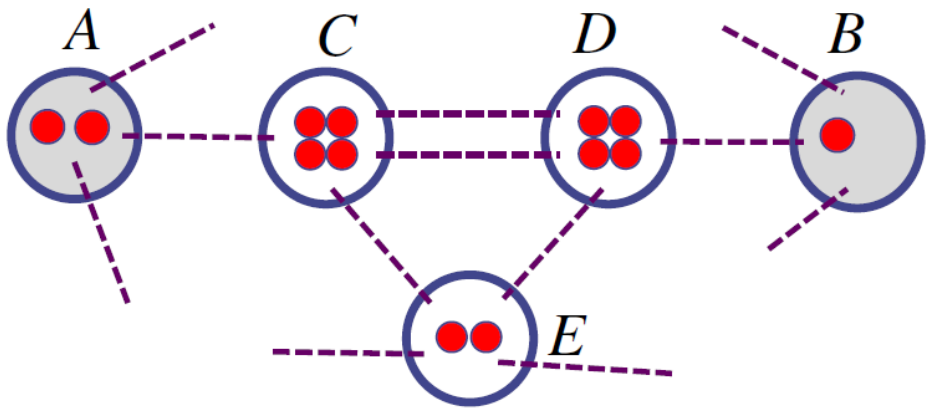
Data Structures
Programming Project #1

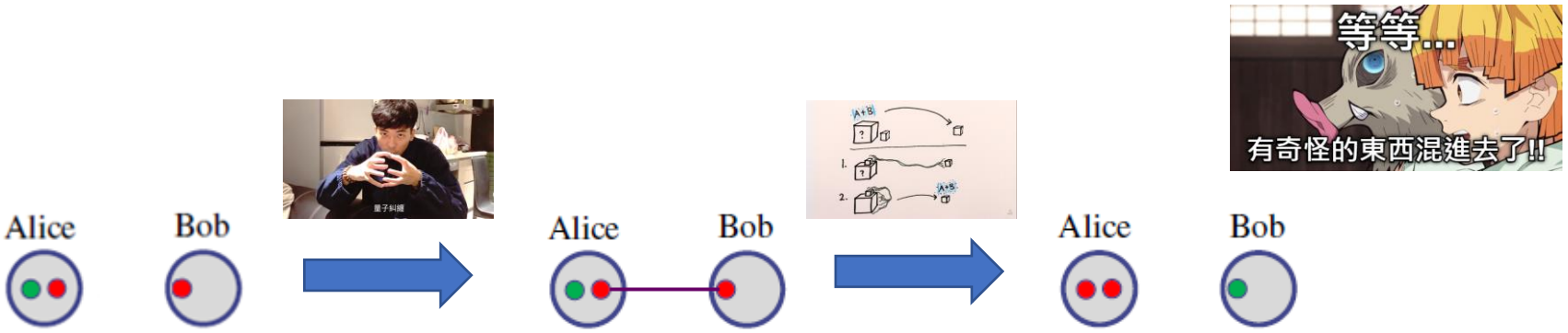
Data Transmission in Quantum Networks

- A quantum network:
- Nodes has a limited amount of quantum memory
- Nodes are interconnected with a limited number of quantum channels (e.g., optical fiber)



Data Transmission in Quantum Networks

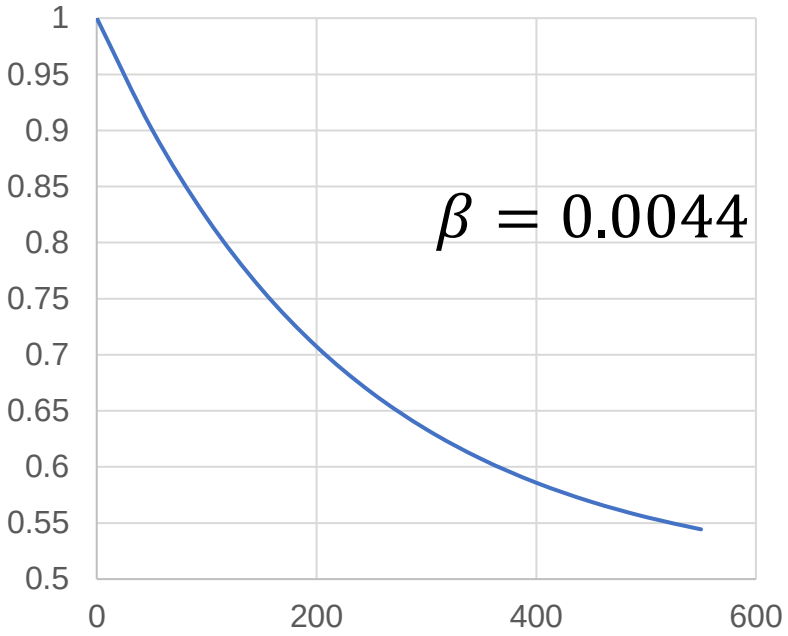
- Entangling (building an entangled link):
  Create an entangled pair between two nodes
- Precondition:
  Two nodes each with a quantum memory are interconnected with a quantum channel

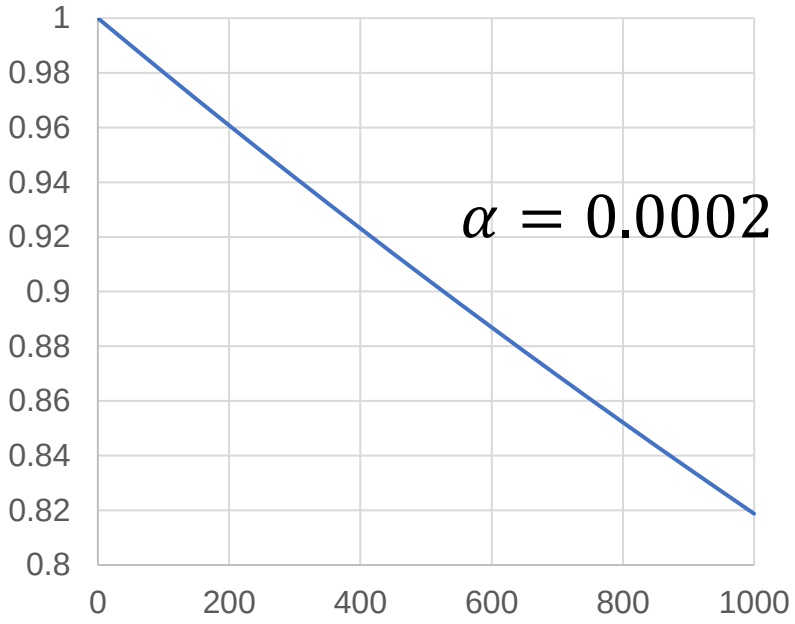


https://www.popularmechanics.com/science/a25699/how-quantum-teleportation-works/

Fidelity and Probability of Entangling

- The fidelity and success probability of building an entangled link is related to the fiber distance l



