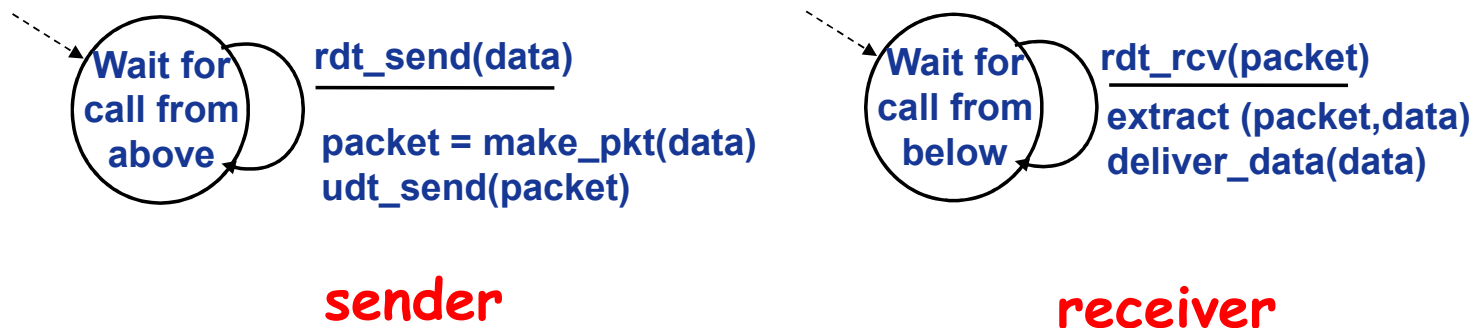


Rdt 1.0: 可靠信道上的可靠数据传输

❖ 底层信道完全可靠

- 不会发生错误(bit error)
- 不会丢弃分组

❖ 发送方和接收方的FSM独立



本讲主题

Rdt 2.0



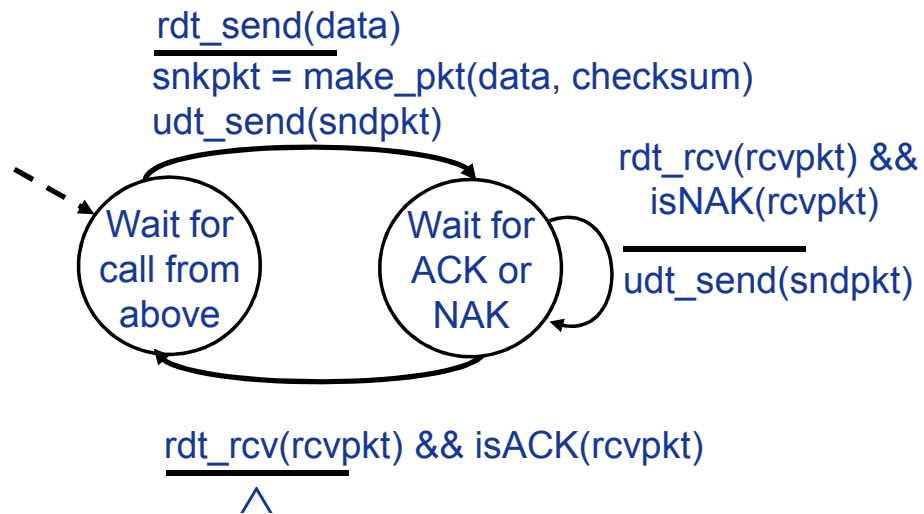
Rdt 2.0: 产生位错误的信道



- ❖ 底层信道可能翻转分组中的位(bit)
 - 利用校验和检测位错误
- ❖ 如何从错误中恢复？
 - 确认机制(Acknowledgements, ACK): 接收方显式地告知发送方分组已正确接收
 - NAK:接收方显式地告知发送方分组有错误
 - 发送方收到NAK后, 重传分组
- ❖ 基于这种重传机制的rdt协议称为ARQ(Automatic Repeat reQuest)协议
- ❖ Rdt 2.0中引入的新机制
 - 差错检测
 - 接收方反馈控制消息: ACK/NAK
 - 重传



