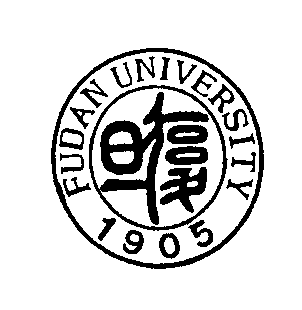
****

**Fudan Switch Simulator Tool**

**设计规格说明书**

目录

[1. 引言 3](#_Toc45219794)

[1.1. 系统目标 3](#_Toc45219795)

[1.2. 运行环境 3](#_Toc45219796)

[1.2.1. 术语定义 3](#_Toc45219797)

[1.2.2. 主要软件需求 3](#_Toc45219798)

[1.2.3. 设计约束/限制 3](#_Toc45219799)

[2. 体系结构设计 4](#_Toc45219800)

[2.1. DFD 4](#_Toc45219801)

[2.1.1. 交换机仿真系统顶层图 4](#_Toc45219802)

[2.1.2. 交换机仿真系统0层图 4](#_Toc45219803)

[2.1.3. 交换机仿真系统加工1子图 4](#_Toc45219804)

[2.1.4. 交换机仿真系统加工2子图 5](#_Toc45219805)

[2.1.5. 交换机仿真系统加工3子图 5](#_Toc45219806)

[2.2. 体系结构 5](#_Toc45219807)

[3. 数据设计 6](#_Toc45219808)

[3.1. 数据对象/数据结构 6](#_Toc45219809)

[3.2. 全局数据 7](#_Toc45219810)

[4. 模块设计说明 7](#_Toc45219811)

[4.1. 端口模块 7](#_Toc45219812)

[4.2. 转发器模块 8](#_Toc45219813)

[4.3. DMA模块 9](#_Toc45219814)

[5. 人机界面规格说明 9](#_Toc45219815)

[5.1. 用户接口 9](#_Toc45219816)

[5.1.1. 添加新交换机 9](#_Toc45219817)

[5.1.2. 添加新客户机 9](#_Toc45219818)

[5.1.3. 查看客户机节点 10](#_Toc45219819)

[5.1.4. 查看交换机节点 10](#_Toc45219820)

[5.2. 人机界面设计 10](#_Toc45219821)

[6. 运行设计 10](#_Toc45219822)

[6.1. 运行模块的组合 10](#_Toc45219823)

[6.2. 运行控制 10](#_Toc45219824)

[7. 出错处理设计 11](#_Toc45219825)

[7.1. MAC地址格式错误 11](#_Toc45219826)

[7.2. MAC地址 11](#_Toc45219827)

[7.3. 连接端口格式错误 11](#_Toc45219828)

[7.4. 重复连接 12](#_Toc45219829)

# 引言

## 系统目标

该系统的目标是实现一个图形化操作的以太网交换机仿真器，用户可以自行设置多台虚拟客户机与交换机组成交换网络，模拟交换网络的工作过程。

## 运行环境

硬件环境：处理器 1G Hz以上，内存（RAM）1G以上，磁盘空间10G以上。

软件环境：Windows 8以上版本操作系统。

### 术语定义

|  |  |
| --- | --- |
| **术语** | **描述** |
| MAC地址 | 一台设备的物理地址，每台设备的MAC地址是该网络中独一无二的。 |
| 冲突域 | 同一个网络上两个设备同时进行传输则会产生冲突，冲突域是会产生冲突的最小范围。 |
| 以太网 | 一种计算机局域网技术。在本系统中均指交换式以太网，与传统以太网相比，交换式以太网中交换机的各个端口隔离了冲突域，从而保证了各个端口的独立带宽。 |
| 以太网帧 | 数据链路层的协议数据单元。 |
| 以太网交换机 | 一种基于MAC地址识别，能够完成以太网帧封装与转发功能的网络设备。 |
| 端口 | 连接其他网络设备的接口。 |
| 转发器 | 处理转发逻辑的模块。 |
| DMA | 直接MAC地址表存取（Direct MAC address table Access），提供对MAC地址表的读写接口。 |