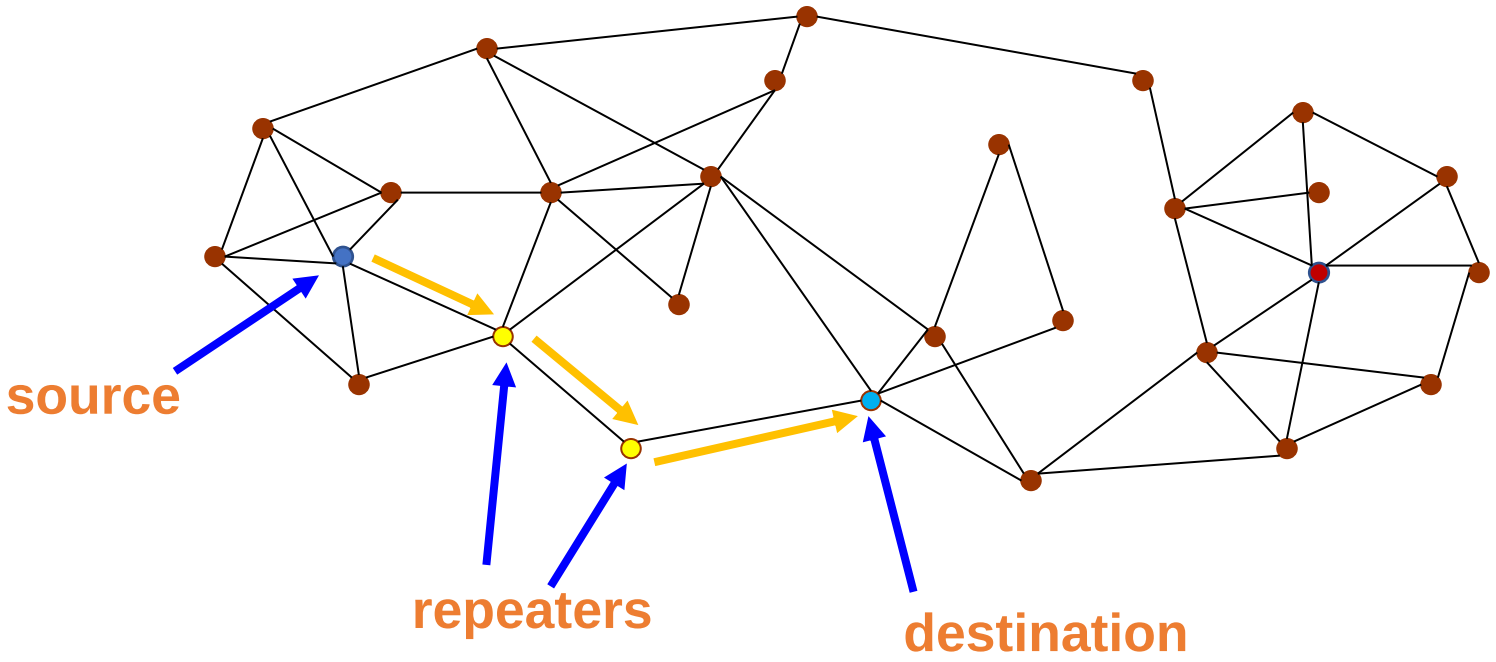


Fidelity: $\frac{1}{2} + \frac{1}{2} e^{-\beta \cdot l}$

Probability: $e^{-\alpha \cdot l}$

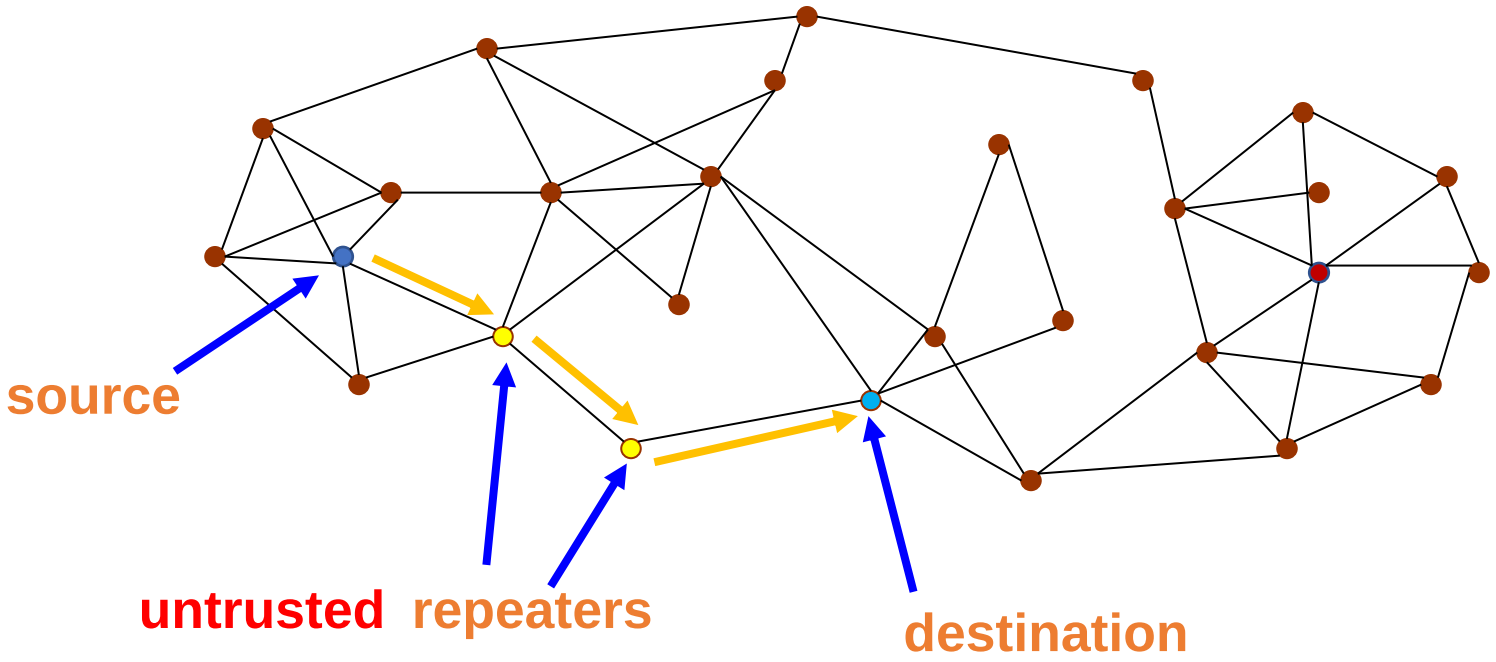
Long-Distance Data Transmission

- The two nodes may be distant from each other
- Classical networks:
  Repeaters use store and forward to transmit packets from a source to a destination



