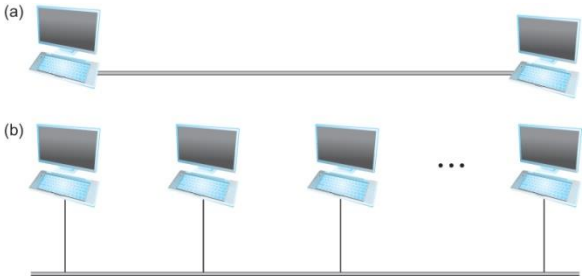
直接链接网络(I)

华中科技大学电子信息与通信学院
通信工程系
陈京文
Email: jwchen@hust.edu.cn
2020.10.2

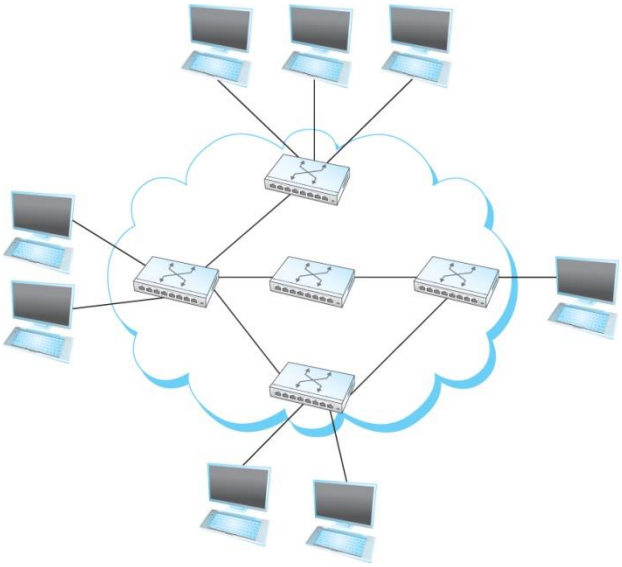
内容提要

- 直连网络概述
  - 基本问题，媒质，链路，编码
- 成帧
  - 面向字节，面向比特
- 差错检测
  - 二维奇偶校验，CRC
- 可靠传输
  - 基本要素，停止等待，滑动窗口

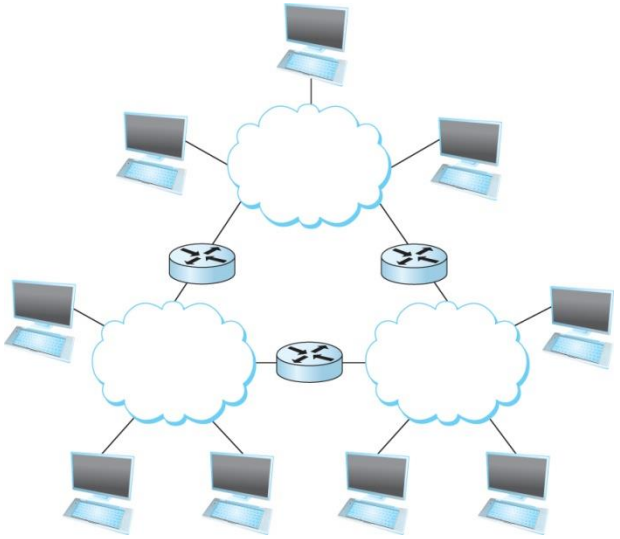
网络连通



直接链接



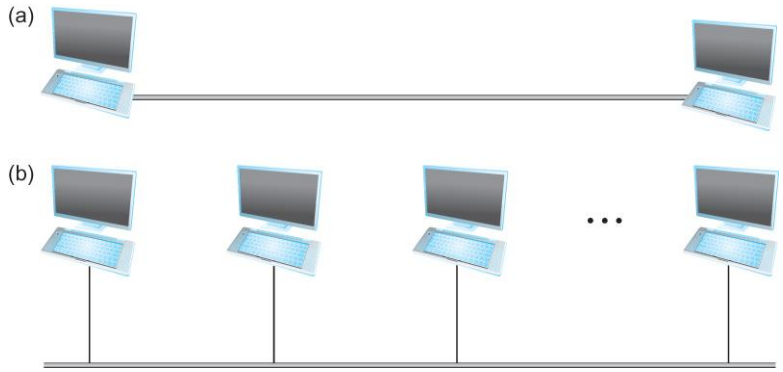
交换网络



互联网络

直接链接网络

- 直接链接网络
  - 主机通过物理媒质或逻辑通道直接相连
- 直接链接类型
  - 点对点，即两台主机通过一条点对点链路连接
  - 多路接入(multiple access)，即多台主机共同接入到一种媒质，实现相互之间的连接
    - 例：802.11无线局域网(Wi-Fi)，共享式以太网，共享式令牌环



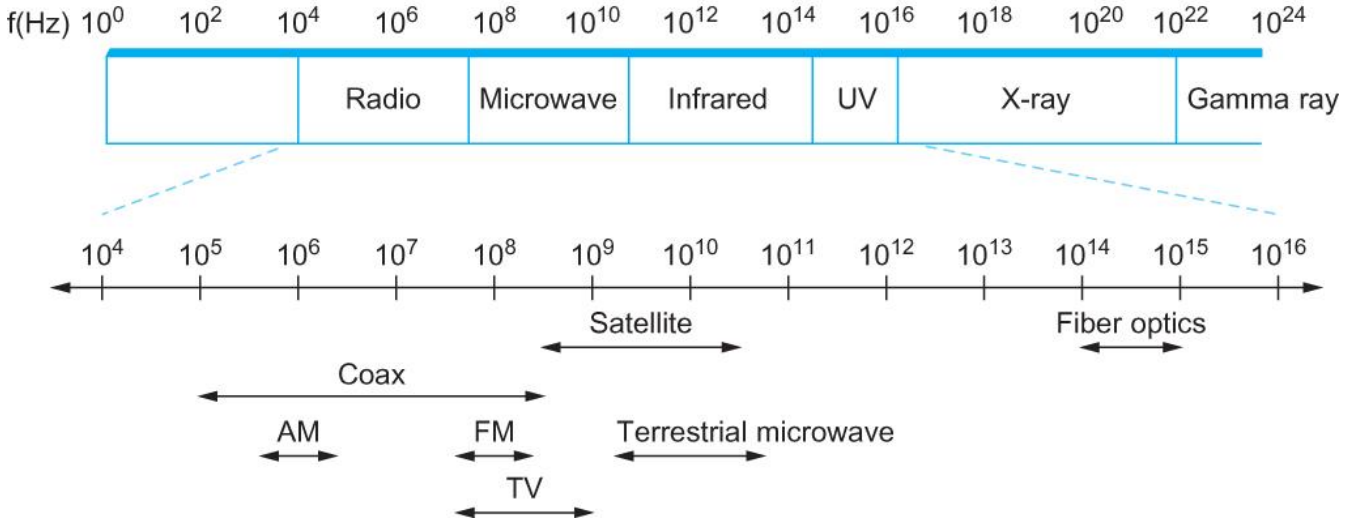
shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)

直接链接网络基本问题

- 编码(Encoding)
  - 如何对通过链路传输的数据比特进行编解码
- 成帧(Framing)
  - 给定收到的一系列连续的比特，如何确定相应的数据帧(不可分割的数据块)，即确定帧的起始和结尾
- 差错检测(Error detection)
  - 检测数据帧传输过程中产生的错误
- 可靠传输(Reliable transmission)
  - 纠正可能的传输错误，实现数据帧的可靠传输
- 媒体接入控制(Media access control)
  - 协调多台主机对于共享链路的接入(使用)

物理媒质

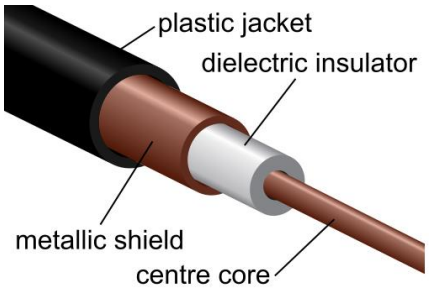


- 传播电磁波信号的物理媒质类型
  - 导波型(guided)：铜线，同轴电缆，光纤，双绞线
  - 非导波型(unguided)：空间(传输无线电信号)
  - 双工相关：全双工(full-duplex)，半双工(half-duplex)，单工(simplex)
- 将比特"编码"至信号
  - 低层次：调制(modulation)
  - 高层次：通过调制，将比特数据编码至两种不同信号
- 链路容量基本限制：香侬信道定理 —— $C = B\log_2(1+S/N)$

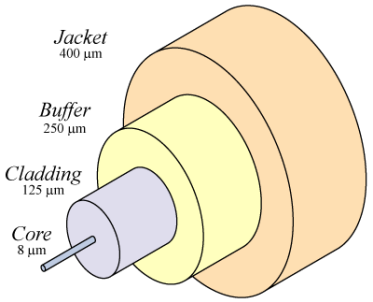
物理媒质：有线

- 双绞线(Twisted pair)
  - Category 5: 10-1000Mbps, 100m
  - Category 6: 10 Gbps

- 同轴电缆(Coaxial cable)
  - Thin-net coax: 10-100Mbps, 200m
  - Thick-net coax: 10-100Mbps, 500m



- 光纤(Fiber optics)
  - 多模(Multimode): 100Mbps, 2km
  - 单模(Single-mode): 100-40000 Mbps, 40km



物理媒质：无线

技术特征

- 无物理线缆(通过自由空间传输)
- 利用电磁波传输信息
- 双向性
- 传播环境的影响
  - 反射
  - 阻挡障碍
  - 干扰

主要类型

- 地面微波
  - 支持高达155Mbps通道
- 局域网(如WiFi)
  - 11Mbps, 54 Mbps, …
- 较大范围(如蜂窝网络)
  - 4G (LTE): ~ 10s Mbps
- 卫星
  - Kbps ~ 45Mbps
  - 270 ms端到端时延
  - 地球同步，低轨道

租用专线(Leased Line)

- 不同速率的例子
  - E1: 2 Mbps
  - OC-12 (STS-12): 622.080 Mbps
  - OC-48 (STS-48): 2.488320 Gbps

