list. For the scope of this assignment, indices should start at 1.

3. `public class Simulator` - This class contains the main method that tests your simulation. You should not use hard-coded numbers. Instead, all values should be received from user input. The following contains the list of parameters you must keep track of:

- `Router dispatcher` - Level 1 router
- `Collection<Router> routers` - Level 2 routers
- `int totalServiceTime` - contains the running sum of the total time each packet is in the network. The service time per packet is simply *the time it has arrived to the Destination minus the time when the packet was created*. When a packet counter reaches 0, dequeue it from the router queue and add the time to the total time. Ignore the leftover Packets in the network when simulation time is up.
- `int totalPacketsArrived` - contains the total number of packets that has been successfully forwarded to the destination. When a packet counter reaches 0, dequeue it from the router queue and increase this count by 1.
- `int packetsDropped` - records the number of packets that have been dropped due to a congested network.
  Note: this can only happen when `sendPacketTo(Collection routers)` throws an exception.
- `double arrivalProb` - the probability of a new packet arriving at the Dispatcher.
- `int numIntRouters` - the number of Intermediate routers in the network.
- `int maxBufferSize` - the maximum number of Packets a Router can accommodate for.
- `int minPacketSize` - the minimum size of a Packet
- `int maxPacketSize` - the maximum size of a Packet
- `int bandwidth` - the maximum number of Packets the Destination router can accept at a given simulation unit
- `int duration` - the number of simulation units

Implement the following methods:

- `public double simulate(/* any arguments above you find fit */)` - runs the simulator as described in the specs. Calculate and return the average time each packet spends within the network.
- `private int randInt(int minVal, int maxVal)` - this will be your helper method that can generate a random number between minVal and maxVal, inclusively. Return that randomly generated number.
- `public static void main(String[] args)` - the main() method will prompt the user for inputs to the simulator. It will then run the simulator, and outputs the result. Prompt the user whether he or she wants to run another simulation.

