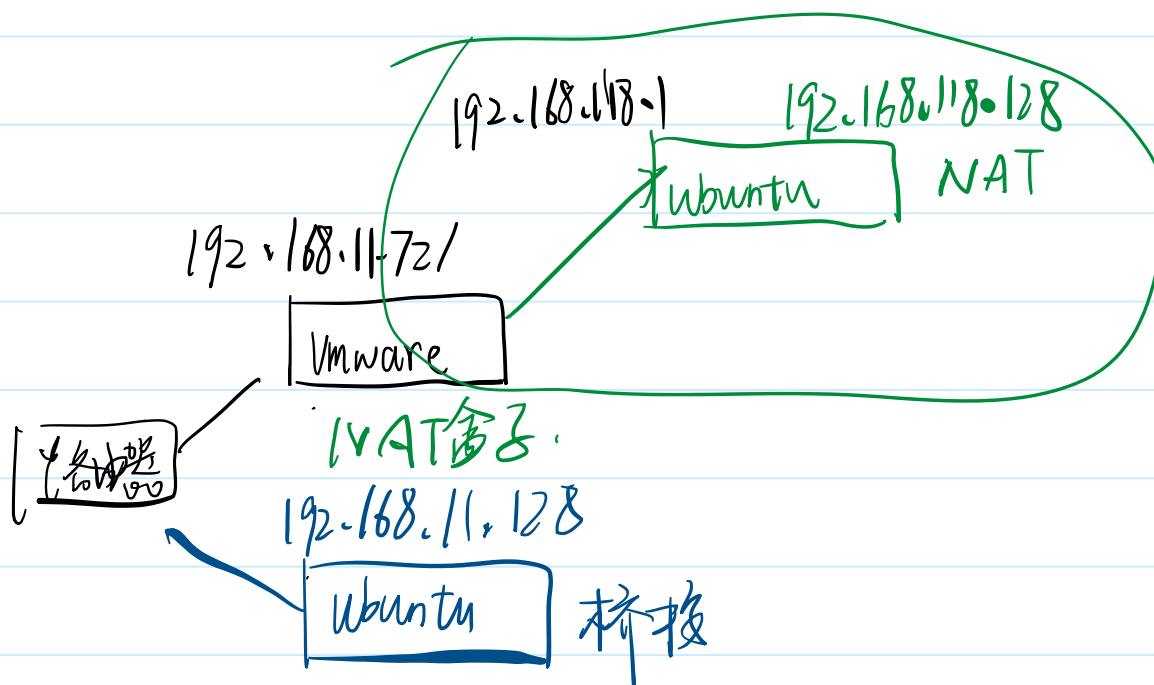


# 虚拟机的网络模式

2023年8月19日 9:58

## 网络连接

- ☐ 桥接模式(B): 直接连接物理网络
- ☐ 复制物理网络连接状态(P)
- ☒ NAT 模式(N): 用于共享主机的 IP 地址



# IP数据报

2023年8月19日 10:05

版本 4bit	首部长度4bit	服务类型8bit	总长度（字节数）16bit	
标识16bit		标志3bit	片偏移13bit	
TTL 8bit	协议 8bit	首部校验和 16bit		
源IP地址 32bit				
目的IP地址 32bit				
可选的选项				
数据				

属于同一个数据包

46bit [0,15] 单位4B, 长度[20,60]

