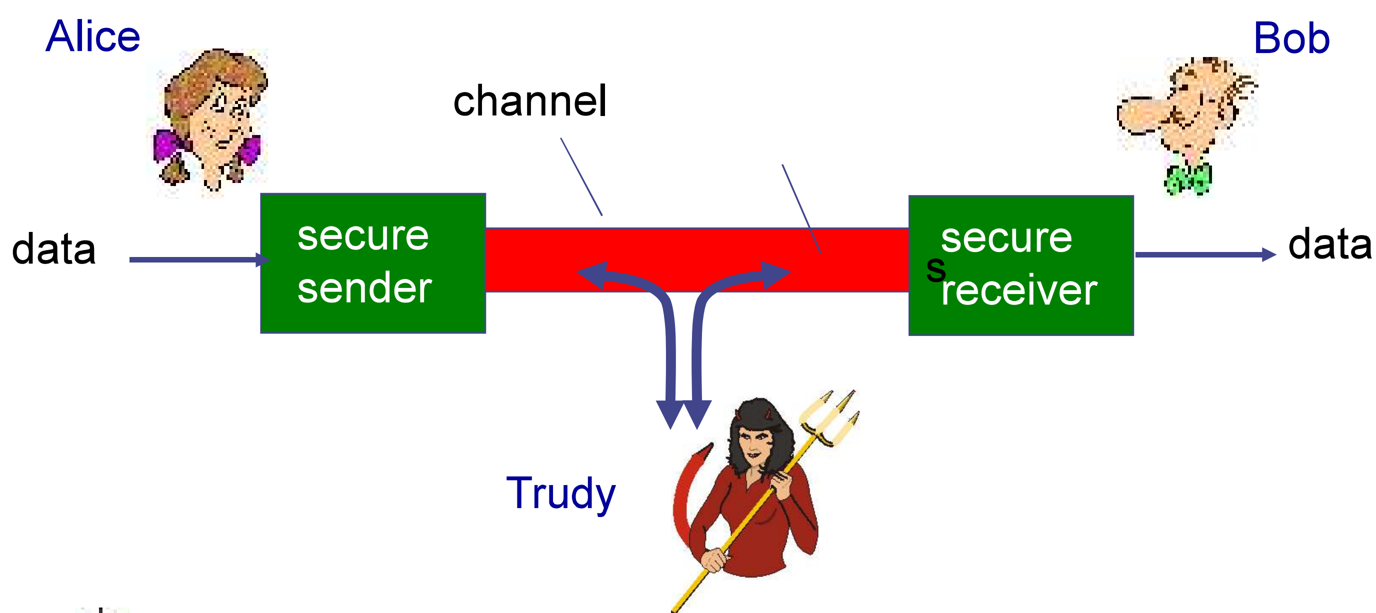
No cloning theorem

- Perfect copies of an unknown quantum state cannot be made!

Quantum key distribution (QKD)

- Existing key distribution methods rely on
  - Complexity of integer factorization
- QKD uses quantum mechanics to detect the presence or absence of an eavesdropper.
- How?
  - If Eve (eavesdropper) attempts to learn the data value, it needs to measure the quantum states
    - The measurement results in collapse of quantum state and is revealed to Alice and Bob.

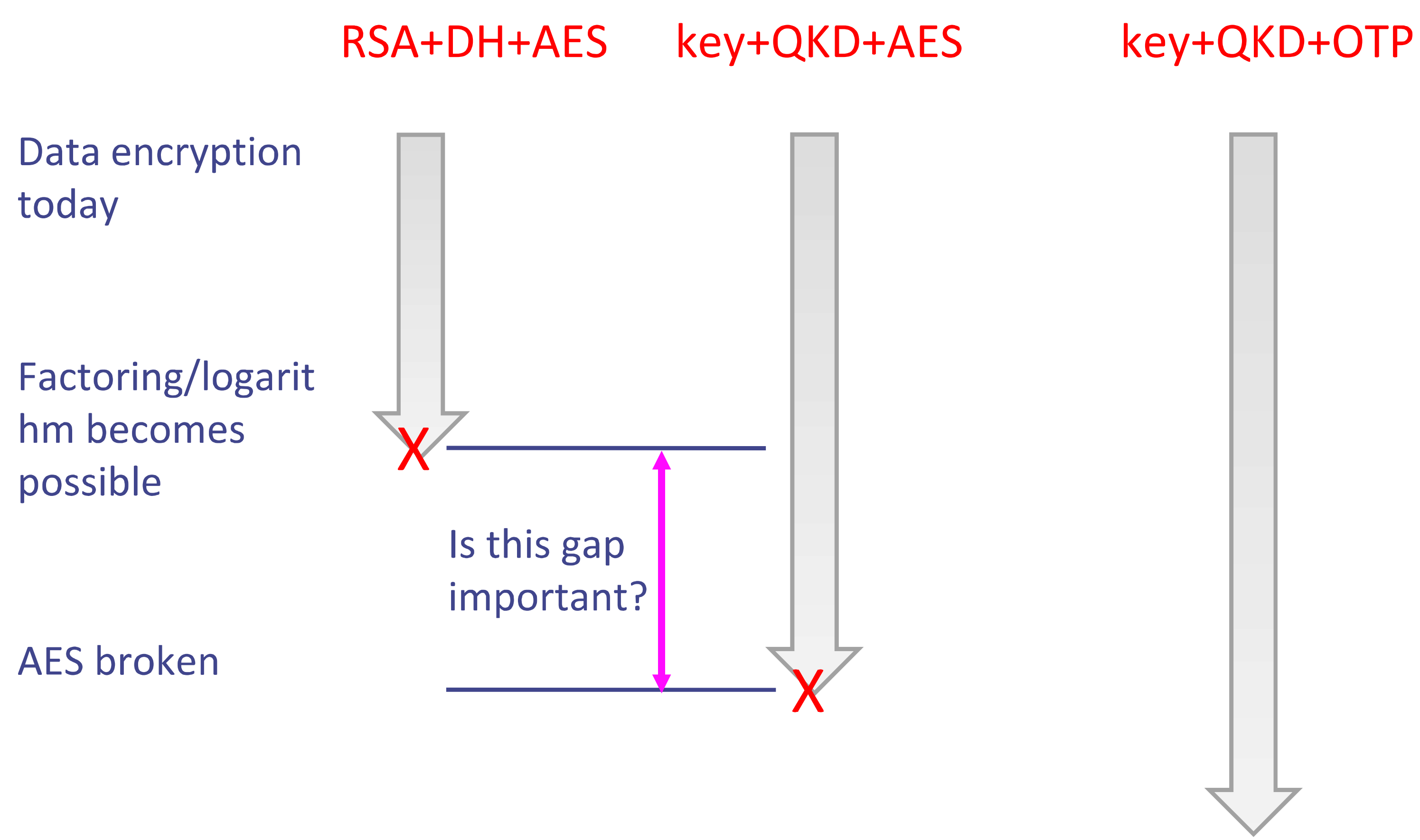
Steps of building a secure channel



- 1. Authentication
- 2. Key agreement
- 3. Encrypted data transfer

Authentication methods	Attacking methods
1. Pre-shared keys	Brute force
2. Public key crypto (RSA)	Factorization
Key agreement methods	
1. Diffie-Hellman	Discrete logarithm
2. Public key crypto (RSA)	Factorization
Data encryption methods	
1. One-time pad	
2. Block cipher (AES)	Brute force (Try all keys)

QKD only solves key agreement!



