



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

立足航天，服务国防，面向国民经济主战场

计算机网络之探赜索隐

主讲人：聂兰顺

本讲主题

TCP拥塞控制



TCP拥塞控制的基本原理

❖ Sender限制发送速率

$\text{LastByteSent} - \text{LastByteAcked} \leq \text{CongWin}$

$$\text{rate} \approx \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/sec}$$

❖ CongWin:

- 动态调整以改变发送速率
- 反映所感知到的网络拥塞

问题：如何感知网络拥塞？

❖ Loss事件=timeout或3个重复ACK

❖ 发生loss事件后，发送方降低速率

如何合理地调整发送速率？

❖ 加性增—乘性减: AIMD

❖ 慢启动: SS



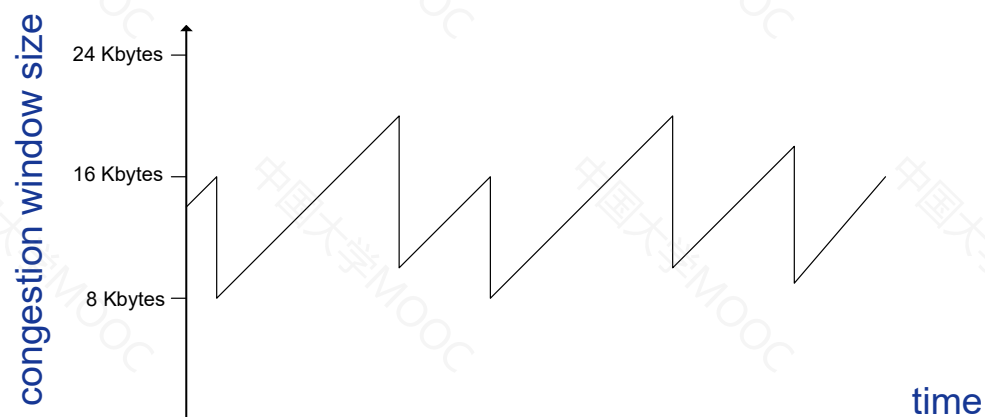
加性增—乘性减: AIMD

❖ **原理:** 逐渐增加发送速率, 谨慎探测可用带宽, 直到发生loss

❖ **方法:** AIMD

- Additive Increase: 每个RTT将CongWin增大一个MSS——拥塞避免
- Multiplicative Decrease: 发生loss后将CongWin减半

锯齿行为: 探测
可用带宽



TCP慢启动: SS

❖ TCP连接建立时, CongWin=1

- 例: MSS=500 byte,
RTT=200msec
- 初始速率=20k bps

❖ 可用带宽可能远远高于初始 速率:

- 希望快速增长

❖ 原理:

- 当连接开始时, 指数性增长

Slowstart algorithm

```
initialize: Congwin = 1
for (each segment ACKed)
    Congwin++
until (loss event OR
      CongWin > threshold)
```