32bit

65535

头

头 载荷

→ 无用

DF Don't Frag  
MF More Frag

当前分片在数据包中的位置

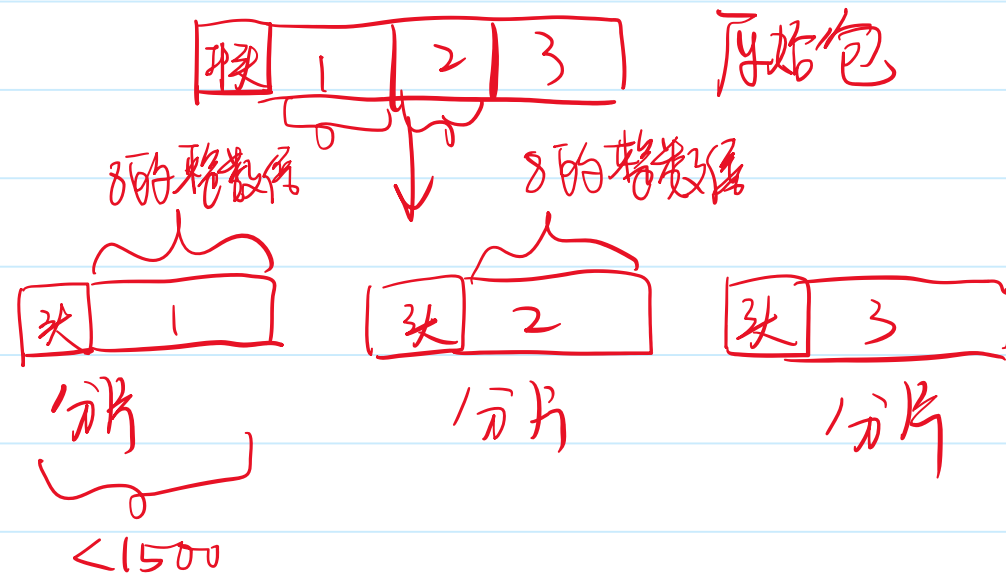
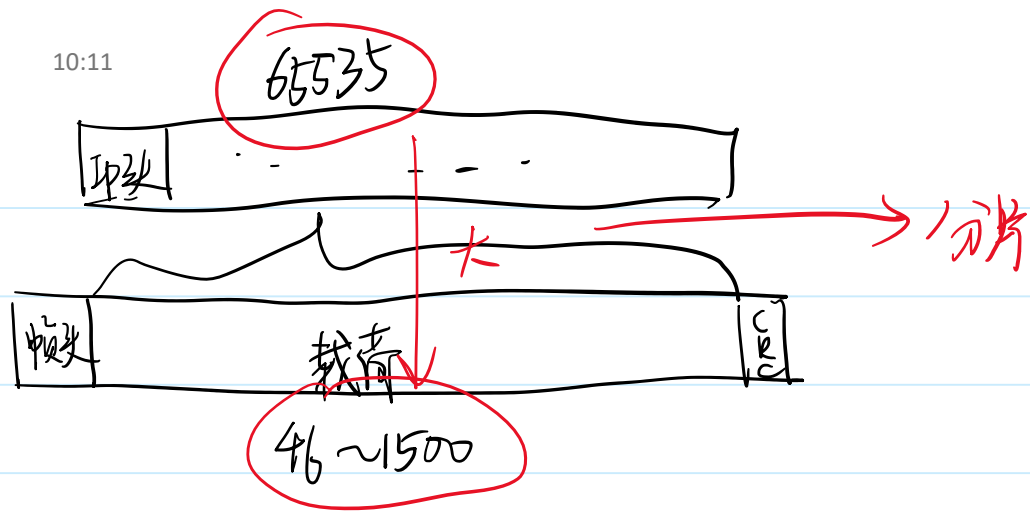
偏移量以8B为单位