RSA+DH+AES
key+QKD+AES
key+QKD+OTP
Data encryption today
Factoring/logarithm becomes possible
Is this gap important?
AES broken

Information sensitivity lifetime

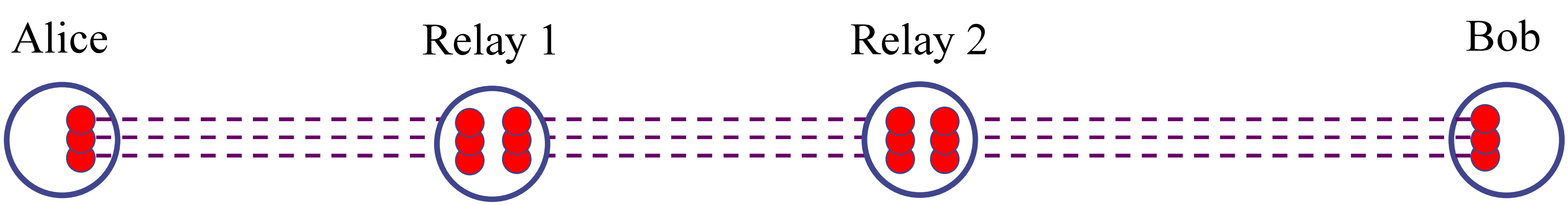
- Many communications are real-time
  - Financial transaction
  - Real-time control
- Retroactive vulnerability
  - Decryption of old stored communication, such as the contents of diplomatic cables
- QKD is used to protect retroactive vulnerability
  - By replacing D-H or RSA

“Open” and “close” communication

- Open: the communication parties do not need to pre-determined
  - Just like the Internet
- Close: there are a closed group of communication parties
  - They hold pre-shared keys
  - QKD can only work for this type of communication
- Why?

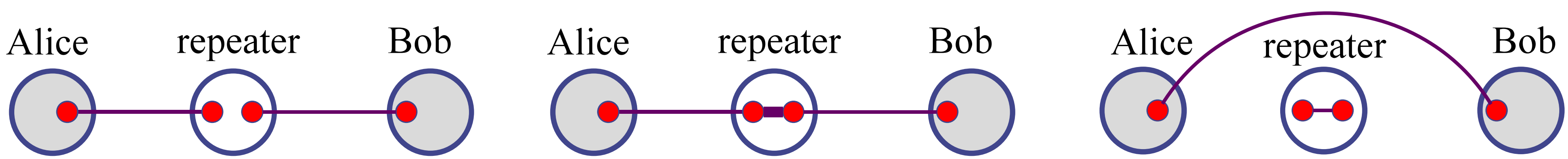
Establish remote entanglements

- Success rate decreases exponentially with channel length
- Solution
  - Parallel channels
  - Relays
    - Relays are called repeaters
    - Trusted or untrusted



Establish remote entanglements via repeaters

- Workflow
  - Build local entanglements
  - Bell state measurement (BSM)
    - With classical communication: repeater to A and B
  - 2-hop entanglement
    - with transitivity



(a) Prepare locally (b) Bell state measurement (c) After BSM

Limitations in quantum networks

- Entanglements decay fast
  - 1.5 sec ~ 1 min

- Solution: synchronized communication model
  - slotted timeline, slots are independent
  - 4 phases per slot

- High resource contention
  - # of qubits in a node
  - quantum channels to neighbors
  - limited in #
  - allocated exclusively

Building multi-hop entanglements

$\leftrightarrow$

Routing in the quantum network
w/ time and resource limitations

Key differences *vs.* classical networking

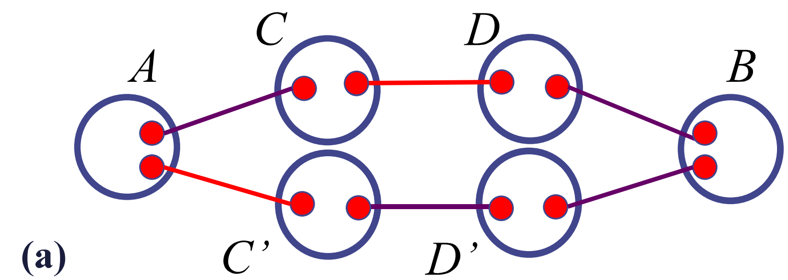
- **Slotted** time: cannot retransmit failed “packet”
- **Fixed assignments**: qubits to channels and qubits for BSMs

- Static and dynamic link states
  - Stable topology (nodes, channels, and qubits)
  - Probabilistic entanglement results: local path recovery

- **Non-additive** routing metric for paths
  - Parallel channels and flexible choices for BSMs

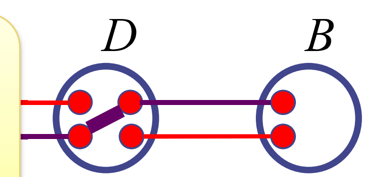
Non-additive routing metric for paths

- Non-overlapping paths
  - Fail at any channel



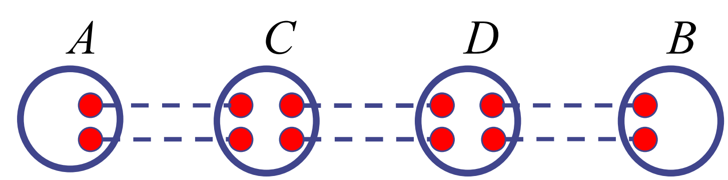
(a)

- Parallel paths
  - Recover



Intuition: prefer wider paths

- Path: width and hops
  - ⟨2,3⟩-path: 2 parallel channels, 3 hops



- General: non-additive

Existing algorithms not suitable

- Packet switching: Link state, distance vector, etc.
  - hop-by-hop
- Multihop wireless network: DSR, AODV, etc.
  - rely highly on probing and retransmission
- Like circuit switching, but
  - further actions required for dynamic link states

Goal

Maximize entanglement delivery in each time slot

(throughput)

Q-PASS

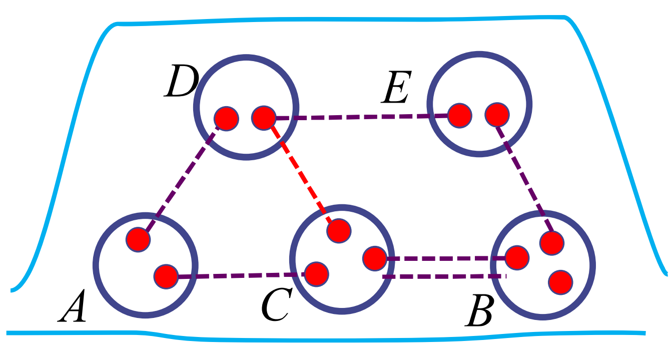
offline routing, segment-based path repair

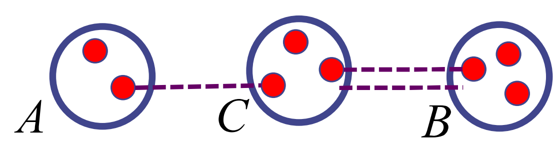
Q-CAST

online routing, $\oplus$-based path repair

Q-CAST: **C**ontention-free p**A**th **S**election at run**T**ime

- Contention-free
  - Any two paths don’t share any qubit or channel
- Extend Dijkstra algorithm for non-additive metric
  - Work for monotonic metrics, optimized speed
  - Monotonic: adding one hop, the path gets worse





