

《计算机网络协议开发》实验报告

第 9 次实验 移动 IP 协议实验

姓名：元玉慧

学号：101220151

10 级 计算机 系 4 班

邮箱：njucsyyh@gmail.com

时间：2013/04/28

一、实验目的

本实验分“角色”，即 MN（移动节点）、FA（外地代理）和 HA（家乡代理）实现移动 IP 的三个技术：代理搜索、注册和包传递。通过本实验，学生可以理解移动 IP 的工作原理，掌握移动 IP 的主要技术。

二、实验设计背景

定义：

移动 IP 技术是指移动用户离开原网络，在基于 IP 的不同网络链路中自由移动和漫游时，不需要修改原有的 IP 地址，仍能享有原网络中的一切权限和服务的技术。它以基本 IP 技术为基础，以让移动设备在不同的网络使用同一固定 IP 地址为手段，以对设备移动性的支持为目的，最终到达移动用户在各种网络中自由漫游的效果。

移动 IP 协议可以看作是一种路由协议，只有与传统意义上的路由协议 OSPF、RIP、BGP 相比，它具有特殊的功能，即将数据报路由到那些可能一直在快速地改变位置的移动节点上。移动 IP 在网络层加入了新的特性，使得移动节点在漫游到外地链路上时，运行在移动设备上的网络应用程序不会因链路的改变而中断，这种特性使得移动节点需要总是通过本地地址进行寻址和通信，利用移动 IP 技术可以保证网间切换对于上层应用透明，当移动节点改变其在互联网上的链路层接入点后，仍然可以保持所有正在进行的通讯。

功能实体：

--移动节点（MN）：可以将接入因特网的位置从一条链路切换到另一条链路上，而仍然保持所有正在进行的通信，并且只使用它的家乡地址（home address）的那些节点；

--家乡代理（HA）：是指位于移动节点家乡链路上的路由器。当移动节点离开家乡网络时，它负责截获所有发往移动节点家乡地址的数据包，并通过隧道将它们转发给移动节点，并且维护移动节点当前位置的信息；

--外地代理（FA）：是指位于移动节点所访问网络（外地链路）上的路由器，为注册的移动节点提供路由服务。它接受移动节点的家乡代理通过隧道发来的报文，进行拆封后发给移动节点。对于移动节点发出的报文，外地代理提供普通路由器的服务。

术语：

--家乡地址/家乡链路：移动节点的家乡地址是指永久分配给该节点的地址。当移动节点切换链路时，家乡地址并不改变，移动节点家乡地址的网络前缀决定了它的家乡链路，即移动节点的家乡链路就是与它的家乡地址具有相同网络前缀的链路。除了极少数特例，移动节点只用家乡地址

和别的节点通信，即移动节点发出的所有包的源 IP 地址都是它的家乡地址，它接收的所有包的目的 IP 地址都是它的家乡地址。

--隧道： 当一个数据包被封装在另一个数据包的净荷中进行传送时，所经过的路径称为隧道。

--转交地址/外地链路： 转交地址是指当移动节点不在家乡网络时，移动节点被赋予的用以反映出移动节点当前所在链路位置的临时 IP 地址。转交地址具有以下特征：转交地址与移动节点当前所在链路相关；当移动节点切换链路时，转交地址也随之改变；送往转交地址的数据包可以通过正常路由机制到达，而不需要用到移动 IP 的相关隧道；当移动节点与其对端节点通讯时，转交地址一般不作为数据包的源 IP 地址或目的 IP 地址使用。移动 IP 协议可以使用两种不同类型的转交地址：

--外地代理转交地址： 是移动节点所注册的外地代理的地址，这个外地代理必须至少有一个端口连接移动节点所在的外地链路。外地代理转交地址可以是外地代理的任一个 IP 地址，只要通过这个地址可以正常与家乡代理通信即可。因此，外地代理转交地址的网络前缀并不一定与外地链路的网络前缀相同，多个移动节点可以同时共用一个外地代理转交地址；

--配置转交地址： 是移动节点从外地链路获得的当地地址，移动节点将之与自己的一个物理网络接口建立关联。其网络前缀必须与移动节点当前所在的外地链路的网络前缀相同。当外地链路上没有外地代理时，移动节点通常采用这种转交地址，一个配置转交地址同时只能被一个移动节点使用；

转交地址是一个与移动节点连接的外地链路紧密相关的 IP 地址，它与移动节点所连接的外地链路最多只有一跳之隔。它要么是有一个端口在外地链路上的外地代理的 IP 地址，要么就是暂时分配给移动节点的一个端口的地址。当移动节点与外地链路相连时，家乡代理利用这个地址向移动节点传送数据包，即转交地址是连接家乡代理和移动节点的隧道的出口。

移动节点向家乡代理注册自己已经获得的转交地址，在注册过程中，如果链路上有外地代理，移动节点就向 FA 请求服务，FA 再将注册包中继给 HA。为保障网络通信的安全性，注册消息需要进行认证处理。

如果注册成功，家乡代理技术得到发往移动节点家乡地址的数据包，并根据移动节点注册到 HA 上的转交地址，通过隧道将这些数据包传送给转交地址，在转交地址处，原始数据包被从隧道中提取出来送给移动节点。

由移动节点发送的数据包，采用家乡地址作为源 IP 地址，使用外地网络上的路由器作为默认的路由器，发送的数据包将通过外地网络的路由器直接发送到通信对端，而无需隧道技术。

三、实验理论要点

移动 IP 协议的实现主要通过 3 个过程来完成，分别是代理发现，注册和包传递。

移动节点利用代理发现机制完成 3 个功能：判定当前连在家乡链路上还是外地链路上；检测它是否切换了链路；当链接在外地链路上时，得到一个转交地址。

代理发现有 2 条消息构成：

-1- 代理通告消息

移动代理利用这个消息向移动节点告知它们的存在，当一个节点在一条链路上被配置成家乡代理或者外地代理时，它就在这条链路上广播或者组播代理通告消息，这使得连到这条链路上的移动节点可以判定该链路上是否有代理存在；

- 2- 代理请求消息

移动节点也可以主动发送代理请求消息，这个消息的目的就是让链路上的所有代理立即发送一条代理通告消息；

代理通告消息和代理请求消息与 ICMP 路由器发现消息中定义的路由器通告消息和路由器请求消息非常相似，事实上，移动 IP 利用了这 2 个消息的格式，并对他们进行了一定的扩展。

代理请求消息：当家乡代理或者外地代理接收到这条消息后，它必须马上响应一条代理广播消息。为与别的 ICMP 消息区分，代理/路由器请求消息的类型域取值为 10；

代理请求消息的格式：

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
类型								编码								校验和															
保留																															