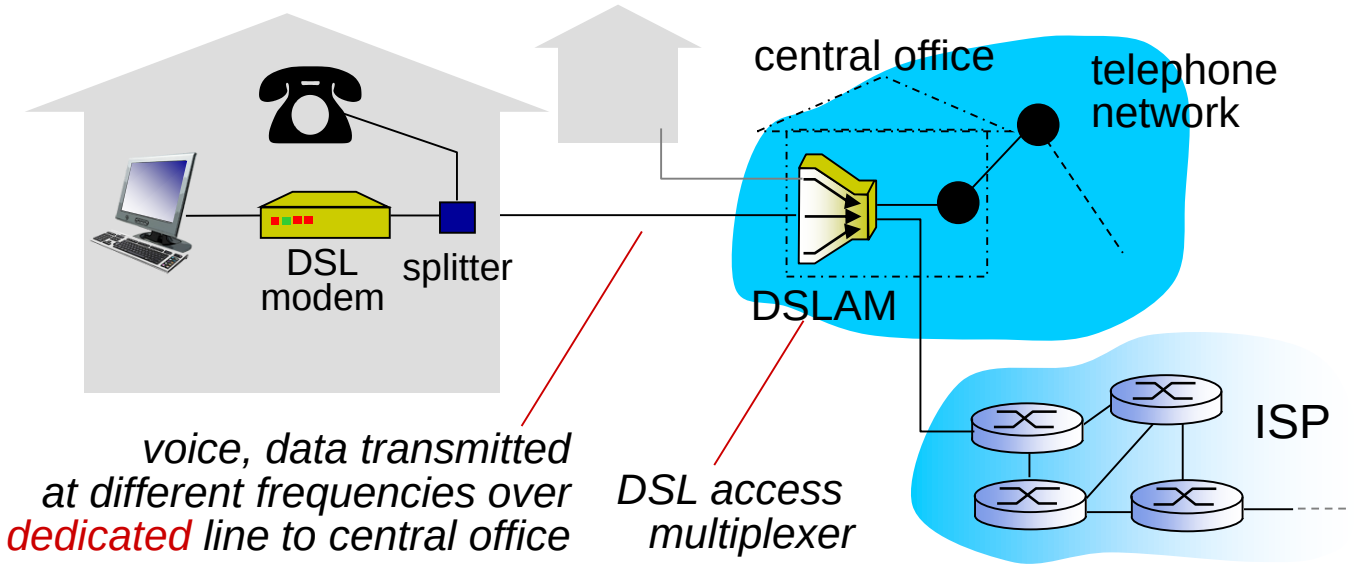
- 通常为运营商提供的一条逻辑电路，如一个高速率逻辑电路帧中的某个时隙

- 运营商通过其传送网提供相应的租用专线服务

接入链路

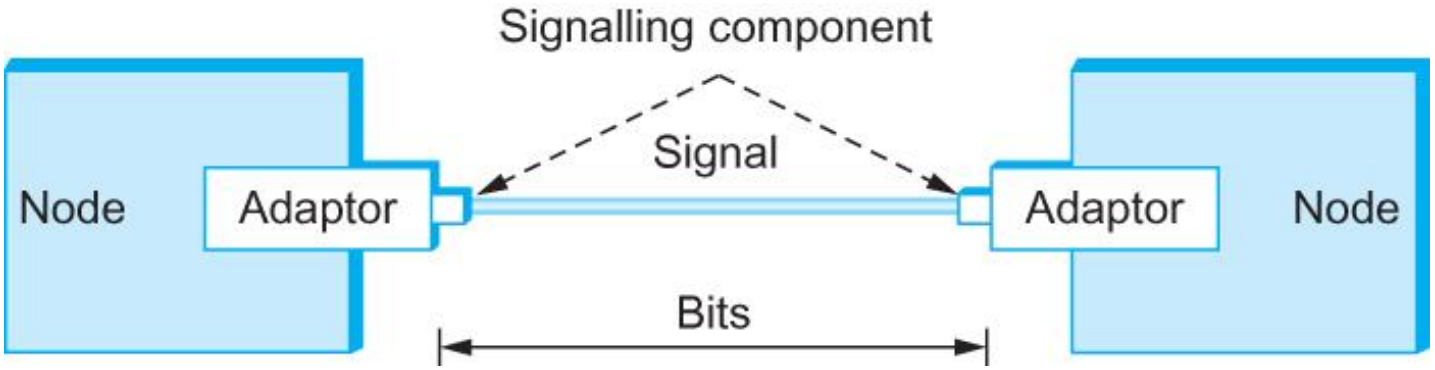
- 也称为“最后一公里(Last-Mile)”链路
  - 通常位于服务提供商至家庭用户的最后一公里
- 例子
  - POTS (Plain Old Telephone Service): 28.8~56 Kbps
  - ISDN (Integrated Services Digital Network): 64~128 Kbps
  - DSL: 16 Kbps ~ 55.2 Mbps，如ADSL
  - Cable modem: 20~40 Mbps
- POTS和ISDN可以提供至公众网络或私有网络的接入服务
  - 公众网络：为公众提供服务的网络，如Internet
  - 私有网络：并不向公众提供服务，如机构私有网络
- xDSL和Cable modem主要用于公众网络的接入，通常不会接入至私有网络

接入链路：xDSL



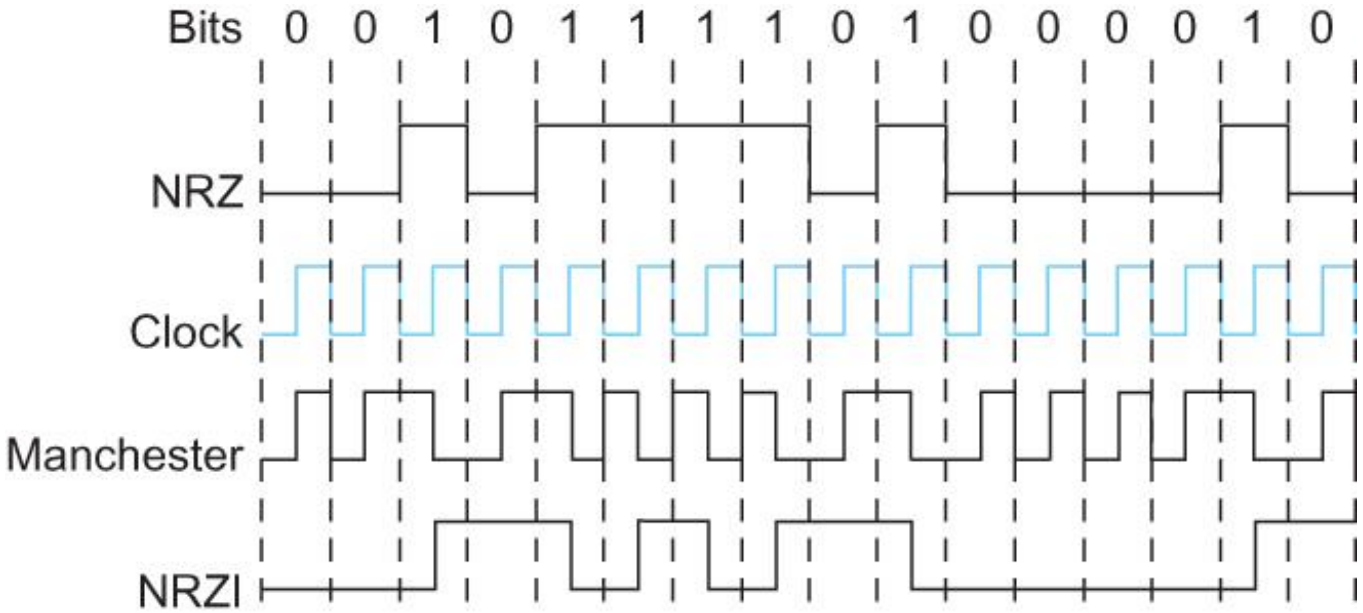
- 采用现有的电话线路，接入至集中的DSLAM (Digital Subscriber Line Access Multiplexer)
  - 数据信息传至Internet
  - 语音信号传至电话网络
- 上行传输速率< 2.5 Mbps (通常 < 1 Mbps)
- 下行传输速率< 24 Mbps (通常 < 10 Mbps)

比特与信号



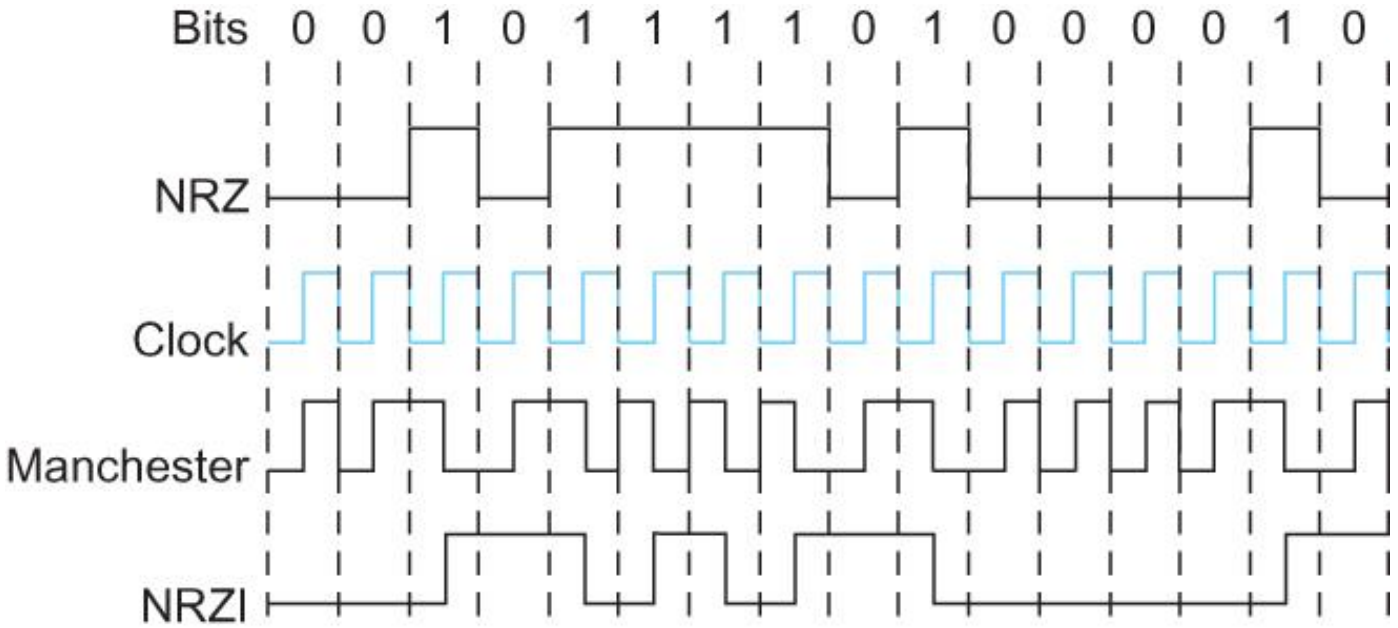
- 比特流编码至信号传输
  - 信号传输：信号模组之间
  - 比特传输：适配器之间
- 编码方式
  - NRZ, NRZI, Manchester, 4B/5B, …

