Quantum Teleportation

$|v\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^2$

One way:

Having many copies of this qubit, Asja can do measurements and ~~we~~ can approximate the values a and b to be $\tilde{a}$ and $\tilde{b}$. And then send it to Balvis using Classical values.

Other way:

If Asja & Balvis has already shared a entangled qubit pair, then Balvis can create the quantum state $|v\rangle$ only by receiving two classical bits information from Asja.

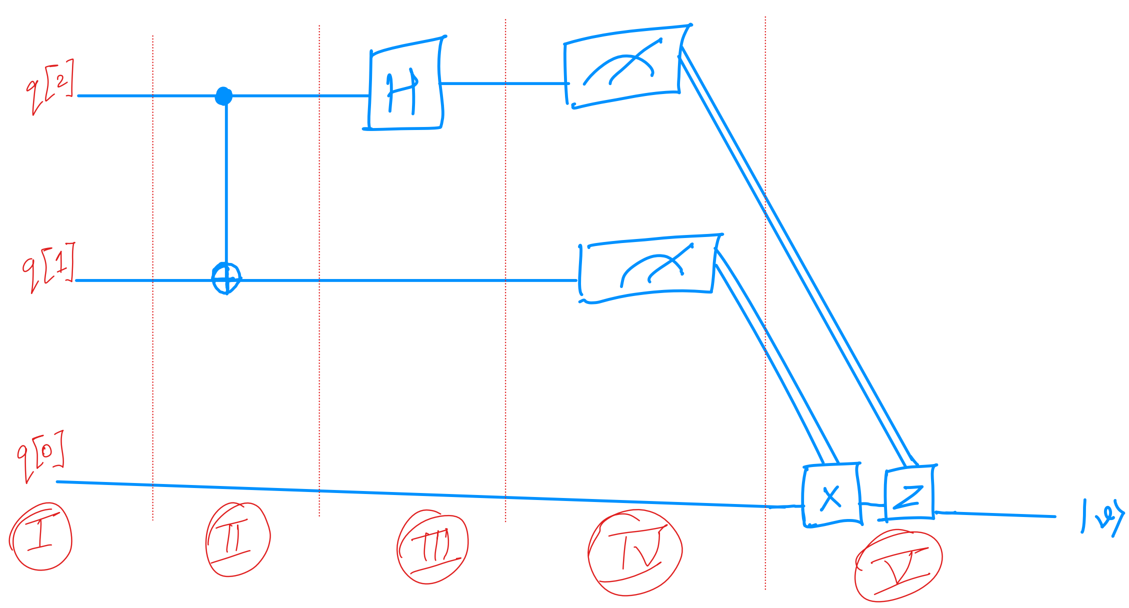
Diagram:

Asja's quantum message $|v\rangle = \begin{pmatrix} a \\ b \end{pmatrix} = a|0\rangle + b|1\rangle$

Asja's entangled qubit

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

Balvis's entangled qubit



I So initially, the quantum state of the system i.e including all the three qubits is given by

$$(a|0\rangle + b|1\rangle)\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) = \frac{1}{\sqrt{2}}\left(a\underbrace{|000\rangle}_{q[2],q[1],q[0]} + a|011\rangle + b|100\rangle + b|111\rangle\right)$$

CTI

II con. targ

$$CNOT(q[2], q[1]) \longrightarrow \frac{1}{\sqrt{2}}\left(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle\right)$$

III

$$H(q[2]) \longrightarrow \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(a|000\rangle + a|100\rangle + \frac{1}{\sqrt{2}}(a|011\rangle + a|111\rangle) + \frac{1}{\sqrt{2}}(b|010\rangle - b|110\rangle) + \frac{1}{\sqrt{2}}(b|001\rangle - b|101\rangle)\right.$$

$$H|0yz\rangle = \frac{1}{\sqrt{2}}|0yz\rangle + \frac{1}{\sqrt{2}}|1yz\rangle$$
$$H|1yz\rangle = \frac{1}{\sqrt{2}}|0yz\rangle - \frac{1}{\sqrt{2}}|1yz\rangle$$

On rearranging such that we separate same of Asja's qubits with Balvis's qubits,

$$= \frac{1}{2}\left(a|000\rangle + b|001\rangle\right) + \frac{1}{2}\left(a|011\rangle + b|010\rangle\right) + \frac{1}{2}\left(a|100\rangle - b|101\rangle\right) + \frac{1}{2}\left(a|111\rangle - b|110\rangle\right)$$

$$= \frac{1}{2}|00\rangle\left(a|0\rangle + b|1\rangle\right) + \frac{1}{2}|01\rangle\left(a|1\rangle + b|0\rangle\right) + \frac{1}{2}|10\rangle\left(a|0\rangle - b|1\rangle\right) + \frac{1}{2}|11\rangle\left(a|1\rangle - b|0\rangle\right)$$

IV

Measurement by Asja on her qubits (q[2], q[1])