代理通告消息：由 IP 首部，ICMP 路由器通告消息和移动代理通告扩展以及其他一些可选扩展组成，本实验不考虑可选扩展，代理通告消息中的 IP 首部被移动节点用来判定它是链接在家乡链路上还是外地链路上，通告消息的类型域取值为 9，编码域取值为 16. 生存期域表明代理发送广播的频率，RFC 上建议以 1/3 的生存期为周期发送代理广播，地址数目与和地址表项大小域分别列出的路由器地址优先级对的数目，以及每对包含 32bit 字的数目，它始终为 2，如果该报文，的总长度比根据地质数目和地址表项大小计算的值要大，那么接收到的报文的其他部分就被认为是扩展部分，如果其中有一个扩展部分为移动代理通告扩展，那么接收到的这个报文就是代理通告报文，否则接收到的报文是 ICMP 路由器通告报文。

0								1								2								3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
类型								编码								校验和																	
地址数目								地址表项大小								生存期																	
路由器地址																																	
优先级																																	
.....																																	
类型								长度								顺序号																	
注册生存期																R	B	H	F	M	G	r	T	保留									
零或多个转交地址																																	

移动代理通告扩展的类型域值是 16，长度域则给出了该扩展域的字节数，不包括类型域和长度域本身，当一个代理重启时，它将顺序号域复位成 0，每发送一个代理通告报文就将序号加 1. 转交地址域列出外地代理的一个或者多个转交地址。

注册：

注册是移动节点通知其他家乡代理它当前的移动绑定，并要求家乡代理将发送到其家乡地址上的报文转发给它的过程。

注册的过程发生在代理发现之后，当移动节点发现它移动回家乡链路上时，就向家乡代理注销移动绑定，并开始像固定主机或者路由器那样进行通信，当移动节点发现它连在一条外地链路上时，它需要先得到一个转交地址，再进行注册。如果得到的是外地代理转交地址，则通过外地代理向家乡代理进行注册；如果得到的配置转交地址，则既可以直接向家乡大力注册，也可以通过外地代理进行注册；这 2 种注册方式都将通过在移动节点和家乡代理之间交换注册请求和注册应答报文来完成。

移动节点进行注册的主要目的是为了将它的转交地址告诉家乡代理，家乡代理可以用这个地址将数据包通过隧道送给移动节点。因此家乡代理必然存有一份移动节点家乡地址和转交地址的对应表，这张表称为绑定表，每一个表项称为绑定表项。注册过程就是新建、修改或删除家乡代理中移动节点的绑定表项的过程。一次注册只在一定的生存时间内有效，移动节点在生存时间过期之前应重新注册。

注册报文的类型域表明这条报文是注册请求还是注册应答，前者类型域取值为 1，后者类型域取值为 3.

注册请求消息：

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
类型								S	B	D	M	G	r	T	x	生存期															
家乡地址																															
家乡代理																															
转交地址																															
标识																															
扩展																															

注册应答消息：

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
类型								编码								生存期															
家乡地址																															
家乡代理																															
标识																															
扩展																															

外地代理如何处理注册请求

外地代理接收到注册请求之后，要对它进行一系列的有效性检查。如果其中有一项检查失败，外地代理就向移动节点发送一条注册应答报文，拒绝这次注册请求，注册应答的代码域给出了拒绝的原因。如果外地代理同一接收移动节点的注册请求，它就更新来访移动节点列表，并将该报文转发给移动节点的家乡代理。外地代理要将注册请求报文的 IP 头和 UDP 头完全剥去，再加上新的 IP、UDP 头后才送给家乡代理。新报文的目标 IP 地址从注册请求报文的家乡代理域中得到，源 IP 地址则为外地代理上发送这个分组的端口的 IP 地址。

在中继注册请求报文前，外地代理要记录下一些信息，用于向移动节点发送注册应答，以及在注册成功后卫移动节点路由 IP 分组。

外地代理要记录的信息包括源数据链路层地址、源 IP 地址、源 UDP 端口号、家乡代理地址、标志域以及请求注册的生存期等。

家乡代理如何处理注册请求

家乡代理在受到注册请求后，也会做一系列和外地代理相似的有效性检查，如果注册请求是无效的，家乡代理会向移动节点发送一条注册应答报文，其中的代码域将注明注册失败的原因。如果注册请求是有效的，且绑定表中没有此移动节点的绑定表项，那么家乡代理将为该移动节点创建一个新的绑定表项，并插入到绑定表中。如果绑定表中已存在此移动节点的信息，家乡代理将根据转交地址，移动节点的家乡地址，生存期值等信息对移动节点的绑定表项进行更新。随后家乡

代理将会根据注册请求包的要求，新建、重建或者是撤销到移动节点的转交地址隧道。另外，家乡代理还将为移动节点提供 ARP 代理服务。最后，家乡代理向移动节点发送注册应答，告知注册成功。注册应答消息中的源 IP 地址、目标 IP 地址、源 UDP 和目标 UDP 端口号只是将注册请求消息中的相应域按照源和目标交换一下。

外地代理如何处理注册应答

外地代理接收到注册应答后，将会对消息进行一系列的有效性检查。外地代理一旦发现应答是无效的，将会产生一个包含适当代码域的注册应答，并发送给移动节点。如果注册应答消息是有效的，外地代理对通过隧道发往移动节点的包进行拆封，并对移动节点发过来的包实行缺省路由器的功能。

移动节点如何处理注册应答

在移动节点接收到注册应答后，移动节点就开始对应答报文进行有效性检查。如果这条应答报文是有效的，那么移动节点就检查代码域，判断这次注册被家乡代理或者外地代理接收还是拒绝。如果代码域表示注册请求已被接收，那么移动节点就可以调整它的路由表，然后就可以开始通信或者继续先前的通信了。如果移动节点在规定的时间内没有收到注册应答，它就会重发注册请求，根据协议的设计原则，移动节点后一次重发的时间间隔要比前一次重发的时间间隔长约 2 倍，直到重发时间间隔达到预先设定的最大值，或者重发次数超过了预先设定的最大值为止。

包传递

当移动节点离开家乡链路后，它的家乡代理把发往移动节点的所有分组转发到移动节点的当前位置，家乡代理可以使用代理 ARP 或者其他有效方法截获发往移动节点的 IP 分组。对于每个截获的分组，家乡代理使用隧道技术把它们发送到移动节点的当前转交地址。如果转交地址是外地代理的 IP 地址，那么这个外地代理就是隧道的终端，此时外地代理从分组中解封出原始数据包，并转发给移动节点。如果移动节点使用配置转交地址作为转交地址，那么分组就将直接通过隧道发送到移动节点上。

当移动节点在外地链路上时，它必须依赖某种方法来确定数据包转发出去的路由器。如果移动节点是通过外地代理注册，代理通告报文里的相关信息为移动节点提供了 2 中选择路由器的方法。第一种方法是选择外地代理作为缺省路由器，它由代理周期发送的代理通告报文的源 IP 地址指明。它要求外地代理必须有能力为移动节点产生的数据包提供路由服务。第二种方法是选择代理中，路由器地址域中出现的任何路由器。当移动节点在外地链路注册了一个配置转交地址，并无需向外地代理注册是，它有 2 种方法选择缺省路由器。如果链路上有一台路由器发送

了 ICMP 路由器通告报文，那么移动节点就将通告报文中的路由器地址域所列出的任何地址作为路由器的地址，如果没有路由器通告报文，那么移动节点依靠它来得到配置转交地址的方法来获得路由器的地址，比如通过 DHCP 应答包得到缺省路由器地址。

