```python
import numpy as np
import pandas as pd
from collections import deque
```

Problem 1.

A.

The packet will require $T + L/R + l/c$ time to finish transmitting

B.

Each packet requires $4\text{kb}/1000\text{kb} = 4\text{ms}$ of processing time at A. propagation delay is fixed 10ms. Thus, the packets can be modelled as follows:

Packet	arrival time	send complete time	receipt time
1	0 ms	4 ms	14ms
2	1 ms	8 ms	18 ms
3	1.5 ms	12 ms	22 ms

C.

```python
res_df = pd.DataFrame(columns= ["Avg. InterArrival Time", "Avg. Packet Length", "Avg. Trans

def simulate(lambd, mu, res_df = res_df, num_packets = 10000):
    arrival_t = np.random.exponential(1/lambd, [num_packets])
    avg_arrival_interval = np.mean(arrival_t)
    length_t = np.random.exponential(1/mu, [num_packets])
    avg_packet_length = np.mean(length_t)
    service_t = np.zeros(num_packets)
    q = deque()
    # q.append((length_t[0], 0)) # length, entrance time
    next_service = 0
    finished_service = 0
    cur_time = 0
    # print(arrival_t, length_t)
    while q or finished_service < num_packets:
        # print(q, next_service, finished_service, cur_time)
        if (q and next_service == num_packets) or (q and q[0][0] < arrival_t[next_service]
            if next_service < num_packets:
                arrival_t[next_service] -= q[0][0]
            cur_time += q[0][0]
            service_t[finished_service] = cur_time - q[0][1]
            finished_service += 1
            q.popleft()
        elif (q and q[0][0] == arrival_t[next_service]):
            next_service += 1
            cur_time += q[0][0]
            service_t[finished_service] = cur_time - q[0][1]
            finished_service += 1
```

```
                q.popleft()
                q.append(length_t[next_service])
            elif (not q) or (q and q[0][0] > arrival_t[next_service]):
                if q:
                    q[0][0] -= arrival_t[next_service]
                cur_time += arrival_t[next_service]
                q.append([length_t[next_service], cur_time])
                next_service += 1
        res_df.loc[f"lambda: {lambd}, mu: {mu}"] = [avg_arrival_interval, avg_packet_length, np
simulate(1, 0.8, res_df)
simulate(1, 1, res_df)
simulate(1, 1.2, res_df)
display(res_df)
```

	Avg. InterArrival Time	Avg. Packet Length	Avg. Transport Time
lambda: 1, mu: 0.8	1.006014	1.238507	1253.221346
lambda: 1, mu: 1	0.998661	0.976398	35.339267
lambda: 1, mu: 1.2	1.012268	0.825468	4.225015

Problem 2.

(Diagrams below)

A.

The state space is $\{0, 1\}$. All are communicating and recurrent because $E[A(t)] < M$

B.

The State space is non-negative integers, and has $|R| = \infty$. However, it is recurrent because $E[A(t)] < M$

C.

The State space is non-negative integers, and has $|R| = \infty$. However, it is recurrent because $E[A(t)] < M$

D.

The State space is non-negative integers, and has $|R| = \infty$. However, it is recurrent because $E[A(t)] < M$

E.

The State space is non-negative integers, and has $|R| = \infty$. However, it is transient because $E[A(t)] > M$

Problem 3.

As from class, to evaluate the stability of the system, we apply the Foster-Lyapunov theorem with $V(x) = x$. With this, we have:

$$
\begin{aligned}
E[V(t+1) - V(t)|X(t) = x_0] &= E[(X(t+1) - M(t+1))^+ + A(t+1) - X(t)|X(t) = x_0] \\
&= E[(X(t+1) - M(t+1))^+ + A(t+1)|X(t) = x_0] - x_0 \\
&= x_0 - E[M(X)] + E[A(t)] - x_0 \quad x_0 > 0 \qquad = E[A(t)] - E[M(X)]
\end{aligned}
$$

