

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



TCP之可靠数据传输

TCP之可靠数据传输

硕士研究生 程浩卿

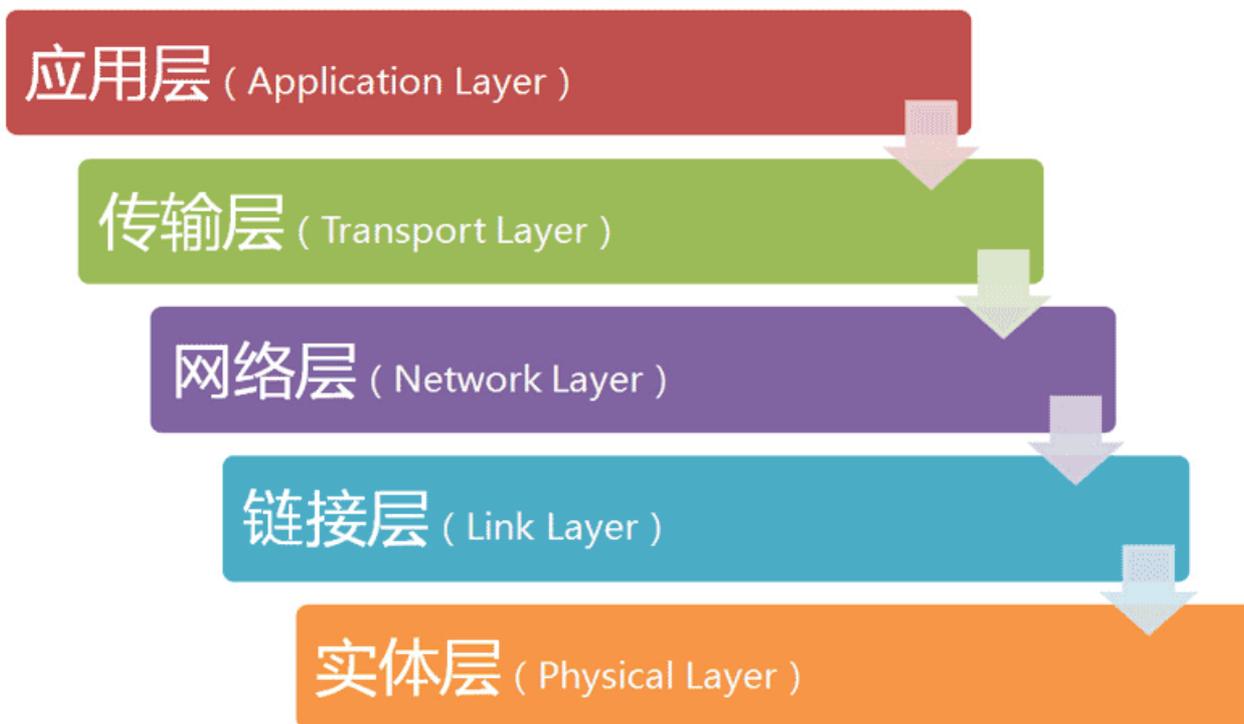
2018年4月8日

- 基本概念
- 可靠数据传输原理
- TCP中的原理实现

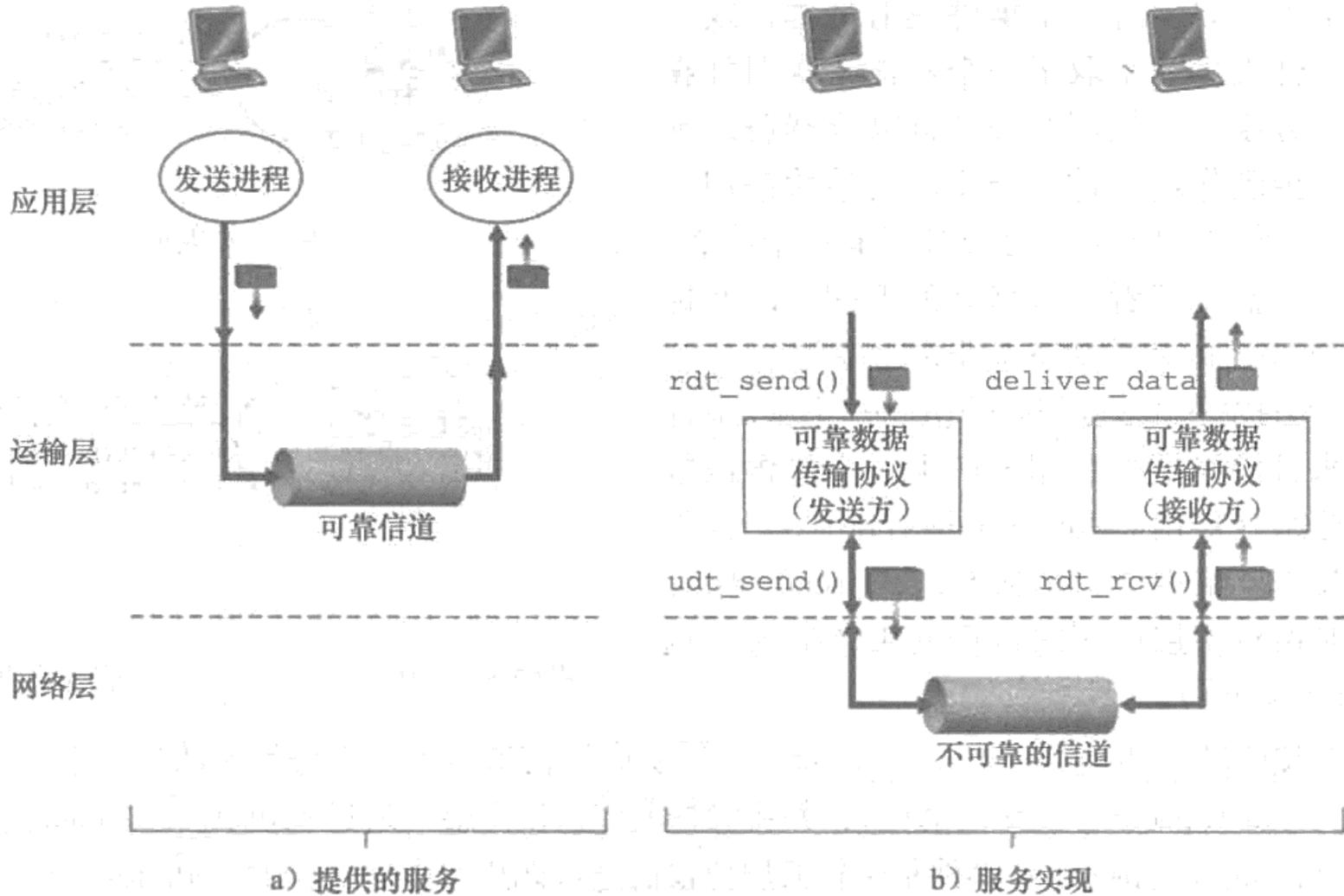


基本概念

- 五层因特网协议栈



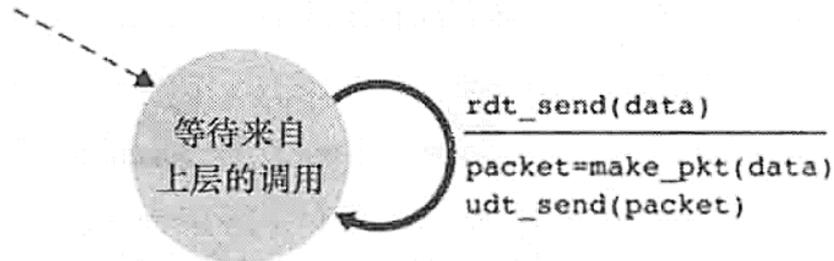
- 套接字 (socket)
 - 应用层与传输层的**接口**，TCP套接字由四元组标识，即 {源端口号，源IP，目的端口号，目的IP}
- 多路分解与多路复用
 - 多路分解即在接收端将传输层报文**定向至不同的套接字**
 - 多路复用即在发送端**从不同的套接字收取**应用层报文



