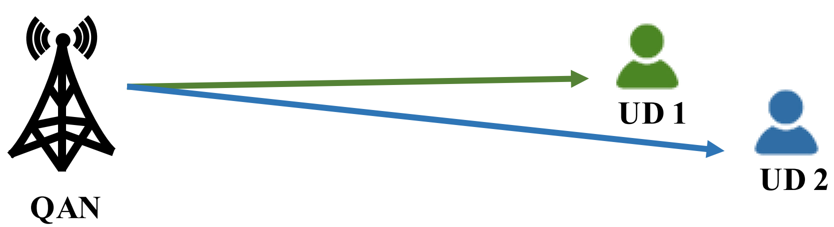
# Data Structures
# Programming Project #1

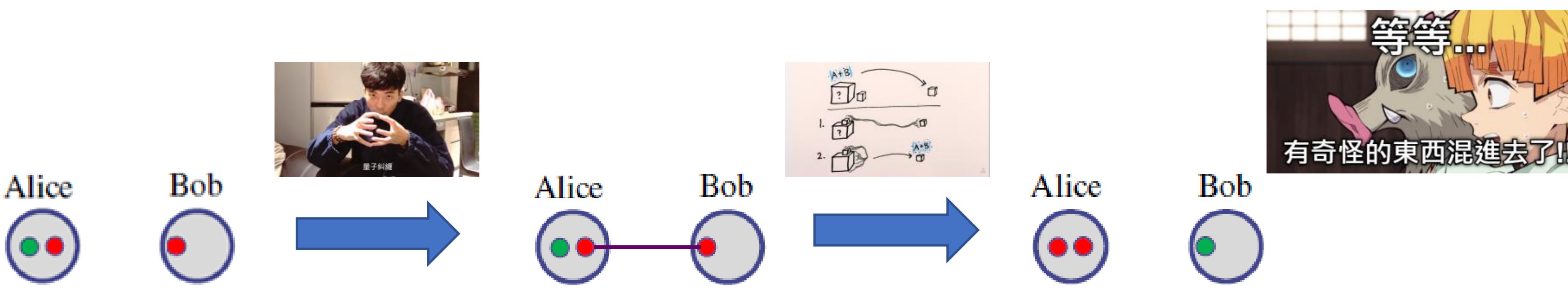
# Entanglement Distribution in FSO QNs

- A quantum access node (QAN) with user devices (UDs) in free-space optical quantum networks (QNs)
- Nodes has quantum memory to store qubits
- Nodes are interconnected with light-of-sight (LoS) free-space optical (FSO) quantum channels
- Entangled pairs are distributed from QAN to UDs
- Applications: teleportation, symmetric encryption



# Distribution and Teleportation in FSO QNs

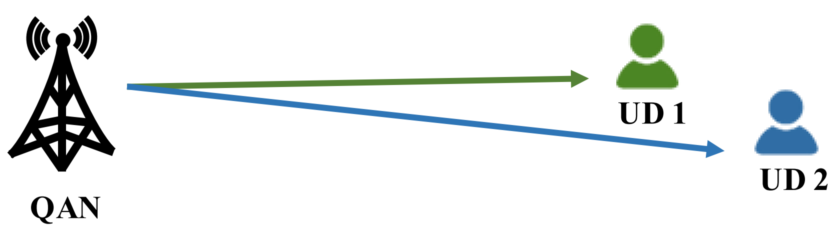
- Entanglement distribution:
  Create an entangled pair between two nodes
- Precondition:
  Two nodes each with a unit of quantum memory are interconnected with a quantum channel
- Teleportation starts as an entangled pair is ready




