

计算机网络

开始连接

华中科技大学电信学院 2016



学习目标

- 理解直连网络的概念，区分节点、网络适配器、链路的特征；
- 了解网络通信编码，包括NRZ、NRZI、Manchester、4B/5B等；
- 理解不同的组帧方法，包括面向字节、面向比特、面向时钟的组帧协议案例；
- 掌握**差错控制**的概念，包括二维奇偶校验、循环冗余校验等，具备简单运算的能力；
- 掌握**可靠传输**的概念和基本实现机制，掌握停止等待、滑动窗口的ARQ算法，具备简单运算的能力；
- 结合以太网的媒介特征，理解以太网的设计要点，掌握**CSMA/CD冲突检测**的原理，具备简单计算的能力；
- 结合无线媒介的特征，理解无线局域网的设计要点，理解**CSMA/CA冲突避免**的原理；

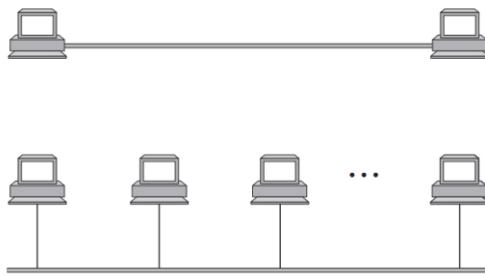


