

## Exp 7: IP 绕送之 TTL

**目的：**了解网络上时间的计算方式与TTL的工作原理。

**摘要：**在 ITS 所组成的网络环境中，发出IP报文并使其溢时，让学生学习TTL(Time to Live)在英特网上的用途与实际会发生的机制。

**时间：**6 hrs。

### 一、网络拓扑

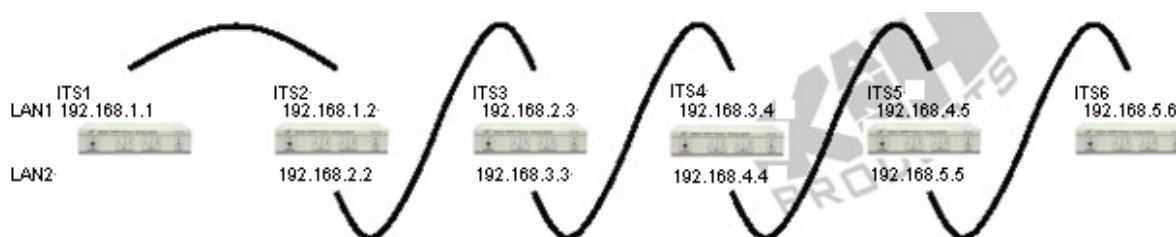


图7.1

### 二、技术背景

TTL(Time to Live)是在英特网协议(IP)里所定义的一个值，包含在IP报文里告诉路由器此数据包还有多久的存活时间。要设定存活时间是因为某些数据包可能没办法顺利在合理的时程内传递到目的端，举例来说，几组设定不是很正确或未设定完整的网络绕送表，导致数据包在传递时陷入无穷循环，不断地在几个相同的子网络内绕来绕去，解决这个问题方法就是设定一个时间，当时间内数据包还没送达目的端就会被扔掉，同时再发出一个ICMP报文通知发送端之前发出去的数据包已溢时(Time Exceeded)让发送端决定是不是要再重传或是启动其它机制。因为在网络上确切估计时间有它的难处，所以TTL 使用了「计算数据包通过的节点数」来取代精确时间量测的方式一个数据包TTL的初始值是系统预设的占了8bits在IP header，每过一个路由器时TTL值会被减1，一直到TTL值变为0时即代表溢时。

### 三、实验步骤

#### 1、网络拓扑连线

1) 在Hubox上将网络连接如 图7.2所示。

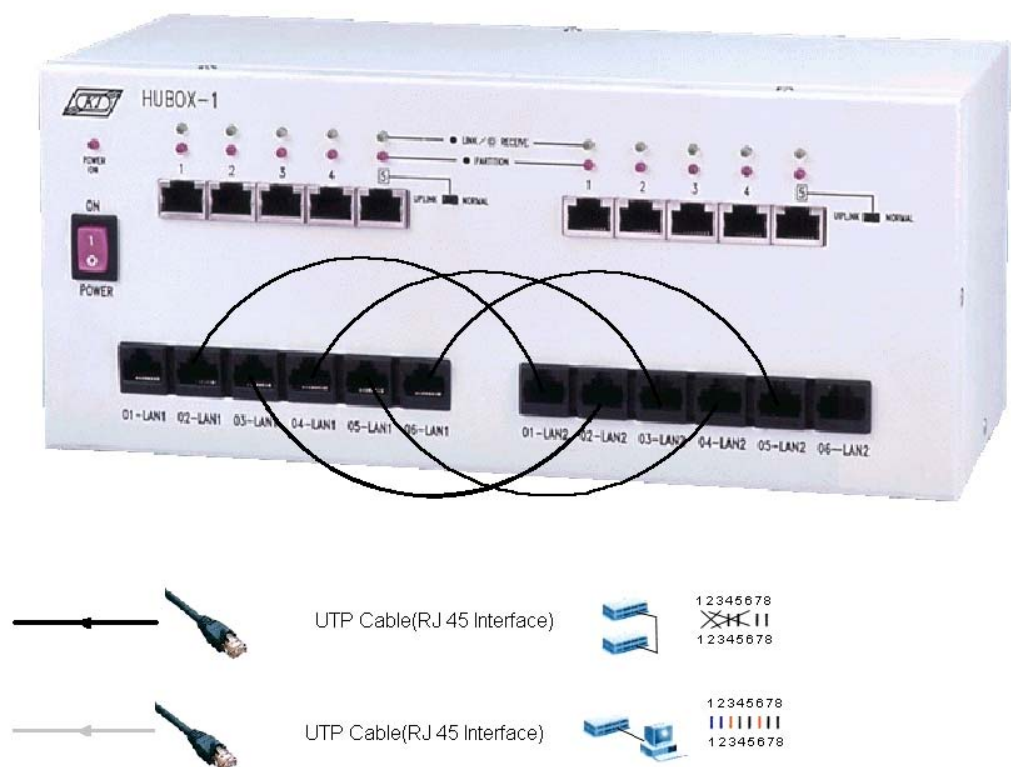


图 7.2

2、观察TTL机制

所有ITS

- 2) 从Tool menu 里选择Network Configuration，打开网络属性配置界面。
- 3) 参考网络拓扑图，设定每台ITS每个interface的IP地址。将 ITS1 and ITS6 设为“Host”模式，其余ITS设为“Gateway” 模式。
- 4) 依照表7.1，配置好每台ITS的路由表，然后点击Set & Close 保存配置信息。此时，IP报文就可以在ITS1 and ITS6之间传送。
- 5) 打开网络信息浏览器（Network Message Browser）,并勾选 Listening On。

