第三单元 数据链路层

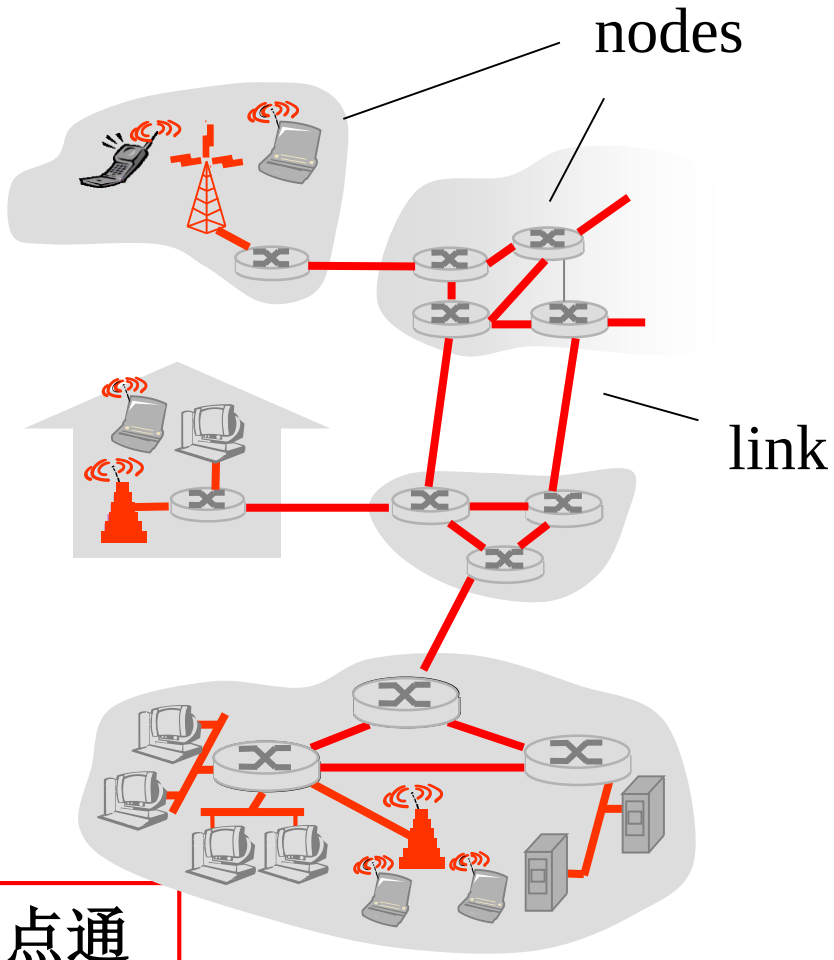
- 概述
- 差错检测
- 可靠数据传输
- 停等协议
- 滑动窗口协议
- PPP协议

2019.3.13

概述

一些术语:

- 主机和路由器是节点(nodes)
- 连接相邻节点的通道是链路(links)
  - 有线链路(wired links)
  - 无线链路(wireless links)
  - 局域网(LANs)
- 第2层的数据包(packet)是帧(frame)

