

Exp 25: IP Aliasing

目的：了解什么是IP 别名(IP aliasing)及其应用。

摘要：介绍一组在网卡上设定多个IP地址的技术IP别名(IP aliasing)。同时通过MDDL的程序编辑，学生可以了解如何去使用IP aliasing这个机制。

时间：4.5 小时。

一、网络拓扑

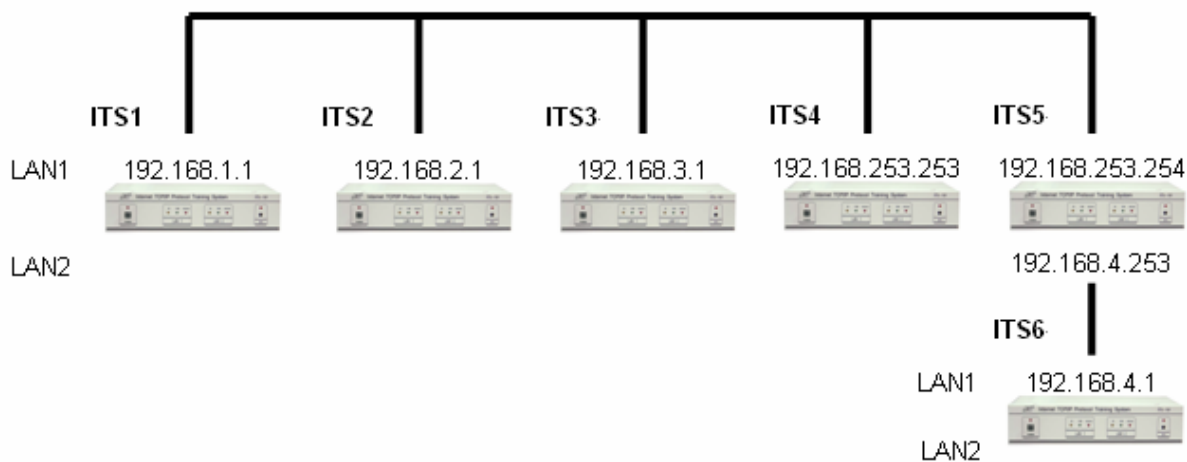


图 25.1

二、技术背景

一般来说，一张实体网络接口，如Ethernet网卡，只会对应一个IP地址，然而这并不是绝对的，IP别名(IP aliasing)就是一种能在一张网卡上配置多个IP地址的技术。透过IP别名的使用，我们可以让局域网络更加灵活与多变，例如在一个实体网络上切割多个子网络，这些子网络彼此不能直接通讯但却能一同透过一个使用IP别名的服务器对外联系。

