A NETWORK SIMULATOR BASED ON OSPF ROUTING PROTOCOL FOR NETWORK DESIGN AND EDUCATIONAL PURPOSES

By

Jorge L. Justiniano Magraner

A Project Report submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING
In
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGUEZ CAMPUS
2007

Approved by:

____________________________ ______________________
Yi Qian, PhD Date
Member, Graduate Committee

____________________________ ______________________
Nayda G. Santiago Santiago, PhD Date
Member, Graduate Committee

____________________________ ______________________
Isidoro Couvertier, PhD Date
President, Graduate Committee

____________________________ ______________________
Jeannette Santos, PhD Date
Representative of Graduate Studies

____________________________ ______________________
Isidoro Couvertier, PhD Date
Chairperson of the Department

Abstract

Computer simulation is the tool of choice for developing and testing new technologies and techniques. It is also commonly used to help teach and train students and other professionals. The Masters project done is a network simulation program whose purpose is to develop and test current networking protocols and which will also help when teaching of people who are beginning to learn how to set up and configure computer networks with the aid of graphical computer interface (GUI). The implemented routing protocol used is Open Shortest Path First (OSPF) which is the most commonly used interior gateway protocol (IGP), but the network simulator could be updated and its functionality expanded by the addition of other network layer protocols and any other protocols of the other layers of the Open Systems Interconnection (OSI) Reference Model upon their implementation.

Resumen

Los simuladores de computadoras son la herramienta preferida para desarrollar y probar nuevas tecnologías y técnicas. También son comúnmente usadas para enseñar y entrenar a estudiantes y otros profesionales. En este proyecto de maestría se hizo un programa que simula de redes de computadoras cuyo propósito es desarrollar y probar protocolos de redes actuales y también ayudar a personas que están empezando aprender a montar y configurar redes de computadoras con la ayudar de una interfase gráfica. El protocolo de enrutamiento simulado es “open shortest path first” (OSPF) cual es el protocolo interior más utilizado, pero este simulador de redes permite ser actualizado y su funcionalidad ser extendida con la adición de otros protocolos del nivel de redes y otros protocolos de otros niveles del modelo de referencia del sistema abierto de interconexiones una vez hayan sido implementados.

Table of Contents

Figure List

1 Introduction

Computer simulation is an excellent tool for studying the behavior of complex systems in our every day life. Inside the computer world, one of the areas where simulations are being heavily used is in the area of computer networking. A network simulator is a valuable tool for researchers to develop, test, and diagnose network protocols. Simulators are economical and practical because any user can carry out experiments without having to buy and set up all the hardware needed. Even if the network being set up is simple and small in size, (e.g. two personal computers, a router, and a server) a big investment would be needed to be made by the user and could continue to grow if other equipment were needed or if the current equipment needs to be repaired or replaced. Another advantage of network simulation is that it provides the user the flexibility of changing different variables in the network with ease. An example of this is if the user wants to simulate a link, they can set up that link up with any amount of bandwidth or propagation delay, which will simulate the use of different types of wire and connections.

Both commercial and academic network simulation software are available for researchers and students to use. Some examples of these simulators are OPNET (3D Visualizer, IPv6 Discrete Event Simulation, Specialized Model) [7], LANforge [1], and SIMULINK [3], which are commercial network simulators. Akaroa2 [10] and Network Simulator 2 (NS- 2) [2] are of academic nature and are available free of charge for any user to use or modify. All these programs simulate real network situations but only a few of them use graphical representation to visually show how the packets are delivered from the source through the different parts of the network to their final destination. Examples of visual simulators are OPNET and SIMULINK.

The purpose of this master's project is to create a network simulator that will have a graphical interface where the user can set up their own network with any hardware components that they want (PC, servers, routers, switches, and different cable types) graphically configure the equipment and then the user can visually see how the packets are sent and routed through the network using the Open Shortest Path First (OSPF) v2 [6] routing protocol which is the most used Interior Gateway Protocol (IGP) and is supported by most vendors [9]. With visual feedback the user will be able to see where packets are lost due to bad configuration, possible collisions and the delivery of correctly received packets. Once the user is satisfied with the configuration they will be able to extract the command line configuration instructions from any of the simulator's routers and use it to configure real routers. The simulator will also have the ability to expand its functionality as other routing protocols can be added upon future development or any other protocol that are part Open System Interconnection (OSI) Reference Model.

In general, this network simulator will be used to introduce students or users that have zero or little networking experience the ability to learn and apply how to set up a small computer network, configure the different hardware components and see how the packets are routed when using the OSPF routing protocol.

2 Literature Review

Network simulators are the tools of choice for researchers that create, develop, and administer current computer networks. Before testing any new protocols or new hardware on existing networks, researchers and developers prefer to simulate them on virtual networks. There are many reasons why they prefer simulations on virtual networks to testing on real networks. The first reason is that network simulators are economical. Besides the cost of creating the simulator, researchers are able to test new technologies or new algorithms without having to actually build or acquire any new hardware. This also includes all the wiring needed like: fiber optics, Ethernet, and copper cables. Simulations also have the advantage of having results that are easily reproducible compared to real networks since virtual networks have the advantage of being free of uncontrollable factors, which can affect the results of the simulation. One final advantage of simulation is that the network topology and parameters can be changed with ease [11].

Although network simulators have many advantages, they also come with many disadvantages. As mentioned before simulators remove uncontrollable factors like external traffic or unforeseen errors, which could be inserted into simulated network to try to simulate real situations but usually removed for simplicity and easier analysis. This simplification translates into not having results equally to those of using real network equipment. Another disadvantage is memory constraints. Simulating a large network with many nodes can take up a lot of the system resources on the computer running the simulator.

There are several projects being done to try to fix the problems that network simulators have. One of these projects was developed by Wang and Kung [11] where they have proposed a way of making a Hi- Fidelity TCP/IP network simulator. In their studies they have found that network simulators have the limitation of providing results “that are not convincing as those produced by real hardware.” This is because much of the complexity that is found in protocols is taken away or simplified. Another limitation that they found is that network simulators are not extensible to include Application Programming Interface (API) that is a set of protocols or routines that developer provides a user in order to develop software applications. By adding an API to a network simulator, applications can be developed and run on simulated nodes to test the networks response time and capabilities. By not having an API, the effect of application level performance of a real system cannot be studied.

To create their Hi-Fidelity network, Wang and Kung proposed to use a Berkley Socket Distribution (BSD) UNIX Tunnel Network Interface, which is a network interface with no real network attached to it. The network was be done by modifying the UNIX Kernel so multiple nodes were simulated by reentering the simulation host multiple times creating a network of BSD UNIX host and routers. The advantage of this approach was that they use of real TCP/IP protocol stack and made use of BSD UNIX systems API making the simulator to be extensible to current or future programs that can run on any node in the simulated network. A disadvantage to this approach is that the simulator was more than three times slower than the real network. This means as more nodes are added to the network, more time was needed to complete a simulation.

Another simulator that is being developed is Nessi created by Jérôme Vernez, Jürgen Ehrensberger, and Stephan Robert [4]. This simulator (developed using the python computer language) was developed towards educational use as it enables students to implement and modify simulation models of protocols. Nessi compared to NS-2 simulator has a lower learning curve and the development of protocols on it is easier and less error prone but at the same time Nessi has the disadvantage of having a much slower simulation running time. Another advantage that Nessi offers is that it uses a graphical interface to control its simulations, observe behavior and produce graphical results.

Li and Depei [13] proposed to construct a parallel network simulator, which will be based on the sequential simulator NS-2. The purpose of creating the parallel network is to resolve the disadvantage simulation time and memory requirements. To accomplish parallelism, parallel discrete event simulation (PDES) techniques are applied to modify the event scheduling mechanism of NS. The reason for making a parallel simulator is, as more design complexity and the size of the simulation network increases so will the time needed to complete the simulation. By using PDES techniques simulation time is cut by dividing large simulation model parts in sub models, which were executed in parallel.

3 Background Information

Before the programming of the project can begin, the first step that needs to taken is to research and thoroughly understand all the basics of computer networks from simple configuration to the functionality of each of the network’s components that will be used in the simulations. The hardware components that will be used in the simulator are hubs, switches, routers, PCs, servers and interconnection mediums that include modem, Ethernet, fiber optic and serial.

The hub is a component that interconnects all of the components that are attaches to its ports together. When one of its connected components sends out a frame from a port that is connected to the hub, the hub receives the frame and sends it out all of its ports (except the port which the frame was received on) letting all the other components receive the frame. Then each receiving component will look at the destination Medium Access Control (MAC) address to see if the frame was intended for it. The downside of using hubs is that if two or more of the components connected to the hub send out information at the same time collisions will occur.

A switch usually works with the data link or Layer 2 of the OSI reference model and its functionality is much like that of a hub in which the switch interconnects all of the components that are attaches to it but with the difference that it only sends out the received frame out of all if its ports if the either the frame is sent with a broadcast address or if the destination is unknown by the switch. For a switch to know about the frame’s destination and know how to forward it, a switches use transparent switching to learn of other locations on the network and fill out its lookup table.

The algorithm used for transparent bridging consists of five parts. The first part of the algorithm is a learning stage where the switch extracts the source MAC address from the received frame and if it is not found in the lookup table, the address is added allowing the switch to forward a packet intended for component that sent the frame. Next if the switch does not know the destination of the packet then the switch sends the frame out of all its ports except for the port that the last frame was received from. This stage is called the flooding stage. When the switch receives a frame that it knows the destination, the switch is in the forwarding stage and sends the frame out of the appropriate port. When a switch detects that the frame received belongs to the same segment or network part, then the frame does not need to be forwarded. This stage is known as the filtering stage. The final stage for transparent bridging is the aging stage where the switch places a time stamp on the lookup table entry that was just used. When a lookup table entry reaches a certain user defined time, the switch erases the entry.

In order to reduce the complexity of a network, an administrator can choose to create virtual local area networks (VLANs) within the network switches. Theses VLANs consist of a group of network nodes that group together in a single broadcast domain [12]. The broadcast domain consists of a portion of a network that will receive broadcast packets from any component within the defined network. If frame needs to be sent from on VLAN to another a router needs to be used between to route the information from on VLAN to another since VLANs cannot communicate between themselves.