Q-CAST Contention-free pAth Selection at runTime

P1: initialize

Receives src-dst pairs

← Node pairs to deliver entanglements: 3↔11 4↔14

P2: entangle locally

select paths based on topo and src-dst pairs

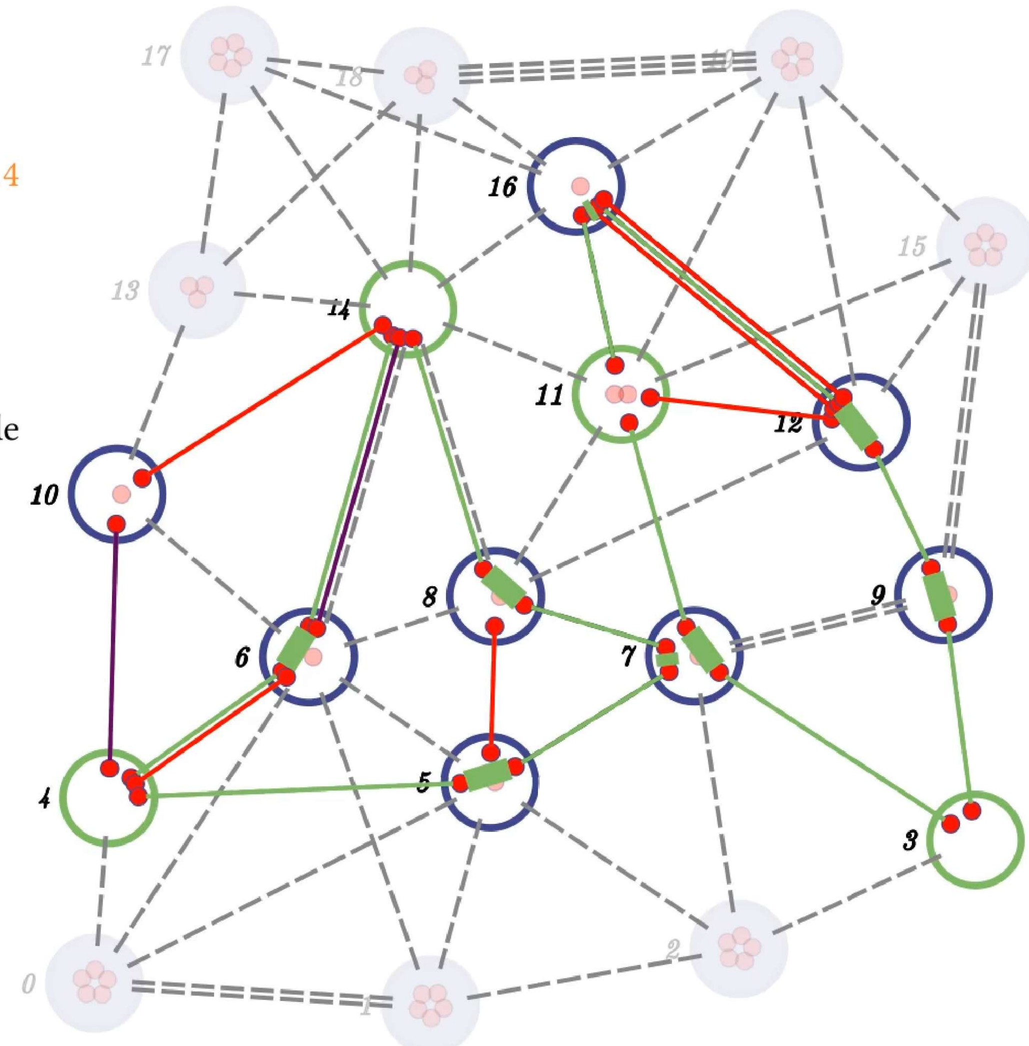
- ✓ The best path for each pair
- ✓ Pick the best of the bests
- ✓ Occupy qubits and channels
- Restart until no path available
- ✓ Partial paths for recovery

