

TCP/IP 通訊協定及應用

Spring 2002
 中央大學 吳曉光博士
<http://wmlab.csie.ncu.edu.tw/course/tcp>



Chapter 22: TCP Persist Timer



Introduction

- Why we use persist timer:
 - If a acknowledgment is lost, we could end up with both sides waiting for the other
 - To prevent deadlock from occurring the sender uses a persist timer
 - These segments from the sender are called window probes
- An Example:
 - We'll invoke the server as
 - `svr4 % sock -i -s -P100000 5555`
 - The server sleep for 100,000 seconds(27.8 hours) before reading from the network
 - The client on host bsd1 and performs 1024-byte writes to port 5555 on the server



An Example

```

1 0.0 bsd1.1027 > svr4.5555: # 1:1025(1024) ack 1 win 4096
2 0.191961 ( 0.1920) svr4.5555 > bsd1.1027: . ack 1025 win 4096
3 0.196950 ( 0.0050) bsd1.1027 > svr4.5555: . 1025:2049(1024) ack 1 win 4096
4 0.200340 ( 0.0034) bsd1.1027 > svr4.5555: . 2049:3073(1024) ack 1 win 4096
5 0.207906 ( 0.0075) svr4.5555 > bsd1.1027: . ack 3073 win 4096
6 0.212676 ( 0.0052) bsd1.1027 > svr4.5555: . 3073:4097(1024) ack 1 win 4096
7 0.216115 ( 0.0034) bsd1.1027 > svr4.5555: P 4097:5121(1024) ack 1 win 4096
8 0.219397 ( 0.0039) bsd1.1027 > svr4.5555: P 5121:6145(1024) ack 1 win 4096
9 0.227882 ( 0.0079) svr4.5555 > bsd1.1027: . ack 5121 win 4096
10 0.233012 ( 0.0051) bsd1.1027 > svr4.5555: P 6145:7169(1024) ack 1 win 4096
11 0.237014 ( 0.0040) bsd1.1027 > svr4.5555: P 7169:8193(1024) ack 1 win 4096
12 0.240961 ( 0.0039) bsd1.1027 > svr4.5555: P 8193:9217(1024) ack 1 win 4096
13 0.402143 ( 0.1612) svr4.5555 > bsd1.1027: . ack 9217 win 0
14 0.351561 ( 4.9494) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
15 0.355971 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
16 10.351714 ( 4.9361) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
17 10.355670 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
18 16.351881 ( 5.9962) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
19 16.355849 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
20 28.352213 ( 11.9964) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
21 28.356178 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
22 52.352874 ( 23.9967) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
23 52.356839 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
24 100.354224 ( 47.9974) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
25 100.358207 ( 0.0040) svr4.5555 > bsd1.1027: . ack 9217 win 0
26 160.355514 ( 59.9977) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
27 160.359479 ( 0.0039) svr4.5555 > bsd1.1027: . ack 9217 win 0
28 220.357575 ( 59.9977) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
29 220.361648 ( 0.0041) svr4.5555 > bsd1.1027: . ack 9217 win 0
30 280.359254 ( 59.9976) bsd1.1027 > svr4.5555: . 9217:9218(1) ack 1 win 4096
31 280.363151 ( 0.0041) svr4.5555 > bsd1.1027: . ack 9217 win 0
    
```

Figure 22.1 Example of persist timer probing a zero-sized window.



An Example

- About persist timer:
 - It always bounded between 5 and 60 seconds
 - The persist state is different from the retransmission timeout is that TCP never gives up sending window probes
- What is Silly Window Syndrome
 - It can be caused by the receiver can advertise small windows and the sender can transmit small amounts of data
- Correct avoidance of the silly window syndrome
 - 1. The receiver must not advertise small windows
 - 2. Sender is not transmitting unless one of conditions is true:
 - a. a full-sized segment can be sent
 - b. we can send at least one-half of the maximum sized window ever advertised
 - c. we can send everything we have and either we are not expecting an ACK



Silly Window Syndrome

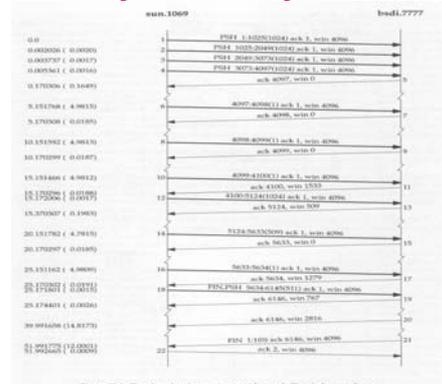


Figure 22.2 Time line showing receiver avoidance of silly window syndrome.



