**TCP 协议规则**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 位源端口 | | | | | | | | | 16 位目的地址 |
| 32 位序列号 | | | | | | | | | |
| 32 位 确认号 | | | | | | | | | |
| 4位首部长度 | 保留（6） | U | A | P | R | S | F | 16位窗口长度 | |
| 16位TCP检验和 | | | | | | | | | 16位紧急指针 |
| 选项 | | | | | | | | | |
| 数据 | | | | | | | | | |

**选项子段：(tcp/ip : p192)**

选项表结束:

无操作:

最大报文长度

窗口扩大因子：

**TCP连接的建立与中止：**

– – 连接的建立：

22:22:46.510758 IP (tos 0x0, ttl 64, id 29789, offset 0, flags [DF], proto TCP (6), length 60)

192.168.1.86.56132 > 192.168.1.80.8068: Flags [S], cksum 0x8425 (incorrect -> 0xdf97),

seq 4036226398, win 29200, options [mss 1460,sackOK,TS val 10656003 ecr 0,nop,wscale 7], length 0

22:22:46.511781 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)

192.168.1.80.8068 > 192.168.1.86.56132: Flags [S.], cksum 0x1ac4 (correct),

seq 519747379, ack 4036226399, win 14480, options [mss 1460,sackOK,TS val 3745757 ecr 10656003,nop,wscale 6], length 0

22:22:46.511843 IP (tos 0x0, ttl 64, id 29790, offset 0, flags [DF], proto TCP (6), length 52)

192.168.1.86.56132 > 192.168.1.80.8068: Flags [.], cksum 0x841d (incorrect -> 0x8139), seq 1, ack 1, win 229, options [nop,nop,TS val 10656004 ecr 3745757], length 0

**– 最大报文长度：**

当一个连接建立是， 双方告知双方的 MSS 来协商最大窗口选项

**→ 连接请求队列：**

1>等待连接的一段有一个固定长度的连接队列， 该队列中的连接已被TCP接受，但还没有被应用层所接受。

2>应用层只有在三次握手重第三个报文段受到后才会知道这个新的连接，

3> 如果对于新的连接，连接队列中没有空间，TCP将不理会受到的SYN，也不发送任何报文段

4> 当用户层接受一个连接时， 他将从队列中移出；

说明：

TCP 服务器无法使客户进程的主动打开失效。当一个新的客户连接传递给服务器的应用程序时，TCP的 三次握手链接已经结束

**→ Nagle 算法：**

该算法应对小分组增加拥赛出现的问题

该算法要求一个TCP链接上最多只能有一个未被确认的未完成的小分组，该分组的确认到达之前不能发送其他的的小分组，相反， TCP收集这些少量的分组，并在确认到来时以一个分组的方式发出去;

**→ RTT 的計算**

ERR = M – A

a ← A + 1/8 Err

D ← D + 1/4(Err| -D)

RTO = A + 4D

**→ 数据的交互：**

慢启动， 快重传， 拥赛避免策略 p235

**→ 坚持计时器**

当对方的通告窗口为0是的测试窗口的策略（TCP/IP） 245