P3

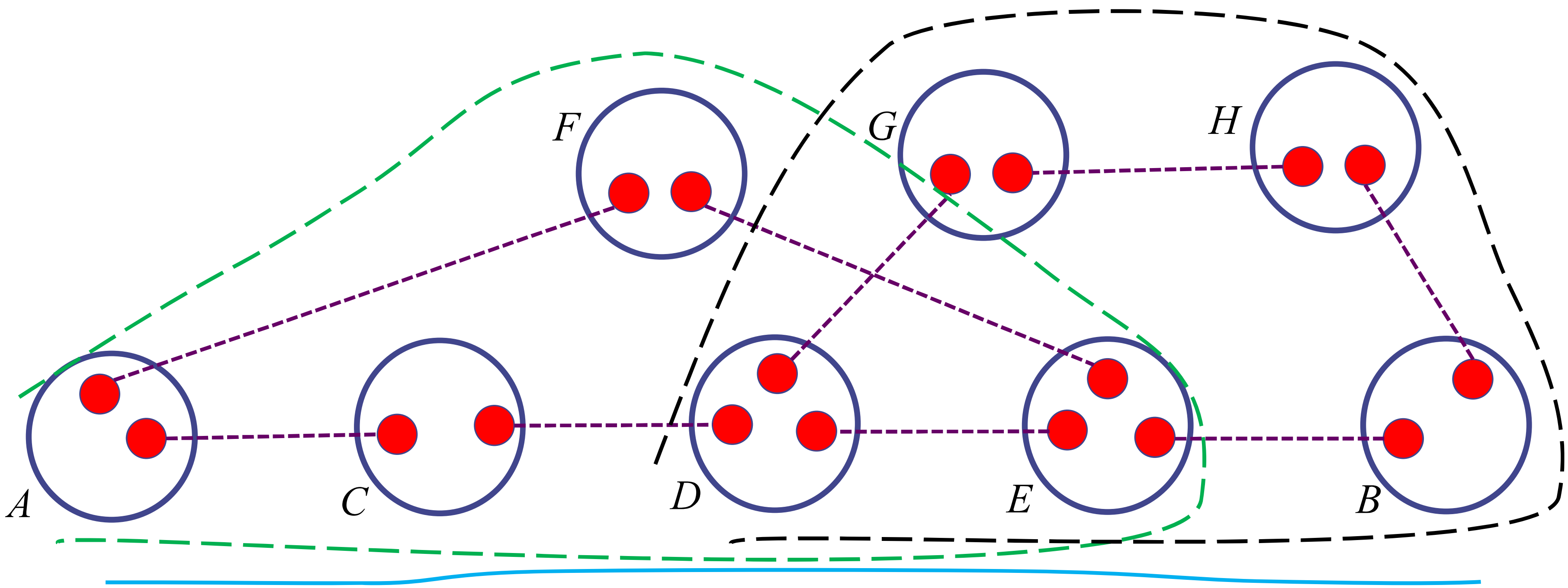
Exchange link state within k-hop

P4: place internal link

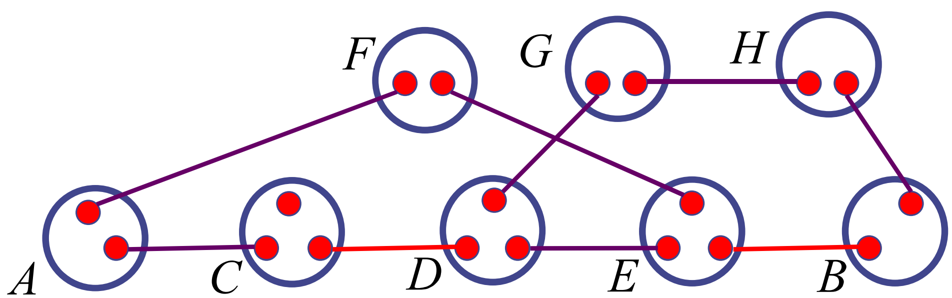
Connect existing local entanglements

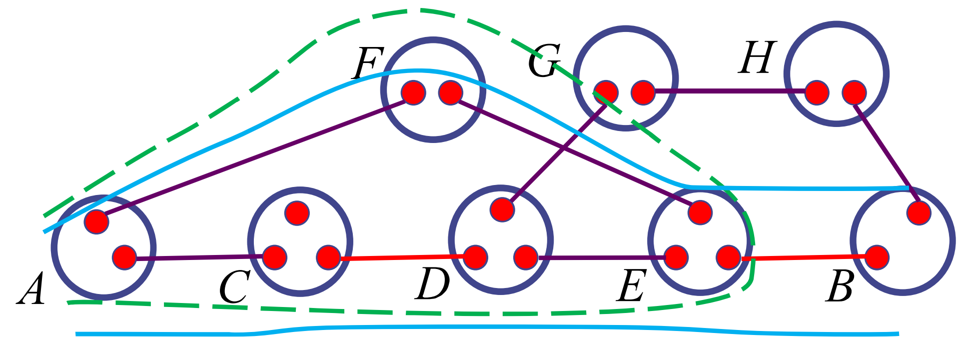


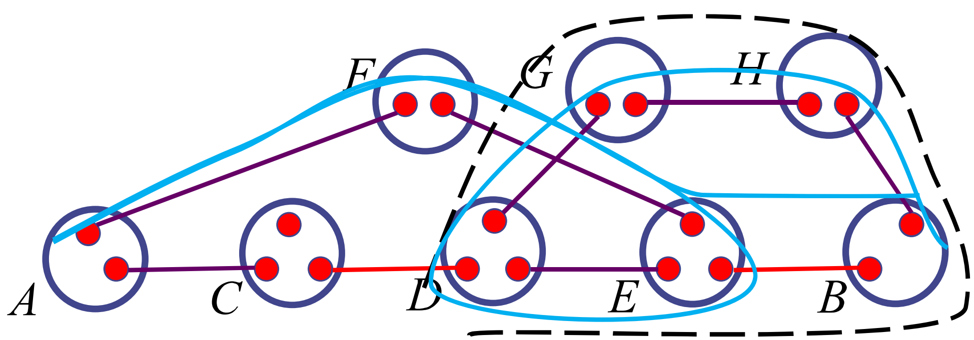
Q-CAST P4: $\oplus$-based path repair

