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可靠数据传输

TCP可靠数据传输概述

- TCP在IP层提供的不可靠服务基础上实现可靠数据传输服务
- 流水线机制
- 累积确认
- TCP使用单一重传定时器
- 触发重传的事件
  - 超时
  - 收到重复ACK
- 渐进式
  - 暂不考虑重复ACK
  - 暂不考虑流量控制
  - 暂不考虑拥塞控制

TCP RTT和超时

- 问题：如何设置定时器的超时时间？
- 大于RTT
  - 但是RTT是变化的
- 过短：
  - 不必要的重传
- 过长：
  - 对段丢失时间反应慢

- 问题：如何估计RTT？
- SampleRTT: 测量从段发出去到收到ACK的时间
  - 忽略重传
- SampleRTT变化
  - 测量多个SampleRTT，求平均值，形成RTT的估计值EstimatedRTT

$$EstimatedRTT = (1-\alpha)*EstimatedRTT + \alpha*SampleRTT$$

指数加权移动平均

典型值：0.125

TCP RTT和超时

定时器超时时间的设置:

- EstimatedRTT + “安全边界”
- EstimatedRTT变化大➔较大的边界

测量RTT的变化值: SampleRTT与EstimatedRTT的差值

$$DevRTT = (1-\beta)*DevRTT + \beta*|SampleRTT-EstimatedRTT|$$

(typically, $\beta = 0.25$)

定时器超时时间的设置：

$$TimeoutInterval = EstimatedRTT + 4*DevRTT$$

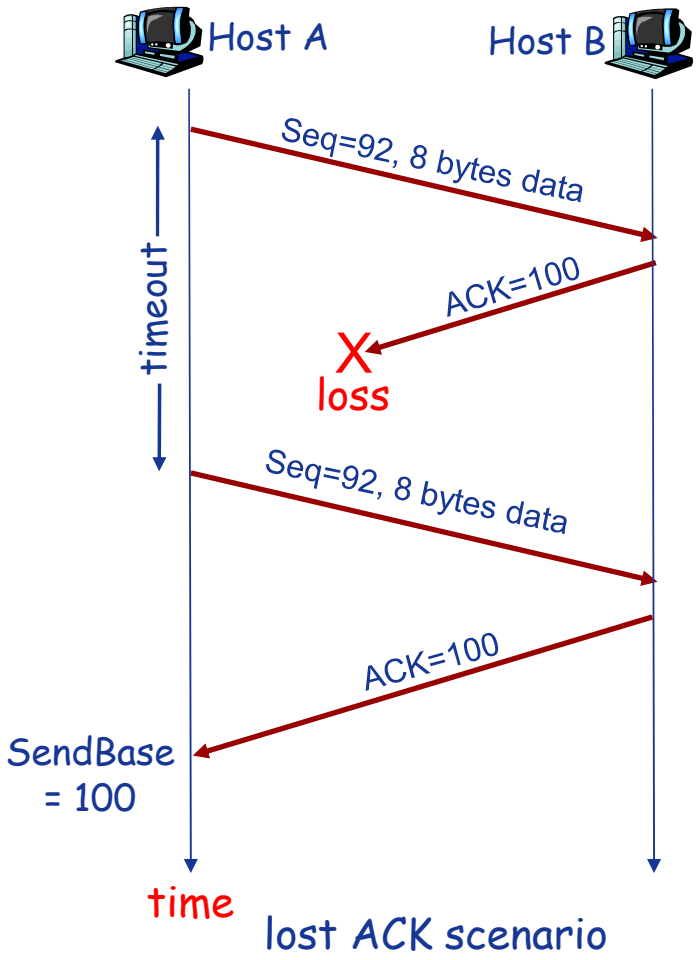
TCP发送方事件

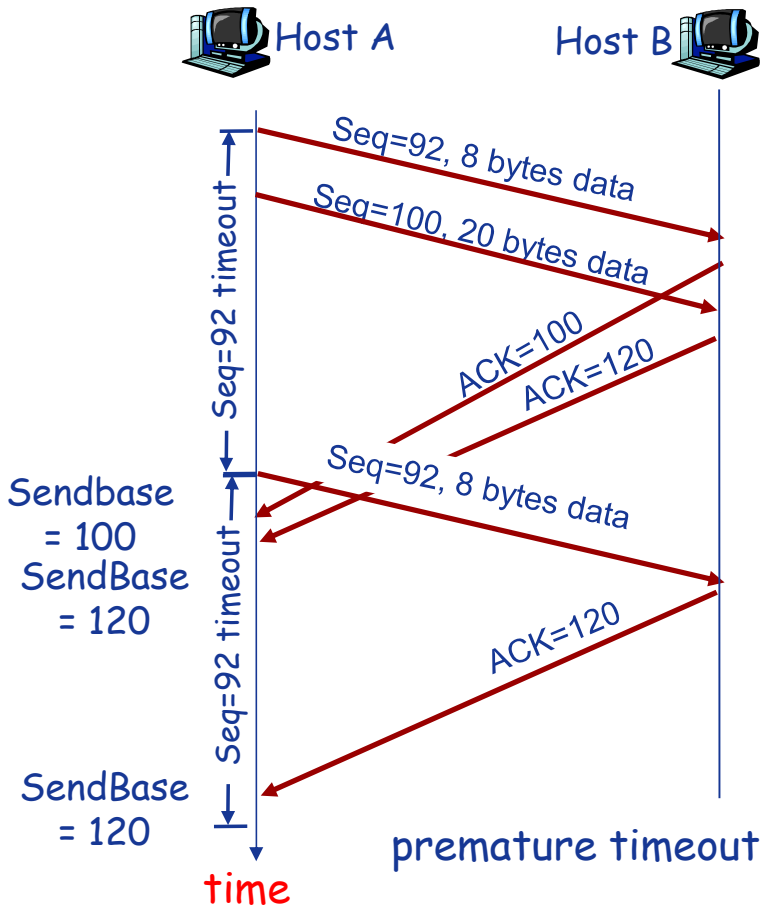
- 从应用层收到数据
  - 创建Segment
  - 序列号是Segment第一个字节的编号
  - 开启计时器
  - 设置超时时间：TimeOutInterval
- 超时
  - 重传引起超时的Segment
  - 重启定时器
- 收到ACK
  - 如果确认此前未确认的Segment
    - 更新SendBase
    - 如果窗口中还有未被确认的分组，重新启动定时器

