

Exp 12: Sliding Window 滑动窗口

目的： 让学生了解滑动窗口(sliding window)机制在TCP 中的工作原理。

摘要： 本实验主要介绍sliding window机制如何调节网络数据包的流量，实验中并通过MDDL 程序语言，让学生进一步了解其算法及运作流程。

时间： 3 hrs。

一、网络拓扑

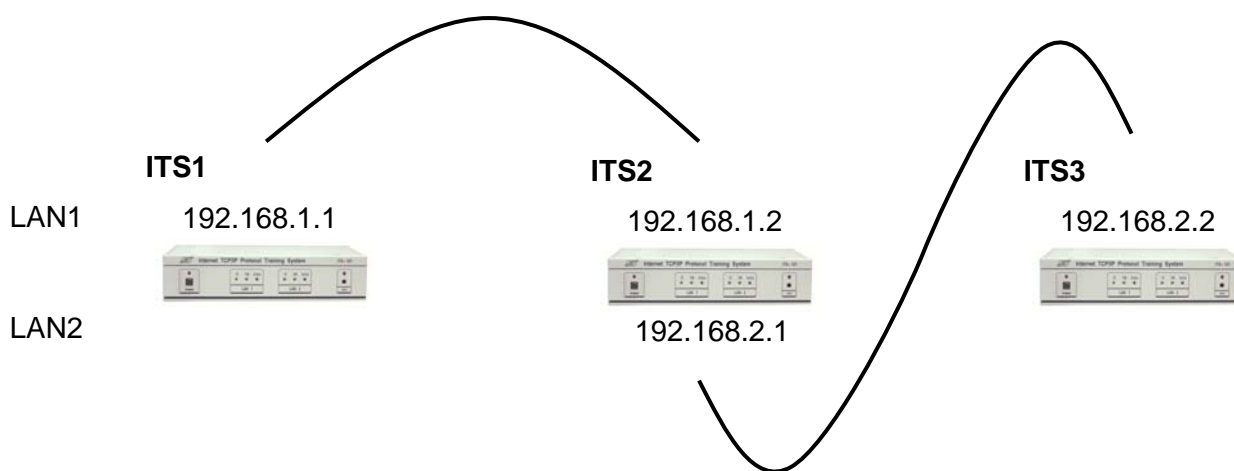


图12.1

二、技术背景

1、流动控制(Flow Control)

1) Idle-RQ

- 非常没有效率。

2) Continuous-RQ

- 发送端连续传送数据包给接收端。

比较下来Continuous-RQ 会是比较好的方案，但万一当接收端已塞满了一堆尚未回应的报文时，就必须有一个机制来通知发送端停止继续传送，sliding window flow control 正是为了解决此问题而发展的机制。

2、滑动窗口(Sliding Window)

1) 设定unacknowledged frames的最大允许量：Send Window Size。

2) 追踪UWE (Upper Window Edge) 及LWE (Lower Window Edge) 的位置。

- 3) UWE: 每执行一次报文传送时UWE 值就加一。
- 4) LWE: 每执行一次报文回应(ACK) 时LWE值就减一。
- 5) 其中发送端要设定(UWE - LWE) Send Window Size。
- 6) 如果当(UWE - LWE) = Send Window Size时, 发送端就自动停止传送封包给接收端。