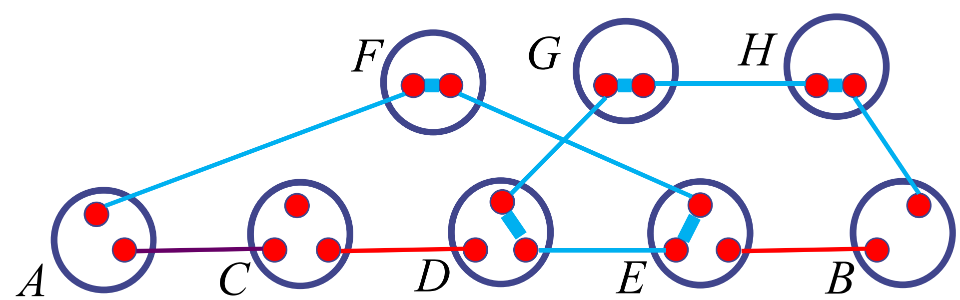


Q-CAST P4: ⊕-based path repair



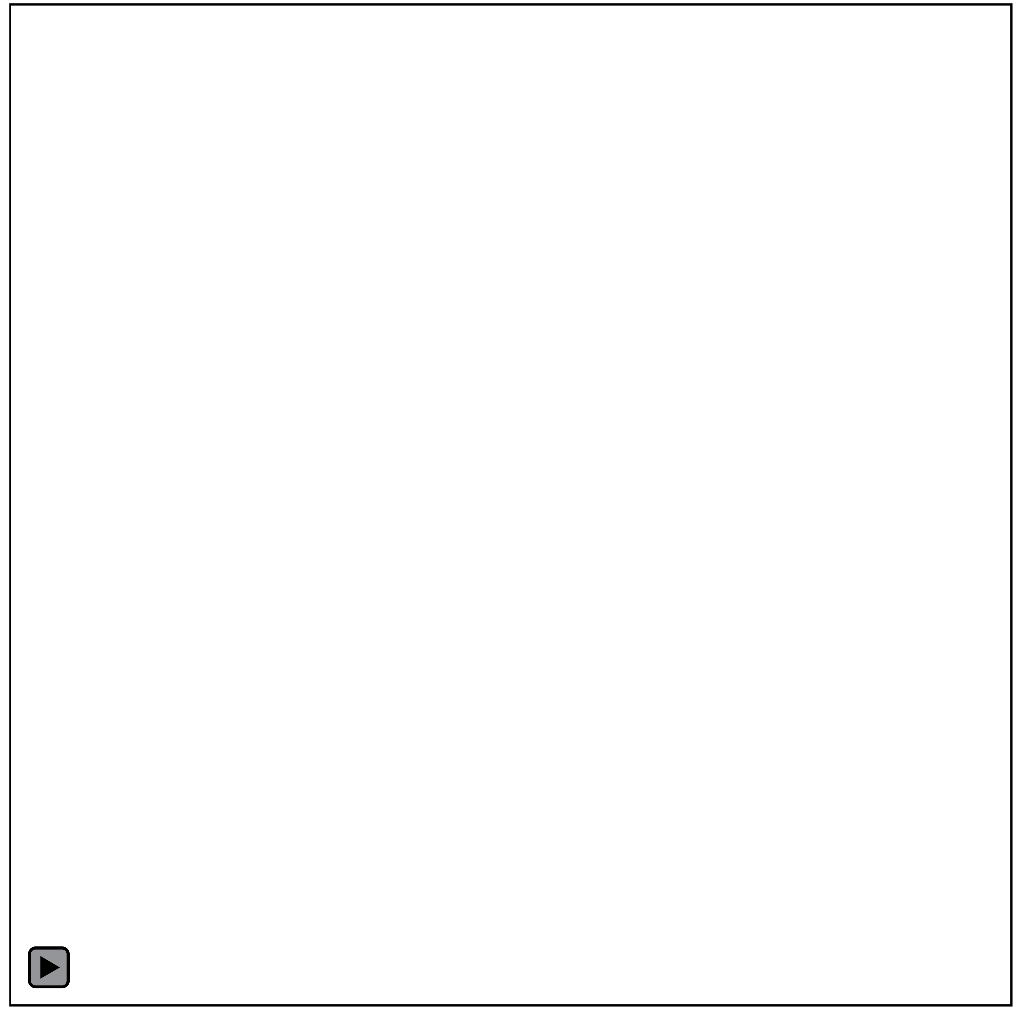




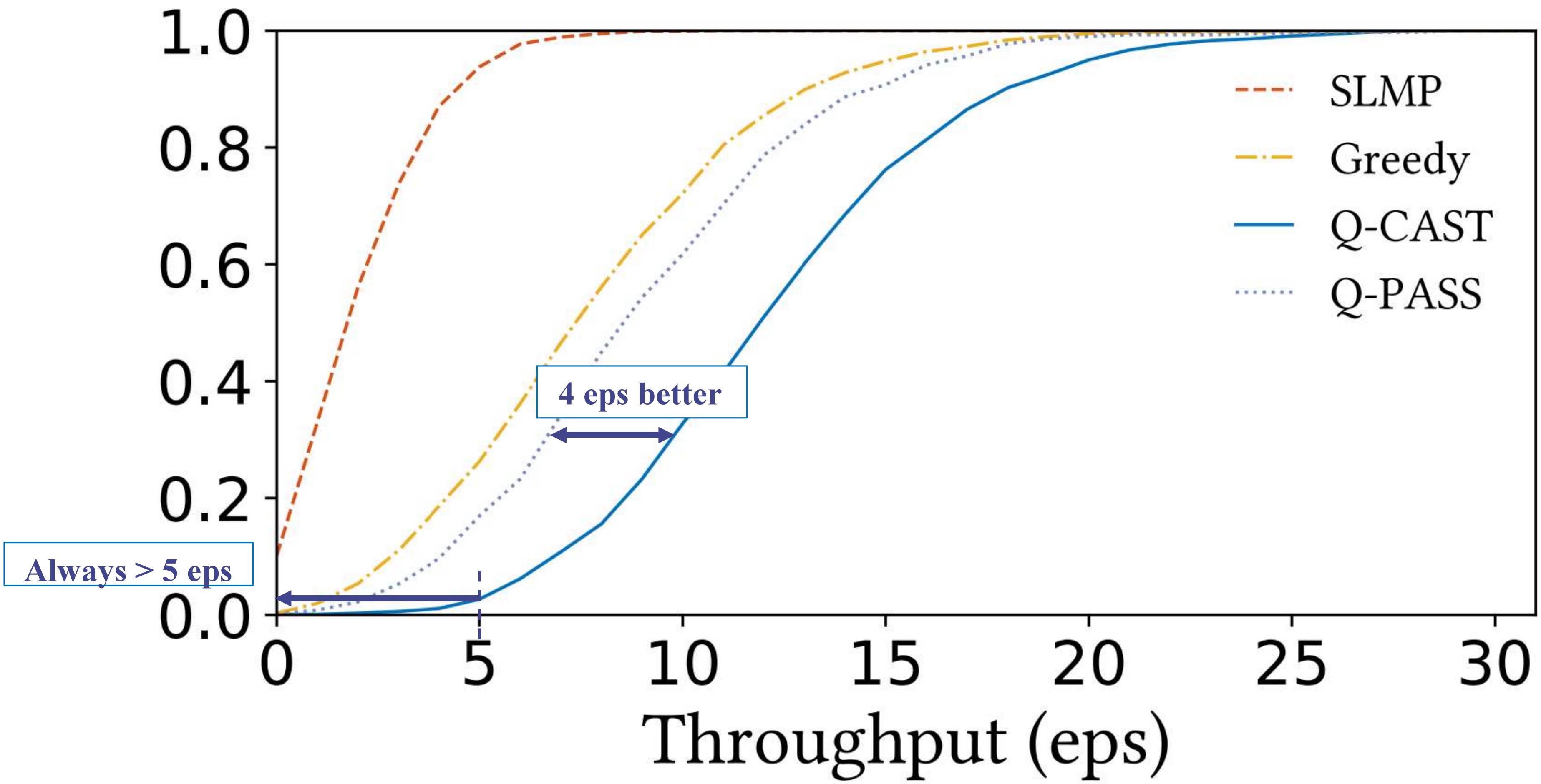


Evaluation

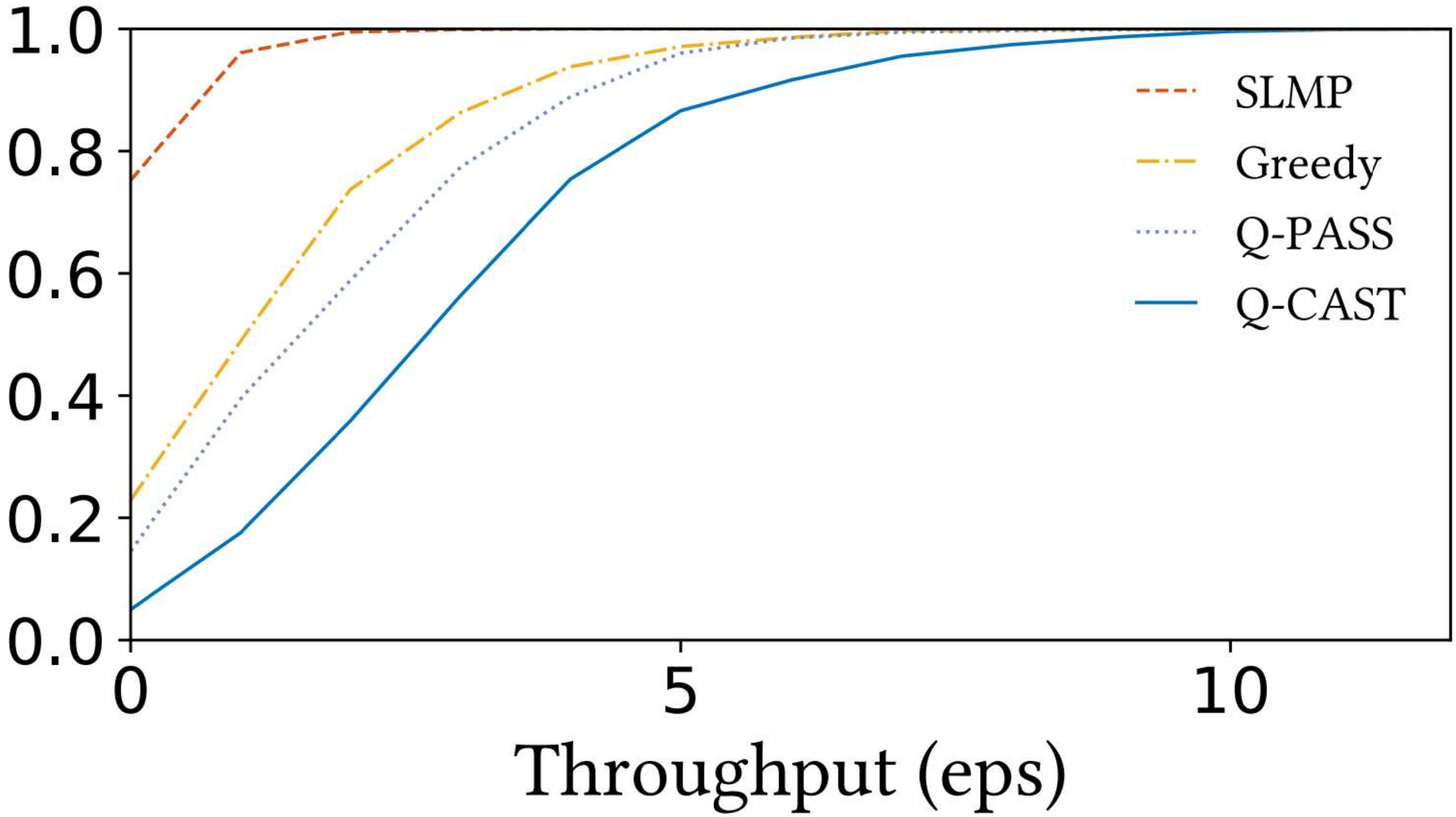
- Algorithms
  - Q-PASS   Q-CAST
  - Greedy   SLMP
- Evaluate
  - Throughput
  - Recovery algorithms