载荷

# 分片

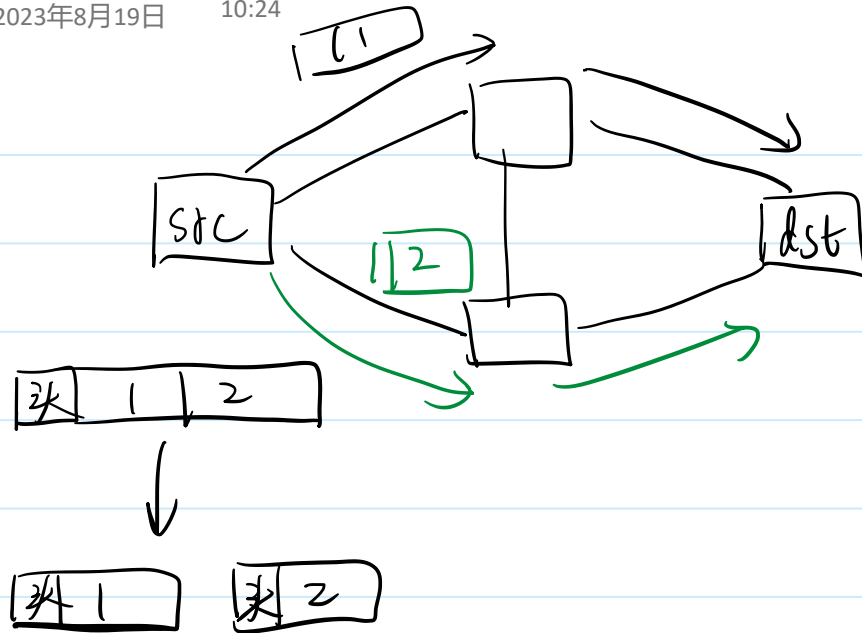
2023年8月19日

10:11



# 尽量不要触发分片

2023年8月19日 10:24

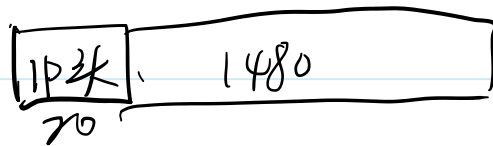


接收方

- ① 乱序
- ② 丢一个, 全丢.

# 分片的例子

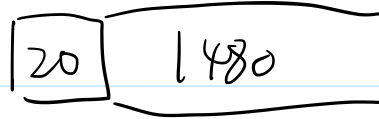
2023年8月19日 10:29



DF = 0

MF = 1

offset = 0



DF = 0

MF = 1

offset = 185



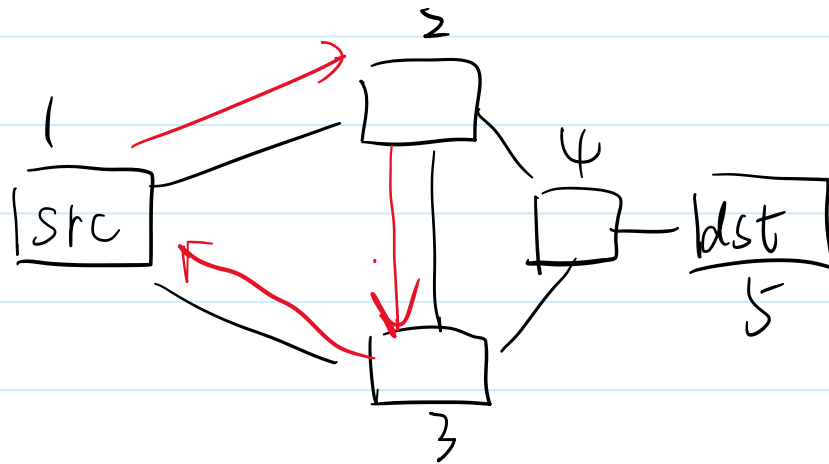
DF = 0

MF = 0

offset = 370

time to live ttl

2023年8月19日 10:33



出报环回以后,避免其影响

# 路由表

2023年8月19日

10:05

把数据交给离终点最近的节点

```
liao:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.168.118.2   0.0.0.0          UG        100    0      0 ens33
169.254.0.0      0.0.0.0          255.255.0.0      U         1000   0      0 ens33
172.17.0.0       0.0.0.0          255.255.0.0      U         0      0      0 docker0
192.168.118.0    0.0.0.0          255.255.255.0    U         100    0      0 ens33
liao:~$ netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask          Flags  MSS  Window  irtt Iface
0.0.0.0          192.168.118.2   0.0.0.0          UG      0 0      0 ens33
169.254.0.0      0.0.0.0          255.255.0.0      U       0 0      0 ens33
172.17.0.0       0.0.0.0          255.255.0.0      U       0 0      0 docker0
192.168.118.0    0.0.0.0          255.255.255.0    U       0 0      0 ens33
liao:~$
```

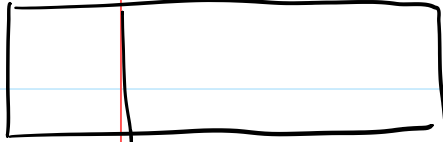
Destination 收到包的终点.