比特流编码：NRZ，NRZI



- NRZ (Non-Return to Zero)
  - 比特为0、1，分别对应两种信号，如低、高电平
  - 接收端从接收信号中恢复出时钟，判决再生，得到原始比特流
  - 主要问题：比特流连续为1或0引起基线漂移、时钟恢复
- NRZI (Non-Return to Zero Inverted)
  - 比特为1：从当前信号跳变；比特为0：保持为当前信号
  - 解决了比特连续为1的问题

比特流编码： Manchester, 4B/5B

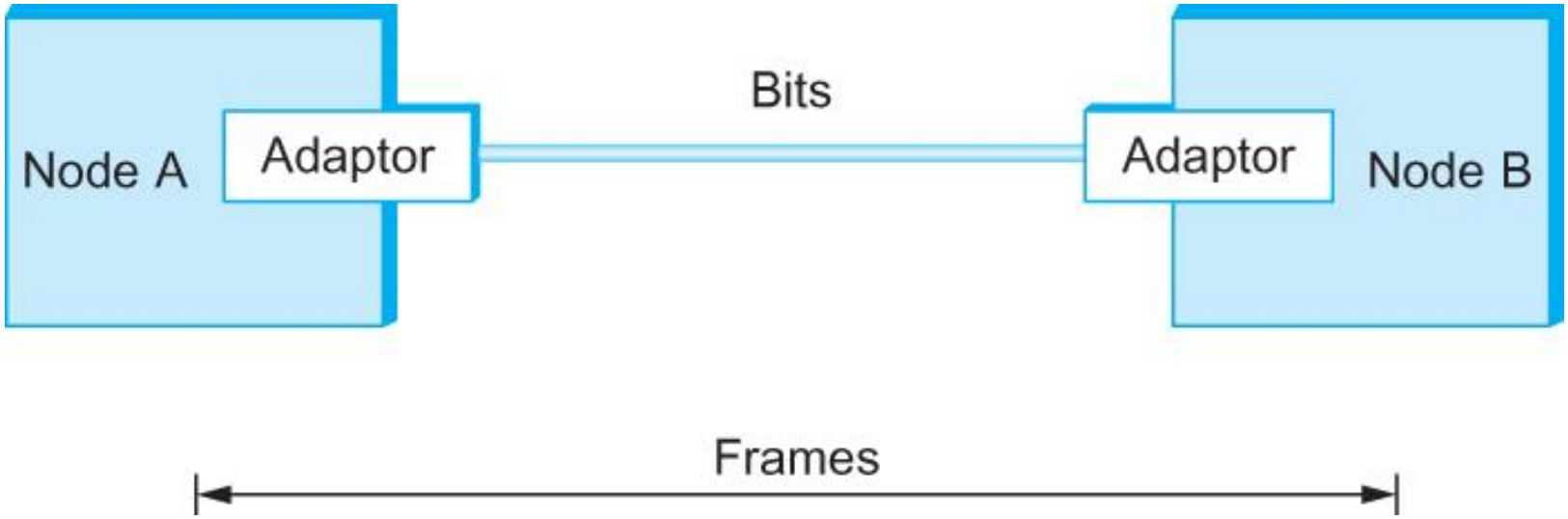


- Manchester编码：将NRZ编码数据与时钟信号异或
  - 比特为0：低至高跳变；比特为1：高至低跳变
  - 效率损失 —— 链路速率只有波特率(信号变化的速率)的一半
- 4B/5B：4比特数据一组，编码为5比特
  - 选择合适的编码，使得5比特编码组最多包含1个前导0，最多包含2个结尾0 —— 相邻两个5比特编码组中最多连续为0比特不超过3
  - 链路效率为80%

内容提要

- 直连网络概述
  - 基本问题，媒质，链路，编码
- 成帧
  - 面向字节，面向比特
- 差错检测
  - 二维奇偶校验，CRC
- 可靠传输
  - 基本要素，停止等待，滑动窗口

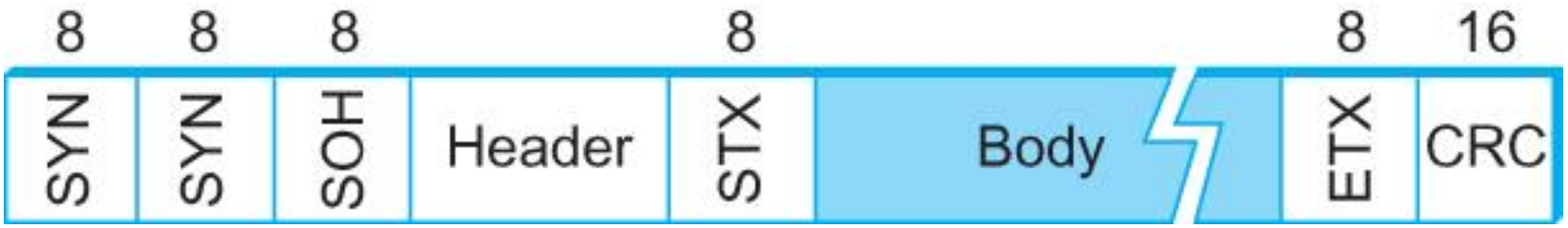
成帧(Framing)



- 两个节点之间的数据传输以一系列不可分割(单独传输)的数据块的形式进行，在数据链路层称为帧(Frame)

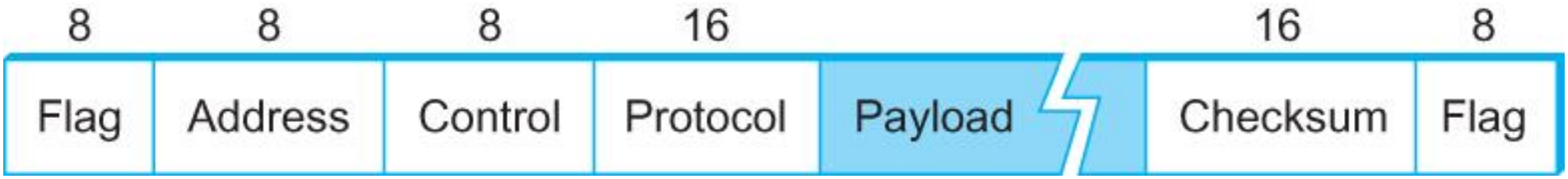
- 成帧(Framing)：在比特流中确定一帧的开始和结尾
  - 面向字节(Byte-oriented)：将每一帧视为一个字节(或字符)集合
  - 面向比特(Bit-oriented)：将每一帧视为一个比特集合

面向字节的成帧：哨兵式(Sentinel)



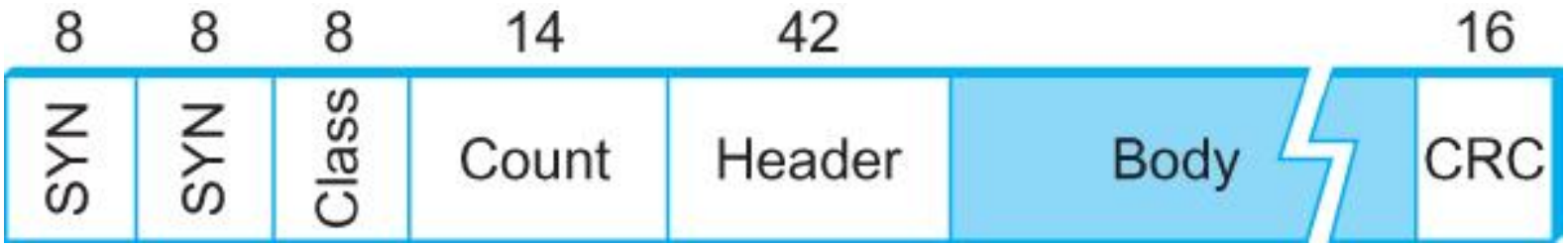
- BISYNC (Binary Synchronous Communication) Protocol
  - 1960年代IBM提出
  - 帧的数据部分处于哨兵字符之间
  - 哨兵字符 —— STX (start of text), ETX (end of text)
- 采用哨兵字符的问题
  - 帧的数据部分可能会出现ETX
- 解决方案 —— 字节填充
  - 在帧的数据部分的ETX之前加入DLE (data-link escape)字符
  - 在帧的数据部分的DLE之前插入额外的DLE字符

面向字节的成帧：哨兵式(续)



- PPP (Point-to-Point Protocol)
  - 在互联网接入链路中广泛使用
  - STX和ETX: `0x7E`
  - 另有LCP、NCP协议用于链路建立过程中相关参数的协商、网络层参数的设置

面向字节的成帧：字节计数



- 用一个字段专门给出帧长度
- DDCMP (Digital Data Communication Protocol)
  - 用于早期的DECNet
  - Count字段给出这一帧的字节长度
- 如定义帧长度的字段出错，会导致随后一系列帧定界错误

面向比特的成帧

- HDLC (High-level Data Link Control)
  - 基于IBM的SDLC，经ITU标准化，用于X.25
- 起始和结尾比特序列
  - **0x7E (01111110)**
- 类似于哨兵字节的问题：**01111110** 会出现在数据部分
  - 解决方案：比特填充