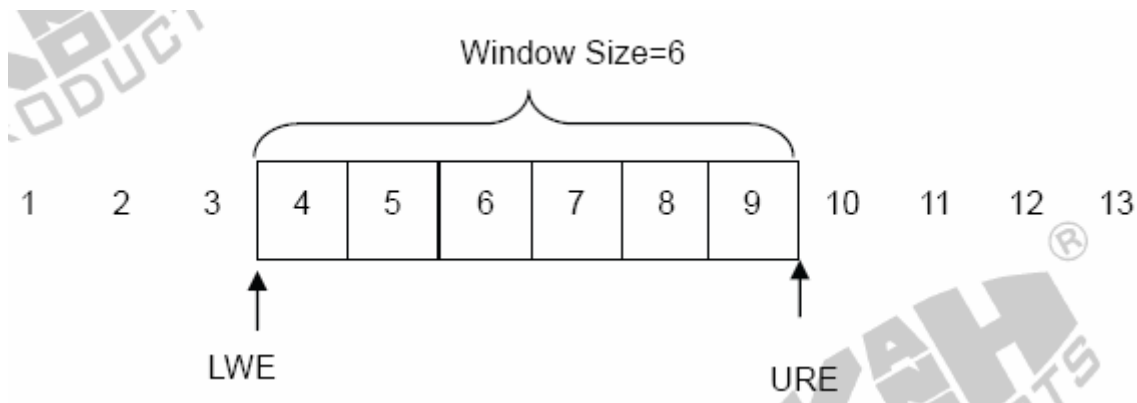


图12.2

3、Sliding Window Control with Variable Window Size

当接收端送出确认报文(ACK)时, 会同时告知发送端其接收端window size的大小(即接收端尚可接收的封包量), 而此window size正意味着接收端目前缓冲区的大小。当接收端回应增加其window size时, 发送端会增大其sliding window 的值, 且继续传出其尚未接获回应确认的报文, 相对的如果接收端响应欲减小其window size 时, 发送端会减低其sliding window 的值, 且停止传出其尚未接获响应确认的报文。

三、实验步骤

1、了解网络拓扑

- 1) 在Hubox 上将网络连接如图12.3所示。

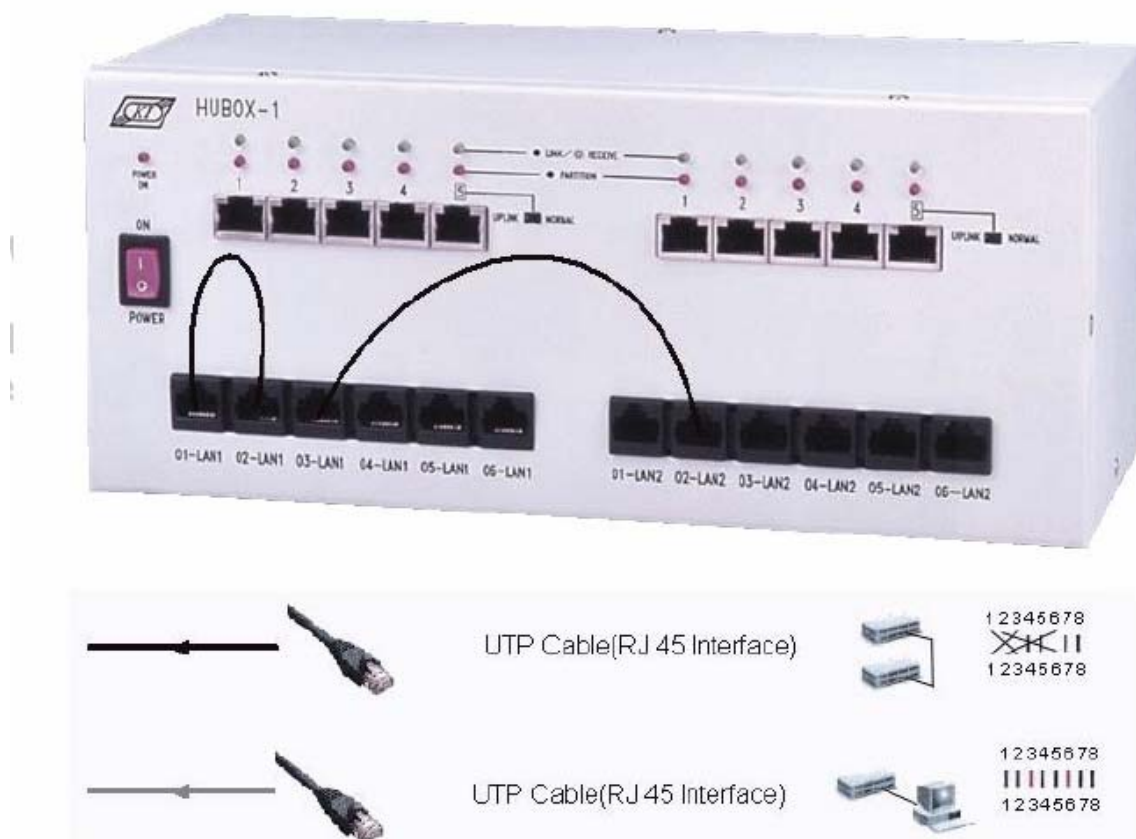


图12.3

2、设置 Host 和 Gateway

- 2) 执行 **XCLIENT.BAT**，打开 ITS 应用软件 KCodes Network Explorer。
- 3) 打开网络封包浏览器 Network Message Browser。