提纲

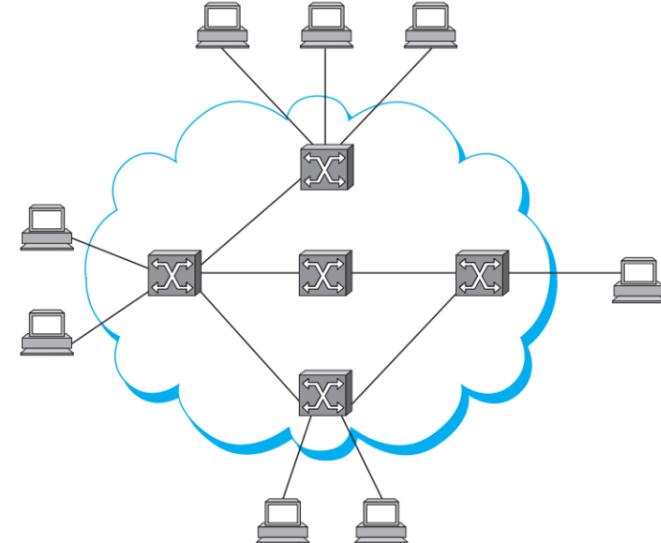
引言

- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

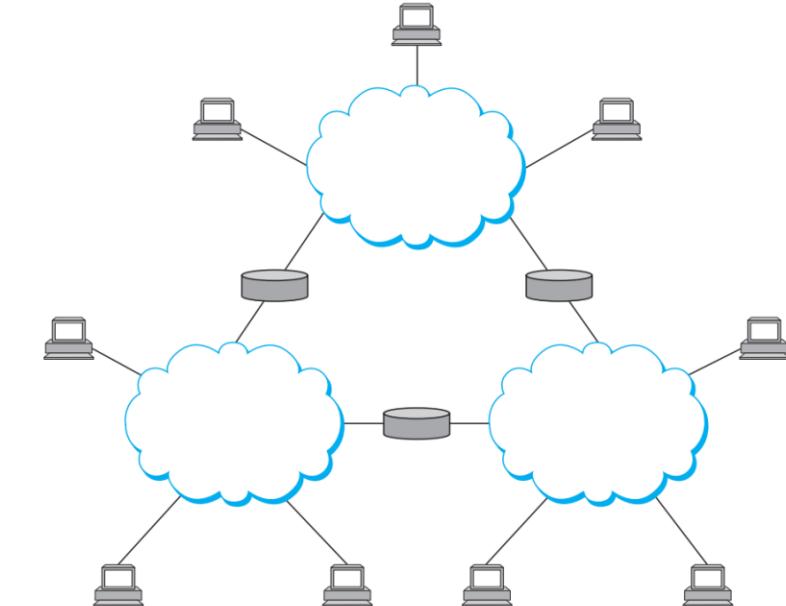
回顾: 网络连通性



直连网络



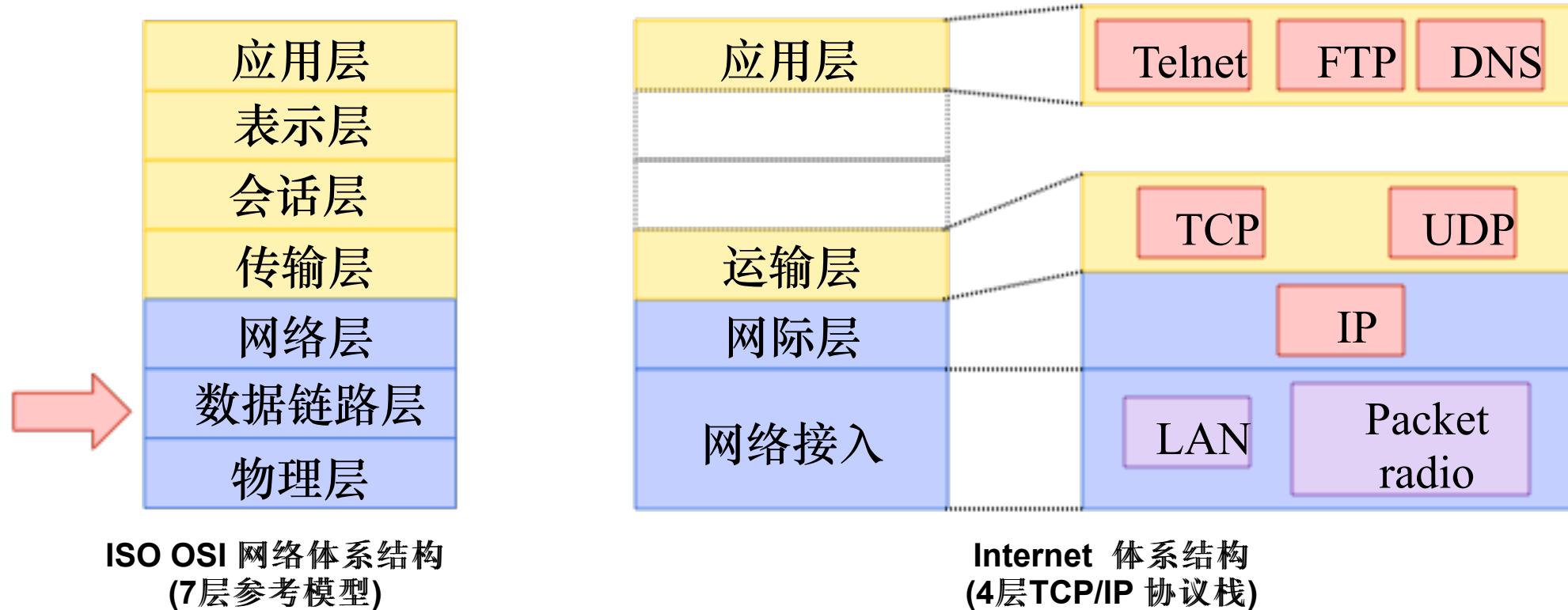
交换网络



网络互联

- 什么是直连网络?
- 所有的主机通过某种物理媒质直接连接

回顾: 网络体系结构



- 直连网络问题在体系结构中所处的层次?
- L1(物理层) 处理物理传输媒质上的数字通信问题, 特别是比特如何表示为模拟信号
- 直连网络的研究主要集中于L2(数据链路层)