Genmask 生成掩码

Gateway 网关, 下一跳的IP. 0.0.0.0 本地链路

# 路由表的使用流程

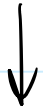
2023年8月19日 11:03



目的ip 192.168.118.2



把目的ip 与 每个Genmask .



得到结果和 Destination 匹配

选择最长匹配项, 下一跳是 Gateway

Kernel IP routing table		
Destination	Gateway	Genmask
0.0.0.0	192.168.118.2	0.0.0.0
169.254.0.0	0.0.0.0	255.255.0.0
172.17.0.0	0.0.0.0	255.255.0.0
192.168.118.0	0.0.0.0	255.255.255.0

→ 0.0.0.0  
→ 192.168.0.0  
→ 192.168.0.0  
→ 192.168.118.0

192.168.1.1 → 0.0.0.0 ✓ → 192.168.118.2  
→ 192.168.0.0 X  
→ 192.168.0.0 X  
→ 192.168.1.0 X



```
liao:~$ sudo route del -net 0.0.0.0 gw 192.168.118.2 dev ens33
[sudo] password for liao:
liao:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
169.254.0.0      0.0.0.0         255.255.0.0     U        1000   0      0 ens33
172.17.0.0       0.0.0.0         255.255.0.0     U         0      0      0 docker0
192.168.118.0    0.0.0.0         255.255.255.0   U        100    0      0 ens33
```

```
liao:~$ ping 14.119.104.254
ping: connect: Network is unreachable
liao:~$ sudo route add -net 0.0.0.0 gw 192.168.118.2 dev ens33
liao:~$ ping 14.119.104.254
PING 14.119.104.254 (14.119.104.254) 56(84) bytes of data.
64 bytes from 14.119.104.254: icmp_seq=1 ttl=128 time=23.0 ms
64 bytes from 14.119.104.254: icmp_seq=2 ttl=128 time=24.9 ms
64 bytes from 14.119.104.254: icmp_seq=3 ttl=128 time=23.8 ms
64 bytes from 14.119.104.254: icmp_seq=4 ttl=128 time=24.6 ms
64 bytes from 14.119.104.254: icmp_seq=5 ttl=128 time=24.2 ms
^C
--- 14.119.104.254 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
```

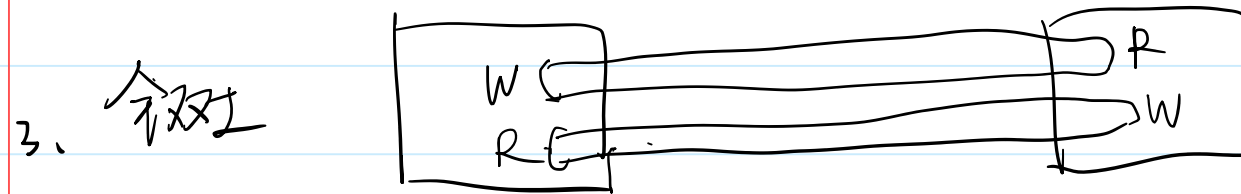
# TCP

2023年8月19日 11:14

"打电话"      A  $\longrightarrow$  B

1. 有连接.      { 断开      无效的交互  
                      已连接      有效的交互.