4. Any exception class that you find fit.

SAMPLE INPUT/OUTPUT:

```
// comment in green, input in red, output in black

// Your output should following the following out format. Do not expect the
numbers to be the same.

Starting simulator...

Enter the number of Intermediate routers: 4

Enter the arrival probability of a packet: 0.5

Enter the maximum buffer size of a router: 10

Enter the minimum size of a packet: 500

Enter the maximum size of a packet: 1500
Enter the bandwidth size: 2

Enter the simulation duration: 25

Time: 1
Packet 1 arrives at dispatcher with size 576.
Packet 2 arrives at dispatcher with size 1044.
Packet 1 sent to Router 1.
Packet 2 sent to Router 2.
R1: {[1, 1, 5]}
R2: {[2, 1, 10]}
R3: {}
R4: {}

Time: 2
Packet 3 arrives at dispatcher with size 922.
Packet 3 sent to Router 3.
R1: {[1, 1, 4]}
R2: {[2, 1, 9]}
R3: {[3, 2, 9]}
R4: {}

Time: 3
Packet 4 arrives at dispatcher with size 1301.
Packet 5 arrives at dispatcher with size 574.
Packet 4 sent to Router 4.
Packet 5 send to Router 1.
R1: {[1, 1, 3], [5, 3, 5]}
R2: {[2, 1, 8]}
R3: {[3, 2, 8]}
R4: {[4, 3, 13]}

Time: 4
Packet 6 arrives at dispatcher with size 1283.
Packet 7 arrives at dispatcher with size 552.
Packet 6 sent to Router 2.
Packet 7 send to Router 3.
R1: {[1, 1, 2], [5, 3, 5]}
R2: {[2, 1, 7], [6, 4, 12]}
R3: {[3, 2, 7], [7, 4, 5]}
R4: {[4, 3, 12]}

Time: 5
No packets arrived.
R1: {[1, 1, 1], [5, 3, 5]}
R2: {[2, 1, 6], [6, 4, 12]}
R3: {[3, 2, 6], [7, 4, 5]}
R4: {[4, 3, 11]}

Time: 6
Packet 8 arrives at dispatcher with size 900.
Packet 8 sent to Router 4.
Packet 1 has successfully reached its destination: +5
R1: {[5, 3, 5]}
R2: {[2, 1, 5], [6, 4, 12]}
R3: {[3, 2, 5], [7, 4, 5]}
R4: {[4, 3, 11], [8, 6, 9]}
```

```
// +5 means it took 5 simulation units for the packet to travel through the
network.

Time: 7
Packet 9 arrives at dispatcher with size 1410.
Packet 10 arrives at dispatcher with size 913.
Packet 9 sent to Router 1.
Packet 10 sent to Router 1.
R1: {[5, 3, 4], [9, 7, 14], [10, 7, 9]}
R2: {[2, 1, 4], [6, 4, 12]}
R3: {[3, 2, 4], [7, 4, 5]}
R4: {[4, 3, 10], [8, 6, 9]}

// According to our selection algorithm, both packets should end up in Router 1.

// Time 8 - 10 not shown in sample i/o

Time: 11
Packet 17 arrives at dispatcher with size 830.
Packet 17 sent to Router 3.
Packet 5 has successfully reached its destination: +8
Packet 2 has successfully reached its destination: +10
R1: {[9, 7, 14], [10, 7, 9], ... ... ...}
R2: {[6, 4, 12], ... ... ...}
R3: {[3, 2, 0], [7, 4, 5], ... ... ...} // Packet 3 will stay in the router buffer
R4: {[4, 3, 6], [8, 6, 9], ... ... ...}

// Packet 3 will not be accepted by the Destination router due to limited
bandwidth. It will stay in the router queue until processed in the next
simulation unit.

Time: 12
Packet 18 arrives at dispatcher with size 1201.
Packet 19 arrives at dispatcher with size 667.
Packet 20 arrives at dispatcher with size 920.
Packet 18 sent to Router 1.
Packet 19 sent to Router 2.
Packet 20 sent to Router 4.
Packet 3 has successfully reached its destination: +10
R1: {[9, 7, 13], [10, 7, 9], ... ... ...}
R2: {[6, 4, 11], ... ... ...}
R3: {[7, 4, 5], ... ... ...}
R4: {[4, 3, 5], [8, 6, 9], ... ... ...}

// Time 13 - 23 not shown in sample i/o

Time: 24
Packet 45 arrives at dispatcher with size 1008.
Packet 46 arrives at dispatcher with size 573.
Packet 45 sent to Router 3.
Network is congested. Packet 46 is dropped.
Packet 6 has successfully reached its destination: +20
R1: {[9, 7, 13], [10, 7, 9], ... ... ...}
R2: {[[11, 8, 7], ... ... ...}
R3: {[12, 8, 6], ... ... ...}
R4: {[8, 6, 2], ... ... ...}

// If all router queues are full, we drop the packet. It cannot be sent through
the network.
```

```
Time: 25
Packet 47 arrives at dispatcher with size 710.
Packet 48 arrives at dispatcher with size 993.
Packet 47 sent to Router 2.
Network is congested. Packet 48 is dropped.
R1: {[9, 7, 12], [10, 7, 9], ... ... ...}
R2: {[[11, 8, 6], ... ... ...}
R3: {[12, 8, 5], ... ... ...}
R4: {[8, 6, 1], ... ... ...}

Simulation ending...
Total service time: 70
Total packets served: 6
Average service time per packet: 11.67
Total packets dropped: 2

// This is the final result output. You will be graded on the trends through
simulations with various parameters. You can find the requirements on the grading
sheet for this assignment.

Do you want to try another simulation? (y/n): n

Program terminating successfully...
```

Course Info | Schedule | Sections | Announcements | Homework | Exams | Help/FAQ | Grades | HOME