|  |  |
| --- | --- |
| 实验名称：ARP协议分析 | |
| 实验台号： | 实验时间：2024/1/2 |
| 实验小组： | |
| 实验目的：  •理解ARP的工作原理；  •了解ARP数据包的格式；  •掌握静态ARP的设置方法；  •理解主机之间的通信过程。 | |
| 实验环境说明：  wireshark  linux | |
| 实验过程、步骤（可另附页、使用网络拓扑图等辅助说明）及结果：  工作原理:   1. pc0想给pc1发送数据, 先从arp缓存表中根据pc1的ip地址查询mac地址，缓存命中转，否则转2 2. pc0发送arp请求报文，广播给本地的所有主机，等待arp响应报文 3. 本地所有主机收到报文，判断目标ip地址是否是本机ip不是则抛弃该报文，如果对的上则带上自己的ip回复一个arp响应报文 4. pc0收到来自pc1的响应报文，把pc1的mac地址和ip地址写入arp   缓存   1. 发送数据   数据包格式：  IMG_256  在ARP表建立前，主机并不知道目标MAC地址，所以在一开始的时候只能通过广播的方式将ARP请求包发送出去，处于同一局域网的主机都能接收到广播的数据包。所以一开始目标MAC地址是FF-FF-FF-FF-FF-FF，而以太网首部的帧类型是有多种，对于ARP数据包来说，其值为0x0806，对于IP数据报来说，其值为0x0800。  在ARP首部一开始的2个字节存储的是硬件类型，表示要知道目标网卡的硬件类型，其中，值为1表示以太网地址；接下来还有2字节的协议类型，其中，0x0800表示IP协议，其他还可能是ICMP/IGMP协议等；接下来有1个字节表示硬件地址长度，指出该报文中硬件地址的长度，对于以太网硬件类型，它的值为6；还有1字节的协议地址长度，如果是ARP协议、IP协议等，该值为4；ARP首部最后的op字段用于记录ARP操作的类型，分别是：ARP请求，其值为1，ARP应答，其值为2。对于ARP首部后面的四个字段分别是源MAC地址、源IP地址、目标MAC地址、目标IP地址。  ip neigh命令:  ip neigh子命令是ip命令的一部分，用于管理arp表。ip命令是是当前管理 Linux 操作系统网络最常用的命令之一。  用法    显示arp表    显示指定网卡的arp表    删除表项    arping 命令:  arping命令可以用来发送arp请求    wireshark过滤arp数据包    arp请求和响应的数据包大小都是42字节头以太帧头部的type值为0x0806指示这是arp类型的数据包另外我们还看的出来请求报文是广播的而响应报文是单播的。  请求报文中最重要的是target ip字段    告诉要找哪台主机，mac地址则是全置为1，说明这是一个广播报文。    响应报文中头部目标地址变成了请求主机的mac地址而源地址变成了路由器的mac地址  无理由arp  wireshark过滤无理由arp数据包    配置静态arp/手动更改arp表项 | |
| 实验总结（遇到的问题及解决办法、体会）：   1. linux平台下arp相关操作和windows平台不一致：学习ip命令和arping命令来替代windows下的arp实现的功能   今天，我深入学习了ARP（地址解析协议）的相关知识。ARP是一个在以太网中用于将32位的IP地址转换为MAC地址的协议。在深入了解其工作原理后，我意识到ARP在计算机网络中的重要性，它使得数据能够在网络中正确地路由。  在学习过程中，我主要关注了ARP请求和应答的工作机制。ARP请求是当一个设备想要知道另一个设备的MAC地址时发送的消息，而ARP应答则是被查询设备对请求的响应。理解这两个过程是理解ARP如何工作的关键。  此外，我还学习了如何在Linux操作系统中使用ip neigh和arping这两个命令来查看和操作ARP表。通过这些命令，我可以查看ARP缓存，添加或删除条目，甚至发送自定义ARP请求。  为了更好地理解ARP的实际运作，我使用Wireshark工具抓取了ARP数据包进行详细分析。Wireshark是一个强大的网络协议分析器，它可以帮助我直观地查看数据包的详细信息，从而更好地理解ARP数据包的构造和内容。  通过这次学习，我对ARP有了更深入的理解，也更加明白了它在日常网络操作中的重要地位。同时，我也意识到了实践操作对于理论知识吸收的重要性。理论知识是基础，但只有通过实际操作，我们才能真正掌握并运用这些知识。 | |
| 器材、工具领用及归还负责人： | 实验记录人： |
| 实验执笔人： | 报告协助人：（签名） |
| 小组成员签名：（签名） | |
| 验收人： | 成绩评定： |