Overall switches provide more advantages than the use of hubs. The first advantage is that with a switch the users can use the complete bandwidth of a line and if a hub is used the bandwidth of a line in shared. Another advantage is that attached components can send out frames at the same time with none or any collisions and using a hub there is a greater chance of having collisions and the probability grows as the amount of components connected to the hub grows. Another advantage of switches is that they can be used to divide the network into segments reducing the complexity and traffic.

PCs and servers are the end user devices found at the end of a network. Most of the user data originates from PCs and goes through the network and reaches another PC or server. For the simulator, both the PC and server will only have one port. The configuration of the PC port includes the type of medium, the IP address for the port, the network submask, the bandwidth and duplex type of the medium. Another parameter that's needs to be entered is the default gateway address that is the address of the router that servers as access point to another network. Instead of a user having to enter the IP address, submask and default gateway manually, the PC can use dynamic host configuration protocol (DHCP) which allows the PC to request an IP address from a DCHP server. For a server in the simulator the only configuration needed is that of the port type, IP address, submask, bandwidth, and duplex type.

A router is a network device that forwards data packets across a network towards its destination. Routers work with the network layer or Layer 3 of the OSI reference model meaning the packet is extracted from data payload of the received frame. In order to forward a packet toward its destination, the router makes use of routing protocols. There are to types of routing methods: static and dynamic. In static routing, the administrator manually configures routes. Meanwhile with dynamic routing, the routing protocol chosen by the administrator is in charge of discovering paths and destinations. In order for dynamic routing to be successful the routing protocol must maintain the routing table and must distribute information in a timely fashion to other router.

There are two types of dynamic routing algorithms: distance vector and link state routing protocols. With distance vector, the router passes the complete routing table to the neighboring router in which it combines it with its own routing table. The distance vector is based on Bellman-Ford algorithm and uses hop count as metric. Examples of distance vector routing algorithms are: Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP). The other dynamic routing algorithm is link state algorithm were each router maintains knowledge of the whole network topology. Each router floods Link State Advertisements (LSA) to the neighboring routers to inform them of the surrounding network. The router then makes use of the LSA to create a Shortest Path Tree (SPT) for the network and its destinations. Examples of link state routing protocols are: Interconnection System to Interconnection System (IS-IS) and OSPF.

Overall distance vector algorithm is easier to configure and debug but has the limitation of network size. Meanwhile link state algorithm work better for lager networks and converges quicker than distance vector algorithms but has the disadvantage that the algorithms are more complex, require more memory, more computations and more bandwidth requirements.

When a person looks at a network like the Internet, it is composed of a collection of other networks. Each of the networks that are under a common administrator is known as an Autonomous System (AS). Within each AS then administrator chooses a common routing protocol known as an Interior Gateway Protocol (IGP). An example of an IGP is OSPF. To communicate with other AS, router contains another routing protocol known as Exterior Gateway Protocol (EGP). An example of EGP is Border Gateway Protocol (BGP).

OSPF is the most used interior gateway protocol. For an AS running OSPF, the AS can be subdivided even more by creating areas. Each area runs a copy the routing algorithm and the area’s topology is invisible to router outside the area. There are two special areas: backbone and stub areas. The backbone area, also known as area 0, is an area that interconnects all the areas in the AS. The stub area is an area that doesn’t accept external routes through it thus reducing the link state database size.

Each router runs the protocol a different way depending on the type of router it is chosen to be. A router can either be set to be of type internal, area boarder, backbone, or boundary router. An internal router is one that lies inside an area. An area boarder router is one that is part of two or more areas. A backbone router is one that has an interface to the backbone area. The last router type is boundary router that is router that exchanges information with other AS and it’s in charge of advertising external routes to the other routers in the AS.

As mentioned before, OSPF is a link state routing algorithm that floods LSA in order to learn about the surrounding network. In OSPF v2 there are four type of LSA: router, network, summary type 3, summary type 4, and AS external LSA. Router LSA’s are created by all the routers in the AS and describe the state of each of the router’s interfaces which is a connection between the router and on of its attached networks. Network LSAs are created by the designated router of an area and contains a list of all the routers that are in the area and adjacent to the designated router. The designated router is chosen for networks that are of type broadcast or non-broadcast multi access networks

(NBMA). LSAs of type summary 3 are created by area boarder routers and describe the destinations found in an area. This LSA is then flooded to the next area.
Summary LSA of type 4 are also created by designated router and describe routes to AS boundary routers. The final LSA is AS External LSA that describes routes to destinations outside the AS. This type of LSA is flooded throughout the entire AS except for areas designated as a stub area.

There are four types of protocol packets that are sent by routers running OSPF. Hello packets are sent by the router to help discover and maintain relationships between routers. When a relationship is established, the router starts to send database description packets which describe the contents of the link state database. Once the database exchange has been completed, the router may wish to ask for either LSAs that are not contained inside its database or for LSAs that are more up to date from those found in its database. To accomplish this, the router sends out Link State Request packet. When sending new or more recent LSAs, routers send out Link State Update Packets. The last protocol packet type is the Acknowledgement Packet that is used to make the flooding procedure reliable.

4 Materials and Methods

The development of this project was done using the Java™ programming language which allowed the created network simulator to run on any platform operating system as long as the user has the Java Virtual Machine installed on their system. The code development of the project has been accomplished with the use of Sun Microsystems NetBeans 3.5.1 IDE (integrated development environment) Java developer [5]. NetBeans was used because of the facility and speed of creating the program's GUI (graphical user interface). Along with the ability of quick GUI development, NetBeans also provided other programming tools for the fast development of the Java classes and interfaces along with an excellent code debugger that was used to find and correct the programming errors.

After all the knowledge gathering about networks, its topologies, and its hardware components, the next step of the program's development was to identify most the classes that were needed and what interfaces were necessary to join those object classes with the graphical part of the program. These initial classes are the ones that represented all the hardware components like: PCs, switches, routers, servers, hubs, ports, and different types of cables. Next group of classes that were identified were those corresponding to OSPF. The classes that remained were those of that provided utility.

The second part of programming the project was the development of the simulators GUI's. For the network setup, as seen in figure 1 below, the GUI was set that the user can select any hardware component from the top toolbar drag and drop the component that they want on to the working area and then the user will be able to inter connect them in any network topology that they wish after clicking on the connect button.

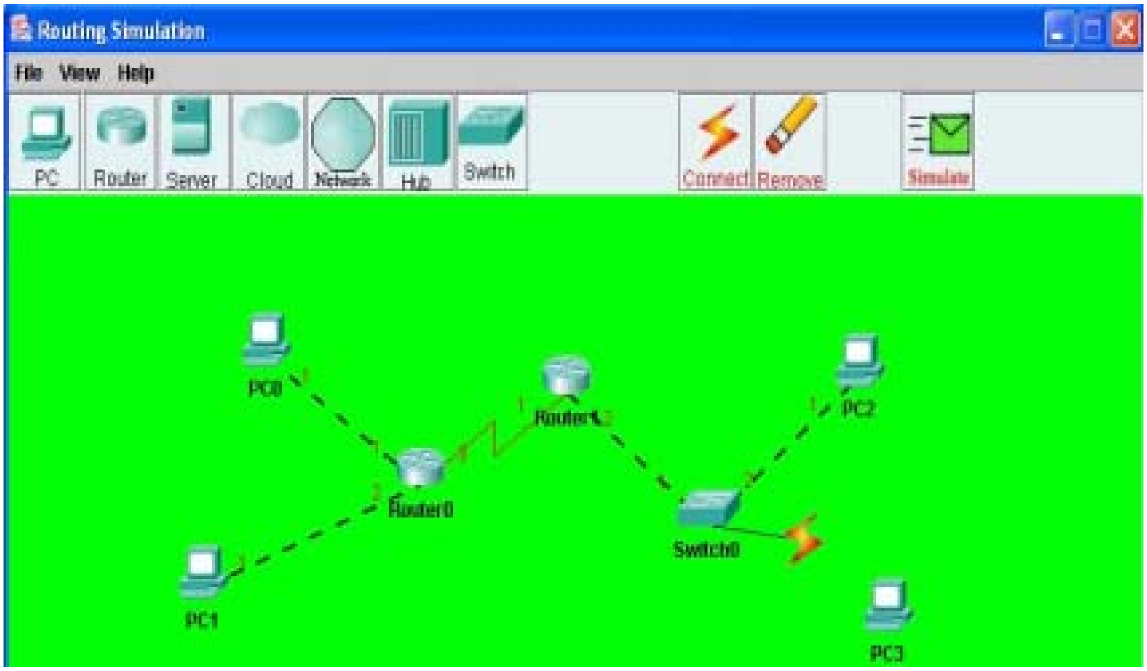
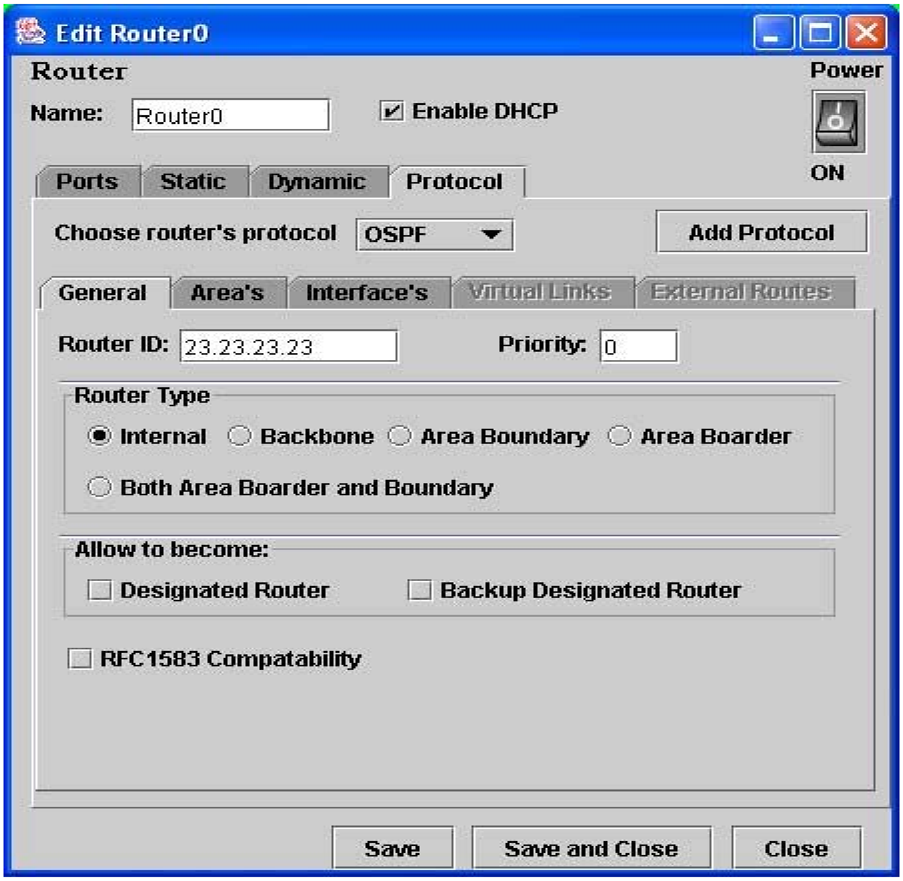