2. 可靠      丢包会重传

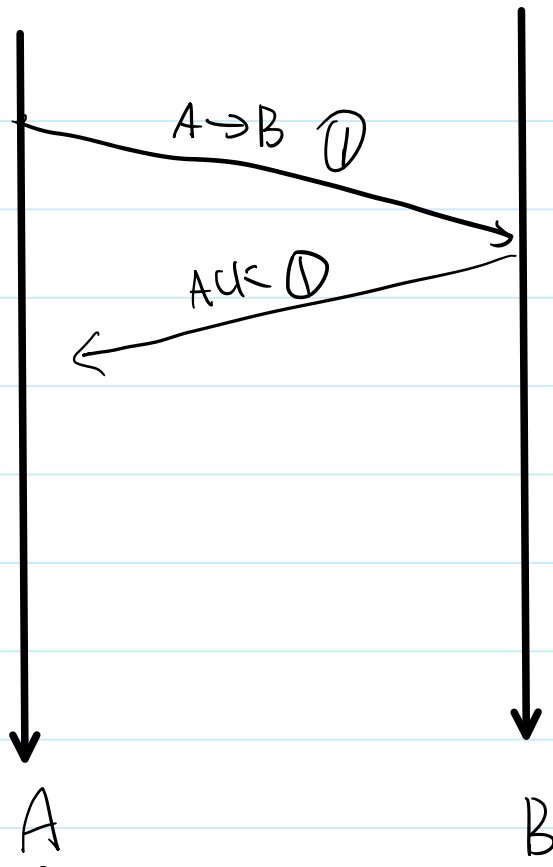


4. 端到端      单播.

# 可靠性

2023年8月19日

11:20

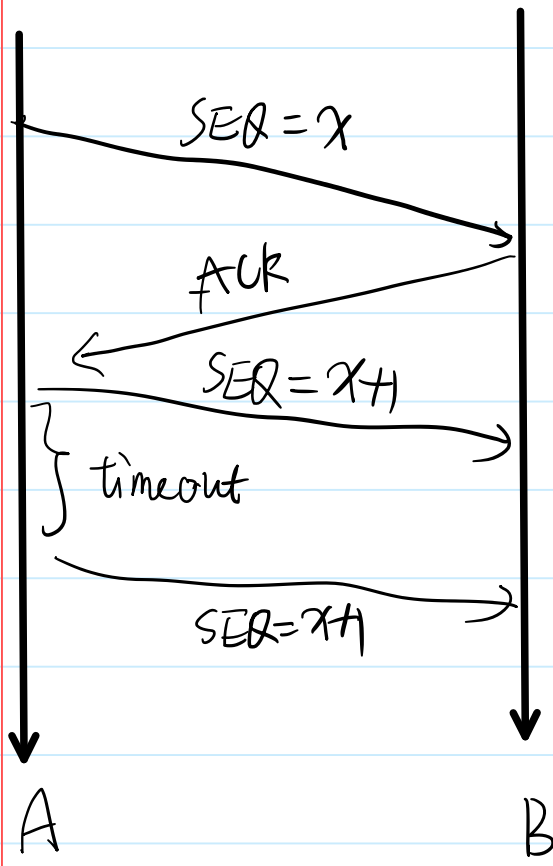


(A → B). B 回复一个 ACK.  
A 未收到 ACK 重传 (A → B)

↪ a. (A → B) 丢了

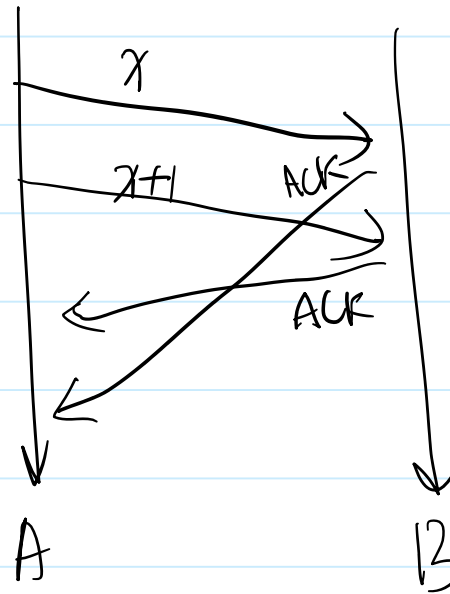
b. ACK 丢了 ⇒ B 收到两次  
(A → B).