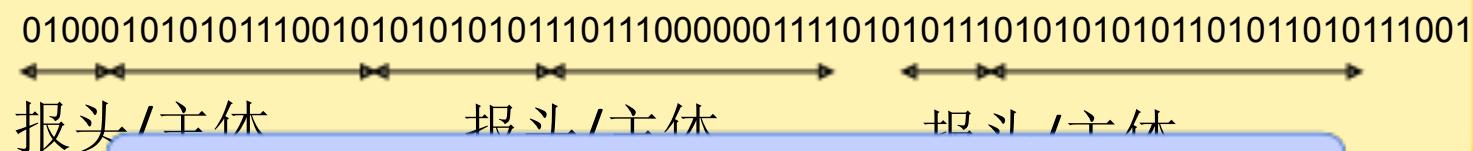
物理层 vs 数据链路层

数据链路
层

传输数据帧



数据帧



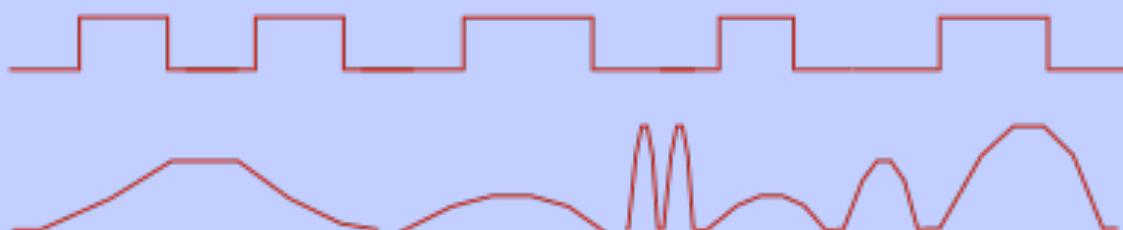
物理
层

比特流

物理层为上层提供比特流传输服务

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

数字信号



模拟信号

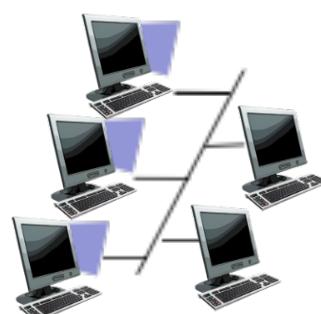


直连网络

- 什么是直连网络?
- 所有的主机通过某种物理媒质直接连接
 - 物理传输媒质: 电缆, 光纤, 空气接口, …
 - 与距离无关: 小的区域(例如, 一栋办公大楼) 或者一个大的区域(例如, 横贯大陆)
- 传播技术
 - 点到点的链路
 - 广播链路
- 本章的研究动机
 - 直连网络是最简单的网络形式
 - 我们从简单的用合适的传输媒质将两台或多台主机连接起来着手研究网络

直连链路类型

- point-to-point
- PPP for dial-up access
- point-to-point link between Ethernet switch, host
- *broadcast (shared wire or medium)*
- old-fashioned Ethernet
- upstream HFC
- 802.11 wireless LAN



