Ipv6 收发实验报告

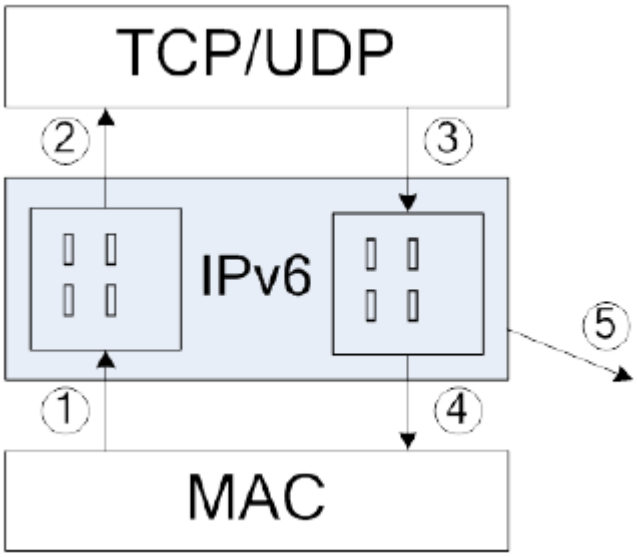
计 21 2012011401 张梦豪

一. 实验目的

通过设计实现主机协议栈中的 IPv6 协议，深入了解网络层协议的基本原理，学习 IPv6 协议基本的分组接收和发送流程。

二. 实验说明

（1）处理流程



客户端接收到测试服务器发送来的 IPv6 分组后，调用接收接口函数 stud_ipv6_recv( )（图中接口函数 1）。学生需要在这个函数中实现 IPv6 分组接收处理的功能。接收处理完成后，调用接口函数 ipv6_SendtoUp( ) 将需要上层协议进一步处理的信息提交给上层协议（图中接口函数 2）；或者调用函数 ipv6_DiscardPkt( )丢弃有错误的分组并报告错误类型（图中函数 5）。

在上层协议需要发送分组时，会调用发送接口函数 stud_ipv6_Upsend( )（图中接口函数 3）。学生需要在这个函数中实现 IPv6 分组封装发送的功能。根据所传参数完成 IPv6 分组的封装，之后调用接口函数 ipv6_SendtoLower( )把分组交给下层完成发送（图中接口函数 4）。

1） IPv6 分组接收流程

在接口函数 stud_ipv6_recv( )中，需要完成下列处理步骤（仅供参考）：

检查所接收到的 IPv6 分组头部的字段，包括版本号（Version）、有效载荷长度（Payload length）、跳数限制（Hop limit）字段。对于出错的分组调用 ipv6_DiscardPkt( )丢弃，并说明错误类型。

检查 IPv6 分组是否该由本机接收。如果分组的目的地址是本机地址或广播地址，则说明此分组是发送给本机的；否则调用 ipv6_DiscardPkt( )丢弃，并说明错误类型。

提取得到上层协议类型，调用 ipv6_SendtoUp( )接口函数，交给系统进行后续接收处理。

2） 发送流程

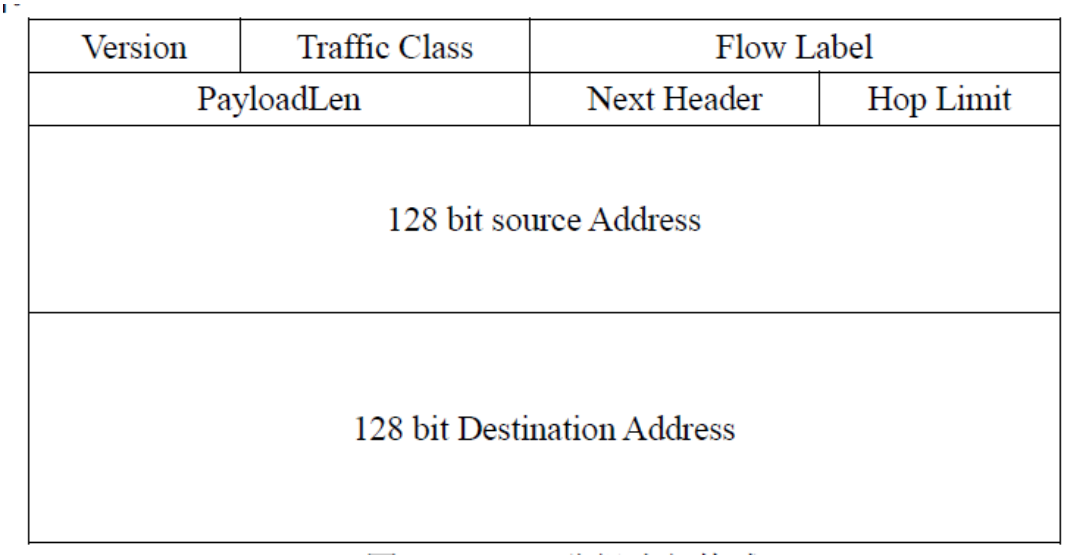
在接口函数 stud_ipv6_Upsend( )中，需要完成下列处理步骤（仅供参考）：

