Appendix B

SICOSYS Tutorial

In this chapter will give rough description of the user interface to SICOSYS illustrated by a simple example. This will help the user get acquainted with the simulator.

B.1 Work Environment

In order to successfully run SICOSYS there are several files that have to be accessible to the simulator executable. This section will describe the directory structure needed by SICOSYS.

Basically the SICOSYS work environment is composed by five files:

`ATCSimul` Is the executable program. It can be anywhere accessible by the user. It can be in a system directory like `/usr/bin`, so it is accessible to many users, or it can be in a user directory when only one user needs it.

`ATCSimul.ini` It is read by SICOSYS at startup and tells it where the three SGML files are. It should be in the working directory. Therefore every user must have his own copy (or more than one).

`Router.sgm` Describes the structure of the routers that can be used. Its location must be pointed by `ATCSimul.ini`.

`Network.sgm` Describes the networks that can be used. Its location must be pointed by `ATCSimul.ini`.

`Simula.sgm` Describes the parameters of the different simulations. Its location must be pointed by `ATCSimul.ini`.

A more detailed description of all files is given in the next section. In addition, there are some tools to help the debugging tasks. These are explained in the Reference Manual

`BView` is a simple program to convert the buffer information file to a readable format.

`xbuffer` is a graphical application to represent the buffer information file.

`xsimul` is a graphical interface to SICOSYS. It executes simulations while presenting the evolution of the latency.

B.2 Configuration Files

All configuration files are coded in SGML. SGML is a markup language similar to HTML, in fact SGML can be considered a superset of HTML. A SGML file consists of a set of tags, each with a variable number of parameters. A tag can contain other tags with in it, conforming a hierarchical structure. This suits the the hierarchical descriptions of routers in SICOSYS perfectly. A simple introduction to the format of the files will be given throughout this section. For a detailed explanation of all the tags that are available see the reference manual.

B.2.1 `ATCSimul.ini`

The first, and most simple, configuration file is `ATCSimul.ini`. SICOSYS uses this file to find the rest of the configuration files it needs. This is very simple, as can be seen in the following example.

```
<SimulationFile  id="../sgm/Simula.sgm"  >
<NetworkFile     id="../sgm/Network.sgm" >
<RouterFile      id="../sgm/Router.sgm"  >
```

The three tags that can be found in this file give the location of each of the three files describing routers, networks and simulations. The locations can be absolute or relative to the working directory.

B.2.2 `Simula.sgm`

This file describes the scenarios that will be simulated. In addition to selecting a network from `Network.sgm`, other simulation parameters are given. These parameters include simulation length, seed for random numbers or traffic pattern and injection rate. Here follows a simple example:

```
<Simulation id="TORUS8x8-CT">
  <Network           id="TORUS8x8-CT">
  <SimulationCycles  id=20000>
  <TrafficPattern    id="MODAL" type="PERFECT-SUFFLE">
  <Seed              id=113>
  <Load              id=0.1>
  <MessageLength     id=1>
  <PacketLength      id=16>
</Simulation>
```

B.2.3 Network.sgm

The `Network.sgm` file describes the way the routers are connected to each other. This is, it describes the topology of the network to be used. The format of this file is fairly simple. Each network is defined by a single tag specifying the topology, the name, the router to be used, the size of the network and the delay of the wires between neighboring nodes. As an example, a 64 node torus network using the router shown above can be described easily as follows.

```
<TorusNetwork id="TORUS8x8-CT" sizeX=8 sizeY=8 router="
   DOR2D-CT" delay=0>
```

B.2.4 Router.sgm

This file has a description of all the routers that can be simulated. In addition to the router structure, there are other parameters that are specified in this description, like the buffer capacity or the delay of each component of the router.

