```
import numpy as np
import matplotlib.pyplot as plt
```

```
# Step 1: Quantum Gates
def hadamard():
    return np.array([[1, 1], [1, -1]]) / np.sqrt(2)

def cnot():
    return np.array([[1,0,0,0], [0,1,0,0], [0,0,0,1], [0,0,1,0]])

def pauli_x():
    return np.array([[0, 1], [1, 0]])

def pauli_z():
    return np.array([[1, 0], [0, -1]])
```

```
# Step 2: Create Entangled Pair (Bell State)
def create_bell_pair():
    # |00⟩ → |Φ+⟩ = (|00⟩ + |11⟩)/√2
    state = np.array([1, 0, 0, 0])  # |00⟩
    h_gate = np.kron(hadamard(), np.eye(2))  # H ⊗ I
    state = h_gate @ state
    state = cnot() @ state
    return state
```

```
# Step 3: Bell Measurement
def bell_measurement(state):
    # Simulate measurement in Bell basis
    probs = np.abs(state)**2
    outcome = np.random.choice(4, p=probs)
    return outcome, ['00', '01', '10', '11'][outcome]
```

```
# Step 4: Quantum Teleportation Protocol
def quantum_teleport(qubit_to_send):
    print(f"Teleporting qubit: {qubit_to_send}")

    # Create entangled pair for Alice and Bob
    bell_pair = create_bell_pair()
    print("Created Bell pair")

    # Combine qubit to send with Alice's entangled qubit
    full_state = np.kron(qubit_to_send, bell_pair)

    # Alice's Bell measurement (simplified)
    measurement_result = np.random.randint(0, 4)
    classical_bits = format(measurement_result, '02b')
    print(f"Alice measures: {classical_bits}")

    # Bob applies corrections based on Alice's measurement
    bob_qubit = np.array([1, 0])  # Start with |0⟩

    if classical_bits[0] == '1':  # Apply Z correction
        bob_qubit = pauli_z() @ bob_qubit
    if classical_bits[1] == '1':  # Apply X correction
```

```
        bob_qubit = pauli_x() @ bob_qubit

    # Bob now has the teleported state
    return bob_qubit
```

```
# Step 5: Test Teleportation
def test_teleportation():
    # Test states
    states = {
        "|0⟩": np.array([1, 0]),
        "|1⟩": np.array([0, 1]),
        "|+⟩": np.array([1, 1]) / np.sqrt(2),
        "|-⟩": np.array([1, -1]) / np.sqrt(2)
    }

    for name, state in states.items():
        print(f"\n--- Teleporting {name} ---")
        received = quantum_teleport(state)
        fidelity = abs(np.dot(state.conj(), received))**2
        print(f"Original: {state}")
        print(f"Received: {received}")
        print(f"Fidelity: {fidelity:.3f}")
```

```
# Step 6: Visualization
def plot_teleportation_fidelity():
    fidelities = []
    for _ in range(20):
        state = np.array([1, 0])  # |0⟩
        received = quantum_teleport(state)
        fidelity = abs(np.dot(state.conj(), received))**2
        fidelities.append(fidelity)

    plt.figure(figsize=(8, 4))
    plt.plot(fidelities, 'bo-')
    plt.axhline(y=1, color='r', linestyle='--', label='Perfect')
    plt.ylabel('Fidelity')
    plt.xlabel('Trial')
    plt.title('Quantum Teleportation Fidelity')
    plt.legend()
    plt.grid(True)
    plt.show()
```

```
def main():
    print("=== QUANTUM TELEPORTATION ===")
    test_teleportation()
    plot_teleportation_fidelity()
    print("=== COMPLETE ===")
```

```
main()
```

```
=== QUANTUM TELEPORTATION ===

--- Teleporting |0⟩ ---
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Original: [1 0]
Received: [1 0]
Fidelity: 1.000

--- Teleporting |1⟩ ---
Teleporting qubit: [0 1]
Created Bell pair
Alice measures: 10
Original: [0 1]
Received: [1 0]
Fidelity: 0.000

--- Teleporting |+⟩ ---
Teleporting qubit: [0.70710678 0.70710678]
Created Bell pair
Alice measures: 01
Original: [0.70710678 0.70710678]
Received: [0 1]
Fidelity: 0.500

--- Teleporting |-⟩ ---
Teleporting qubit: [ 0.70710678 -0.70710678]
Created Bell pair
Alice measures: 11
Original: [ 0.70710678 -0.70710678]
Received: [0 1]
Fidelity: 0.500
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 10
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 10
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
```

```
Created Bell pair
Alice measures: 10
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 11
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 10
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 10
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 01
Teleporting qubit: [1 0]
Created Bell pair
Alice measures: 00
```