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



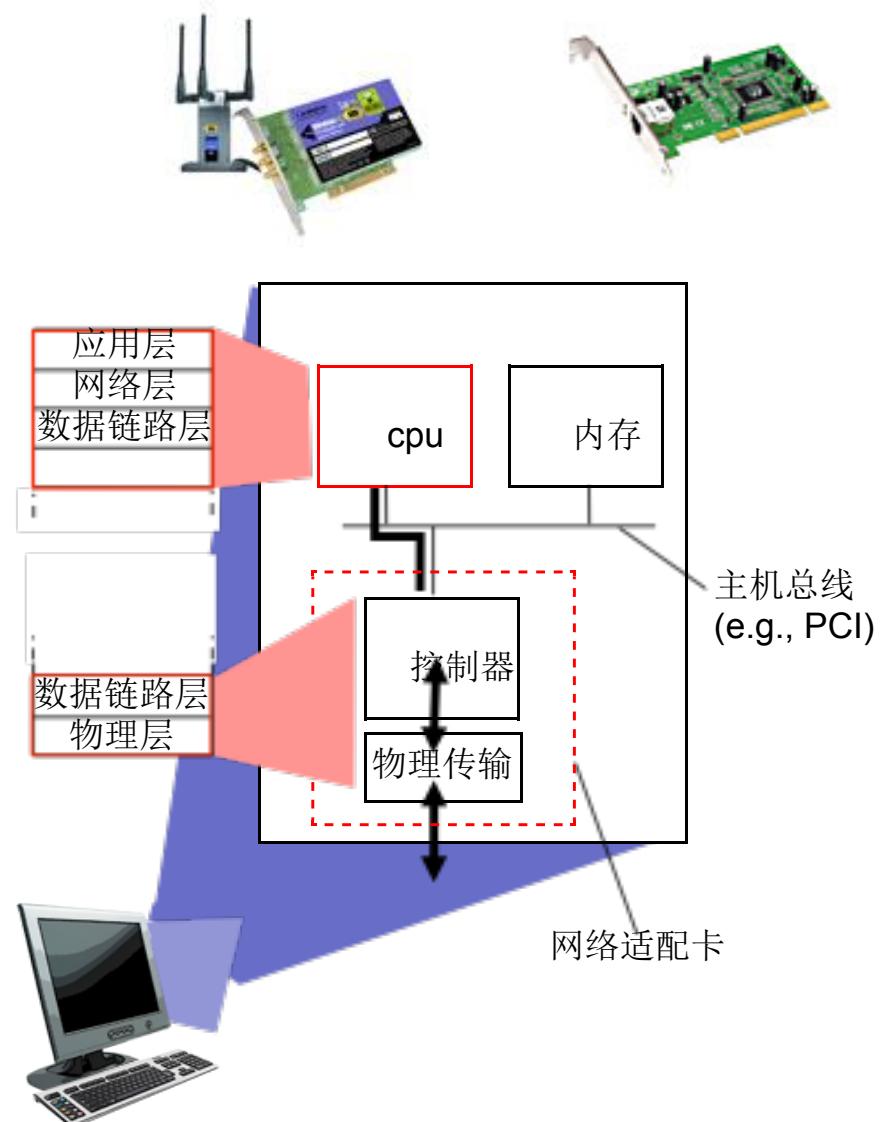
shared
RF
(satellite)



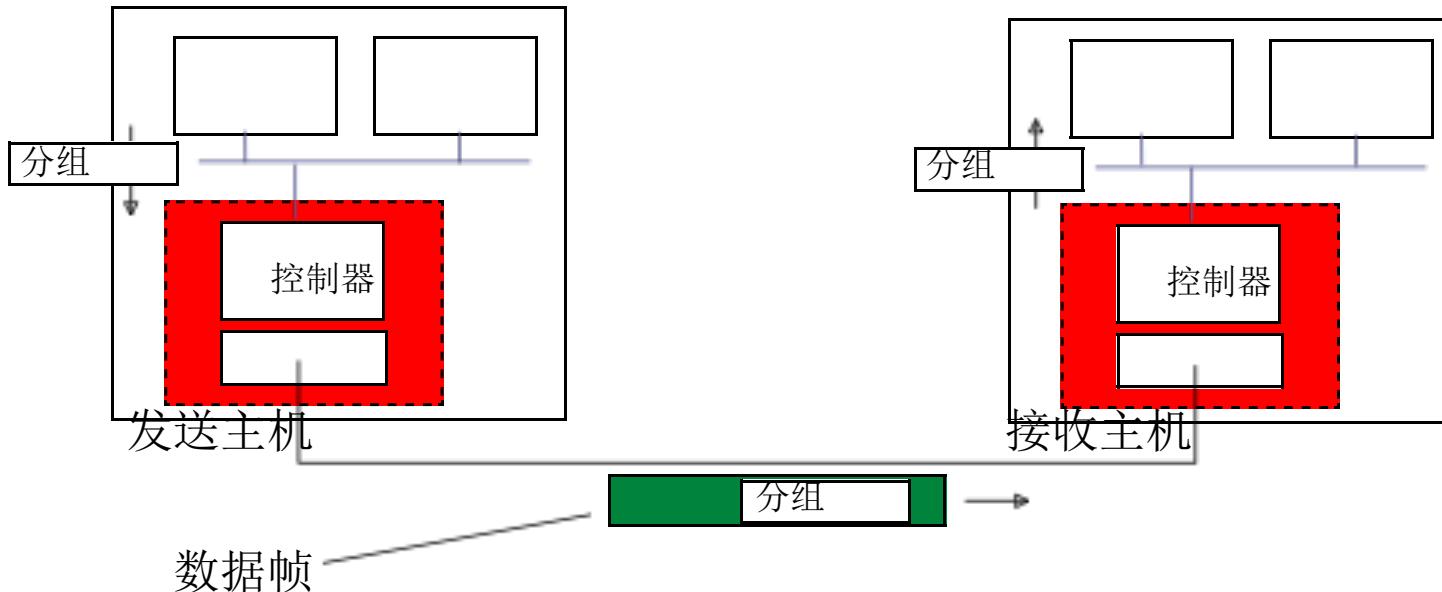
humans at a
cocktail party
(shared air, acoustical)