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

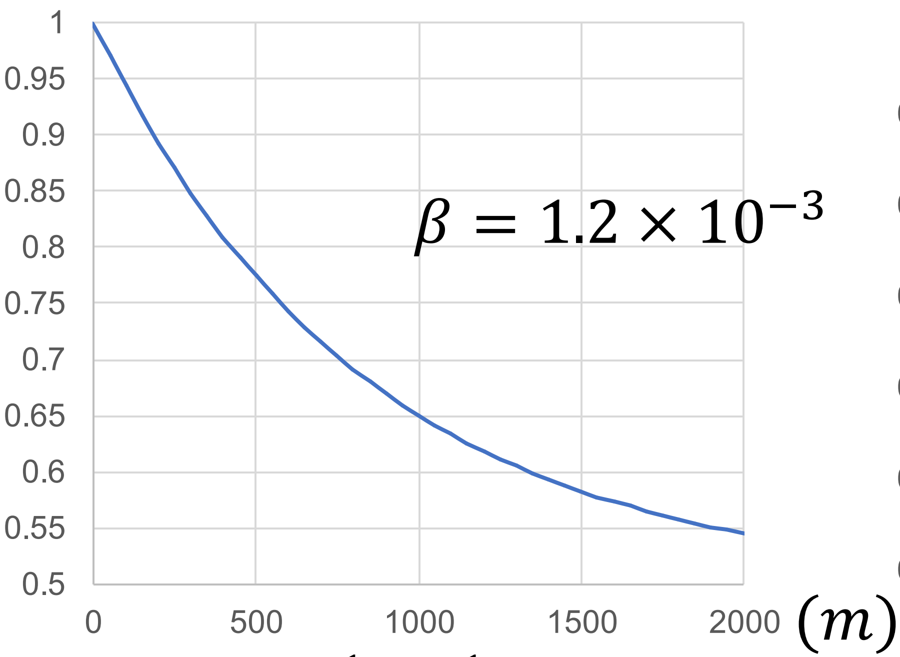
# UD's Request and Requirements

- Each UD has an all-or-nothing request with a profit, a fidelity threshold, and a demand expected key rate
- Either reject or accept it
- Once accepted, the resulting fidelity should meet the fidelity threshold and the received key rate should be no less than the demand expected key rate

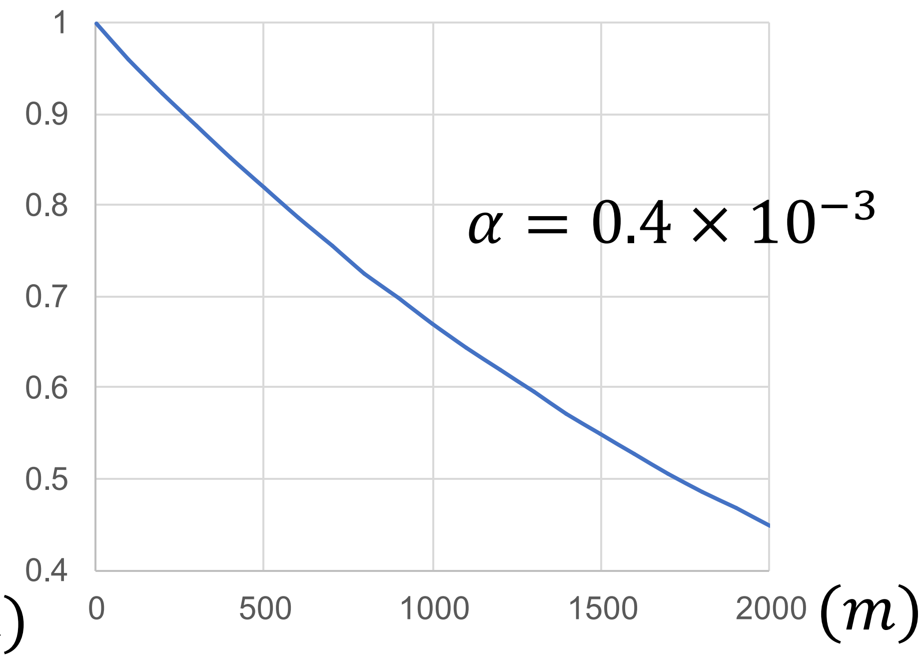


# Fidelity and Probability of Entangling

- The success probability of establishing an entangled pair and the resulting fidelity are related to the FSO distance $l$ $(m)$ (here we adopt **binary state**)