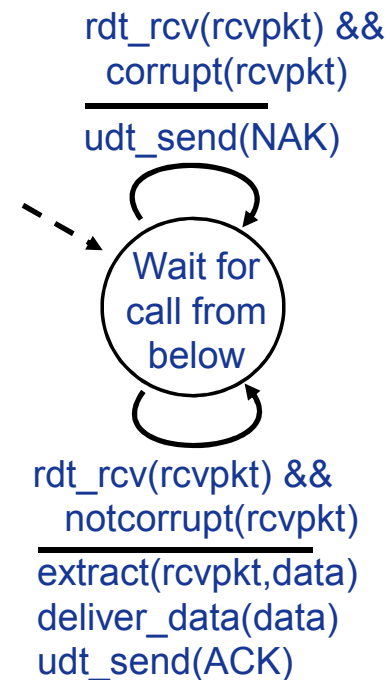
Rdt 2.0: FSM规约



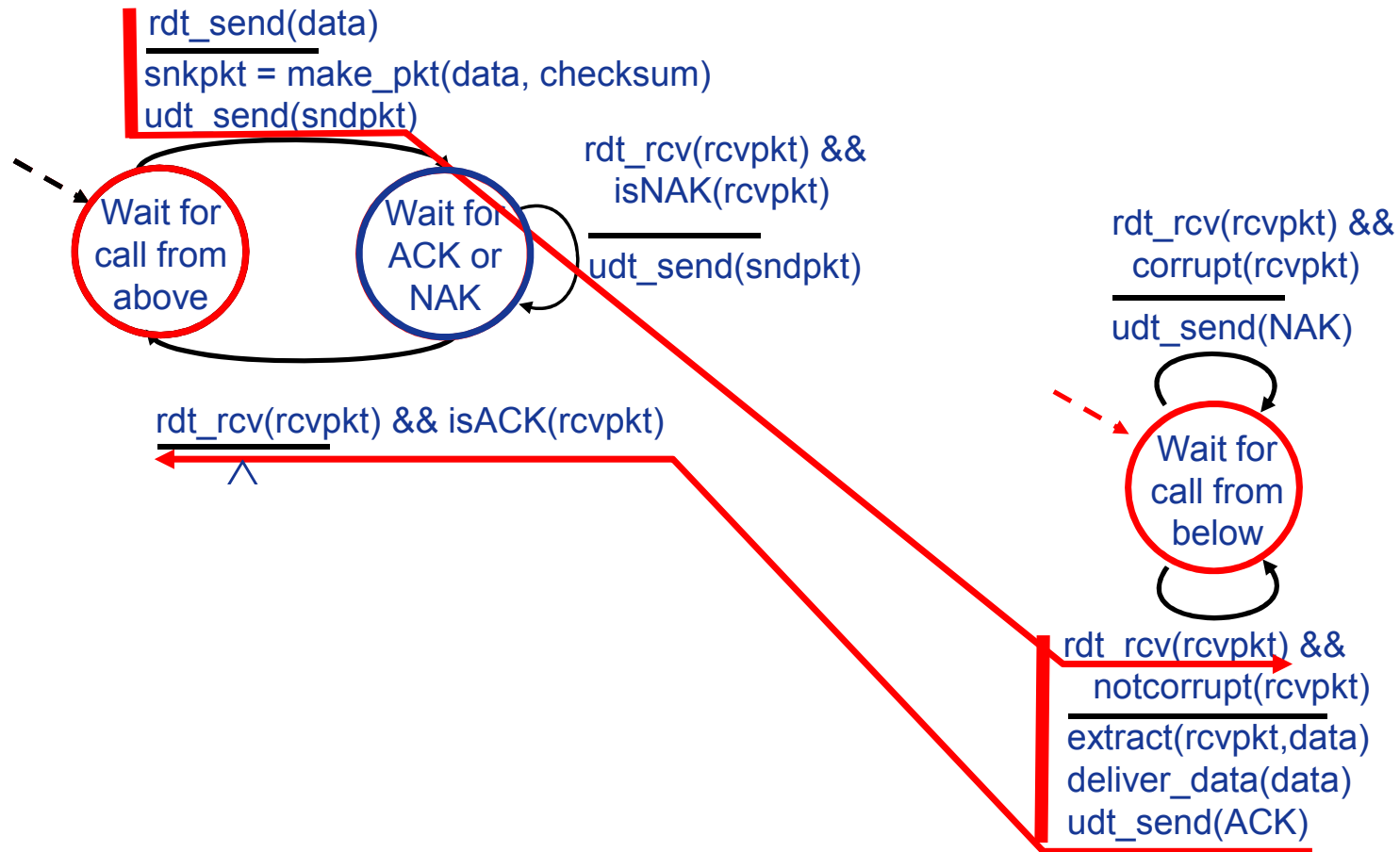
sender

停一等协议

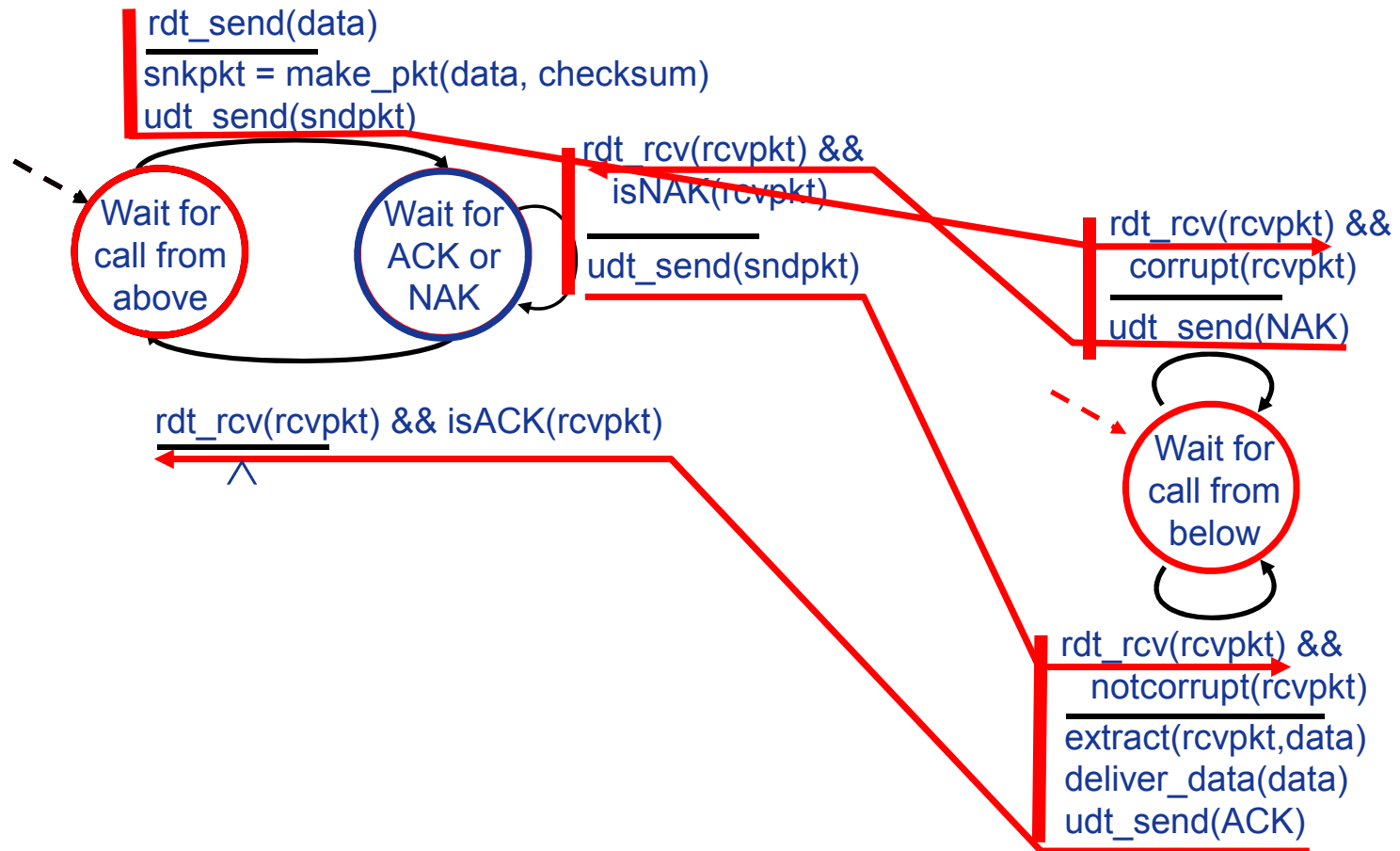
receiver



Rdt 2.0: 无错误场景



Rdt 2.0: 有错误场景



本讲主题

Rdt 2.1和2.2



Rdt 2.0有什么缺陷？

❖ 如果ACK/NAK消息发生错误/被破坏(corrupted)会怎么样？

- 为ACK/NAK增加校验和，检错并纠错
- 发送方收到被破坏ACK/NAK时不知道接收方发生了什么，添加额外的控制消息
- 如果ACK/NAK坏掉，发送方重传
- 不能简单的重传：产生重复分组