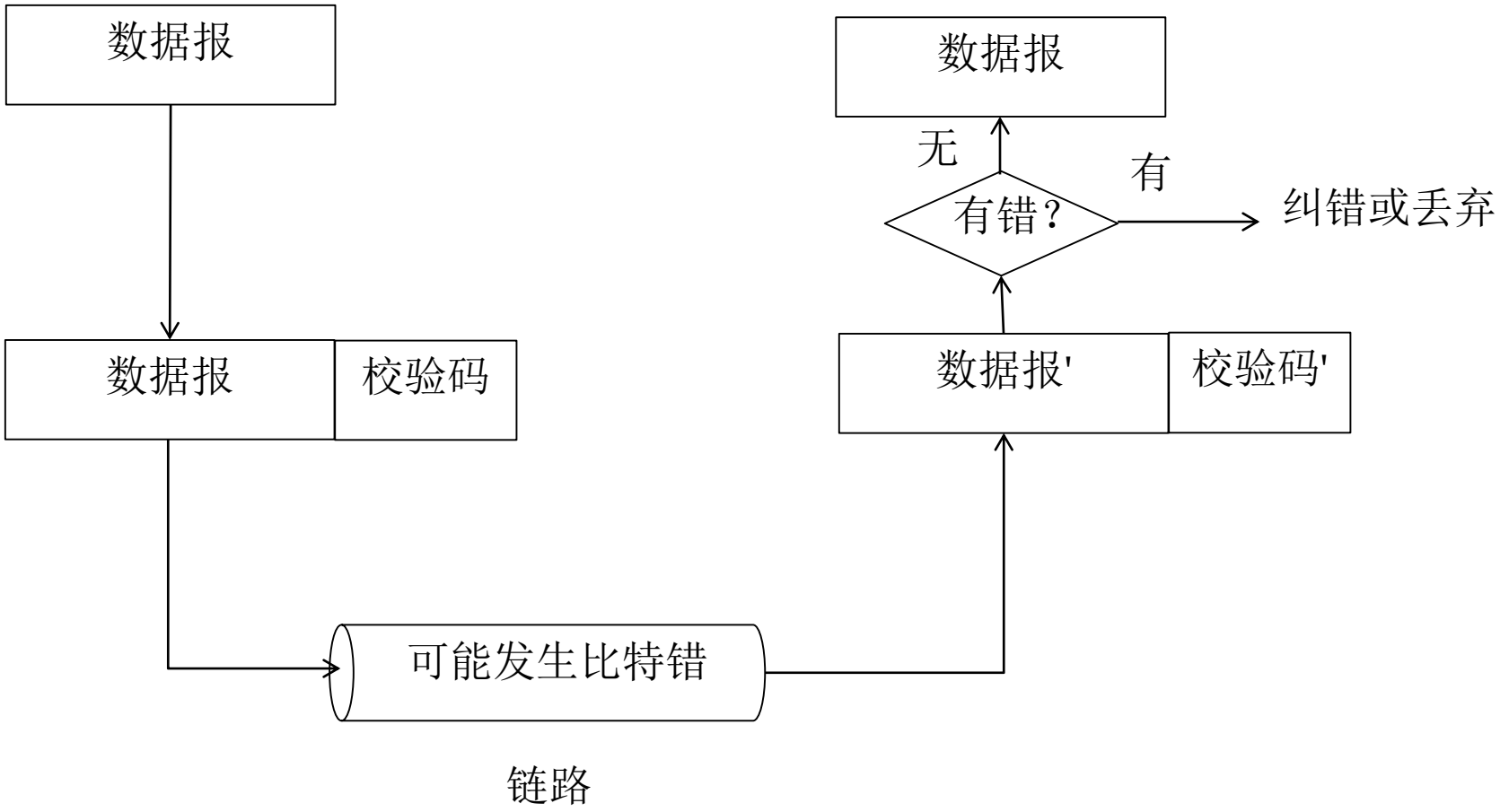


数据链路层负责把数据包从一个节点通过链路（直连网络或物理网络）传给相邻的另一个节点。

功能

- 形成帧 (framing)
- 差错检测(error detect)：比特错，纠错
- 差错控制(error control)：丢包、重复、错序。
  流控制(flow control)
- 介质访问控制(medium access control): 多路访问，碰撞(collision)

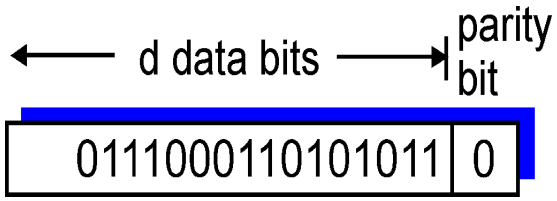
差错检测



奇偶校验

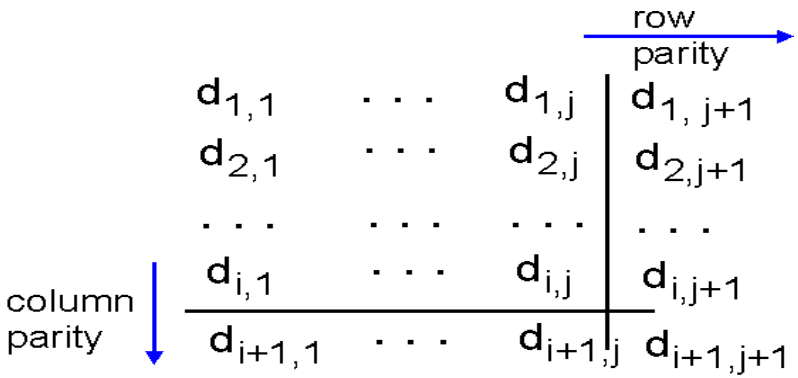
Single Bit Parity:

Detect single bit errors



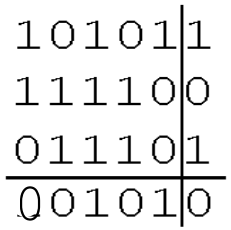
Two Dimensional Bit Parity:

