

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | IPv4分组收发与转发 | | | | | |
| 姓名 | 瞿久尧 | | 院系 | 计算学部 | | |
| 班级 | 2037101 | | 学号 | 120L022314 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物207 | | 实验时间 | 2022.10.21 | | |
| 实验课表现 | 出勤、表现得分(10) |  | 实验报告  得分(40) |  | 实验总分 |  |
| 操作结果得分(50) |  |
| 教师评语 | | | | | | |
|  | | | | | | |

****

|  |
| --- |
| 实验目的： |
| **IPv4分组收发：**  IPv4 协议是互联网的核心协议，它保证了网络节点（包括网络设备和主机）在网络层能够按照标准协议互相通信。IPv4 地址唯一标识了网络节点和网络的连接关系。在我们日常使用的计算机的主机协议栈中，IPv4 协议必不可少，它能够接收网络中传送给本机的分组，同时也能根据上层协议的要求将报文封装为 IPv4 分组发送出去。  本实验通过设计实现主机协议栈中的 IPv4 协议，让学生深入了解网络层协议的基本原理，学习 IPv4 协议基本的分组接收和发送流程。  另外，通过本实验，学生可以初步接触互联网协议栈的结构和计算机网络实验系统，为后面进行更为深入复杂的实验奠定良好的基础。  **IPv4分组转发：**  本实验需要将实验模块的角色定位从通信两端的主机转移到作为中间节点的路由器上，在 IPv4分组收发处理的基础上，实现分组的路由转发功能。  网络层协议最为关注的是如何将IPv4分组从源主机通过网络送达目的主机，这个任务就是由路由器中的 IPv4 协议模块所承担。路由器根据自身所获得的路由信息，将收到的IPv4分组转发给正确的下一跳路由器。如此逐跳地对分组进行转发，直至该分组抵达目的主机。IPv4 分组转发是路由器最为重要的功能。  本实验设计模拟实现路由器中的 IPv4 协议，可以在原有 IPv4 分组收发实验的基础上，增加 IPv4 分组的转发功能。对网络的观察视角由主机转移到路由器中，了解路由器是如何为分组选择路由，并逐跳地将分组发送到目的主机。本实验中也会初步接触路由表这一重要的数据结构，认识路由器是如何根据路由表对分组进行转发的。 |
| 实验内容： |
| **IPv4分组收发：**   1. 实现 IPv4 分组的基本接收处理功能   对于接收到的IPv4分组，检查目的地址是否为本地地址，并检查 IPv4 分组头部中其它字段的合法性。提交正确的分组给上层协议继续处理， 丢弃错误的分组并说明错误类型。   1. 实现 IPv4 分组的封装发送   根据上层协议所提供的参数，封装 IPv4 分组，调用系统提供的发送接口函数将分组发送出去。  **IPv4分组转发：**   1. 设计路由表数据结构   设计路由表所采用的数据结构。要求能够根据目的 IPv4 地址来确定 分组处理行为（转发情况下需获得下一跳的 IPv4 地址）。路由表的数据 结构和查找算法会极大的影响路由器的转发性能，有兴趣的同学可以深 入思考和探索。   1. IPv4 分组的接收和发送   对前面实验（IP 实验）中所完成的代码进行修改，在路由器协议栈的 IPv4 模块中能够正确完成分组的接收和发送处理。具体要求不做改变，参见“IP实验”。   1. IPv4 分组的转发   对于需要转发的分组进行处理，获得下一跳的 IP 地址，然后调用发送接口函数做进一步处理。 |
| 实验过程： |
| 1. **实验环境设置**      1. **IPv4分组收发** 2. **选择实验内容**      1. **流程图**   接收函数 stud\_ip\_recv() 流程图：  发送函数 stud\_ip\_Upsend() 流程图：   1. **数据报内容错误检测** 2. **版本号错误：**判断版本号是否为4        1. **头部长度错误**：判断头部长度是否大于等于5        1. **生存时间错误：**判断生存时间是否大于0        1. **目的地址错误：**判断目的地址是否是本机IPv4地址或广播地址        1. **头部校验和错误：**计算校验和并判断是否等于0xffff     每次对两个字节进行反码加运算：     1. **IPv4分组转发** 2. **选择实验内容**      1. **新建数据结构：map<unsigned int, unsigned int> routeTable**   map结构的路由表，第一位存的是目的地址dest，第二位存的是下一跳地址nexthop，映射关系为 。   1. **流程图**   路由表初始化函数stud\_Route\_Init()流程图：  路由增加函数stud\_route\_add()流程图：  C:\Users\jeffery\AppData\Local\Microsoft\Windows\INetCache\Content.Word\路由添加.png  路由转发函数stud\_fwd\_deal()流程图： |
| 实验结果： |
| 1. **IPv4分组收发** 2. **编译结果：**      1. **测试结果：**      1. **错误数据报具体数据** 2. **版本号错误**   错误版本号：3，应为：4     1. **头部长度错误**   错误头部长度：1，应为：大于等于5的整数     1. **TTL错误**   错误TTL：0，应为：大于0的整数     1. 目标地址错误   错误目标地址：192.165.89.51，应为：10.0.0.3     1. **校验和错误**   错误校验和：0x03E8，应为0x4AFC     1. **IPv4分组转发** 2. **编译结果：**      1. **执行结果：** |
| 问题讨论： |
| * 在存在大量分组的情况下如何提高转发效率？   在路由转发函数的实现中，时间消耗与路由表的查找效率关系密切，若使用遍历查找的数据结构来构造路由表，其查找的时间复杂度是，分组数量大时，效率低下。因此，可以使用基于hash表查找的数据结构构造路由器，减少查找时间，例如map结构。 |
| 心得体会： |
| 本次实验的主要难点在于数据报中各信息的截取及校验和的计算。C语言提供的位操作可以很方便地提取版本号、首部长、TTL、校验和等信息，而不需要遍历整个数据报进行查找。在计算校验和过程中，由于对反码运算不熟练，卡了一段时间，需要重新复习一下这些基础知识。 |