三、实验步骤

1、了解网络拓扑

1) 在Hubox 上将网络连接如图25.2所示。

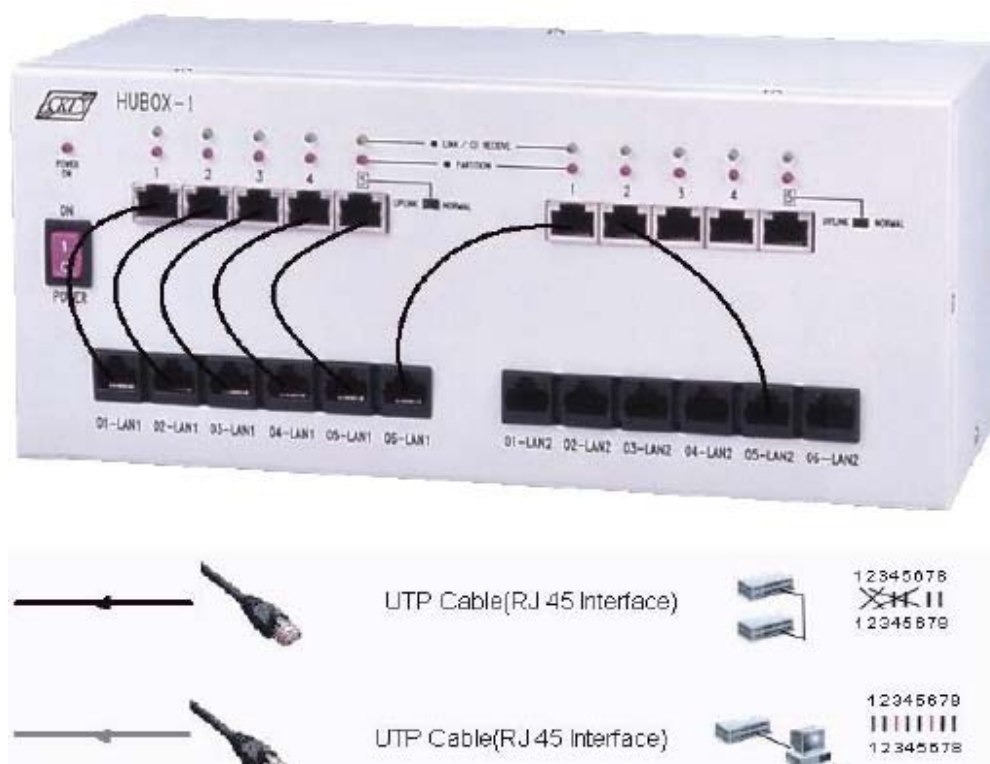


图 25.2

2、IP Alias Application 实验

A. 初始设置:

- 2) 执行 **XCLIENT.BAT**，打开 ITS 应用软件 KCodes Network Explorer。
- 3) 从 Tool 菜单部分打开“网络配置”对话框 (**Network Configuration**)。

ITS1 设置如下:

- 4) 定义 Interface 1 的 IP 地址为“**192.168.1.1**”并且定义 Interface 1 的 Subnet Mask 为“**255.255.255.0**”。在路由表部分 (Routing Table), “Destination ”and “Mask”部分输入“**0.0.0.0**”; “Gateway”部分输入“**192.168.1.253**”(见图 25.3)。

ITS2 设置如下:

- 5) 定义 Interface 1 的 IP 地址为“**192.168.2.1**”并且定义 Interface 1 的 Subnet Mask 为“**255.255.255.0**”。在路由表部分 (Routing Table), “Destination ”and “Mask”部分输入“**0.0.0.0**”; “Gateway”部分输入“**192.168.2.253**” 模式设定为 “**Host**”, 然后点击“**Set & Close**”按钮。

ITS3 设置如下:

- 6) 定义 Interface 1 的 IP 地址为“**192.168.3.1**”并且定义 Interface 1 的 Subnet Mask 为“**255.255.255.0**” 在路由表部分 (Routing Table), “Destination ”and

“Mask”部分输入“**0.0.0.0**”；“Gateway”部分输入“**192.168.3.253**”。模式设定为“**Host**”，然后点击“**Set & Close**”按钮。

ITS4 (IP Alias) 设置如下：

- 7) 定义 Interface 1 的 IP 地址为“**192.168.253.253**”并且定义 Interface 1 的 Subnet Mask 为“**255.255.255.0**”在路由表部分 (Routing Table), “Destination ”and “Mask”部分输入“**0.0.0.0**”；“Gateway”部分输入“**192.168.253.254**”。模式设定为“**Gateway**”，然后点击“**Set & Close**”按钮。

ITS5 (Gateway) 设置如下：

- 8) 定义 Interface 1 的 IP 地址为“**192.168.253.254**”，定义 Interface 2 的 IP 地址为“**192.168.4.253**”。定义 Interface 1 和 Interface 2 的 Subnet Mask 为“**255.255.255.0**”。在路由表部分 (Routing Table), “Destination ”and “Mask”部分输入“**0.0.0.0**”；“Gateway”部分输入“**192.168.253.253**”模式设定为“**Gateway**”，然后点击“**Set & Close**”按钮。