In the following example the description of a simple 2D router is shown.

```
<Router id=DOR2D-CT inputs=4 outputs=4 bufferSize=64
        bufferControl=CT routingControl=DOR-BU >
   <Injector id="INJ">
   <Consumer id="CONS">

   <Buffer id="BUF1" type="X+"   dataDelay=2  size=32>
   <Buffer id="BUF2" type="X-"   dataDelay=2>
   <Buffer id="BUF3" type="Y+"   dataDelay=2>
   <Buffer id="BUF4" type="Y-"   dataDelay=2>
   <Buffer id="BUF5" type="Node" dataDelay=2>
```

```
<Routing id="RTG1" type="X+"   headerDelay=1 dataDelay
   =0>
<Routing id="RTG2" type="X-"   headerDelay=1 dataDelay
   =0>
<Routing id="RTG3" type="Y+"   headerDelay=1 dataDelay
   =0>
<Routing id="RTG4" type="Y-"   headerDelay=1 dataDelay
   =0>
<Routing id="RTG5" type="Node" headerDelay=1 dataDelay
   =0>

<Crossbar id="CROSSBAR" inputs=5 outputs=5 type="CT"
      headerDelay=1 dataDelay=1>
   <Input  id=1  type="X+">
   <Input  id=2  type="X-">
   <Input  id=3  type="Y+">
   <Input  id=4  type="Y-">
   <Input  id=5  type="Node">
   <Output id=1  type="X+">
   <Output id=2  type="X-">
   <Output id=3  type="Y+">
   <Output id=4  type="Y-">
   <Output id=5  type="Node">
</Crossbar>

<Connection id="C01" source="INJ" destiny="BUF5">
<Connection id="C02" source="CROSSBAR.5" destiny="CONS">
<Connection id="C03" source="BUF1" destiny="RTG1">
<Connection id="C04" source="BUF2" destiny="RTG2">
<Connection id="C05" source="BUF3" destiny="RTG3">
<Connection id="C06" source="BUF4" destiny="RTG4">
<Connection id="C07" source="BUF5" destiny="RTG5">
<Connection id="C08" source="RTG1" destiny="CROSSBAR.1">
<Connection id="C09" source="RTG2" destiny="CROSSBAR.2">
<Connection id="C10" source="RTG3" destiny="CROSSBAR.3">
<Connection id="C11" source="RTG4" destiny="CROSSBAR.4">
<Connection id="C12" source="RTG5" destiny="CROSSBAR.5">

<Input  id="1" type="X+"   wrapper="BUF1">
<Input  id="2" type="X-"   wrapper="BUF2">
<Input  id="3" type="Y+"   wrapper="BUF3">
<Input  id="4" type="Y-"   wrapper="BUF4">
<Output id="1" type="X+"   wrapper="CROSSBAR.1">
```

```
   <Output id="2" type="X-"   wrapper="CROSSBAR.2">
   <Output id="3" type="Y+"   wrapper="CROSSBAR.3">
   <Output id="4" type="Y-"   wrapper="CROSSBAR.4">
</Router>
```

In the `Router.sgm` file there can be any number of routers, each defined by a ROUTER tag. In the ROUTER tag, there are various attributes defining the name of the router, the number of inputs and outputs, the size of the buffers, the buffer control algorithm and the routing function. The complete definition of the internal structure of the router has to be between the ROUTER and the corresponding /ROUTER tag.

B.3 Running Simulations

Having the definition of the router, network and simulation, every thing is ready to execute the simulation. This is done with the `ATCSimul` program. It accepts various command line switches, all are optional except for one, *-s <simulation_ID>*. It tells `ATCSimul` which simulation must be executed. The rest of the switches override the simulation parameters set in the SIMULATION tag like *-l <load>* sets the traffic load and *-c <cycles>* modifies the number of cycles to simulate. There are also some switches that enable tracing and debugging functions.

In order to run the simulation shown in the examples above, the following command should typed:

```
$ ATCSimul -s TORUS8x8-CT
```

The output of the simulation is fairly simple. It consists of a listing of the statistical values observed in the simulation. By executing the command above, the following output is obtained.

```
***************************************
 Network        = Toro(8,8,1)
 buffer control = CT
 routing control= DOR-BU
 Started at : Tue May  8 08:16:31 2001
 Ended at   : Tue May  8 08:18:29 2001
***************************************
Traffic Pattern         = Perfect Shuffle
Seed                    = 113
Cycles simulated        = 20000
Buffers size            = 64
Messages length         = 1 packet(s)
```

```
Packets length          = 16 flits
Supply load             = 9.76125 %
Real load               = 9.73375 %
Supply Thr.             = 6.2472 flits/cycle
Throughput              = 6.2296 flits/cycle
Average distance        = 4.13395
Messages generated      = 7809
Messages received       = 7787
Messages to inject      = 8
Total message latency   = 42.6705
Network message latency = 40.8305
Buffer message latency  = 1.83999
Maximum message latency = 159
Simulation time         = 00:01:58
```

The header presents information of the network and router that were simulated. Following are the simulation parameters and then the simulation results, the meanings of which are explained below:

`Supply load` the traffic load presented to the network. This value is nomalised to the network bisection.