### 主要软件需求

交换机需要实现接收和解析数据帧、过滤和转发数据帧、维护MAC地址表的功能。此外，系统的响应速度需要足够快，能够支持包含较多节点的网络；系统需要能够稳定地工作，在异常时能够恰当处理而不致使计算机崩溃；系统的图形化界面需要简洁易用。

### 设计约束/限制

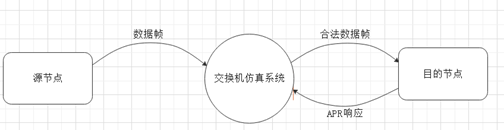
* 时间约束：开发过程在10天内完成。
* 预算约束：预算为0。
* 技术约束：
  + 开发环境：Windows 10操作系统。
  + 开发工具：后端使用Pycharm IDE，前端使用PyQt 5工具包。
  + 开发语言：Python 3.6。
  + 部署方式：单机部署。
* 人员约束：开发小组总共4人。

# 体系结构设计

## DFD

### 交换机仿真系统顶层图

交换机仿真系统的顶层图设计如下：

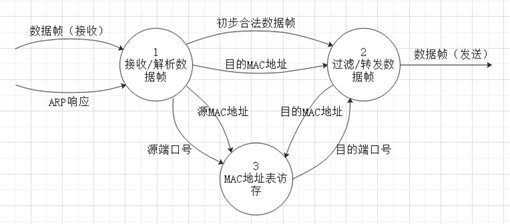


图：顶层图

交换机仿真系统可以接收从源节点发送过来的数据帧，并根据数据帧的目的MAC地址，将其转发给目的节点。目的节点在收到数据帧后，会单播一个ARP响应帧给交换机。

### 交换机仿真系统0层图

交换机仿真系统的0层图设计如下：

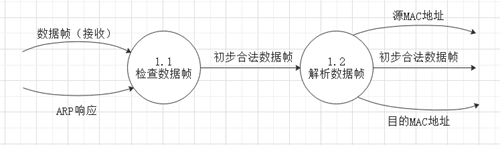


图：0层图

交换机仿真系统的功能主要由三部分组成：接收/解析数据帧，MAC地址表访存，过滤/转发数据帧。

### 交换机仿真系统加工1子图

交换机仿真系统的加工1子图设计如下：

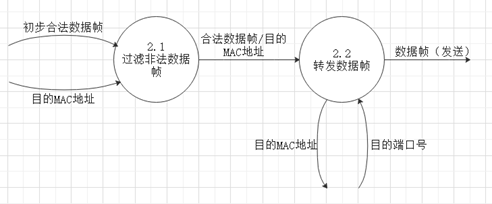


图：加工1子图

加功1主要负责接收/解析数据帧，分为两个子加工：检查数据帧和解析数据帧。前一个子加工负责接收源节点发送来的数据帧以及目的节点发送来的ARP响应帧，并根据不同的转发方式控制信号，对数据帧进行不同的检查，输出初步合法的数据帧给第二个子加工，由其负责解析数据帧，输出数据帧的源MAC地址，目的MAC地址。

### 交换机仿真系统加工2子图

交换机仿真系统的加工2子图设计如下：

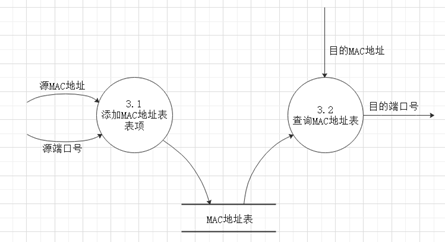


图：加工2子图

加工2主要负责过滤/转发数据帧，分为两个子加工：过滤非法数据帧和转发数据帧。前一个子加工负责过滤源MAC地址与目的MAC地址相同的数据帧，后一个子加工负责查询MAC地址表，并根据查询结果转发数据帧。