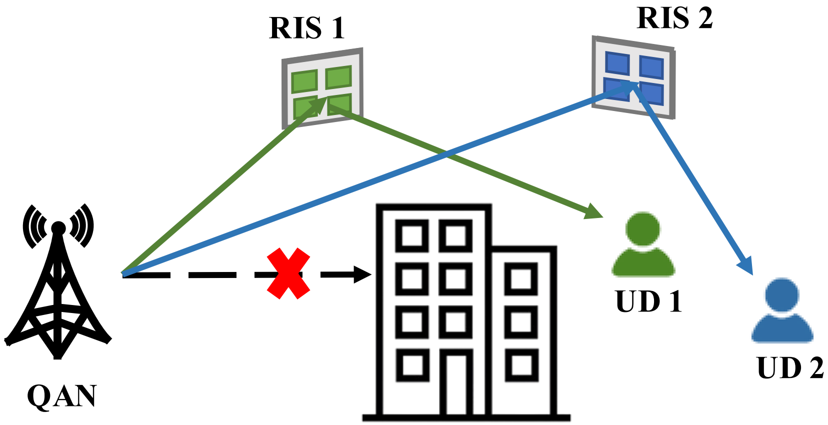

Fidelity: $\frac{1}{2} + \frac{1}{2} e^{-\beta \cdot l}$




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

# NLoS Challenge and RIS

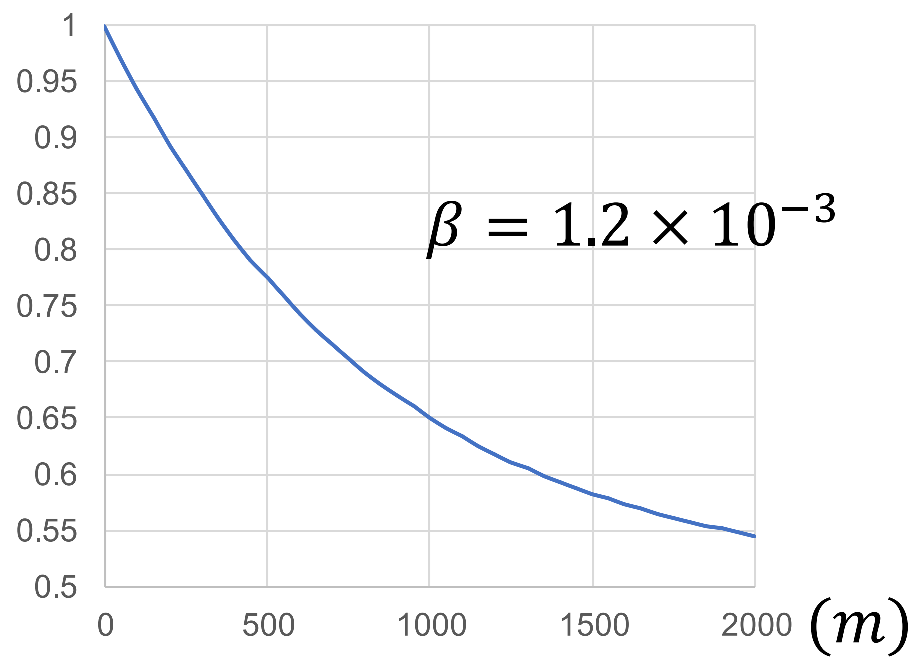
- Two nodes may be non-line-of-sight due to blockage by buildings or other obstacles
- A reconfigurable intelligent surfaces (RIS) reflects the signal via phase shift so that a blocked UD can still receive it (i.e., virtual LoS)
- Each RIS can be assigned to at most one UD



# Penalty of RISs – Longer Optical Path

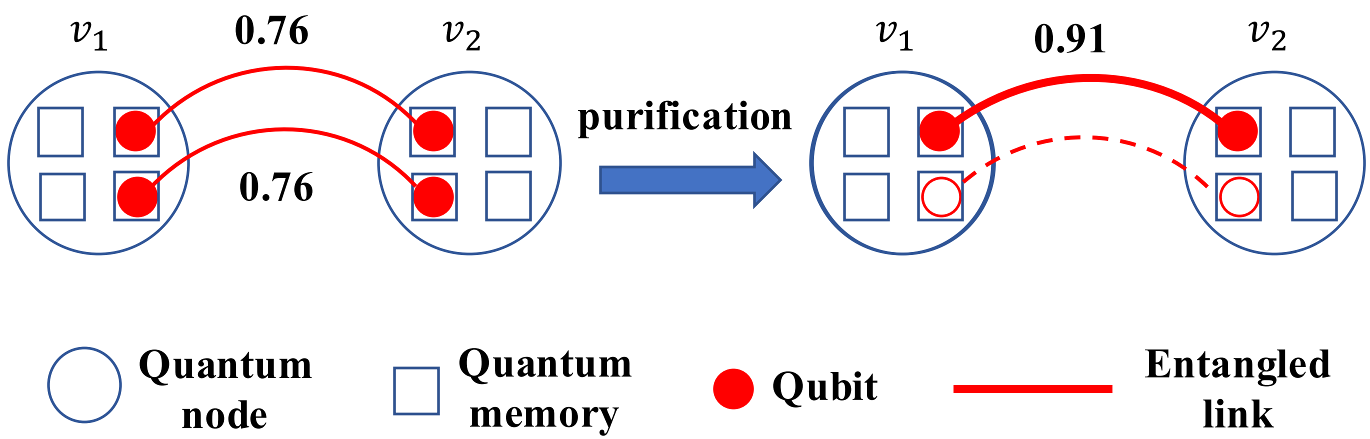
- An inherent penalty of RIS is the longer optical path, leading to degraded fidelity
- The fidelity could be terrible (i.e., cannot meet the fidelity threshold) when the FSO distance $l$ is overlength