ITS1 (Host) 设置如下：

- 4) 根据拓扑结构，定义 Interface 1 的 IP 地址为“**192.168.1.1**”子网掩码设为“**255.255.255.0**”MTU 设为“**1500**”。然后点击“**Add new routing entry**”按钮见（图 12.4）
- 5) 定义 Destination 为“**192.168.2.0**”，MASK 为“**255.255.255.0**”，Gateway 为“**192.168.1.2**”（见图 12.5）最后点击 **Update** 按钮。
- 6) 模式选择“**Host**”，之后点击“**Set & Close**”按钮。

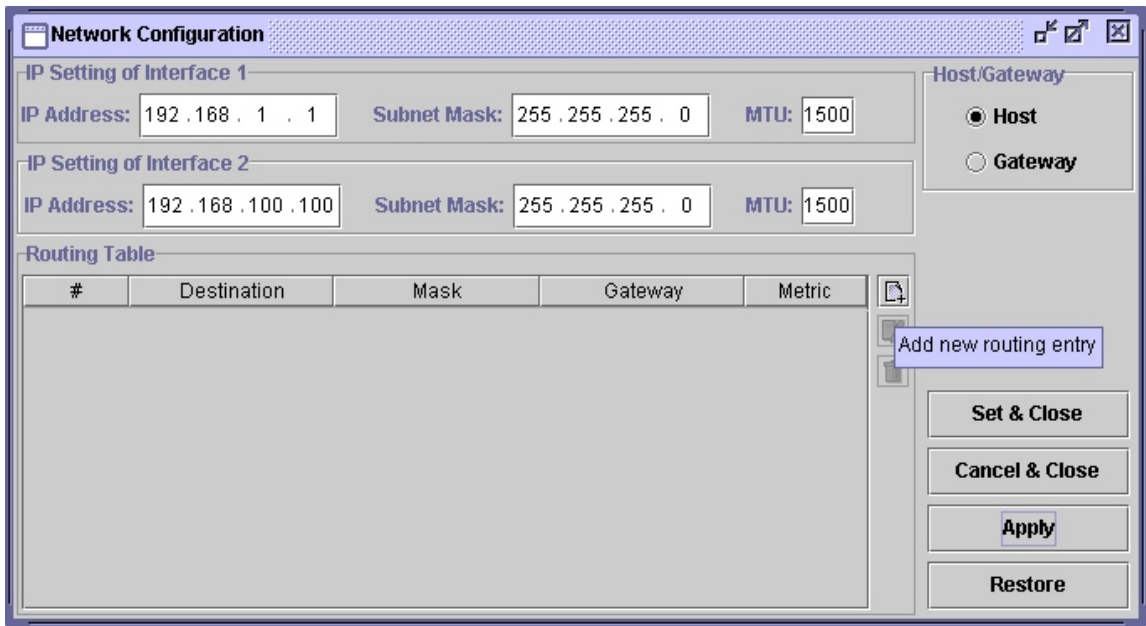


图 12.4

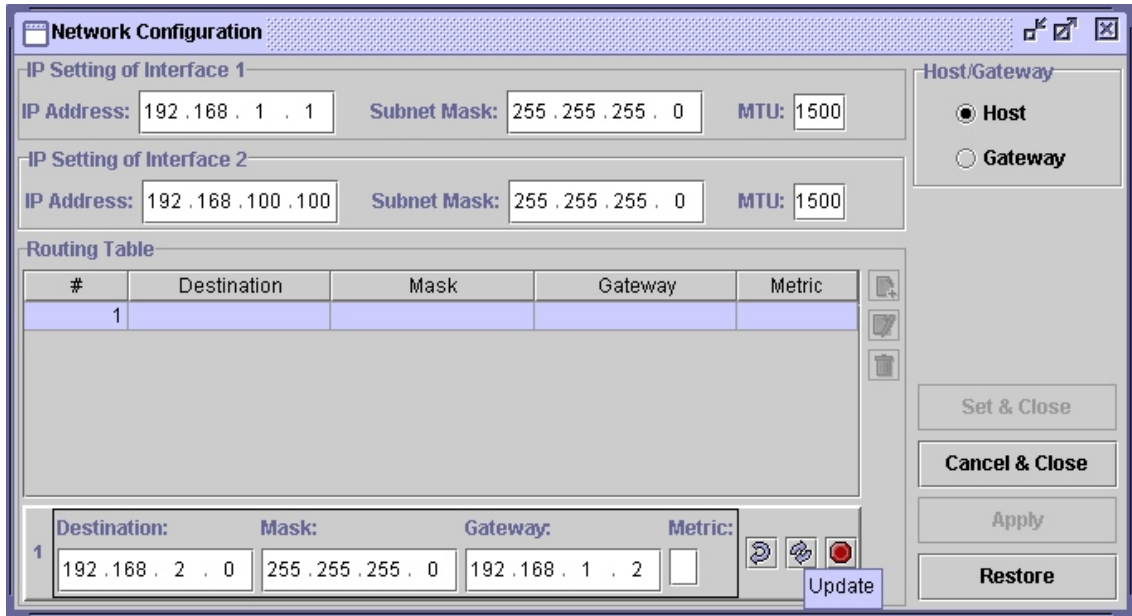


图 12.5

ITS3 (Host) 设置如下:

- 7) 根据拓扑结构, 定义 Interface 1 的 IP 地址为“192.168.2.2”子网掩码设为“255.255.255.0”MTU 设为“1500”。然后点击“Add new routing entry”按钮。
- 8) 定义 Destination 为“192.168.1.0”, MASK 为“255.255.255.0” into Mask, Gateway 为“192.168.2.1”最后点击 **Update** 按钮。
- 9) 模式选择“**Host**”, 之后点击“**Set & Close**”按钮。

ITS2 (Gateway) 设置如下:

- 10) 根据拓扑结构 A 定义 Interface 1 的 IP 地址为“192.168.1.2”, 并且定义 Interface

2 的 IP 地址为“**192.168.2.1**”（见图 12.6）

- 11) 模式选择“**Gateway**”之后点击“**Set & Close**”按钮。现在，我们已经设置好了路由表，下面可以开始实验。

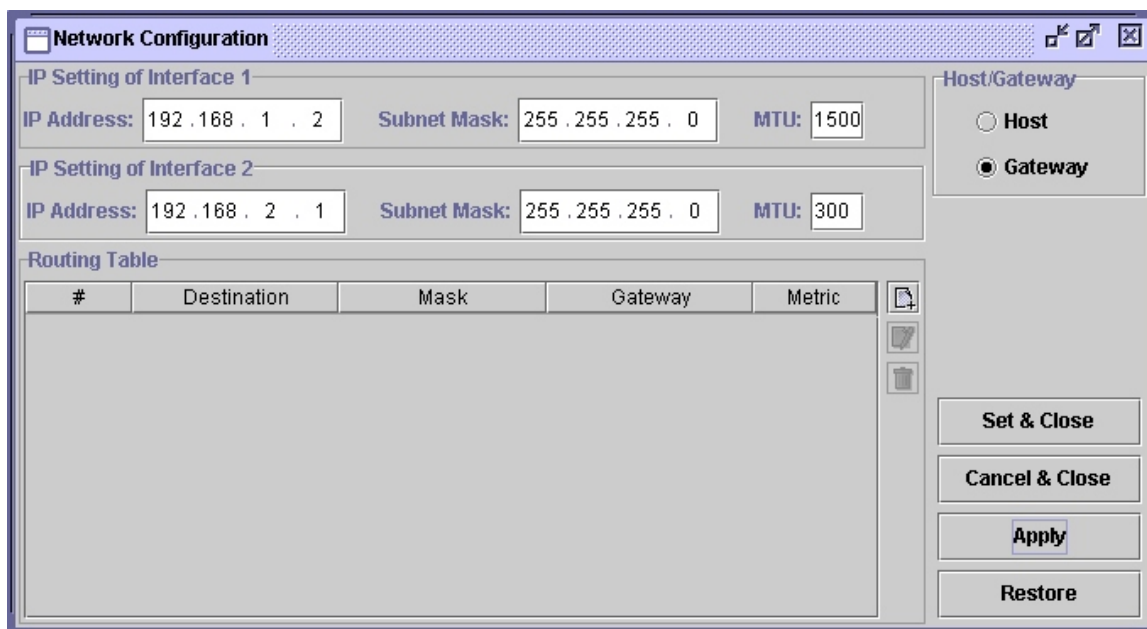


图 12.6

ITS2 操作如下：

- 12) 打开网络封包浏览器（Network Message Browser）界面，同时主意是否打开了监听状态。（**Listening On**）
- 13) 在 Reactor menu 下面点击 **MDDL Reactor Panel** 打开 MDDL 平台（MDDL Editor）。
- 14) 点击 **Load** 按钮，调用 C: \XClient \Data \Mddl \Tutorial \Ex10 \PktLost4.mddl 程序，最后点击 **Upld** 按钮。（Pktlost4 程序的定义如下：每发送 5 个封包会自动丢弃第四个封包）

ITS3 操作如下：

- 15) 打开网络封包浏览器（Network Message Browser）界面，同时主意是否打开了监听状态。（**Listening On**）
- 16) 在 Reactor menu 下面点击 **MDDL Reactor Panel** 打开 MDDL 平台（MDDL Editor）。
- 17) 点击 **Load** 按钮，调用 C: \XClient \Data \Mddl \Tutorial \Ex12 \SlidingWindowReceiver.mddl，最后点击 **Upld** 按钮。

ITS1 操作如下:

- 18) 打开网络包浏览器 (Network Message Browser) 界面, 同时主意是否打开了监听状态。(Listening On)
- 19) 在 Reactor menu 下面点击 **MDDL Reactor Panel** 打开 MDDL 平台 (MDDL Editor)。
- 20) 点击 **Load** 按钮, 调用 C: \XClient \Data \Mddl \Tutorial \Ex12 \SlidingWindowSender.mddl, 最后点击 **Upld** 按钮。
- 21) 打开 IP 封包的发送界面 (IP Datagram Sender)。在 Protocol 部分定义“7”, 输入 Destination IP 为 “192.168.2.2”, 数据段部分输入 “check”(见图 12.7)。