三、实验设计要点

充分理解移动 IP 协议，了解它的 3 个“角色”的主要作用，能够实现以下功能：

-1-代理发现：移动节点功能的实现

根据系统提供的参数组装并发送代理请求报文，收到代理应答消息后，判断出该代理是外地代理还是家乡代理，如果是外地代理就提交转交地址（实验默认使用外地代理地址作为转交地址）

-2-注册：移动节点的功能的实现

根据系统提供的参数组装并发送注册请求消息，收到注册应答消息后，根据报文的内容，修改移动节点的路由表。

-3-注册：外地代理功能的实现

收到注册请求报文后，将报文重新重新组装，然后中继到家乡代理。从家乡代理收到注册应答后，判断本次注册是否成功，成功则需要修改外地代理的路由表，最后将该应答消息中继到移动节点。

-4-注册：家乡代理功能的实现

对于收到的注册请求报文，进行合法性的检查，如果合法，则修改它的绑定表，最后发送注册应答消息给外地代理。

-5-包传递：家乡代理功能的实现

如果收到的目的地址是移动节点家乡地址的数据包，查找绑定表，将该包进行 IP 封装，然后发送给外地代理，否则，查找路由表，进行正常的 IP 转发。

-6-包传递：外地代理功能的实现

如果收到目的地址是移动节点家乡地址的数据包，查找路由表，将该包进行解封装，然后发送给移动节点。如果收到；来自移动节点的数据包，需要转发该数据包。

三、实验内容

ICMP 请求消息报文格式：

代理请求报头信息格式（请求注册报文头部还有 IP 和 ICMP 首部）

为与其他 ICMP 消息区分，代理/路由器请求消息的类型域取值为 10，编码域取值为 0；


```

//8字节的ICMP请求消息
typedef struct ICMP_req{
    unsigned char type;
    unsigned char code;
    unsigned short checksum;
    unsigned int reserved;
}* icmpheader;

```

注册请求消息格式：

```

//注册请求消息
typedef struct Regi_req{
    unsigned char type;
    //特殊标志字段（略去）
    unsigned char TAG;
    unsigned short ttl;
    //家乡地址
    unsigned int MN_addr;
    //家乡代理
    unsigned int HA_addr;
    //外地代理
    unsigned int FA_addr;
    //标识
    unsigned int iden_low;
    unsigned int iden_high;
}* request;

```

注册应答消息：

```

//注册应答消息
typedef struct Regi_rep{
    unsigned char type;
    unsigned char code;
    unsigned short ttl;
    unsigned int MN_addr;
    unsigned int HA_addr;
    unsigned int iden_low;
    unsigned int iden_high;
}* reply;

```

校验和计算以及路由添加函数

```

unsigned short checksum(unsigned short * buf, unsigned short count){
|
//移动节点路由表添加函数
void MN_add(unsigned int dest, unsigned int mask, unsigned int nexthop, unsigned int if_no){
//家乡代理路由表添加函数
void HA_add(unsigned int dest, unsigned int mask, unsigned int nexthop, unsigned int if_no){
//外地代理路由表添加函数
void FA_add(unsigned int dest, unsigned int mask, unsigned int nexthop, unsigned int if_no){

```

【1】移动节点发送代理请求报文函数：

```
//移动节点发送代理请求报文函数
int stud_MN_icmp_send()
{
    icmpheader buf = new ICMP_req;
    memset(buf, 0, sizeof(ICMP_req));
    buf->type = 10;
    buf->code = 0;
    //结构体指针类型转换限制
    //ICMP报文的校验和字段计算方法是将高16位和低16位相加即可，需要交换字节序
    buf->checksum = 0xFFFF;
    buf->reserved = htonl(0);
    //发送ICMP请求报文，从ICMP首部开始，字节序是网络字节序。
    icmp_sendIpPkt((unsigned char*)buf, sizeof(ICMP_req));
    return 0;
}
```

【2】移动节点接收代理应答报文函数

```
//移动节点接收代理应答报文函数
int stud_MN_icmp_recv(char *buffer,unsigned short len)
{
    //ipheader buf = (ipheader)buffer;
    unsigned int host_addr = getHAAddress();
    unsigned int src_addr = ntohl(*(int*)(buffer+12));
    unsigned int dest_addr = ntohl(*(int*)(buffer+16));
    //收到家乡代理的应答
    if(src_addr == host_addr)
        return 2;
    //收到外地代理的应答
    else{
        submitCareofadd(src_addr);
        return 1;
    }
    return 0;
}
```

【3】移动节点发送注册请求报文函数

```
//移动节点发送注册请求报文函数
int stud_MN_send_Regi_req(char*pdata,unsigned char len,unsigned short ttl,unsigned int HA_addr,
    unsigned int FA_addr,unsigned int MN_Haddr,unsigned int iden_low,unsigned int iden_high)
{
    request buf = new struct Regi_req;
    memset(buf, 0, 24);
    buf->type = 1;
    buf->TAG = 0;
    buf->ttl = htons(ttl);
    buf->HA_addr = htonl(HA_addr);
    buf->FA_addr = htonl(FA_addr);
    buf->MN_addr = htonl(MN_Haddr);
    buf->iden_high = htonl(iden_high);
    buf->iden_low = htonl(iden_low);
    char *buffer = new char[24+len];
    memcpy(buffer, buf, 24);
    memcpy(buffer+24, pdata, len);
    //发送移动节点注册请求消息的系统函数
    MN_regi_sendIpPkt( (unsigned char*)buffer, len+24);
    return 0;
}
```

【4】移动节点接收注册应答报文

```
//移动节点接收注册应答报文函数
int stud_MN_recv_Regi_rep(char *pbuffer,unsigned short len,unsigned int if_no)
{
    reply buf = (reply)(pbuffer+28);
    if(buf->code != 1 && buf->code != 0)
        return 1;
    //注册成功后，默认路由 只需要指定下一跳就可以了
    //在哪里使用了默认网关？
    MN_add(0, 0, getFAHAddress(), if_no);
    return 0;
}
..
```

【5】外地代理接收注册请求报文

```
//外地代理接收注册请求报文函数
int stud_FA_recv_Regi_req(char *pbuffer,unsigned short len, unsigned int mask,unsigned int if_no)
{
    request buf = (request)(pbuffer+28);
    int HA_addr = ntohl(buf->HA_addr);
    int FA_addr = ntohl(buf->FA_addr);
    int MN_addr = ntohl(buf->MN_addr);
    mask_global = mask;
    port_global = if_no;
    FA_sendregi_req(pbuffer+28, len-28, HA_addr, FA_addr);
    return 0;
}
```

