第十六节 拥塞控制与连接管理

一、课程目标

掌握教材 5.8-5.9。

二、课程内容

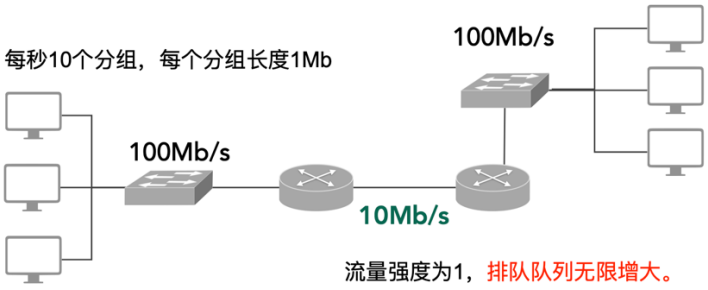
【第一部分：拥塞与拥塞控制】

1、**拥塞**：在某段时间，若通信双方对网络中某资源的需求超过了该资源所能提供的可用部分，**网络性能变坏**。常见原因包括

（1）链路容量不足、资源分配不均衡；

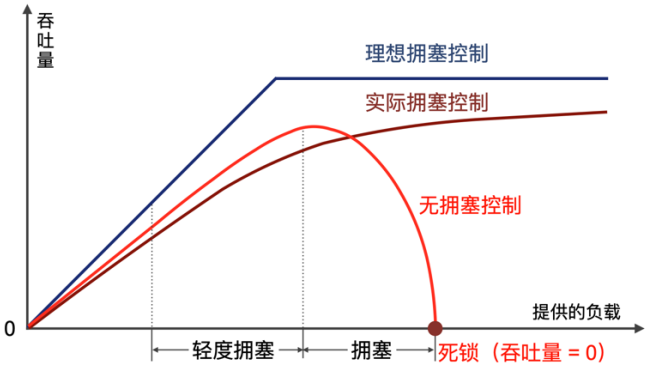
（2）路由器缓存空间、流量分布不均衡；

（3）处理机速度太慢。



网络拥塞的解决是一个系统工程。例如，路由器没有足够的缓存空间，它就会丢弃一些新到的分组。分组被丢弃时，发送这一分组的源点就会重传这一分组，甚至可能还要重传多次。这样会引起更多的分组流入网络和被网络中的路由器丢弃。拥塞引起的重传不会缓解网络的拥塞，反而会加剧网络的拥塞。

2、**拥塞控制**：**防止过多的数据注入到网络中，使网络中的路由器或链路不致过载。**前提是网络能够承受现有的网络负荷。拥塞控制是一个全局性的过程:涉及到所有的主机、所有的路由器，以及与降低网络传输性能有关的所有因素。