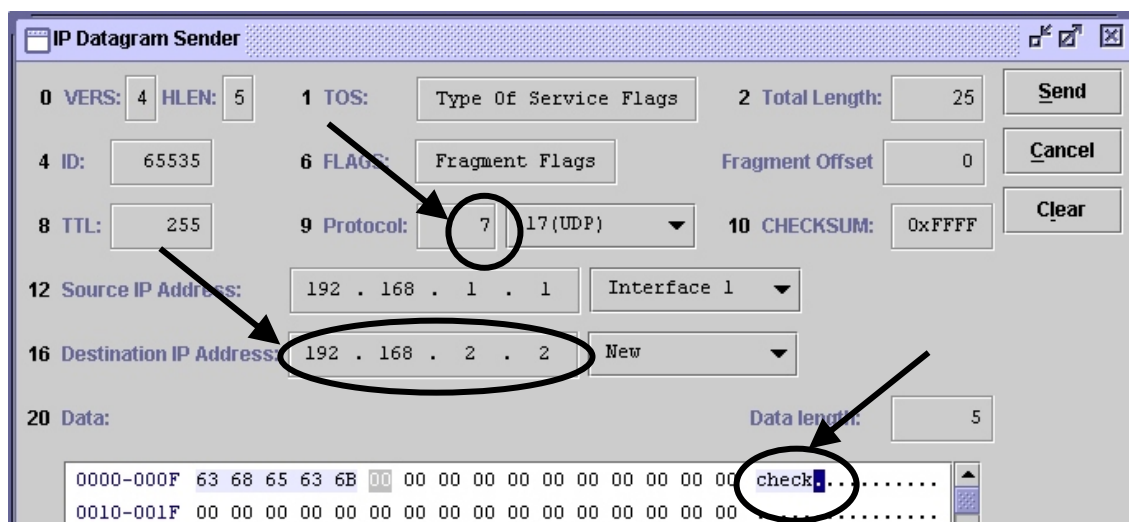


图 12.7

- 22) 最后, 点击“**Send**”按钮。ITS1 机将会发送一个 IP 包给 ITS3, ITS3 会回馈一个 ACK 给 ITS1。见图 12.8, 打开 ITS1 的网络封包浏览器, 我们可以发现 window size 的大小为 3。当 IP 封包被丢弃后, 处于重发状态, 此时再手动发一个新的 IP 封包, window size 变为 1; 如果在重发状态中发送了两个新的 IP 封包, window size 变为 0 (即可用的窗口以满) 见图 12.9(Usable Window Size = SND.UNA + SND.WND - SND.NXT)。

1	15:42:09.46	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	16	0	00.00.00.00.63.68...
3	15:42:09.46	O	/I	ARP	00:94:02:14:01:35	FF:FF:FF:FF:FF:FF			60				00.01.08.00.06.04...
4	15:42:09.46	I	1	ARP	00:94:02:14:01:11	00:94:02:14:01:35			60				00.01.08.00.06.04...
5	15:42:09.52			<Usrc>									ACK 01
6	15:42:09.51	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	13	0	00.01.01.04.00.00...
7	15:42:09.53			<Usrc>									NXT 01
8	15:42:09.54			<Usrc>									UWIND03
9	15:42:10.78	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	17	0	00.01.00.00.63.68...
10	15:42:10.80	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	17	0	00.01.00.00.63.68...
11	15:42:10.87			<Usrc>									ACK 02
12	15:42:10.86	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	14	0	00.02.01.04.00.00...
13	15:42:10.88			<Usrc>									NXT 02
14	15:42:10.89			<Usrc>									UWIND03
15	15:42:11.79	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	18	0	00.02.00.00.63.68...
16	15:42:11.82	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	18	0	00.02.00.00.63.68...
17	15:42:11.88			<Usrc>									ACK 03
18	15:42:11.87	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	15	0	00.03.01.04.00.00...
19	15:42:11.89			<Usrc>									NXT 03
20	15:42:11.90			<Usrc>									UWIND03