【6】外地代理接收注册应答报文函数

```
//外地代理接收注册应答报文函数
int stud_FA_recv_Regi_rep(char *pbuffer,unsigned short len)
{
    reply buf = (reply)(pbuffer+28);
    //ipheader head = (ipheader)pbuffer;
    unsigned int dest_addr = ntohl(*(int*)(pbuffer+16));
    int MNaddr = ntohl(buf->MN_addr);
    int FAaddr = getFAHAddress();
    if(buf->code==0 || buf->code==1){
        FA_add(MNaddr, mask_global, MNaddr, port_global);
        FA_sendregi_rep((pbuffer+28), len-28, MNaddr, FAaddr);
        return 0;
    }
    else{
        FA_sendregi_rep((pbuffer+28), len-28, MNaddr, FAaddr);
        return 1;
    }
}
..
```

【7】家乡代理接收注册请求报文函数

//家乡代理注册请求报文函数

```
int stud_HA_recv_Regi_req(char *pbuffer,unsigned short len,unsigned int mask,unsigned int if_no)
{
    request buf = (request)(pbuffer+28);
    int MN_addr = ntohl(buf->MN_addr);
    int HA_addr = ntohl(buf->HA_addr);
    int FA_addr = ntohl(buf->FA_addr);
    unsigned int src = ntohl(*(int*)(pbuffer+12));
    unsigned int dest = ntohl(*(int*)(pbuffer+16));

    if(MN_addr == getMNHAddress()){
        HA_add(MN_addr, mask, src, if_no);
        HA_regi_sendIpPkt(3, 0, src, dest, MN_addr, HA_addr);
        return 0;
    }
    else{
        HA_regi_sendIpPkt(3, 131, src, dest, MN_addr, HA_addr);
        return 1;
    }
}
```

【8】家乡代理接收发往移动节点的报文的函数

```
int stud_packertrans_HA_recv(char *pbuffer,unsigned short len)
{
    unsigned int dest = ntohl(*(int*)(pbuffer+16));
    unsigned int src = ntohl(*(int*)(pbuffer+12));

    unsigned int dest_in = ntohl(*(int*)(pbuffer+36));
    unsigned int src_in = ntohl(*(int*)(pbuffer+32));

    int HA_addr = getHAHAddress();
    int MN_addr = getMNHAddress();
    int FA_addr = getFAHAddress();
    unsigned char protocol = pbuffer[9];

    if(dest == MN_addr){
        if(protocol==4){
            if(dest == dest_in){
                if(src == FA_addr) { HA_DiscardPkt(pbuffer,len); return 1; }
                else{
                    *(int*)(pbuffer+16) = htonl(FA_addr);
                    HA_forward_ipv4packet(pbuffer,len);
                }
            }
            else
                HA_send_Encap_packet(pbuffer,len,FA_addr,HA_addr);
        }
        else
            HA_send_Encap_packet(pbuffer,len,FA_addr,HA_addr);
    }
    else{
        stud_HA_route_node* p = g_HA_route_table;
        while(p != NULL){
            if(dest == p->dest){
                break;
            }
            p=p->next;
        }
    }
}
```

```

        if (p == NULL) {
            HA_DiscardPkt(pbuffer, len);
            return 1;
        }
        else{
            pbuffer[8]--;
            *((short *) (pbuffer+10)) = 0;
            *((short *) (pbuffer+10)) = checksum((unsigned short *) (pbuffer), 20);
            HA_forward_ipv4packet(pbuffer, len);
        }
    }
    return 0;
}

```

【9】外地代理接收发往移动节点的封装报文的函数

```

int stud_packertrans_FA_recv(char *pbuffer, unsigned short len)
{
    unsigned char protocol = (pbuffer[9]);
    unsigned int HA_addr = getHAHAddress();
    unsigned int MN_addr = getMNHAddress();
    unsigned int dest = ntohl(*(int *) (pbuffer+16));
    unsigned int src = ntohl(*(int *) (pbuffer+12));

    //printf("\n Protocol: %d\n", protocol);
    if(src == HA_addr){
        if(protocol==4){
            FA_send_ipv4_toMN(pbuffer+20, len-20);
        }
        else
            FA_DiscardPkt(pbuffer, len);
    }
    else if(src == MN_addr){
        unsigned short check = checksum((unsigned short *) (pbuffer), 20);
        if(check == 0){
            pbuffer[8]--;
            *((short *) (pbuffer+10)) = 0;
            *((short *) (pbuffer+10)) = checksum((unsigned short *) (pbuffer), 20);
            FA_forward_ipv4packet(pbuffer, len);
        }
        else FA_DiscardPkt(pbuffer, len);
    }
    return 0;
}

```

四、实验结果及报文分析

-1-代理搜索，移动节点功能的实现

对应于 1, 2 号报文：

Protocol 部分是 0x01, 对应于 ICMP 代理请求消息；

1 号报文分析：移动节点发送代理请求消息 (type:10 code 0), 目的地址是组播地址：

244.0.0.2

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun...	192.168.0.9	224.0.0.2	ICMP	Router solicitation	10.1 代理搜索-移动节点功能的实现
2	Sun...	10.0.0.1	255.255.2...	ICMP	Mobile IP Advertisement	10.1 代理搜索-移动节点功能的实现
3	Sun...	192.168.0.9	10.0.0.1	UDP	UDP 4660 > 434	10.2 注册-移动节点功能的实现
4	Sun...	10.0.0.1	192.168.0.9	UDP	UDP 434 > 4660	10.2 注册-移动节点功能的实现

Ethernet II, Src: 00:0D:04:00:00:0A, Dst: 00:0D:01:00:00:0A
Version :4, Src: 192.168.0.9, Dst: 224.0.0.2
Internet Control Message Protocol
Type: 10 (Router solicitation)
Code: 0
Checksum: 0xFFFF5 [correct]
ICMP Data(4 bytes)

2 号报文分析：移动节点发布代理通告消息（type: 9 code:16），目的地址是广播地址：
255.255.255.255

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun...	192.168.0.9	224.0.0.2	ICMP	Router solicitation	10.1 代理搜索-移动节点功能的实现
2	Sun...	10.0.0.1	255.255.2...	ICMP	Mobile IP Advertisement	10.1 代理搜索-移动节点功能的实现
3	Sun...	192.168.0.9	10.0.0.1	UDP	UDP 4660 > 434	10.2 注册-移动节点功能的实现
4	Sun...	10.0.0.1	192.168.0.9	UDP	UDP 434 > 4660	10.2 注册-移动节点功能的实现

Ethernet II, Src: 00:0D:01:00:00:0A, Dst: 00:0D:04:00:00:0A
Version :4, Src: 10.0.0.1, Dst: 255.255.255.255
Internet Control Message Protocol
Type: 9 (Mobile IP Advertisement)
Code: 16
Checksum: 0xFFFF6 [correct]
ICMP Data(20 bytes)

-2- 注册-移动节点功能的实现

此功能对应于 3 号和 4 号报文

3 号报文：移动节点发送注册请求消息：

源地址：192.168.0.9

目的地址：10.0.0.1

类型：0x01

生存期：0x0010 是 16s

家乡地址是：192.168.0.9 (C0 A8 00 09)

家乡代理是：192.168.0.3 (C0 A8 00 03)

转交地址是：10.0.0.1 (0A 00 00 01)

