路由转发实验

孙佳钰 2015K8009929051

2018 年 11 月 1 日

1 实验内容

路由表保存了网络到网关和端口的映射，本实验需要实现如下过程：

- 路由器转发流程：从数据包中提取出目的 IP 后，从本地路由表中寻找转发端口然后进行转发。
- ARP 协议与 ARP 缓存：在寻找到目的 IP 后，利用 ARP 协议将目的 IP 的 MAC 地址填充进去，然后进行数据包发送。
- ICMP 协议格式：当上两步发生错误时，需要用 ICMP 数据包发送错误信息；同时也要支持 ping 操作的数据包处理。

2 实验流程

由于代码太多，故报告中没有加入代码部分，完整代码可见附件。

2.1 路由器转发流程

1. 收到数据包并判断出该数据包为 IP 数据包后，将收到包的端口、数据包和数据包长度一同传入 $handle\_ip\_packet$ 中。在该函数中首先判断该数据包的目的地址，如果地址为收到包的端口的地址，则判断该包是否是 ping 请求，是则调用 $icmp\_send\_packet$ 向端口发送 ping 回答，否则释放包；若地址不是收到包的端口的地址，则调用 $ip\_forward\_packet$ 进行转发。

2. 在 $ip\_forward\_packet$ 函数中，首先利用 $longest\_prefix\_match$ 函数将目的地址进行最长前缀匹配查找，如果找不到路由表项，就调用 $icmp\_send\_packet$ 向端口发送目的网络不可达信息。

如果找到了，先判断该数据包的 ttl 是否大于 1，不大于 1 就丢掉并调用 $icmp_send_packet$ 向端口发送超时信息，否则就将 ttl 减 1，重新计算校验和。然后判断找到的路由表项的网关地址是否为 0，为 0 则表明数据包的目的 IP 地址就在下一网段中，即可直接将目的地址传入

iface_send_packet_by_arp 进行发送；不为 0 则表明数据包还需要经过转发，则应将网关地址传入 *iface_send_packet_by_arp* 进行发送。

3. 在 *longest_prefix_match* 函数中进行最长前缀匹配。先设定最大掩码为 0，返回结点为 NULL，然后对路由表进行遍历。对每一表项，如果其掩码大于最大掩码，并且其目的地址与掩码的与和传入的目的地址与掩码的与相等，则将最大掩码更新为该掩码，将返回结点更新为该结点。

2.2 ARP 协议与 ARP 缓存

ARP 协议负责 IP 地址与 MAC 地址之间的映射。

1. 收到数据包并判断出该数据包为 ARP 数据包后，将收到包的端口、数据包和数据包长度一同传入 *handle_arp_packet* 中。在该函数中首先判断数据包的目的 IP 是否是收到数据包的端口的 IP，不是则释放数据包。是的话再判断收到的是否是请求数据包，是则调用 *arp_send_reply* 进行回复；不是则表明是回复数据包，调用 *arpcache_insert* 将数据包中的源 IP 地址与源 MAC 地址的对应插入到 ARP 表中。

2. *arp_send_reply* 函数中，需要自己组建 ARP 数据包并发送。其中链路层首部中的源地址为收到数据包的端口的地址，目的地址为传入数据包中的源地址。ARP 首部中的源硬件地址和目的硬件地址与链路层首部中的相同；源协议地址为收到数据包的端口的地址，目的协议地址为收到数据包中的源协议地址。其余如类型等字段根据各自要求进行填充。组好后调用 *iface_send_packet* 进行发送。

3. *arp_send_request* 函数负责在查询 ARP 表失败时广播请求数据包。其中链路层首部中的源地址为收到数据包的端口的地址，目的地址为 FF:FF:FF:FF:FF:FF。ARP 首部中的源硬件地址与链路层首部中的相同，目的硬件地址为 00:00:00:00:00:00；源协议地址为收到数据包的端口的地址，目的协议地址为作为参数传入的 IP 地址。其余如类型等字段根据各自要求进行填充。组好后调用 *iface_send_packet* 进行发送。

4. *arpcache_lookup* 函数负责在 ARP 表中查询作为参数传入的 IP 地址是否存在 MAC 映射。存在则返回 1，不存在返回 0。

5. *arpcache_append_packet* 函数负责将传入的数据包插入到 ARP 等待队列中。先从现有的等待队列中查询是否有与目的 IP 相同的项，若有则可直接插入到该项数据包队列的队尾。若没有则要新建一项，IP 设置为传入包的目的 IP，重试次数设置为 0，然后将数据包插入到该项数据包队列，将该项插入到等待队列，然后调用 *arp_send_request* 发送请求数据包。

6. *arpcache_insert* 函数负责将查询到的 IP 地址和 MAC 地址的对应插入到 ARP 表中。首先看 ARP 表是否已满，未满可直接插入，已满则随机替换一项。然后到等待队列中查询是否有与目的 IP 相同的项，若有则将该项等待队列中的所有排队数据包逐一将 MAC 地址填入并调用 *iface_send_packet* 发送，然后将相应等待表项释放。