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



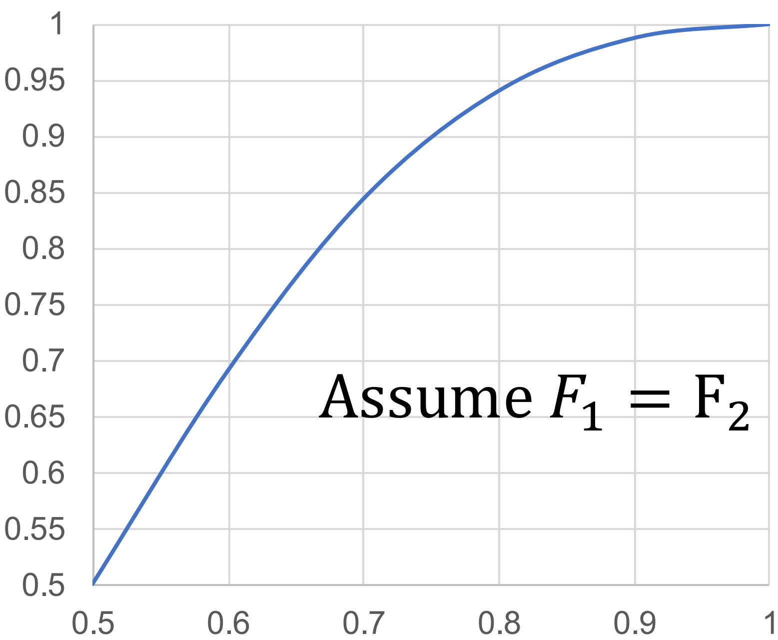
# Remedy of Fidelity Loss – Purification

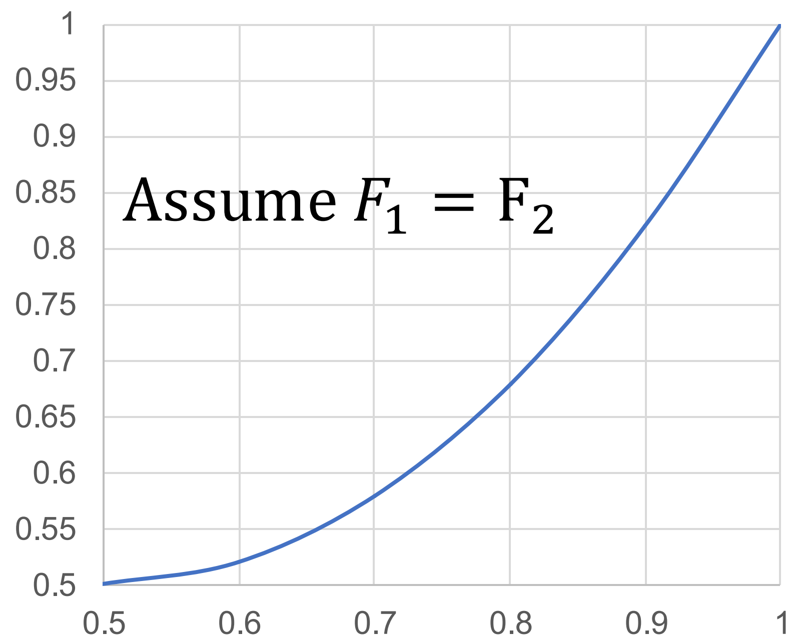
- An inherent penalty of RIS is the longer optical path, leading to degraded fidelity
- Entanglement purification improves fidelity at the cost of consuming an additional entangled pair



