Chapter 5 System Design

This chapter details the design considerations for the MPLS VPN simulator. In keeping with the object-oriented programming approach that has been chosen, the design aspects of the MPLS VPN simulator focuses on the design of object classes.

The chapter begins with the system architecture design. It discusses the overall architecture design for MPLS VPN. Then, it follows by the object-oriented design, which details how the new MPLS VPN components are integrated into the current UMJaNetSim simulator.

The third section describes the design of the object classes for the MPLS VPN components. Each of the classes is described individually of their attribute as well as method. It is followed by a discussion on the topology design.

The final section summarizes the details of this chapter.

5.1 System Architecture Design

The new MPLS VPN simulator design is categorized into three sections. There is MPLS VPN configuration design, MPLS VPN simulation design and MPLS VPN detail logging design. The following details each of the design precisely.

5.1.1 MPLS VPN Configuration Design

One of the functionality of the MPLS VPN simulator is to let the user configure the VPN before the simulation start. In order to ease the user during configuration, the simulator needs to provide GUI during configuration. Figure 5.1 shows the detail of the MPLS VPN configuration design.

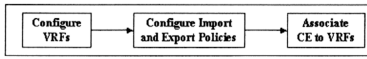

Figure 5.1 MPLS VPN Configuration Design

The simulator allows the user to predefined a few VPNs. This process can be automatically or manually. The predefined VPN information will contain the VPN name, the route distinguisher and the VPN color. During the VRF configuration, the user needs to select the PE-Router, input a VRF name and select the VPN name. The simulator will define the route distinguisher based on the VPN name selected.

This process is followed by the configuration of import ad export policies. The user needs to select the predefined VRF name and their import and export policies. The simulator will store the export route target and the import route target in two different lists for each VRFs. Then, the user needs to configure the PE-CE links where the CE interface is associated to the previously defined VRFs.

5.1.2 MPLS VPN Simulation Design

MP-iBGP is used to advertise customer VPN routes between PE-Routers that learned from connected CE routers. All MP-BGP sessions are internal BGP sessions because the session is between two routers that belong to the same autonomous system. Figure 5.2 shows the detail of the MPLS VPN simulation design.

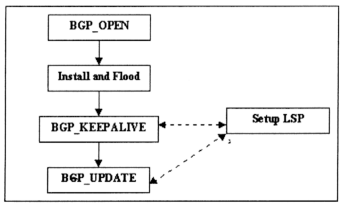

Figure 5.2 MPLS VPN Simulation Design

When the simulation starts, PE-Router will schedule an event to generate the BGP_OPEN message. The BGP_OPEN message contains the route distinguisher, a list of export route target and the router id. After the BGP_OPEN message has been generated, the message will be flooded to all the PE-Router. When others PE-Routers receive the BGP_OPEN message,

verification is performed to check whether the import route target match the export route target of the BGP_OPEN message. If it doesn't match, the message is dropped. If it matches, a BGP_KEEPALIVE message is generated and stored in a list. The BGP_KEEPALIVE message will contain the destination ip, which is the advertising PE-Router IP and the source IP. The BGP_KEEPALIVE message is to notify the advertising PE-Router that the PE-Router is alive. Figure 5.3 shows the BGP_OPEN flooding process.