比特填充

- 发送方传送5个连续为1的比特时
  - 在下一比特前插入一个为0的比特

- 接收方收到5个连续为1的比特时
  - 如随后的比特为0：移除这一比特
  - 如随后两个比特为10：帧结尾
  - 如随后两个比特为11：错误

- 比特填充的特点
  - 可能会导致两个连续的帧接收失败
  - 帧长度取决于帧中的数据载荷长度

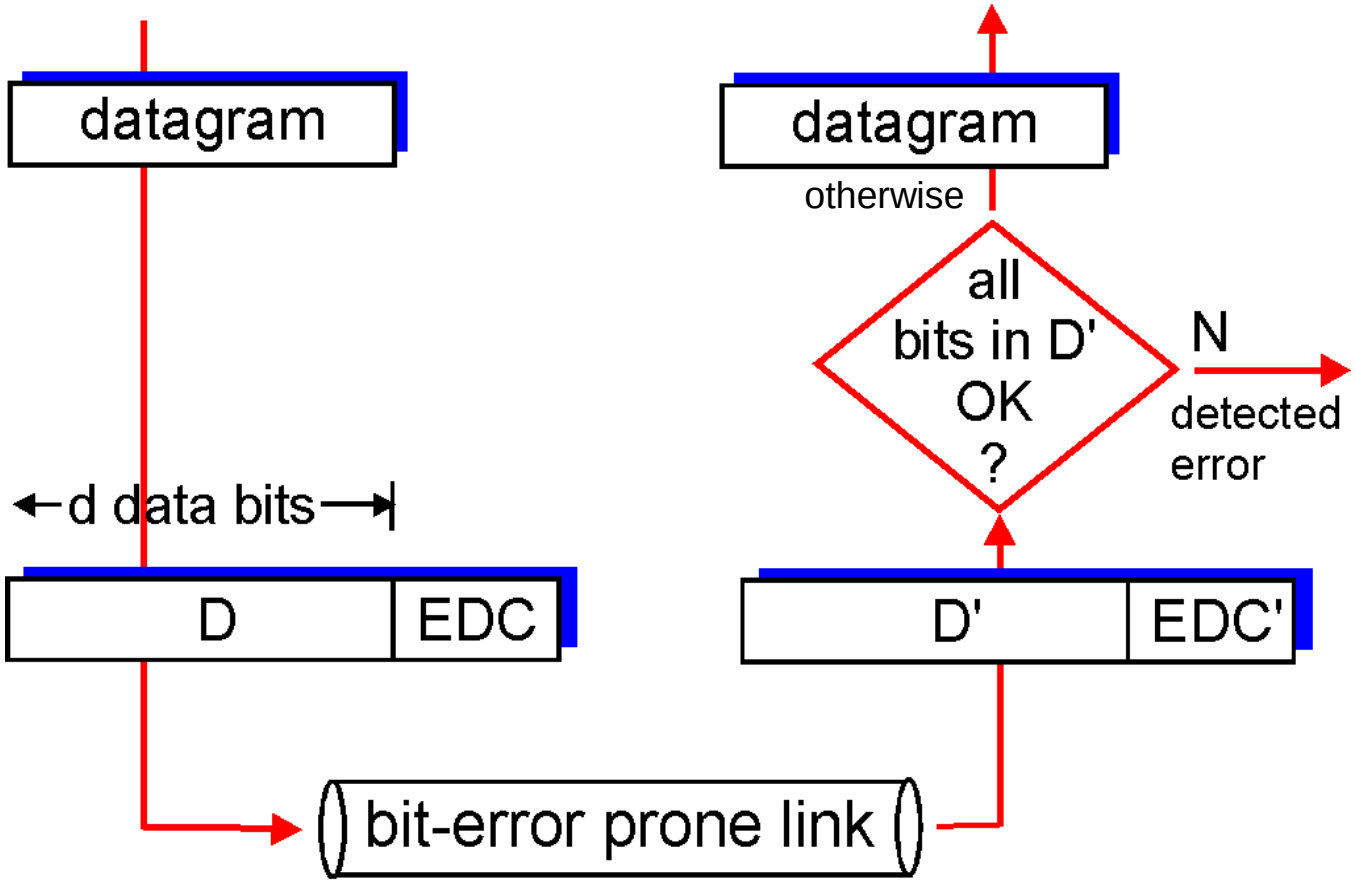
内容提要

- 直连网络概述
  - 基本问题，媒质，链路，编码
- 成帧
  - 面向字节，面向比特
- 差错检测
  - 二维奇偶校验，CRC
- 可靠传输
  - 基本要素，停止等待，滑动窗口

比特出错及纠错

- 比特错误问题
  - 传输过程中的噪声、干扰引起
    - 误码率(Bit Error Rate, BER)：取决于链路类型，如光纤链路出错率明显低于无线链路
  - 如未能检测到比特错误，会导致数据接收错误
- 许多网络应用(如Web)不能接受任何数据错误，需要可靠的数据传输
- 为了实现可靠传输，需要
  - 首先检测收到的帧/分组中的错误，
  - 继而纠正错误(参见本讲义下一部分)

错误检测

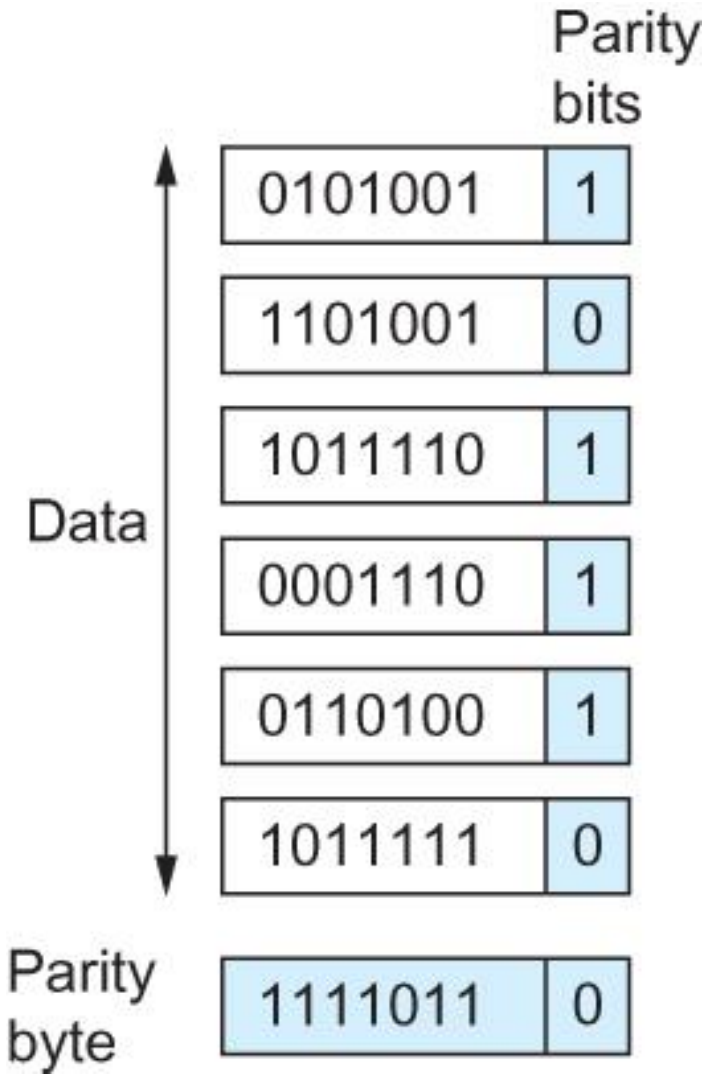


EDC = Error Detection Code (额外添加的码元)
D = 待保护数据 (包括载荷和首部等)

错误检测(续)

- 错误检测方法并非百分百可靠
  - 可能会错过极少数错误
  - 较长的检错码(EDC)通常会提供较好的检测和纠错性能
- 主要目标
  - 低冗余度
    - 令n表示数据长度，k表示ECC长度
    - 要求k远小于n
  - 高检错率
    - 最大化错误被检测到的概率
- 几种常见的检错码
  - 二维奇偶校验(Two-Dimensional Parity)
  - Internet校验和(Internet Checksum)
  - 循环冗余校验(Cyclic Redundancy Check, CRC)

二维奇偶校验



- 效果
  - 可检测到所有1、2、3比特错误，和大部分4比特错误
- 用于BISYNC协议中传输ASCII编码字符

CRC

- 循环冗余校验(CRC)
  - 理论基础源于名为有限域(Finite Fields)的数学分支

- 相关数学符号
  - 需传输数据：比特数据视为n-1阶多项式***M(x)***
    - 示例：8比特数据10011010 —— $M(x)= x^7+x^4+x^3+x^1$
  - 检错码：多项式***P(x)***
    - 计算基于k阶生成多项式***C(x)***

      示例：$C(x)= x^3+x^2+1$，其中$k=3$
    - $C(x)$的选择对于检错性能有着重要影响

CRC (续)

- 检错：接收方验证$P(x)$能否被$C(x)$整除(模2算术意义)
  - 无错：余数为0
  - 有错：余数不为0

- 如何计算$P(x)$
  - $M(x)$乘以x^k (即在$M(x)$之后添加个k个)，得到$T(x)$
  - $T(x)$除以(模2算术)$C(x)$，得到余数
  - $T(x)$减去(模2算术，相当于异或)上述余数，即为$P(x)$ — 编码数据
    - 相当于原始数据结尾处附加上述除法运算的余数

