

ipv4 转发实验报告

计 21 2012011401 张梦豪

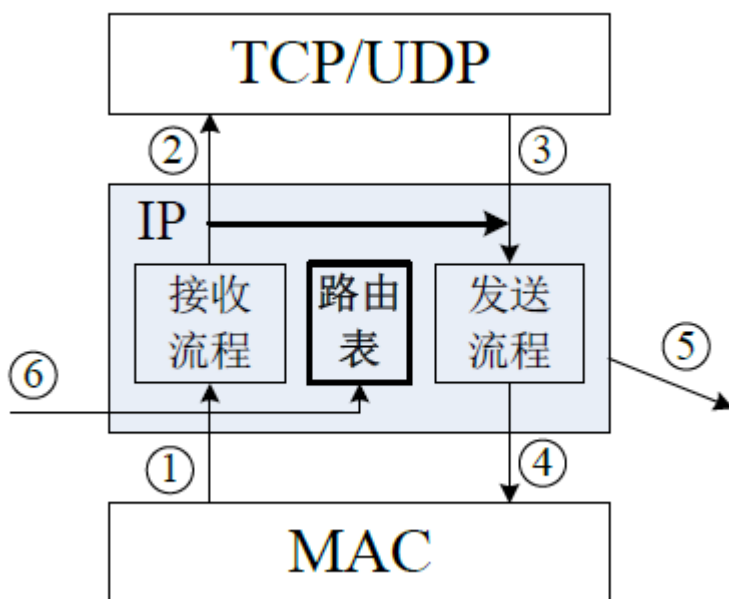
一. 实验目的

1. 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。
2. 初步接触路由表这一数据结构，认识路由器是如何根据路由表确定分组转发的下一跳。

二. 实验说明

分组转发是路由器最重要的功能。分组转发的依据是路由信息，以此将目的地址不同的分组发送到相应的接口上，逐跳转发，并最终到达目的主机。本实验要求按照路由器协议栈的 IPv4 协议功能进行设计实现，接收处理所有收到的分组（而不只是目的地址为本机地址的分组），并根据分组的 IPV4 目的地址结合相关的路由信息，对分组进行转发、接收或丢弃操作。

(1) 转发流程



如图，在下层接收接口函数 `Stud_fwd_deal()` 中（图中接口函数 1），实现分组接收处理。主要功能是根据分组中目的 IPv4 地址结合对应的路由信息对分组进行处理。分组需要上交，则调用接口函数 `Fwd_LocalRcv()`（图中接口函数 2）；需要丢弃，则调用函数 `Fwd_DiscardPkt()`（图中函数 5）；需要转发，则进行转发操作。转发操作的实现要点包括，TTL 值减 1，然后重新计算头校验和，

最后调用发送接口函数 `Fwd_SendtoLower()`（图中接口函数 4）将分组发送出去。注意，接口函数 `Fwd_SendtoLower()` 比前面实验增加了一个参数 `pNxtHopAddr`，要求在调用时传入下一跳的 IPv4 地址，此地址是通过查找路由表得到的。另外，本实验增加了一个路由表配置的接口（图 2.1 中函数 6），要求能够根据系统所给信息来设定本机路由表。

（2）路由表的设计

在本实验中分组接收和发送过程中都需要引入路由表的查找步骤。路由器的主要任务是进行分组转发，它所接收的多数分组都是需要进行转发的，而不像主机协议栈中 IPv4 模块只接收发送给本机的分组；另外，路由器也要接收处理发送给本机的一些分组，如路由协议的分组（RIP 实验中会涉及到）、ICMP 分组等。如何确定对各种分组的处理操作类型，就需要根据分组的 IPv4 目的地址结合路由信息进行判断。

一般而言，路由信息包括地址段、距离、下一跳地址、操作类型等。在接收到 IPv4 分组后，要通过其目的地址匹配地址段来判断是否为本机地址，如果是则本机接收；如果不是，则通过其目的地址段查找路由表信息，从而得到进一步的操作类型，转发情况下还要获得下一跳的 IPv4 地址。发送 IPv4 分组时，也要拿目的地址来查找路由表，得到下一跳的 IPv4 地址，然后调用发送接口函数做进一步处理。在前面实验中，发送流程中没有查找路由表来确定下一跳地址的步骤，这项工作由系统来完成了，在本实验中则作为实验内容要求学生实现。需要进一步说明的是，在转发路径中，本路由器可能是路径上的最后一跳，可以直接转发给目的主机，此时下一跳的地址就是 IPv4 分组的目地址；而非最后一跳的情况下，下一跳的地址是从对应的路由信息中获取的。因此，在路由表中转发类型要区分最后一跳和非最后一跳的情况。

路由表数据结构的设计是非常重要的，会极大地影响路由表的查找速度，进而影响路由器的分组转发性能。本实验中虽然不会涉及大量分组的处理问题，但良好且高效的数据结构无疑会为后面的实验奠定良好的基础。链表结构是最简单的，但效率比较低；树型结构的查找效率会提高很多，但组织和维护有些复杂，可以作为提高的要求。具体数据结构的设计，可以在实践中进一步深入研究。

三. 实验内容及实现思路

整个实验中我使用链表作为记录路由表的数据结构。

(1) `stud_Route_Init()`函数中, 对链表进行初始化。

(2) `stud_route_add()`函数中, 完成路由分组的增加, 把路由加入到链表的尾部。

(3) `stud_fwd_deal()`函数中, 首先判定是否为本机接收的分组, 如果是则调用 `fwd_LocalRcv()`; 然后判断分组是否超时, 如果超时则丢弃分组; 否则按照最长匹配在链表中查找路由分组, 并向相应接口进行转发, 这里需要注意 TTL 和校验和的变化, 查找失败则调用 `fwd_DiscardPkt()`。

具体实验代码见 `ipv4.cpp`。

四. 思考问题

(1) 转发处理中, 先判断目的地址是否为本机地址, 还是先对 TTL 进行处理, 在实现上有何不同。

解: 如果先判断目的地址是否为本机地址, 那么在后面需判断 $TTL \leq 0$ 则丢弃; 如果先对 TTL 进行处理, 那么需判断 $TTL < 0$ 丢弃, 然后在判断是否为本机地址。不同之处在于判断时是否包含 0。

(2) 调研路由表的数据结构有哪几种实现方式, 并比较这些实现方式的优缺点。

解: 路由表主要有链表和树两种实现方式。链表数据结构比较简单, 但是查询花费的时间较多; 树实现比较复杂, 但是查询效率相对较高。

五. 实验总结

这个实验总体比较简单, 书上的说明比较清楚, 通过这个实验, 我对 ipv4 分组头部格式及其转发过程有了更加清晰的认识, 同时也对路由表这个结构有了深刻的理解。