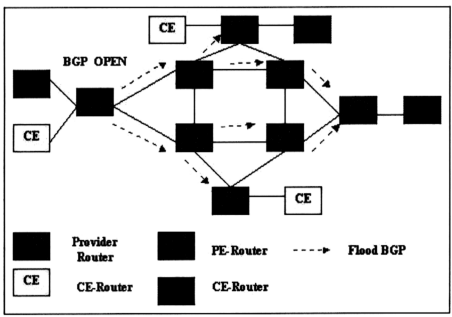

Figure 5.3 BGP_OPEN Flooding Process

After all the PE-Routers send their BGP_OPEN message, the PE-Router will schedule an event that sends the BGP_KEEPALIVE message back to the advertising PE-Router. It will check whether the LSP had already established between the two PE-Routers. It needs to setup the LSP if it doesn't exist. When the advertising PE-Router receives the BGP_KEEPPALIVE message, the advertising PE-Router will generate the BGP_UPDATE message and store it in a list. The BGP_UPDATE message will contain the route distinguisher, a list of export route target, a list of destination IP, which are the IP address and subnet mask of the CE-Routers and a list of VPN labels that associated with the destination IP. The VPN labels are generated automatically by the PE-Router. Besides that, the BGP_UPDATE message also contains the next hop PE-Router ID.

After all the PE-Router send their BGP_KEEPALIVE message, the PE-Router will schedule an event that sends the BGP_UPDATE message. Before the BGP_UPDATE message is sent, it needs to check whether the LSP had already established between the two PE-Routers. After the LSP had been established, the BGP_UPDATE message is sent to destination PE-Router through MPLS label switching.

When the destination PE-Router receives the BGP_UPDATE message, it will filter the BGP_UPDATE message and import the routes into the relevant VRFs. The routes information will contain the destination IP, subnet mask, VPN label, IGP label and the next hop PE-Router ID. The information will be stored in the forwarding table of VRFs.

The MPLS VPN backbone has been setup completely. The IP packets send by the VPN customer will reach the destination based on VPN label and IGP label through label stacking.

5.1.3 MPLS VPN Detail Logging Design

During the simulation, all the process that are running need to be log into the log file in order to let the user verify the correctness of the simulator. The users will know whether the MPLS VPN backbone has been setup by checking the log file. Besides that, the simulator also provides the GUI for viewing VPN detail. Figure 5.4 shows the detail of the MPLS VPN detail logging design.

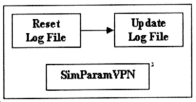

Figure 5.4 MPLS VPN Detail Logging Design

Before the simulation start, the simulator will allow the user to reset the log file. During the simulation, the process running will automatically update into different log file based on simulation step.

5.2 Object Oriented Design

MPLS VPN components will inherit all the features from the SimComponent by adding more properties and methods on the ATMLSR and the IPBTE component. Besides that, new parameter is added and the parameter will inherit from SimParameter. Figure 5.5 shows the MPLS VPN simulator objects.

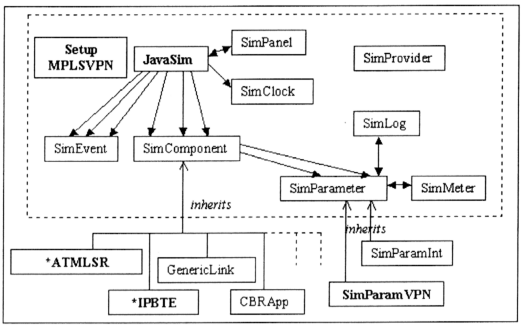

Figure 5.5 MPLS VPN Simulator Objects

5.3 Class Design

This section gives a description on classes design of the MPLS VPN simulator. The properties and the functionality of each class are discussed.

5.3.1 SetupMplsVPN Class

The SetupMplsVPN class is the class that performs all the configuration of VRFs and other VPN details. This class is called when the user select tools and Setup MPLS VPN from the menu list. The SetupMplsVPN class must perform the following functions:

- This class must provide three configuration steps for the user to setup the MPLS VPN. The first step is designed for the user to select the PE-Router, define a VRF name and select a VPN for the VRF. The second step is designed for the user to select the VRF and configure the import and export policies. The final step is designed for the user to associate the CE to the relevant VRF.
- This class must let the user to modify the configuration.
- This class must display the updating list in a table.
- This class must update the VRF into the correct PE-Router.

Table 5.1 lists the design of the major attributes and methods of the SetupMplsVPN class.