模2算术运算

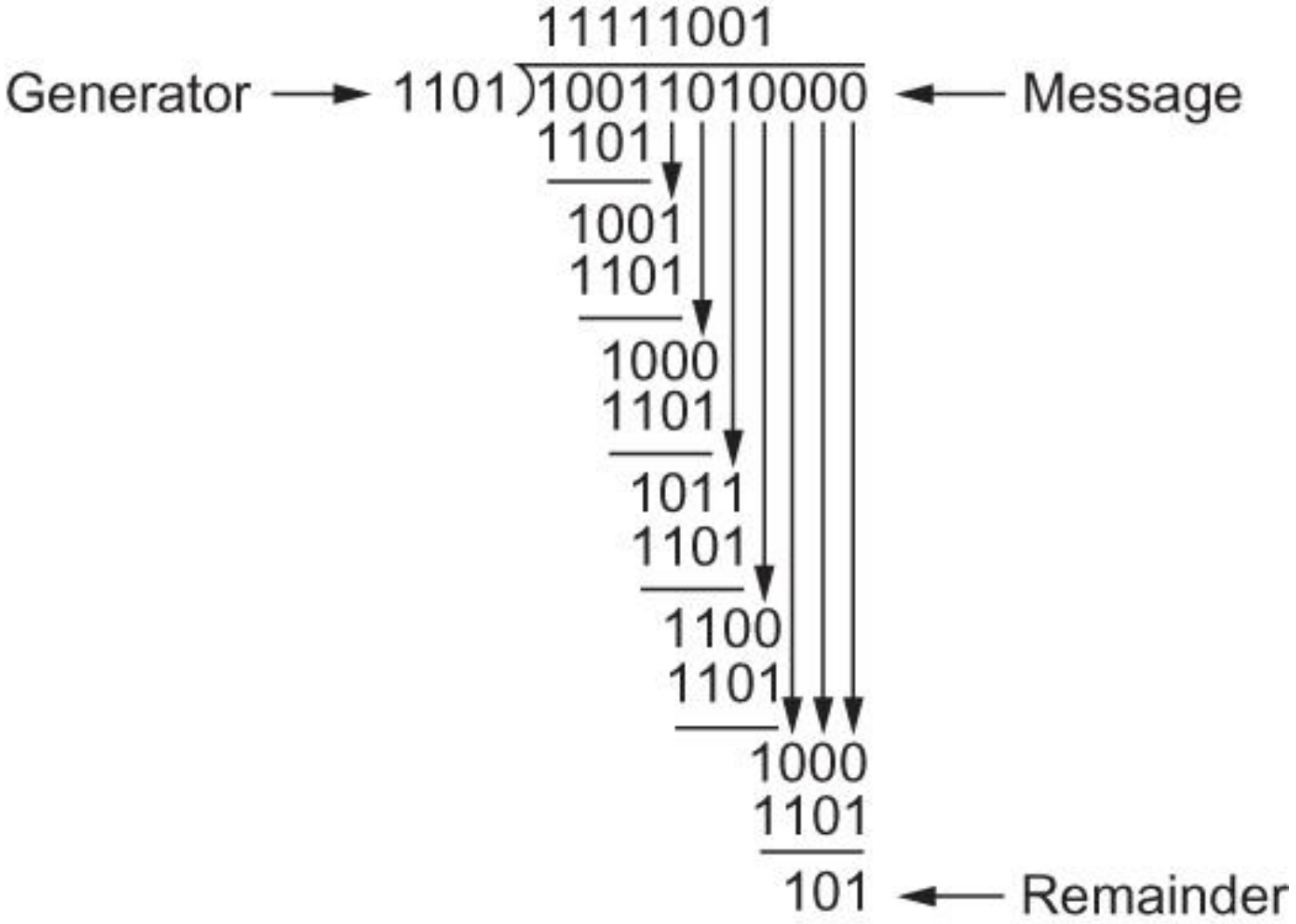
- 类似于普通二级制算术，但无需向相邻位进位和借位
- 示例

```
101 +     101 +     1011 +          101 -     101 -     1011 -
010       001       0111            010       001       0111
---       ---       ----            ---       ---       ----
111       100       1100            111       100       1100
```

- 模2加法和减法相当于异或运算

a	b	$a \otimes b$
0	0	0
0	1	1
1	0	1
1	1	0

CRC运算示例


11111001
Generator
1101
10011010000
Message
1101
1001
1101
1000
1101
1011
1101
1100
1101
1000
1101
101
Remainder

CRC生成多项式

- 生成多项式$C(x)$的选择
  - 采用合理的$C(x)$可以检测到
    - 所有1或2比特、任意奇数个比特、长度少于k的连续比特错误
    - 部分长度大于k的连续比特错误
  - 采用某些$C(x)$可以检测到特定的比特错误
- 常见国际标准CRC生成多项式
  - CRC-8: x^8+x^2+x+1
  - CRC-10: $x^{10}+x^9+x^5+x^4+x+1$
  - CRC-12: $x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16: $x^{16}+x^{15}+x^2+1$
  - CRC-CCITT: $x^{16}+x^{12}+x^5+1$
  - CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
- 易于采用硬件电路(k-bit移位寄存器和异或门)实现CRC

内容提要

- 直连网络概述
  - 基本问题，媒质，链路，编码
- 成帧
  - 面向字节，面向比特
- 差错检测
  - 二维奇偶校验，CRC
- 可靠传输
  - 基本要素，停止等待，滑动窗口

可靠传输

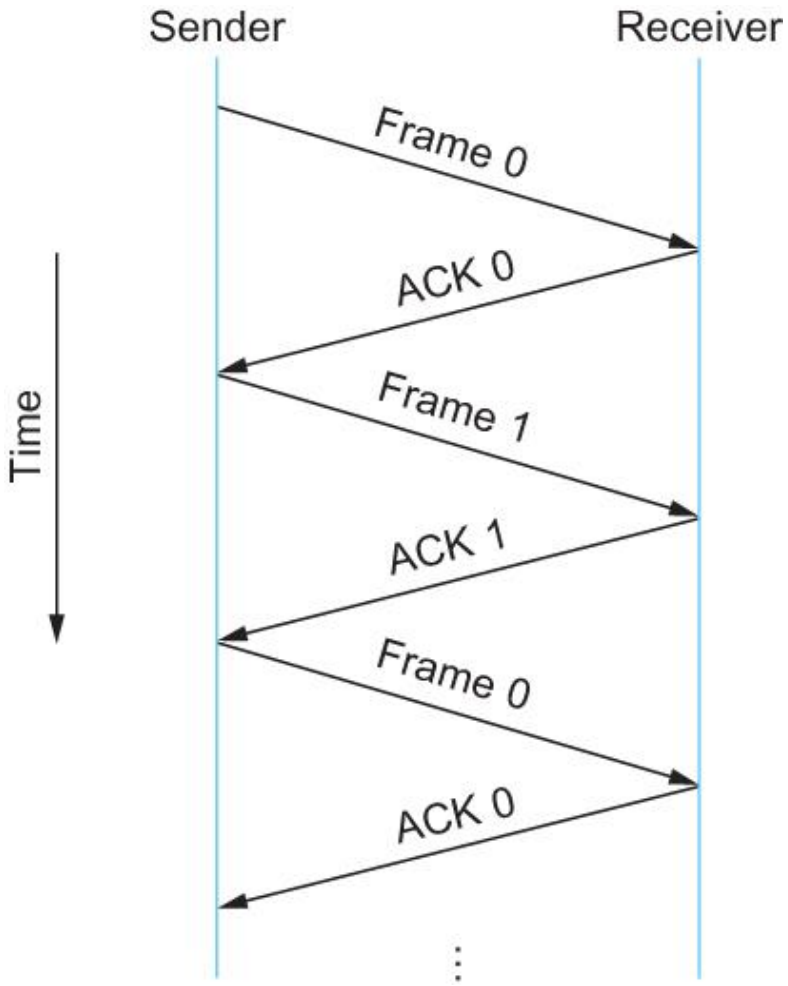
- 可靠传输问题：如何恢复丢弃的分组
  - 原因：比特错误导致分组出错，因此丢弃
    - 注：另一导致分组丢弃的原因是缓存溢出(特别是上层)
- 可靠传输的技术方案
  - 重传(retransmission)：接收方检测到错误时，通知发送方分组出错，发送方因此重传数据分组，例如ARQ (参见下一页)
  - 纠错码(Error Correction Code, ECC)：发送方在待传数据之外添加纠错码，接收方采用纠错码特定的算法重构数据
    - 纠错的水平取决于编码类型
      - 一般而言，添加的编码比特越多，纠错能力越强
    - 但并不能保证100%的错误都能纠正
  - 重传 vs. ECC?
- 可靠分组传输的技术解决方案适用于所有层，不仅仅是数据链路层

ARQ

- 自动重发请求(Automatic Repeat reQuest, ARQ)
  - 确认(Acknowledgement, ACK)：接收方通知发送方正确地收到一帧或多帧
  - 超时(Timeout)：发送方在发出帧之后，启动关联的定时器，经过将在设定时间后超时
  - 重传(Retransmission)：如果关联定时器超时，发送方重传相应的帧至接收方
  - 帧序号(Sequence number)：用于辨识帧，避免重复

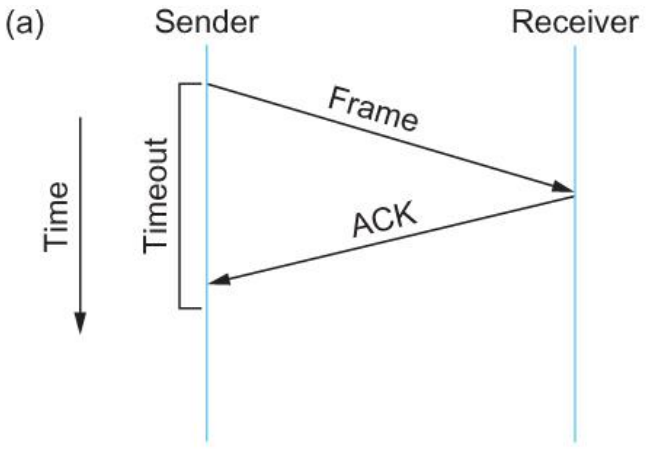
- ARQ的类型
  - 停止等待(stop-and-wait)
  - 滑动窗口(sliding window)

停止等待(Stop-and-Wait)