根据所传参数（如数据大小），来确定分配的存储空间的大小并申请分组的存储空间。

按照 IPv6 协议标准填写 IPv6 分组头部各字段。注意，各字段内容都要转换成网络字节序。

完成 IPv6 分组的封装后，调用 ipv6_SendtoLower( )接口函数完成后续的发送处理工作，最终将分组发送到网络中。

（2）IPv6 分组头部格式



- Version：4 位协议版本号，为 6。
- Traffic Class：传输类型，占 8 位。
- Flow Label：流量标签：占 20 位。

- Payload Length：16 位无符号整数，IPv6 载荷，也就是说跟在 IPv6 头部后面的部分，以 8 bits 为单位。
- Next Header：8 位，标识紧跟 IPv6 头部之后的头部类型。
- Hop Limit：8 位无符号整数，每经过一个节点减 1，减为 0 后该报文被丢弃。
- Source Address：源地址，报文的产生者。
- Destination Address：目的地址：报文的接收者。

三．实验内容及实现思路

（1）在 int stud_ipv6_recv(char *pBuffer, unsigned short length)函数中，首先检查所接收到的 IPv6 分组头部的字段，包括版本号（Version）和跳数限制（Hop limit）字段。对于出错的分组调用 ipv6_DiscardPkt( )丢弃，并说明错误类型。之后再检查 IPv6 分组是否该由本机接收，比对地址，如果分组的目的地址不是本机地址（IPv6 无广播地址），则说明此分组不是发送给本机的，丢弃并说明错误类型；否则调用 ipv6_SendtoUp( )接口函数，交给系统进行后续接收处理。

（2）在 int stud_ipv6_Upsend(char *pData, unsigned short len, ipv6_addr *srcAddr, ipv6_addr *dstAddr, char hoplimit, char nexthead)中，根据数据大小与 IPv6 头部大小之和（40Bytes）来确定分配的存储空间的大小并申请分组的存储空间。按照 IPv6 协议标准填写 IPv6 分组头部各字段，注意转换成网络字节序。完成 IPv6 分组的封装后，调用 ipv6_SendtoLower( )接口函数完成后续的发送处理工作，最终将分组发送到网络中。

具体实现见 ipv6.cpp。

四．思考问题

1） 比较 IPv6 收发实验与 IPv4 收发实验的异同。

解：两者的基本思路和处理流程是大致相同的，在分组接收流程中，都是先检查头部的合法性，然后再做是否接受的判断；在分组发送中，都是先进行头部的封装，然后在进行发送。

不同之处在于两个协议的细节不同，由于两个协议头部定义的不同，故检查的主要字段是不同的，ipv4 需要检查头部长度（IHL）和校验和（Header Checksum），而 ipv6 没有；ipv6 需要检查有效负荷长度（Payload Length）而

ipv4 没有；而且两者在做是否应该由本机接收时也是有区别的，ipv4 有广播地址而 ipv6 不存在广播地址，故 ipv6 不需要检查广播地址。

2） IPv6 头部有 128 位，分析以不同的单位（字节，字和双字）进行存储的特点。

解：IPv6 头部主要由以下单位组成：

版本（4 位）：IP 版本，设置为 6。

流量类型（8 位）：执行与 IPv4 头部中的服务类型相同的功能。

流标签（20 位）：用于标识一个流，其目的是：不需要在分组中进行深度搜索，路由器就能识别应该以类似方式处理的分组。字段由源设置，在转发路由上不应该被修改。

净荷长度（16 位）：因为头部长度固定为 40 字节，所以指明净荷长度就能确定这个分组的长度。

下一个头部（8 位）：本字段扩展了 IPv4 头部中协议号的功能。

跳数限制（8 位）：该字段类似于 IPv4 的 TTL。它定义了 IP 数据报文所能经过的最大跳数。

源 IPv6 地址（128 位）。

目的 IPv6 地址（128 位）。

五．实验总结

通过本次实验，让我对 ipv6 协议有了更加深刻的理解，也让我看到了 ipv6 与 ipv4 的不同之处以及它特殊的魅力，ipv6 比 ipv4 更加简洁，更加高效，更多的地址空间满足了人们日益高涨的地址需求，更简洁的头部信息大大提高了路由器的处理能力，总体感觉收获很大。