目录

1. 题目与设计要求…………………………………………………3
2. 系统功能概述图…………………………………………………4
3. 算法描述…………………………………………………………6
4. 界面设计…………………………………………………………6
5. 主要程序片段……………………………………………………8
6. 系统运行与调试结果…………………………………………..11
7. 小组任务分工说明……………………………………………..12
8. 还存在的问题与今后的解决思路……………………………..12
9. 个人感受………………………………………………………..12

10、 参考文献………………………………………………………..13

**一、题目与设计要求**

**1. 题目**：网桥转发表处理模拟工具设计

**2. 设计要求**：给定局域网络图和几个转发信息，能够按照逆向学习算法，自动生成转发表并图形显示；此后的转发信息能够参照转发表正常工作。输入输出内容能够保存。

**3. 操作系统：**Windows 10 教育版

**4. 运行环境：**

* Python 3.5.2
* Anaconda Jupyter Notebook
* Microsoft Office Visio
* Microsoft Office Word

**Python模块**

* Tkinter
* Pandas
* PIL
* Pyinstaller

**二、系统功能概述图**

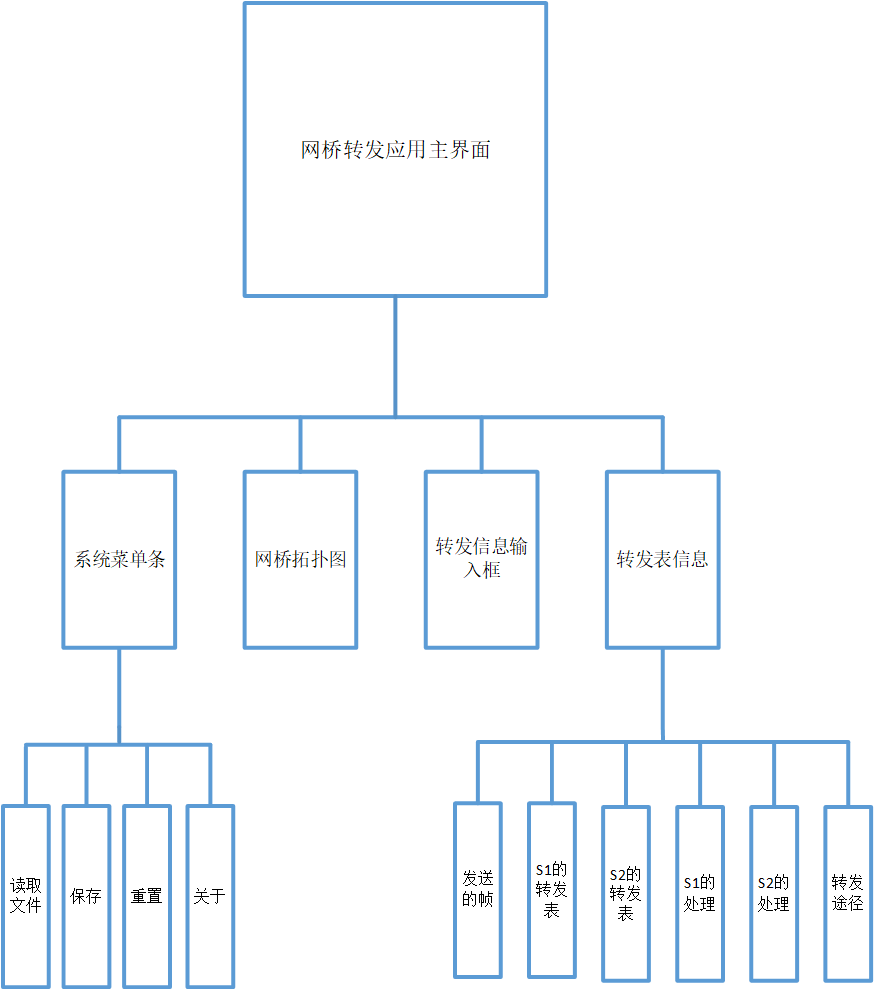
****

图1功能框图

图2功能流程图

**三、算法描述**

当网桥刚接入时，转发表是空的，网桥通过逆向学习来获取转发信息并逐步建立转发表。

逆向学习算法指的是：

网桥通过检查达到帧的源地址以及输入端口来发现目的节点及其对应的输出端口。

随着收到的帧不断增多，转发表就逐渐建立完备。

对于未知的目的节点，网桥采用扩散转发法，向所有端口发送。

一旦未知结点开始发送且发送的帧到达网桥，网桥就可以通过逆向学习法获取转发信息，填入转发表种，随后就可以按转发表来转发。

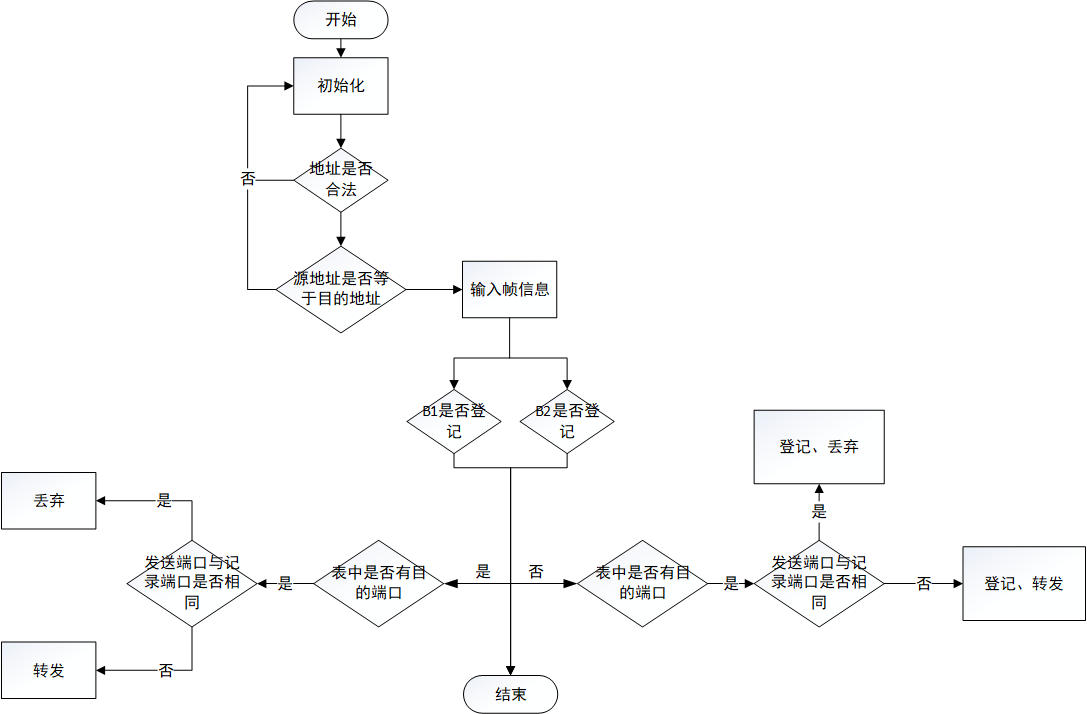
****

图3算法流程图

**四、界面设计**

1.初始主界面

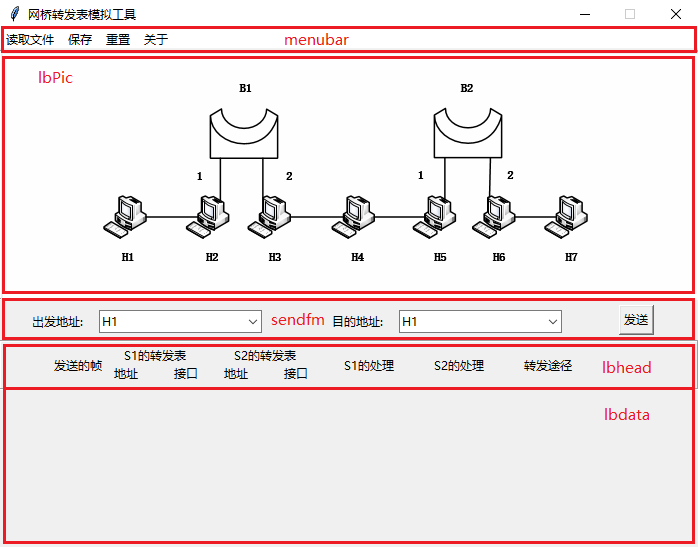


图4

界面最上方设置菜单条，用于执行读取、保存文件和重置运行数据的操作。同时可以在“关于”选项中查看相关人员信息以及系统说明。

在菜单条下首先是模拟网桥网络拓扑结构的网络拓扑图，可以向使用者展示通过不同主机通讯时网桥的工作结构。

接下来是数据的输入和执行区域，逐次从已知的出发地址向目的地址发送。数据输入框设计为带下拉条的形式，这样可以方便使用并防止使用者输入范围外的数据。用户通过点击“发送”按钮执行转发任务。