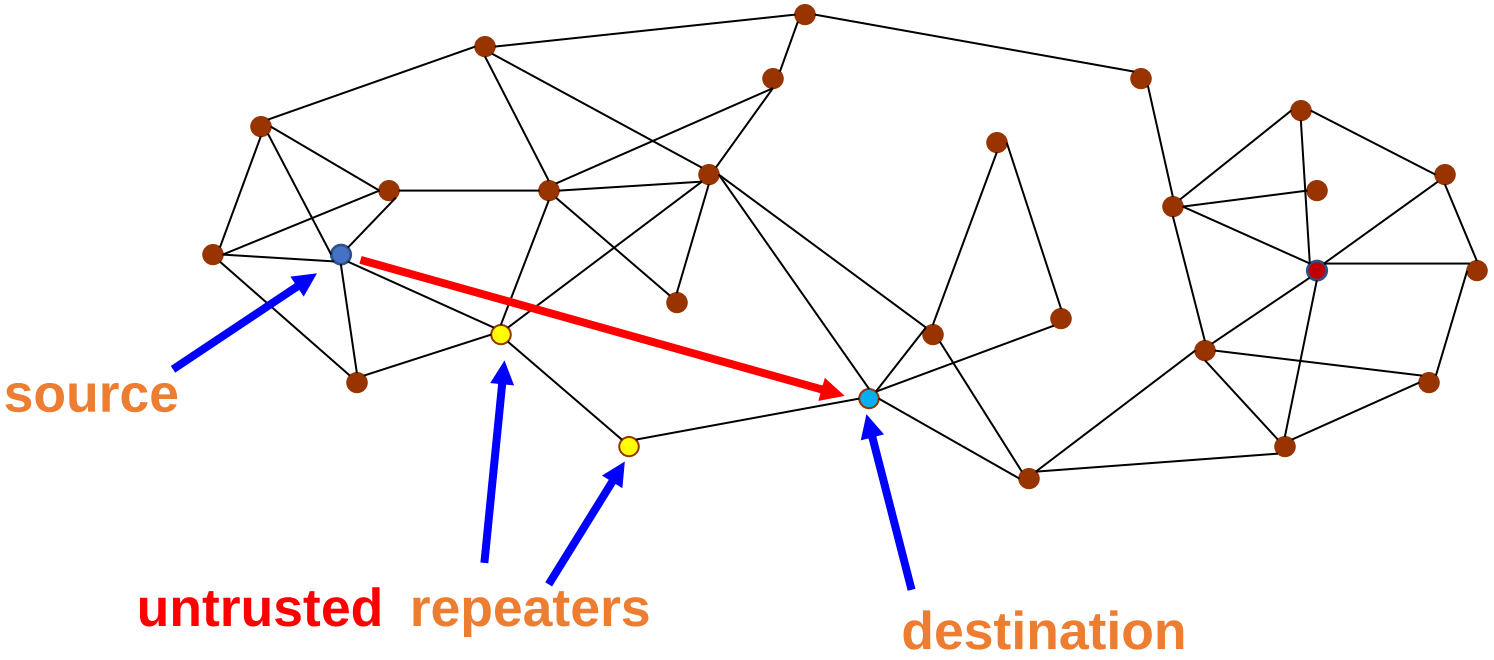
Drawback of Store-and-Forward

- However, the data qubit may visit untrusted repeaters
- It could be destroyed, peeked at, or faked



Abandon Store-and-Forward

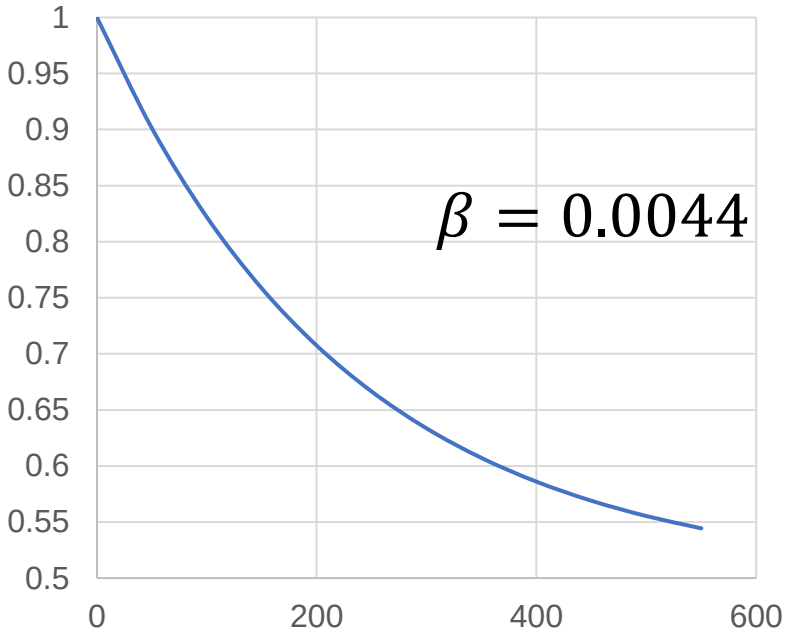
- Can we send the data qubits without letting repeaters know?
- Yes, all-optical-switching technique can bypass repeaters and construct a long entangled link



Issue of Long-Distance Bypassing

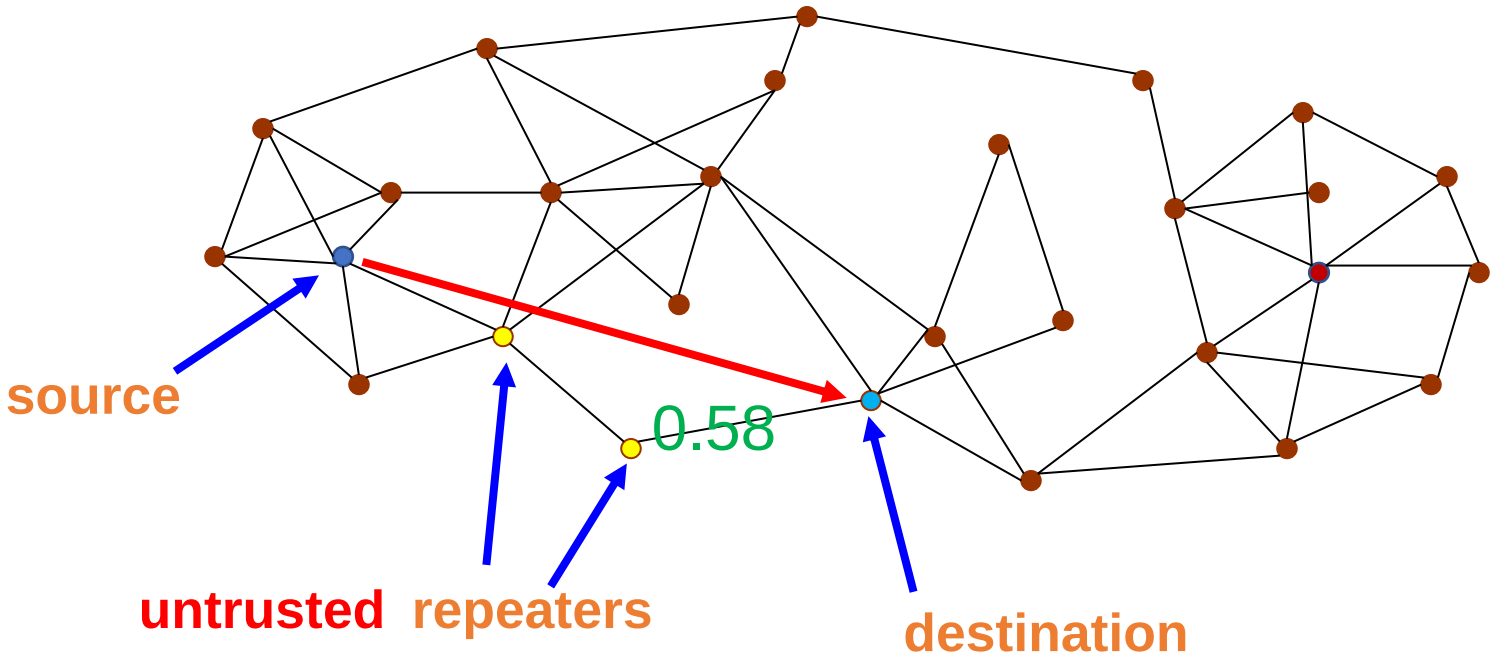
- Can we send the data qubits without letting repeaters know?
- Yes, all-optical-switching technique can bypass repeaters and construct a long entangled link
- However, the fidelity could be terrible when the fiber distance l is overlength