Each state has a probability of $\frac{1}{4}$ to occur

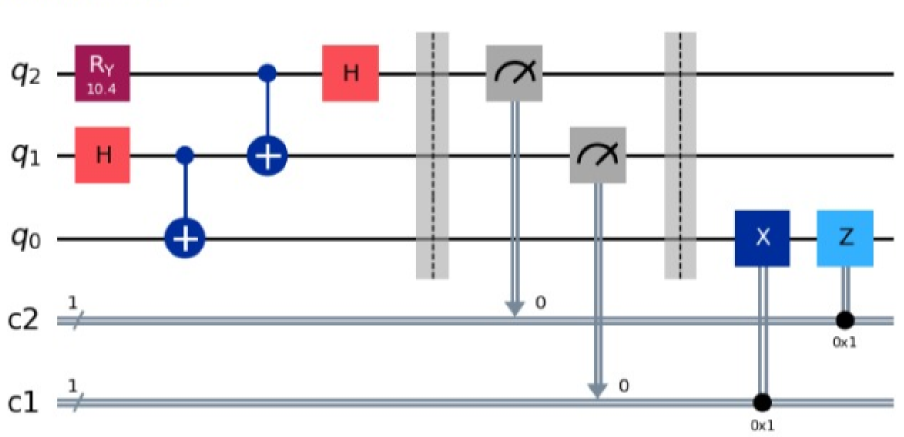
V

If she measure's	then Balvis's qubit is in the state
00	$|\psi_{00}\rangle \equiv a|0\rangle + b|1\rangle \longrightarrow |\psi\rangle$
01	$|\psi_{01}\rangle \equiv a|1\rangle + b|0\rangle \xrightarrow{\text{NOT}} |\psi\rangle$
10	$|\psi_{10}\rangle \equiv a|0\rangle - b|1\rangle \xrightarrow{Z} |\psi\rangle$
11	$|\psi_{11}\rangle \equiv a|1\rangle - b|0\rangle \xrightarrow{\text{NOT} \ \& \ Z} |\psi\rangle$

- No Cloning

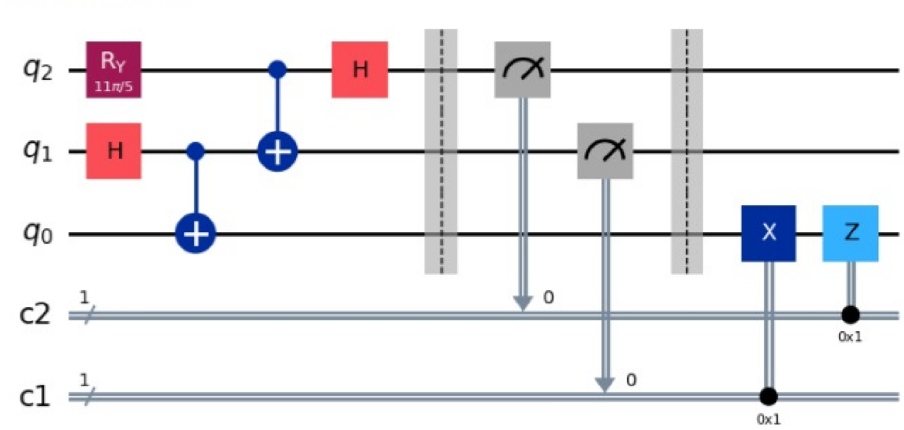
```python
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from qiskit_aer import StatevectorSimulator
from math import pi, cos, sin
from random import randrange

# quantum circuit with three qubits and two bits
q =  QuantumRegister(3,"q")
c1 = ClassicalRegister(1,"c1")
c2 = ClassicalRegister(1,"c2")
qc = QuantumCircuit(q,c1,c2)

# rotate the first qubit by random angle
r = randrange(100)
theta = 2*pi*(r/100) # radians
print("the picked angle is",r*3.6,"degrees and",theta,"radians")
a = cos(theta)
b = sin(theta)
print("a=",round(a,4),"b=",round(b,4))
qc.ry(2*theta,q[2])

# creating an entanglement between q[1] and q[0]
qc.h(q[1])
qc.cx(q[1],q[0])

# CNOT operator by Asja on her qubits where q[2] is the control qubit
qc.cx(q[2],q[1])

# Hadamard operator by Asja on q[2]
qc.h(q[2])

qc.barrier()

# the measurement done by Asja
qc.measure(q[2],c2)
qc.measure(q[1],c1)

qc.barrier()

# post-processing done by Balvis
qc.x(q[0]).c_if(c1,1)
qc.z(q[0]).c_if(c2,1)

# draw the circuit
display(qc.draw(output='mpl',reverse_bits=True))

# read the state vector
job = StatevectorSimulator().run(qc,optimization_level=0,shots=1)
current_quantum_state=job.result().get_statevector(qc).data
print("the state vector is")
for i in range(len(current_quantum_state)):
    print(round(current_quantum_state[i].real,4))
print()

classical_outcomes = ['00','01','10','11']

for i in range(4):
    if (current_quantum_state[2*i].real != 0) or (current_quantum_state[2*i+1].real != 0):
        print("the classical outcome is",classical_outcomes[i])
```

```
the picked angle is 298.8 degrees and 5.215043804959056 radians
a= 0.4818 b= -0.8763
/tmp/ipython-input-1068032031.py:40: DeprecationWarning: The method ``qiskit.circuit.instructionset.Inst
  qc.x(q[0]).c_if(c1,1)
/tmp/ipython-input-1068032031.py:41: DeprecationWarning: The method ``qiskit.circuit.instructionset.Inst
  qc.z(q[0]).c_if(c2,1)
```




```
the state vector is
-0.0
0.0
0.4818
-0.8763
0.0
0.0
0.0
0.0

the classical outcome is 01
```

```
the picked angle is 198.0 degrees and 3.455751918948773 radians
a= -0.9511 b= -0.309
/tmp/ipython-input-1068032031.py:40: DeprecationWarning: The method ``qiskit.circuit.instructionset.Ins
  qc.x(q[0]).c_if(c1,1)
/tmp/ipython-input-1068032031.py:41: DeprecationWarning: The method ``qiskit.circuit.instructionset.Ins
  qc.z(q[0]).c_if(c2,1)
```




```
the state vector is
-0.0
0.0
-0.0
0.0
-0.9511
-0.309
0.0
0.0

the classical outcome is 10
```