最后是数据执行后的结果显示列表，列名分为发送的帧、S1的转发表、S2的转发表、S1的处理、S2的处理以及转发途径。执行一次范例发送H1->H5如下图所示：

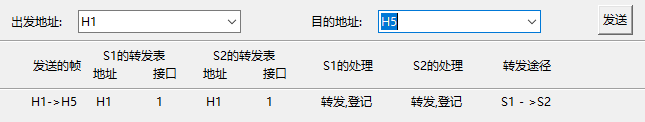


图5

2转发信息超过八条，系统弹出提示。



图6

3点击“保存”弹出询问保存提示。



图7

4点击“读取文件”出现弹窗，选择读取路径。

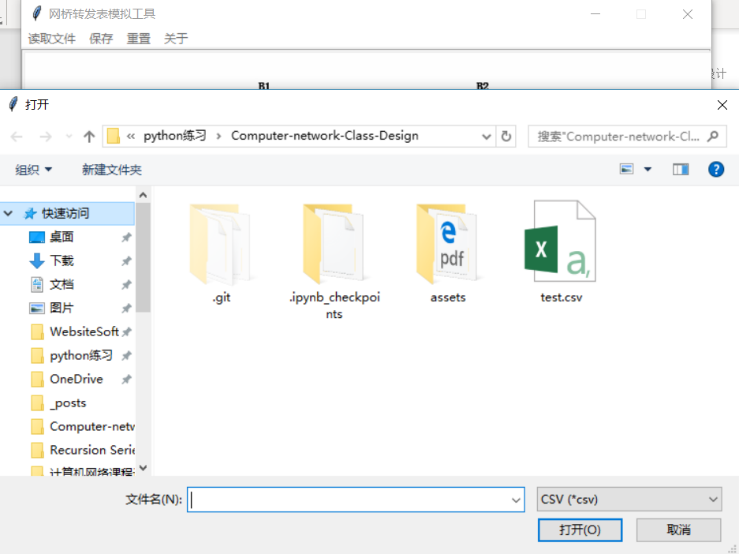


图8

**五、主要程序片段**

**1.发送事件函数：**

def btnSend():

global i,srcAddr,desAddr

if(i > 7):

tkinter.messagebox.showinfo('提示','转发信息已到达最大数据')

return

dataChip = ["","","","","","","",""]

if(not flag): #flag为true，不执行此步骤

srcAddr = varSrc.get()

desAddr = varDes.get()

if(srcAddr == desAddr):

return

handleB1 = ""

handleB2 = ""

routPort = ""

flagb1 = True #表示两个转发表都要转发

flagb2 = True

seqS = [0,0]

if(srcAddr in lan1):

seqS = [0,1]

elif(srcAddr in lan2):

seqS = [1,0]

elif(srcAddr in lan3):

seqS = [1,0]

else:

return

if(desAddr not in lan1):

if(desAddr not in lan2):

if(desAddr not in lan3):

return

count = 0

while(count!=2):

if(seqS[count] == 0 and flagb1):

if(srcAddr not in S1):

S1[srcAddr] = hostSegOne[srcAddr]

handleB1 = "转发,登记"

Label(lbdata, text=srcAddr).place(x=65,y=20\*i)

Label(lbdata, text=S1[srcAddr]).place(x=125,y=20\*i)

dataChip[2 \* count + 1] = srcAddr

dataChip[2 \* count + 2] = S1[srcAddr]

if(desAddr in S1):

if(S1[desAddr] == hostSegOne[srcAddr]):

handleB1 = "登记,丢弃"

flagb2 = False

else:

handleB1 = "登记,转发"

routPort += "S1 - >"

else:

if(hostSegOne[srcAddr] == 1):

routPort += "S1 - >"

else:

routPort += "S1 - >"

else:

if(desAddr in S1):

if(S1[desAddr] == hostSegOne[srcAddr]):

handleB1 = "丢弃"

flagb2 = False

else:

handleB1 = "转发"

else:

handleB1 = "转发"

elif(seqS[count] == 1 and flagb2):

if(srcAddr not in S2):

S2[srcAddr] = hostSegTwo[srcAddr]

handleB2 = "转发,登记"

Label(lbdata, text=srcAddr).place(x=175,y=20\*i)

