

## 为什么需要拥塞控制?

第一部分: 模型与崩溃

## TCP 与流量控制

TCP 提供流量控制的双向字节流

“流量控制”：发送方尊重接收方的容量

但是.....网络的容量呢?

## 单流、单跳模型

## 从发送方的角度, 数据包的三个可能位置

1. 在瓶颈队列中
2. 在链路上传输中
3. 在接收方, 确认信息正在传回发送方







## 窗口: 对未确认字节数的上限

接收方的窗口大小限制了从发送方角度看的未确认字节数。

未确认意味着已发送, 且未被确认或判定为丢失。

问: 如果窗口大小非常小 (例如 1 字节) 会怎样?

问: 如果窗口大小非常大呢?

如果接收方的窗口大小非常大呢?

发送方传输了过多的段。大部分溢出路由器的队列并被丢弃。

我们称之为“拥塞”。

发送方必须一次又一次地重新发送相同的字节。最终, 字节流从接收方的 TCP 正确输出。

问: 这为什么不好?

## 无限制发送的问题: 崩溃与公平性

1. 迫使路由器丢弃大量数据包可能导致“拥塞崩溃”。

需求很大, 但网络没有做有用的工作。

2. 当某些流发送过多时, 其他流会被饿死。

网络表现出糟糕的“公平性”。

**崩溃!**

**崩溃!**

## 导致崩溃的一个简单方法

## 为什么需要拥塞控制?

第二部分: 公平性与目标

## 无限制发送的问题: 崩溃与公平性

1. 迫使路由器丢弃大量数据包可能导致“拥塞崩溃”。

需求很大, 但网络没有做有用的工作。

2. 当某些流发送过多时, 其他流会被饿死。

网络表现出糟糕的“公平性”。

公平性: 什么是划分网络的正确方式?

网络资源是有限的。

流共享同一网络。它们不可能都获得全部资源。

什么是分配这块蛋糕的最佳方式?



## 资源分配示例

A→a B→b C→c 总计

20 10 0 30 (最大利用率)

10 0 10 20 (对 C 最优)

0 0 20 10 (崩溃! )

15 5 5 25 (最大最小公平: 最差结果尽可能好)

16 6 4 26 (比例公平: 改善 x 需要造成  $>1/x$  的损害)

## 资源分配的数学

## Alpha 公平性

## 其他群体目标

最小化流完成时间 (平均下载)

最小化页面加载时间 (包含多个下载的网站)

最大化效能 (= 吞吐量 / 延迟)

.....

## 本单元的其余内容

让流共享网络并防止崩溃的算法被称为“拥塞控制”。

在网络中, 几乎任何涉及去中心化资源分配的问题 = 拥塞控制。

接下来的重要问题:

窗口大小应该是多少?

流应该如何学习合适的窗口大小?