Figure 1: Network Setup Window

When the user double clicks on a piece of hardware already placed in the working area an event triggers a window to appear which will allow the user to view, edit, and save the hardware's configuration settings for the chosen device. Figure 2 below show the window that appears to setup any router.



Name: Router0

☑ Enable DHCP

Choose router's protocol: OSPF

Router ID: 23.23.23.23

Priority: 0

Router Type

● Internal ○ Backbone ○ Area Boundary ○ Area Boarder

○ Both Area Boarder and Boundary

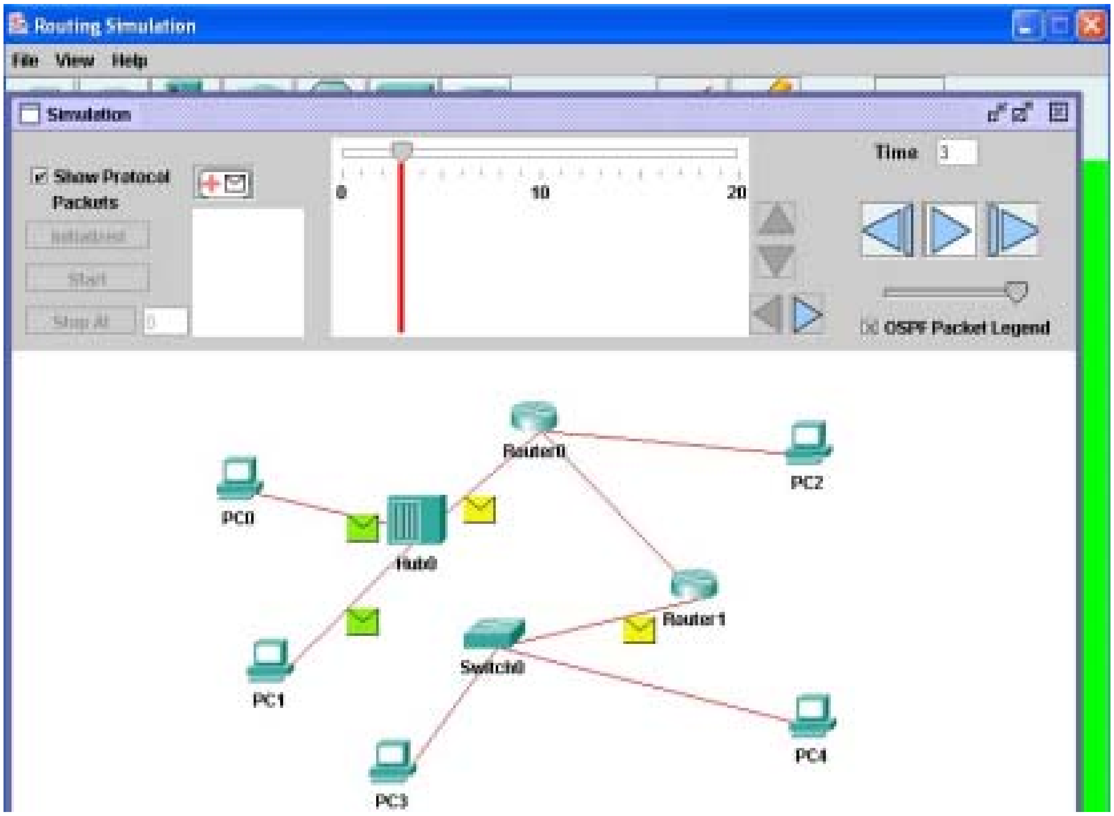
Allow to become:

☐ Designated Router ☐ Backup Designated Router

☐ RFC1583 Compatability

Figure 2: Router Configuration Window

Once the user finishes the network setup and configuration, the program is then passed to the network's simulation GUI. This part of the program relies on a lot of drawing and heavy animation. The timeline line, data and protocol packets have been drawn with the use of Java's Graphics2D class and other graphical libraries to draw the two-dimensional images while the Thread class was used to control the start and end of the animations. As seen in figure 3, the screen shot shows four packets that are sent at time 3 in the simulation which are drawn from its source and redrawn several points further over the line that connects the two components until the packet is over the destination causing the thread that draws a particular packet to stop.


☑ Show Protocol Packets

Time 3

☒ OSPF Packet Legend

Figure 3: Simulation Window while running the simulation

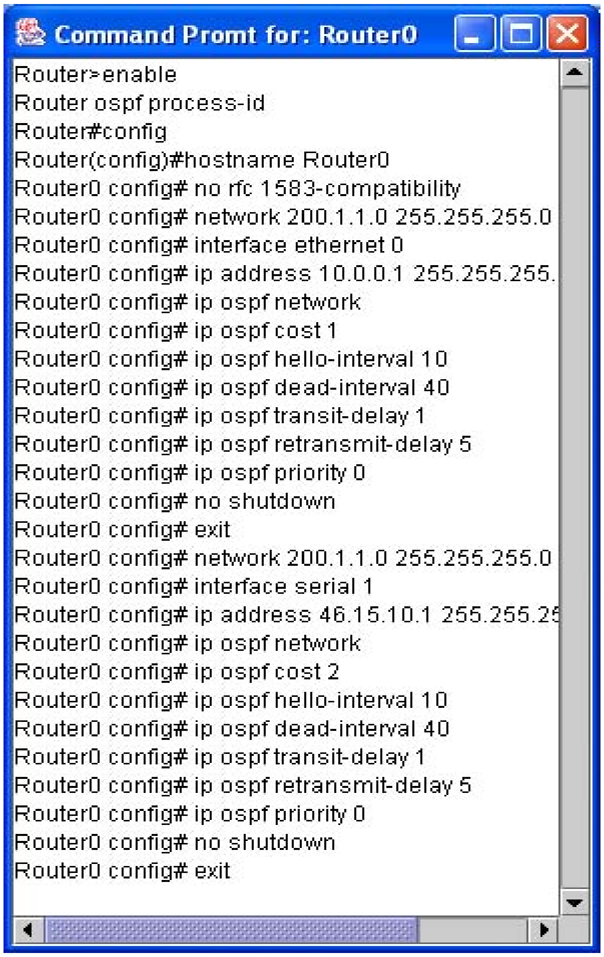
The last stage of the programming of the simulator was to debug program as a whole. Although most of the protocol functionality was tested during the early stages of the simulator's development through the command prompt, extensive testing was performed during this final stage because most of the errors usually occur during integration of all the parts. To test the functionality of the OSPF protocol, different examples of networks using OSPF were taken from OSPF RFC and from Cisco tutorials on OSPF. These examples were placed in the simulator and the resulting routing table was compared to the one in the examples. If the same amount of destinations appeared in both routing tables and the destinations match in all the routing table parameters then network was taken as working and correctly configured. Also the amount of LSAs sent by a router was checked in the same manner as the routing table.

The final stage of the project was to help improve human computer interaction and also to lower the learning curve of the simulator program.

4 Discussion

An application was developed were a user can create a computer network with any topology, with different types of wiring cables and with several of the most used hardware components found in today’s computer networks. The only limitation of the network size is given the amount of area found in the working panel, which is the area the user has to place all the network devices. For that reason, the network simulator has a limit of thirty devices. Another reason for the limit on the amount of devices that can be used is that during the simulation the network a lot of computations are made and the drawing of packets of the simulation takes up many of the users system’s resources. Each router, which is the hardware device in the simulator with the largest number variables and functions, has an initial size of 5904 bytes just for its variables. During the simulation this size of the amount of variables increases plus the amount of functions used also increase that can cause the system to decrease its speed and performance.

The configuration of each of the hardware components is done through a GUI allowing for faster configuration of the device. For router, the user can also extract the equivalent command line instructions (as seen in figure 4 below) allowing the user to copy the instructions and use them to configure real routers.



```
Router>enable
Router ospf process-id
Router#config
Router(config)#hostname Router0
Router0 config# no rfc 1583-compatibility
Router0 config# network 200.1.1.0 255.255.255.0
Router0 config# interface ethernet 0
Router0 config# ip address 10.0.0.1 255.255.255.
Router0 config# ip ospf network
Router0 config# ip ospf cost 1
Router0 config# ip ospf hello-interval 10
Router0 config# ip ospf dead-interval 40
Router0 config# ip ospf transit-delay 1
Router0 config# ip ospf retransmit-delay 5
Router0 config# ip ospf priority 0
Router0 config# no shutdown
Router0 config# exit
Router0 config# network 200.1.1.0 255.255.255.0
Router0 config# interface serial 1
Router0 config# ip address 46.15.10.1 255.255.25
Router0 config# ip ospf network
Router0 config# ip ospf cost 2
Router0 config# ip ospf hello-interval 10
Router0 config# ip ospf dead-interval 40
Router0 config# ip ospf transit-delay 1
Router0 config# ip ospf retransmit-delay 5
Router0 config# ip ospf priority 0
Router0 config# no shutdown
Router0 config# exit
```

Figure 4: Command line configuration for Router 0
after setup through GUI

After the network has been connected and configured, the user can then simulate the network by going into the simulation window. There the user has the options to either start the simulation with the network already initialized where all the routers having their routing tables filled out. The next simulation mode the user can choose is to have the simulation start out from scratch allowing the user see all the protocol packets (if the user has this option selected) that each router must send out in order to obtain knowledge about the surrounding network and help it build its routing table for the autonomous system. The last possible simulation mode is one were the user can set a stop time and the simulation runs continuously until reaching the user defined time. From that point on the user continues the simulation manually by advancing the time through either the play button or skip button (the third button from left to right as seen in figure 5 below).