We know that $E[M(x)] = \frac{K}{2}$. It remains to evaluate $A(t)$, which we are told is conditional on a binary markov chain itself.

$$\begin{aligned} E[A(t)] &= E[A_1(t)(1 - S(t)) + A_2(t)S(t)] \\ &= E[p_1(1 - S(t)) + p_2 S(t)] \end{aligned}$$

from the given definitions of i.i.d $A_1(t)$ and $A_2(t)$. Thus, it suffices to evaluate the EV of $S(t)$. We aren't given the values of this binary chain, so assume that the first state has value k and the second one j. Given that it is binary, we know that it is stable, and it suffices to evaluate the EV of the chain.

The transition matrix of this chain is $\begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}$, which we want to find what it converges to.

$$\begin{aligned} \pi_0(1 - p) + \pi_1(q) = \pi_0 \\ \pi_0(p) + \pi_1(1 - q) = \pi_1 \end{aligned}$$

for $\pi(0) + \pi(1) = 0$ results in $\pi = \begin{bmatrix} \frac{q}{p+q} \\ \frac{p}{p+q} \end{bmatrix}$. This implies that the EV of this system is $\frac{jq+kp}{p+q}$, so $E[A(t)] = \frac{(jq+kp)(1-p_1+p_2)}{p+q}$. This needs to be less than $\frac{K}{2}$, which is a necessary and sufficient condition for stability.

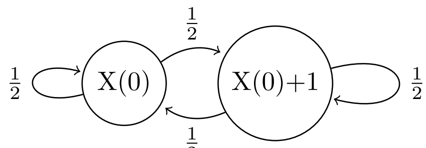

Figure 1: A.

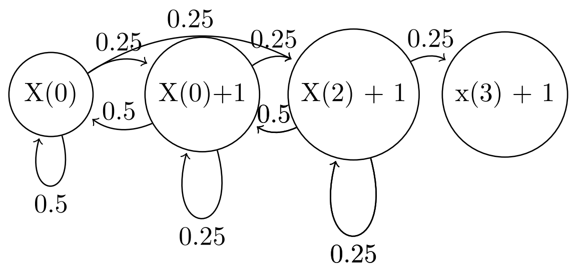

Figure 2: B.

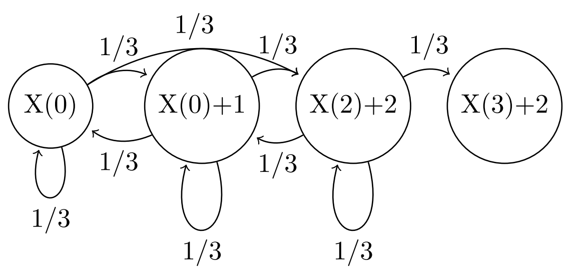

Figure 3: C.

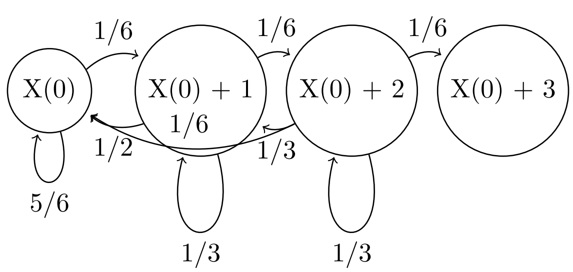

Figure 4: D.

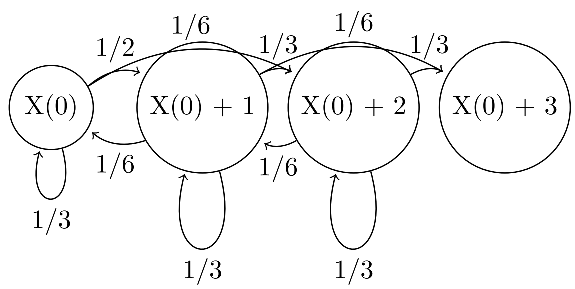

Figure 5: E.