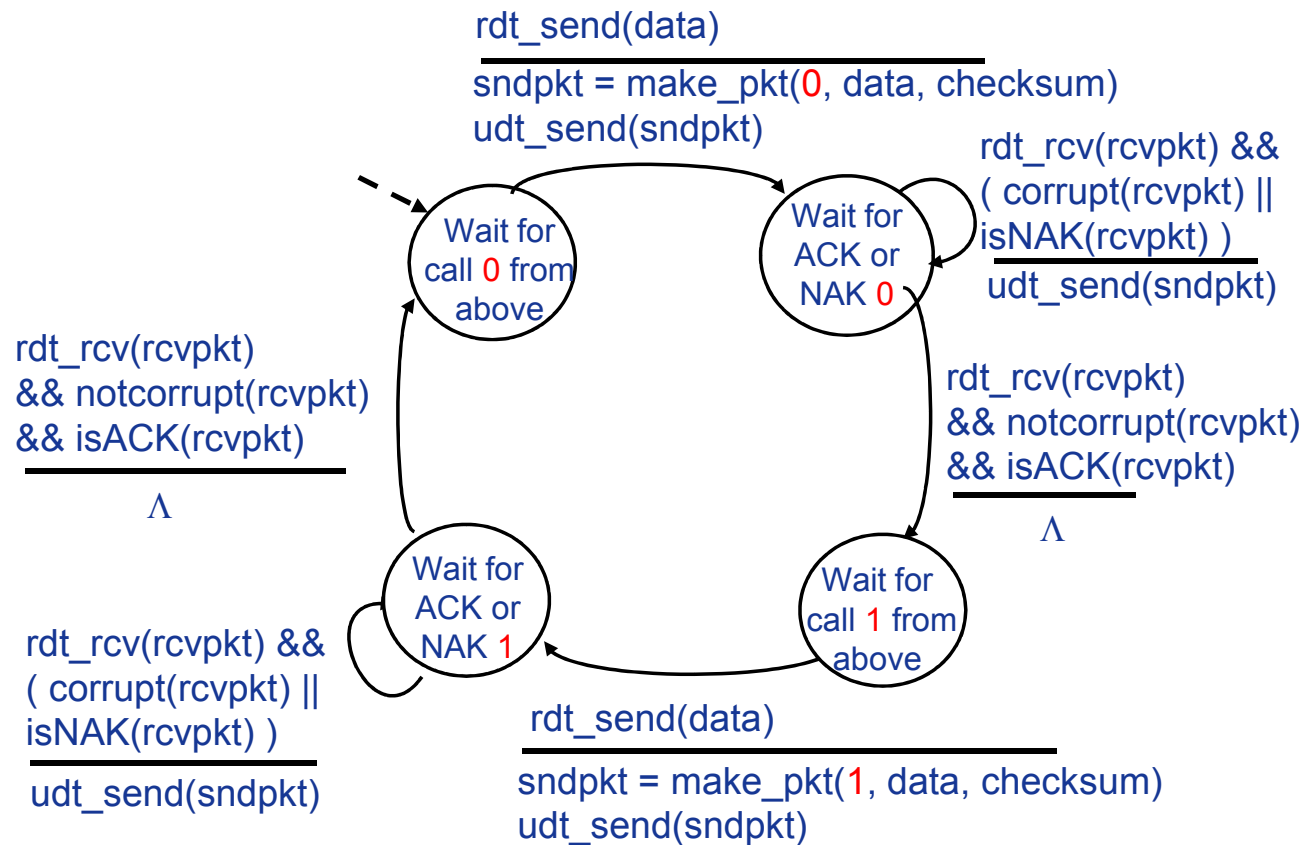
❖ 如何解决重复分组问题？

- 序列号(Sequence number): 发送方给每个分组增加序列号
- 接收方丢弃重复分组

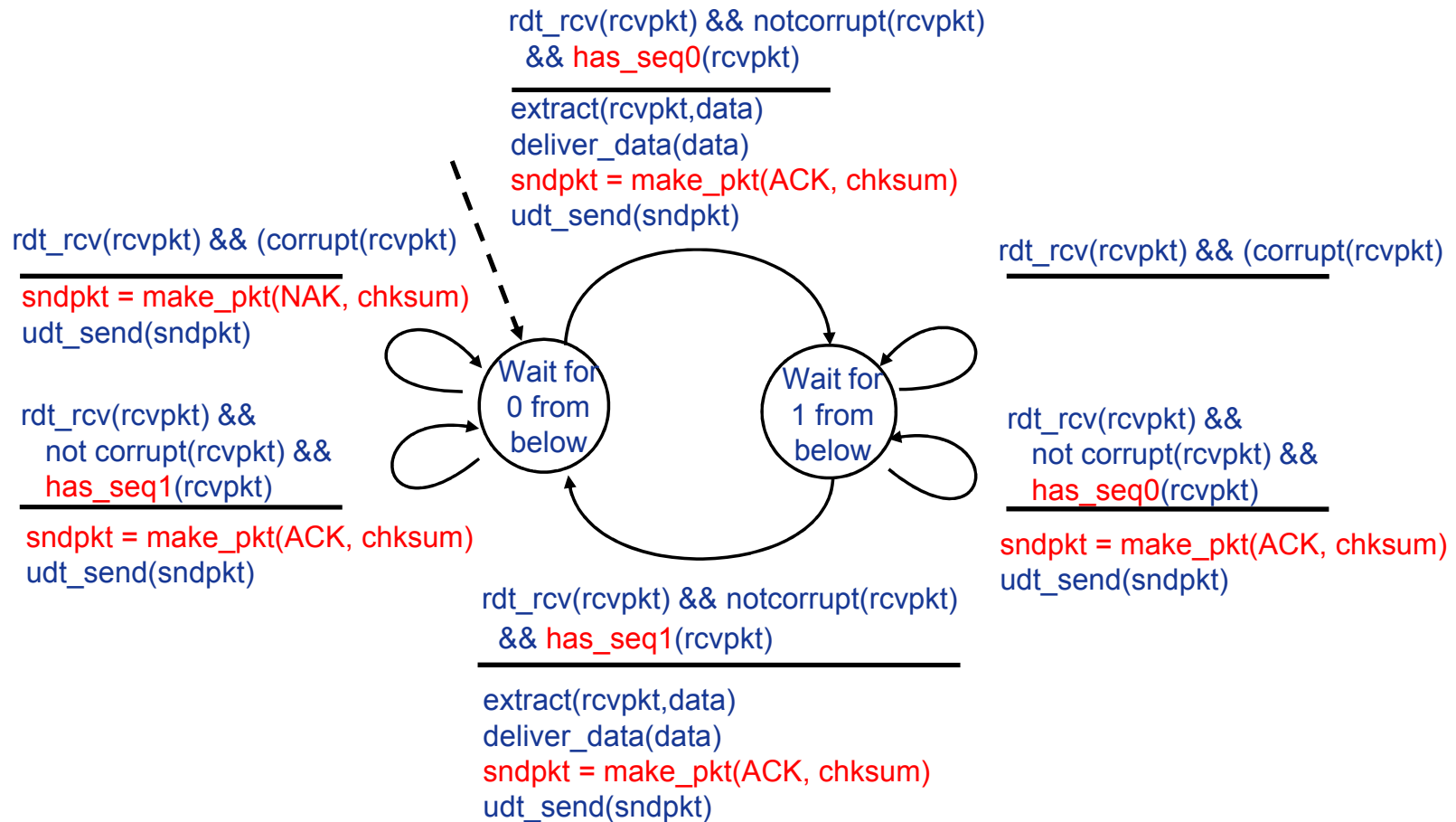
stop and wait
Sender sends one packet,
then waits for receiver
response