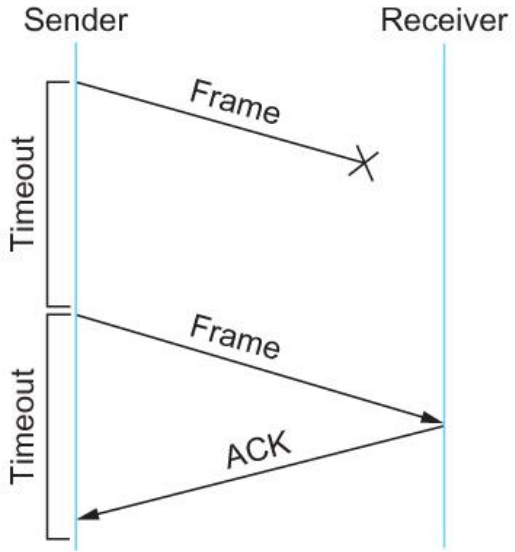


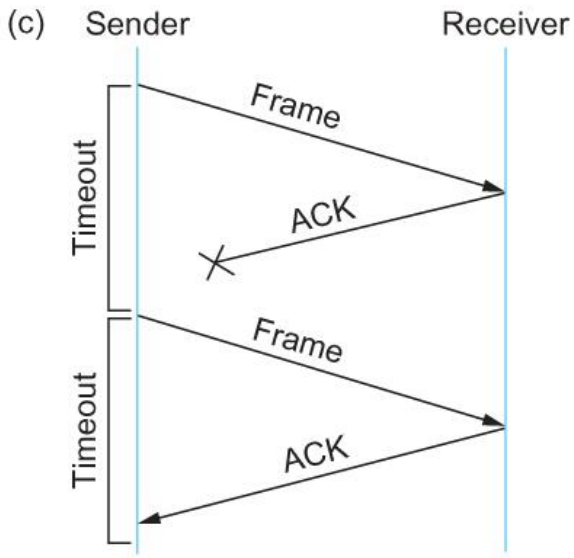
停止等待的可能场景

正常情况

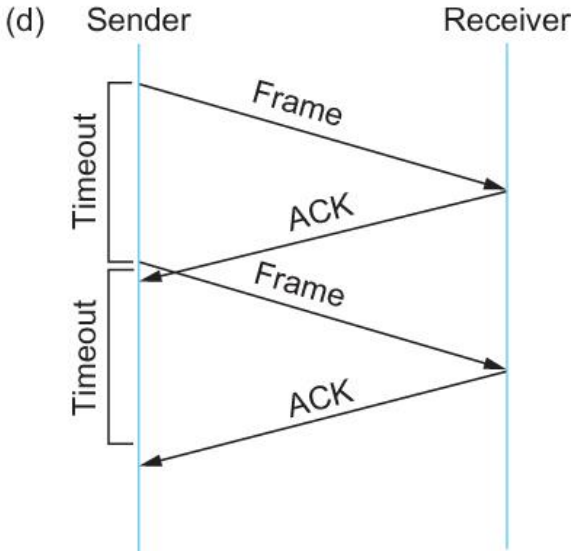


帧丢失：
超时后重传





ACK丢失：
超时后重传

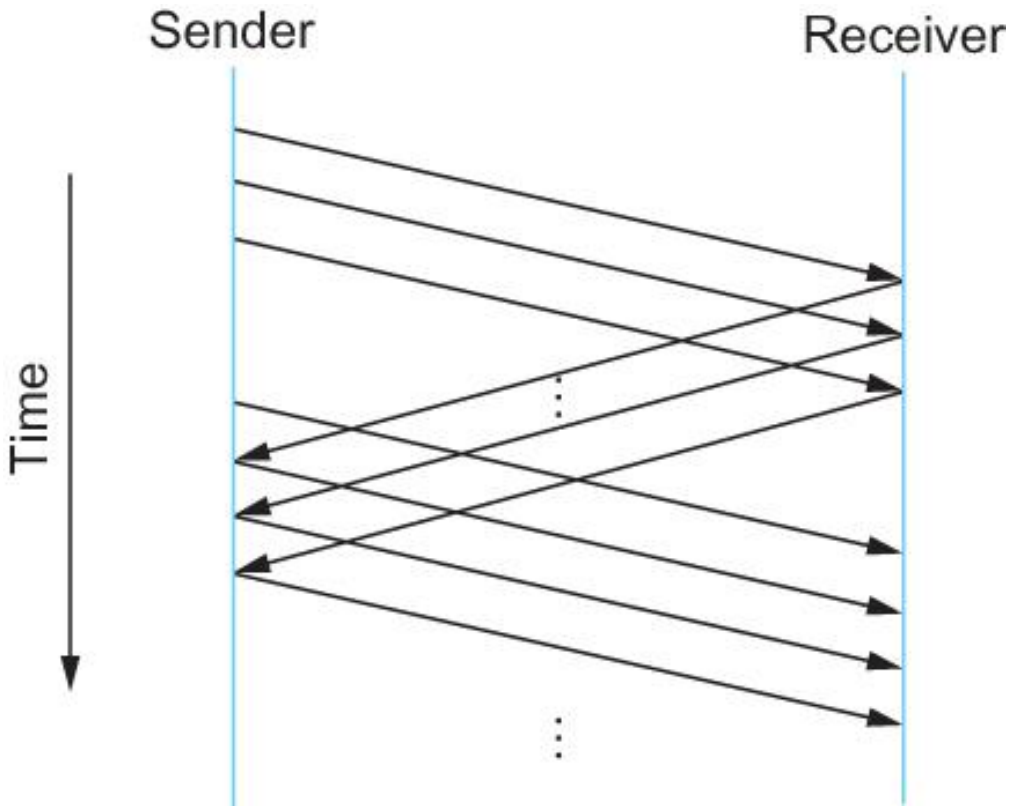


ACK迟到：
超时后重传

停止等待算法的特点

- 最简单的ARQ
- 1比特顺序号已然够用
- 链路带宽利用效率低
  - 例如：链路带宽为2 Mbps, RTT为45 ms, 帧长1.5KB
    - 每个RTT内，发送方只能发送1帧
    - 实际数据发送率为1500x8/0.045 = 266.7Kbps
- 改进：滑动窗口(Sliding window)

滑动窗口算法的核心思路



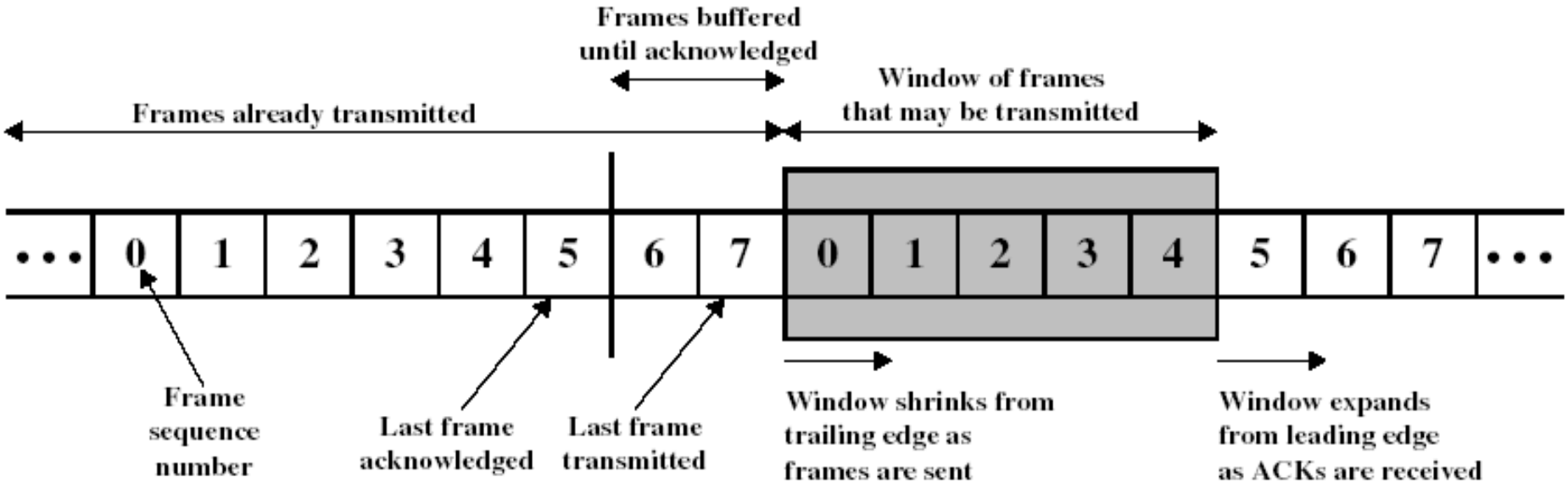
发送方批量发送，以充满管道(Keep the pipe full)

发送方

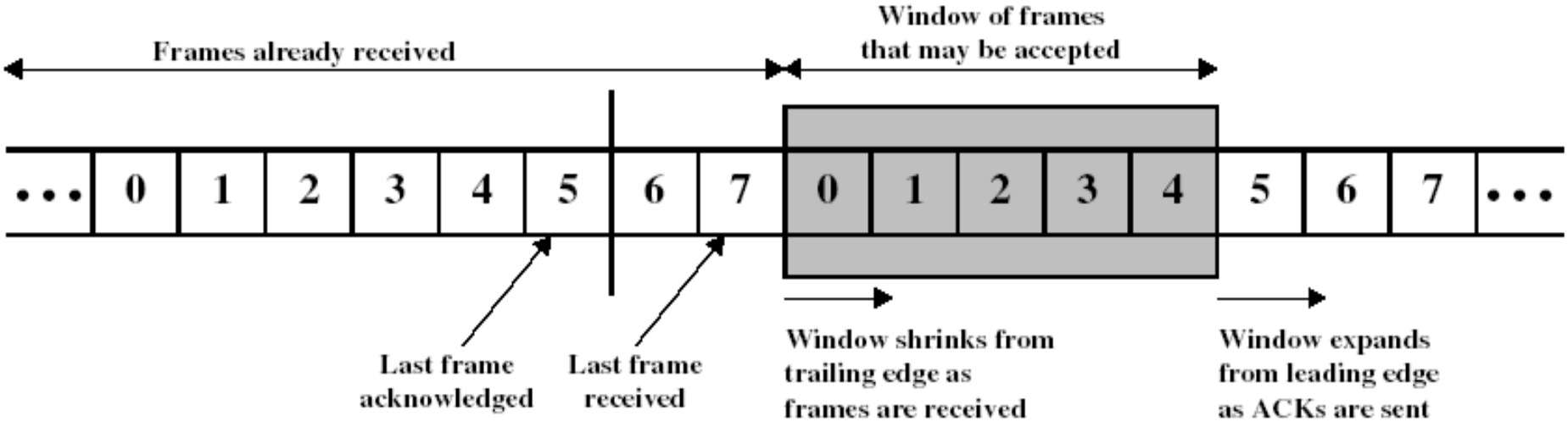
- 维护一个关于帧序号(SeqNum)的发送窗口
  - 只能发送SeqNum在发送窗口范围内的帧
  - 发送帧时，发送窗口缩减相应数据量
  - 收到ACK时，发送窗口扩展相应数据量
  - 发送窗口大小(Send Window Size, SWS)：能够未经收方确认而连续发送的帧数据量上界
  - 意即：未收到相应ACK之前，发送方可连续发送SWS数据量的帧
- 已发送的帧，还需缓存(以备重传)直至收到接收方的确认
- 帧序号
  - 固定比特长度：k比特
  - 取值范围：$0 \sim 2^k-1$