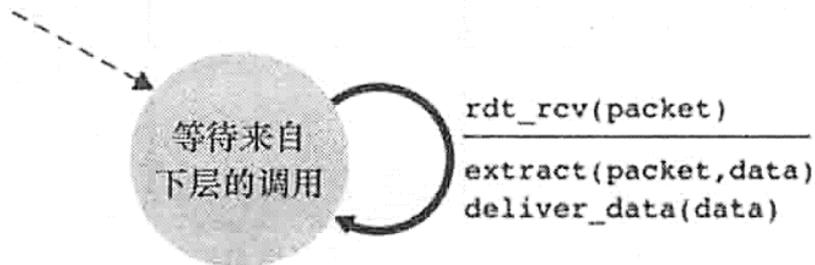


可靠数据传输原理

- 最理想的情况，底层信道（传输层以下）是完全可靠的
 - 发送端的传输层只有一个状态，即等待应用层的调用
 - 接收端的传输层只有一个状态，即等待网络层的调用

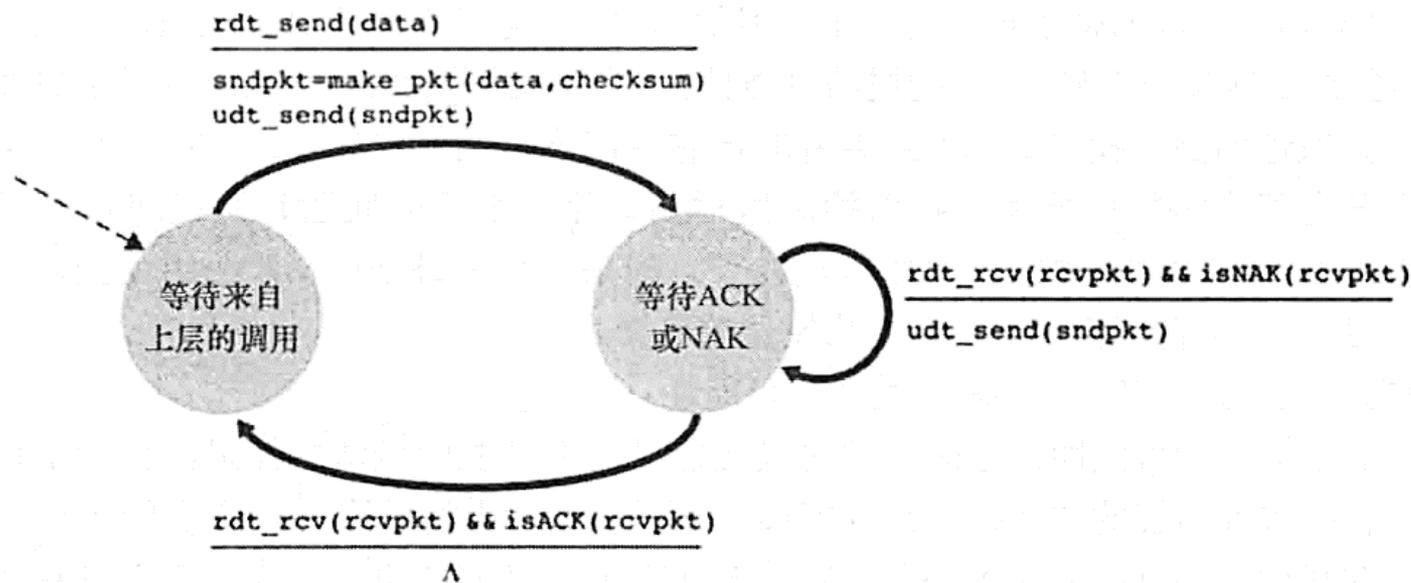


a) rdt1.0: 发送端

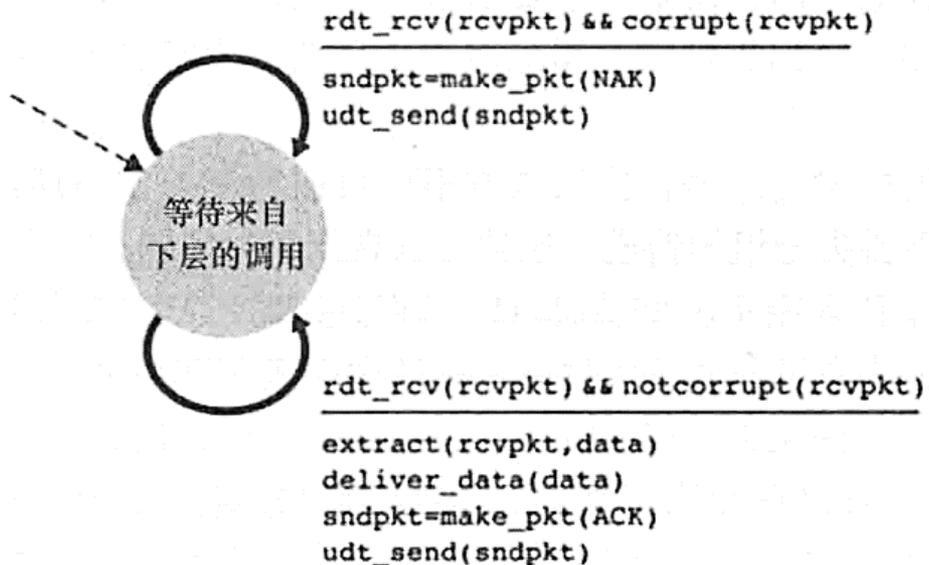


b) rdt1.0: 接收端

- 考虑分组受损的情况（不包括分组丢失）
 - 接收端需要对报文进行**差错检验**并给予**反馈**
 - 发送端根据反馈选择是否进行**重传**，等待反馈期间不会发送新的数据，即为**停等协议**