Detect and correct single bit errors

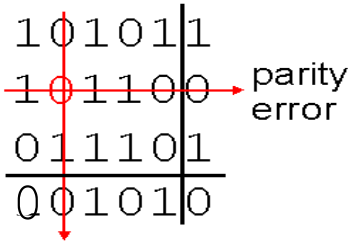


哪位有错？

1 0 1 1 1
1 0 0 0 1
0 1 1 1 1
1 0 1 0 1
1 1 0 0 0



no errors



correctable single bit error

检验和

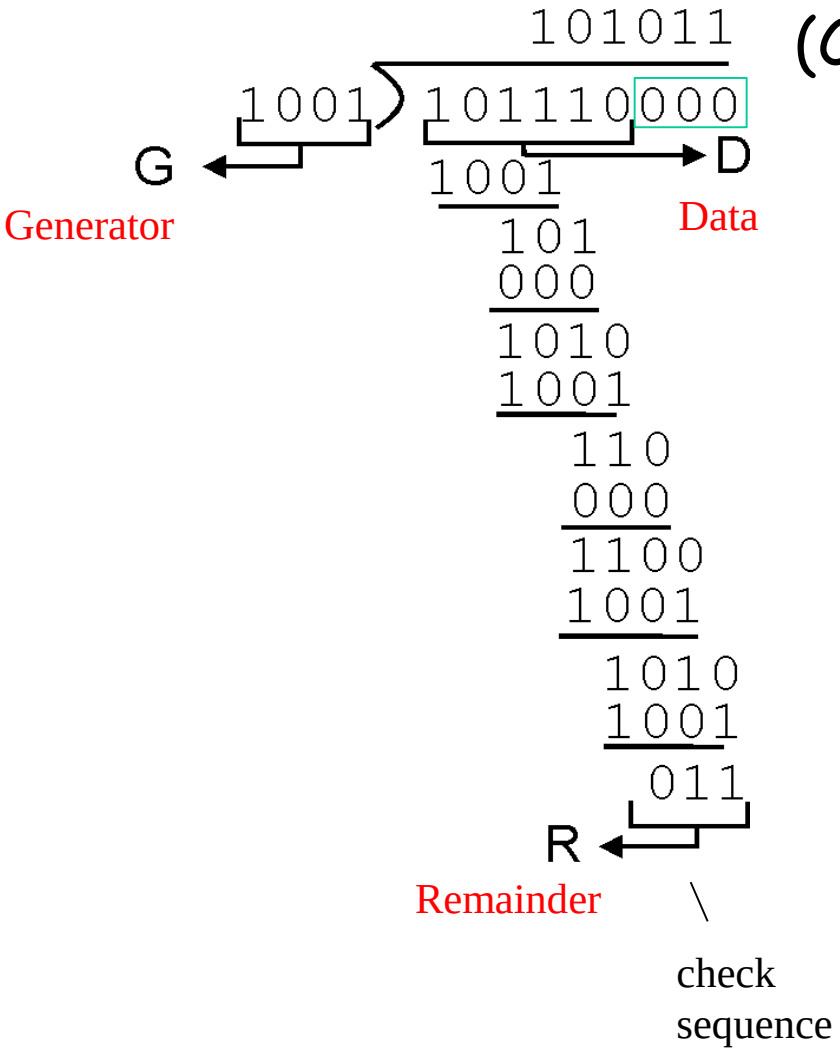
(Checksum)

```
  10000110 10000111
+ 00000100 01000100
-------------------
  10001010 11001011
+ 11000000 00000000
-------------------
 1 01001010 11001011
+                  1
-------------------
  01001010 11001100
```

反码：1011 0101 0011 0011

由于需要使用加法器实现算法，校验和一般不用于数据链路层，而是更高层，例如，IP层和传输层。

循环冗余校验码



(Cyclic Redundancy Check, CRC)

最后结果：101110 011

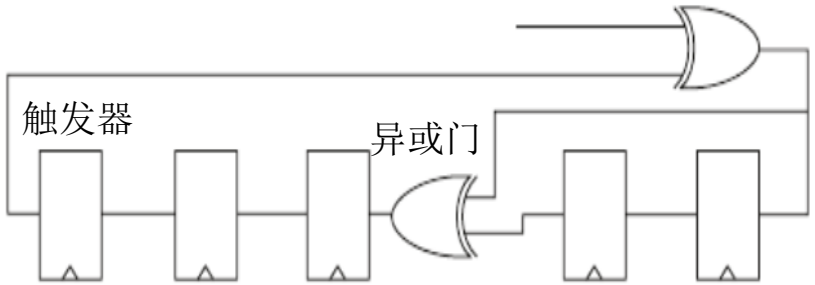
数据　校验码

如果传输过程中没有出现比特错，接收方用相同的除数去除数据加CRC校验码，余数应该为0。

采用模2除法：做减法时没有借位，类似于按位异或。

链路层常用CRC检验，因为容易用硬件实现，速度快，检错率很高。

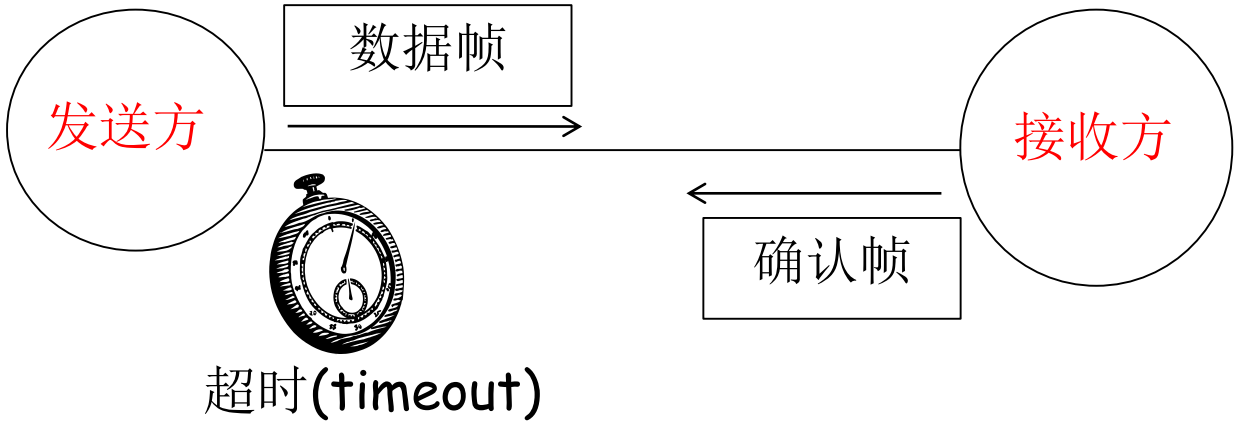
CRC-CCITT($x^{16}+x^{12}+x^5+1$)的错误检测能力：可以检出所有随机奇数位错误和双位错，可以检出所有长度小于等于16位的突发错；对于长度等于17位的突发错误，检错率为99.9969%；长度大于等于18位的突发错误，检错率为99.9985%。