↪ 为了让 B 区分重传包和新包重复内容  
引入一个序号机制



定时器 只有发送方有.

ACK不会自动重发. 一定是数据重发, 导致ACK重发



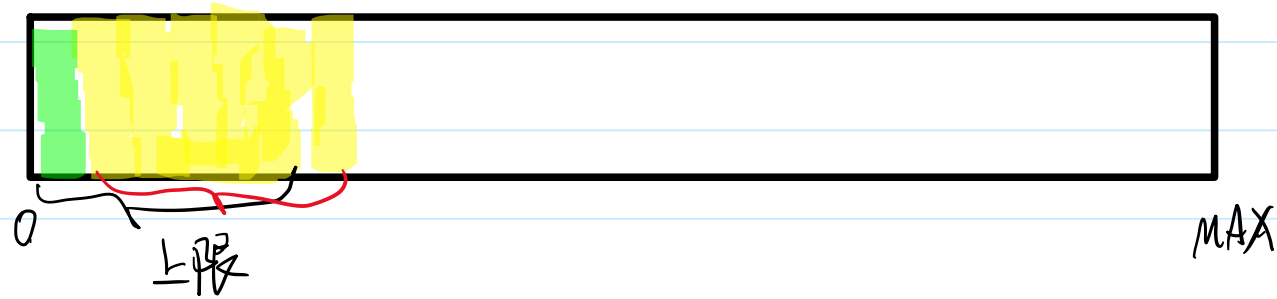
在ACK中包含 SEQ的信息.

# 滑动窗口

2023年8月19日 11:33

已发送并确认的数据有限

序号的集合



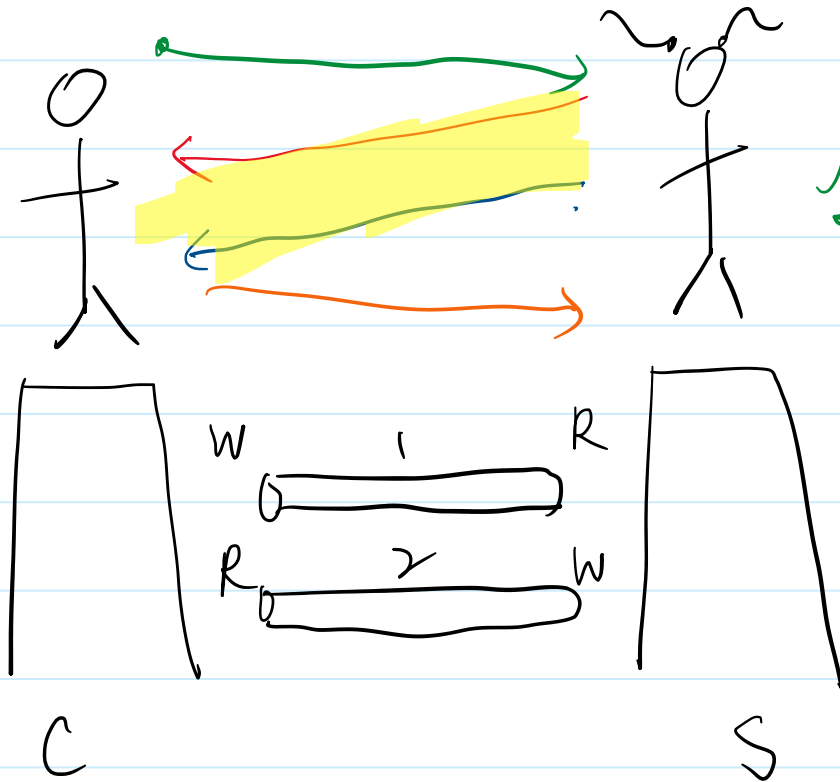
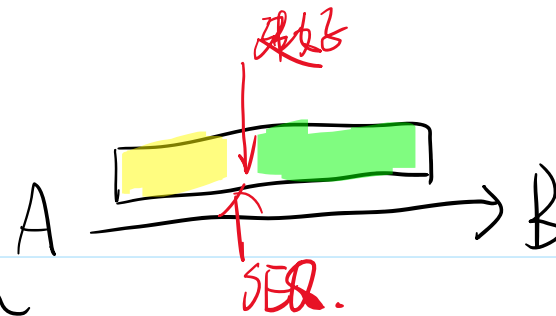
滑动窗口：保持上限，集合平移