a) rdt2.0: 发送端



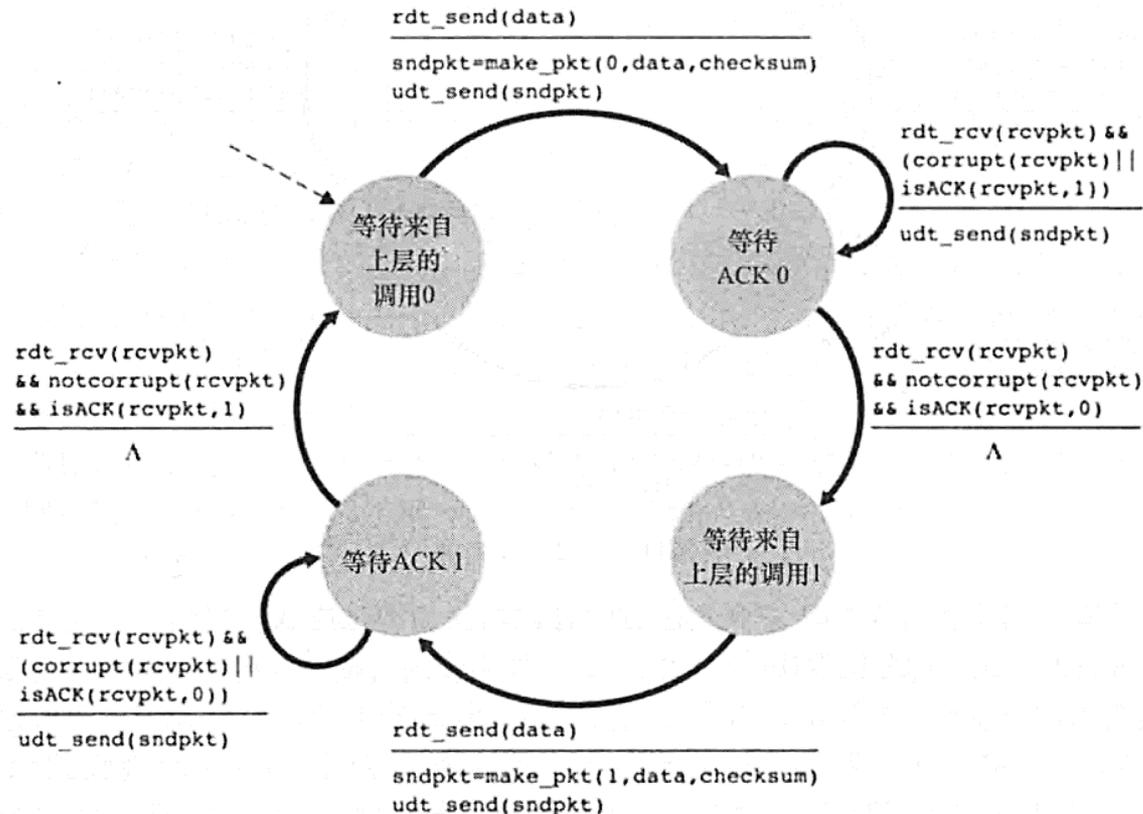
b) rdt2.0: 接收端

- RDT2.0存在的问题

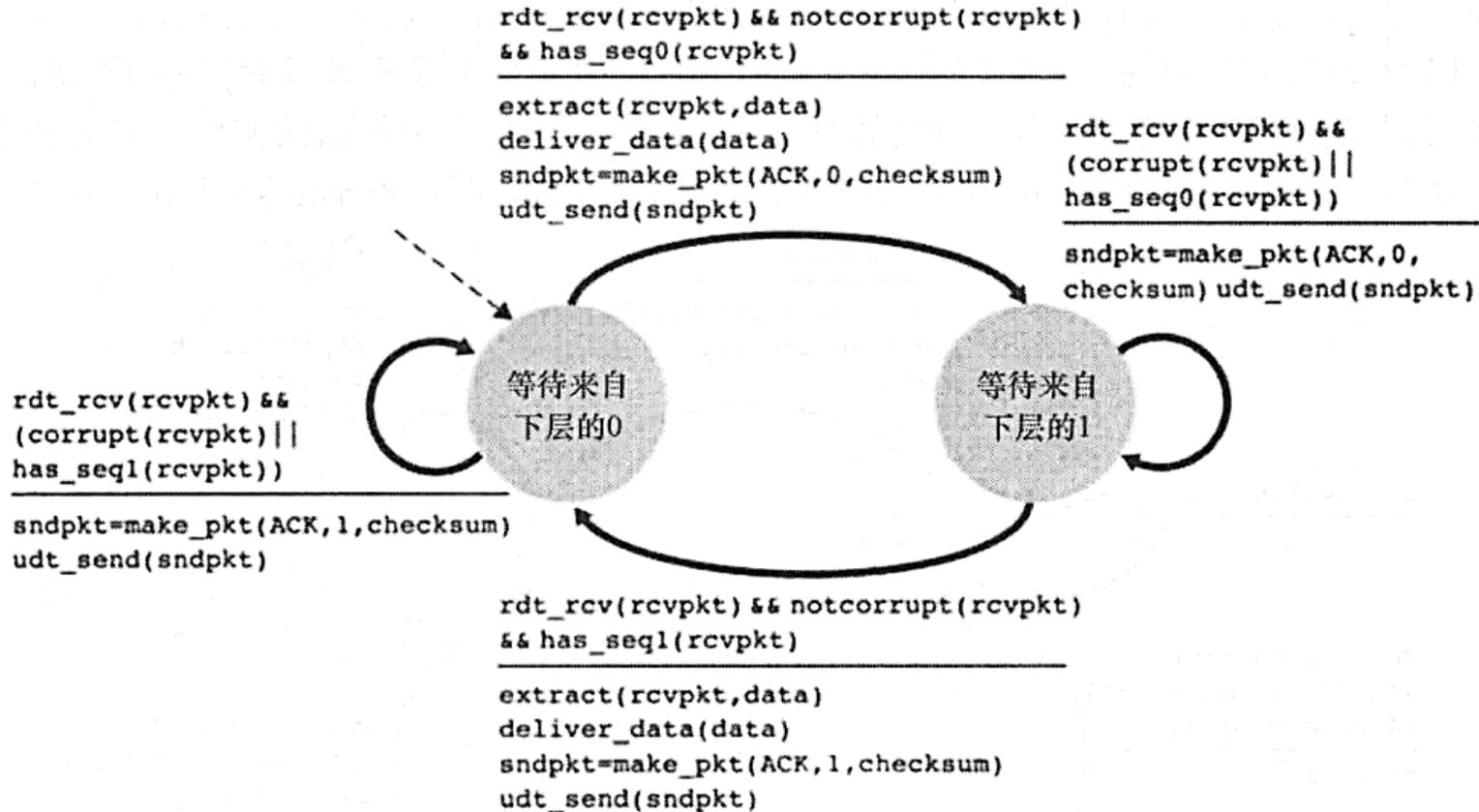
- 反馈报文也有可能受损

- 解决办法，当发送端收到模糊的反馈应答时，就直接重传

- 重传策略将会带来**冗余分组**，致使接收端无法分辨
 - 发送端在分组中**添加序号字段**，使接收端得以分辨当前到来的分组**是新的分组还是重传分组**



RDT2.1



- 将分组丢失纳入考虑
 - 在发送端发送分组的时候**启动一个定时器**，如果响应超时，则重传分组，重启定时器
- 可靠数据传输协议
 - 需要检验和、序号、定时器、反馈机制这些技术协同工作