TCP发送端程序

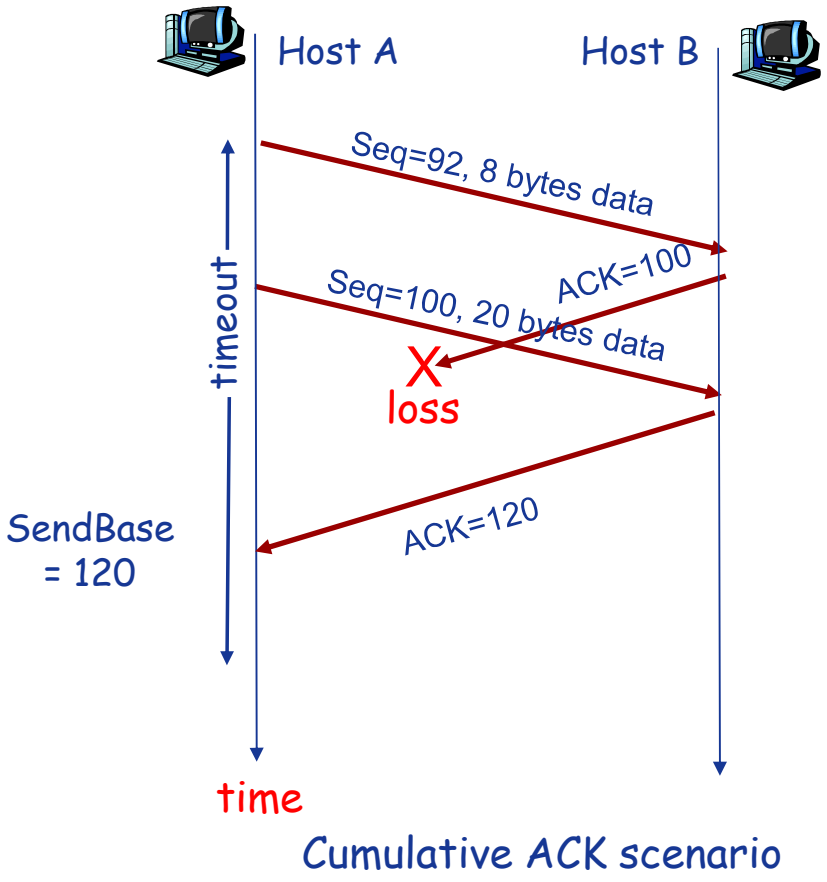
```
NextSeqNum = InitialSeqNum
SendBase = InitialSeqNum

loop (forever) {
  switch(event)

  event: data received from application above
      create TCP segment with sequence number NextSeqNum
      if (timer currently not running)
          start timer
      pass segment to IP
      NextSeqNum = NextSeqNum + length(data)

  event: timer timeout
      retransmit not-yet-acknowledged segment with
          smallest sequence number
      start timer

  event: ACK received, with ACK field value of y
      if (y > SendBase) {
          SendBase = y
          if (there are currently not-yet-acknowledged segments)
              start timer
          }

} /* end of loop forever */
```

TCP重传示例


Host A
Host B
timeout
Seq=92, 8 bytes data
ACK=100
X
loss
Seq=92, 8 bytes data
ACK=100
SendBase
= 100
time
lost ACK scenario


Host A
Host B
Seq=92 timeout
Seq=92, 8 bytes data
Seq=100, 20 bytes data
ACK=100
ACK=120
Seq=92, 8 bytes data
Sendbase
= 100
SendBase
= 120
ACK=120
Seq=92 timeout
SendBase
= 120
time
premature timeout

TCP重传示例



Cumulative ACK scenario

TCP ACK生成: RFC 1122, RFC 2581

Event at Receiver	TCP Receiver action
Arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed	Delayed ACK. Wait up to 500ms for next segment. If no next segment, send ACK
Arrival of in-order segment with expected seq #. One other segment has ACK pending	Immediately send single cumulative ACK, ACKing both in-order segments
Arrival of out-of-order segment higher-than-expect seq. # . Gap detected	Immediately send *duplicate ACK*, indicating seq. # of next expected byte
Arrival of segment that partially or completely fills gap	Immediate send ACK, provided that segment startsat lower end of gap

快速重传机制

- TCP的实现中，如果发生超时，超时时间间隔将重新设置，即将超时时间间隔加倍，导致其很大
  - 重发丢失的分组之前要等待很长时间
- 通过重复ACK检测分组丢失
  - Sender会背靠背地发送多个分组
  - 如果某个分组丢失，可能会引发多个重复的ACK
- 如果sender收到对同一数据的3个ACK，则假定该数据之后的段已经丢失
  - **快速重传**：在定时器超时之前即进行重传

快速重传算法

```
event: ACK received, with ACK field value of y
            if (y > SendBase) {
                SendBase = y
                if (there are currently not-yet-acknowledged segments)
                    start timer
              }
            else {
                increment count of dup ACKs received for y
                if (count of dup ACKs received for y = 3) {
                    resend segment with sequence number y
                 }
```

a duplicate ACK for already ACKed segment

fast retransmit

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！