→ { 内存限制  
接收方处理能力

# 建立连接

2023年8月19日 11:40

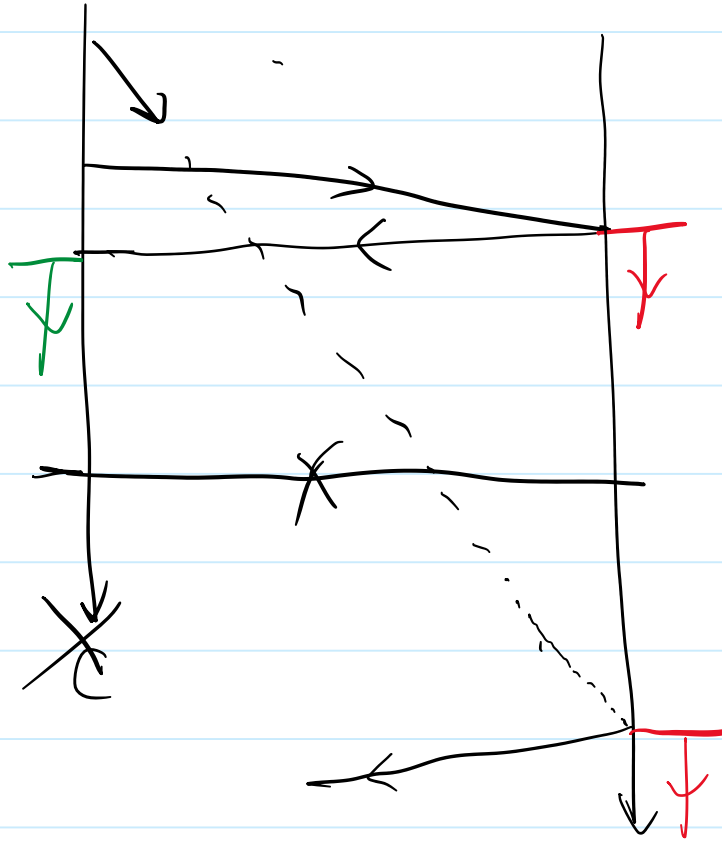
从未连接 → 已连接



- 1 → 发送时. C 知道 1 的 W 已开.  
→ 到达时 S 知道 1 的 R, W 已开
- 2 → 到达时. C 知道 1 的 R 已开  
→ 发送时. S 知道 2 的 W 已开  
→ 到达时. C 知道 2 的 R, W 已开
- 3 → 到达时. S 知道 2 的 R 已开