TCP中的原理实现

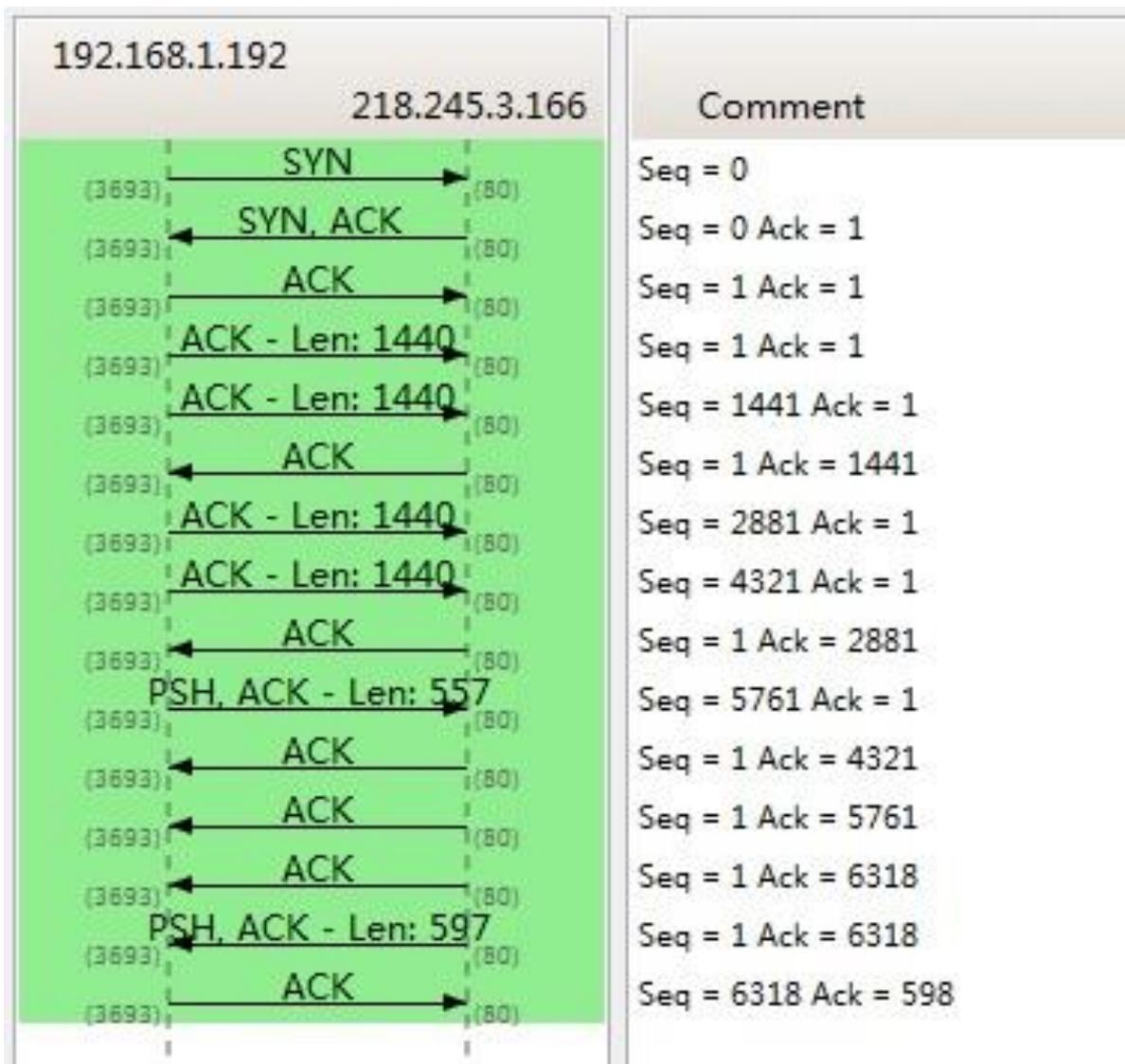
- TCP报文段结构



- SequenceNum (序列号)
 - TCP连接中, 为传送的字节流 (数据) 中的每一个字节按顺序编号
 - 如果一个TCP报文段的序号为301, 它携带了100字节的数据, 就表示这100个字节的数据的字节序号范围是[301, 400], 该报文段携带的第一个字节序号是301, 最后一个字节序号是400

- AckNum (确认号)
 - TCP协议中，一般采用**累积确认**的方式，即每传送多个连续 TCP段，可以只对最后一个TCP段进行确认