Fidelity: $\frac{1}{2} + \frac{1}{2} e^{-\beta \cdot l}$



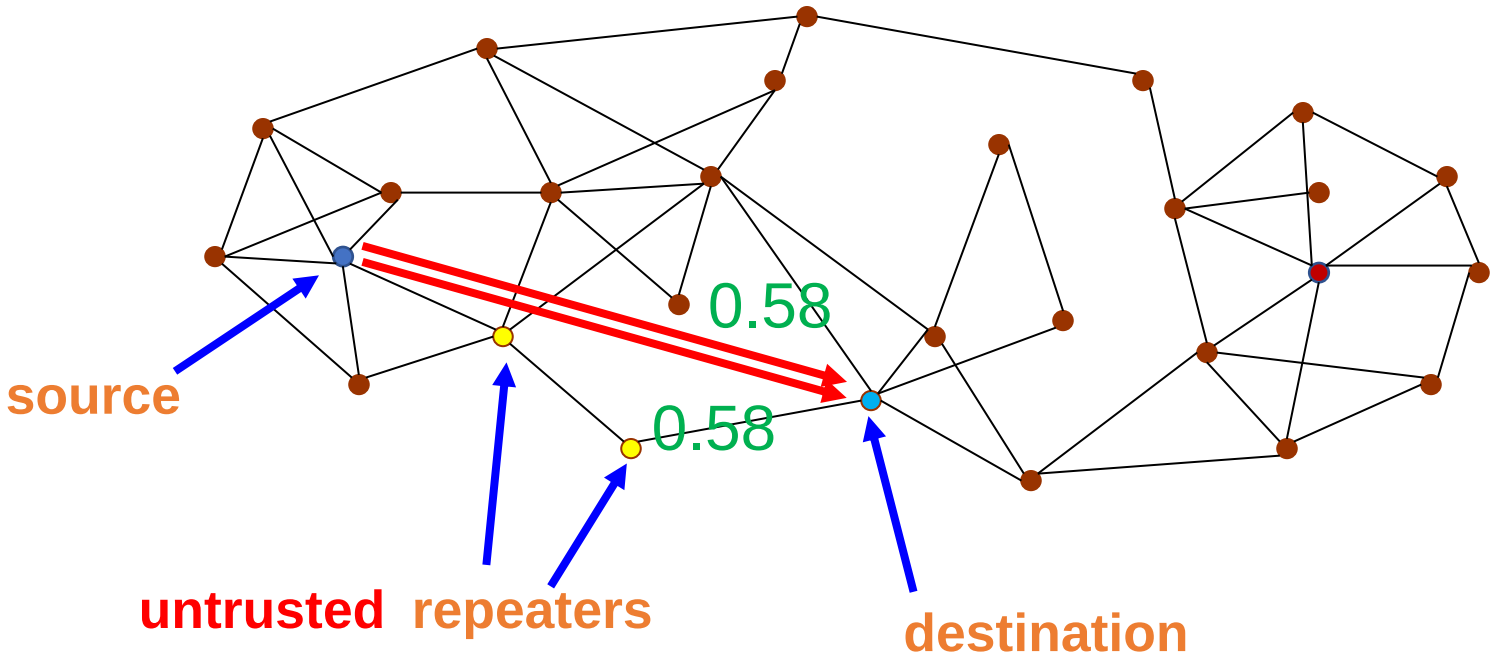
Remedy of Fidelity Loss

- Can we fix the fidelity loss due to distance?
- Yes, via Entanglement Purification
- Sacrifice a link to raise the other link's fidelity



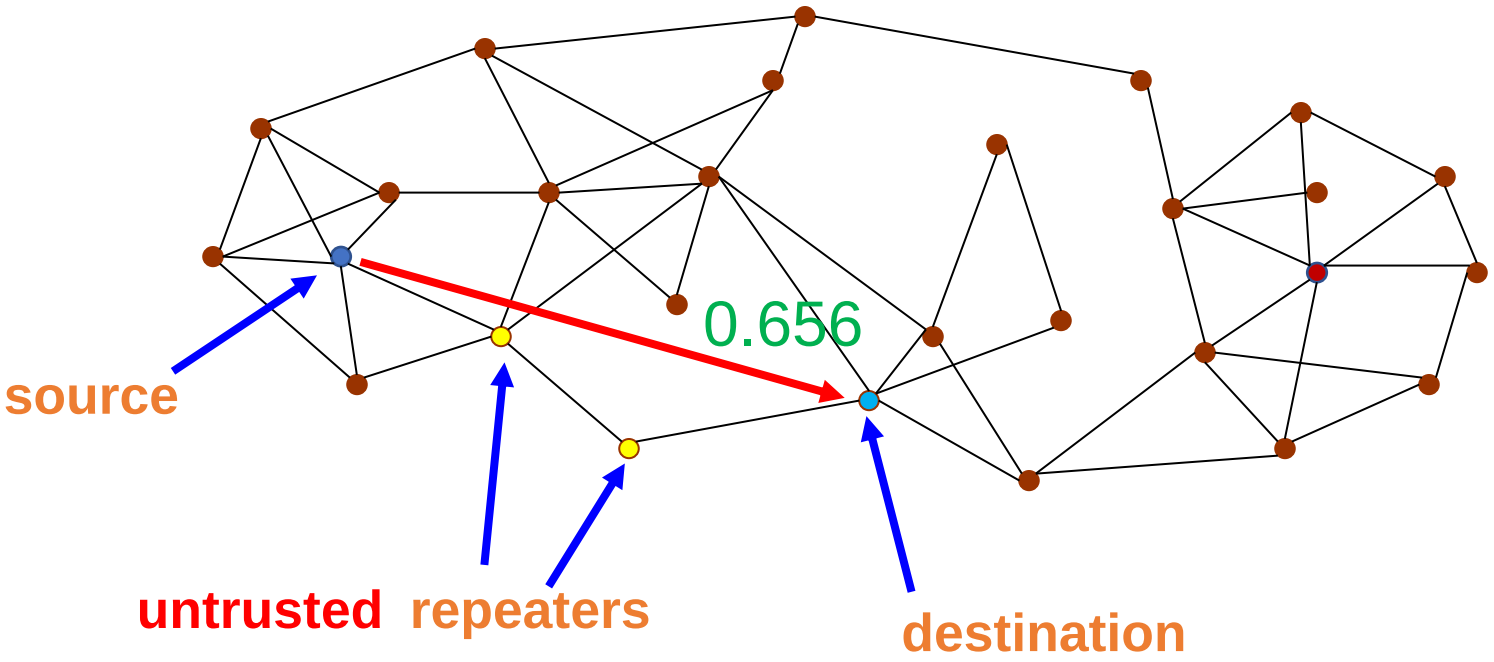
Remedy of Fidelity Loss

- Can we fix the fidelity loss due to distance?
- Yes, via Entanglement Purification
- Sacrifice a link to raise the other link's fidelity