标识和扩展字段：全 0

1	Sun...	192.168.0.9	224.0.0.2	ICMP	Router solicitation	10.1 代理搜索-移动节点功能的实现
2	Sun...	10.0.0.1	255.255.2...	ICMP	Mobile IP Advertisement	10.1 代理搜索-移动节点功能的实现
3	Sun...	192.168.0.9	10.0.0.1	UDP	UDP 4660 > 434	10.2 注册-移动节点功能的实现
4	Sun...	10.0.0.1	192.168.0.9	UDP	UDP 434 > 4660	10.2 注册-移动节点功能的实现

+	Ethernet II, Src: 00:0D:04:00:00:0A , Dst: 00:0D:01:00:00:0A
+	Version :4, Src: 192.168.0.9 , Dst: 10.0.0.1
+	User Datagram Protocol, Src Port: 4660, Dst Port: 434
+	Data(46 bytes)


```

0000 00 0D01 00 00 0A00 0D04 00 00 0A08 00 45 00
0010 00 4A00 00 00 00 FF11 F0F0 C0A8 00 09 0A00
0020 00 0112 3401 B200 36 00 00 01 00 00 10 C0A8
0030 00 09 C0A8 00 03 0A00 00 01 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00

```

4 号报文：移动节点接收到注册应答消息

源地址：10.0.0.1

目的地址：192.168.0.9

类型：0x03

编码：0x00(编码值为 0 或者 1 表示接受注册)

生存时间：0x0010 16s

家乡地址是：192.168.0.9 (C0 A8 00 09)

家乡代理是：192.168.0.3 (C0 A8 00 03)

标识和扩展字段：全 0

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun...	192.168.0.9	224.0.0.2	ICMP	Router solicitation	10.1 代理搜索-移动节点功能的实现
2	Sun...	10.0.0.1	255.255.2...	ICMP	Mobile IP Advertisement	10.1 代理搜索-移动节点功能的实现
3	Sun...	192.168.0.9	10.0.0.1	UDP	UDP 4660 > 434	10.2 注册-移动节点功能的实现
4	Sun...	10.0.0.1	192.168.0.9	UDP	UDP 434 > 4660	10.2 注册-移动节点功能的实现

+	Ethernet II, Src: 00:0D:01:00:00:0A , Dst: 00:0D:03:00:00:0A
+	Version :4, Src: 10.0.0.1 , Dst: 192.168.0.9
+	User Datagram Protocol, Src Port: 434, Dst Port: 4660
+	Data(42 bytes)


```

0000 00 0D03 00 00 0A00 0D01 00 00 0A08 00 45 00
0010 00 46 00 00 00 00 40 11 AFF5 0A00 00 01 C0A8
0020 00 09 01 B212 3400 32 00 00 03 00 00 10 C0A8
0030 00 09 C0A8 00 03 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00

```

-3-注册外地代理功能的实现

图中的 1, 2, 3, 4 号报文完成外地代理功能：

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun...	192.168.0.9	10.0.0.1	UDP	UDP 4660 > 434	10.3 注册-外地代理功能的实现
2	Sun...	10.0.0.1	192.168.0.3	UDP	UDP 4696 > 434	10.3 注册-外地代理功能的实现
3	Sun...	192.168.0.3	10.0.0.1	UDP	UDP 434 > 4696	10.3 注册-外地代理功能的实现
4	Sun...	10.0.0.1	192.168.0.9	UDP	UDP 434 > 4660	10.3 注册-外地代理功能的实现
5	Sun...	10.0.0.1	192.168.0.3	UDP	UDP 4665 > 434	10.4 注册-家乡代理功能的实现
6	Sun...	192.168.0.3	10.0.0.1	UDP	UDP 434 > 4665	10.4 注册-家乡代理功能的实现

1 号报文是外地代理收到了移动节点发来的注册请求消息：

源地址：192.168.0.9

目的地址：10.0.0.1

类型：0x01

生存期：0x0010 是 16s

家乡地址是：192.168.0.9 (C0 A8 00 09)

家乡代理是：192.168.0.3 (C0 A8 00 03)

转交地址是：10.0.0.1 (0A 00 00 01)

标识和扩展字段：全 0

+

Ethernet II, Src: 00:0D:01:00:00:0A , Dst: 00:0D:03:00:00:0A

+

Version :4, Src: 192.168.0.9 , Dst: 10.0.0.1

+

User Datagram Protocol, Src Port: 4660, Dst Port: 434

●

Data(46 bytes)