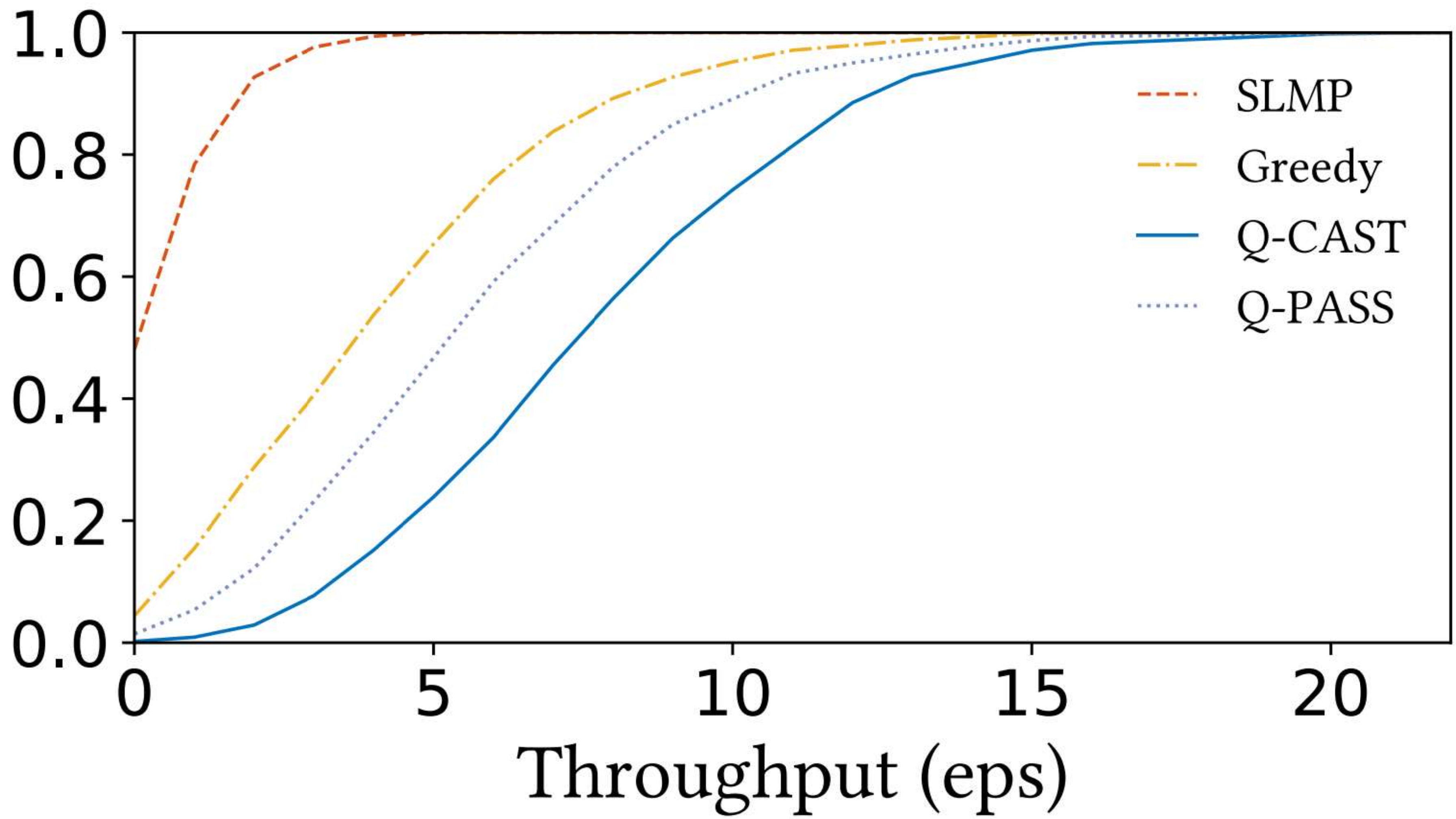
- Reference setting $n = 100, E_p = 0.6, q = 0.9, k = 3, E_d = 6, m = 10$



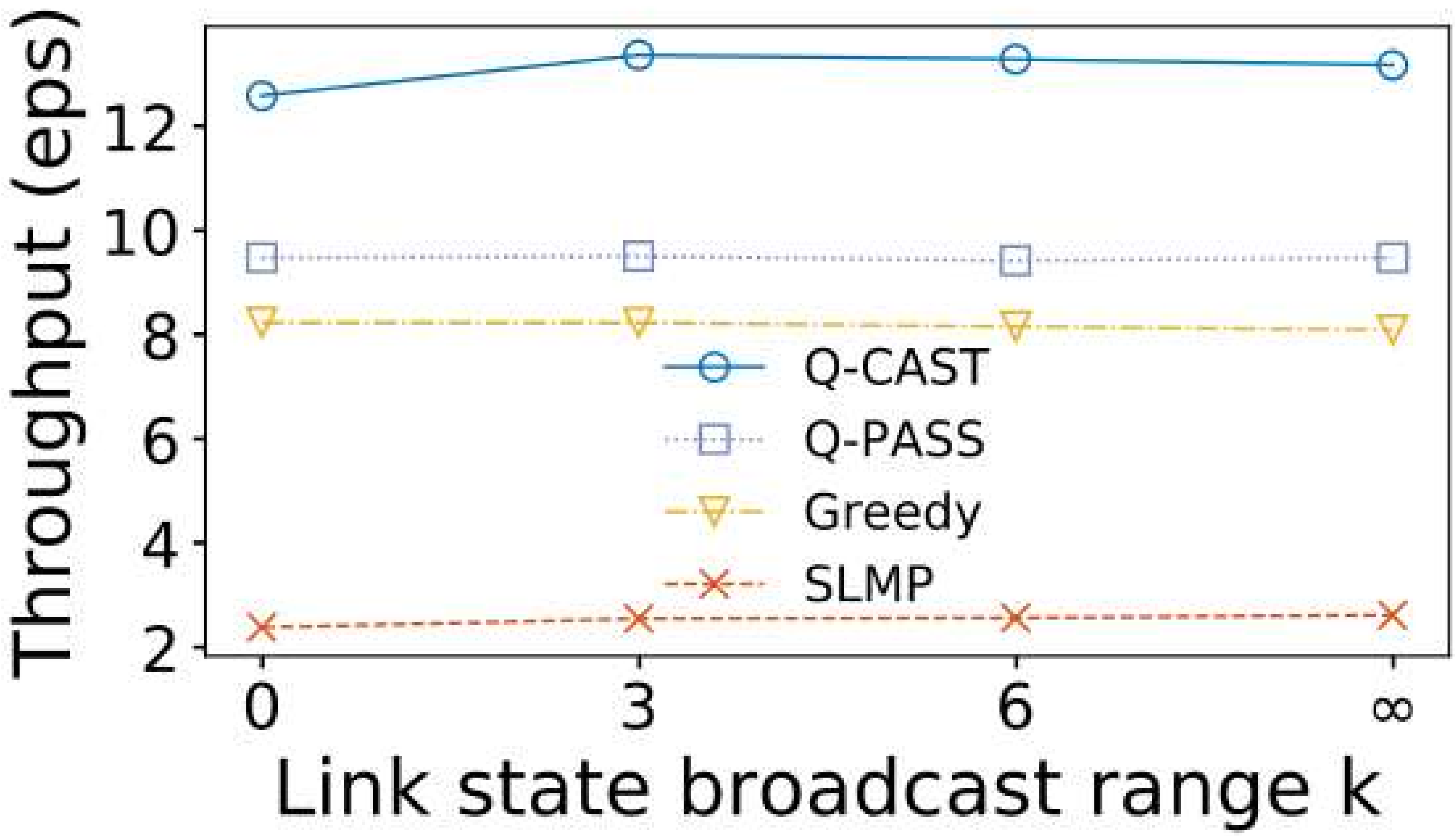
reference setting n = 100, E_p = 0.6, q = 0.9, k = 3, E_d = 6, m = 10