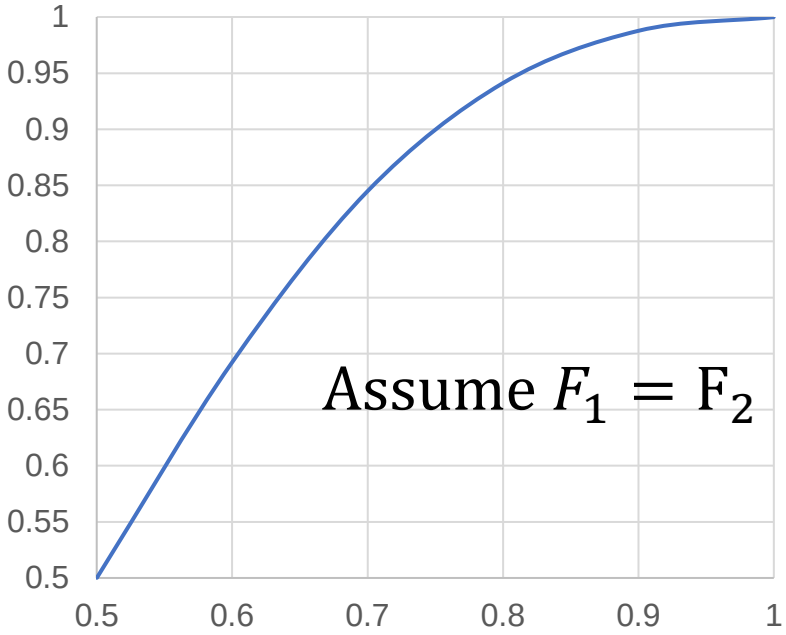
Remedy of Fidelity Loss

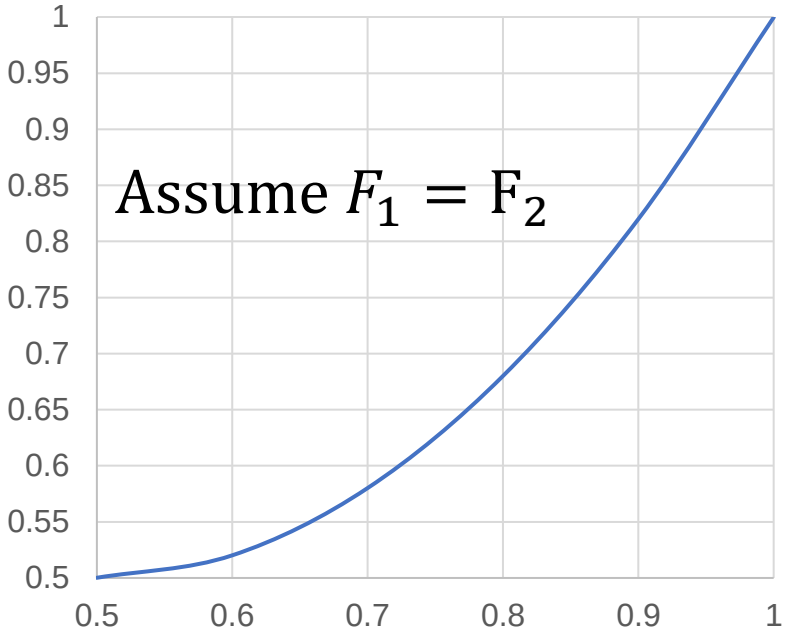
- Can we fix the fidelity loss due to distance?
- Yes, via Entanglement Purification
- Sacrifice a link to raise the other link's fidelity



Fidelity and Probability of Purification

- Purification can raise the fidelity with a probability
- For simplicity, here we adopt **binary state**





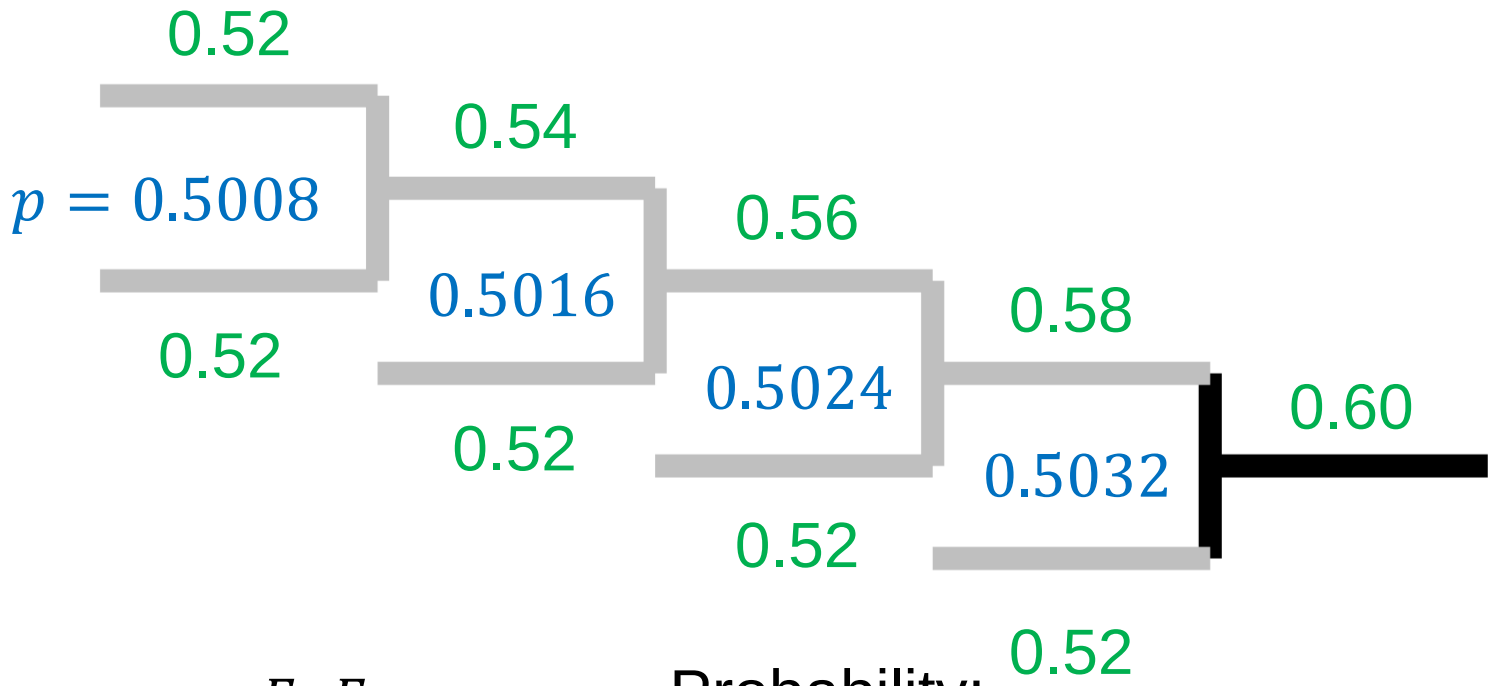
Fidelity: $\frac{F_1 \cdot F_2}{F_1 \cdot F_2 + (1-F_1)\cdot(1-F_2)}$