### 交换机仿真系统加工3子图

交换机仿真系统的加工3子图设计如下：

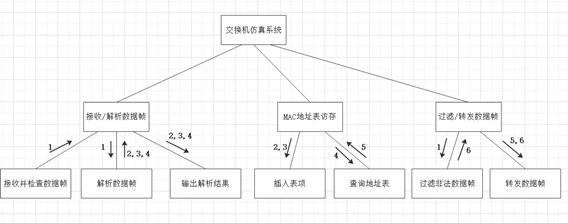


图：加工3子图

加工3主要负责MAC地址表的访存，分为两个子加工：添加MAC地址表表项和查询MAC地址表。前一个子加工负责将源数据帧的相关信息加入到MAC地址表中，建立起源MAC地址与源端口号之间的映射；后一个子加工主要负责根据输入的目的MAC地址，查询对应的端口号。

## 体系结构

由上面的DFD图，导出交换机仿真系统的结构图如下：



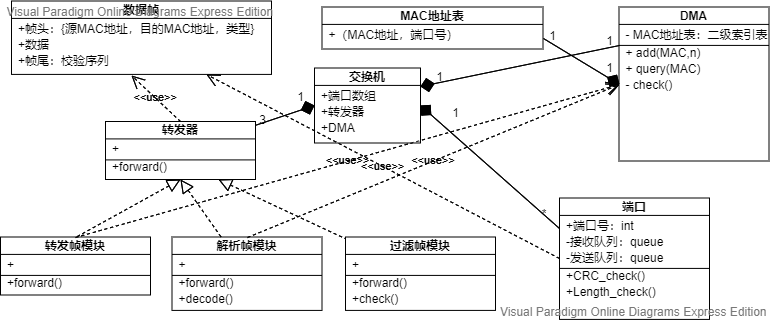
图：结构图

# 数据设计

## 数据对象/数据结构

|  |  |  |
| --- | --- | --- |
| 对象名称 | 数据结构 | 说明 |
| MAC地址表 | 二级索引表 | 由于MAC地址共有6个字节，前三个字节表示供货商，后三个字节表示序列号，因此可以将MAC地址表设计成二级索引表：外层索引是供货商号，内层索引是序列号，每个表项包含了MAC地址对应的端口号和该表项的老化时间。 |
| 端口缓存区 | 队列 | 包括了接受缓存队列和发送缓存队列，调度算法为FCFS。 |
| 端口 | 对象数组 | 一台交换机的端口号是固定的，因此可以用一个数组记录。每一个数组元素是一个自定义对象，数据包括了端口缓存区，函数包括了CRC校验函数，残帧检查函数。 |
| 转发器 | 自定义对象 | 转发器是交换机的核心部分，解析帧模块，过滤帧模块，转发帧模块都是转发器的实现，构成流水线结构。每个模块定义了其特有的功能。 |
| DMA | 自定义对象 | 如果让需要访问MAC地址表的转发帧模块和解析帧模块直接对MAC地址表进行操作，那么模块负担太重，成为流水线上的性能瓶颈。因此，在交换机中引入DMA，使MAC地址表对于交换机的其他部分透明。DMA的数据包括了MAC地址表；函数包括了添加表项函数，清理无效表项函数，查找表项函数。 |
| 数据帧 | 自定义对象 | 帧头包含了3个字段：前两个字段分别为6字节长的目的地址字段和源地址字段，第三个字段为2字节的类型字段，里面包含的信息用来标志上一层使用的是什么协议；数据部分包含了一个字段，长度不定；帧尾也只有一个字段，为4字节长，包含的信息是帧校验序列FCS（使用CRC校验） |