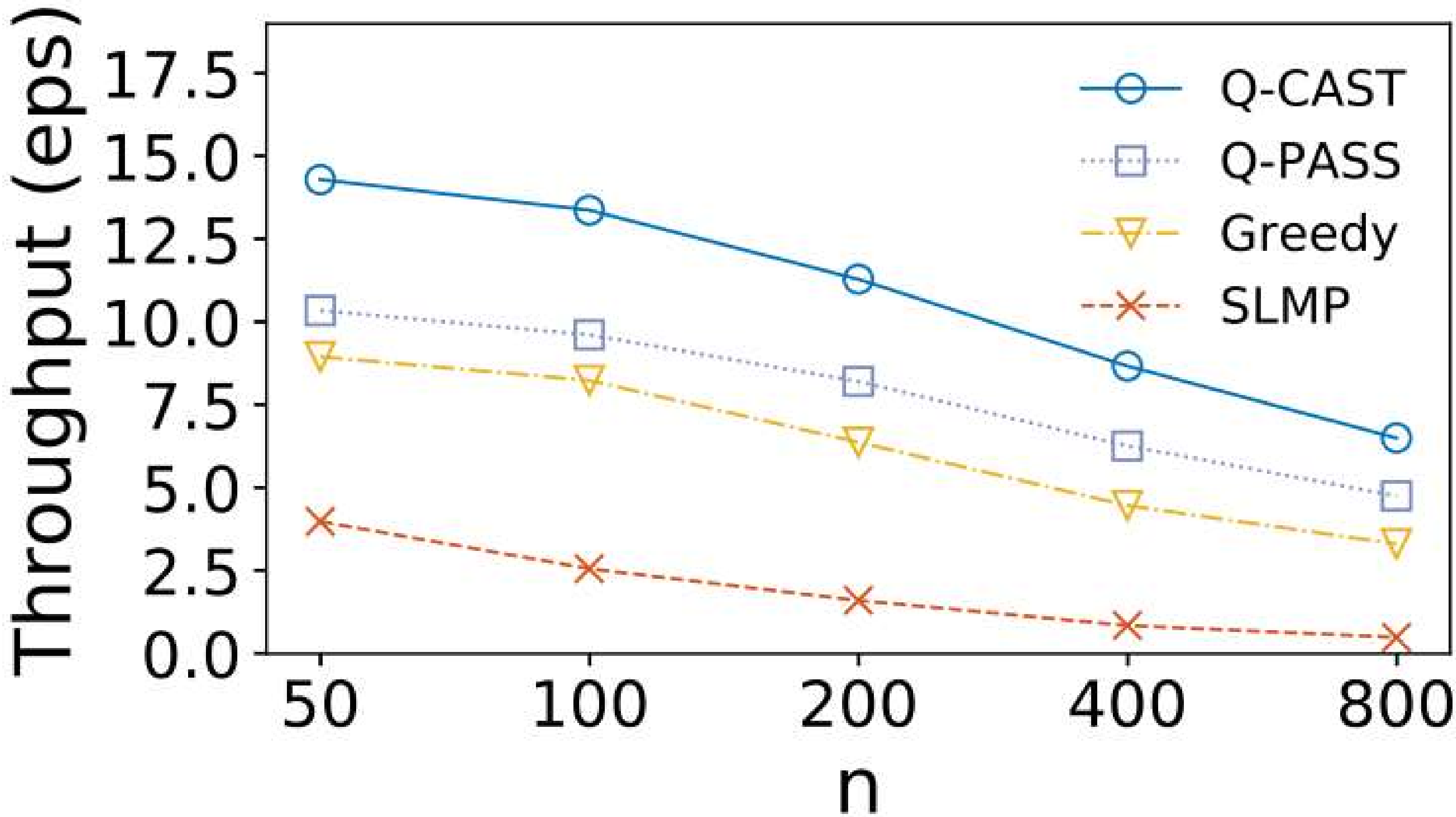


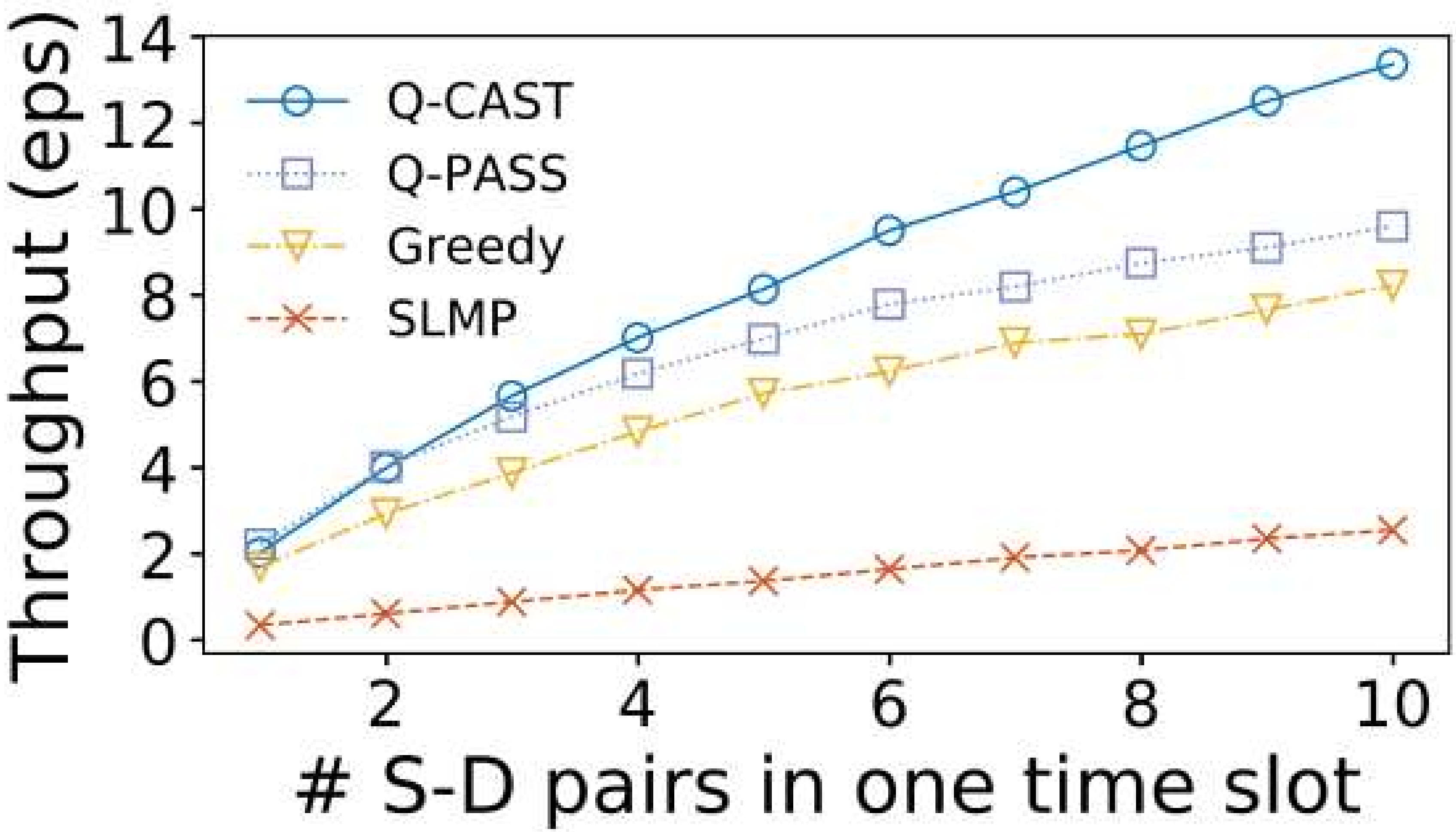
$n = 100$, $\boldsymbol{E_p = 0.3}$, $q = 0.9$, $k = 3$, $E_d = 6$, $m = 10$

