

## 6. 拥塞控制原理

### 6.1 拥塞：

非正式的定义："太多的数据需要网络传输，超过了网络的处理能力"

与流量控制不同

拥塞的表现

分组丢失（路由器缓冲区溢出）

分组经历比较长的延迟（在路由器的队列中排队）

网络TOP10的问题

拥塞的原因/代价：场景1

2个发送端，2个接收端

一个路由器，具备无限大的缓冲

输出链路带宽：R

没有重传

每个连接的最大吞吐量： $R/2$

代价：当进入的速率接近链路带宽R时，延迟增大（流量强度趋向1）

拥塞的原因/代价：场景2

一个路由器，有限的缓冲

分组丢失时候，发送端重发

应用层的输入=应用层的输出

传输层的输入包括重传>应用层的输入

理想化：发送端有完美的信息

发送端知道什么时候路由器的缓冲是可用的

只在缓冲区可用的时候发送

不会丢失

传输层的输入包括重传=应用层的输入

掌握信息丢失

分组可以丢失，在路由器由于缓冲区满而被丢弃

如果知道分组丢失，发送方重新分组

传输层的输入包括重传>应用层的输入

代价：

为了达到一个有效输出，网络需要做更多的工作（大量丢失造成的重传）

没有必要的重传，链路中包括了多个分组的拷贝

是那些没有丢失，经历的时间比较长（拥塞状态）但是超时的分组

降低了的"goodput"

拥塞的原因/代价：场景3

见图

入网拥塞出现死锁（只进不出）

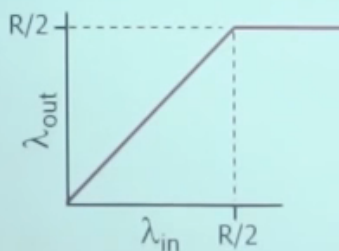
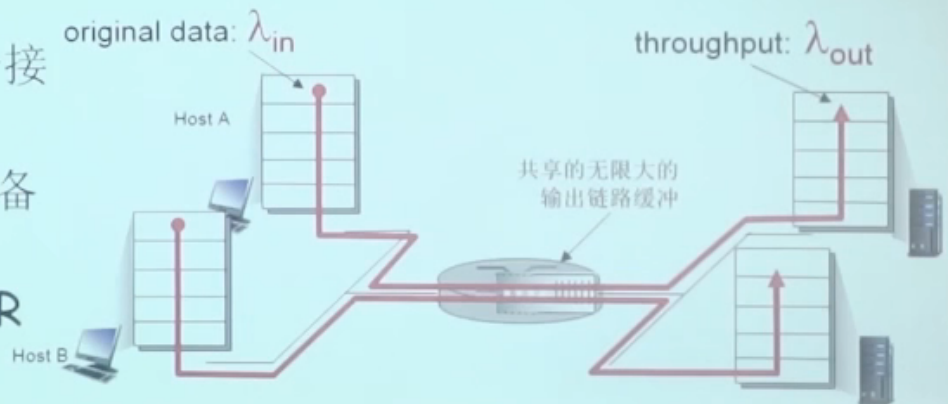
转发少的节点优先，导致转发多的节点需要等待，等待时间一就就超时重发