类图表示如下：



## 全局数据

全局控制信号transmitCtrl={直通转发，碎片隔离，存储转发}，控制端口对到来数据帧进一步响应。

# 模块设计说明

## 端口模块

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **模块名** | **设计者** | **模块所在文件** | **模块所在库** | **调用本模块的模块名** | **模块调用的其他模块名** |
| 端口 | 李纳川 | port.py | switch | Switch | 无 |
| **功能概述** | | | **处理过程概述** | | |
| 端口是交换机与外界交换的窗口。交换机的端口能实时监听并接收网络中的数据帧，并能根据控制信号transmitCtrl的不同选择进行存储转发、碎片隔离、直通操作。一个端口在同一时刻只接收一个数据帧。交换机的通过端口发送数据帧，一个端口在同一时刻只发送一个数据帧。 | | | 当端口检测的数据帧到达后，便会接受该数据帧。由于链路层具备冲突检测功能，所以不会出现多个帧同时到达的情况。接受数据帧后，若控制信号transmitCtrl=存储转发，便将数据帧发送到接受缓冲队列，并随后在CRC校验模块对数据帧进行循环冗余码校验，丢弃错误帧。若控制信号transmitCtrl=碎片隔离，则将数据帧放入残帧检查模块，丢弃帧长大于等于64字节的数据帧。若transmitCtrl=直通，则不进行额外处理，直接将数据帧发送到帧解析模块。  当某个转发帧从交换机内部到达端口时，会将其加入发送缓冲区。端口会实时检测发送缓冲区，若发送缓冲队列不为空，便将发送缓冲队列的队首数据帧出队，发送到外部网络。 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **内部接口** | | | | |
| **名称** | **意义** | **数据类型** | **有效范围** | **I/O标志** |
| DataReceived | 端口收到转发器发来的数据帧 | Array | transmitCtrl=直通，任意；  transmitCtrl=碎片隔离，>=64字节；  transmitCtrl=存储转发，64-1518字节。 | received=1,收到；received=0,未收到 |
| DataSent | 端口发送给转发器的数据帧 | Array | transmitCtrl=直通，任意；  transmitCtrl=碎片隔离，>=64字节；  transmitCtrl=存储转发，64-1518字节。 | received=1,收到；received=0,未发送。 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **外部接口** | | | | | |
| **名称** | **意义** | **数据类型** | **有效范围** | **I/O标志** | **格式** |
| FramesReceived | 端口收到发送数据帧的机器发来的数据帧 | Array | 64-1518字节 | FramesReceived=1,收到。  FramesReceived=0,未收到。 | Ethernet V2;  IEEE802.3/LLC |
| FramesSent | 端口发给接受数据帧机器的数据帧 | Array | 64-1518字节 | FramesSent  =1,收到。  FramesSent  =0,未收到。 | Ethernet V2;  IEEE802.3/LLC |

## 转发器模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模块名** | **设计者** | **模块所在文件** | | **模块所在库** | **调用本模块的模块名** | **模块调用的其他模块名** |
| 转发器 | 刘婧漪 | Forward.py | | switch | Switch | DMA |
| **功能概述** | | | **处理过程概述** | | | |
| 转发器是交换机的核心部分，其基本功能是向下一个交换机部件转发信息流。解析帧模块，过滤帧模块，转发帧模块都是转发器的实现，构成流水线结构。解析帧模块的功能是解析数据帧的源MAC地址和目的MAC地址；过滤帧模块的功能是过滤源MAC地址和目的MAC地址相同的数据帧；转发帧模块的功能是根据MAC地址表记录的信息，在相应的端口转发数据帧。 | | | 解析帧模块   * 输入：数据帧，到来端口 * 输出：数据帧，源MAC地址，目的MAC地址，到来端口 * 操作：按照数据帧格式，解析出帧中的源MAC地址，目的MAC地址，将数据帧，源MAC地址，目的MAC地址，帧的到来端口传递给交换机的过滤帧模块 ；同时，调用DMA的add接口更新MAC地址表。   过滤帧模块   * 输入：数据帧，源MAC地址adr1，目的MAC地址adr2，到来端口 * 输出：数据帧，目的MAC地址adr2，到来端口，数据帧 * 操作：判断adr1和adr2所处的网段是否相同，若相同，则丢弃该数据帧，否则将数据帧，adr2，帧的到来端口传递给交换机的转发帧模块。   转发帧模块   * 输入：数据帧，目的MAC地址，到来端口 * 输出：转发端口，目的MAC地址 * 操作：判断目的MAC地址类型（主机标识段是否为全1），如果是广播地址，则向除接收端口外，端口数组中记录的所有端口转发；如果目的地址如果不是广播地址，则调用DMA的query接口查找MAC地址表，找目的MAC地址对应的端口：找到了，通过目的MAC地址对应的端口转发。找不到，通过除到来端口外，端口数组中记录的所有端口转发。 | | | |