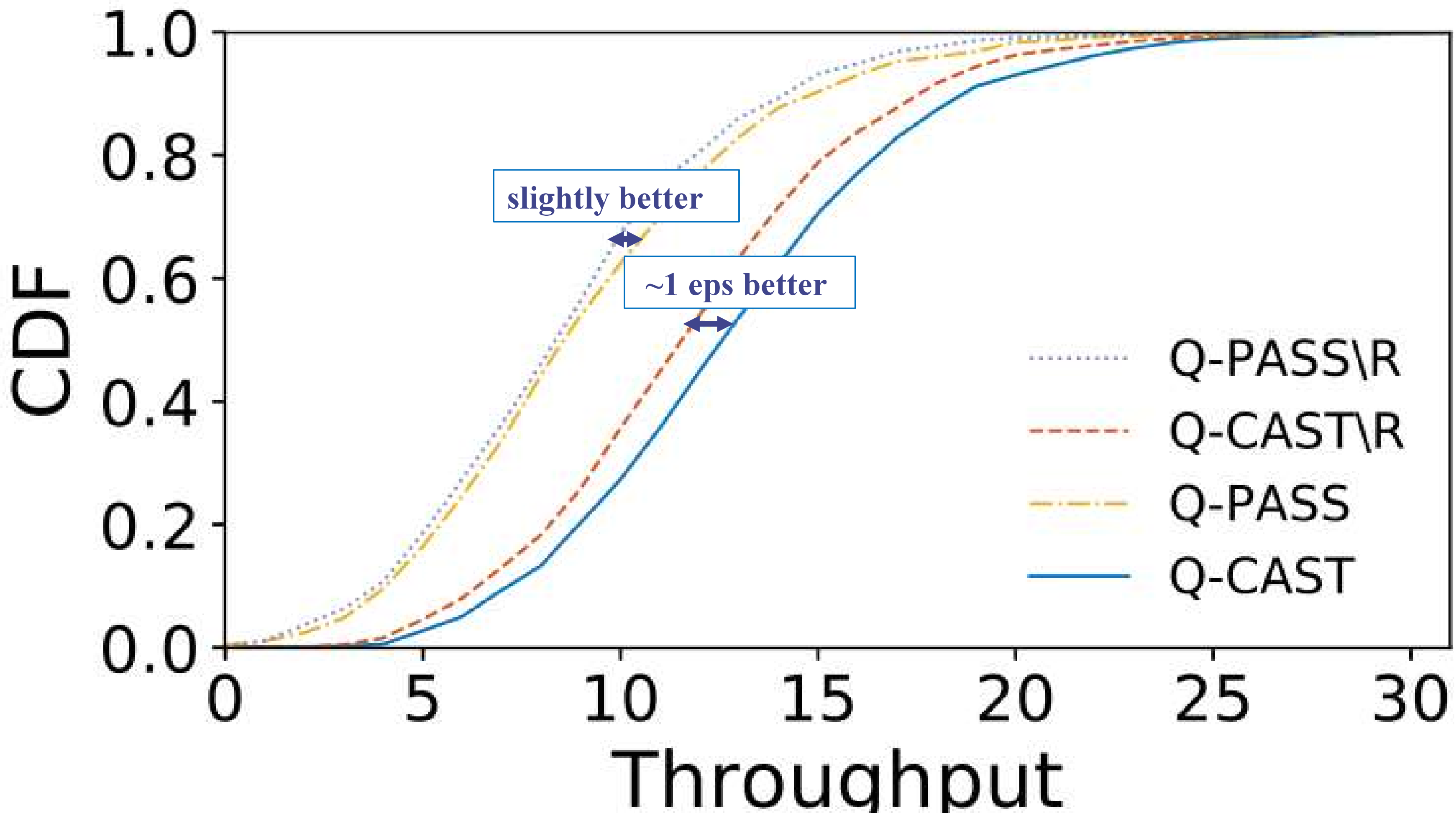


$\boldsymbol{n} = \mathbf{400}$, $E_p = 0.6$, $q = 0.9$, $k = 3$, $E_d = 6$, $m = 10$


Throughput (eps)
12
10
8
6
4
2
0
3
6
∞
Link state broadcast range k
Q-CAST
Q-PASS
Greedy
SLMP


Throughput (eps)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
50
100
200
400
800
n
Q-CAST
Q-PASS
Greedy
SLMP

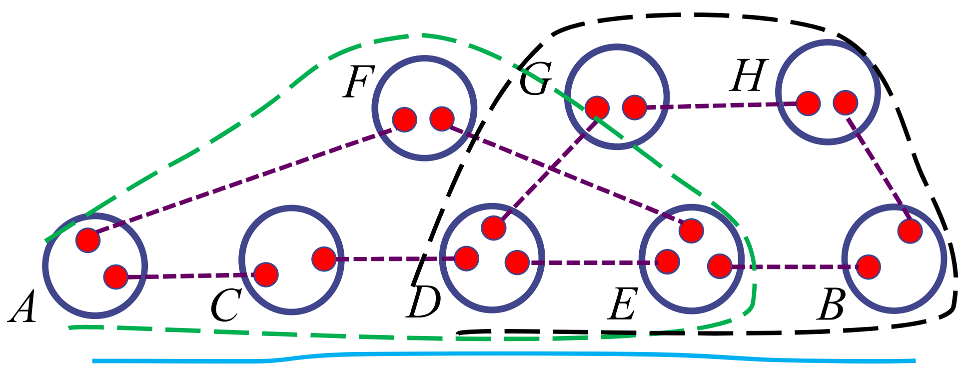

Throughput (eps)
14
12
10
8
6
4
2
0
2
4
6
8
10
S-D pairs in one time slot
Q-CAST
Q-PASS
Greedy
SLMP

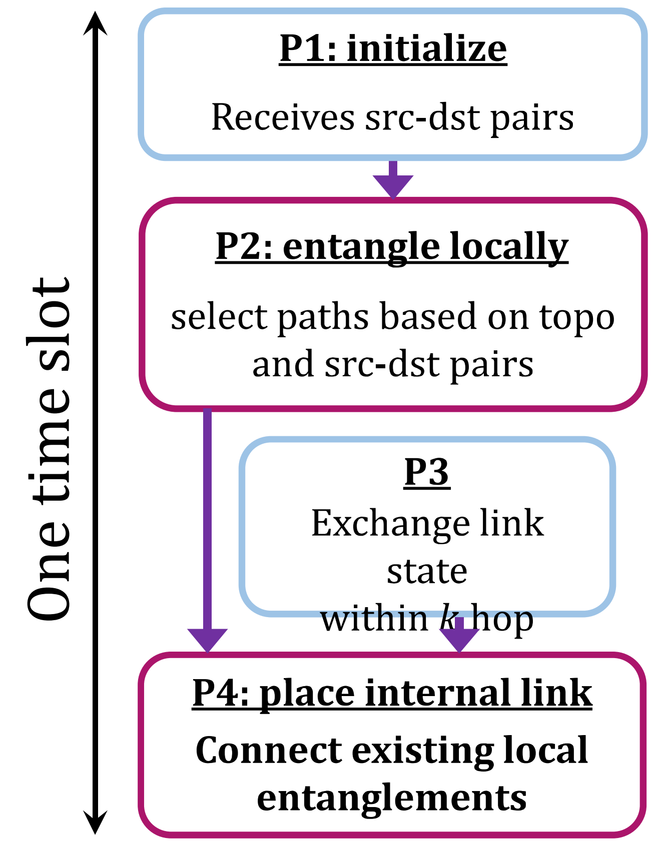

slightly better
~1 eps better
Q-PASS\R
Q-CAST\R
Q-PASS
Q-CAST
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Throughput

Conclusion

- Quantum network model
  - Peculiarity: slotted time, probabilistic link states, non-additive metric
- Algorithms
  - Q-PASS: offline routing
  - Q-CAST: online routing
- Smarter algorithms? Proved fairness? QoS?
- Opensource: https://github.com/QianLabUCSC/QuantumRouting

Thank you


A
C
D
E
B
F
G
H


One time slot
P1: initialize
Receives src-dst pairs
P2: entangle locally
select paths based on topo and src-dst pairs
P3
Exchange link state within k hop
P4: place internal link
Connect existing local entanglements