ITS6 设置如下：

- 9) 定义 Interface 1 的 IP 地址为“**192.168.4.1**”并且定义 Interface 1 的 Subnet Mask 为“**255.255.255.0**”在路由表部分 (Routing Table), “Destination ”and “Mask”部分输入“**0.0.0.0**”；“Gateway”部分输入“**192.168.4.253**”模式设定为“**Host**”，然后点击“**Set & Close**”按钮 (因为我们给每台 ITS 设置了不同的子网 IP 地址, ITS1 和 ITS2 虽然在同一个物理网络内，但他们还是在不同的子网中)。

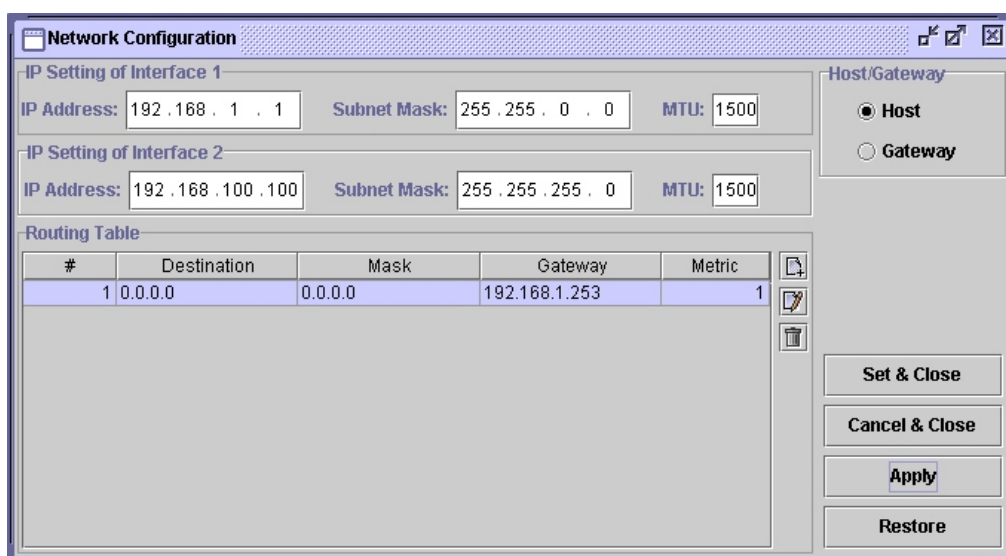


图 25.3

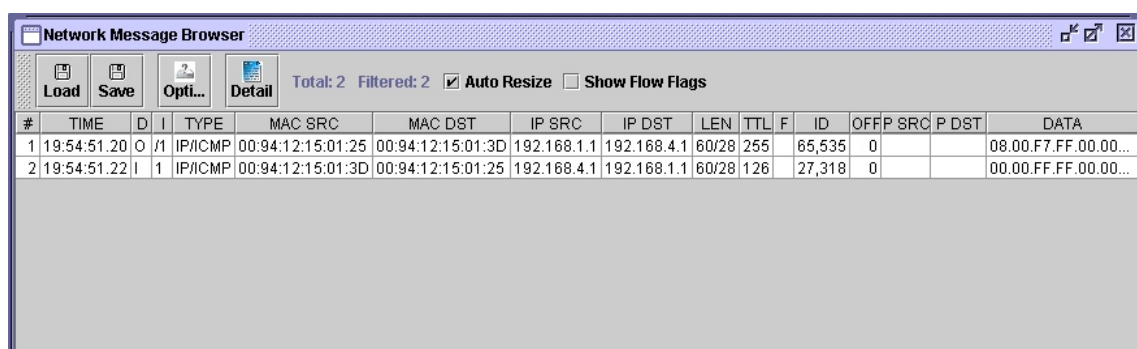
B. IP Aliasing 实验:

ITS4 设置如下:

- 10) 打开网络封包浏览器 (Network Message Browser). 检查是否打开 **Listening**。
- 11) 打开 MDDL 编辑界面。
- 12) 点击 **Load** 按钮。调用 C: \XClient \Data \Mddl \Tutorial \Ex25 \IPAlias.mddl, 然后点击 **Upld** 按钮。

