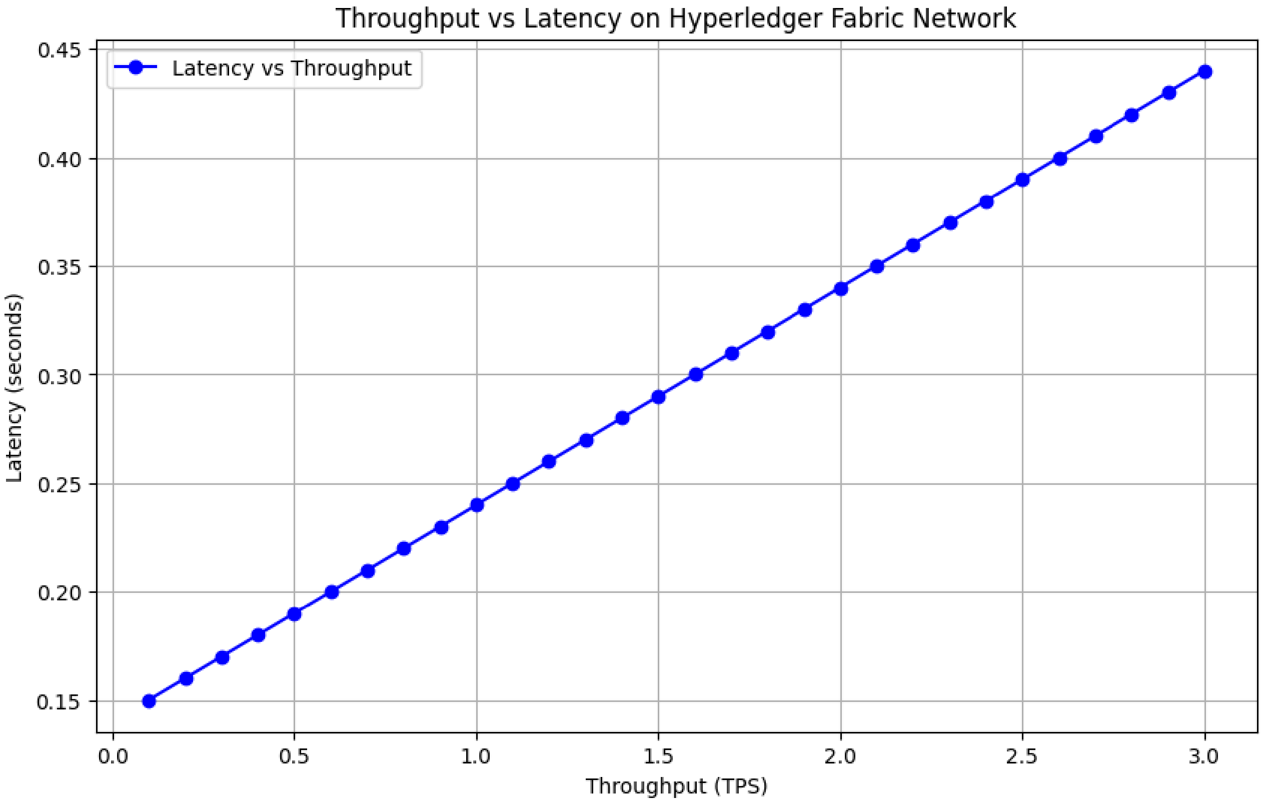
```
import csv
with open('throughput_latency_sim_network.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    print("Column Names:", reader.fieldnames)

# Continue with the previous logic to check if the columns are correct
throughput = []
latency = []
```

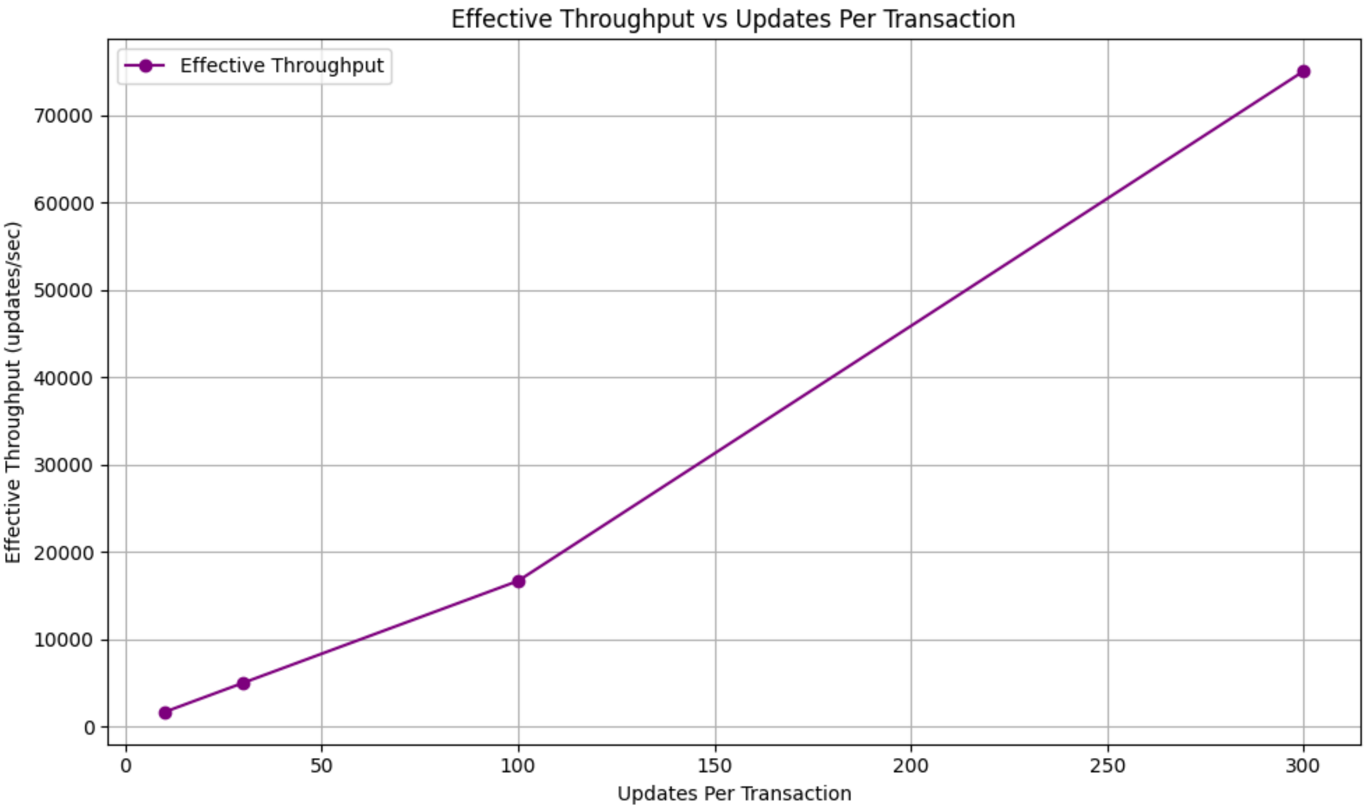
```
Column Names: ['Transaction ID', 'Submit Time', 'Commit Time', 'Latency (ms)', 'Throughput (TPS)']
```

```
import matplotlib.pyplot as plt
import csv

# Read the CSV data
throughput = []
latency = []

with open('throughput_latency.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        # Convert Latency from ms to seconds for easier comparison
        latency.append(float(row['Latency (ms)']) / 1000)
        throughput.append(float(row['Throughput (TPS)']))

# Print the collected data to verify
print("Throughput:", throughput)
print("Latency:", latency)

# Plot the data
plt.figure(figsize=(10, 6))
plt.plot(throughput, latency, marker='o', color='b', label='Latency vs Throughput')
plt.title('Throughput vs Latency on Hyperledger Fabric Network')
plt.xlabel('Throughput (TPS)')
plt.ylabel('Latency (seconds)')
plt.grid(True)
plt.legend()
plt.show()
```

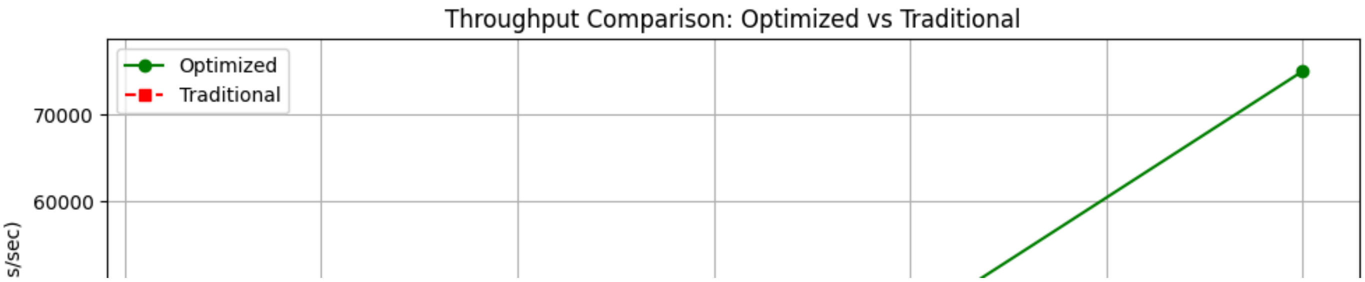
```
Throughput: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.
Latency: [0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3, 0.31, 0.32, 0.33
```



```
import matplotlib.pyplot as plt
import csv

updates = []
throughput = []