$G(X)=X^5+X^2+1$

可靠数据传输

(Reliable Data Transfer)

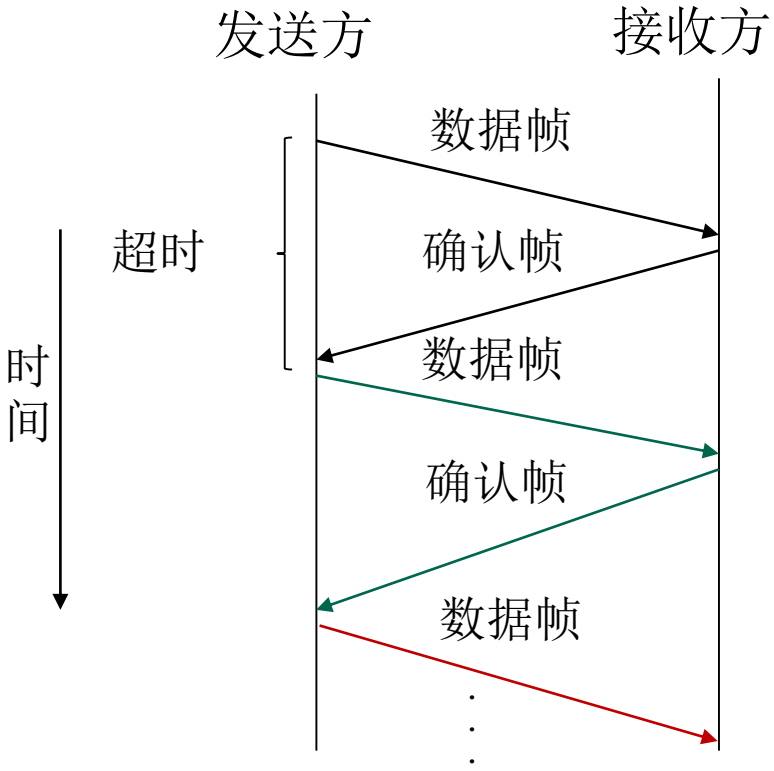


如果收到的帧出现错误又不能纠正，该帧将被丢弃，即出现丢包错误(loss)。自动重发请求(Automatic Repeat reQuest，ARQ)通过确认的方式重传丢失的包，它的方法是每发送一帧都启动一个超时定时器，如果它的确认帧在其超时时间内到达就删除该定时器，否则，将重传该帧并同时重启其定时器。

确认帧(Acknowledgement Frame)是一个控制帧，接收方把它发给发送方用来表示已经收到了它的数据帧。

后面讲的停等协议和滑动窗口协议都是ARQ协议。

停等协议（stop-and-wait）



只有收到前一个数据帧的确认才可以发送下一个数据帧。

例子：1000Mbps 链路，15 ms传播延迟，1KB的帧，停等协议，该链路的吞吐量是多少？

帧长L=1KB

数据传输率R=1000Mbps

往返时间(RTT)=2*15ms=30ms

发送延迟(L/R)=1KB/1Gbps = 0.008ms

吞吐量(throughput)=L/(RTT+L/R)

=1KB/30.008ms

=0.266Mbps

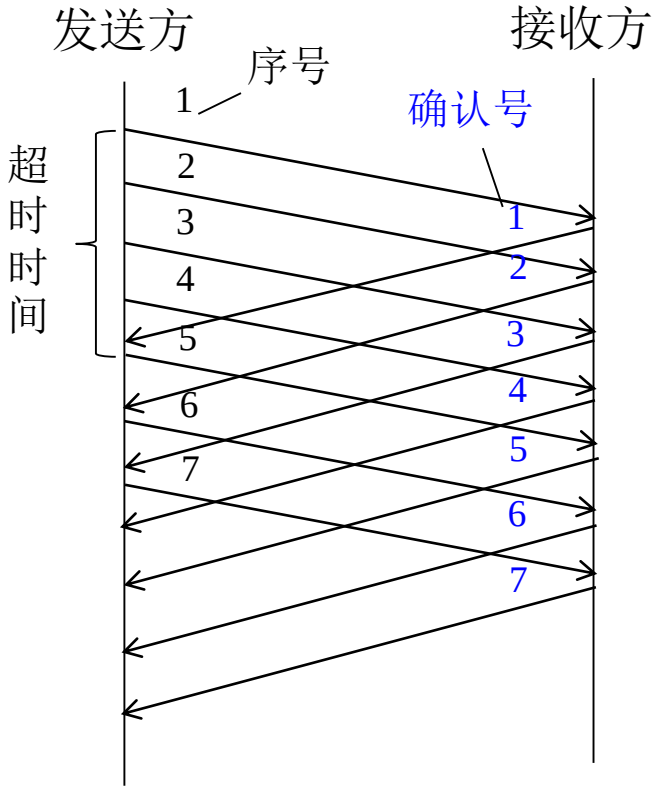
错误：数据帧丢失，确认帧丢失，超时

序号问题：至少需要几个序号？

滑动窗口协议

(Sliding Window)