0000

00 0D 03 00 00 0A 00 0D 01 00 00 0A 08 00 45 00

0010

00 4A 00 00 00 00 40 11 AF F1 C0 A8 00 09 0A 00

0020

00 01 12 34 01 B2 00 32 00 00 01 00 00 10 C0 A8

0030

00 09 C0 A8 00 03 0A 00 00 01 00 00 00 00 00 00

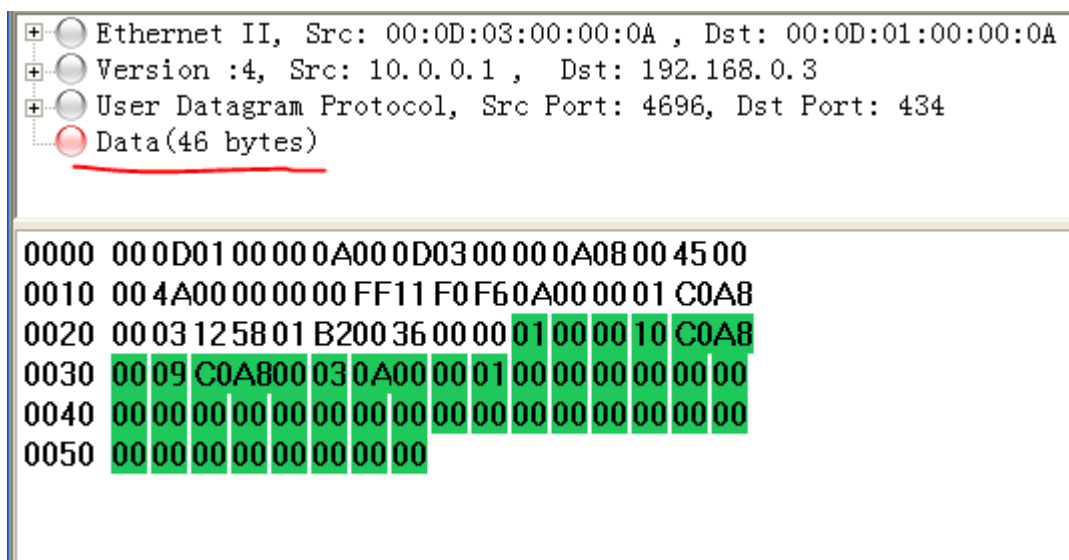
0040

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0050

00 00 00 00 00 00 00 00

2 号报文报文内容和 1 号报文相同，只是将注册请求消息转发给了家乡代理（目的地址改变为家乡代理地址：192.168.0.3）



3 号报文是家乡代理收到了家乡代理节点发来的注册应答消息：

源地址：192.168.0.3

目的地址：10.0.0.1

类型：0x03 表明报文是注册应答消息

Code: 64-88 表示外地代理注册失败代码

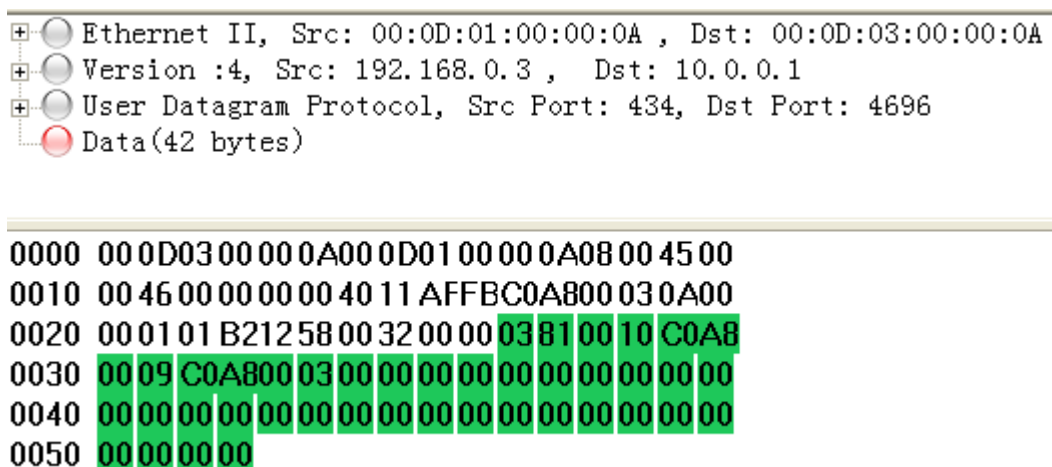
128-136 表示家乡代理注册失败代码

所以 0x81 表示家乡代理失败代码，家乡代理主机不可达；

生存时间：0x10

家乡地址：192.168.0.9 (C0 A8 00 09)

家乡代理地址：192.168.0.3 (C0 A8 00 03)

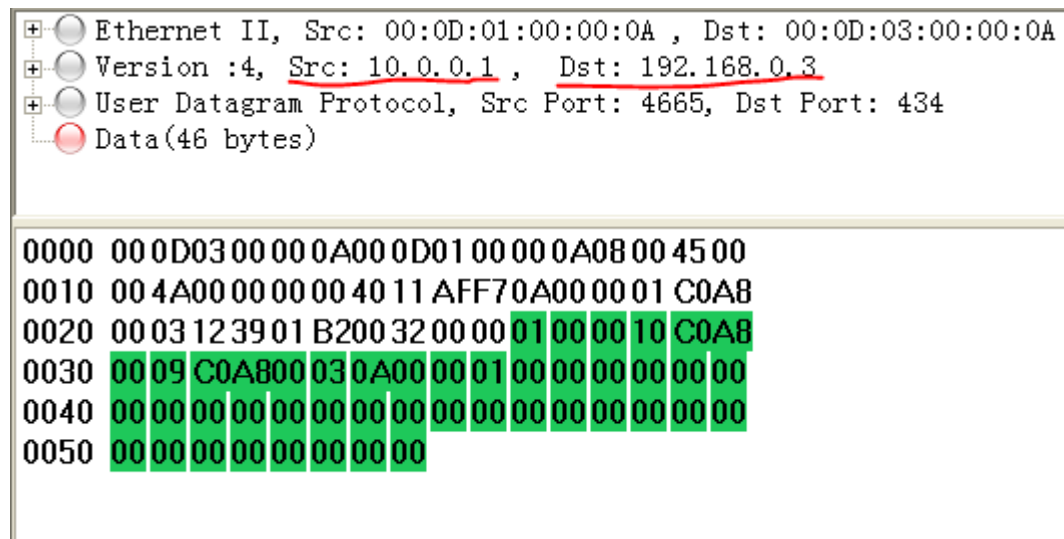


4 号报文的 data 内容与 3 号报文相同，只是将注册应答消息转发给了移动节点（目的地址是：192.168.0.9）

-4-注册家乡代理功能的实现

对应于 5, 6 号报文：

5 号报文：家乡代理收到了注册请求报文；



Wireshark packet capture for packet 5. The packet list shows Ethernet II, Version 4, User Datagram Protocol, and Data (46 bytes). The packet details pane shows the following fields:

- Ethernet II, Src: 00:0D:01:00:00:0A, Dst: 00:0D:03:00:00:0A
- Version :4, Src: 10.0.0.1, Dst: 192.168.0.3
- User Datagram Protocol, Src Port: 4665, Dst Port: 434
- Data(46 bytes)

The packet bytes pane shows the following hex data:

```
0000 000D0300000A000D0100000A08004500
0010 004A000000004011AFF70A000001C0A8
0020 0003123901B20032000001000010C0A8
0030 0009C0A800030A000001000000000000
0040 00000000000000000000000000000000
0050 0000000000000000
```

源地址：10.0.0.1

目的地址：192.168.0.3

类型：0x01

生存期：0x0010 是 16s

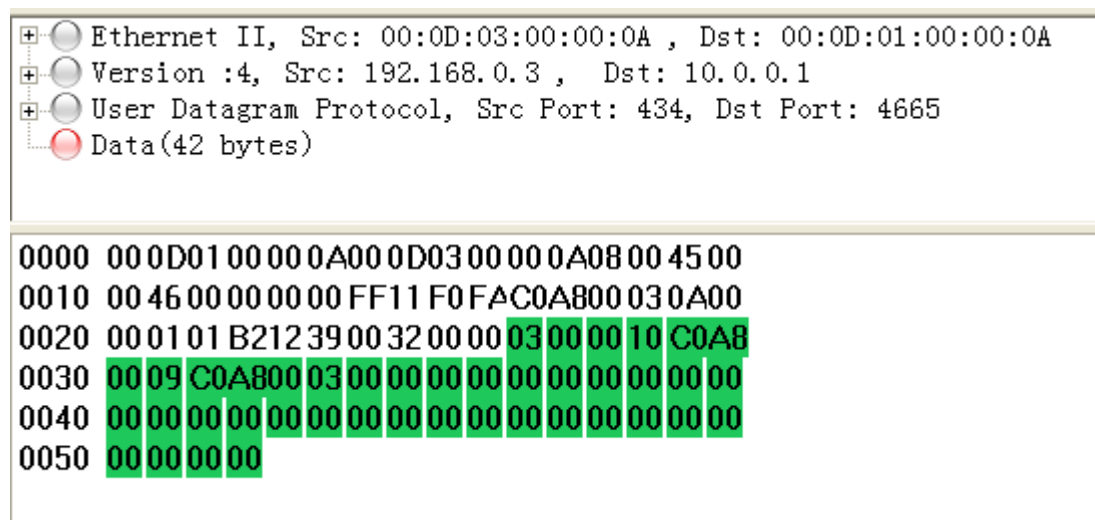
家乡地址是：192.168.0.9 (C0 A8 00 09)

家乡代理是：192.168.0.3 (C0 A8 00 03)

转交地址是：10.0.0.1 (0A 00 00 01)