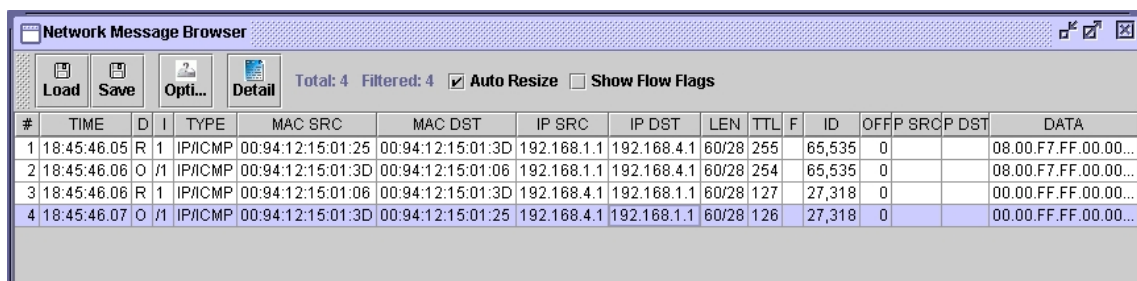
ITS 1, 2, 3, 5 and 6 设置如下:

- 13) 打开网络封包浏览器 (Network Message Browser). 检查是否打开 **Listening**。
- 14) 根据实验 4 部分(图 4.5), 由 ITS1 发送 ICMP Echo Request 封包至 ITS6 图 25.4 显示了 ITS1 发送了一个 ICMP Echo Request 封包并且接收到一个 ICMP Echo Reply 封包。图 25.5 显示了 ITS4 (IP Alias) 接收到一个 ICMP Echo Request 封包, 并且绕送给 ITS6。图 25.6 显示了 ITS6 接收到 ICMP Echo Request 封包并且发送一个 ICMP Echo Reply 封包。



#	TIME	D	I	TYPE	MAC SRC	MAC DST	IP SRC	IP DST	LEN	TTL	F	ID	OFFP	SRC	P	DST	DATA
1	19:54:51.20	O	/1	IP/ICMP	00:94:12:15:01:25	00:94:12:15:01:3D	192.168.1.1	192.168.4.1	60/28	255		65,535	0				08.00.F7.FF.00.00...
2	19:54:51.22	I	1	IP/ICMP	00:94:12:15:01:3D	00:94:12:15:01:25	192.168.4.1	192.168.1.1	60/28	126		27,318	0				00.00.FF.FF.00.00...

图 25.4



#	TIME	D	I	TYPE	MAC SRC	MAC DST	IP SRC	IP DST	LEN	TTL	F	ID	OFFP	SRC	P	DST	DATA
1	18:45:46.05	R	1	IP/ICMP	00:94:12:15:01:25	00:94:12:15:01:3D	192.168.1.1	192.168.4.1	60/28	255		65,535	0				08.00.F7.FF.00.00...
2	18:45:46.06	O	/1	IP/ICMP	00:94:12:15:01:3D	00:94:12:15:01:06	192.168.1.1	192.168.4.1	60/28	254		65,535	0				08.00.F7.FF.00.00...
3	18:45:46.06	R	1	IP/ICMP	00:94:12:15:01:06	00:94:12:15:01:3D	192.168.4.1	192.168.1.1	60/28	127		27,318	0				00.00.FF.FF.00.00...
4	18:45:46.07	O	/1	IP/ICMP	00:94:12:15:01:3D	00:94:12:15:01:25	192.168.4.1	192.168.1.1	60/28	126		27,318	0				00.00.FF.FF.00.00...

图 25.5

The screenshot shows a 'Network Message Browser' window with a toolbar (Load, Save, Opti..., Detail) and status 'Total: 2 Filtered: 2'. It contains a table of network messages:

#	TIME	D	I	TYPE	MAC SRC	MAC DST	IP SRC	IP DST	LEN	TTL	F	ID	OFF	P SRC	P DST	DATA
1	18:45:38.35	R	2	IP/ICMP	00:94:12:15:02:06	00:14:38:15:9D:C8	192.168.1.1	192.168.4.1	60/28	254		65,535	0			08.00.F7.FF.00.00...
2	18:45:38.35	R	1	IP/ICMP	00:94:12:15:01:06	00:94:12:15:01:3D	192.168.4.1	192.168.1.1	60/28	128		27,318	0			00.00.FF.FF.00.00...