Label(lbdata, text=S2[srcAddr]).place(x=235,y=20\*i)

dataChip[2 \* count + 1] = srcAddr

dataChip[2 \* count + 2] = S2[srcAddr]

if(desAddr in S2):

if(S2[desAddr] == hostSegTwo[srcAddr]):

handleB2 = "登记,丢弃"

flagb1 = False

else:

handleB2 = "登记,转发"

routPort += "S2 - >"

else:

if(hostSegOne[srcAddr] == 1):

routPort += "S2 - >"

else:

routPort += "S2 - >"

else:

if(desAddr in S2):

if(S2[desAddr] == hostSegTwo[srcAddr]):

handleB2 = "丢弃"

flagb1 = False

else:

handleB2 = "转发"

else:

handleB2 = "转发"

if(srcAddr in lan2):

flagb1 = True

count = count + 1

#print(i)

routPort = routPort[:-4]

Label(lbdata, text=srcAddr + "->" + desAddr).place(x=0,y=20\*i)

Label(lbdata, text=handleB1).place(x=290,y=20\*i)

Label(lbdata, text=handleB2).place(x=380,y=20\*i)

Label(lbdata, text=routPort).place(x=470,y=20\*i)

dataChip[0] = srcAddr + "->" + desAddr

dataChip[5] = handleB1

dataChip[6] = handleB2

dataChip[7] = routPort

dataIO.append(dataChip)

i = i + 1

分别算出转发表S1和S2的数据和端口，然后分别通过源地址，目的端口，发送端口和目的地址判断是否“登记”、“丢失”、“转发”。

**2.读取数据函数：**

def tableRead():

#root.withdraw()

global i

global flag,srcAddr,desAddr

if(i > 7):

tkinter.messagebox.showinfo('提示', '请重置后再进行读取')

return

file\_path = filedialog.askopenfilename(filetypes = [('CSV', 'csv')])

if(file\_path):

flag = True #设置为从文件获取数据

info=pd.read\_csv(file\_path,encoding = 'gbk')

for j in info['发送的帧']:

if(i < 8):

srcAddr = j.split('-')[0]

desAddr = j.split('>')[1]

btnSend() #调用发送数据的函数

flag = False

读取选中文件中“发送的帧”列中的数据，循环调用发送事件的函数，通过重新发送数据将文件中的数据展示。读取的数据条数加上系统原有的数据最多不超过8条。由于读取过程是重新执行btnsend函数，因此不会出现读取前有重复数据而出错的情况。

**六、系统运行与调试结果**

**调试错误记录**

2019年1月17日11：28分调试修复

1、每次测试结果记录会进行保留，下一次结果保存会紧接前次结果后面。

2、出发地址校验功能正常；目的地址校验功能暂未解决，如目的地址错误，网桥转发表能进行转发。

解决方案：

每次在保存之后，对列表dataIO进行清空，解决保存追加的问题，并且加上与源地址类似的校验。

2019年1月21日15：30分调试修复

1、第一次输入出发地址和目的地址，点击发送，无误，第二次未更改出发地址，只更改目的地址，转发表S1、S2无标记无显示。

2、读取文件，重置后无法再次正常发送帧

3、文件数大于转发表最大容量，读取到转发表展示时，会重复弹出提示窗口

解决方案：

在读取文件之后，将转发信息获取方式为从输入框获取。在弹窗之前进行判断，设置最多只出现一次弹窗。



图18.*多次弹出提示窗口*

**七、小组分工说明**

**张宇**：逆向学习算法的实现，程序总体功能设计

**王含艺**：python的Tkinter库交互式界面布局设计和实现

**吴昊航**：程序框架、系统功能流程图、系统调试、报告撰写

**八、还存在的问题与今后的解决思路**

还存在的问题：界面无法动态生成

解决思路：

1. 到网上查阅资料，找一些基本图形，分别代表主机和网桥。
2. 添加Tkinter画布Canvas计算位置
3. 获取网络拓扑结构数据，将常量用变量替代
4. 获取转发情况，更新转发表

**九、参考文献**

【1】张晓明. 计算机网络课程设计. 北京理工大学出版社,2016 年 8 月

【2】 张晓明. 计算机网络教程（第 2 版）. 清华大学出版社,2017 年 9 月

【3 】吴功宜 等. 计算机网络高级软件编程技术. 清华大学出版社,2008 年 1 月

【python基础】

Tkinter小构件之canvas 画布：

http://www.cnblogs.com/libra-yong/p/6250183.html

TKinter之菜单：

<http://www.cnblogs.com/kongzhagen/p/6148737.html>

tkinter中frame布局控件：

https://www.cnblogs.com/anita-harbour/p/9315472.html

Python tkinter下拉列表框(Combobox)：

<https://blog.csdn.net/zwliang98/article/details/80092143>

Python Tkinter clearing a frame

https://stackoverflow.com/questions/15781802/python-tkinter-clearing-a-frame