Rdt 2.1: 发送方, 应对ACK/NAK破坏



Rdt 2.1: 接收方, 应对ACK/NAK破坏



Rdt 2.1 vs. Rdt 2.0

❖ 发送方:

- 为每个分组增加了序列号
- 两个序列号(0, 1)就够用, 为什么?
- 需校验ACK/NAK消息是否发生错误
- 状态数量翻倍
 - 状态必须“记住”“当前”的分组序列号

❖ 接收方

- 需判断分组是否是重复
 - 当前所处状态提供了期望收到分组的序列号
- 注意: 接收方无法知道ACK/NAK是否被发送方正确收到

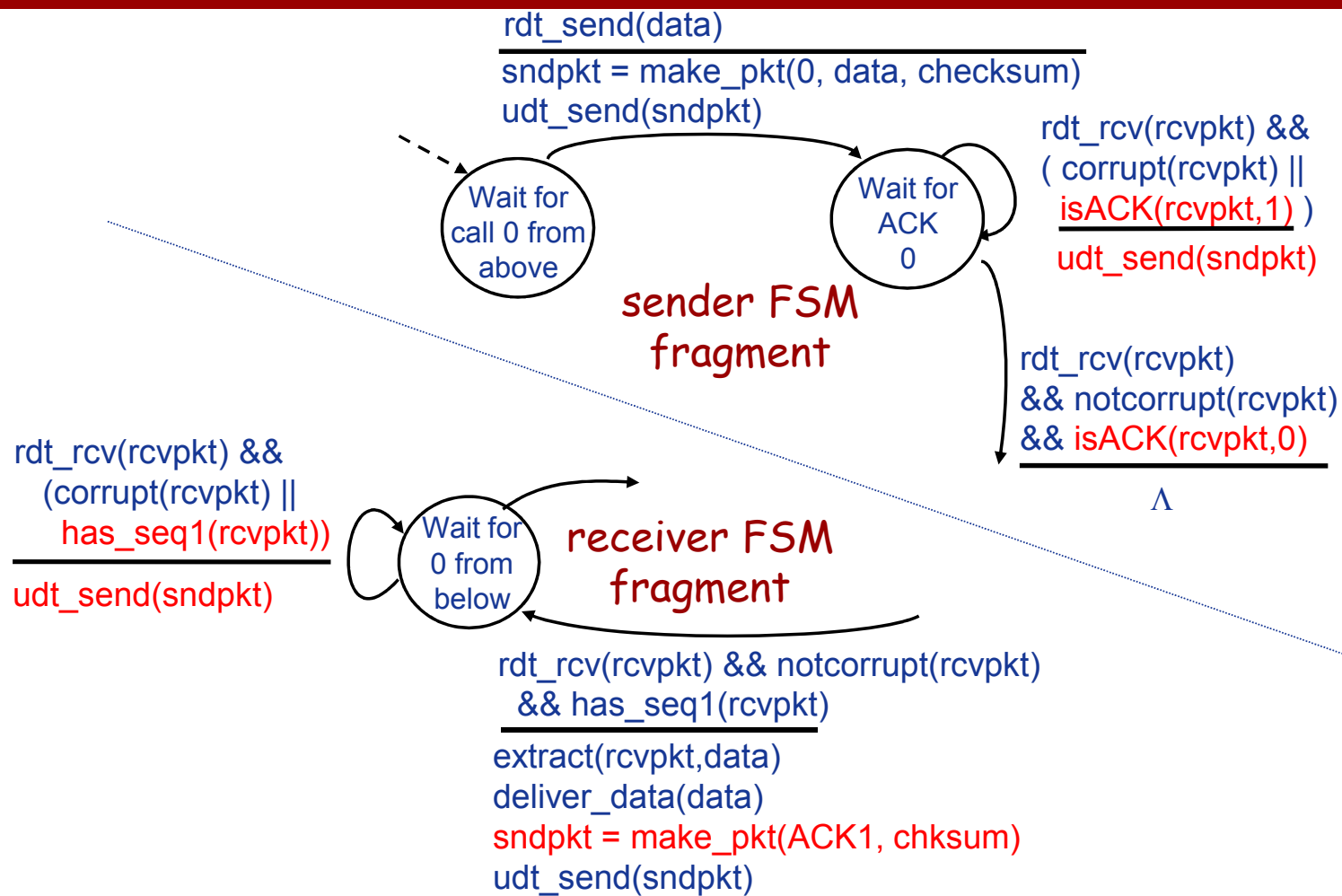


Rdt 2.2: 无NAK消息协议

- ❖ 我们真的需要两种确认消息(ACK + NAK)吗?
- ❖ 与rdt 2.1功能相同，但是只使用ACK
- ❖ 如何实现?
 - 接收方通过ACK告知最后一个被正确接收的分组
 - 在ACK消息中显式地加入被确认分组的序列号
- ❖ 发送方收到重复ACK之后，采取与收到NAK消息相同的动作
 - 重传当前分组