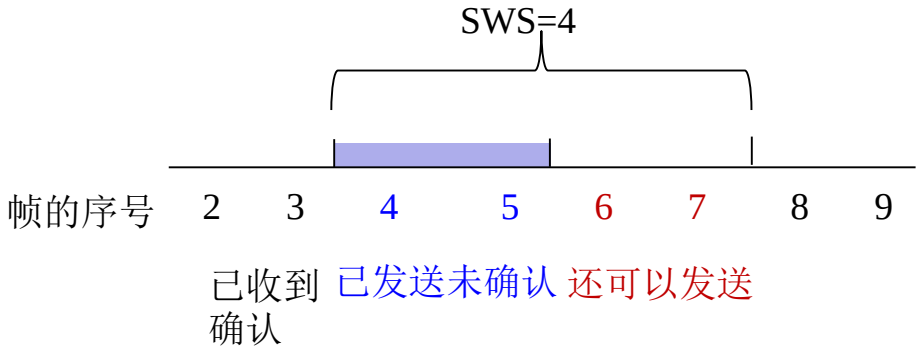
滑动窗口协议是一种ARQ协议，它不需要等待已发送的数据帧的确认帧回来就可以连续发送多个帧，其个数由发送窗口来控制。



发送窗口(Sending Window)是个连续发送数据帧的可用序号范围，主要用于**流控制**(flow control)。

发送窗口大小(Sending Window Size, SWS)表示发送窗口的大小，也是发送缓冲区的大小。

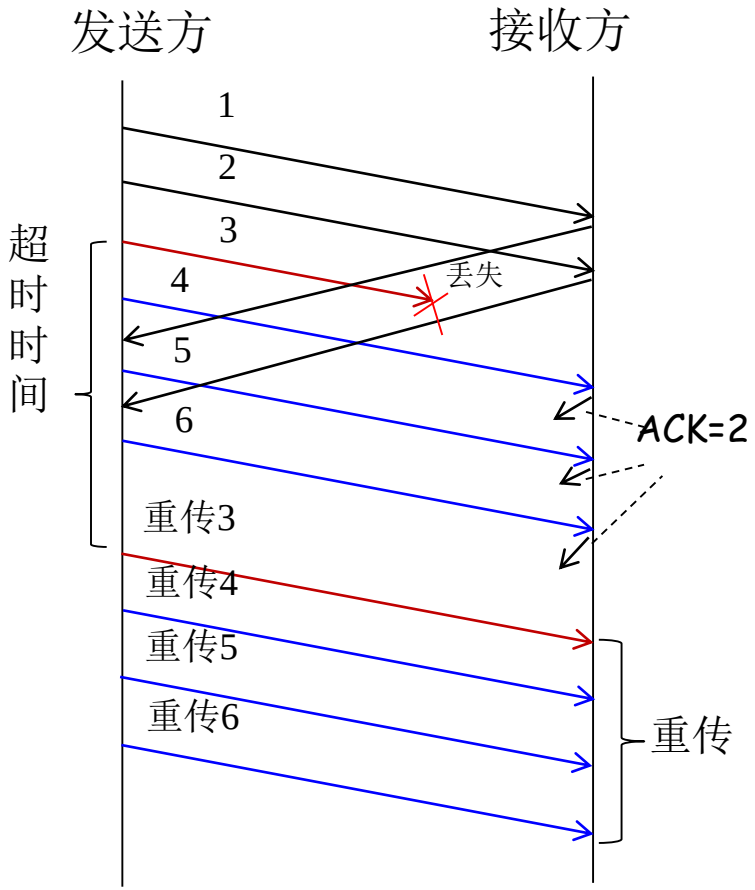
如果接收方发送的**确认帧**(Acknowledgement Frame)的确认号为ACK，表示序号为ACK以及之前的数据帧已经全部收到并已交给上层协议。



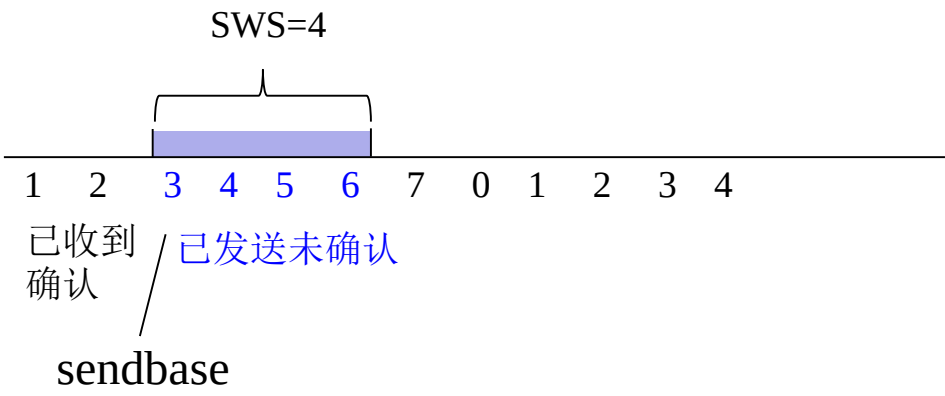
回退N协议

(go back N)

