Instruction to run the Hot-Potato Routing Simulation

The simulation allows the user to input 6 parameters.
The diameter of the torus network, the number of processors that the simulation is to be run on, the duration of the simulation, the number of packet injection applications, whether the routers should absorb sleeping packets, and the number of kernel processes.

The simulation may also be run in sequential or parallel mode.
The simulation parameters may be input when the program prompts the user for them.
The simulation parameters may also be input through the command line interface.

The following example shows both of these methods.
The following example is shows how to run the simulation in parallel mode.

Parallel Mode

User Prompted Input Parameters

The simulation may be run using the following command and inputting the requested parameters as follows:

./routing-par
What is the number of rows in the routing matrix? 8
What is the number of processors? 4
What is the simulation duration? 100,000
How many routers should inject packets in percentage (i.e. 0, 50, 100)? 25
Should a router absorb a sleeping packet? 1
Number of KPs : 8

This method can also be accomplished by redirecting an input parameter file.

Command Line Input Parameters

The simulation can also be run using command line arguments.
The following command runs the same simulation as above:

```
                N    PEs SimTime   %Sources Absorb? KPs  Redirect output
./routing-par 8    4    100,000  25        1       8    >  out-par1.txt
```

Note that the output is redirected to a file.

Sequential Mode

The above example can also be run in sequential mode as follows:

./routing-seq
What is the number of rows in the routing matrix? 8
What is the number of processors? 4
What is the simulation duration? 100,000
How many routers should inject packets in percentage (i.e. 0, 50, 100)? 25
Should a router absorb a sleeping packet? 1
Number of KPs : 8

Command Line Input Parameters

```
               N    PEs SimTime   %Sources Absorb? KPs  Redirect output
./routing-seq 8    4    100,000   25        1       8    >  out-par1.txt
```