 - 发送方发送了一个**报文段序号为301**的TCP段，这个段携带了 **100字节数据**，则接收方应当回复的**确认号是401**，它表示接收方已经收到了字节序号为 [0, 400] 的数据，现在期望你发送字节序号为401以及以后的数据

TCP中的原理实现



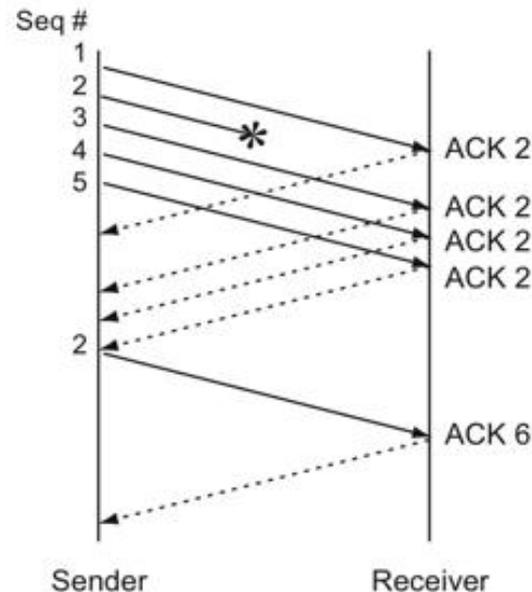


- TCP重传机制
 - SeqNum和Ack是以字节数为单位，所以Ack的时候，不能跳着确认，只能确认最大的连续收到的包
- 发送端发了1,2,3,4,5一共五份数据，接收端收到了1, 2，于是回Ack 3，然后收到了4（注意此时3没收到），此时的TCP会怎么办？

