

## 如何实现拥塞控制?

如何找到合适的窗口?

## 网络模型

5 Mbit/s

RTT = 100 ms

## 什么是合适的拥塞窗口?

理想的未确认字节总数 = 带宽 × 延迟积 (BDP)。

使链路始终忙碌, 瓶颈队列中没有积压。

对于单个流, BDP 是该流的理想窗口。

(N 个流: 每个流可以使用  $cwnd = BDP/N$ )

无丢包窗口: 任何小于  $BDP + \text{最大队列大小}$  的值。

但是……理想  $cwnd$  的值在运行时是未知的!

TCP 发送方不知道:

瓶颈链路速率

最小 RTT (无排队时)

争夺同一瓶颈的其他流的数量

那么……如何在没有全知视角的情况下逼近正确的拥塞窗口?

## 一种可能: 成功时增加, 丢包时减少

从较小的 cwnd 值开始 (例如 3 个段)

成功时 (段被完全确认), 每个 RTT 增加 1 个段

每确认一个字节:  $cwnd += (\text{段大小}) / cwnd$

丢包时, 假设发生拥塞。将 cwnd 减半!

丢包推断条件:

段在很久以前发送, 仍未被确认

或者多个后来发送的段已被确认

这被称为“加性增、乘性减”算法 (AIMD)。

AI = +1, MD = /2

问: 为什么丢包时减少幅度这么大?

### 一种可能: 成功时增加, 丢包时减少 (续)

从较小的 cwnd 值开始 (例如 3 个段)

成功时 (段被完全确认), 每个 RTT 增加 1 个段

每确认一个字节:  $cwnd += (\text{段大小}) / cwnd$

丢包时, 假设发生拥塞。将 cwnd 减半!

丢包推断条件:

段在很久以前发送, 仍未被确认

或者多个后来发送的段已被确认

这被称为 “加性增、乘性减” 算法 (AIMD)。

AI = +1, MD = /2

问: 为什么丢包时减少幅度这么大?

## 慢启动: 初始阶段的指数增长!

从  $cwnd = 3$  个段开始, 以每 RTT 增加 1 个段的速率增长实在太慢了  
但从  $cwnd =$  接收方窗口开始又不可接受!

折中方案称为“慢启动”:

起始  $cwnd = 3$  个段

成功时 (段被完全确认), 每个字节确认后增加一个字节

每确认一个字节:  $cwnd++$

首次丢包时, 将  $cwnd$  减半并转为 AIMD。

问: 路由器缓冲区 (最大队列) 的理想值是多少?

5 Mbit/s

RTT = 100 ms

多个 TCP 流的情况呢?

请在线观看 (视频 4-4)。