

# 《计算机网络协议开发》实验报告

## 第五次实验：重叠网络协议栈开发

姓名：吕玉龙

学号：191220076

2019级计算机科学与技术院/系

邮箱：1931015836@qq.com

时间：2022.5.16

## 一、实验目的

熟悉传输控制层信号协议的设计、实现。熟悉传输控制层协议是如何建立连接，以及如何解决信号数据包的丢失和损坏。

## 二、成品功能

完成了 pdf 中 5.3 所要求的所有基本功能，即简单重叠网络 SON 实现，可以完成验收展示。

## 三、实验内容

### 1) 邻居表

这次实验需要在四个不同的网络主机上运行，同时需要解析网络拓扑，所以首先需要能够解析本机的 ip 和主机名。上网搜寻资料后，主机名可以用 `gethostname()` 函数获得，根据主机名解析则需要如下的函数

```
int hostname_to_ip(char * hostname , char* ip)
{
    struct hostent *he;
    struct in_addr **addr_list;
    int i;
    if ((he = gethostbyname(hostname)) == NULL)
    {
        // get the host info
        perror("gethostbyname");
        return 1;
    }

    addr_list = (struct in_addr **) he->h_addr_list;
    for(i = 0; addr_list[i] != NULL; i++)
    {
        //Return the first one;
        strcpy(ip , inet_ntoa(*addr_list[i]) );
        return 0;
    }

    return 1;
}
```

有了必要的准备就可以解析拓扑文件了。这里使用了 `fscanf` 进行解析。将解析到的主机 ip 和主机 id 都存储到给定的结构体中。同时，邻居数也可以通过解析拓扑得到，这里我没有放在 `topology.c` 文件中。

```
//这个函数为邻居表中指定的邻居书照原样分配，并建立连接。如果分配成功，
int nt_addconn(nbr_entry_t* nt, int nodeID, int conn);

int get_neighbor();
```

## 2) 邻居之间的连接

节点号小的邻居等待节点号大的邻居连接时，需要判断是哪个节点到来的连接。所以这里需要用到 `accept` 函数中的参数，得到邻居的 ip 地址，然后和邻居表中存储的 ip 地址做比较，就可以得到邻居之间的套接字描述符了。

```
printf("%d\n", num_connect);
while(1){
    if(num_connect == 0) break;
    struct sockaddr_in ss;
    socklen_t slen = sizeof(ss);
    int fd = accept(listenfd, (struct sockaddr *) &ss, &slen);
    if (fd < 0) {
        printf("error connect");
    }
    printf("receive request\n");
    for(int i = 0; i < neighbor_num; i++){
        if(strcmp(inet_ntoa(ss.sin_addr), nt[i].nodeIP) == 0){
            num_connect--;
            nt[i].conn = fd;
        }
    }
}
printf("all connect\n");
}
```

`waitsip` 函数则等待来自 sip 的连接。连接建立之后，使用 `getpktToSend` 函数接收来自 sip 的包，判断是否需要转发的包或是路由更新报文。

```
while(1){
    if(getpktToSend(&pkt, &nextnode, sip_conn) > 0){
        pkt.header.src_nodeID = myid;
        if(nextnode == BROADCAST_NODEID){
            printf("broadcasting\n");
            for(int i = 0; i < neighbor_num; i++){
                if(sendpkt(&pkt, nt[i].conn) > 0){
                    printf("succeed sending to neighbour\n");
                }
            }
        }
        else{
            int next_son;
            for(int i = 0; i < neighbor_num; i++){
                if(nt[i].nodeID == nextnode){
                    next_son = nt[i].conn;
                }
            }
            if(sendpkt(&pkt, next_son) > 0){
                printf("succeed sending to neighbour\n");
            }
        }
    }
    else
        break;
    memset(&pkt, 0, sizeof(sip_pkt_t));
}
```

sip 和 son 通信使用的接收和发送的函数基本相同，不同的是发送包的结构，只要在 `seg` 发送函数上稍作修改即可。

## 3) SIP

sip 连接时使用本机的 ip 地址即可。