代价：

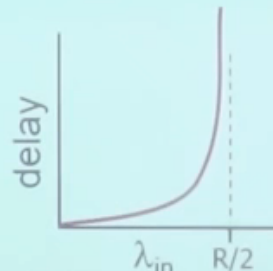
当分组丢失时，任何"关于这个分组的上传传输能力"都被浪费了

# 拥塞的原因/代价: 场景1

- 2个发送端, 2个接收端
- 一个路由器, 具备无限大的缓冲
- 输出链路带宽:  $R$
- 没有重传



❖ 每个连接的最大吞吐量:  $R/2$

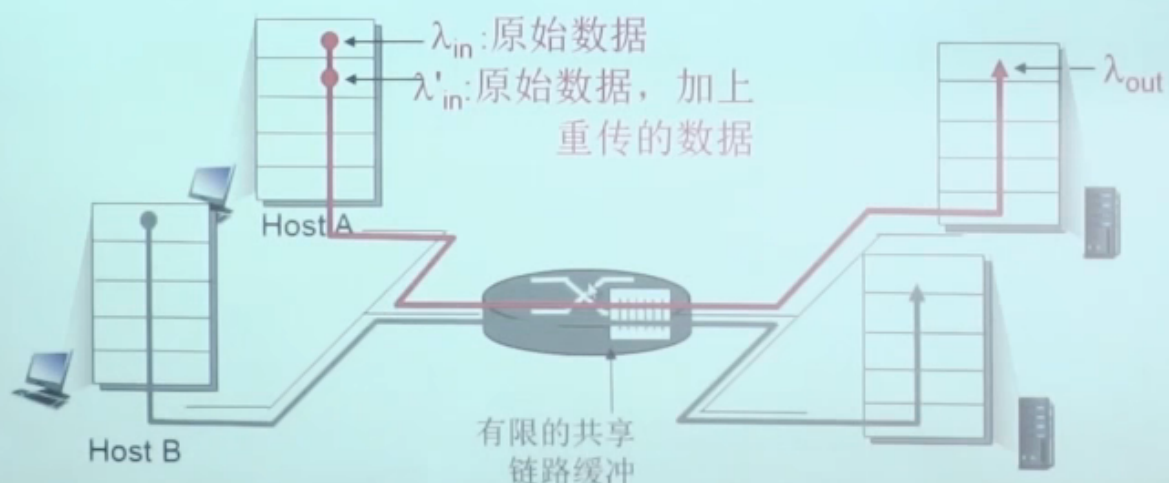


❖ 当进入的速率 $\lambda_{in}$ 接近链路链路带宽 $R$ 时, 延迟增大

Transport Layer 3-109

# 拥塞的原因/代价: 场景2

- 一个路由器, 有限的缓冲
- 分组丢失时, 发送端重传
  - 应用层的输入=应用层输出:  $\lambda_{in} = \lambda_{out}$
  - 传输层的输入包括重传:  $\lambda'_{in} \geq \lambda_{in}$



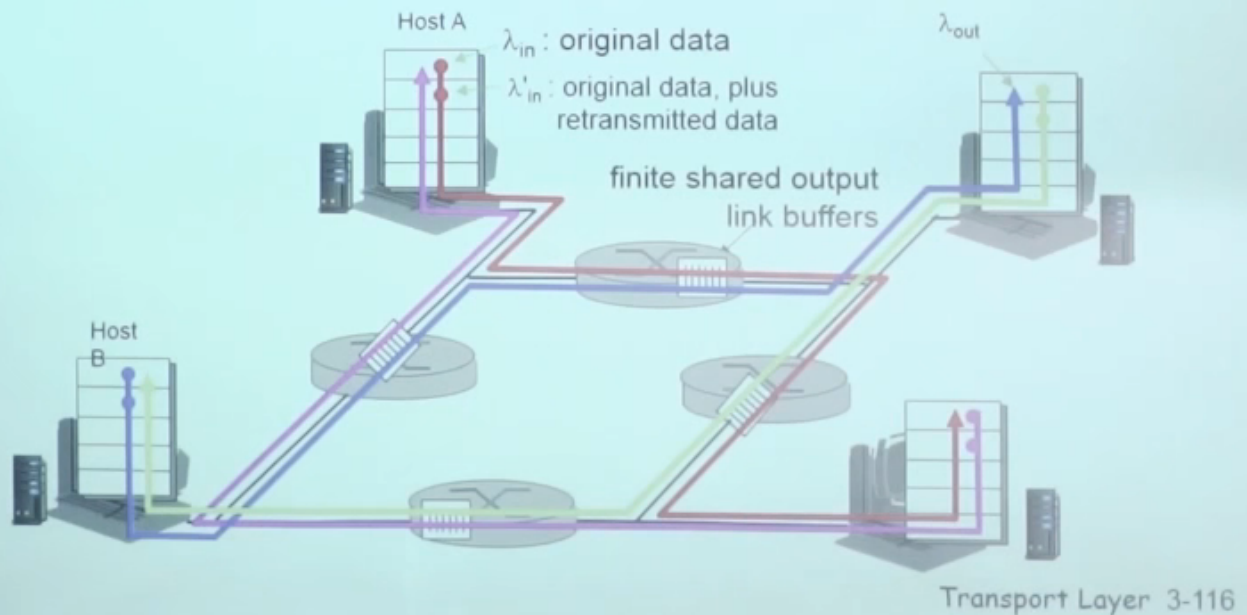
Transport Layer 3-110

# congestion的原因/代价: 场景3

- ❑ 4个发送端
- ❑ 多重路径
- ❑ 超时 / 重传

Q: 当  $\lambda_{in}$   $\lambda'_{in}$  增加时, 会发生什么?

A: 当红色的  $\lambda'_{in}$  增加时, 所有到来的蓝色分组都在最上方的队列中丢弃了, 蓝色吞吐  $\rightarrow 0$



## 6.2 拥塞控制的方法

拥塞控制的目的: 网络不拥塞的情况下, 尽可能的提升传输速率

如果没有网络拥塞

超时重发/丢失重发大量增多

--->传输层输入占据大量带宽, 导致输出变慢

--->加剧超时/丢失, 更多输入

--->到达极致就是只进不出, 形成死锁

2种常用的拥塞控制方法:

端到端拥塞控制:

没有来自网络的显式反馈

端系统根据延迟和丢失时间推断是否拥塞

超时重发, 丢失重发, 快速重发都是拥塞的信号

TCP采用的方法

网络辅助的拥塞控制:

路由器提供给端系统以反馈信息

单个bit置位, 显示有拥塞 (SNA, DECbit, TCP/IP ECN, ATM)

显式提供发送端可以采用的速率

ATM ABR拥塞控制: ATM: 异步传输网络

ABR: available bit rate:

弹性服务

如果发送端的路径"轻载"

发送方使用可用带宽

如果发送端的路径拥塞:

发送方限制其发送的速率到一个最小的保障速率上

RM(资源管理)信元

由发送端发送，在数据信元中间隔插入

RM信元中的比特被交换机设置（"网络辅助"）

NI bit: no increase in rate（轻微拥塞）速率不要增加了

CI bit: congestion

ER：最多为你这对连接提供多少带宽

indication: 拥塞指示

发送端发送的RM信元被接收端返回，接收端不做任何改变