标识和扩展字段：全 0

6 号报文：家乡代理发送注册应答消息



Wireshark packet capture for packet 6. The packet list shows Ethernet II, Version 4, User Datagram Protocol, and Data (42 bytes). The packet details pane shows the following fields:

- Ethernet II, Src: 00:0D:03:00:00:0A, Dst: 00:0D:01:00:00:0A
- Version :4, Src: 192.168.0.3, Dst: 10.0.0.1
- User Datagram Protocol, Src Port: 434, Dst Port: 4665
- Data(42 bytes)

The packet bytes pane shows the following hex data:

```
0000 000D0100000A000D0300000A08004500
0010 004600000000FF11F0FAC0A800030A00
0020 000101B212390032000003000010C0A8
0030 0009C0A8000300000000000000000000
0040 00000000000000000000000000000000
0050 00000000
```

源地址：192.168.0.3
目的地址：10.0.0.1
类型：0x03
生存期：0x0010 是 16s
家乡地址是：192.168.0.9 (C0 A8 00 09)
家乡代理是：192.168.0.3 (C0 A8 00 03)
转交地址是：10.0.0.1 (0A 00 00 01)
标识和扩展字段：全 0

-5-包传递-家乡代理功能的实现

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun...	20.0.0.2	192.168.0.9	UDP	UDP 17667 > 13314	10.5 包传送-家乡代理功能的实现
2	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3, Dst: 10.0.0.1	10.5 包传送-家乡代理功能的实现
3	Sun...	11.0.0.2	192.168.0.9	IP	Version 4, Src: 11.0.0.2, Dst: 192.168.0.9	10.5 包传送-家乡代理功能的实现
4	Sun...	11.0.0.2	10.0.0.1	IP	Version 4, Src: 11.0.0.2, Dst: 10.0.0.1	10.5 包传送-家乡代理功能的实现
5	Sun...	11.0.0.2	192.168.0.9	IP	Version 4, Src: 11.0.0.2, Dst: 192.168.0.9	10.5 包传送-家乡代理功能的实现
6	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3, Dst: 10.0.0.1	10.5 包传送-家乡代理功能的实现
7	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3, Dst: 10.0.0.1	10.6 包传送-外地代理功能的实现

测试的报文由 6 条，主要选取其中的正常封装转发的 2 条报文进行分析：
报文 5 是：家乡代理收到一个目的地址为移动节点 192.168.0.9 的报文；(data 为 42 字节)
报文 6 是：将报文 2 封装后的报文，加了一个 IP 首部，(data 为 62 个字节) 目的地址填为外地代理的地址，12 号报文的 data 部分其实就是 11 号报文的 IP 报文。

Ethernet II, Src: 00:0D:01:00:00:0A , Dst: 00:0D:03:00:00:0A

Version :4, Src: 11.0.0.2 , Dst: 192.168.0.9

Data(42 bytes)

0000 000D0300000A000D0100000A08004500

0010 003E000000004004AF090B000002C0A8

0020 00094500002A000000004000AE250C00

0030 0002C0A8000500000000000000000000

0040 00000000000000000000000000000000

☒ Ethernet II, Src: 00:0D:03:00:00:0A , Dst: 00:0D:01:00:00:0A
☒ Version :4, Src: 192.168.0.3 , Dst: 10.0.0.1
☒ Data(62 bytes)

```

0000 000D0100000A000D0300000A08004500
0010 0052000000004004AFFCC0A800030A00
0020 00014500003E000000004004AF090B00
0030 0002C0A800094500002A000000004000
0040 AE250C000002C0A80005000000000000
0050 00000000000000000000000000000000
  
```

封装后的 IP 报文头的协议部分字段是 0x04

6	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3
7	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3

☐ Flags: 0
☐ Fragment offset: 0
☐ Time to live: 64
☒ Protocol: IP (0x04)
☐ Header checksum: 0xAFFC [correct]
☐ Source: 192.168.0.3

```

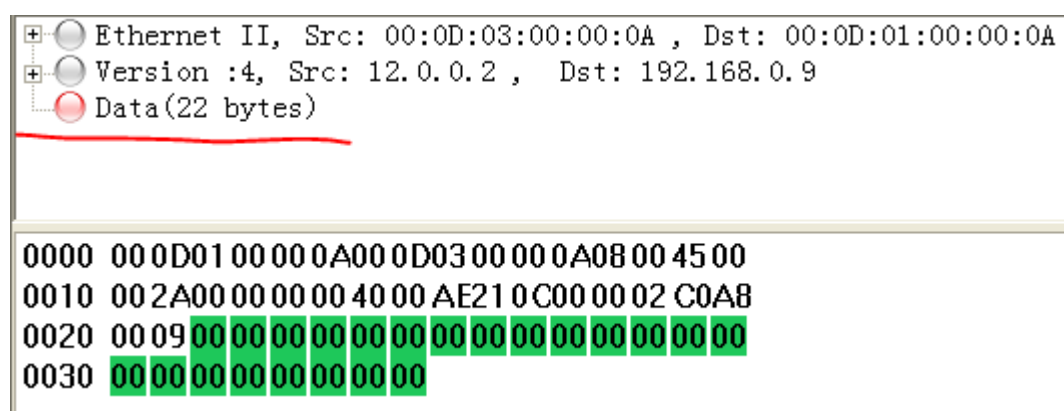
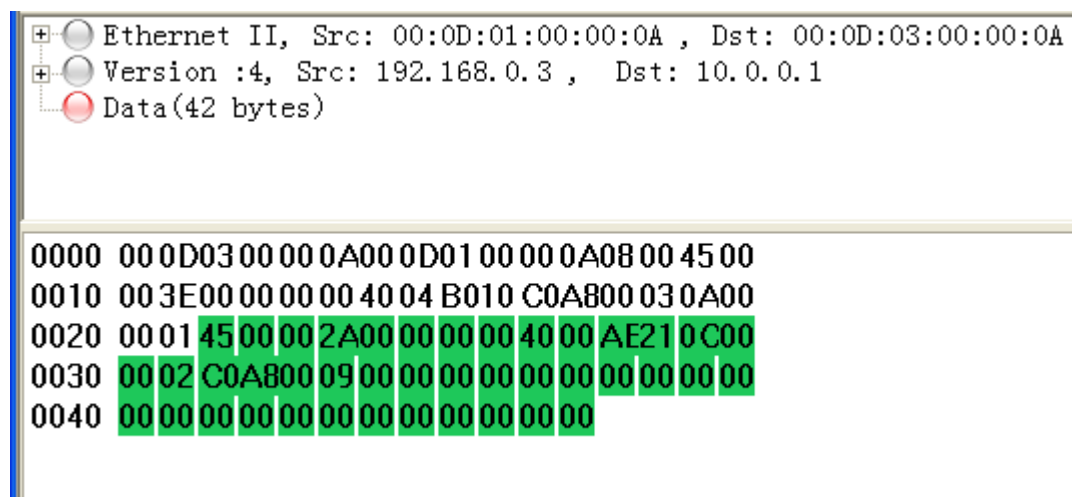
0000 000D0100000A000D0300000A08004500
0010 0052000000004004AFFCC0A800030A00
0020 00014500003E000000004004AF090B00
0030 0002C0A800094500002A000000004000
0040 AE250C000002C0A80005000000000000
0050 00000000000000000000000000000000
  