# Fidelity and Probability of Purification

- Purification can raise the fidelity with a probability
- Note that we adopt **binary state**



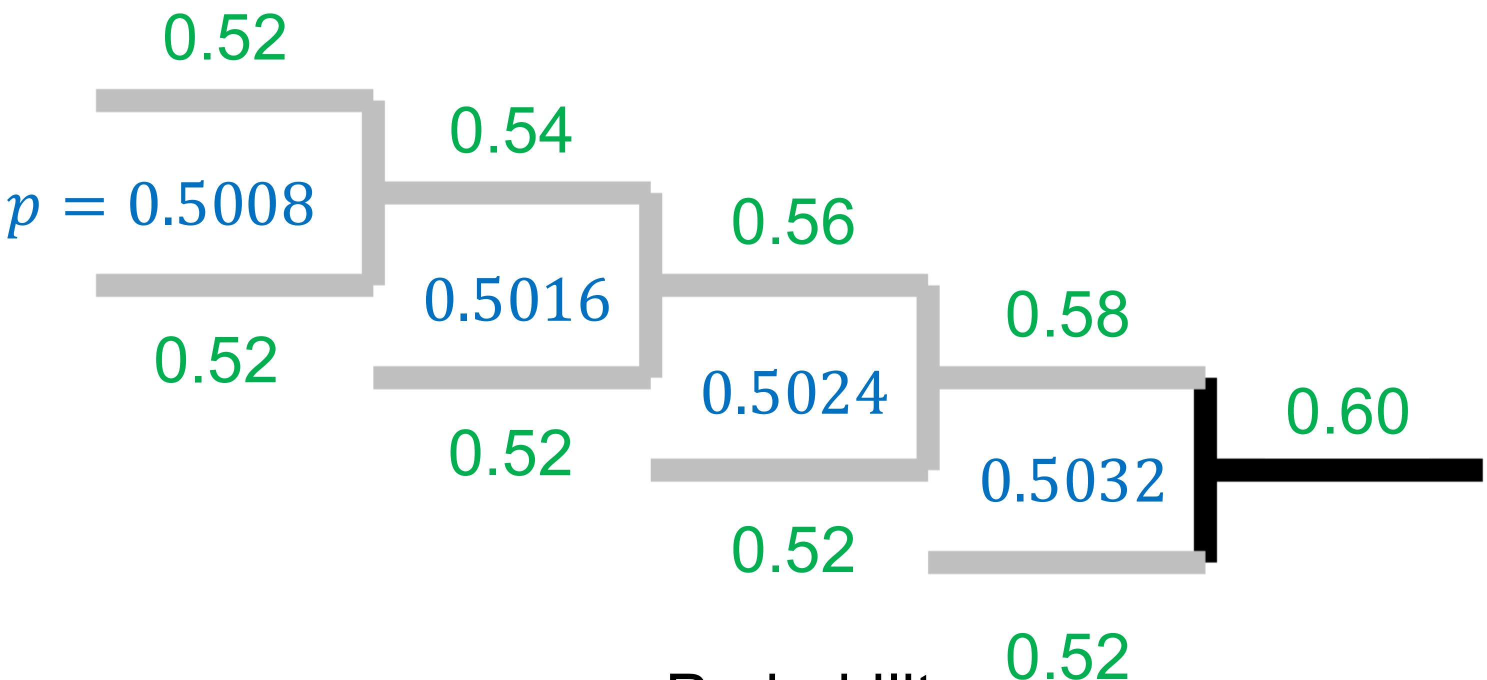

Fidelity: $\dfrac{F_1 \cdot F_2}{F_1 \cdot F_2 + (1 - F_1) \cdot (1 - F_2)}$

Probability:

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

# Fidelity and Probability of Purification

- We can conduct multiple rounds of purification
- 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:

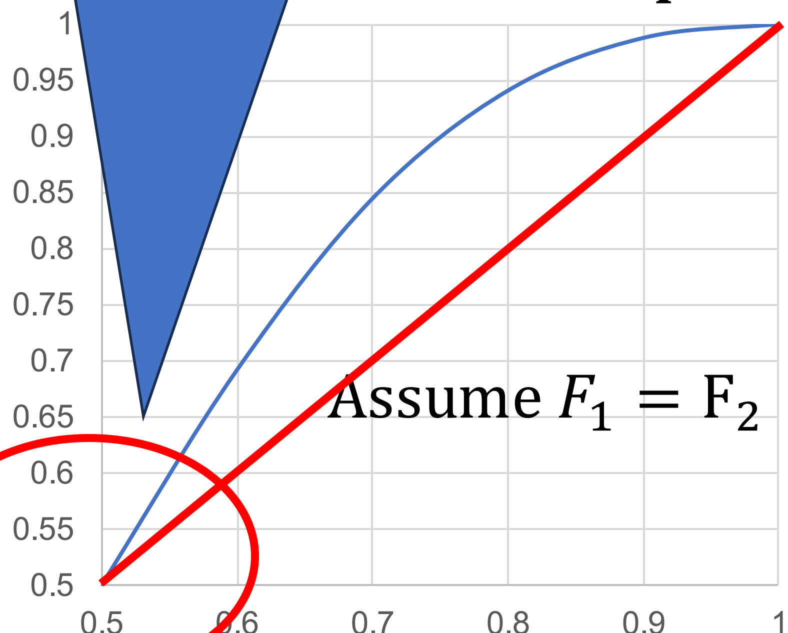
$F_1 \cdot F_2 + (1 - F_1) \cdot (1 - F_2)$

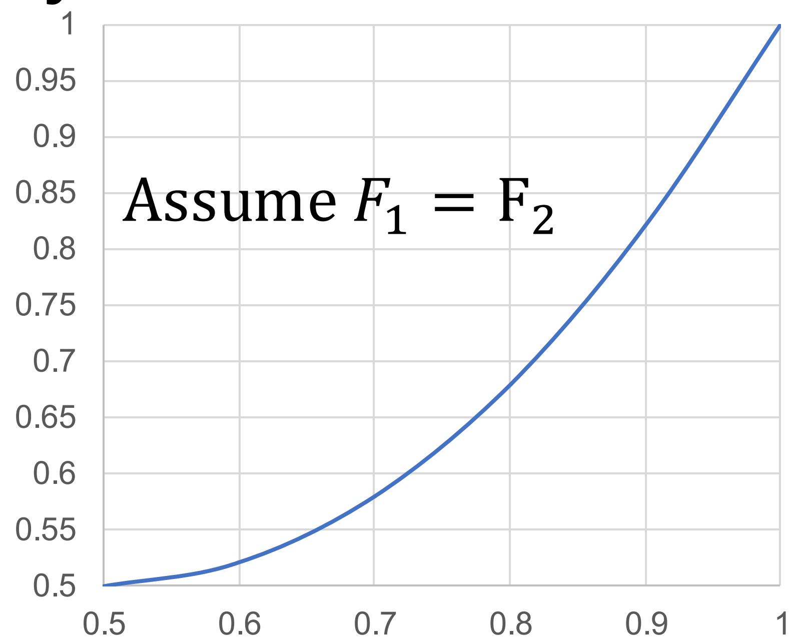
# Fidelity and Probability of Purification

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

the fidelity with a probability

that we adopt **binary state**




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

Probability:

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

# Summary of Operations

| | Fidelity | Probability |
|---|---|---|
| Entangling | Distance $\uparrow$<br>Fidelity $\downarrow$ | Distance $\uparrow$<br>Probability $\downarrow$ |
| Purification | $\uparrow$ | $\downarrow$ |

# Summary of Formulas (1/4)

- UD: $u$
- Indirect path via RIS or direct path to the QAN: $r$

- The success probability of entangling from the QAN to UD $u$ via path $r$:

$$P_e(u, r) = e^{-\alpha \cdot l(u,r)}$$

- The resulting fidelity:

$$F_e(u, r) = \frac{1}{2} + \frac{1}{2} e^{-\beta \cdot l(u,r)}$$

# Summary of Formulas (2/4)

- Fidelity of two entangled pairs: $F_1$ and $F_2$
- The success probability of purification on two entangled pairs with fidelity $F_1$ and $F_2$:
$$P_p(F_1, F_2) = F_1 \times F_2 + (1 - F_1) \times (1 - F_2)$$
- The resulting fidelity:
$$F_p(F_1, F_2) = \frac{F_1 \cdot F_2}{F_1 \cdot F_2 + (1 - F_1) \cdot (1 - F_2)}$$