Rdt 2.2 FSM片段



本讲主题

Rdt 3.0



Rdt 3.0

❖ 如果信道既可能发生错误，也可能丢失分组，怎么办？

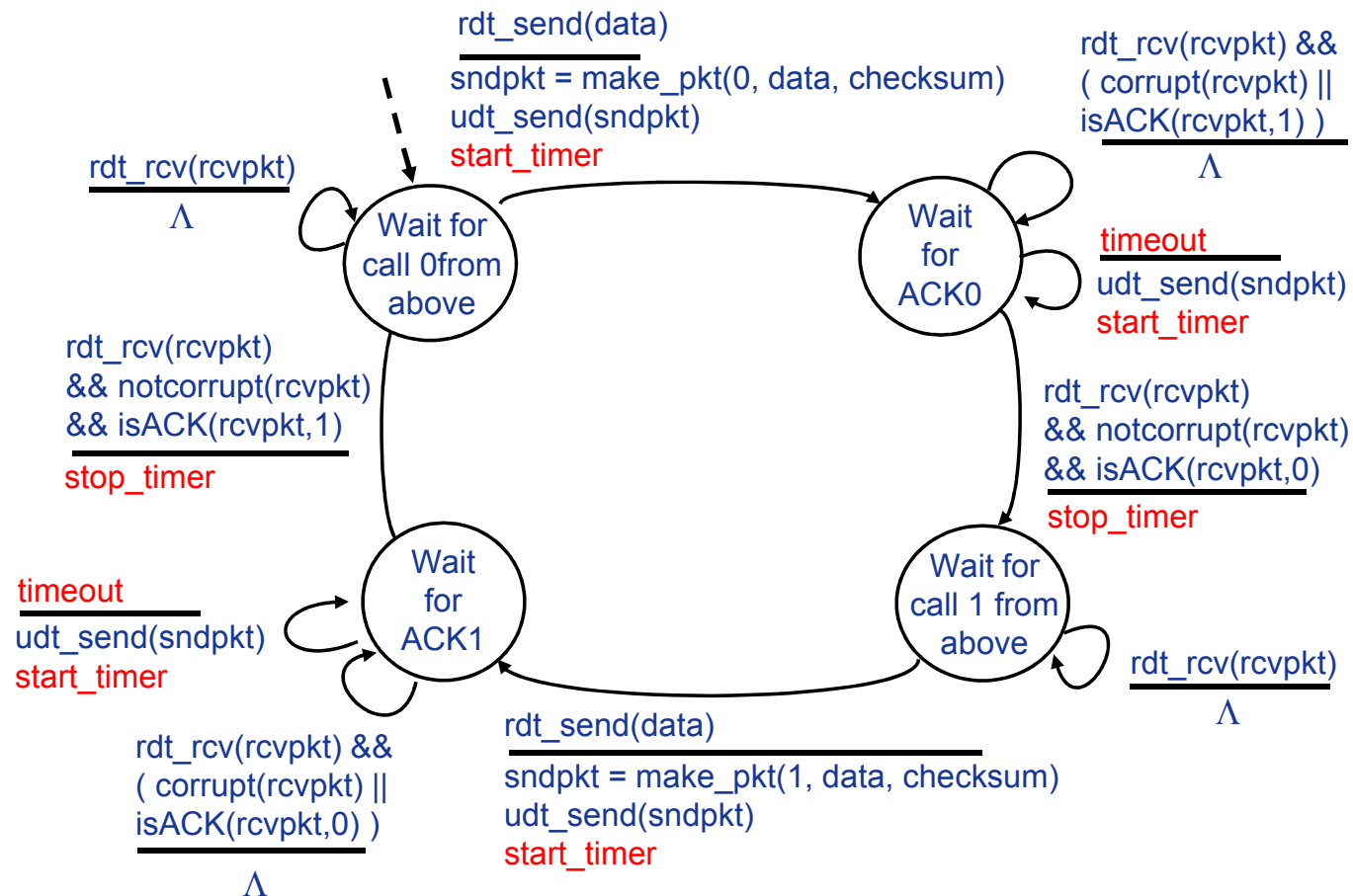
- “校验和 + 序列号 + ACK + 重传” 够用吗？

❖ 方法：发送方等待“合理”时间

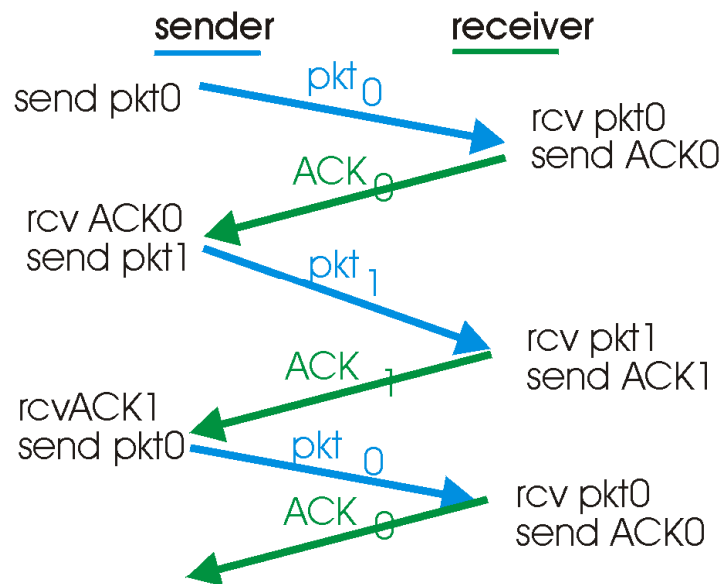
- 如果没收到ACK，重传
- 如果分组或ACK只是延迟而不是丢了
 - ⑩ 重传会产生重复，序列号机制能够处理
 - ⑩ 接收方需在ACK中显式告知所确认的分组
- 需要定时器