Table 5.1 Attributes and Method of SetupMplsVPN

Major Attribute	Major Methods
A list of VRF	Configure VRF
A list of import and export policies	Configure import and export policies
A list of CE-PE link.	Associate CE to VRF
	Update PE-Router with the VRF

5.3.2 SetupIP Class

The SetupIP class is the class that used to setup the IPBTE source network and subnet mask. This class is called when the user select Setup Component IP from the menu list. This class must perform the following functions:

- This class must provide the user to input the source network and subnet mask.
- This class must verify the correctness of the input.
- This class must update the IPBTE with the correct source network and subnet mask.

Table 5.2 lists the design of the major attributes and methods of the SetupIP class.

Table 5.2 Attributes and Method of SetupIP

Major Attribute	Major Methods
IP	Get IP and Subnet mask value
Subnet mask	Verify the correctness of input
IPBTE Name	Set the IP and Subnet mask value to the selected IPBTE

5.3.3 SimVPN Class

The SimVPN class is the class that used to predefined VPN name and the route distinguisher. This class is called when the user select Pre-Defined VPN from the menu list. This class must perform the following functions:

- This class must let the user predefined VPN automatically or manually.
- The class will predefined a VPN with their name, route distinguisher and the VPN color.

Table 5.3 lists the design of the major attributes and methods of the SimVPN class.

Table 5.3 Attributes and Method of SimVPN

Major Attribute	Major Methods
VPN Name	Set VPN Name
Route Distinguisher	Set Route Distinguisher
VPN Color	Set VPN Color

5.3.4 IPBTE Class

The IPBTE class is the class that acts like a CE-Router in the MPLS VPN backbone. The existing IPBTE class needed to be modified in order to support the MPLS VPN backbone. This class must perform the following functions:

- This class must set the VPN state to true after the VPN had been setup.
- This class must change the color of the name based on the VPN site selected.

Table 5.4 lists the design of the major attributes and methods of the IPBTE class.

Table 5.4 Attributes and Method of IPBTE

Major Attribute	Major Methods
VPN State	Set VPN State
VPN Color	Set Color

5.3.5 ATMLSR Class

The ATMLSR class is the class that performs the functionality of PE-Router as well as P-Router. The existing ATMLSR class needed to be modified in order to support the MPLS VPN backbone. This class must perform the following functions:

- This class must be able to support VPN and non-VPN.
- If this class act like a PE-Router, it must be able to generate BGP_OPEN message that include route distinguisher, a list of export route target and the router ID. After generate the BGP_OPEN message, this class must be able to flood the message to others ATMLSR.
- This class must be able to verify the export and import route target.
- This class must be able to generate BGP_KEEPALIVE message.
- This class must be able to setup LSP among PE-Routers
- This class must be able to generate BGP_UPDATE message.
- This class must import the route into the correct VRF.
- This class must log all the simulation process in the log files.

Table 5.5 lists the design of the major attributes and methods of the ATMLSR class.

Table 5.5 Attributes and Method of ATMLSR

Major Attribute	Major Methods
A list of VRFs	Generate BGP_OPEN message
BGP Class	Flood and install LSA
A list of LSP	Generate BGP_KEEPALIVE message
A list of LSA	Setup LSP
CallRecord Class	Perform LDP label mapping
A list BGPtable	Generate BGP_UPDATE message
A list BGPUpdate	Perform label stacking and label forwarding
VF Class	Log MP-iBGP update
	Log IP packet forwarding

5.3.6 SimParamVPN Class

The SimParamVPN class is the class that let the user view the VPN detail that has been setup. This class is called when the user click the VPN Detail Button at the ATMLSR properties dialog box. The SimParamVPN class must perform the following functions:

- This class must display the VPN detail for the PE-Router
- This class must list out all the VRFs that have been setup for the PE-Router.

Table 5.6 lists the design of the major attributes and methods of the SimParamVPN class.