回退N协议是一种滑动窗口协议，出现超时的时候将重传所有已发送且未收到确认的帧。



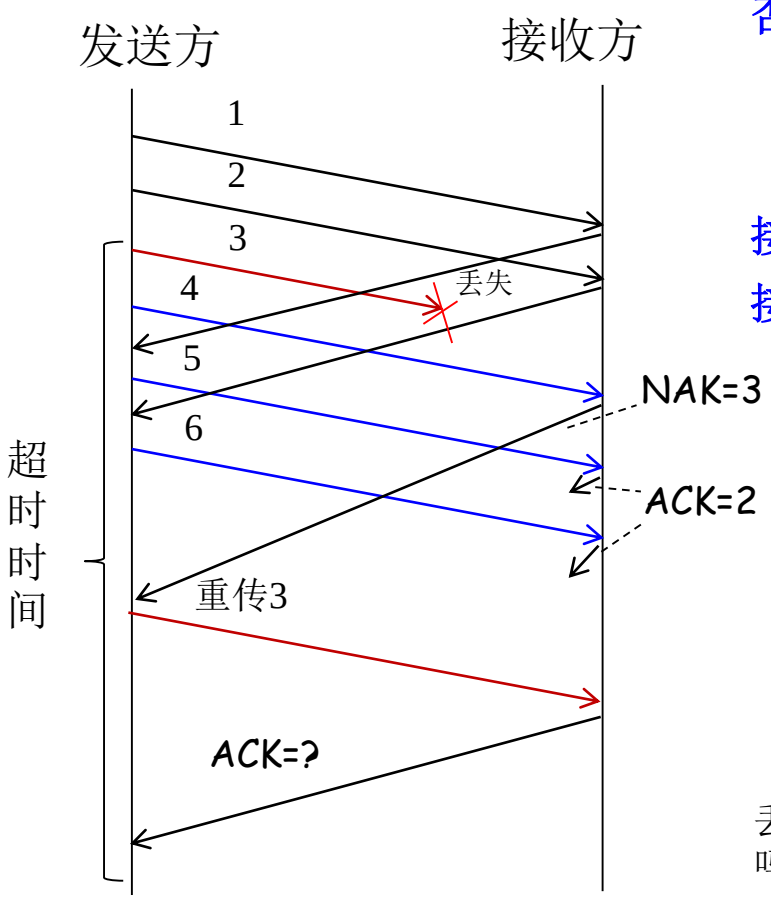
序号可以循环使用：



选择性重传

(Selective Repeat)

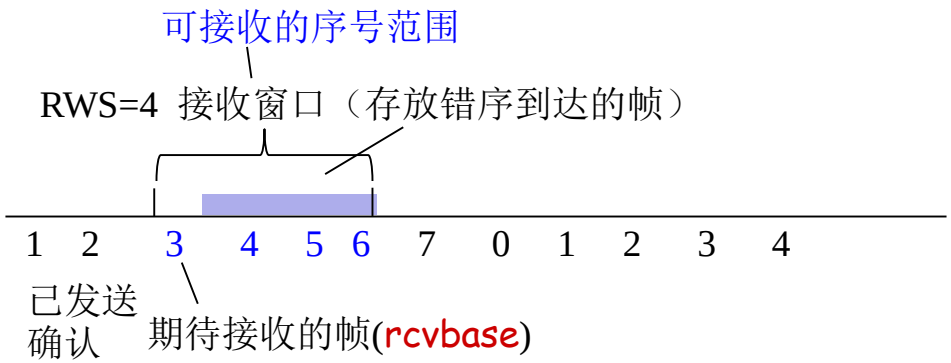
选择性重传协议通过发送NAK帧要求发送方单独重传丢失的帧。



否定性确认帧(Negative Acknowledgement,NAK)用于表示这一帧之前的数据帧全部收到并已交给上层协议，要求发送方重传这一帧。每个帧只发送一次NAK。

接收窗口用于确定应该保存哪些帧，用序号范围表示。

接收窗口大小(Receiving Window Size, RWS)表示接收窗口的大小，也是接收缓冲区的大小。



丢失NAK是否会出错？丢失多个帧怎么办？没有后续帧会出错吗？ 如何设置超时时间?