Probability:

$$F_1 \cdot F_2 + (1-F_1)\cdot(1-F_2)$$

Fidelity and Probability of Purification

- Purification can raise the fidelity with a probability
- For simplicity, here we adopt **pumping**



Fidelity: $\frac{F_1 \cdot F_2}{F_1 \cdot F_2 + (1-F_1) \cdot (1-F_2)}$

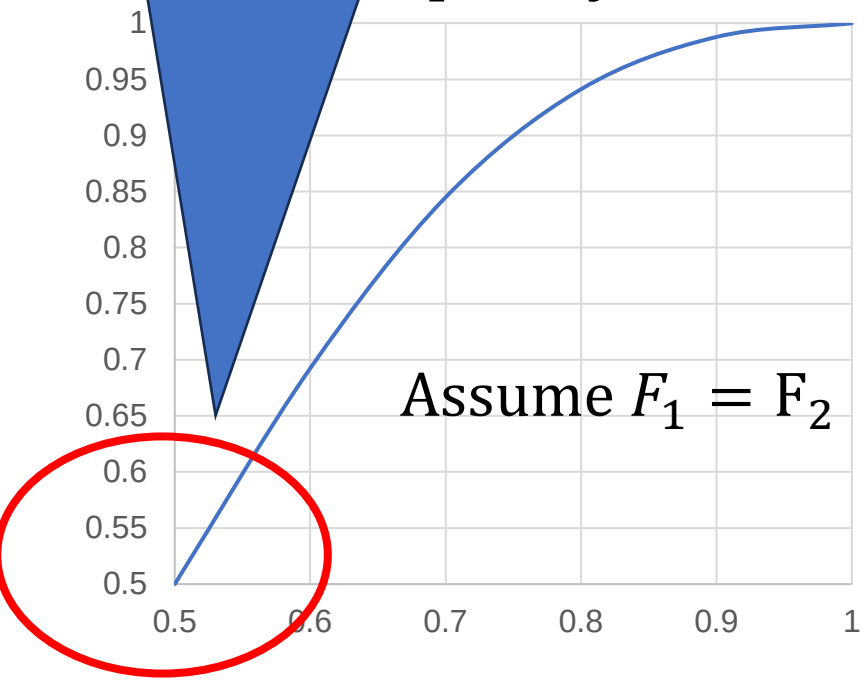
Probability: $F_1 \cdot F_2 + (1 - F_1) \cdot (1 - F_2)$

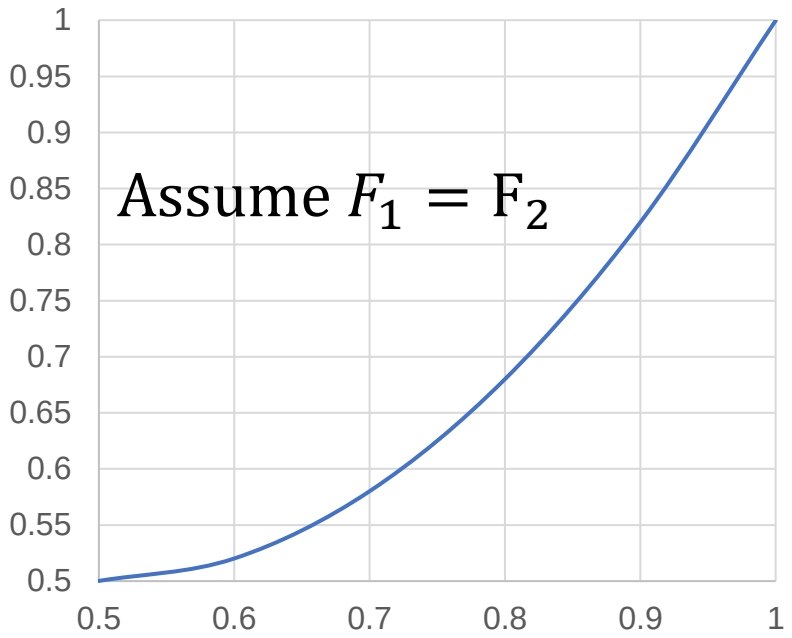
Fidelity and Probability of Purification

It is difficult to fix the fidelity when the fidelity is low

the fidelity with a probability

plicity, here we adopt **binary state**





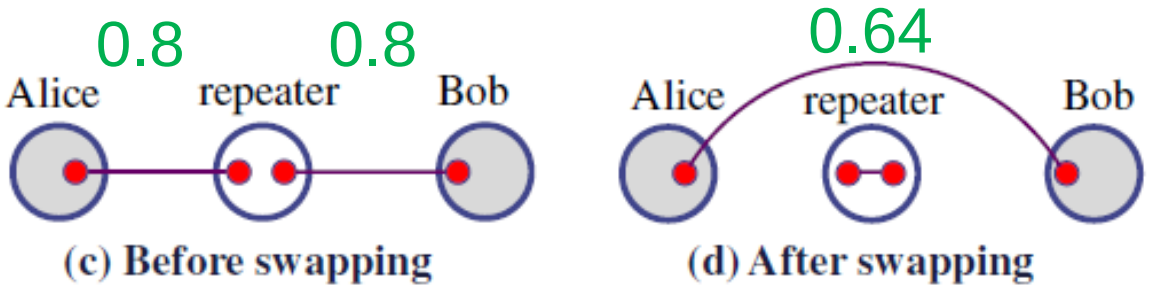
Fidelity: $\frac{F_1 \cdot F_2}{F_1 \cdot F_2 + (1-F_1)\cdot(1-F_2)}$

Probability:

$$F_1 \cdot F_2 + (1-F_1)\cdot(1-F_2)$$

Path Decomposition

- Can we construct links and then merge them?
- Yes, via Entanglement Swapping
- Conduct the swapping operation on the intermediate repeater to merge two links
- Alice has a data qubit for Bob