with open('high_throughput_results.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        updates.append(int(row['Updates Per Transaction']))
        throughput.append(float(row['Effective Throughput (updates/sec)']))

plt.figure(figsize=(10, 6))
plt.plot(updates, throughput, marker='o', linestyle='-', color='purple', label='Effective Throughput')
plt.title('Effective Throughput vs Updates Per Transaction')
plt.xlabel('Updates Per Transaction')
plt.ylabel('Effective Throughput (updates/sec)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
import csv

updates = []
optimized_throughput = []
traditional_throughput = []

with open('high_throughput_comparison.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        if row['Method'] == 'Optimized':
            updates.append(int(row['Updates Per Transaction']))
            optimized_throughput.append(float(row['Effective Throughput (updates/sec)']))
        elif row['Method'] == 'Traditional':
            traditional_throughput.append(float(row['Effective Throughput (updates/sec)']))

plt.figure(figsize=(10, 6))
plt.plot(updates, optimized_throughput, marker='o', linestyle='-', color='green', label='Optimized')
plt.plot(updates, traditional_throughput, marker='s', linestyle='--', color='red', label='Traditional')
plt.title('Throughput Comparison: Optimized vs Traditional')
plt.xlabel('Updates Per Transaction')
plt.ylabel('Effective Throughput (updates/sec)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```


Throughput Comparison: Optimized vs Traditional
Optimized
Traditional
70000
60000
s/sec)