Selective repeat

sender

data from above:

- if next available seq # in window, send pkt

timeout(n):

- resend pkt n, restart timer

ACK(n) in [sendbase,sendbase+N-1]:

- mark pkt n as received
- if n smallest unACKed pkt, advance window base to next unACKed seq #

receiver

pkt n in [rcvbase, rcvbase+N-1]

buffer

- if n=rcvbase, (1) deliver buffered, in-order pkts to upper layer
  (2) advance window to the pkt x expected to receive,
  (3) rcvbase set to x
  (5) send ACK(rcvbase -1)
- if n!=rcvbase,
  send NAK(rcvbase-1)
  if not sent within some time
  or send ACK(rcvbase -1)

pkt n not in [rcvbase,rcvbase-1]

- discard the pkt
- send ACK(rcvbase-1)

提高滑动窗口协议的效率

选择性确认(Selective Acknowledgement)

接收方把已收到的帧的序号告诉发送方。当发送方要重传帧的时候，不会发送这些帧。

延迟确认(Delayed Acknowledgement)

接收方收到一帧后并不立即发送确认帧，而是等待一段时间再发送。

捎带确认(Piggybacking)

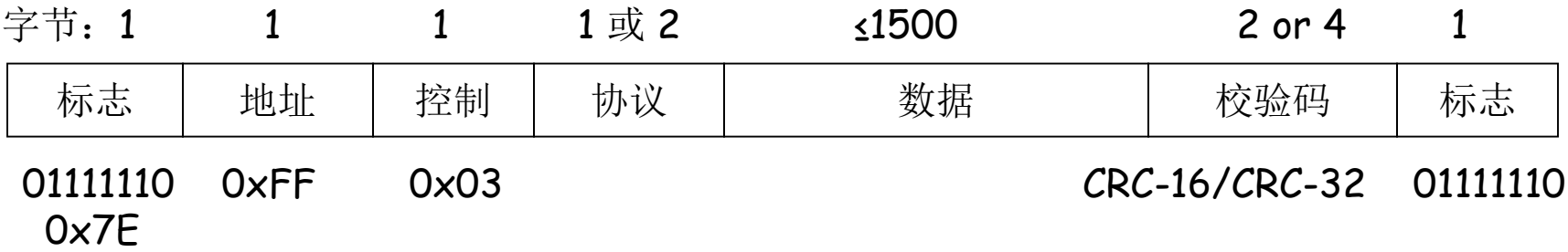
通信双方其实是全双工方式工作。接收方在发数据给对方时顺便把确认号也告诉对方。

PPP 协议

RFC 1557

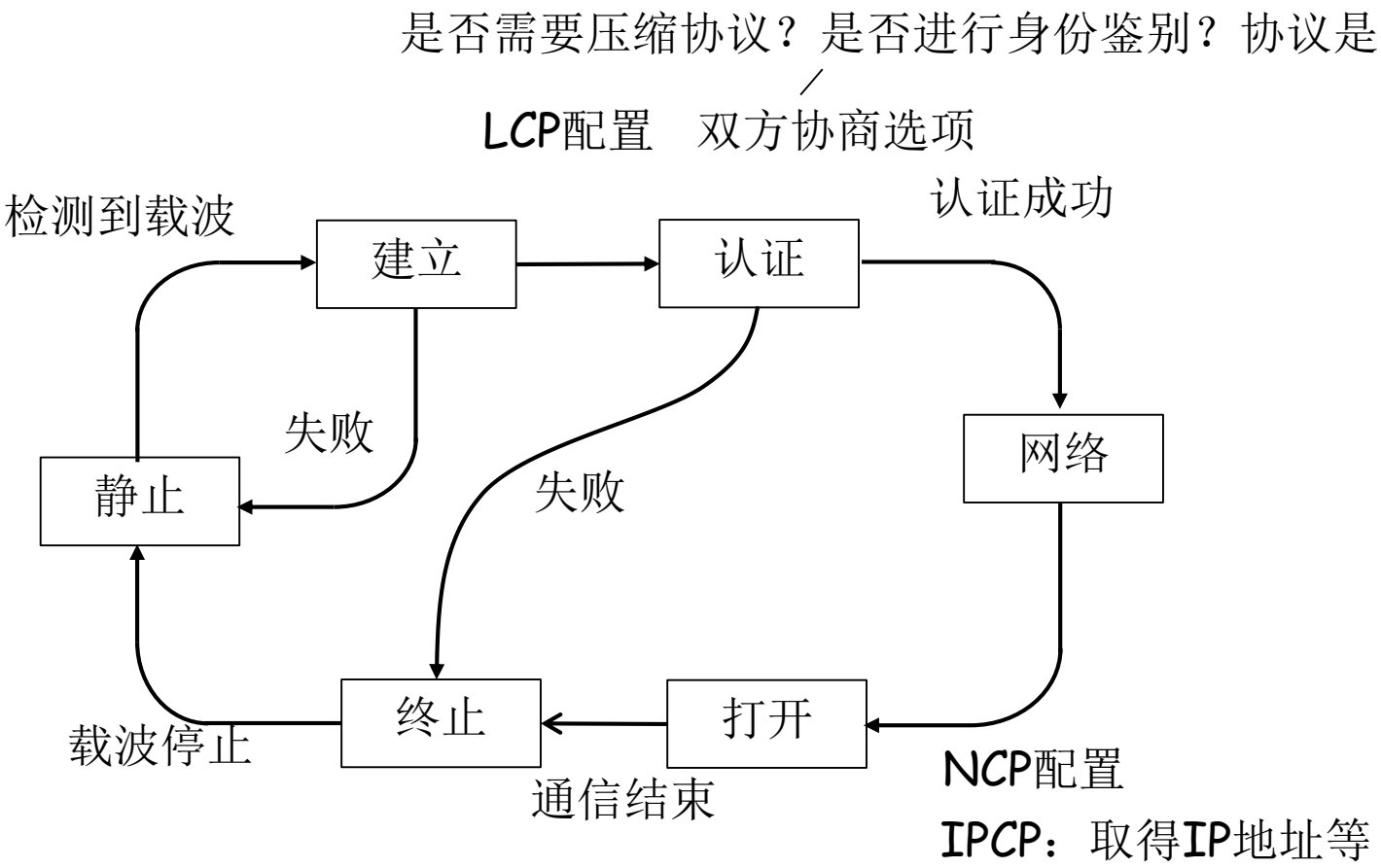
- PPP协议(Point-to-Point Protocol) 是点到点网络的数据链路层协议。
- PPP协议是根据HDLC(High-Level Data Link Control)协议进行设计的，主要用于串行电缆(V.35) 、电话线(MODEM) 等各种串行链路。
- PPP协议可以提供连接认证、传输加密和压缩功能。
- PPP协议可以为各种网络层协议提供服务：因特网的IP协议、Novell公司的IPX协议、苹果公司的AppleTalk协议。
- PPP协议的多链路捆绑技术可以通过将通信两端之间的多条通信链路捆绑成一条虚拟的链路而达到扩充链路可用带宽的目的。
- ADSL的PPPoE协议和VPN中的PPTP协议都采用了PPP协议进行封装。