```

-5-包传递-外地代理功能的实现

7	Sun...	192.168.0.3	10.0.0.1	IP	Version 4, Src: 192.168.0.3, Dst: 10.0.0.1	10.6 包传递-外地代理功能的实现
8	Sun...	12.0.0.2	192.168.0.9	IP	Version 4, Src: 12.0.0.2, Dst: 192.168.0.9	10.6 包传递-外地代理功能的实现
9	Sun...	192.168.0.9	11.0.0.1	TCP	Bogus TCP header length must be at least 20	10.6 包传递-外地代理功能的实现
10	Sun...	192.168.0.9	11.0.0.1	TCP	Bogus TCP header length must be at least 20	10.6 包传递-外地代理功能的实现

主要分析报文 7，8 来说明外地代理接收到封装后的数据包进行解封转发处理

报文 7 的 data 为 42 个字节，然后 8 为解封后的报文，data 为 22 字节长度。



五、思考问题

-1-移动 IP 协议的工作机制

家乡代理 HA 和外地代理 FA 周期性的组播或者广播一条被称为代理通告的消息来宣告自己的存在。

移动节点周期性地接收到代理通告消息，检查其中的内容以确定自己是连在家乡链路还是外地链路上，当它链接在家乡链路时，移动节点就像固定节点一样工作，不再利用移动 IP 功能。如果移动节点发现它链接在外地链路上，则启用移动 IP 的功能。

连接在外地链路上的移动节点需要一个代表它当前所在位置的转交地址，这个地址可以是外地代理转交地址或者是配置转交地址。如果收不到外地代理的通告消息，移动节点就使用配置转交地址，这可以通过一个常规的 IP 地址配置规程得到，比如用 DHCP 动态配置或者手工静态完成。

移动节点在向家乡代理注册自己已经获得的转交地址，在注册过程中，如果链路上有外地代理，移动节点就向 FA 请求服务，FA 再将注册包中寄给 HA,为保障网络通信的安全性，注册消息需要进行认证处理。

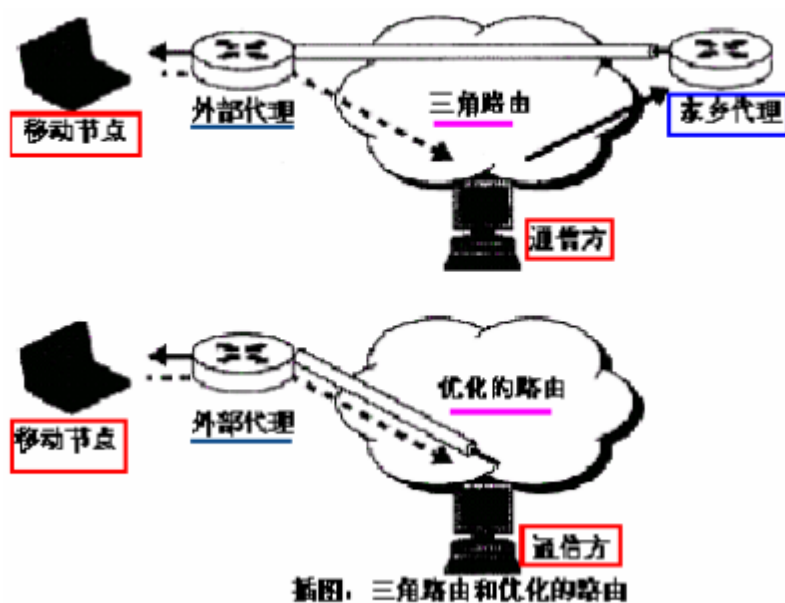
如果注册成功，家乡代理通过 ARP 代理结束得到发往移动节点家乡地址的数据包，并根据移动节点注册到 HA 上的转交地址，通过隧道将这些数据包传送给转交地址，在转交地址处，原始数据包被从隧道中提取出来送给移动节点。

由移动节点发出的数据包，采用家乡地址作为源 IP 地址，使用外地网络的路由器作为默认的路由器，发送的数据包将通过外地网络的路由器直接发送到通信对端，而无需隧道技术。

-2-基本移动 IP 协议会面临三角路由的问题，解释该问题并提出解决方法；

三角路由问题：通信主机发往移动主机的分组必须经过本地代理，从而从移动主机发往 Ch 的分组是直接发送的，2 个方向的通信不是同一路径，产生三角路由问题，这在移动主机远离本地代理，通信主机和移动主机相邻的情况下效率尤其低下。

6.5. 路由操作和三角路由问题



移动节点可以把他的转交地址直接通知给通信方，并让它们把分组直接传到移动节点，即旁路掉家乡代理，但是这种优化，在延迟和资源消耗方面，可能比三角路由更加有效，因为一般而言分组在他们通往目的地的路径上会跨越较少的链路。

解决方案：

可以采用绑定和隧道技术，具体过程是：当发往移动主机的数据报首先到达本地网络时，本地网络立刻发出绑定请求，虽有发送数据的路由代理将对后继要发送的数据报进行分装，采用隧道技术直接送到移动主机，而不是发送到本地网络再通过转交地址到达外地网，这样就解决了路由的优化问题，但其负面的影响是增加了地址转换的开销，由于隧道技术必须对数据进行封装和拆

封，因此对路由代理的软件进行修改。同时绑定即时的应用会引起网络安全性方面的问题，破坏者可以利用绑定技术，使数据沿着他所指的方向进行传送，从而达到窃取数据的目的。

六、实验中遇到的问题

此次试验遇到了很多问题：

- 【1】 注册消息时：关于 ICMP 报文的校验码字段计算方法，后来在网上搜过后，才知道 ICMP 校验码计算方法和 IPV4 报文的校验计算方法是不同的，而且校验算法是没有固定格式的，不同的报文类型的计算方法都是不一样的。按照 2 个字节对齐进行反码加法，同时需要转换字节序
- 【2】 添加默认路由的时候，对默认路由认识不清楚，默认路由只需要设置下一跳字段为 `getFAHAddress()` 即可，其他字段均为 0；
- 【3】 外地代理请求注册报文函数和外地代理接收注册报文函数之间需要传递全局参数掩码和接受报文的接口号。
- 【4】 遇到了很多细节的地方没有进行字节序转行会报各种错误。
- 【5】 家乡代理接收发往移动节点报文的函数；部分实现比较繁琐，开始时没有理解，后来分类计算各种情况，开始时会报报文都不长度或者 TTL 等各种字段的错误。
- 【6】 计算校验和字段时，遇到了一些问题，需要将报文的校验和字段初始为 0，然后调用校验计算函数计算校验和即可。

七、实验的启示/意见和建议

此次实验大约花费了 30+ 小时吧！包括写代码+调代码+写报告 主要实验太烦琐了，很多细节都要仔细学习，不过通过此次实验加深了自己对于 ICMP, 以及用户隧道实现等链路层的封装机制有了更深的理解，同时也提升了一定的编程能力。