## DMA模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模块名** | **设计者** | | **模块所在文件** | **模块所在库** | **调用本模块的模块名** | **模块调用的其他模块名** |
| 直接MAC地址表存取 | 李泳桦 | | DMA.py | switch | Switch | 无 |
| **功能概述** | | **处理过程概述** | | | | |
| DMA是直接与存储器相连的控制模块，负责管理MAC地址的高效存储，支持添加MAC表项、查询目的MAC地址对应的端口号、定时清除老化项的功能。 | | 若调用add接口，则输入的MAC地址和端口号被加入MAC地址表，如果该表项已经存在，则重置其老化时间；若不存在，则添加一条表项，初始化老化时间为零。若调用query接口，则根据输入的MAC地址查询对应表项，然后输出端口号。MAC表内部具有高效的索引结构辅助功能的实现。模块中设有计时器，会定时检测老化时间超过某一阈值的表项，并自动删除该表项。 | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **意义** | **数据类型** | **有效范围** |
| add | 交换机需要调用add接口添加或更新MAC表项。 | 输入1：int  输入2：int | 输入1：0~2^{48}-1  输入2：0~MAX\_PORT\_NUM |
| query | 交换机需要调用query接口查询目标MAC地址的转发端口。 | 输入1：int  输出：int | 输入1：0~MAX\_PORT\_NUM  输出：0~0~2^{48}-1 |

# 人机界面规格说明

## 用户接口

### 添加新交换机

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **意义** | **数据类型** | **格式** |
| 添加新交换机接口 | 可以向网路中添加一台新的交换机 | 字符串 | 该接口接收一个参数，用来表示新交换机要连接到的端口号。格式为‘a:b’，其中a为已存在的某个交换机编号，b为此交换机的端口号，b∈[1,4]。 |

### 添加新客户机

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **意义** | **数据类型** | **格式** |
| 添加新客户机接口 | 可以向网络中添加一台新客户机 | 字符串 | 该接口接收两个个参数，第一个参数用来表示新客户机的MAC地址，格式为"xx:xx:xx:xx:xx:xx:xx"，其中x∈[a-zA-Z0-9]；第二个参数用来表示新客户机要连接到的端口号。格式为‘a:b’，其中a为已存在的某个交换机编号，b为此交换机的端口号，b∈[1,4]。 |

### 查看客户机节点

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **意义** | **数据类型** | **格式** |
| 查看客户机节点接口 | 可以查看客户机节点的数据流，包括接收，发送情况。 | 无 | 无，点击对应图标即可 |

### 查看交换机节点

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **意义** | **数据类型** | **格式** |
| 查看交换机节点接口 | 可以查看交换机节点的数据流，包括各个端口的接收，发送情况，以及MAC地址表。 | 无 | 无，点击对应图标即可 |

## 人机界面设计

本软件的人机界面设计如下：主界面上应该专门有一块地方用来放置计算机网络，同时界面上应有按钮可以控制添加交换机或添加客户机节点到网络中。计算机网络中每个节点都可以点击查看改节点的数据流信息，包括接收发送数据帧以及MAC地址表。