21	15:42:13.25	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	19	0	00.03.00.00.63.68...
22	15:42:13.25	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	19	0	00.03.00.00.63.68...
23	15:42:14.00	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	20	0	00.04.00.00.63.68...
24	15:42:14.00	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	20	0	00.04.00.00.63.68...
25	15:42:14.46			<Usrc>									SACK04
26	15:42:14.47			<Usrc>									ACK 03
27	15:42:14.45	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	16	0	00.03.01.03.00.00...
28	15:42:14.48			<Usrc>									NXT 05
29	15:42:14.49			<Usrc>									UWIND01

图 12.8

30	15:42:14.02	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	21	0	00.05.00.00.63.68...
31	15:42:14.95	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	21	0	00.05.00.00.63.68...
32	15:42:15.02			<Usrc>									SACK05
33	15:42:15.03			<Usrc>									ACK 03
34	15:42:15.01	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	17	0	00.03.01.02.00.00...
35	15:42:15.04			<Usrc>									NXT 06
36	15:42:15.05			<Usrc>									UWIND00
37	15:42:18.61	O	/I	IP/C8	00:94:02:14:01:35	00:94:02:14:01:11	192.168.1.1	192.168.2.2	60/29	255	22	0	00.03.00.00.63.68...
38	15:42:18.69			<Usrc>									ACK 06
39	15:42:18.68	I	1	IP/C8	00:94:02:14:01:11	00:94:02:14:01:35	192.168.2.2	192.168.1.1	60/27	254	18	0	00.06.01.04.00.00...
40	15:42:18.70			<Usrc>									NXT 06
41	15:42:18.72			<Usrc>									UWIND03

图 12.9

四、实验讨论

- 1、在实验中，如果报文在传输时我们将连接ITS 的网络线拔除，会发生什么状况？
- 2、改写PkLost4.mddl 程序让连续两个IP datagram 都会lost，如果在ITS 1 尚未 retransmission 之前，再多送一个IP datagram，此时的window size 大小为何？