3、**拥塞控制的一般原理（解决办法）：**

由于拥塞控制是个动态问题，因此拥塞控制很难设计。主要有两种思路：

（1） 开环控制方法：在设计网络时考虑发生拥塞的因素，力求网络不产生拥塞。

（2） 闭环控制方法：监测网络系统以便检测到拥塞在何时、何处发生；将拥塞发生的信息传送到可采取行动的地方；调整网络系统的运行以解

决出现的问题。

4、拥塞度量指标（指标上升标志拥塞增长），包括

（1） 由于缺少缓存空间而被丢弃的分组的百分数；

（2） 平均队列长度；

（3） 超时重传的分组数；

（4） 平均分组时延；

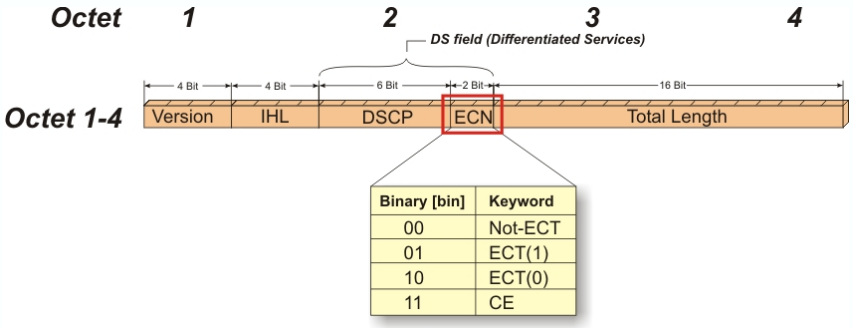
（5） 分组时延的标准差，等等

5、拥塞通知：一般在监测到拥塞发生时，要将拥塞发生的信息传送到产生分组的源站。（当然，通知拥塞发生的分组同样会使网络更加拥塞）

拥塞控制的时机选择：过于频繁，会使系统产生不稳定的振荡；过于迟缓，采取行动又不具有任何价值。

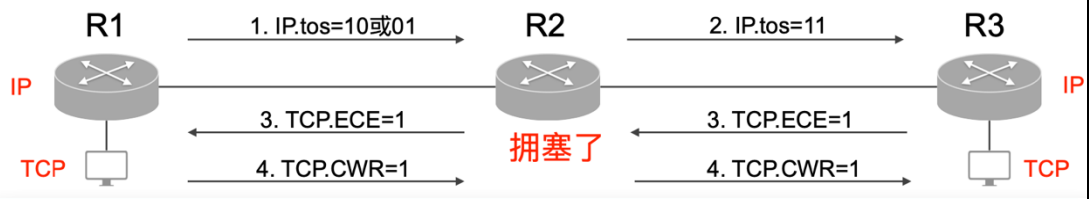
显式拥塞控制 ECN（Explicit Congestion Notification）

RFC 3168 定义，通过在 IP 头部嵌入拥塞指示器和在 TCP 头部嵌入拥塞确认实现。下面是 IP 头部的前四个帧的格式，00 表示无拥塞或者未开启 ECN 功能，01/10 代表使能 ECN 功能，11 表示拥塞：



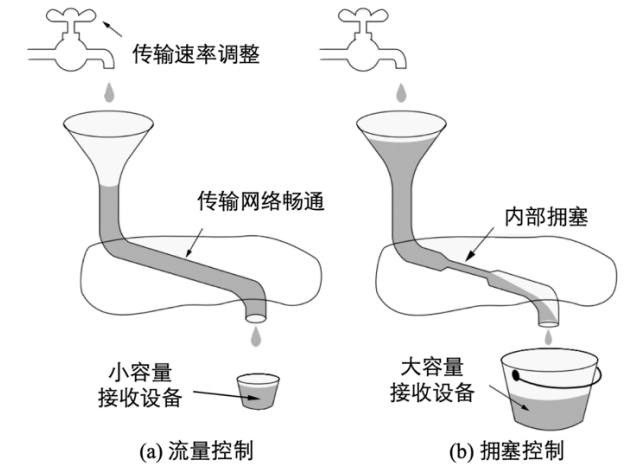
处理流程：

发送端设置 IP 首部为 10 或者 01，表示使能 ECN。数据包经过拥塞处时，由路由器或交换机更改 ECN 为 11；接收端收到 ECN 位为 11 的 TCP 报文，设置 TCP 首部中的 ECE 标志；发送端收到设置 ECE 标志的 ACK 时，减小发送窗口，并在发送下一个 TCP 中设置 CWR 标志。接收方停止设置 ECE 标志。



6、流量控制与拥塞控制的区别

流量控制	拥塞控制
流量控制是端到端的问题(接收端控制发送端)，点对点通信量的控制。抑制发送端发送速率，以便使接收端来得及接收。	初衷是防止过多数据注入到网络，使网络中的路由器或链路不致过载。 但是某些拥塞控制算法是向发送端发送控制报文，并告诉发送端，网络已出现麻烦，必须放慢发送速率，与流量控制是很相似。



【第二部分：TCP 拥塞控制算法】

7、TCP 拥塞控制算法基本概念：

（1）TCP 采用基于窗口的方法进行拥塞控制，属于闭环控制方法。

（2）TCP 发送方维持一个拥塞窗口 CWND (Congestion Window)，拥塞窗口的大小取决于网络的拥塞程度，并且动态变化；发送端利用拥塞窗口根据网络的拥塞情况调整发送的数据量；**网络没有出现拥塞，拥塞窗口增大一些**，以便发送更多的分组，提高网络的利用率；**网络出现拥塞或有可能出现拥塞，拥塞窗口减小一些**，减少注入到网络中的分组数。