接收方

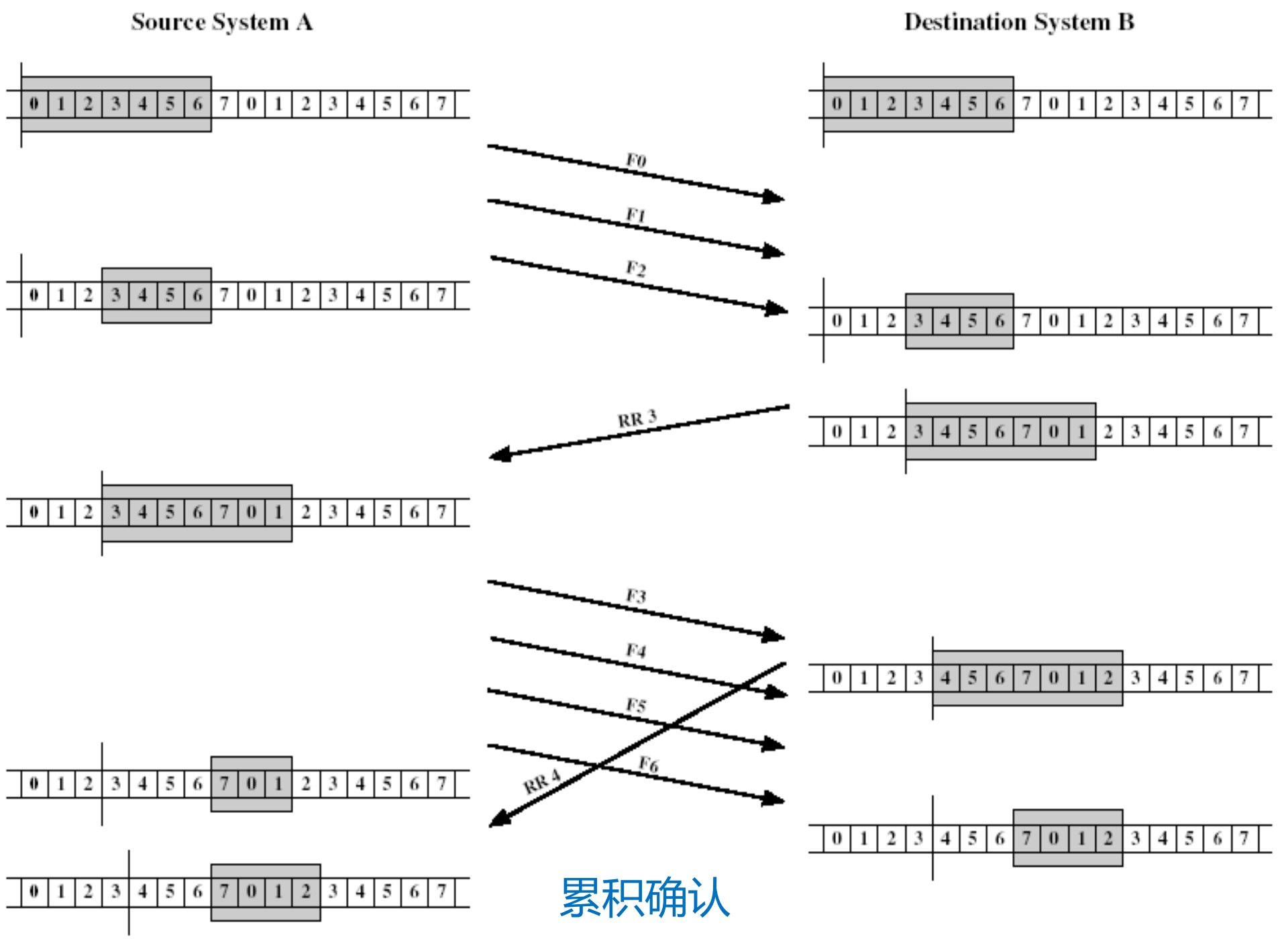
- 维护一个关于帧序号(SeqNum)的接收窗口
  - 只能接收SeqNum在接收窗口范围内的帧
  - 帧的接收：接收窗口缩减相应数据量
  - ACK的发送：接收窗口扩展相应数据量
  - 接收窗口大小(Receive Window Size, SWS)：给出可接收且未进行确认的帧数据量上界
- 确认正确收到的帧
  - 累积确认：确认消息中的帧序号MaxSeqNumRcvd，表示所有SeqNum < (或<=) MaxSeqNumRcvd的帧全部正确收到
    - 例如：正确收到帧2、3、4，回复MaxSeqNumRcvd = 5的ACK消息，以确认成功收到SeqNum < 5的帧(即帧2、3、4)
  - 每帧确认：确认每一个争取收到的帧
    - 需要更多处理
  - 多帧确认：单个ACK消息中确认多个帧的收讫
    - ACK帧结构复杂

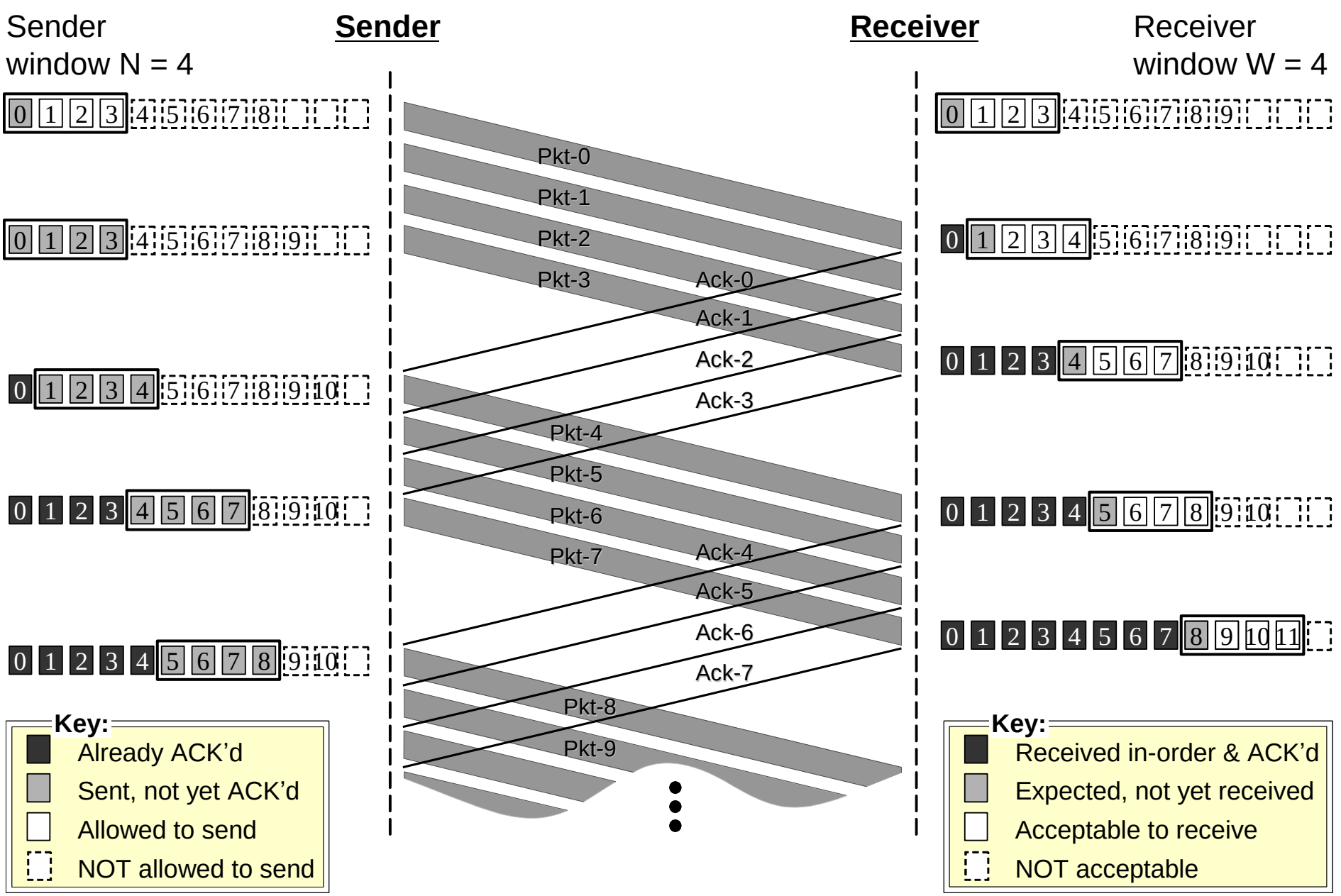


(a) Sender's perspective



(b) Receiver's perspective


Source System A
Destination System B
F0
F1
F2
RR 3
F3
F4
F5
F6
RR 4
累积确认


Sender
Sender
Receiver
Receiver
window N = 4
window W = 4
Pkt-0
Pkt-1
Pkt-2
Pkt-3
Ack-0
Ack-1
Ack-2
Ack-3
Pkt-4
Pkt-5
Pkt-6
Pkt-7
Ack-4
Ack-5
Ack-6
Ack-7
Pkt-8
Pkt-9
Key:
Already ACK'd
Sent, not yet ACK'd
Allowed to send
NOT allowed to send
Key:
Received in-order & ACK'd
Expected, not yet received
Acceptable to receive
NOT acceptable

每帧确认

帧丢失情况下的确认

- 偶发性帧丢失会导致发送端降速，不能有效利用链路容量
  - 发送方收不到丢失帧的确认，只能等候定时器超时，同时也不能进一步滑动发送窗口
    - 例如：发送了帧6、7、8，发送窗口消耗了3帧大小，帧6丢失，帧7、8正确收到
  - 如接收方采用合理的确认方式，有助于发送方及早知道，从而能够及早重发和释放发送窗口

- 三种方法
  - 重复确认(Additional Acknowledgement)
  - 负面确认(Negative Acknowledgement, NAK)
  - 选择性确认(Selective Acknowledgement)