REACTOR PROGRAMS**1、SlidingWindowSender.mddl**

```
VAR1.SND_UNA      = 0W;                // SND_UNA initialization.

VAR1.SND_NXT      = VAR1.SND_UNA;       // SND_NXT initialization.

VAR1.SND_WND      = 3W;                // SND_WND initialization.


VAR2[0, 3]        = {192, 168, 1, 1};   // SRC Address.

VAR2[4, 7]        = {192, 168, 2, 2};   // DST Address.


VAR4[0,1]         = 0W;                // SACK   (Selective Acknowledgment)


IP_OUT_HANDLER

{

    IF( S.IP_ADDRDST != VAR2[4, 7] || S.IP_PROT == CNST_IP_PROT_KDP )

        RETURN;

    DISCARD_MESSAGE;

//

    IF(VAR1.SND_NXT - (VAR1.SND_UNA + VAR1.SND_WND) <  32768W )

    IF(VAR1.SND_NXT - (VAR1.SND_UNA + 3) <  32768W )

        RETURN;

    ADD_TO_POOL 20 WITH_DATA

    {

        T.[0]                = 6                ,

        T.[1]                = 5                ,

        T.[2,].KDP_ID        = VAR1.SND_NXT      ,

        T.[2,].KDP_ACK       = 0                ,

        T.[2,].KDP_WINDOW_SIZE = 0                ,

        T.[2,].KDP_DATA      = S.IP_DATA
```



```

}
```

```

SEND_OUT_IP WITH_DATA
```

```

{
    T.IP_PROT                = CNST_IP_PROT_KDP  ,
    T.IP_ADDRDST             = VAR2[4, 7]        ,
    T.IP_DATA.KDP_ID         = VAR1.SND_NXT      ,
    T.IP_DATA.KDP_ACK        = 0                 ,
    T.IP_DATA.KDP_WINDOW_SIZE = 0                ,
    T.IP_DATA.KDP_DATA       = S.IP_DATA
}
```

```

VAR1.SND_NXT = VAR1.SND_NXT + 1W;
```

```

}
```

```

TIMER_WITH_PERIOD 1000
```

```

{
```

```

    FOR_EVERY_ELEMENT_IN_POOL 20
```

```

    {
```

```

        PE[0] = PE[0] - 1;
```

```

        IF(PE[0] == 0)
```

```

        {
```

```

            PE[1] = PE[1] - 1;
```

```

            IF(PE[1] == 0)
```

```

            {
```

```

                GENERATE_USER_SYMSG WITH_DATA
```

```

                {
```

```

                    TARGET = "Communication Aborted!"
```

```

                }
```

```

                REMOVE_CURRENT_POOL_ELEMENT;
```

```
        }  
    ELSE  
    {  
        PE[0] = 6;  
        SEND_OUT_IP WITH_DATA  
        {  
            T.IP_PROT          = CNST_IP_PROT_KDP      ,  
            T.IP_ADDRDST       = VAR2[4, 7]            ,  
            T.IP_DATA          = PE.[2,]  
        }  
    }  
}  
}
```

IP_IN_HANDLER

```
{  
    IF( S.IP_ADDRSRC != VAR2[4, 7] || S.IP_PROT != CNST_IP_PROT_KDP ||  
        S.IP_DATA.KDP_ACK != 1)  
        RETURN;
```

```
    IF (S.IP_DATA.[5,6] != VAR1.SND_UNA)
```

```
    {
```

GENERATE_USER_MSG WITH_DATA

```
    {  
        T.[4] = ((S.IP_DATA.[5,6])/10)+0X30 ,  
        T.[5] = ((S.IP_DATA.[5,6])%10)+0X30 ,  
        TARGET = "SACK"
```

```

    }
    VAR4[0,1] = S.IP_DATA.[5,6] ;
}

```

GENERATE_USER_MSG WITH_DATA

```

{
    T.[4] = ((S.IP_DATA.KDP_ID)/10)+0X30,
    T.[5] = ((S.IP_DATA.KDP_ID)%10)+0X30,
    TARGET = "ACK "
}