7. *arpcache_sweep* 函数在独立于查询发送等操作所在进程之外的进程中工作。首先遍历 ARP 表项，将插入超过一定时间的表项置为无效。然后遍历 ARP 等待队列，如果某队列项重试次数大于一定次数，则对队列中的所有数据包回复目的 IP 不可达的 ICMP 数据包，然后释放

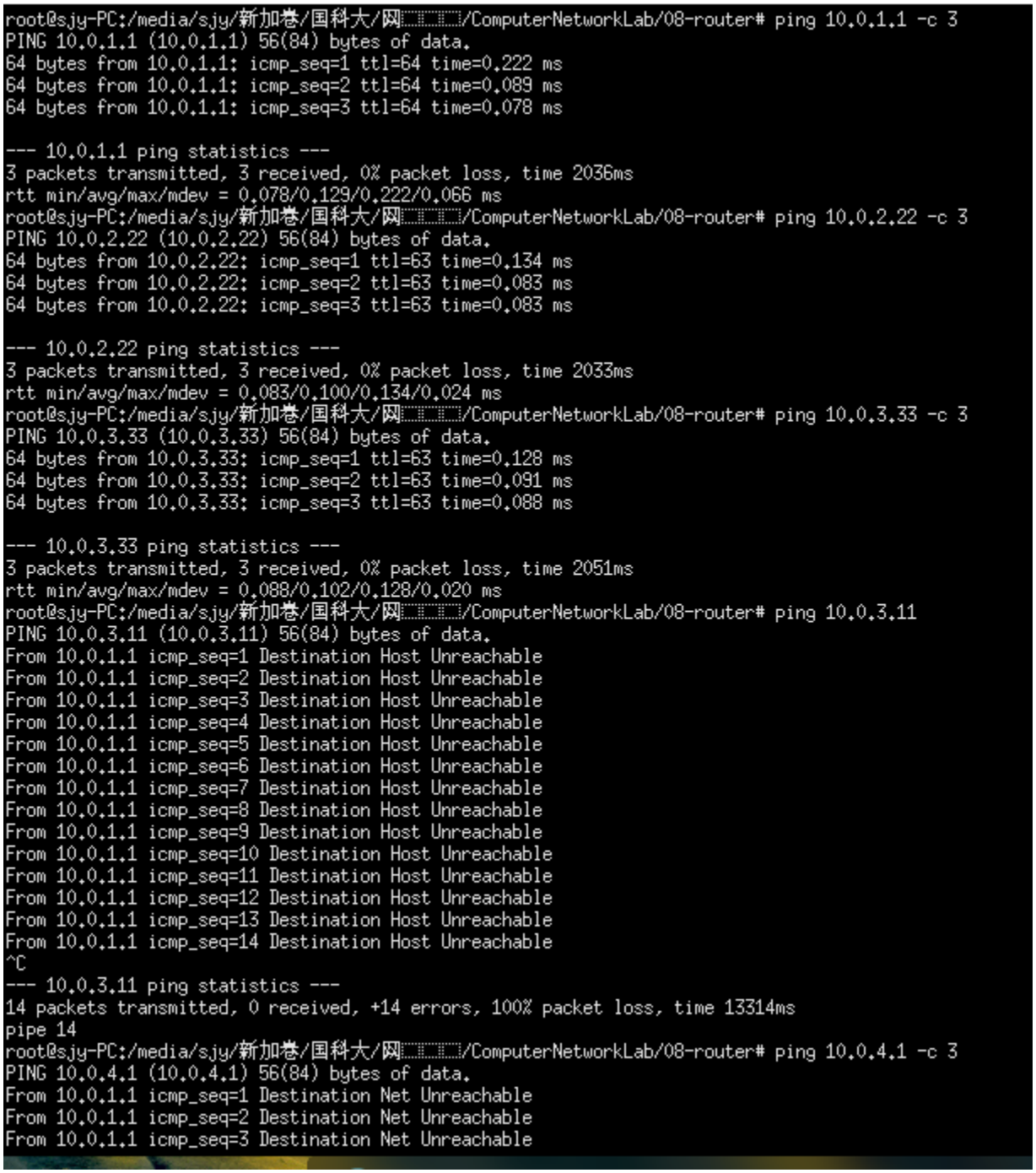
数据包、删除队列项；如果重试次数未达到一定次数，则调用 $arp\_send\_request$ 重新发送请求数据包，并将重试次数加 1。