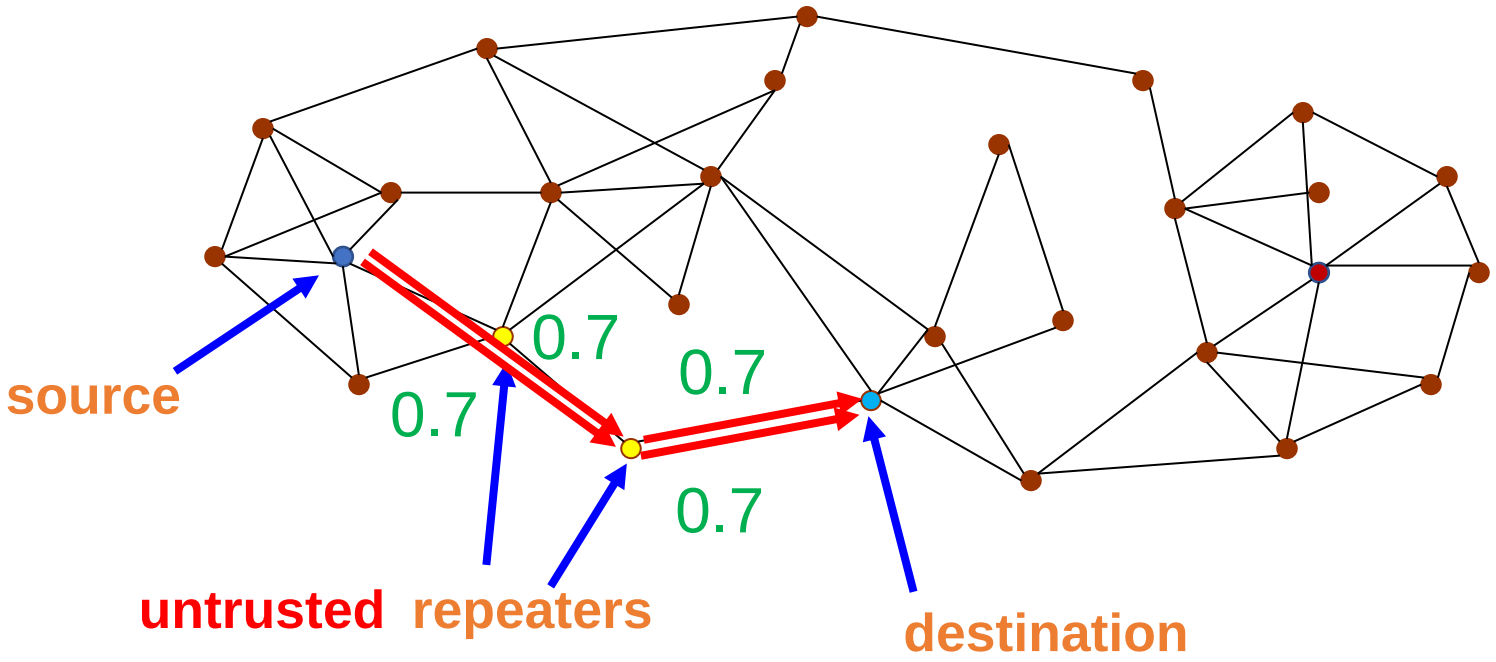


(c) Before swapping (d) After swapping

Fidelity: $F_1 \cdot F_2$ Probability: e.g., $q = 0.7$

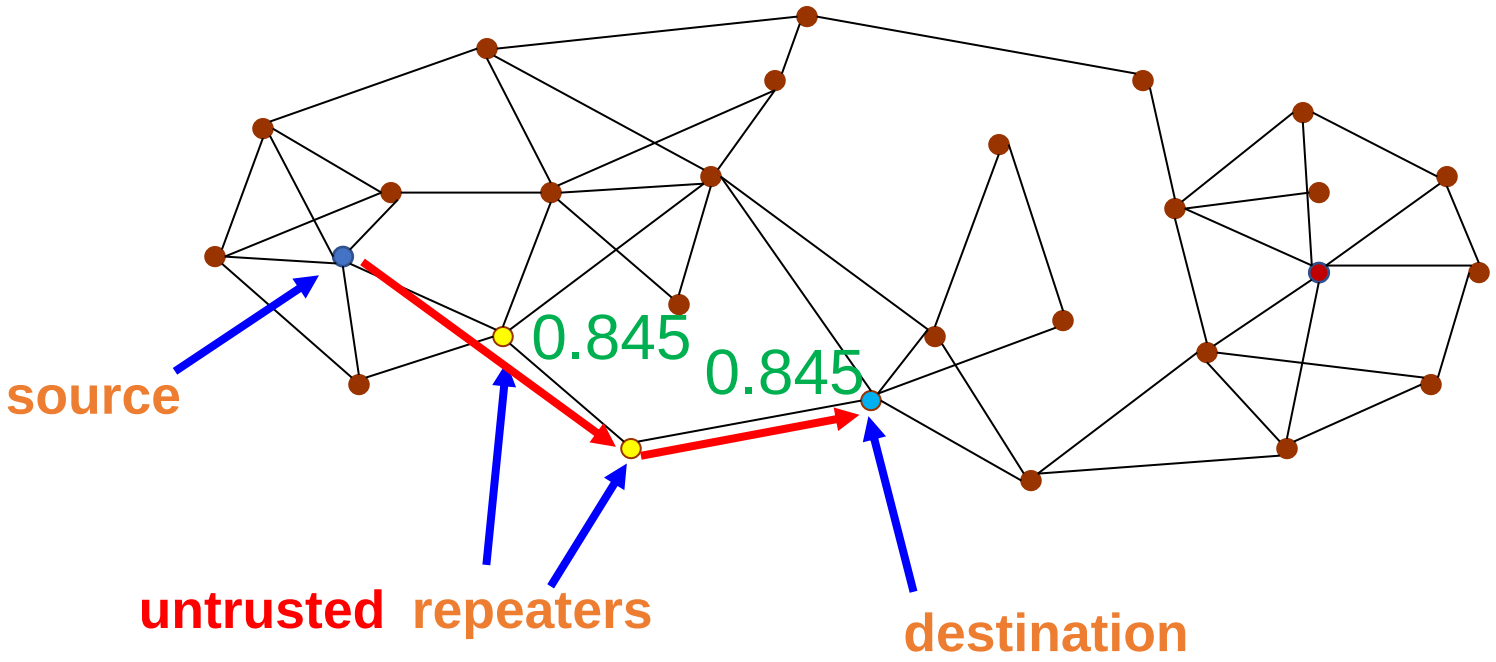
Path Decomposition

- Can we construct links and then merge them?
- Yes, via Entanglement Swapping
- Conduct the swapping operation on the intermediate repeater to merge two links