PPP数据帧



- PPP协议采用HDLC协议的广播地址(0xFF)和无编号帧(0x03)。
- PPP协议采用字节填充法(Byte-stuffing): 信息字段出现的标志字节(0x7E)用0x7D-5E 替换，0x7D用0x7D-5D替换。所有小于0x20的字节(控制字节)都加上值0x20，并在前面加上转义字符0x7D，例如，0x01用0x7D-21替换。
- PPP协议的16位CRC校验码的除数为$x^{16} + x^{12} + x^5 + 1$。
- 协议字段(Protocol)指明上层协议，例如, 0x802B-IPX，0x0021-IP，0xC021-LCP，0x8021-IPCP等 。
- 可以省略地址和控制字节(头部压缩)。PPP协议还可以进行TCP压缩和数据压缩。
- PPP协议没有纠错功能，也没有流控制和确保有序的功能。

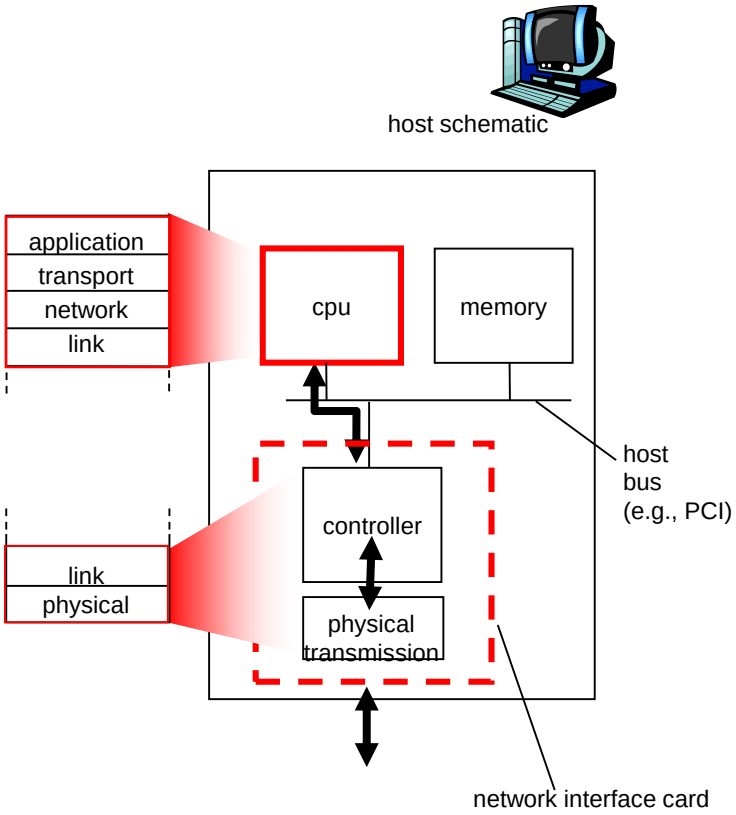
PPP协议的状态图



Link Control Protocol(LCP)
Network Control Protocol (NCP): IPCP(Internet)，IPXCP(Novell)

链路层的实现

- 链路层主要在网络接口卡(*network interface card*，NIC)及其驱动程序 上实现。路由器是在接口模块上实现。



总结

- 概述：数据链路层的功能是什么？
- 差错检测：奇偶校验、校验和、CRC校验码
- 可靠数据传输
- 停等协议
- 滑动窗口协议
- PPP协议