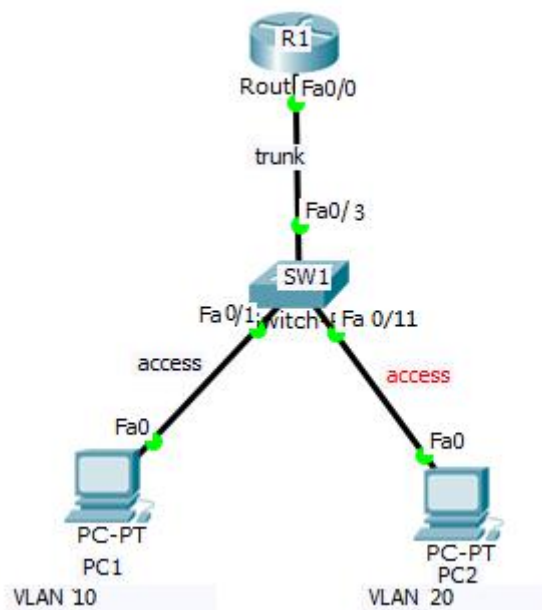


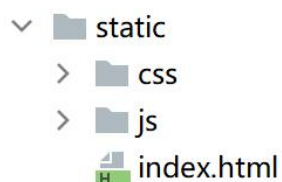
1. 软件架构设计

本系统前端采用 html，通过 http 请求和后端交互。后端采用 SpringBoot 框架，收到前端的请求后调用脚本。脚本采用 Python 来实现，具体使用 telnetlib 包来与路由器进行连接及通信。系统底层网络拓扑图如下图所示。本系统涉及一台路由器和一台交换机，分别命名为 Router 和 Switch，另外还需要两台 PC 机，IP 地址和 Vlan 分别为 192.168.10.1、Vlan10 和 192.168.20.1、Vlan20。



2. 软件体系设计

3.1 前端包结构



index.html 是主页面，主要是添加，修改交换机和路由器的配置，js 包下是放的主页面的 js 脚本，css 包下是放的主页面样式的 css 文件。

3.2 前后端接口

1、配置交换机的接口 /api/setSwitch，请求类型是 post 请求

相应的参数：host_ip，password_login，password_enable，vlan1，interface1，vlan2，interface2，interface_router。

在前端主页面输入相应的交换机的配置信息，调用后端的/api/setSwitch 接口，将参数传入，调用准备好的 python 脚本，进行交换机的配置，配置成功后，返回配置成功，否则返回配置失败。

2、配置路由器的接口 /api/setRouter，请求类型是 post 请求

相应的参数：host_ip, password_login, password_enable, gateway1, mask1, vlan1, gateway2, mask2, vlan2.

在前端主页面输入相应的路由器的配置信息，调用后端的/api/setRouter 接口，将参数传入，调用准备好的 python 脚本，进行路由器的配置，配置成功后，返回配置成功，否则返回配置失败。

3.3 后端文件结构

文件名	主要功能
SetSwitch.java	对设置交换机的请求做路由转发
SetRouter.java	对设置路由器的请求做路由转发
DoSetSwitch.java	完成交换机配置的功能类
DoSetRouter.java	完成路由器配置的功能类

4. 软件详细设计与实现

4.1 Telnet 类的实现

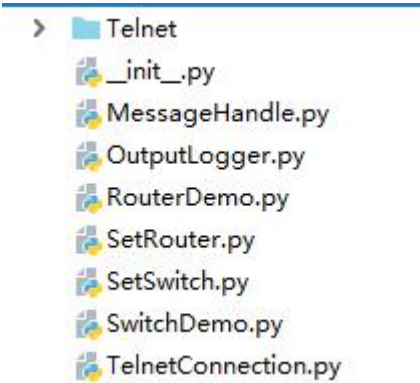


图 4.1 Telnet 包结构

TelnetConnection 类，负责与路由器和交换机进行交互，并且提供路由器与交换机的登录/退出、命令执行操作，其主要功能使用了 telnetlib 库进行实现。

loginHostRouter(host_ip, password_login, password_enable)：登录路由器

loginHostSwitch(host_ip, password_login, password_enable)：登录交换机

executeOneCommand(command, cmd_type): 执行一条命令，返回输出结果

executeSomeCommand(commands, cmd_type): 执行多条命令，返回输出结果

logoutHost(): 退出登录

MessageHandle 类，负责处理从控制台返回的脏数据

handleMsgFromSwitch(msg):处理从交换机输出的脏数据

handleMsgFromRouter(msg):处理从路由器输出的脏数据

handleAllMsg(msgs): 处理返回的字符串列表

SetRouter 类，负责传入参数配置路由器

setRouter(host_ip,password_login,password_enable,gateway1,mask1,vlan1,gateway2,mask2,vlan2):
配置路由器，将网段划分到不同 vlan 中

SetSwitch 类，负责传入参数配置交换机

setSwitch(host_ip,password_login,password_enable,vlan1,vlan2,interface1,interface2,router_interface):配置交换机，划分 vlan

4.2 自动化配置单臂路由的实现

路由器配置的实现：

服务端提供了一个接口/api/setRouter，此接口的描述如下：

接口名：/api/setRouter；

请求参数：

host_ip: IP 地址, password_login: Login 密码, password_enable: Enable 密码, gateway1:

Gateway-1, mask1: Gateway-1 对应的 Mask, vlan1: Gateway-1 对应的 Vlan, gateway2:

Gateway-2, mask2: Gateway-2 对应的 Mask, vlan2: Gateway-2 对应的 Vlan

请求类型：POST

用户在前端填写好路由器配置的信息后，点击配置路由器按钮，应用将向后端发送配置路由器的请求。

路由器

IP地址

Login密码

Enable密码

Gateway-1

Gateway-1对应的Mask

Gateway-1对应的Vlan

Gateway-2

Gateway-2对应的Mask

Gateway-2对应的Vlan

配置路由器

交换机配置的实现：

服务端提供了一个接口/api/setSwitch，此接口的描述如下：

接口名：/api/setSwitch；

请求参数：

host_ip: IP 地址, password_login: Login 密码, password_enable: Enable 密码, vlan1: Vlan1, interface1: Vlan1 对应的接口, vlan2: Vlan2, interface2: Vlan2 对应的接口, interface_router: 路由器与交换机连接的接口

请求类型：POST

用户在前端填写好交换机配置的信息后，点击配置交换机按钮，应用将向后端发送配置交换机的请求。

交换机

IP地址

Login密码

Enable密码

Vlan-1对应的接口

Vlan-1对应的接口

Vlan-2

Vlan-2对应的接口

路由器与交换机连接的接口

配置交换机

4.3 校验单臂路由配置的实现

不同 Vlan 里的主机虽然网段不同，但依然可以相互 ping 通。

5. 系统主要功能描述