图 25.6

四、实验讨论

- 1、试着从ITS 1 发出ICMP Echo Request 给ITS 2 或ITS 3, 然后观察其封包传递过程。
- 2、ITS 1 是否能够直接地将封包丢给ITS 2 或 ITS 3? 如果不行, 它透过什么方式?

REACTOR PROGRAM

1、IPAlias.mddl

```
// IP Alias

VAR1[0,3]      = { 192, 168, 1, 253 }      ; // ITS 1
VAR1[4,7]      = { 192, 168, 2, 253 }      ; // ITS 2
VAR1[8,11]     = { 192, 168, 3, 253 }      ; // ITS 3

VAR2[0]        = 1                          ; // 192.168.253.253
VAR2[1]        = 2                          ;
VAR2[2,5]      = { 192, 168, 253, 254 }     ; // default gateway

ETHER_IN_HANDLER VAR2[0]
{
    IF(S.ETHER_TYPE==CNST_ETHER_TYPE_ARP&&S.ETHER_ARP_OP==CNST_ETHER_A
    RP_OP_ARPREQ )
    {

        IF(S.ETHER_ARP_IPADDRTRGT==VAR1[0,3]||S.ETHER_ARP_IPADDRTRGT==VAR1[4,7]||
        S.ETHER_ARP_IPADDRTRGT==VAR1[8,11])
```

```
{
    SEND_OUT_ETHER_FROM_INTERFACE VAR2[0] WITH_DATA
    {
        T
            = S
            ,
        T.ETHER_MACADDRDST = S.ETHER_MACADDRSRC
            ,
        T.ETHER_MACADDRSRC = MYMAC(VAR2[0])
            ,
        T.ETHER_ARP_OP      = CNST_ETHER_ARP_OP_ARPREPLY,
        T.ETHER_ARP_MACADDRSNDR = MYMAC(VAR2[0])
            ,
        T.ETHER_ARP_IPADDRSNDR  = S.ETHER_ARP_IPADDRTRGT
            ,
        T.ETHER_ARP_MACADDRTRGT =
S.ETHER_ARP_MACADDRSNDR
            ,
        T.ETHER_ARP_IPADDRTRGT  = S.ETHER_ARP_IPADDRSNDR
    }
    DISCARD_MESSAGE;
}

}

}

IP_ROUTING_HANDLER
{

    IF(!ISMYIPADDR(S.IP_ADDRDST)&&S.IP_ADDRDST!=VAR1[0,3]&&S.IP_ADDRDST!=V
AR1[4,7]&&S.IP_ADDRDST!=VAR1[8,11])
    {
        IF(S.IP_TTL>1)
        {
            IF((S.IP_ADDRDST.[0,2]== MYIPADDR(VAR2[0]).[0,2])||(S.IP_ADDRDST.[0,2] ==
VAR1[0,3].[0,2]) ||(S.IP_ADDRDST.[0,2]==VAR1[4,7].[0,2])||(S.IP_ADDRDST.[0,2] ==
VAR1[8,11].[0,2]))
```

```

        {
            SEND_OUT_IP_FROM_INTERFACE VAR2[0] WITH_NEXTHUB
S.IP_ADDRDST
            WITH_DATA
            {
                T                                = S                                ,
                T.IP_TTL                        = S.IP_TTL - 1                        ,
                T.IP_HEADERCHKSUM = {0, 0}                                ,
                T.IP_HEADERCHKSUM = CHECKSUM(T.IP_HEADER)
            }
            DISCARD_MESSAGE;
        }

ELSE
{
    IF(S.IP_ADDRDST.[0,2]== MYIPADDR(VAR2[1]).[0,2])
    {
        SEND_OUT_IP_FROM_INTERFACE VAR2[1] WITH_NEXTHUB
S.IP_ADDRDST
        WITH_DATA
        {
            T                                = S                                ,
            T.IP_TTL                        = S.IP_TTL - 1                        ,
            T.IP_HEADERCHKSUM = {0, 0}                                ,
            T.IP_HEADERCHKSUM = CHECKSUM(T.IP_HEADER)
        }
        DISCARD_MESSAGE;
    }
}