`Real load` the traffic load actualy transmited by the network. This value is normalized to the network bisection.

`Supply Thr.` a different view of the load presented to the network, in terms of injected flits per simulated cycle.

`Throughput` the number of flits the network consumed per simulated cycle.

`Average distance` the average distance traveled by the messages.

`Messages generated` the amount of generated messages.

`Messages received` the amount of received messages.

`Messages to inject` the amount of messages that are still to be injected.

`Total message latency` The mean total latency. This is, since the message is generated until it is consumed.

`Network message latency` The mean network latency. This is, since the messages is injected in the network until it is consumed.

`Buffer message latency` The mean time the messages spend waiting to be injected in the network.

`Maximum message latency` the maximum total latency registered in the simulation.

`Simulation time` The time needed to complete the simulation.

B.4 Helper Applications

The SICOSYS work environment includes three applications that are very useful to detect and locate malfunctions. The two first are concerned with the buffer usage within the routers. The third is an interface to SICOSYS that presents a graphical view of the evolution of the simulation while its running.

B.4.1 Examinig Buffers

When designing a router it is very interesting to know how full do the routers get in order to optimize their size. SICOSYS can write a file containing samples of the buffer ocupation, this is done by specifying a filename after the parameter *-b*. The samples are taken every 100 cycles. Because the size of this file can get very big, it is written in a binary format.

There are two tools to inspect the buffer ocupation file. `BView` is a very simple program that calculates the average ocupation for each buffer and presents it in a table. For example, running random traffic in a 4 node torus network with bubble routers gives an the following output.

```
Network size      = (2,2,1)
Number of buffers = 5
Buffer Size       = 32

Mean buffer ocupation
 (0,0,0)   (1,0,0)   (0,1,0)   (1,1,0)
BUF5  size=32
   0.945      1.17      1.15      1.38
BUF4  size=32
   0.525      0.62      0.55      0.52
BUF3  size=32
   0.485      0.66     0.445      0.45
BUF2  size=32
   0.625      0.39     0.695      0.32
BUF1  size=32
   0.355      0.38       0.5     0.605
```

The other tool is graphical tool designed using Trolltech's QT GUI application framework. The tool gives a graphical view of the buffer ocupation. `xbuffer` loads the buffer file and does an animation of the ocupation of a certain buffer of each router. The View→Select Buffers option presents a lists of the buffers in a router. The user can choose which buffers will be represented. By pressing the Start button the program opens a window for each selected buffer. The windows are divided in cells, one for each router in the network, the colour of the cells vary from white to black depending on the ocupation of the buffer in that router. The figure B.1 shows a screenshot of a 64 node torus.