## 2次握手行不行

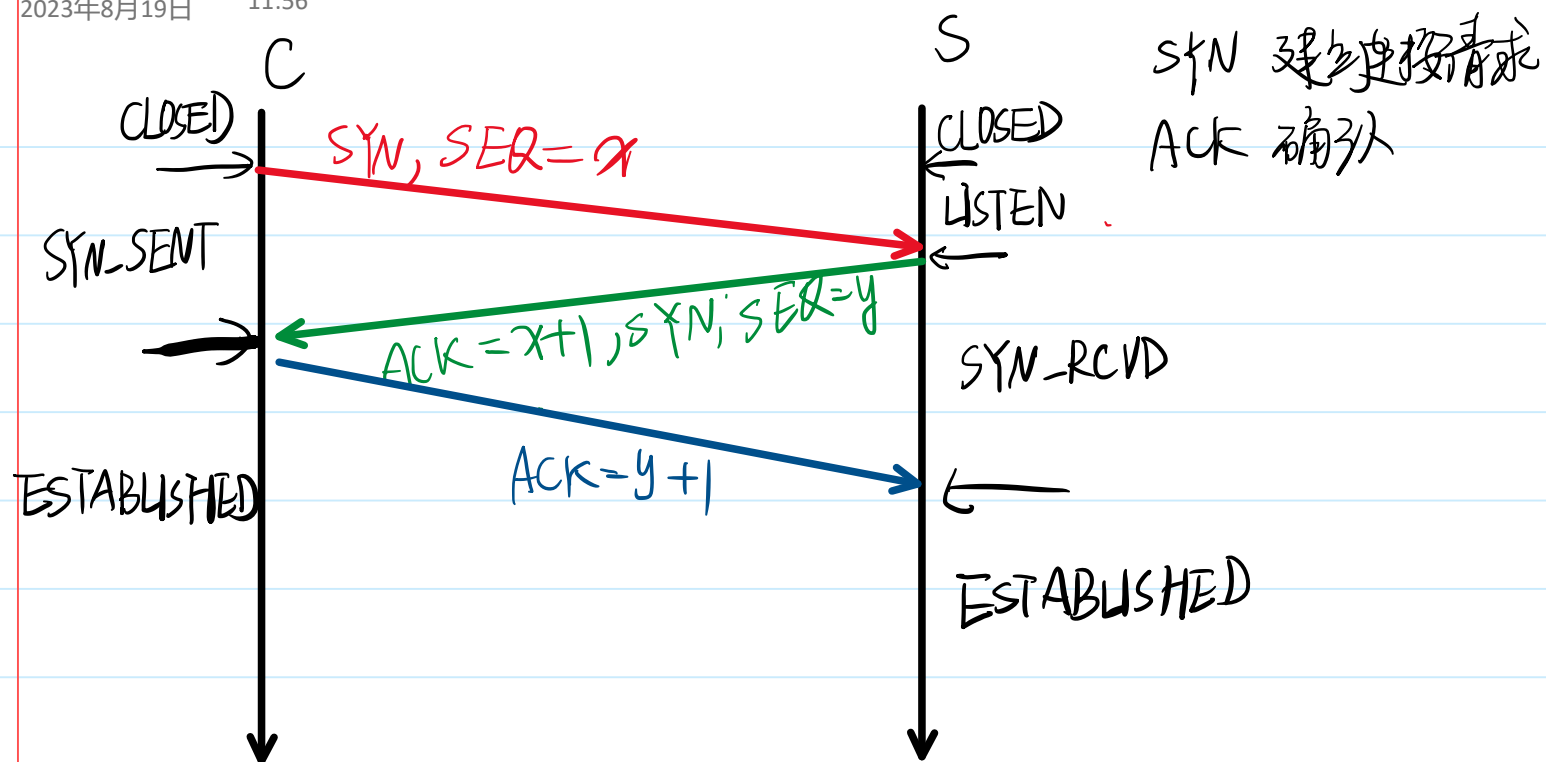
2023年8月19日 11:51



# TCP握手时状态的变化

2023年8月19日

11:56





# netstat

2023年8月19日 12:02

```
liao:LinuxDay20$ netstat -an
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53825	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.53:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:6379	0.0.0.0:*	LISTEN
tcp	1	0	127.0.0.1:57592	127.0.0.1:53825	CLOSE_WAIT
tcp	0	0	127.0.0.1:43460	127.0.0.1:53825	ESTABLISHED
tcp	0	0	192.168.118.128:22	192.168.118.1:4412	ESTABLISHED
tcp	0	0	127.0.0.1:53825	127.0.0.1:43460	ESTABLISHED
tcp	0	316	192.168.118.128:22	192.168.118.1:5045	ESTABLISHED
tcp	1	0	127.0.0.1:57600	127.0.0.1:53825	CLOSE_WAIT