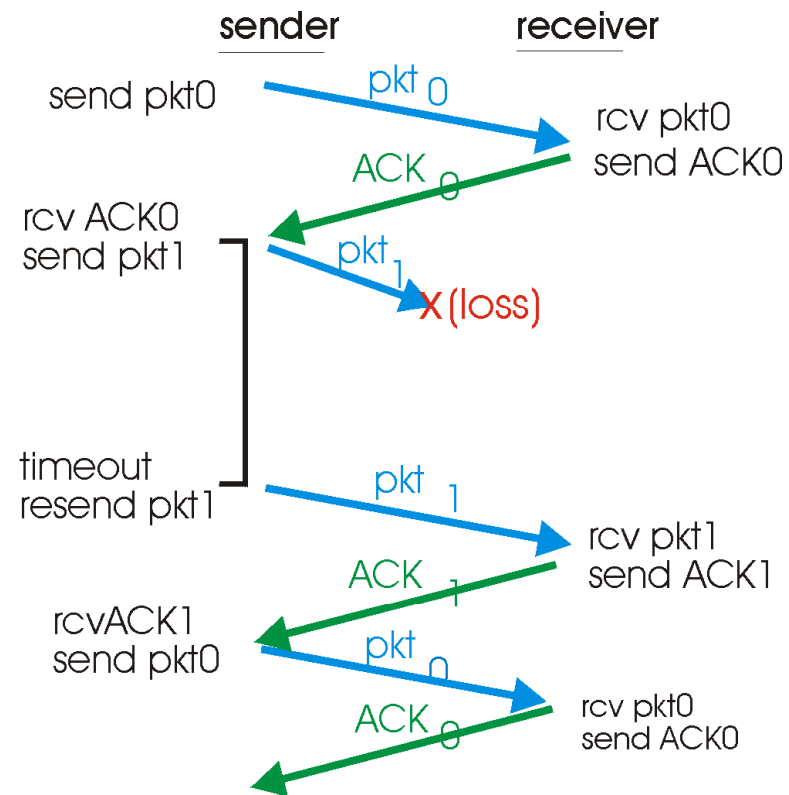
Rdt 3.0发送方FSM



Rdt 3.0示例(1)



