tr_fast and tr_slow are immediate transitions with probability = 1
but the priority of tr_fast is greater than the priority
of tr_slow, so whenever the fast server is free, a job waiting in “buf” will go to it.

tr_fastserv and tr_slowserv are timed transitions with rates of mu_f and mu_s, respectively.
Problem #2 in HW#3

The places and transitions are the same for all three cases.

Associate an enabling function each with cpu_failure, cpu_repair, mem_failure and mem_repair.

Associate a rate function each with cpu_failure, cpu_repair, mem_failure and mem_repair.

The three cases differ in how you define the enabling functions and the rate functions.

You need to code and run 3 separate SPNP programs for the three cases.
Problem #3 in HW#3
(\(\alpha = 2\), so a token represents one-half slot)

A new transition “T8” is added to account for subrating for low-priority clients

For the homework in which only the middle partition exists, the enabling condition for “T8” is \(\text{mark}("RS") == 0\)

For case study #2 in which all three partitions exist, the enabling condition for “T8” is \(\text{mark}("RL") == 0 \&\& \text{mark}("RS") == 0\)

Reward assignment function for calculating the population of low-priority clients:

```cpp
reward_type
population_low_priority()
{return (mark("SLL") + mark("SL")/2);}
```
Problem #4 in HW#3

Five places: CPU, temp, disk1, disk2, and term each holding tokens (jobs).
Place “temp” is a temporary place for holding jobs departing from CPU.

Initially 4 tokens (jobs) are in place “term”.

Four timed transitions: TCPU, Tdisk1, Tdisk2, and Tterm
Only Tterm has a variable rate. All others have fixed rates.

Three immediate transitions: Tp0, Tp1, Tp2 with probabilities of 0.1, 0.667, and 0.233 respectively.
Place “temp” is the input place to these three immediate transitions.