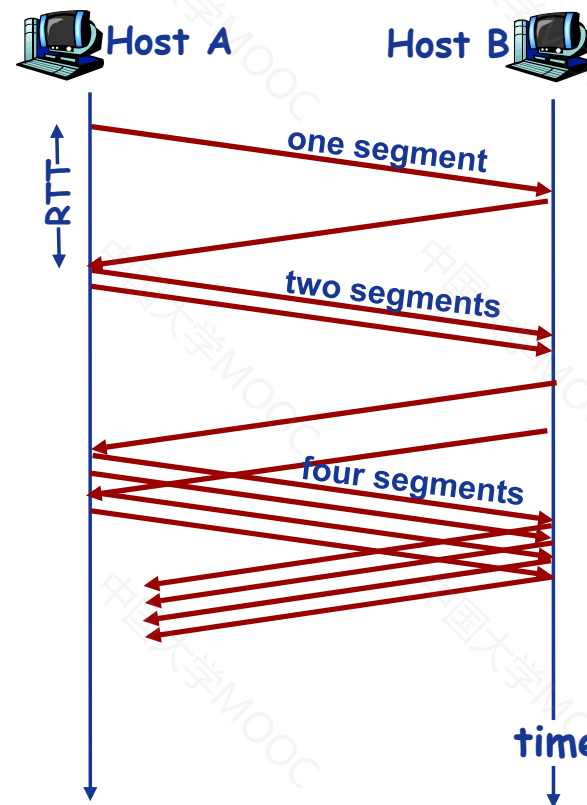


TCP慢启动: SS

❖ 指数性增长

- 每个RTT将CongWin翻倍
- 收到每个ACK进行操作

❖ 初始速率很慢, 但是快速攀升



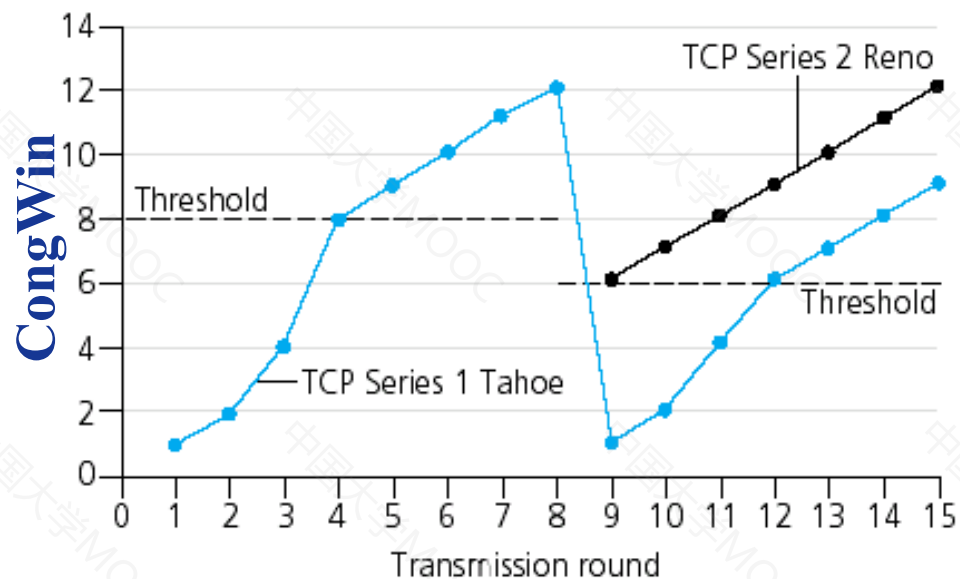
Threshold变量

Q: 何时应该指数性增长切换为线性增长(拥塞避免)?

A: 当CongWin达到Loss事件前值的1/2时.

实现方法:

- ❖ 变量 **Threshold**
- ❖ Loss事件发生时, **Threshold** 被设为Loss事件前**CongWin**值的1/2。



Loss事件的处理

❖ 3个重复ACKs:

- CongWin切到一半
- 然后线性增长

❖ Timeout事件:

- CongWin直接设为1个MSS
- 然后指数增长
- 达到threshold后, 再线性增长

Philosophy:

- ❑ 3个重复ACKs表示网络还能够传输一些 segments
- ❑ timeout事件表明拥塞更为严重



TCP拥塞控制：总结

- ❖ When CongWin is below Threshold, sender in **slow-start** phase, window grows exponentially.
- ❖ When CongWin is above Threshold, sender is in **congestion-avoidance** phase, window grows linearly.
- ❖ When a **triple duplicate ACK** occurs, Threshold set to $\text{CongWin}/2$ and CongWin set to Threshold.
- ❖ When **timeout** occurs, Threshold set to $\text{CongWin}/2$ and CongWin is set to 1 MSS.



TCP拥塞控制

State	Event	TCP Sender Action	Commentary
Slow Start (SS)	ACK receipt for previously unacked data	$\text{CongWin} = \text{CongWin} + \text{MSS}$, If ($\text{CongWin} > \text{Threshold}$) set state to "Congestion Avoidance"	Resulting in a doubling of CongWin every RTT
Congestion Avoidance (CA)	ACK receipt for previously unacked data	$\text{CongWin} = \text{CongWin} + \text{MSS} * (\text{MSS} / \text{CongWin})$	Additive increase, resulting in increase of CongWin by 1 MSS every RTT
SS or CA	Loss event detected by triple duplicate ACK	$\text{Threshold} = \text{CongWin} / 2$, $\text{CongWin} = \text{Threshold}$, Set state to "Congestion Avoidance"	Fast recovery, implementing multiplicative decrease. CongWin will not drop below 1 MSS.
SS or CA	Timeout	$\text{Threshold} = \text{CongWin} / 2$, $\text{CongWin} = 1 \text{ MSS}$, Set state to "Slow Start"	Enter slow start
SS or CA	Duplicate ACK	Increment duplicate ACK count for segment being acked	CongWin and Threshold not changed



TCP拥塞控制算法

Th = ?

CongWin = 1 MSS

/* slow start or exponential increase */

While (No Packet Loss and CongWin < Th) {
 send CongWin TCP segments
 for each ACK increase CongWin by 1
}

/* congestion avoidance or linear increase */

While (No Packet Loss) {
 send CongWin TCP segments
 for CongWin ACKs, increase CongWin by 1
}

Th = CongWin/2

If (3 Dup ACKs) CongWin = Th;

If (timeout) CongWin=1;



例题

❖ 一个TCP连接总是以1 KB的最大段长发送TCP段，发送方有足够多的数据要发送。当拥塞窗口为16 KB时发生了超时，如果接下来的4个RTT（往返时间）时间内的TCP段的传输都是成功的，那么当第4个RTT时间内发送的所有TCP段都得到肯定应答时，拥塞窗口大小是多少？

❖ 解：threshold=16/2=8 KB, CongWin=1 KB, 1个RTT后, CongWin=2 KB, 2个RTT后, CongWin=4 KB, 3个RTT后, CongWin=8 KB, Slowstart is over; 4个RTT后, CongWin=9 KB





哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

立足航天，服务国防，面向国民经济主战场

谢谢!