```

GENERATE_USER_MSG WITH_DATA

```

{
    T.[4] = ((VAR1.SND_NXT)/10)+0X30,
    T.[5] = ((VAR1.SND_NXT)%10)+0X30,
    TARGET = "NXT "
}

```

IF(S.IP_DATA.KDP_ID - VAR1.SND_UNA >= 32768W)

RETURN;

IF(VAR1.SND_NXT - S.IP_DATA.KDP_ID >= 32768W)

RETURN;

DISCARD_MESSAGE;

FOR_EVERY_ELEMENT_IN_POOL 20

```

{
    IF(PE[2,].IP_DATA.KDP_ID - S.IP_DATA.KDP_ID >= 32768W)

```

```
        REMOVE_CURRENT_POOL_ELEMENT;

    }

    VAR1.SND_UNA = S.IP_DATA.KDP_ID;
    VAR1.SND_WND = S.IP_DATA.KDP_WINDOW_SIZE;

    GENERATE_USER_MSG WITH_DATA
    {
        T[4] = ((VAR1.SND_UNA + 3 - VAR1.SND_NXT)/10)+0X30,
        T[5] = ((VAR1.SND_UNA + 3 - VAR1.SND_NXT)%10)+0X30,
        TARGET = "UWND"
    }
}
```

2、SlidingWindowReceiver.mddl

```
VAR1.RCV_NXT    = 0W;                // RCV_NXT initialization.
VAR1.RCV_WND    = 4W;                // RCV_WND initialization.

VAR2[0, 3]      = { 192, 168, 2, 2 }; // SRC Address.
VAR2[4, 7]      = { 192, 168, 1, 1 }; // DST Address.

VAR3[4, 5] = 0W;                    // Some pointer.

IP_IN_HANDLER
{
    IF( S.IP_ADDRSRC != VAR2[4, 7] || S.IP_PROT != CNST_IP_PROT_KDP ||
        S.IP_DATA.KDP_ACK != 0W )
        RETURN;

    DISCARD_MESSAGE;
```

```
IF(S.IP_DATA.KDP_ID - VAR1.RCV_NXT >= 32768W)
```

```
    RETURN;
```

```
IF(S.IP_DATA.KDP_ID - (VAR1.RCV_NXT + VAR1.RCV_WND) < 32768W)
```

```
    RETURN;
```

```
LOOK_FOR_ONE_ELEMENT_IN_POOL 21 WITH_CONDITION (PE.IP_DATA.KDP_ID ==  
    S.IP_DATA.KDP_ID)
```

```
    RETURN;
```

```
VAR1.RCV_WND = VAR1.RCV_WND - 1W;
```

```
GENERATE_USER_MSG WITH_DATA
```

```
{
```

```
    TARGET = VAR1.RCV_WND
```

```
}
```

```
ADD_TO_POOL 21 WITH_CONDITION (S.IP_DATA.KDP_ID - PE.IP_DATA.KDP_ID < 32768W)
```

```
WITH_DATA
```

```
{
```

```
    T = S
```

```
}
```

```
FOR(VAR3[4, 5] = VAR1.RCV_NXT;;VAR3[4, 5] = VAR3[4, 5] + 1W)
```

```
{
```

```
    LOOK_FOR_ONE_ELEMENT_IN_POOL 21 WITH_CONDITION (PE.IP_DATA.KDP_ID ==  
        VAR3[4, 5])
```

```
    {
```

```
        VAR1.RCV_WND = VAR1.RCV_WND + 1W;
```

```
        CONTINUE;

    }

    ELSE

        BREAK;

}

VAR1.RCV_NXT = VAR3[4, 5];

FOR_EVERY_ELEMENT_IN_POOL 21 WITH_CONDITION(PE.IP_DATA.KDP_ID –
VAR1.RCV_NXT >= 32768W)

    REMOVE_CURRENT_POOL_ELEMENT;

SEND_OUT_IP WITH_DATA
{
    T.IP_PROT                = CNST_IP_PROT_KDP,
    T.IP_ADDRDST              = VAR2[4, 7]        ,
    T.IP_DATA.KDP_ID          = VAR1.RCV_NXT      ,
    T.IP_DATA.KDP_ACK         = 1                ,
    T.IP_DATA.KDP_WINDOW_SIZE = VAR1.RCV_WND

}

}
```