# Summary of Formulas (3/4)

- # rounds of pumping purification: $n \in \mathbb{Z}_0^+$
- The success probability of executing $n$ rounds of pumping purification between the QAN and UD $u$ via path $r$:

$$P(u,r,n) = \begin{cases} P_p\big(F(u,r,n-1), F_e(u,r)\big) \cdot P(u,r,n-1), & \text{if } n \geq 1 \\ 1, & \text{if } n = 0 \end{cases}$$

- The resulting fidelity:

$$F(u,r,n) = \begin{cases} F_p\big(F(u,r,n-1), F_e(u,r)\big), & \text{if } n \geq 1 \\ F_e(u,r), & \text{if } n = 0 \end{cases}$$

# Summary of Formulas (4/4)

- The fidelity threshold of UD $u$: $F_{th}(u) \geq 0$
- The expected number of key rate of UD $u$: $d(u) \geq 0$

- The minimum number of required entangled pairs to generate one purified entangled pair that meets the fidelity threshold $F_{th}(u)$:

$$n(u,r) = \min\{n \in \mathbb{Z}_0^+ \mid F(u,r,n) \geq F_{th}(u)\}$$

- Satisfying $d(u)$ requires the QAN to **roughly** provide entangled pairs to UD $u$ at a distribution rate of

$$s(u,r) = \frac{d(u)}{P_e(u,r)} \cdot \frac{n(u,r)+1}{P(u,r,n(u,r))}$$

# Programming Project #1: Entanglement Distribution Optimization

- **Given:**
- A QAN with a limited entanglement generation rate
- Multiple RISs, each has its coverages (i.e., a subset of UDs)
- Multiple UDs, each has an all-or-nothing request with a profit
  - Requirements: fidelity threshold and demand expected key rate
- Each node has a coordinate position
- Assume that quantum memory/channels are always sufficient

- **Goal:** maximize the total profit

- **Constraints:**
- Each UD's request is either accepted or rejected
- The resulting fidelity and received key rate of accepted requests must meet the requirements

- The grade is proportional to total weighted received

# The Competition

- The grade is proportional to the total profit
- Basic: 60 (deadline)
  - A baseline solution (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
- Performan
  - [0%, 30%
  - [30%, 50
  - [50%, 75
  - [75%, 85
  - [85%, 90
  - [90%, 95
  - [95%, 10


t

eadline)

- Homework assistant (superb deadline)
  - +10

# The Competition

- The grade … t
- Basic: 60 (
  - A baselin
  - (Meet the
- Performan
  - [0%, 30%
  - [30%, 50
  - [50%, 75
  - [75%, 85
  - [85%, 90
  - [90%, 95
  - [95%, 10


We have
TIME LIMIT!

- Homework assistant (superb deadline)
  - +10

# 相信你們在做完作業以後

# Baseline Algorithm

- For each UD (from ID 0 to ID n-1)
  - Calculate $s(u, r)$ for each available path $r$ that can serve UD $u$
    - Note that $s(u, r)$ should not be greater than the residual entanglement generation rate of the QAN
    - Note that each path is either indirect via an available RIS or direct toward the QAN
    - Each available RIS is assigned to at most one UD
  - If such a path exists, accept UD $u$'s request and select the available path with the minimum value of $s(u, r)$ to serve UD $u$
    - If there is a tie, select the path via the RIS with a smaller ID
  - Once accepted, receive its profit and reduce the residual entanglement generation rate of the QAN by $s(u, r)$

# Input Sample: use scanf

Format:

#UDs #RISs alpha beta Ent_Gen_Rate
QAN_X_ Position QAN_Y_ Position #Covered_UDs
Covered_UD1_ID Covered_UD2_ID …
UD_ID X_Position Y_ Position Profit Exp_Rate Fidelity_Th
…
RIS_ID X_ Position Y_ Position #Covered_UDs
Covered_UD1_ID Covered_UD2_ID …
…

Ex:

```
2 1 0.4 1.2 55
500 500 0
0 1000 1000 10 100 0.7
1 0 0 8 100 0.7
0 1000 500 1
0
```

# Output Sample: use printf

Format:

#AccUDs

Acc_UD_ID    Used_RIS_ID_(if no RIS: -1)

...

Ex:

1

0    0

# Note

- Superb deadline: 10/9 Thu
- Deadline: 10/16 Thu
- 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