Silly Window Syndrome



Time	Segmental #(bytes 0-3)	Sender TCP	Receiver TCP	Application	Sequence number	Window
0.000	1	1024(1024)			1024	1024
0.000	2	1024(2048)			1024	2048
0.000	3	1024(3072)			1024	3072
0.000	4	1024(4096)			1024	4096
0.000	5	1024(5120)			1024	5120
0.000	6	1024(6144)			1024	6144
0.000	7	1024(7168)			1024	7168
0.000	8	1024(8192)			1024	8192
0.000	9	1024(9216)			1024	9216
0.000	10	1024(10240)			1024	10240
0.000	11	1024(11264)			1024	11264
0.000	12	1024(12288)			1024	12288
0.000	13	1024(13312)			1024	13312
0.000	14	1024(14336)			1024	14336
0.000	15	1024(15360)			1024	15360
0.000	16	1024(16384)			1024	16384
0.000	17	1024(17408)			1024	17408
0.000	18	1024(18432)			1024	18432
0.000	19	1024(19456)			1024	19456
0.000	20	1024(20480)			1024	20480
0.000	21	1024(21504)			1024	21504
0.000	22	1024(22528)			1024	22528
0.000	23	1024(23552)			1024	23552
0.000	24	1024(24576)			1024	24576
0.000	25	1024(25600)			1024	25600
0.000	26	1024(26624)			1024	26624
0.000	27	1024(27648)			1024	27648
0.000	28	1024(28672)			1024	28672
0.000	29	1024(29696)			1024	29696
0.000	30	1024(30720)			1024	30720
0.000	31	1024(31744)			1024	31744
0.000	32	1024(32768)			1024	32768
0.000	33	1024(33792)			1024	33792
0.000	34	1024(34816)			1024	34816
0.000	35	1024(35840)			1024	35840
0.000	36	1024(36864)			1024	36864
0.000	37	1024(37888)			1024	37888
0.000	38	1024(38912)			1024	38912
0.000	39	1024(39936)			1024	39936
0.000	40	1024(40960)			1024	40960
0.000	41	1024(41984)			1024	41984
0.000	42	1024(43008)			1024	43008
0.000	43	1024(44032)			1024	44032
0.000	44	1024(45056)			1024	45056
0.000	45	1024(46080)			1024	46080
0.000	46	1024(47104)			1024	47104
0.000	47	1024(48128)			1024	48128
0.000	48	1024(49152)			1024	49152
0.000	49	1024(50176)			1024	50176
0.000	50	1024(51200)			1024	51200
0.000	51	1024(52224)			1024	52224
0.000	52	1024(53248)			1024	53248
0.000	53	1024(54272)			1024	54272
0.000	54	1024(55296)			1024	55296
0.000	55	1024(56320)			1024	56320
0.000	56	1024(57344)			1024	57344
0.000	57	1024(58368)			1024	58368
0.000	58	1024(59392)			1024	59392
0.000	59	1024(60416)			1024	60416
0.000	60	1024(61440)			1024	61440
0.000	61	1024(62464)			1024	62464
0.000	62	1024(63488)			1024	63488
0.000	63	1024(64512)			1024	64512
0.000	64	1024(65536)			1024	65536
0.000	65	1024(66560)			1024	66560
0.000	66	1024(67584)			1024	67584
0.000	67	1024(68608)			1024	68608
0.000	68	1024(69632)			1024	69632
0.000	69	1024(70656)			1024	70656
0.000	70	1024(71680)			1024	71680
0.000	71	1024(72704)			1024	72704
0.000	72	1024(73728)			1024	73728
0.000	73	1024(74752)			1024	74752
0.000	74	1024(75776)			1024	75776
0.000	75	1024(76800)			1024	76800
0.000	76	1024(77824)			1024	77824
0.000	77	1024(78848)			1024	78848
0.000	78	1024(79872)			1024	79872
0.000	79	1024(80896)			1024	80896
0.000	80	1024(81920)			1024	81920
0.000	81	1024(82944)			1024	82944
0.000	82	1024(83968)			1024	83968
0.000	83	1024(84992)			1024	84992
0.000	84	1024(86016)			1024	86016
0.000	85	1024(87040)			1024	87040
0.000	86	1024(88064)			1024	88064
0.000	87	1024(89088)			1024	89088
0.000	88	1024(90112)			1024	90112
0.000	89	1024(91136)			1024	91136
0.000	90	1024(92160)			1024	92160
0.000	91	1024(93184)			1024	93184
0.000	92	1024(94208)			1024	94208
0.000	93	1024(95232)			1024	95232
0.000	94	1024(96256)			1024	96256
0.000	95	1024(97280)			1024	97280
0.000	96	1024(98304)			1024	98304
0.000	97	1024(99328)			1024	99328
0.000	98	1024(100352)			1024	100352
0.000	99	1024(101376)			1024	101376
0.000	100	1024(102400)			1024	102400

Figure 28.8 Sequence of events for receiver avoidance of the silly window syndrome.



Summary



- TCP's persist timer is set by one end of a connection but has been stopped because the other end has advertised a zero-sized window
- TCP's avoidance of the silly window syndrome is to prevent TCP from advertising small windows or from sending small segments