```

```
        ELSE    //default gateway
        {
            SEND_OUT_IP_FROM_INTERFACE VAR2[0] WITH_NEXTHUB
VAR2[2,5]
            WITH_DATA
            {
                T                = S                ,
                T.IP_TTL          = S.IP_TTL - 1      ,
                T.IP_HEADERCHKSUM = {0, 0}            ,
                T.IP_HEADERCHKSUM = CHECKSUM(T.IP_HEADER)
            }
            DISCARD_MESSAGE;
        }
    }
}

IP_RECEIVED_HANDLER
{
    IF
    (
        S.IP_PROT                == CNST_IP_PROT_ICMP        &&
        S.IP_DATA.ICMP_TYPE_CODE == CNST_ICMP_TYPE_CODE_ECHOREQ
    )
    {
        IF(ISMYIPADDR(S.IP_ADDRDST)||S.IP_ADDRDST==VAR1[0,3]||S.IP_ADDRDST==VAR1[4
,7]||S.IP_ADDRDST==VAR1[8,11])
```



```

{
    IF((S.IP_ADDRSRC.[0,2]== MYIPADDR(VAR2[0]).[0,2])||(S.IP_ADDRSRC.[0,2] ==
    VAR1[0,3].[0,2]) ||(S.IP_ADDRSRC.[0,2]==VAR1[4,7].[0,2])||(S.IP_ADDRSRC.[0,2] ==
    VAR1[8,11].[0,2]))

    {
        SEND_OUT_IP_FROM_INTERFACE VAR2[0] WITH_NEXTHUB
S.IP_ADDRSRC
        WITH_DATA
        {
            T                                = S                                ,
            T.IP_TTL                        = S.IP_TTL - 1                        ,
            T.IP_ADDRSRC                    = S.IP_ADDRDST                        ,
            T.IP_ADDRDST                    = S.IP_ADDRSRC                        ,
            T.IP_DATA.ICMP_TYPE_CODE =
            CNST_ICMP_TYPE_CODE_ECHOREPLY ,
            T.IP_DATA.ICMP_CHKSUM          = {0x00, 0x00}                        ,
            T.IP_DATA.ICMP_CHKSUM          = CHECKSUM(T[20,])                    ,
            T.IP_HEADERCHKSUM              = {0, 0}                              ,
            T.IP_HEADERCHKSUM              = CHECKSUM(T.IP_HEADER)
        }
        DISCARD_MESSAGE;
    }
ELSE
{
    IF(S.IP_ADDRSRC.[0,2]== MYIPADDR(VAR2[1]).[0,2])
    {
        SEND_OUT_IP_FROM_INTERFACE VAR2[1] WITH_NEXTHUB
S.IP_ADDRSRC
        WITH_DATA

```

```
        {  
            T                                = S                ,  
            T.IP_TTL                        = S.IP_TTL - 1      ,  
            T.IP_ADDR SRC                  = S.IP_ADDR DST      ,  
            T.IP_ADDR DST                  = S.IP_ADDR SRC      ,  
            T.IP_DATA.ICMP_TYPE_CODE      =  
            CNST_ICMP_TYPE_CODE_ECHOREPLY ,  
            T.IP_DATA.ICMP_CHKSUM          = {0x00, 0x00}      ,  
            T.IP_DATA.ICMP_CHKSUM          = CHECKSUM(T[20,]) ,  
            T.IP_HEADER_CHKSUM              = {0, 0}            ,  
            T.IP_HEADER_CHKSUM              = CHECKSUM(T.IP_HEADER)  
        }  
        DISCARD_MESSAGE;  
    }  
}  
}  
}  
}
```