（3）TCP 拥塞控制算法包括四个阶段：慢开始 (slow-start)、拥塞避免 (congestion avoidance)、快重传 (fast retransmit)、快恢复 (fast recovery)。

8、慢开始 (Slow Start)，目的是用来确定网络的负载能力或拥塞程度。

基本思路：由小到大逐渐增大拥塞窗口数值。当主机在已建立的 TCP 连接上开始发送数据时，并不清楚网络当前的负荷情况。如果立即把大量数据字节注入到网络，那么就有可能引起网络发生拥塞。

参数：

（1）发送方最大报文段 Sender Maximum Segment Size（通常取 MTU 或 MSS)：发送端可传输的最大 Segment 的大小。

MTU：以太网帧长度最小为 64 字，最大为 1518 字节，某些场景下巨型帧（Jumbo Frame）能达到 9000 字节。去掉以太网头部 14B 和帧校验 4B，以太网 MTU 为 1500 字节。

MSS：MSS = MTU - IP header 头大小 - TCP 头大小。其中，IPv4 头固定 20 字节，TCP 头最小为 20 字节，因此，TCP MSS 最大为 1460 字节。

（2）慢开始门限 ssthresh：一个状态变量，防止拥塞窗口 CWND 增长过大引起网络拥塞。

（3）拥塞窗口 CWND：单次允许发送的数据多少。

算法流程：

（1）初始化拥塞窗口 CWND 为 1-2 个 SMSS（旧规定）或者 **2-4 个 SMSS（RFC 5681 新规定）。**

SMSS > 2190 字节，CWND = 2 × SMSS 字节，且不得超过 2 个报文段；

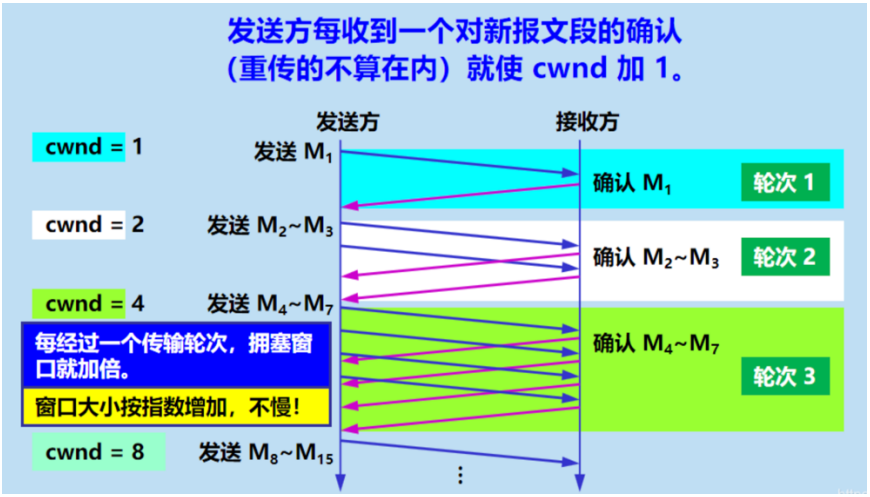
SMSS > 1095 字节且 SMSS ≤ 2190 字节，CWND = 3 × SMSS 字节，且不得超过 3 个报文段；

SMSS ≤ 1095 字节，CWND = 4 × SMSS 字节，且不得超过 4 个报文段;

（2）在每收到一个对新的报文段的确认后，把拥塞窗口增加最多一个 SMSS 的数值（窗口大小实际按指数增加，并不慢）：

拥塞窗口 CWND 每次的增加量 = min (N, SMSS)

其中 N 是原先未被确认的、但现在被刚收到的确认报文段所确认的字节数；当 N<SMSS 时，拥塞窗口每次的增加量小于 SMSS；采用逐步增大发送方的拥塞窗口的方法，注入数据至网络的速率更加合理。



（3）当 CWND 逐渐增大，需要其与慢开始门限 ssthresh 比较，判断是否继续保持慢开始还是进入拥塞避免。

当 CWND<ssthresh 时，使用慢开始算法;