# 运行设计

## 运行模块的组合

1. transmitCtrl=直通，广播

运行模块：端口、转发器（解析帧模块、过滤帧模块、转发帧模块）。

1. transmitCtrl=直通，单播

运行模块：端口、转发器（解析帧模块、过滤帧模块、转发帧模块）、DMA（解析帧模块、转发帧模块、控制模块）。

1. transmitCtrl=碎片隔离，广播

运行模块：端口（残帧检查模块）、转发器（解析帧模块、过滤帧模块、转发帧模块）。

1. transmitCtrl=碎片隔离，单播

运行模块：端口（残帧检查模块）、转发器（解析帧模块、过滤帧模块、转发帧模块）、DMA（解析帧模块、转发帧模块、控制模块）。

1. transmitCtrl=存储转发，广播

运行模块：端口（CRC校验模块）、转发器（解析帧模块、过滤帧模块、转发帧模块）

1. transmitCtrl=存储转发，广播

运行模块：端口（CRC校验模块）、转发器（解析帧模块、过滤帧模块、转发帧模块）、DMA（解析帧模块、转发帧模块、控制模块）。

1. 静置交换机超过其自动老化时间

运行模块：DMA（控制模块）。

## 运行控制

1. transmitCtrl=直通，通讯方式：广播

端口接受到数据帧后直接将它发送到转发器的帧解析模块。在帧解析模块，交换机获取数据帧的类型，源MAC地址和目的MAC地址，随后进行过滤非法帧操作。在过滤帧模块会检查源MAC地址和目的MAC地址是否相同，若相同，则丢弃该数据帧，否则将数据帧、源MAC地址、目的MAC地址、帧的到来端口传递给转发器的转发帧模块。转发帧模块向除接收端口外所有端口转发

1. transmitCtrl=直通，通讯方式：单播

端口接受到数据帧后直接将它发送到转发器的帧解析模块。在帧解析模块，交换机获取数据帧的类型，源MAC地址和目的MAC地址，随后进行过滤非法帧操作。在过滤帧模块会检查源MAC地址和目的MAC地址是否相同，若相同，则丢弃该数据帧，否则将数据帧、源MAC地址、目的MAC地址、帧的到来端口传递给转发器的转发帧模块。转发帧模块将目的地址发送到DMA的转发帧模块，在此模块将目的MAC地址和帧的到来端口发送给DMA的解析帧模块，并遍历MAC地址表，查询目标地址项是否存在。若目标地址项存在，向转发器的转发帧模块返回该MAC地址映射到的端口号, 否则返回空值。DMA的解析帧模块将（目的MAC地址，帧的到来端口）添加进MAC地址表中，若已有该表项，则将它覆盖，同时将它的老化时间置零。

1. transmitCtrl=碎片隔离，通讯方式：广播

端口接受到数据帧后将其交给残帧检查模块，残帧检查模块检查帧长是否大于等于64字节，是的话便丢弃该数据帧，否则将数据帧发送到转发器的帧解析模块。随后的操作与transmitCtrl=直通，通讯方式：广播相同。

1. transmitCtrl=碎片隔离，通讯方式：单播

端口接受到数据帧后将其交给残帧检查模块，残帧检查模块检查帧长是否大于等于64字节，是的话便丢弃该数据帧，否则将数据帧发送到转发器的帧解析模块。随后的操作与transmitCtrl=直通，通讯方式：单播相同。

1. transmitCtrl=存储转发，通讯方式：广播

端口接受到数据帧后将其放入接收缓冲区并在CRC校验模块对数据帧进行CRC（循环冗余码校验）检查，处理残帧、超长帧。随后的操作与transmitCtrl=直通，通讯方式：广播相同。

1. transmitCtrl=存储转发，通讯方式：单播

端口接受到数据帧后将其放入接收缓冲区并在CRC校验模块对数据帧进行CRC（循环冗余码校验）检查，处理残帧、超长帧。随后的操作与transmitCtrl=直通，通讯方式：广播相同。

1. 静置交换机超过其自动老化时间

DMA以深度优先的方式遍历MAC地址表，找到老化时间超过设定的自动老化时间的表项，清除。若某一外层索引对应的所有表项都被删除，那么也将它从外层索引表中删除。

# 出错处理设计

## MAC地址格式错误

概述：当向网络添加新客户机时，输入的MAC地址不符合规定格式。

系统处理方法：弹出消息提示框，告知用户MAC地址不符合条件，同时告知用户正确的MAC地址格式。然后取消本次操作。

## MAC地址

概述：当向网络中添加新客户机时，输入的MAC地址已经存在。

系统处理方法：弹出消息提示框，告知用户MAC地址不允许重复，然后取消本次操作。

## 连接端口格式错误

概述：当向网络添加新客户机或新交换机时，输入的连接端口号不符合规定格式。

系统处理方法：弹出消息提示框，告知用户连接端口号不符合条件，同时告知用户正确的连接端口号格式。然后取消本次操作。

## 重复连接

概述：有多台设备（交换机/客户机）同时连接到同一个端口。

系统处理方法：弹出消息提示框，告知用户不允许重复连接，然后取消本次操作。