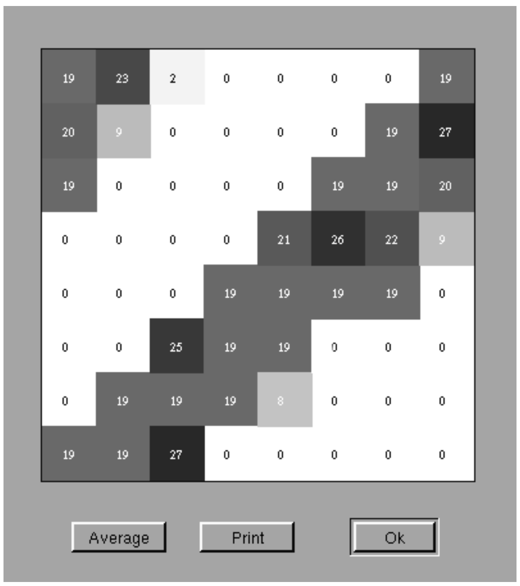

Figure B.1: Screenshot of `xbuffer`

B.4.2 Observing Latency Evolution

Another interesting point in the performace analisys of routers and networks is to know the evolution of the latency. `xsimul` is a simple front-end to SICOSYS, also based on Trolltech's QT GUI application framework. With then File→Open option the user can select a simulation from the `Simula.sgm` file. To further configure the simulation, the Options→Parameters option enables the user to set simulation parameters, such as traffic pattern or simulation lenght (See figure B.2). And by clicking the Run button the simulation starts. During the simulation the latency is plotted versus the simulation cycles in real time, enabling the user to ver the evolution of the network easily. Figure B.3 shows a screenshot of `xsimul`.

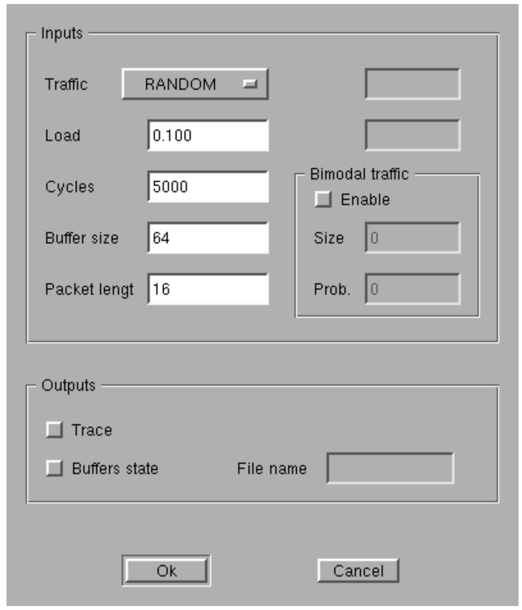

Figure B.2: Screenshot of Options→Parameters dilog box

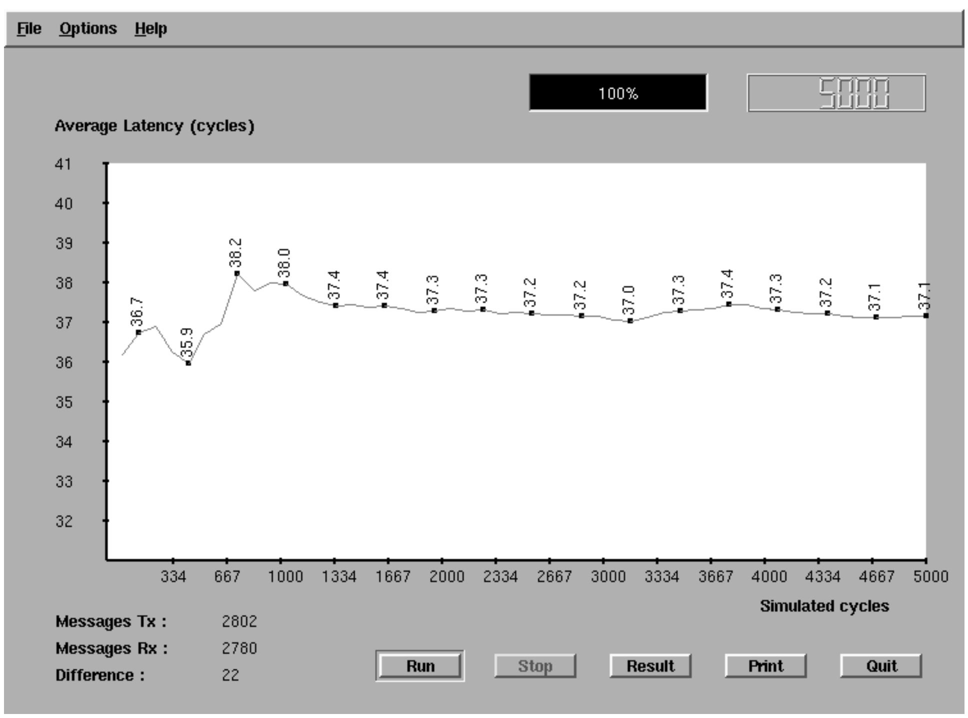

Figure B.3: Screenshot of `xsimul`