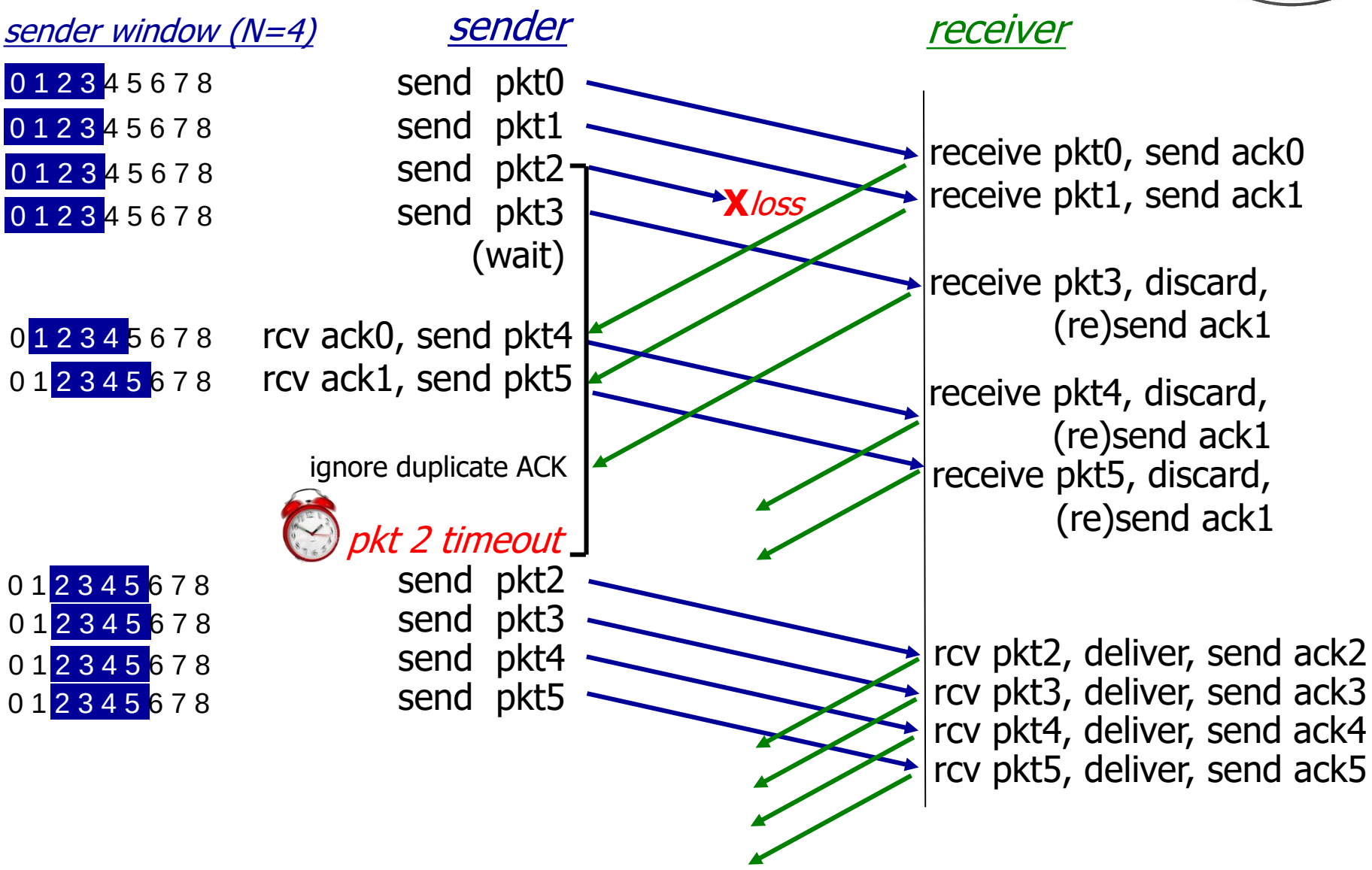
帧丢失情况下的确认(续)

- 重复确认
  - 接收方重复确认前一帧的收到，如前例，接收方重复确认帧5
  - 发送方可将收到的重复确认作为帧丢失的线索
- 负面确认
  - 接收方负面确认某一帧，表示应收到(因为后续帧已收到)，但没有
    - 如前例，负面确认帧6
  - 功效与发送方超时机制重复，同时增加了接收方实现的复杂性
- 选择性确认
  - 接收方只确认正确收到的帧，发送方据此可获知帧丢失信息
    - 通常采用多帧确认，原理上也可采用每帧确认
    - 如前例，单个确认消息确认帧5、7、8
  - 增加了接收方实现的复杂性
- 通常采用重复确认

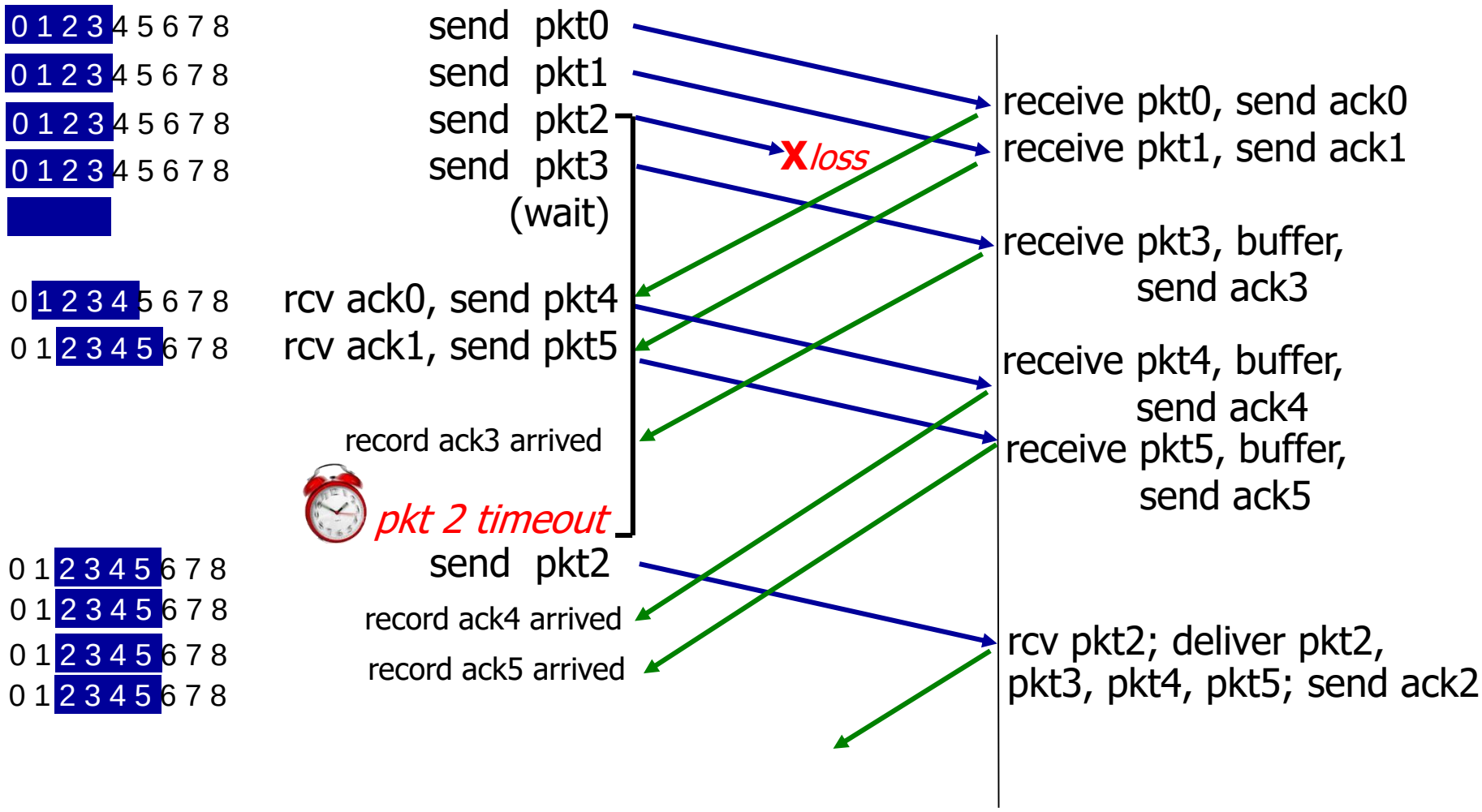
帧丢失情况下的重传

- 回退N帧(Go-back-N)重传
  - 设发送窗口大小SWS为N
  - 如序号为k的帧丢失，则重传序号为$k \sim k+N-1$的所有帧
  - 正确收到的帧也需要重传，因此效率不高
  - 接收方无需缓存失序帧，因此接收缓存简单
- 选择性重传(Selective Repeat)
  - 仅仅重传丢失的帧
  - 效率高
  - 接收方需要缓存失序帧
- 通常采用选择性重传

回退N帧重传：每帧+重复确认



选择性重传：每帧确认


sender window (N=4)
sender
receiver
0 1 2 3 4 5 6 7 8
send pkt0
send pkt1
send pkt2
send pkt3
(wait)
receive pkt0, send ack0
receive pkt1, send ack1
X loss
receive pkt3, buffer,
send ack3
rcv ack0, send pkt4
rcv ack1, send pkt5
receive pkt4, buffer,
send ack4
receive pkt5, buffer,
send ack5
record ack3 arrived
pkt 2 timeout
send pkt2
record ack4 arrived
record ack5 arrived
rcv pkt2; deliver pkt2,
pkt3, pkt4, pkt5; send ack2

窗口大小

- 由于帧序号(SeqNum)的循环使用，需要区分不同轮次循环中具有相同序号的帧(实为不同帧)
  - 设帧序号使用*k*比特，发送和接收窗口大小需满足

    SWS + RWS <= 2^k

- 关于帧到达顺序的隐含假设：传输过程中帧不会重新排列(reordered)
  - 适用于链路层传输
  - 但对于网络层及以上层，并不成立

滑动窗口算法的应用

- 可靠数据传输
- 保持正确的分组接收顺序
  - 链路层能够维持正确的帧接收顺序，将收到帧所封装的数据分组按照正确顺序交给上层协议实体
- 流量控制
  - 同步发送方发送数据、接收方接受数据的速率
  - 主要实现方式：由接收方通知发送方，自己当前接收窗口的有效大小

小结

- 直接链接网络基础技术
  - 基本问题，传输媒质，链路，编码，成帧
- 可靠传输
  - 错误检测：二维奇偶校验，CRC
  - 滑动窗口：基本要素，工作原理，确认方式，重传方法，窗口大小
- 参考文献
  - 教材2.1~2.5节
  - [KR12] 3.4节