

Quantum teleport example

Damien Nguyen

Huawei 2012 Laboratories

September 9, 2019

damien1@huawei.com



Outline

1 Quantum Teleportation

Introduction

Programming quantum teleport with ProjectQ

Quantum Teleportation

Introduction

Quick recap'

Computational basis states

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

General qubit state

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

Hadamard basis states

$$|+\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Setting up the initial states

Alice has an unknown state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ that she wants to transmit to Bob.

Alice and Bob share a pair of entangled qubits $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Let's write the full joint qubit state:

$$|\psi_{\text{full joint}}\rangle = \frac{1}{\sqrt{2}} [\alpha |0\rangle (|00\rangle + |11\rangle) + \beta |1\rangle (|00\rangle + |11\rangle)]$$

Change to the Hadamard basis

Alice changes the basis of her unknown state to the Hadamard basis

$$\begin{aligned} H|\psi\rangle &= H(\alpha |0\rangle + \beta |1\rangle) \\ &= \frac{1}{\sqrt{2}} [\alpha (|0\rangle + |1\rangle) + \beta (|0\rangle - |1\rangle)] \end{aligned}$$

Now rewrite Alice's and Bob's joint qubit state (after the Hadamard transform):

$$\begin{aligned} |\psi_{\text{full joint}}\rangle &= \frac{1}{\sqrt{2}} \left[\frac{\alpha}{\sqrt{2}} (|0\rangle + |1\rangle) (|00\rangle + |11\rangle) + \frac{\beta}{\sqrt{2}} (|0\rangle - |1\rangle) (|00\rangle + |11\rangle) \right] \\ &= \frac{1}{2} \left[\alpha (|000\rangle + |011\rangle + |100\rangle + |111\rangle) + \beta (|001\rangle + |010\rangle - |101\rangle + |110\rangle) \right] \end{aligned}$$

Change to the Hadamard basis (contd.)

$$|\psi_{\text{full joint}}\rangle = \frac{1}{2} \begin{bmatrix} |00\rangle (\alpha |0\rangle + \beta |1\rangle) \\ |01\rangle (\alpha |1\rangle + \beta |0\rangle) \\ |10\rangle (\alpha |0\rangle - \beta |1\rangle) \\ |11\rangle (\alpha |1\rangle - \beta |0\rangle) \end{bmatrix}$$

Note that the state of the third qubit is very similar to the Alice's $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$!

If Alice measures her two qubits, Bob's qubit will collapse to one of the above states.

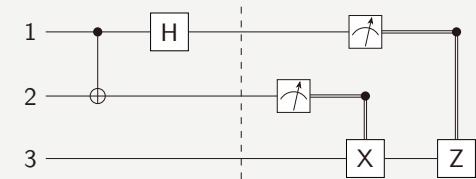
Bob only needs to know the two measurements results and apply some correction to his qubit if required and... Voilà!

Programming quantum teleportation

Quantum circuit

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$



Recall, at the dashed line, the state of our qubits is given by:

$$|\psi_{\text{full joint}}\rangle = \frac{1}{2} \begin{bmatrix} |00\rangle (\alpha|0\rangle + \beta|1\rangle) \\ |01\rangle (\alpha|1\rangle + \beta|0\rangle) \\ |10\rangle (\alpha|0\rangle - \beta|1\rangle) \\ |11\rangle (\alpha|1\rangle - \beta|0\rangle) \end{bmatrix}$$

Implementation I

```

1 from projectq import MainEngine
2 from projectq.ops import CNOT, H, Measure, X, Rz, Z
3 from projectq.meta import Dagger
4
5 def create_state(eng, qb):
6     H | qb
7     Rz(1.21) | qb
8
9
10 eng = MainEngine()
11
12 # ...

```

Implementation II

```

1 # ...
2
3 # Create a Bell pair
4 b1, b2 = eng.allocate_quireg(2)
5 H | b1
6 CNOT | (b1, b2)
7
8 # Create state to send
9 psi = eng.allocate_qubit()
10 create_state(eng, psi)
11
12 # ...

```

Implementation III

```

1  # ...
2
3  # Entangle psi with Alice's other qubit
4  CNOT | (psi, b1)
5
6  # Alice measures her two qubits
7  H | psi
8  Measure | psi
9  Measure | b1
10 msg_to_bob = [int(psi), int(b1)]
11
12 # ...

```

Implementation IV

```

1  # ...
2  # Bob received the message and applies the corrections (if needed)
3  if msg_to_bob[0]:
4      Z | b2
5  if msg_to_bob[1]:
6      X | b2
7
8  # Test if teleport successful: uncompute the state
9  with Dagger(eng):
10     create_state(eng, b2)
11 del b2 # will throw if not in classical state
12 eng.flush()

```

Whole code

```

1  from projectq import MainEngine
2  from projectq.ops import (CNOT, H, Measure,
3                             X, Rz, Z)
4  from projectq.meta import Dagger
5
6  def create_state(eng, qb):
7      H | qb
8      Rz(1.21) | qb
9
10 eng = MainEngine()
11
12 # Create a Bell pair
13 b1, b2 = eng.allocate_quireg(2)
14 H | b1
15 CNOT | (b1, b2)
16
17 # Create state to send
18 psi = eng.allocate_qubit()
19 create_state(eng, psi)
20
21 # Entangle with Alice's
22 CNOT | (psi, b1)
23
24 # Alice measures her two qubits
25 H | psi
26 Measure | psi
27 Measure | b1
28 msg_to_bob = [int(psi), int(b1)]
29
30 # Bob applies the corrections (if needed)
31 if msg_to_bob[0]:
32     Z | b2
33 if msg_to_bob[1]:
34     X | b2
35
36 # Test if teleport successful: uncompute
37 with Dagger(eng):
38     create_state(eng, b2)
39 del b2 # will throw if not in classical state
40 eng.flush()

```

Thank you for your attention

Any questions?

Bring digital to every person, home and organization for a fully connected intelligent world.

Copyright © 2019 Huawei Technologies Switzerland AG. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future products, portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice