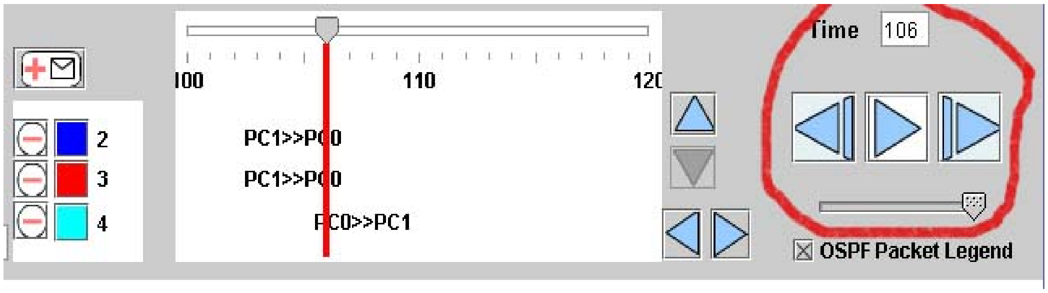


Figure 5: Timeline control buttons.

When the user hits either the play or skip button, the simulation will advance one time unit. In the simulator it is assumed that lines don’t have delays associated with them therefore allowing frames or packets to go from one device to another in just one time unit. Also all the computations done inside each hardware device take one time unit. The reason for this is to simplify the simulator and since performance measurements are out of the initial scope of the simulator that is of helping users learn to configure basic networks using OSPF routing protocol. Delays and its measurements could be implemented in another time to make the simulator more powerful allowing the user to measure performance and throughput of using different types of interconnecting mediums.

To add delays to the simulator, Line class of the simulation needs to be changed that when a frame or packet is received, it will calculate the delay depending on the type and bandwidth of the medium and the length of the medium between the two hardware devices. The PacketInfo class needs to be change as a variable needs to be added which will control the speed of the frame or packet during the visual part of the simulation. This way the user can visually see how packets move thru slower mediums.

For any routing protocol, the user can left click on any router at anytime during the simulation and a popup menu will appear (see figure 6) with the option of either looking at the routers configuration settings as seen in figure 7 below or the router's current routing table seen in figure 8.

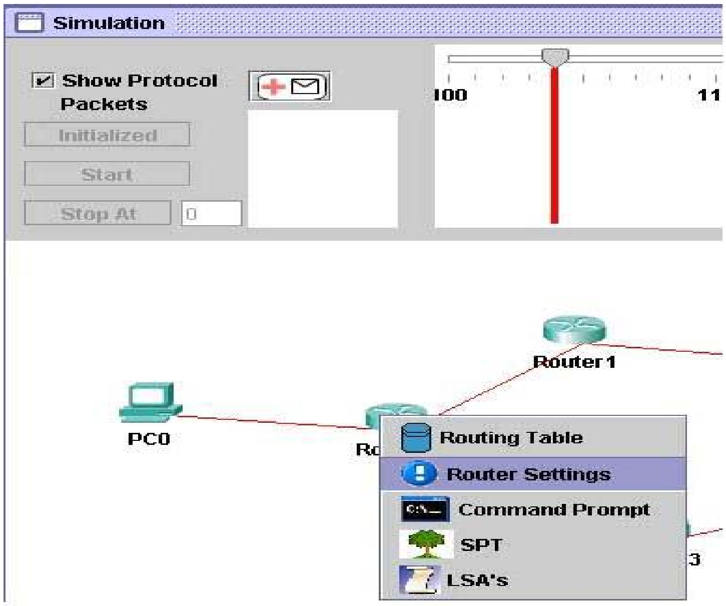
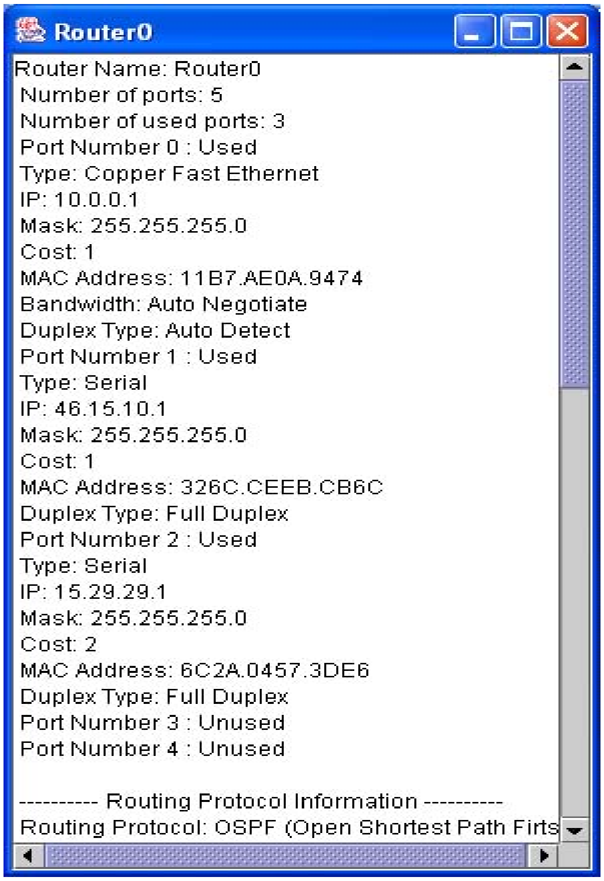


Figure 6: Drop down menu that appears when clicking on a router

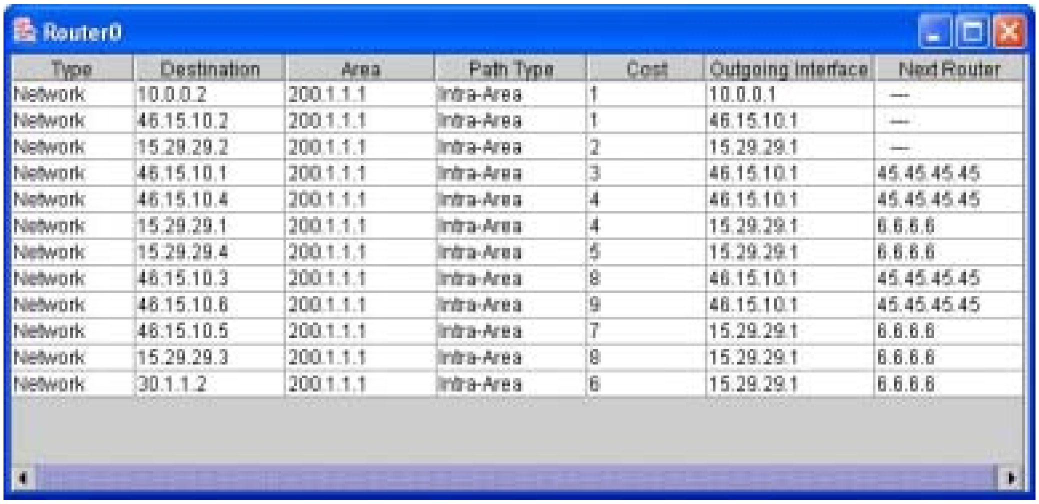


```
Router Name: Router0
 Number of ports: 5
 Number of used ports: 3
 Port Number 0 : Used
 Type: Copper Fast Ethernet
 IP: 10.0.0.1
 Mask: 255.255.255.0
 Cost: 1
 MAC Address: 11B7.AE0A.9474
 Bandwidth: Auto Negotiate
 Duplex Type: Auto Detect
 Port Number 1 : Used
 Type: Serial
 IP: 46.15.10.1
 Mask: 255.255.255.0
 Cost: 1
 MAC Address: 326C.CEEB.CB6C
 Duplex Type: Full Duplex
 Port Number 2 : Used
 Type: Serial
 IP: 15.29.29.1
 Mask: 255.255.255.0
 Cost: 2
 MAC Address: 6C2A.0457.3DE6
 Duplex Type: Full Duplex
 Port Number 3 : Unused
 Port Number 4 : Unused

 ---------- Routing Protocol Information ----------
 Routing Protocol: OSPF (Open Shortest Path Firts
```

Figure 7: Window showing a router’s general settings

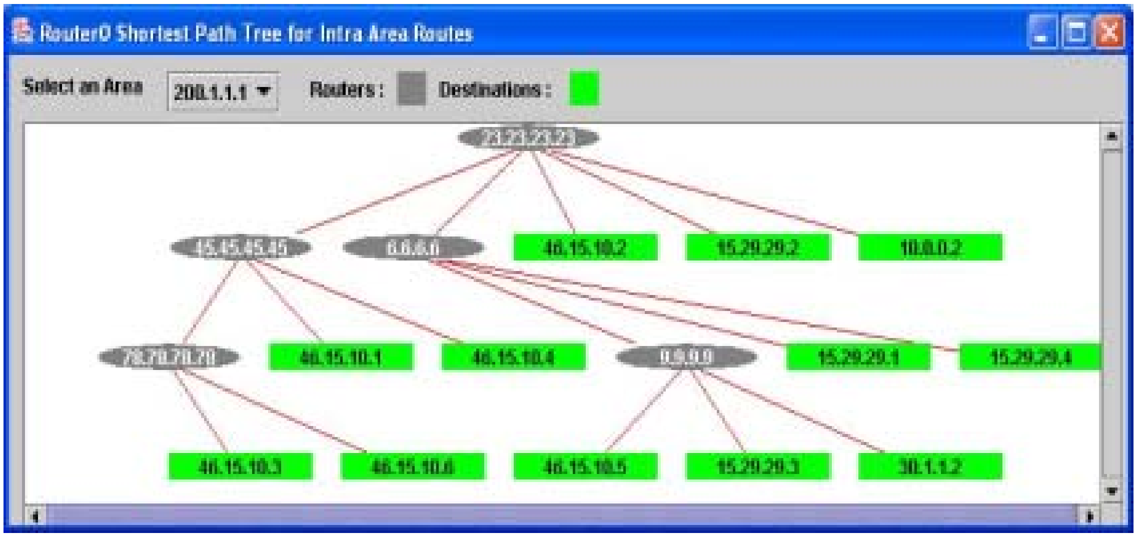
In figure there is an example routing table for a router running the OSPF routing protocol. From this routing table the user is able to see all the destinations the router is aware of, the type of destination, the area ID of where the destination autonomous system (AS), the type of path, the outgoing interface address and the cost of forwarding by this interface.



Type	Destination	Area	Path Type	Cost	Outgoing Interface	Next Router
Network	10.0.0.2	200.1.1.1	Intra-Area	1	10.0.0.1	---
Network	46.15.10.2	200.1.1.1	Intra-Area	1	46.15.10.1	---
Network	15.29.29.2	200.1.1.1	Intra-Area	2	15.29.29.1	---
Network	46.15.10.1	200.1.1.1	Intra-Area	3	46.15.10.1	45.45.45.45
Network	46.15.10.4	200.1.1.1	Intra-Area	4	46.15.10.1	45.45.45.45
Network	15.29.29.1	200.1.1.1	Intra-Area	4	15.29.29.1	6.6.6.6
Network	15.29.29.4	200.1.1.1	Intra-Area	5	15.29.29.1	6.6.6.6
Network	46.15.10.3	200.1.1.1	Intra-Area	8	46.15.10.1	45.45.45.45
Network	46.15.10.6	200.1.1.1	Intra-Area	9	46.15.10.1	45.45.45.45
Network	46.15.10.5	200.1.1.1	Intra-Area	7	15.29.29.1	6.6.6.6
Network	15.29.29.3	200.1.1.1	Intra-Area	8	15.29.29.1	6.6.6.6
Network	30.1.1.2	200.1.1.1	Intra-Area	6	15.29.29.1	6.6.6.6

Figure 8: Window showing routing table for
a router using the OSPF routing protocol

Other options will appear in the popup menu when the user using OSPF as the routing protocol. The user can look at any time the Link State Advertisements (LSA) that the router has generated and received. The user can also see the router's current shortest path tree for intra-area path for the chosen area (as seen in the figure 9 below). In this window all the routers for the area will appear as grey ovals while host interfaces and other interface appear as green rectangles which represent leaves.



Select an Area: 200.1.1.1

Figure 9: Window showing intra-area routes for Router 0

During or after the completion of the simulation, the user can chose to send out data packets for either PC's or servers to another PC or server. To accomplish this, the user only has to click on add packet button shown in figure 10 (button with plus sign and letter icon on it within red circle) then the simulator waits for the user to click first on the source and then click on the destination. These data packets are empty and are only used by the user to visually see that the network is working as their packets are sent from a source to their destination.

If the user would like to send some specific type of data like a file, a picture, or any other type data then the PC class and the Icon class have to be modified to allow the functionality of the user being able to select the data that he or she wishes to send out. Then the size of the data has to be taken into consideration. If the data is too big then it has to be broken up into smaller pieces where one piece will be sent out every time unit. Meanwhile on the destination side, a function can be implemented that shows what either the PC or server (which is the end devices of the network) has received till the moment. This functionality can actually be an application that runs on Layer 7 of the OSI reference model.

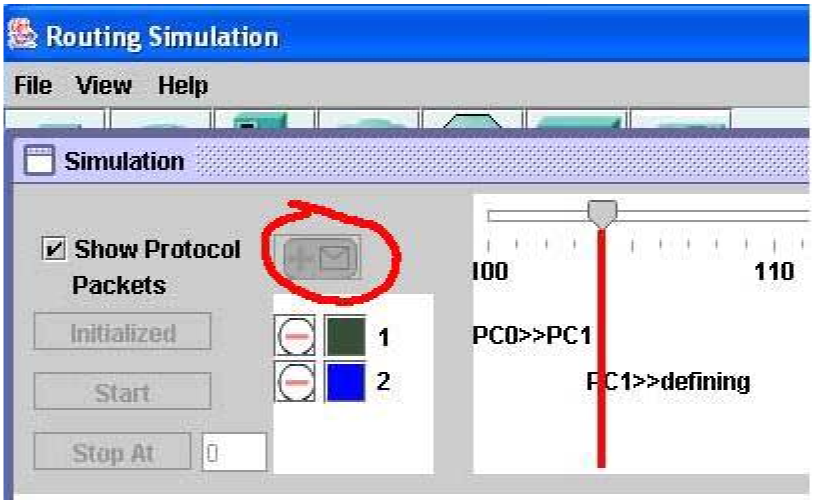


Figure 10: Add user data packet in the simulation window

When the user packet goes thru the network to a destination, the receiving component provides a visual response letting the user know if the packet that the component has received is intended for it. If the packet was correctly received, a green check mark will appear on the component otherwise a red “X” will appear over the component. In figure 11 the hub in the simulation receives the packet from PC0 and sends it out all its ports except from the one it was received on. Since the packet was intended for PC2, the green checkmark appear over it while the “X” appears over PC1 since the destination address cannot match.

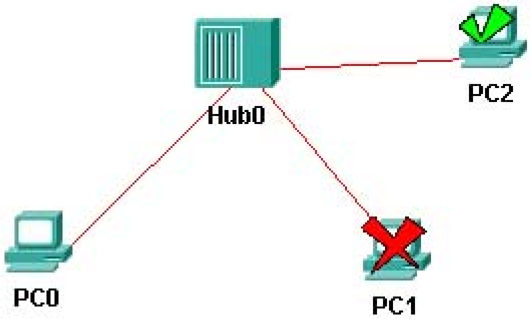


Figure 11: Visual response to received packet by PC2 and PC1

Another visual response provided by the simulator is that when collisions occur. When a hub receives two packets from different ports a collision occurs. To acknowledge this occurrence, a big yellow image simulating a crash appears over the hub as shown in figure 12. The collision image can also appear in the middle of a connection due to a collision of two components having port defined as half duplex and the ports are sending and receiving at the same time. This event also occurs when incompatible duplex types are defined between two connecting ports.

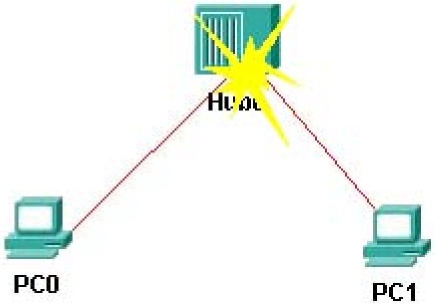


Figure 12: Representation of a packet collision occurring

5 Conclusion

The objectives of this project were completed. A graphical network simulator was developed that allows users to create a network with any topology, with different types of wiring cables and with several of the most used hardware components found in today's computer networks. The configuration of each of the hardware components is done through different GUI's.

After the network has been connected and configured, the user can then simulate the network by going into the simulation window. There the user has three options on how to run the simulation. The first is with the network already initialized with all the routers having their routing tables completed. The second simulation mode allows the user to begin with all of the components their start-up mode allowing the user to see all the protocol packets that each router must send in order to obtain knowledge about the surrounding network and help it build its routing table for the autonomous system. The final simulation mode will let the user set a stopping time for the simulation. This feature is mainly intended for debug and testing of future protocols.

During the simulation the user can click on any router and look at its configuration settings and its current routing table. The user also has an option which allows the user to see equivalent command line instructions which can be used to configure real routers in a faster way as the user only has to copy and paste the instructions on the router. When using OSPF as the routing protocol for the AS the user can also look at the LSA's that it has generated and received and that router's view of the current shortest path tree for the network.

The network simulator also has the ability to be upgraded by adding other routing protocols or other higher or lower layer protocols that belong on the OSI model.

5 Bibliography

[1] Candelatech, “LANforge”, *http://www.candelatech.com/*, (08/23/2005).

[2] Fall, Kevin and Varadhan, Kannan. “The Network Simulator - ns2: Manual,” *http://www.isi.edu/nsnani/ns/nsdocumentatioii.htm1* (28/02/02).

[3] Mathworks, “SimulLink,” *http://www. mathworks.com,* (28/02/02).

[4] Vernez,J.; Ehrensberger, J.; and Robert, S.; “Nessi: a python network simulator for fast protocol development”, Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, 2006 11th International Workshop on
8-9 June 2006, pages 67 – 71.

[5] Sun Micro Systems, “The Official Netbeans Homepage,” http:/*/www.netbeans.org,* (08/20/2005).

[6] Moy, J. “OSPF v2 Standards Track,” http:/*/www.cisco.com,* (08/20/2005).

[7] OPNET Technologies, Inc. “OPNET products home page”, *http://www.opnet.com/products/home.html,* (02/28/2002).

[8] “Synopsys, CoWare Frontier Design.” System C User’s Guide, version 1.2, 2001

[9] Tanenbaum, Andrew. *Computer Networks*. 3 ed.(New Jersey, Prentice-Hall Inc, 1996).

[10] University of Canterbury, New Zealand, “Project Akaroa home page,” *http://www* **.***cosc.canterburv.ac.nz/research/RGhet sim/siniulation group/akaroa,* (08/02/2002).

[11] Wang, S.Y., and Kung, H.T., “A simple methodology for constructing extensible and high-fidelity TCP/IP network simulators”, INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE vol 3, 21-25 March 1999 pages 1134 – 1143.

[12] Cisco, “How LAN Switches Work”, *http://www.cisco.com/en/US/tech/tk389/tk390/technologies_tech_note09186a00800a7af3.shtml,* (Jun 11, 2007)

[13] Li, Jue., and Qian, Depei. “A Practical Approach for Constructing a Parallel Network Simulator”, Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference. 27-29 Aug. 2003, pages 655 - 659

Appendix A – User Manual

1. Network Setup

The first step of the simulation program is to setup the network on the working panel which is the green area on main window. To place a network device on it, the user clicks on the button with the icon of the device that is on the tool bar. This causes the cursor to take the shape of the device as seen in figure 13 below were the cursor has taken the shape of a router since the user has chosen to place another router on the working panel.

Once the user left clicks on any place on the working panel, the device will appear on the working panel with a default name given to the device. The default name is the type of device with a number next to it. If the user right clicks anywhere in the main window while the cursor has the shape of a device, the cursor will again take its normal shape and cancel the placement of the device to the working area. If the user left clicks on anywhere other that the working panel, no new device will be added. If the click was over another device, remove or connect button, the cursor will change to that device or function.

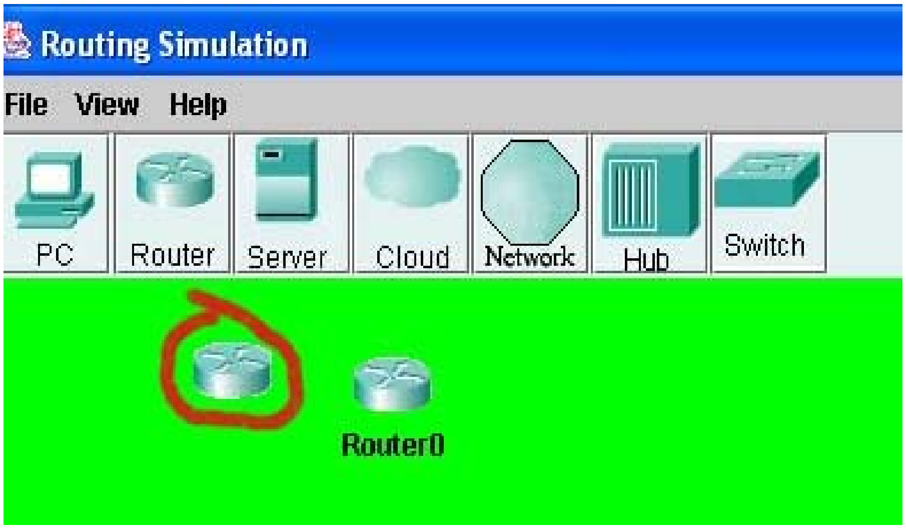
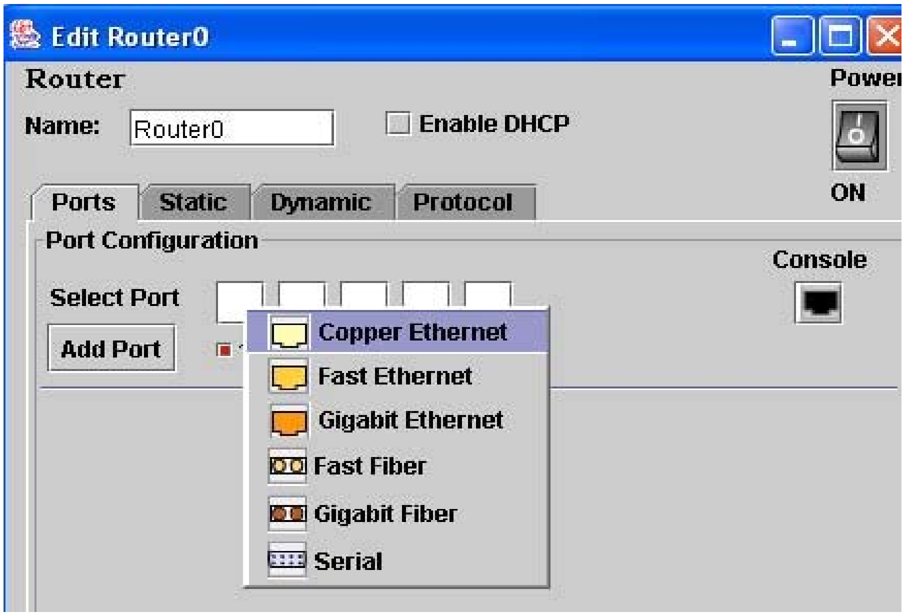


Figure 13: Cursor takes shape of a router.

Once the user has placed all the devices on the working panel, the user has to connect the devices. To do this the user clicks on the “Connect” button on the toolbar. This causes the cursor to change to the connect icon. Then the user left clicks on the first device. When the user moves the mouse way from the first component a line appears that is attached to the first device and the mouse cursor (see figure 1 above). When the user clicks on the second device the line will be attached to both devices and automatically defined. A connection between routers or networks is predefined as a serial line. Interconnection between PCs, servers, hubs, switches, and routers are of type Ethernet by default. If for example the user wishes that the connection between routers is of type Ethernet instead of serial, the user has to define a port of type Ethernet either in the first device that is going to be connected or in both devices. For the user to remove the connect cursor, the user can right click anywhere in the main window or click on another button causing the cursor to change.

To configure a device, the user has to double click on the device causing the configuration GUI for the device to appear. If the user wants to define a port, the user has to left click on an empty port causing a popup menu to appear as seen in figure 14. The user then selects a type of port from the menu. See section 1.1 of the appendix to learn about individual device configuration.



Name: Router0

□ Enable DHCP

Powe ON

Figure 14: Port type definition popup menu.

If the user is not satisfied with the location of a device on the working panel, the user can move the device by clicking on the device and hold the mouse button while the user drags the device to the new location. The user can also erase devices and lines. To do this, the user clicks on the "Remove" button. This causes the cursor to take the shape of half a pencil with an eraser. To remove a device or line the user left clicks on the device or line. If the user clicks on a device that has lines connected to it, all the line will be erased. To remove the remove cursor, the user has to right click anywhere on the main window or click on another button.

2. Device Configuration

PC – User must configure port type, IP address, submask, bandwidth, and duplex type for the only port. Also the user must enter the default gateway. If the PC is connected to a server or router with DHCP enabled, the user can click on the "DHCP" button which will automatically assign the IP address, submask, and default gateway.

Server – User must configure port type, IP address, submask, bandwidth, and duplex type for the only port.

Hub – No configuration is needed. The user only has to open the configuration window and either click "Save" or "Save and Close" button.

Network – The network is an abstract component which connects routers in a broadcast network. The only configuration needed for the network is to define the ports that are connected to it. This can be done manually or automatically. To configure manually, the user does it by entering the same port values as found at the other end of the line on the router. To avoid doing this task manually, the user can leave the configuration of the network till the end. Then the user can click on the "Get Network" button that will automatically fill the port information. To save the network values the user has to either click "Save" or "Save and Close" button. The reason for the needed configuration is that the visual part of the simulation needs to know which port the packet was sent from in order to later send it out all the other lines that are not connect to this port.

Switch – The user can make use of any switch without making any configurations on to it. The user only has to open the component and save it by clicking on either the "Save" or "Save and Close" button. If the user wishes to use virtual local area networks (VLANs), the user must first go to the VLAN tab and create all the VLANs. To create a VLAN the user must enter the VLANs number and name. After creating all the VLANs the user then selects the ports tab and selects the VLAN associated with each port.

Router – The router is the most complex device to configure for the network simulator. First the user has to configure all the port settings for all the connected ports with "on" status. The router is predefined with five slots for ports, but if the user needed more ports, then they will have to click on the "Add Port" button. This action will add an extra port on the router. The user can add up to four extra ports for the router. Unless the user saves the router settings, the newly added will not appear the next time you open the configuration window for the router.

If the user needs to define static routes, user needs to click on the static routes tab. There the user can enter up to ten different static routes.

When the user wants to configure the routing protocol, the user has to click on the protocol tab. On the top part of panel for the protocol tab, there is a drop down menu that the user can chose the routing protocol. The only option that appears is OSPF since it's the only implemented routing protocol.

For OSPF, the first tab contains the general configuration settings. Here the user enters the router id, the priority number for the router, selects type of router, the user also selects if the router can become the designated router or the backup designated router.

The next tab contains the configuration for the areas. Here the user specifies the id number for the area and the type of area. The area type can be a regular area, the backbone area or a stub area in which the user has to enter the cost for the area. If the router is an area boarder router, the user can add up to four more areas. The final configuration for an area is the range of addresses the area contains. To set this up, the user must first select a row on the area table that contains an area. Then the user can either click on the "Edit" or "Add Range" buttons. If the user clicks on the "Edit" button the user will be able to edit all the area's settings including adding or removing the address ranges. If the user clicks on the "Add Range" button, the user will be able to only add or remove address ranges. The user will be able to add up to ten addresses.

Tab number three on the protocol panel contains the configuration for the router interfaces. In order to configure an interface, the user must have first completed the port configuration and must have had defined at least one area. To configure the interface, the user first selects a port. That interface gains the IP address, submask, and cost of that port. Then the user selects the type of network the interface is connected to and the area associated with the interface. The final configuration needed for the interface is hello interval, router dead interval, the retransmission interval, and delay interval. These values must be the same for all the routers in the AS. For that reason default values have already filled out and can be changed by the user at any time. In order to save the protocol settings, the user must make sure that all the ports have been associated with an interface otherwise an error could occur.

The final two tabs are only shown when either the router has been selected as either a border router or a boundary router. For boarder routers, the virtual link tab becomes enabled. In this tab the user can define the end point of the virtual link, the transit area associated with the virtual link, as well as the hello interval, router dead interval, the retransmission interval, and delay interval. In order for virtual links to work the user must configure the virtual link in both of the area boarder routers. The user has a limit of five possible virtual links. For routers selected as boundary router, the external routes tab becomes enabled. Here the user can specify up to five different external routes. If an exterior gateway protocol were to be implemented, the router could produce even more external routes.

3. Simulation of the network

When the user has finished configuring and saving all the devices on the working panel the user can then go test the network configuration. To reach the simulation GUI the user has to click on the “Simulate” button found on the toolbar. Inside the simulation window the user has the option of choosing how to run the simulation. As seen in figure 15 below there are three buttons that contain the options. If the user hits button with the word “Initialized” on it, the simulator will start with the routing table already created. There are two ways of letting the simulator know at what time to stop. The first method used is when each router has sent out three consecutive hello packets out of all its interfaces, the router knows that no new information is going to be received. The second method is that the simulator stops at time 1500. The reason for this is to avoid loops caused by bad configuration.

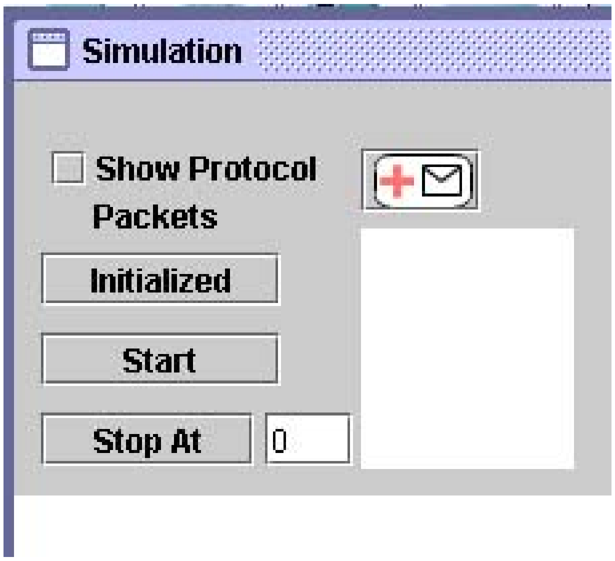


Figure 15: Simulation mode buttons

If the user hits the “Start” button the simulation starts at time 0 allowing the user to see all the protocol packets that need to be sent to complete the routing table and know about the topology of the AS. The final simulation mode allows the user to stop the simulation at a user-defined time. This feature allows the user to see what happens from a certain point on and its very useful for debugging of future protocols.

On the top right side of the simulation GUI the user has three buttons to controls the timeline and the flow of the simulation (see figure above). When the user hits the skip button (see figure 16 below) the simulation will increase one time unit. If the user presses the play button (see figure 17 below) the play button will change its icon to a pause icon (figure 18 below). Play action will increase the simulation time by one unit and will wait for all the packets to be delivered to their next hop. When all the packets have been sent, the simulator’s time will again increase repeating previous cycle. If the user wants to stop the continuous play operation, the user has only to hit the play button that has the pause icon. If the user wants to see packets sent at a previous time, the user has to click on the skip back button see in figure 19 below. This action will cause the animation to go back any number of time units in the simulation. Doing this does not reverse time inside network devices.

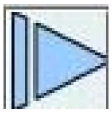


Figure 16: Skip button

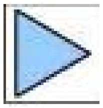


Figure 17: Play button

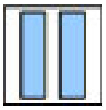


Figure 18: Pause icon

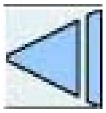


Figure 19: Skip back button

The user has several other options for controlling the display of the simulation. If the user does not want to see the protocol packet, the user can uncheck the check box labeled as “See Protocol Packet”. The user can also control the speed of the moving packet by the slider that’s located under the play button (see figure 5 above).

4. **Saving the project**

Before the user selects the project to be saved, the user must make sure all the devices have been configured and saved; otherwise errors can occur while saving the project. To save a project, the user has to click the File menu and select “Save” option. A file chooser window appears where the user chooses the directory to save the project. The user can chose any name to save the project and has to end it with ".ns" extension. Otherwise the file will not appear in the file chooser window when trying to open the project again.

The user will be ask to save the project any time the user select the "New" option from the file menu or when the user selects of the "Exit" option.

5. **Adding routing protocols to router**

To add a new routing protocol to a router, the user must edit Router, RouterGUI, MainWindow, Icons, and SimulationGUI classes. Inside the RouterGUI class the user has to add a new JPanel that contains all the configuration settings for the new protocol. The user will also have to modify the save function for the router. Inside the Router class, the user will have to add to receivePacket function to accept the new protocol and an object variable that will hold the protocol class. Inside the SimulationGUI class, the user has to make changes the protocol packet legend. Here the user will have to make a new popup menu with the type of packet the new protocols sends out. If the user wants to show some type of information about the protocol, like in the case of OSPF where user can either see the LSAs the router has received or the shortest path tree for the router, the user will have to update the popup menu that is found in the Icons class. The final change that needs to be made is in the save function inside the MainWindow class. This function has to be updated to save the settings for the new protocol that the user has implemented.

6. **Adding hardware devices**

To add any other network device the user has to add a button for that component on the toolbar of the MainWindow class. A new window GUI must be created to allow the configuration of the new device. IconDisplayer2 has to be modified to include the new device and allow the configuration window to be opened. Also ConnectLine class has to be modified to define the default connection type. A new window GUI must be created to allow the configuration of the new device.

Appendix B – Class Diagrams

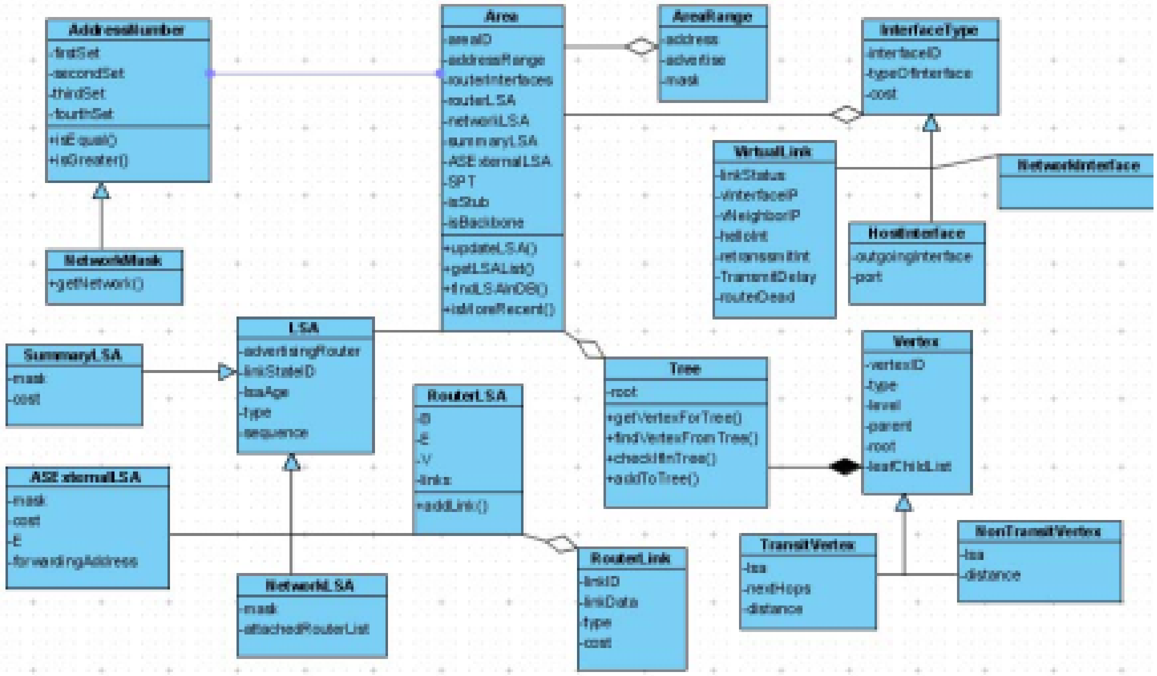


Figure 20: Class Diagram for Area Data Structure

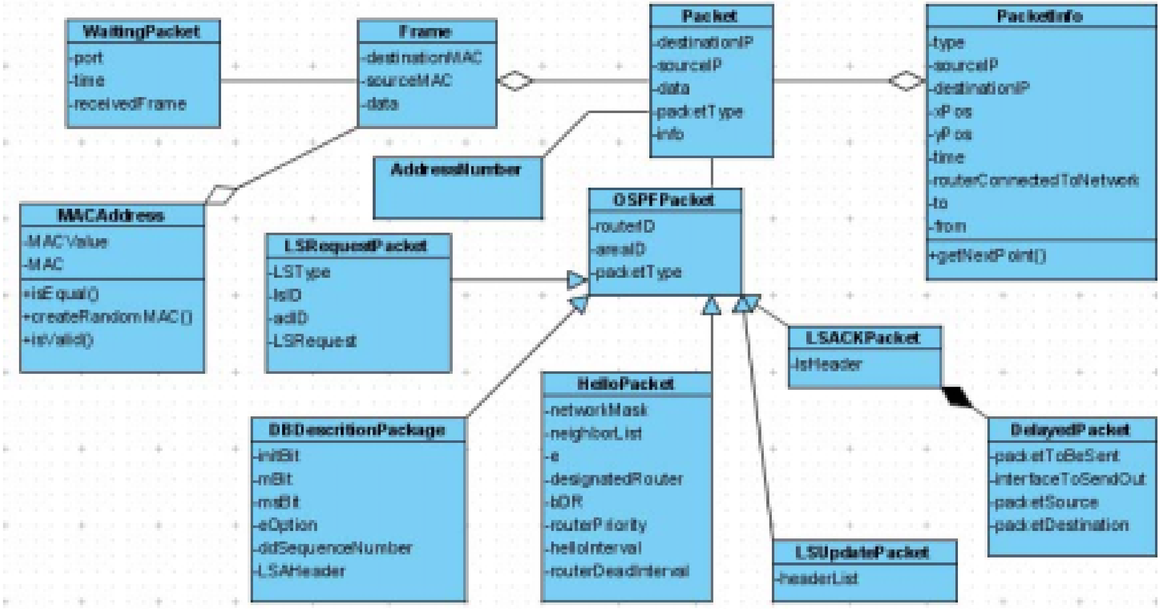


Figure 21: Class Diagram for frames and packets

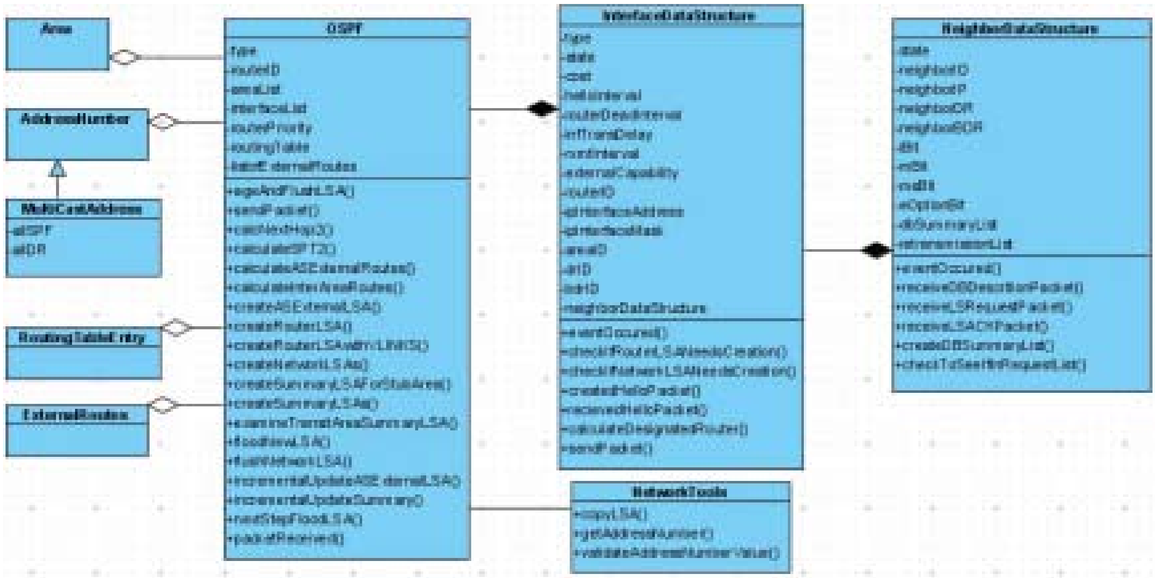


Figure 22: Class Diagram for OSPF

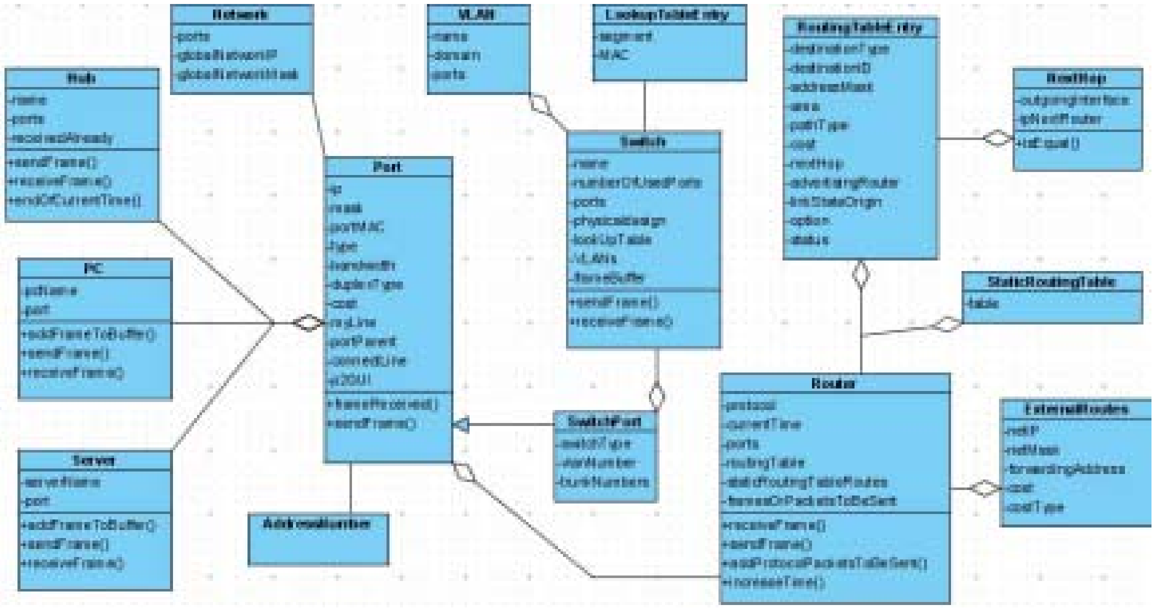


Figure 23: Class Diagram for Hardware Components

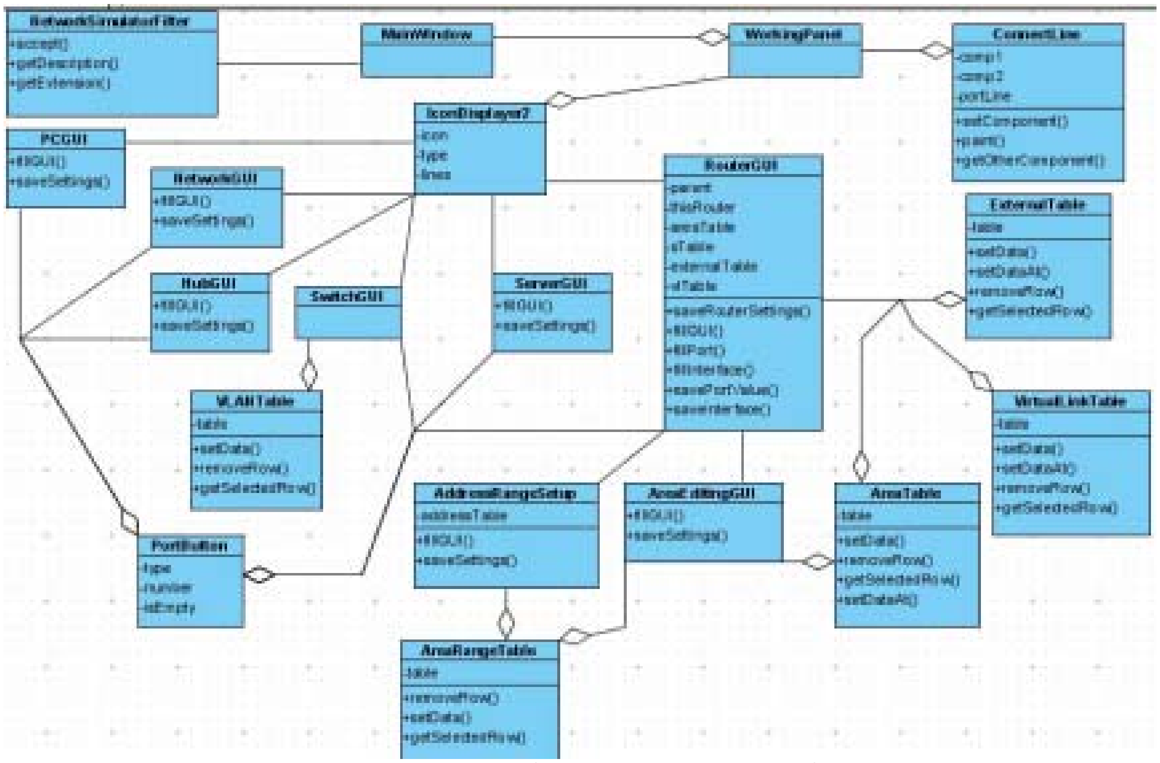


Figure 24: Class Diagram for network setup and configuration

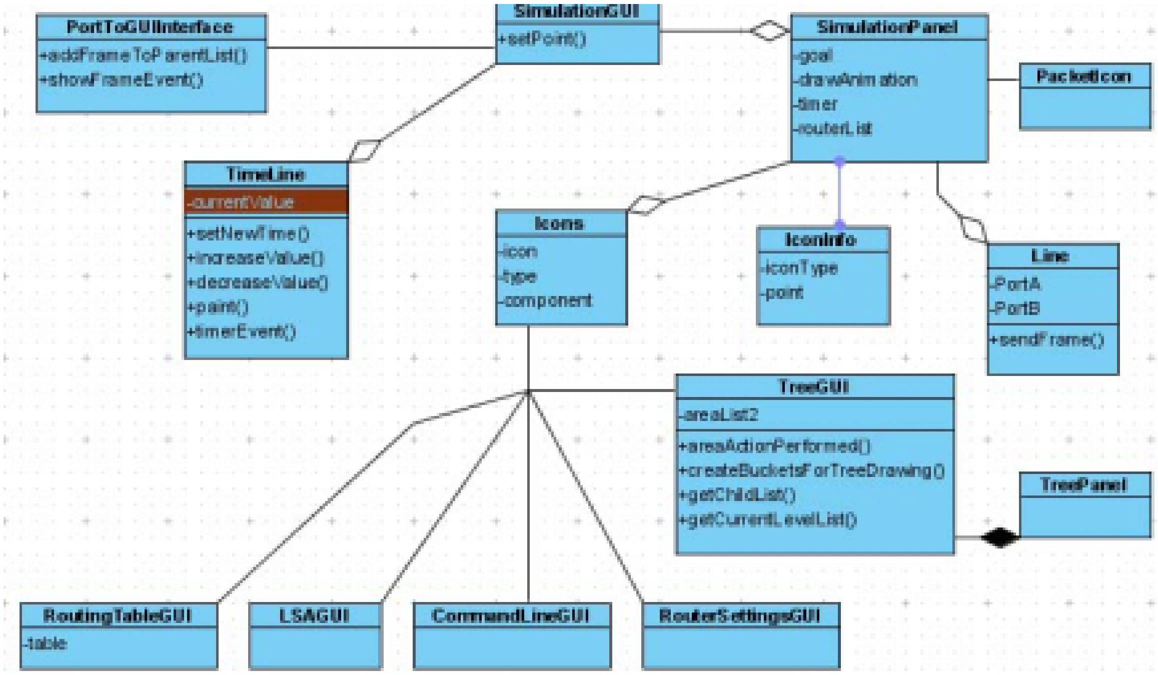


Figure 25: Class Diagram for Simulation