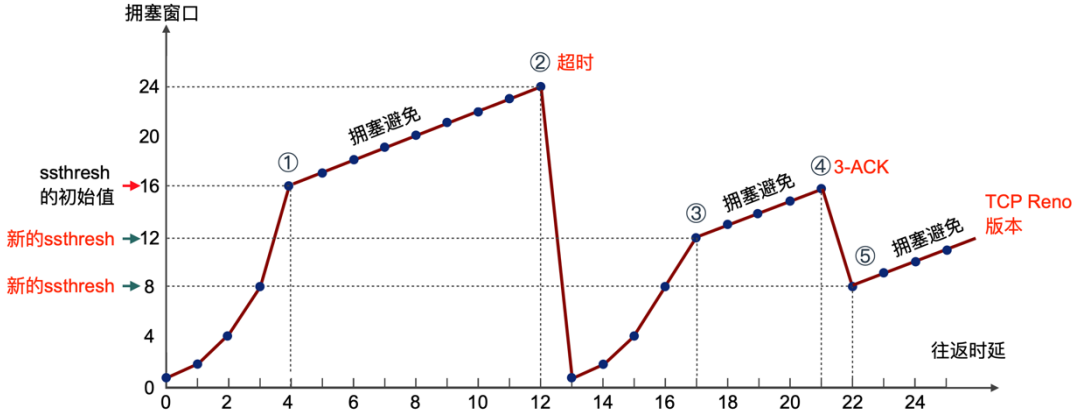
当 CWND>ssthresh 时，停止使用慢开始算法而改用拥塞避免算法;

当 CWND=ssthresh 时，既可使用慢开始算法，也可使用拥塞避免算法。

（4）无论慢开始阶段还是在拥塞避免阶段，只要发送方判断网络出现拥塞(重传定时器超时)，令 **ssthresh = max(CWND/2，2)且 CWND = 1，继续执行慢开始算法**。从而迅速减少主机发送到网络中的分组数，使得发生拥塞的路由器有足够时间把队列中积压的分组处理完毕。

9、拥塞避免 (Congestion avoidance)，目的是用来确定网络的负载能力或拥塞程度。

基本思路：让拥塞窗口 CWND 缓慢地增大，避免出现拥塞；每经过一个 RTT，拥塞窗口 CWND=CWND+1，使拥塞窗口 CWND 按线性规律缓慢增长，在拥塞避免阶段，具有“加法增大” (Additive Increase) 的特点。



在执行慢开始算法时，拥塞窗口 CWND=1，发送第一个报文段

发送方每收到一个对新报文段的确认 ACK，就把拥塞窗口值加 1，然后开始下一轮的传输（请注意，横坐标是传输轮次，不是时间）。因此拥塞窗口 CWND 随着传输轮次按指数规律增长

当拥塞窗口 CWND 增长到慢开始门限值 ssthresh 时（图中的点❶，此时拥塞窗口 CWND=16），就改为执行拥塞避免算法，拥塞窗口按线性规律增长

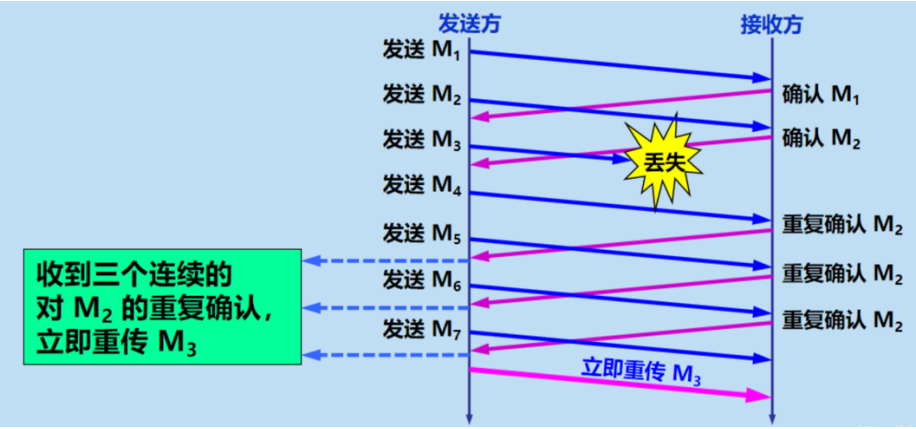
当拥塞窗口 CWND=24 时，网络出现了超时（图中的点❷），发送方判断为网络拥塞。于是调整门限值 ssthresh=CWND/2=12，同时设置拥塞窗口 CWND=1，进入慢开始阶段

按照慢开始算法，发送方每收到一个对新报文段的确认 ACK，就把拥塞窗口值加 1 当拥塞窗口 CWND=ssthresh=12 时（图中的点❸，这是新的 ssthresh 值），改为执行拥塞避免算法，拥塞窗口按线性规律增大

当拥塞窗口 CWND=16 时（图中的点❹），出现了一个新的情况，就是发送方一连收到 3 个对同一个报文段的重复确认（图中记为 3-ACK）。发送方改为执行快重传和快恢复算法。

10、快重传算法，目的是与快恢复配合防止因为报文段丢失重新进入到慢启动的过程

基本思路：发送方只要一连收到三个重复确认，就知道接收方确实没有收到报文段，因而应当立即进行重传（即“快重传”），这样就不会出现超时，发送方也不就会误认为出现了网络拥塞。



发送方发送 M3 时传输过程中 M3 丢失，但是接收方接收到了 M4 报文段，那么接收方必须对 M2 进行重复的确认

当发送方接收到连续 3 个对 M2 的确认后，发送方就知道了接收方没有接收到 M3，应当立即重传 M3，这也就是快重传的由来

这样也不会出现超时，发送方也不会认为这是网络拥塞。

11、快恢复算法，目的是与快重传配合防止因为报文段丢失重新进入到慢启动的过程

基本思路：当发送端收到连续三个重复的确认时，由于发送方现在认为网络很可能没有发生拥塞，因此现在不执行慢开始算法，而是执行快恢复算法 FR (Fast Recovery) 算法。具体操作包括：

（1）慢开始门限 ssthresh=当前拥塞窗口 CWND/2；

（2）新拥塞窗口 CWND=慢开始门限 ssthresh；

（3）开始执行拥塞避免算法，使拥塞窗口缓慢地线性增大。

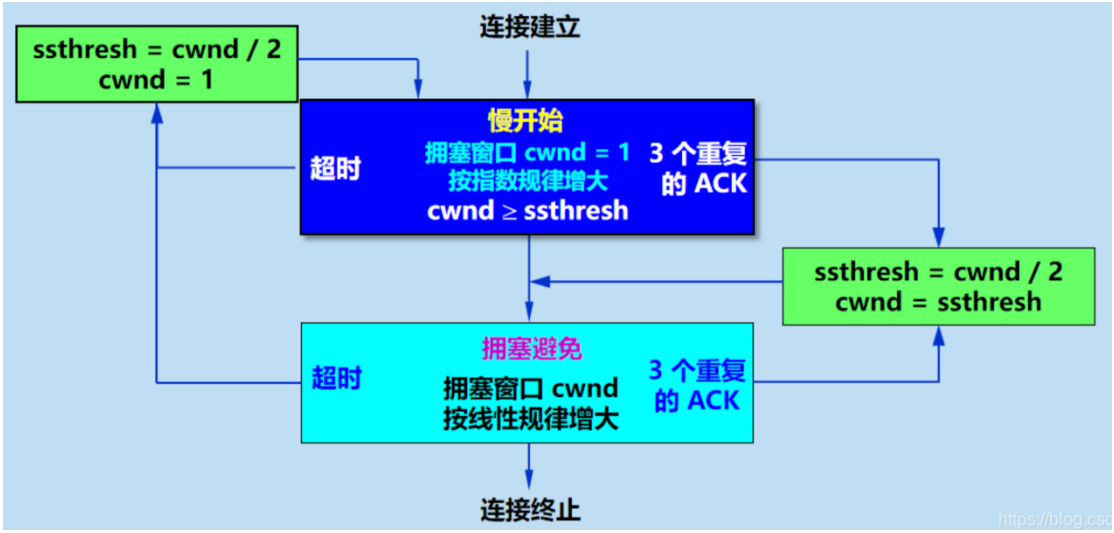
12、加法增大，乘法减小（AIMD）算法

在拥塞避免阶段，拥塞窗口是按照线性规律增大的。这常称为“加法增大”AI (Additive Increase)。

当出现超时或 3 个重复的确认时，就要把门限值设置为当前拥塞窗口值的一半，并大大减小拥塞窗口的数值。这常称为“乘法减小”MD (Multiplicative Decrease)。

二者合在一起就是所谓的 AIMD 算法。

13、拥塞控制流程图



慢开始，每个 RTT 内，每收到一个确认，拥塞窗口值加 1；

拥塞避免，每个 RTT 内，不论收到多少确认，拥塞窗口加 1；超时重传或收到三个连续重复确认，门限值=cwnd*0.5，即加法增大，乘法减小；

快重传，收到三个连续重复确认，立即重传丢失的报文段；

快恢复，收到三个连续重复确认，门限值=cwnd*0.5，执行拥塞避免算法。

14、发送窗口的上限值

发送方的发送窗口的上限值应当取为接收方窗口 rwnd 和拥塞窗口 cwnd 这两个变量中较小的一个。

发送窗口上限值=Min(接收窗口，拥塞窗口)=拥塞窗口

当 rwnd<cwnd 时，是接收方的接收能力限制发送窗口的最大值；

当 cwnd<rwnd 时，则是网络的拥塞限制发送窗口的最大值。

15、主动队列管理 AQM

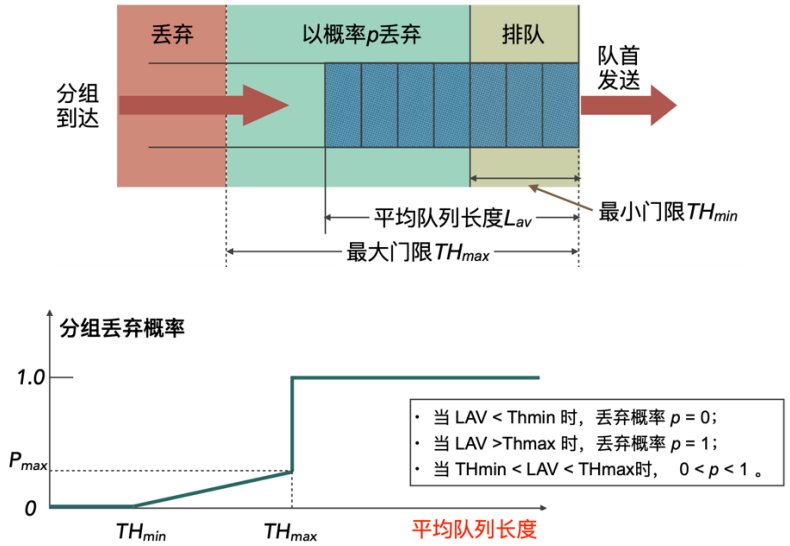
背景：在最简单的情况下，路由器的队列通常采用先进先出(FIFO)规则与尾部丢弃策略(tail-drop policy)。当队列已满时，以后再到达的所有分组将都被丢弃。由于是不区分 TCP 连接的无差别丢弃，这可能导致所有 TCP 连接同一时间进入慢开始状态，通信量突然下降，正常后，通信量突然增大，称为全局同步问题。

基本原理：不要等到路由器的队列长度已经达到最大值时才不得不丢弃后面到达的分组，在队列长度达到某个值得警惕的数值时（即当网络拥塞有了某些拥塞征兆时），就主动丢弃到达的分组。AQM 有不同实现方法，代表性方案是随机早期检测 RED(Random Early Detection)。

随机早期检测 RED

将队列的平均队长作为决定拥塞避免机制是否应被处罚的随机函数的参数，增

加了在队列长度变得太大之前平滑瞬时拥塞的可能性，减少了同时使多个流受分组丢弃影响的可能性。



【连接管理】

16、TCP 连接的三个阶段：连接建立、数据传送、连接释放。

17、连接建立作用：

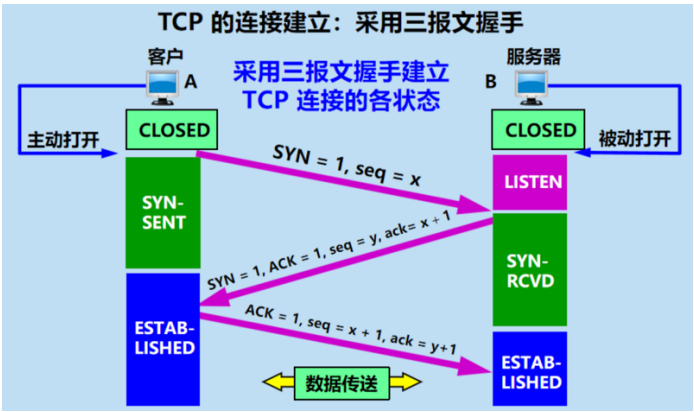
（1）使每一方能够确知对方的存在。

（2）允许双方协商一些参数(如最大窗口值、是否使用窗口扩大选项和时间戳选项以及服务质量等)；

（3）对运输实体资源(如缓存大小、连接表中的项目等)进行分配。

18、TCP 连接的建立采用客户服务器方式：主动发起连接建立的应用进程叫做客户(client)；被动等待连接建立的应用进程叫做服务器(server)。

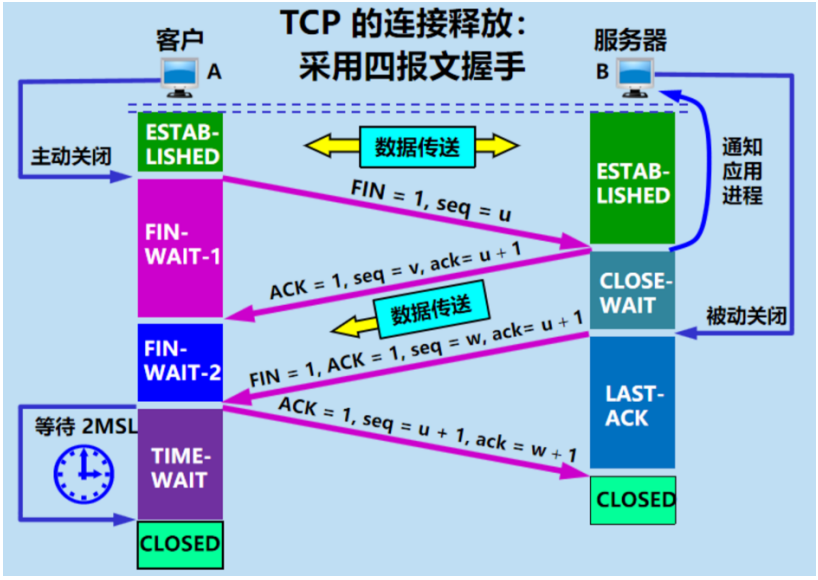
19、三次握手：TCP 建立连接的过程叫做握手，握手需要在客户和服务器之间交换三个 TCP 报文段。称之为三报文握手。采用三报文握手主要是为了防止已失效的连接请求报文段突然又传送到了，因而产生错误。



注意：大写的 ACK 和小写的 ack 代表的含义不同，大写 ACK 代表确认位，小写 ack 代表确认号，ack=x+1 代表之前序号 x 报文已经正确收到了，并期待收到 x+1 报文，服务器 B 也选择了序号 seq=y，这里的 x 和 y 代表双方各自发送的序号的独立的。

20、连接释放：TCP 连接释放过程比较复杂，数据传输结束后，通信的双方

都可释放连接。TCP 连接释放过程是四报文握手（又称**四次挥手**）。



注意：如果 Client 直接进入 CLOSED 状态，Server 有可能没有收到 Client 最后回复的 ACK，Server 超时之后继续发送 FIN。但 Client 已经进入 CLOSED 状态了，服务器重发的 FIN 找不到对应的连接，Server 就会收到 RST 而不是 ACK。这种情况不会造成数据丢失，但导致 TCP 协议不符合可靠连接的要求；Client 不是直接进入 CLOSED 状态，而是要保持 TIME_WAIT，当再次收到 FIN 的时候，能够保证对方收到 ACK，最后正确地关闭连接。