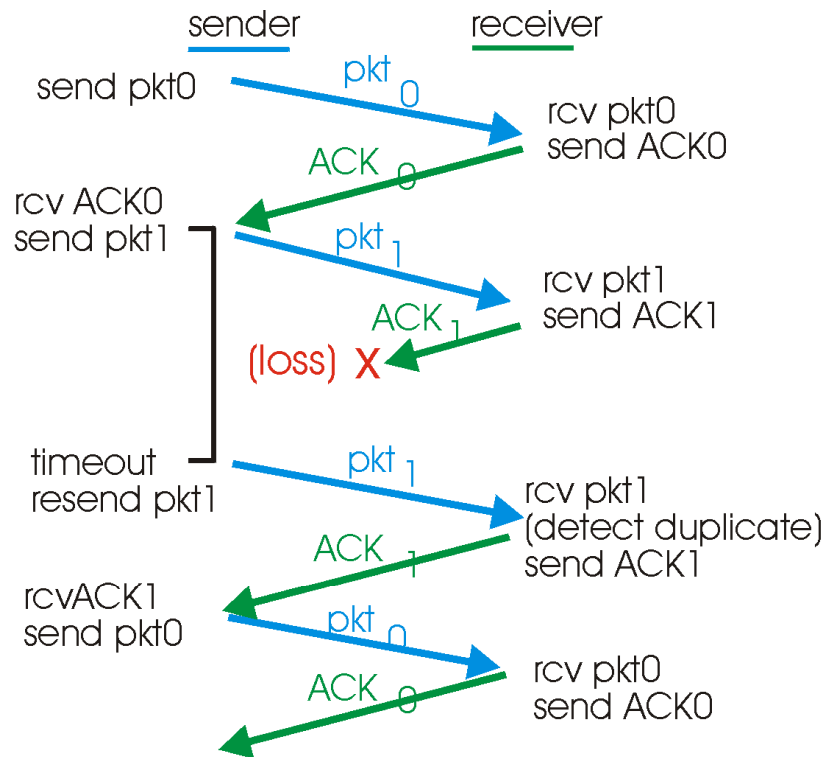
(a) operation with no loss



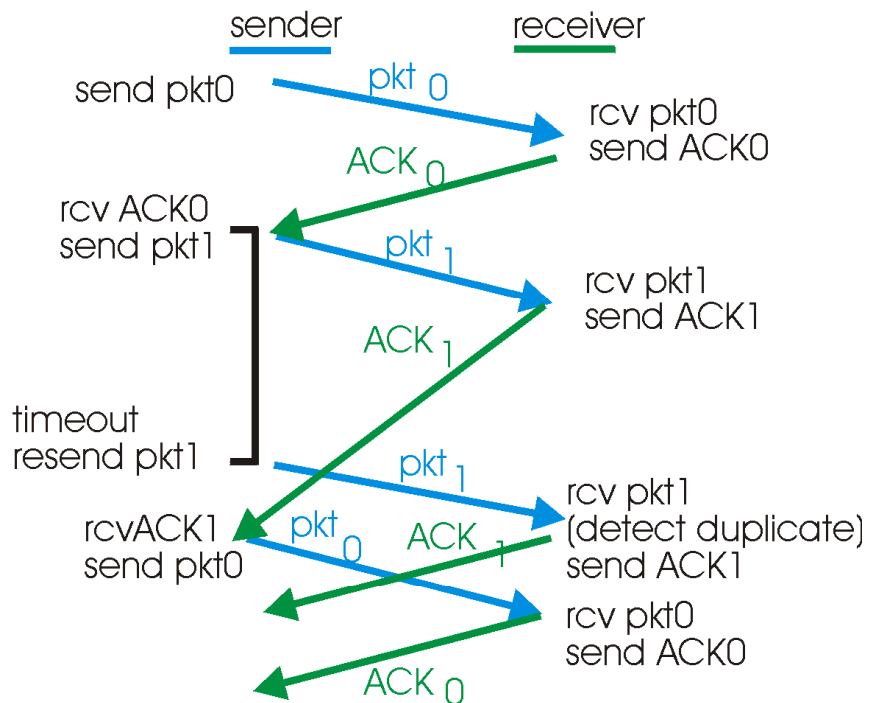
(b) lost packet



Rdt 3.0示例(2)



(c) lost ACK



(d) premature timeout



Rdt 3.0性能分析

- ❖ Rdt 3.0能够正确工作，但性能很差
- ❖ 示例：1Gbps链路，15ms端到端传播延迟，1KB分组

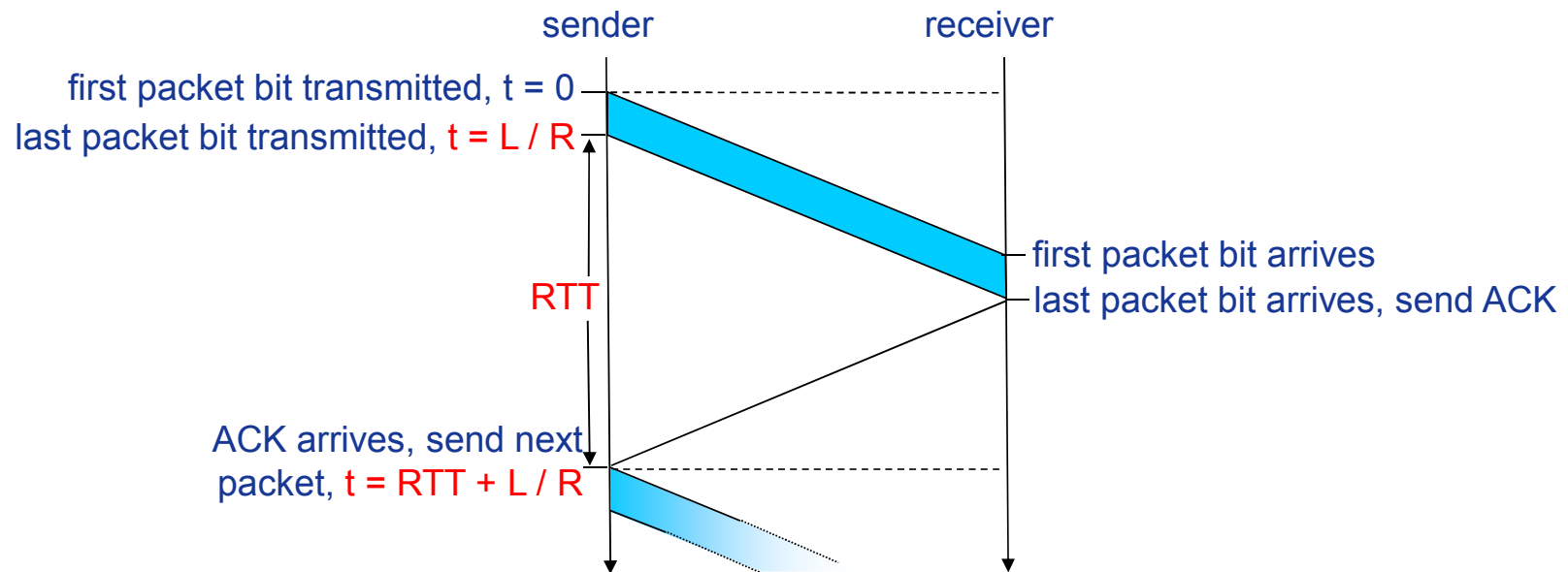
$$T_{\text{transmit}} = \frac{L \text{ (packet length in bits)}}{R \text{ (transmission rate, bps)}} = \frac{8\text{kb/pkt}}{10^9 \text{ b/sec}} = 8 \text{ microsec}$$

- 发送方利用率：发送方发送时间百分比

$$U_{\text{sender}} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

- 在1Gbps链路上每30毫秒才发送一个分组→33KB/sec
- 网络协议限制了物理资源的利用

Rdt 3.0: 停等操作



$$U_{\text{sender}} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