Machine No.	Destination	Mask	Gateway
ITS1	0.0.0.0	0.0.0.0	192.168.1.2
ITS2	0.0.0.0	0.0.0.0	192.168.2.3
ITS3	192.168.1.0	255.255.255.0	192.168.2.2
	0.0.0.0	0.0.0.0	192.168.3.4
ITS4	192.168.5.0	255.255.255.0	192.168.4.5
	0.0.0.0	0.0.0.0	192.168.3.3
ITS5	0.0.0.0	0.0.0.0	192.168.4.4
ITS6	0.0.0.0	0.0.0.0	192.168.5.5

表7.1

ITS1

- 6) 发送一个目的地址为“192.168.5.6”的ICMP Echo Request 报文，确定你可以收到由ITS6发送回来的ICMP Echo Reply 报文。
- 7) 发送一个目的地址为“192.168.6.6”的 ICMP Echo Request报文，我们可以观察到一个现象，表7.1的路由表虽然可以让每台ITS相互之间可以正确的传送信息，但对于目的地址“192.168.6.6”.的这个报文传送到ITS4的时候却被转发到ITS3，然后ITS3又会将该报文转发到ITS4，如此形成了循环转发状态。最后该报文由于TTL溢时被丢弃。

ITS6

- 8) 发送一个目的地址为“192.168.0.6”的 ICMP Echo Request ，同步骤7形成了循环转发状态，最终该报文会因为TTL溢时而被丢弃。

**3、利用MDDL产生TTL溢时**所有 ITS

- 9) 重制网络信息浏览器（Network Message Browser）。

ITS2 到 ITS5

- 10) 从Reactor menu 中选择MDDL Reactor Panel。
- 11) 在MDDL编辑平台上，点击Load按钮。调用TTLTimeExceeded.mddl程序（路径为C:\XClient\Data\Mddl\Tutorial\Ex07\TTLTimeExceeded.mddl），最后按下Upld 按钮。

ITS1

- 12) 发送一个目的地址为“192.168.6.6”的ICMP Echo Request报文，同步骤7，在ITS3和ITS4之间形成循环转发的状态，最终该报文会因TTL溢时被丢弃。

**四、实验讨论**

- 1、在此实验案例里，ITS的网络绕送表是否有错，又或者有缺陷？
- 2、如何解决步骤7与8中，封包会陷入无穷循环的问题？

**REACTOR PROGRAM****1、TTLTimeExceeded.mddl**

```
IP_ROUTING_HANDLER
{
```

```

    IF(ISMYIPADDR(S.IP_ADDRDST))
        RETURN;
    IF((S.IP_ADDRDST.[0,2]!=MYIPADDR(1).[0,2])&&(S.IP_ADDRDST.[0,2]!=MYIPADDR(2).[0,2]))
        //compare with netmask 255.255.255.0
        RETURN;
    IF(S.IP_TTL>1)
    {
        SEND_OUT_IP WITH_DATA
        {
            T                = S                ,
            T.IP_TTL          = S.IP_TTL - 1      ,
            T.IP_HEADERCHKSUM = {0, 0}           ,
            T.IP_HEADERCHKSUM = CHECKSUM(T.IP_HEADER)      }
        }
    ELSE
    {
        SEND_OUT_IP WITH_DATA
        {
            T                = S                ,
            T.IP_TTL          = 255              ,
            T.IP_ADDRSRC      = MYIP(2)          ,
            T.IP_ADDRDST      = S.IP_ADDRDST     ,
            T.IP_DATA.ICMP_TYPE_CODE = {0X0B,0X00} ,
            T.IP_DATA.ICMP_ECHO_ID   ={0x00,0x00} ,
            T.IP_DATA.ICMP_ECHO_SEQNUM ={0x00,0x00} ,
            T.IP_DATA.[8,35]        =S.[0,27]    ,
            T.IP_DATA.[36,39]        ="STOP"      ,
            T.IP_DATA.ICMP_CHKSUM    = {0x00, 0x00} ,
            T.IP_DATA.ICMP_CHKSUM    = CHECKSUM(T[20,]) ,
            T.IP_LEN              = LENGTH(T)      ,
            T.IP_HEADERCHKSUM      = {0, 0}        ,
            T.IP_HEADERCHKSUM      = CHECKSUM(T.IP_HEADER)
        }
    }
    DISCARD_MESSAGE;
}

```