如果 Client 直接进入 CLOSED 状态之后，又向 Server 发起一个新连接，有可能新连接和老连接的端口号是相同的。如果前一次连接的某些数据仍然滞留在网络中，这些数据在建立新连接之后才到达 Server，TCP 协议就认为数据是属于新连接的，这样就和真正的新连接的数据包发生混淆。Client 在 TIME_WAIT 状态等待 2 倍 MSL，这样可以保证本次连接的所有数据都从网络中消失(Server 与 Client 的连接已经释放，这期间 Server 收到的 TCP 报文段，直接丢弃)。

MSL 是报文最长寿命时间，指一个 TCP 报文段在网络中最长的生存时间，它是一个估计值，通常设置为 2 分钟。

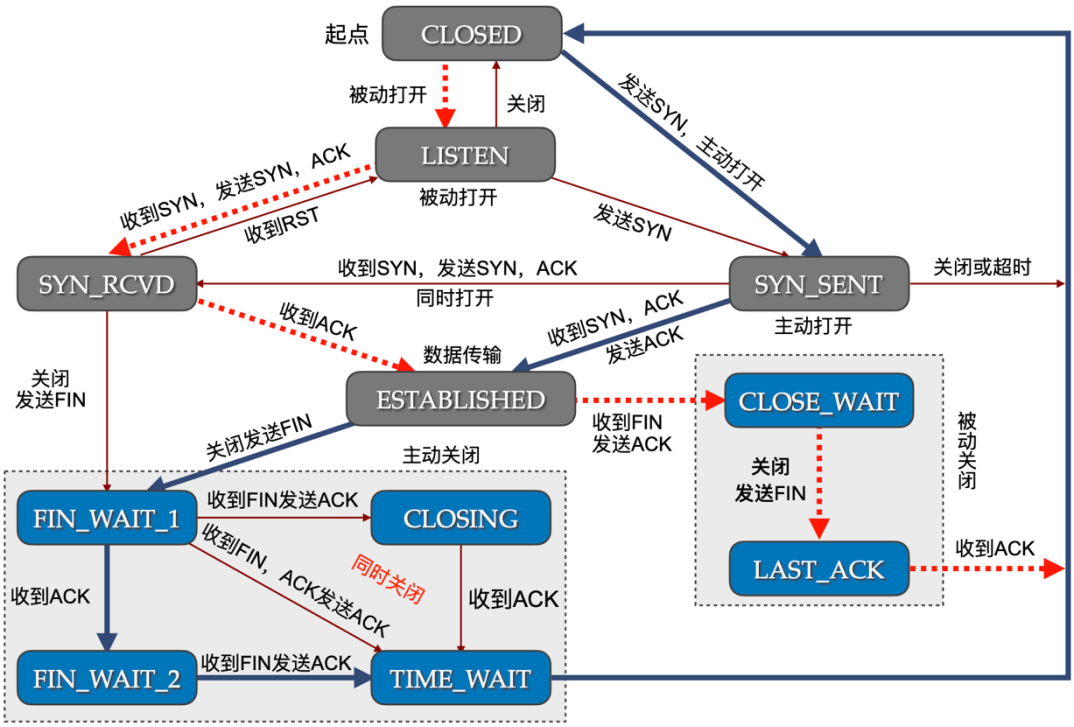
21、保活计时器：用来防止在 TCP 连接出现长时期的空闲。通常设置为 2 小时。若服务器过了 2 小时还没有收到客户的信息，它就发送探测报文段。若发送了 10 个探测报文段(每一个相隔 75 秒)还没有响应，就假定客户出了故障，因而就终止该连接。

22、TCP 有限状态机

其中，粗实线箭头表示对客户进程的正常变迁。

粗虚线箭头表示对服务器进程的正常变迁。

细线箭头表示异常变迁。



三、重点习题

P253：全部

四、参考资料

https://www.bilibili.com/video/BV1oo4y1j7Eb/?vd_source=ae7c78da643dd0aeb736fa2a6cca9565