Table 5.6 Attributes and Method of SimParamVPN

Major Attribute	Major Methods
A list of VRF A list of import and export policies A list of CE	Display the VRF Detail Display import and export policies Display the CE associate to the VRF

5.4 Topology Design

The MPLS VPN makes it possible for a number of topologies that are hard to implement in either the overlay VPN model or the peer-to-peer VPN models. Numerous topologies can be deployed using the MPLS VPN architecture and its associated tools. This section will describe the three common topology designs for the MPLS VPN simulator.

5.4.1 Intranet Topology

The Intranet topology between multiple sites is the simplest VPN topologies that can provision using the MPLS VPN architecture. This topology is the basic VPN network structure that provides any-to-any connectivity between sites using the enhanced peer-to-peer models. Furthermore, this topology allows overlapping IP address between multiple VPNs. Figure 5.6 shows an example of the Intranet topology [1]. Both customer sites have any-to-any, non-redundant, Intranet service from the MPLS VPN backbone with only one CE to PE connection.

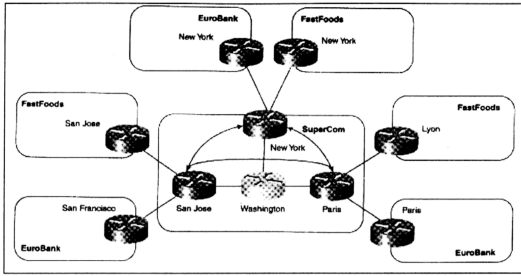

Figure 5.6 MPLS VPN Intranet Topology

5.4.2 Intranet and Extranet Integration Topology

Extranet support within the context of the MPLS VPN architecture simply involved the import of routes from one VRF into a different VRF that services another VPN site. This will create a new set of topology, which is the Intranet and Extranet integration. Figure 5.7 shows an example of the Intranet and Extranet integration topology [1].

From the figure, the two organizations in two different VPN are capable of communicating directly across the MPLS VPN backbone as the organizations import each other's route into their relevant VRFs. However, this type of connectivity didn't allow the overlapping of address between the two VPN customers. This means that the address must be unique between the two organizations.

5.4.3 Central Service Topology

Central service topology is another common topology that must be implemented with MPLS VPN technology. In this topology, the client sites can access services on central servers located at one or more central sites. However, the client sites are not able to communicate with each other. This topology normally provides many services to the client. One of the most popular services is the application hosting in which a service provider provides access to

applications residing on common servers. Figure 5.8 shows an example of the central service topology [1].

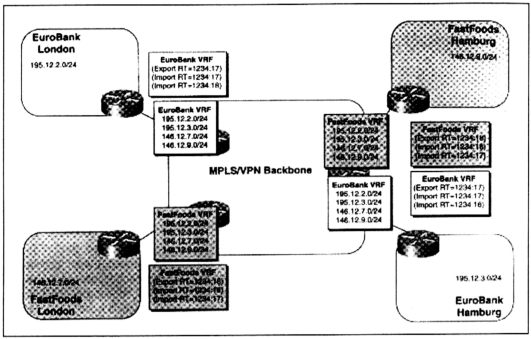

Figure 5.7 MPLS VPN Intranet and Extranet Integration

Under this topology, the clients' site needs to be placed into different VRF, as the clients can't communicate with each other. Furthermore, every client site must use different route distinguisher to prevent route conflict between the client sites.

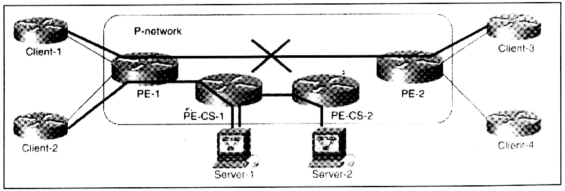

Figure 5.8 MPLS VPN Central Services Topology

5.5 Chapter Summary

This chapter covers the major design issues for the MPLS VPN simulator. This includes an overview of the system architecture, which focus on MPLS VPN configuration design, MPLS VPN simulation design as well as MPLS VPN detail logging design. The classes design give an illustration of the defined attributes and methods for each class. This final section discusses the topology design.