- 超时重传
 - 一种是不回Ack，死等3，当发送方发现收不到3的Ack超时后，会重传3。一旦接收方收到3后，Ack回4——意味着3和4都收到了。

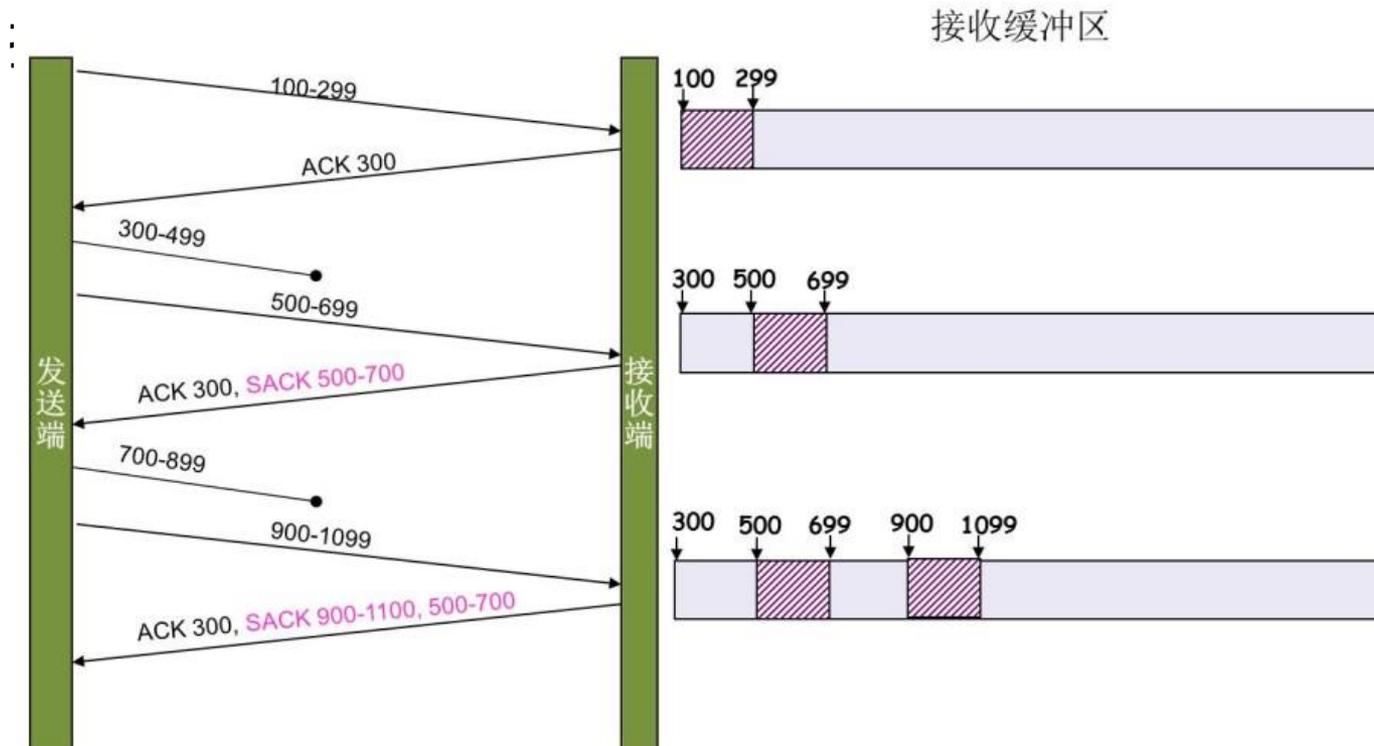
- 快速重传

- 不以时间驱动，而以数据驱动重传。也就是说，如果，包没有连续到达，就Ack最后那个可能被丢了的包，如果发送方连续收到3次相同的Ack，就重传。Fast Retransmit的好处是不用等timeout了再重传。



- SACK方法

- Selective Acknowledgment (参看RFC 2018)，这种方式需要在TCP头里加一个SACK的东西，ACK还是Fast Retransmit的ACK，SACK则是汇报收到的数



谢谢!

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。