2.3 ICMP 协议

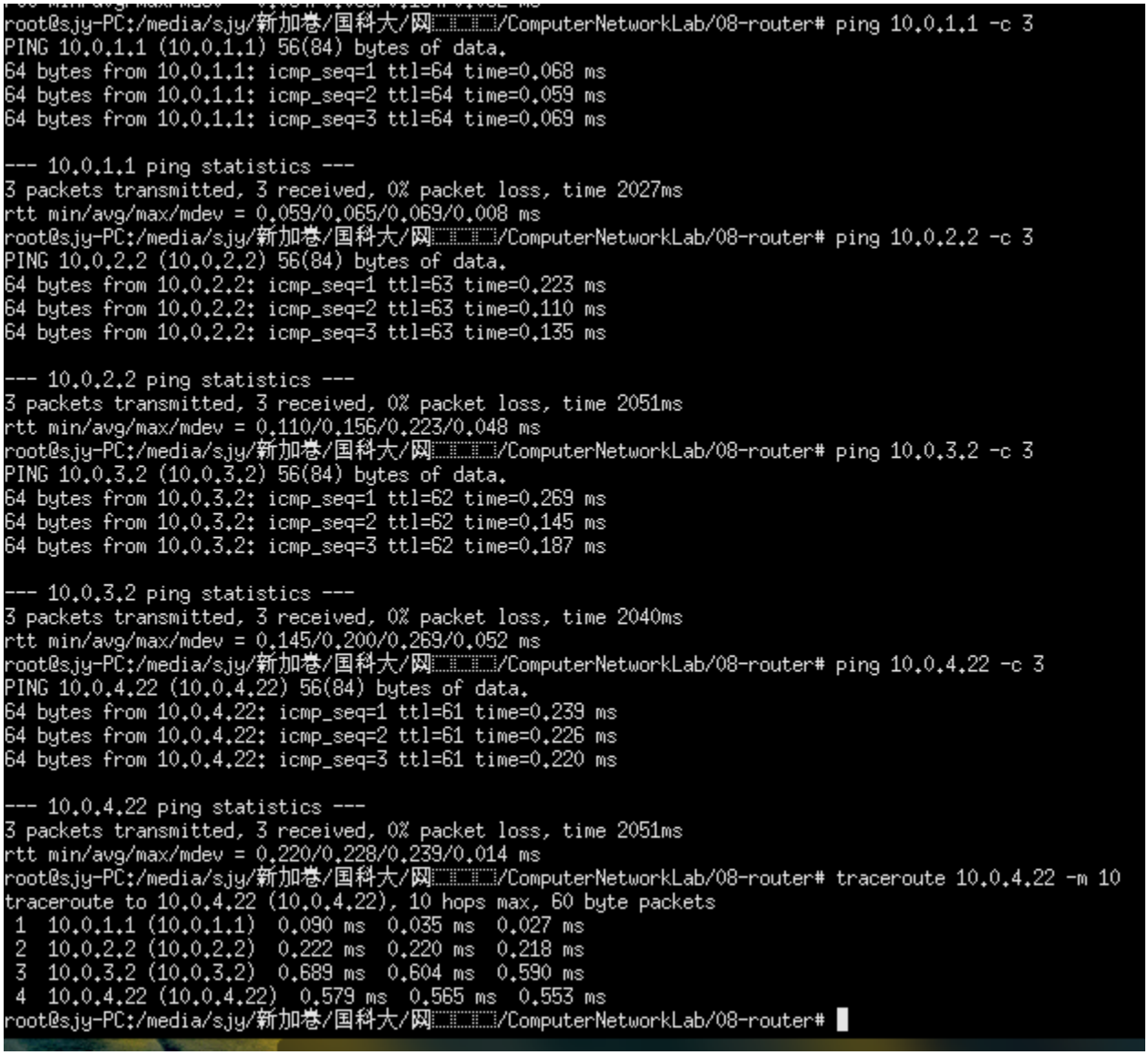
1. 在 $icmp\_send\_packet$ 函数中，需要自己组建 ICMP 数据包并进行发送。链路层首部中的的源地址为传入数据包中的目的地址，目的地址为传入数据包中的源地址。IP 首部中的源地址为接收数据包的端口的地址，目的地址为传入数据包中的源地址。其余如类型、长度、校验和等字段根据各自要求进行填充。将输入数据包释放后调用 ip_send_packet 发送组好的数据包。

2. ip_send_packet 函数是 IP 文件中的，但只在 ICMP 部分中使用，它负责发送本路由器产生的 ICMP 数据包。步骤与转发函数相似：首先利用 $longest\_prefix\_match$ 函数将目的地址进行最长前缀匹配查找，由于 ICMP 产生的数据包都发给接收到的数据包的发送 IP，所以肯定能找到路由表项。然后判断找到的路由表项的网关地址是否为 0，为 0 则表明数据包的目的 IP 地址就在下一网段中，即可直接将目的地址传入 $iface\_send\_packet\_by\_arp$ 进行发送；不为 0 则表明数据包还需要经过转发，则应将网关地址传入 $iface\_send\_packet\_by\_arp$ 进行发送。

3 实验结果及分析



上图是实验内容 1 的结果。



上图是实验内容 2 的结果，具体拓扑可见所附压缩包中文件 *my_topo.py*。