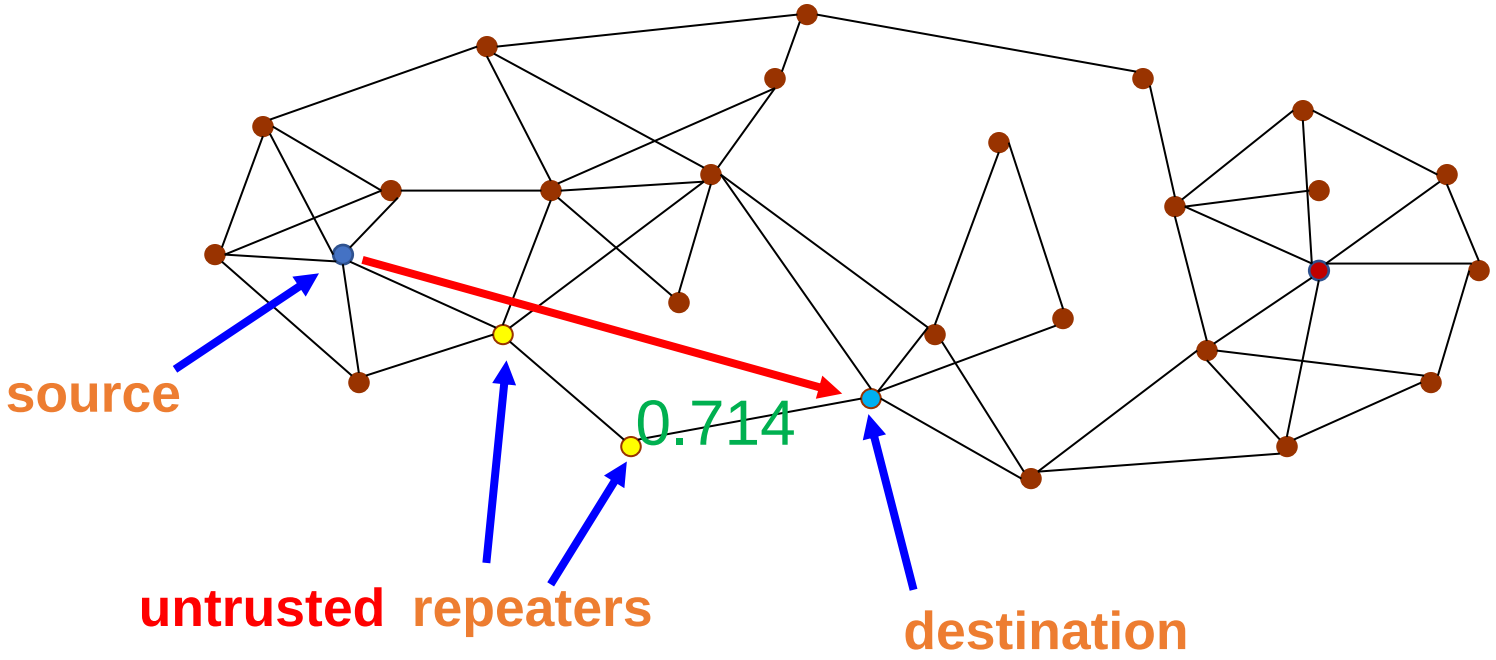
Path Decomposition

- Can we construct links and then merge them?
- Yes, via Entanglement Swapping
- Conduct the swapping operation on the intermediate repeater to merge two links



Path Decomposition

- Can we construct links and then merge them?
- Yes, via Entanglement Swapping
- Conduct the swapping operation on the intermediate repeater to merge two links



Path Decomposition

- Can we construct links and then merge them?
- Yes, via Entanglement Swapping
- Conduct the swapping operation on the intermediate repeater to merge two links

- For simplicity, we assume that purification operations are conducted before swapping operations

Summary of Operations

	Fidelity	Probability
Entangling	Distance ↑, it ↓	Distance ↑, it ↓
Bypass	Distance ↑, it ↓	Distance ↑, it ↓
Purification	↑	↓
Swapping	↓	↓

System Model & Problem Formulation

- Given:
- A path in a quantum network
- Each node has a limited mount of memory
- (Assume fiber channels are always sufficient)
- A fidelity threshold

- Goal: maximize the success probability

- Constraints:
- The fidelity must be no less than the threshold

Programming Project #1: Swapping and Purification Optimization

- Input:
  - A node-weighted path $P = (V, E)$ with parameters α, β, q
  - A fidelity threshold T (which is calculated by you)
- Procedure:
  - Calculate generated links using entangling / bypassing while meeting the memory capacity constraint
  - Estimate fidelity after purification and swapping, meeting the fidelity threshold
  - Maximize the success probability
- Output:
  - The number of generated entangled links between nodes

Programming Project #1: Swapping and Purification Optimization

- Input:
  - A node-weighted path $P = (V, E)$ with parameters α, β, q
  - A fidelity threshold T (which is calculated by you)
- Procedure:
  - Calculate generated links using entangling / bypassing while meeting the memory capacity constraint
  - Estimate fidelity after purification and swapping, meeting the fidelity threshold
  - Maximize the success probability
- Output:
  - The number of generated entangled links between nodes

Programming Project #1: Swapping and Purification Optimization

- Input:
  - A node-weighted path $P = (V, E)$ with parameters α, β, q
  - A fidelity threshold T (which is calculated by you)
- Procedure:
  - Calculate generated links using entangling / bypassing while meeting the memory capacity constraint
  - Estimate fidelity after purification and swapping, meeting the fidelity threshold
  - Maximize the success probability
- Output:
  - The number of generated entangled links between nodes
- The grade is proportional to success probability

The Competition

- The grade is proportional to success probability
- Basic: 60 (deadline)
  - A baseline solution for threshold (see the following pages)
  - (Meet the coding style requirements)
- Performance ranking (decided after the deadline)
  - [0%, 30%) (bottom): +0
  - [30%, 50%): + 5
  - [50%, 75%): + 10
  - [75%, 85%): + 15
  - [85%, 90%): + 20
  - [90%, 95%): + 25
  - [95%, 100%] (top): + 30
- Homework assistant (superb deadline)
  - +10

The Competition

- The grade
- Basic: 60 (
  - A baselin
  - (Meet the
- Performar
  - [0%, 30%
  - [30%, 50
  - [50%, 75
  - [75%, 85
  - [85%, 90
  - [90%, 95
  - [95%, 10
- Homework assistant (superb deadline)
  - +10

The Competition

- The grade
- Basic: 60 (
  - A baselin
  - (Meet the
- Performar
  - [0%, 30%
  - [30%, 50
  - [50%, 75
  - [75%, 85
  - [85%, 90
  - [90%, 95
  - [95%, 10
- Homework assistant (superb deadline)
  - +10

相信你們在做完作業以後

Baseline Algorithm for Fidelity Threshold

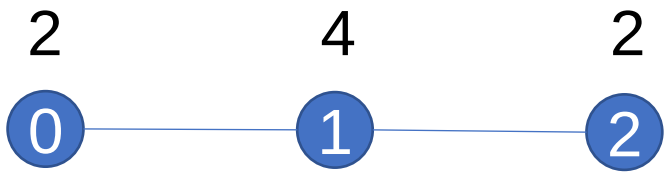
- Initially, construct an entangled link between every pair of adjacent nodes
- After that, every hop on path has an entangled link

- Modifying the solution by a greedy strategy:
- Iteratively examine every hop on the path and choose the hop that can increase the most fidelity by adding one more entangled link for purification
- Repeat it until no more entangled link can be constructed for purification (i.e., there is no more available memory)

- Conduct purification to derive the link fidelity for each hop
- Merge all links by swapping and acquire the resulting fidelity

- The approach does not leverage bypassing

Input Sample: use scanf



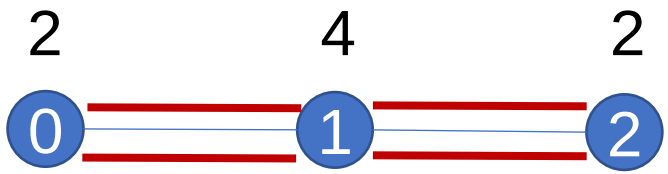
Format:

#Nodes alpha beta SwapProb
NodeID #QuantumMemories
…
LinkID Distance
…

Ex:

3 0.0002 0.0044 0.7
0 2
1 4
2 2
0 200
1 200

Output Sample: use printf



Format:

nodeID1 nodeID2 #entangledLinks

...

Ex:

0 1 2

1 2 2

Note

- Superb deadline: 10/8 Tue
- Deadline: 10/15 Tue
- Pass the test of our online judge platform
- Submit your code to E-course2
- Demonstrate your code remotely or in person with TA
- **C Source code (i.e., only .c)**
- Show a good programming style