数据链路层在那里实现?

- 每台主机
- 数据链路层以适配器的形式实现 (aka 网络接口卡NIC) 或者集成在芯片上
 - 以太网卡, 802.11无线网卡; 以太网芯片
- 通常数据链路层和物理层联合实现
- 挂载在主机的系统总线上
- 包含硬件、软件和固件



适配器之间的通信



- **发送端**
- 将分组封装为数据帧
- 增加差错检测、可靠传输、流控等功能
- **接收端**
- 完成差错检测、实施可靠传输、流控等
- 提取分组并交付之上层协议



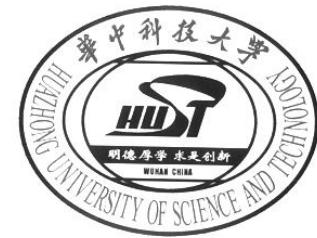
核心问题：

如何连接到网络？

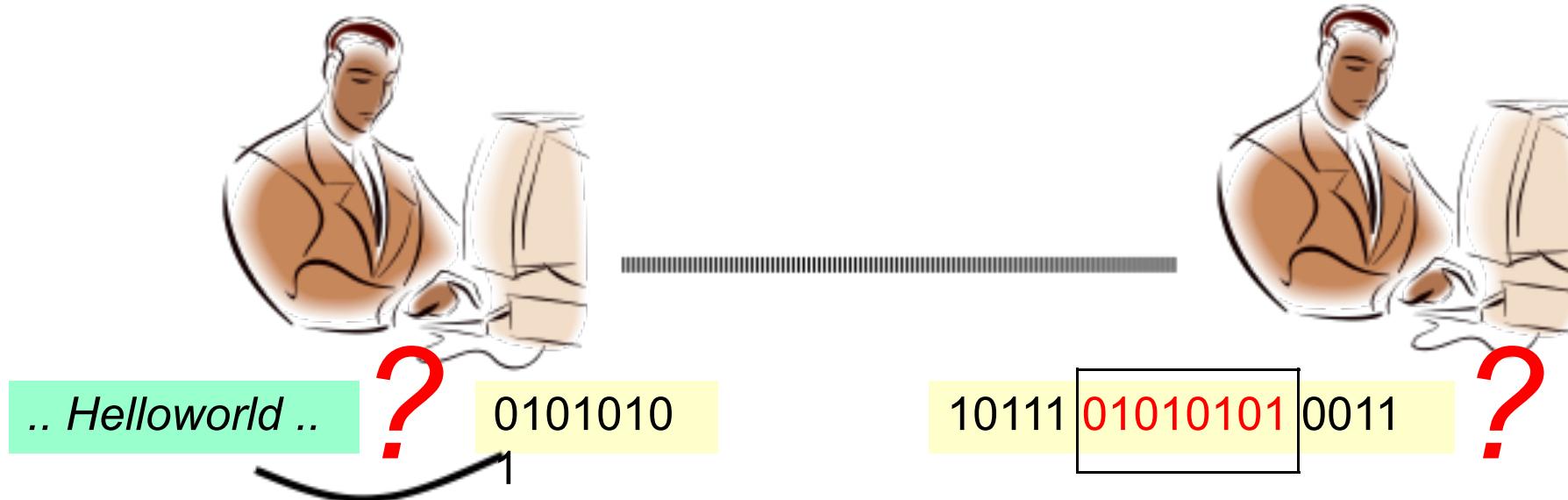


提纲

- 引言
核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

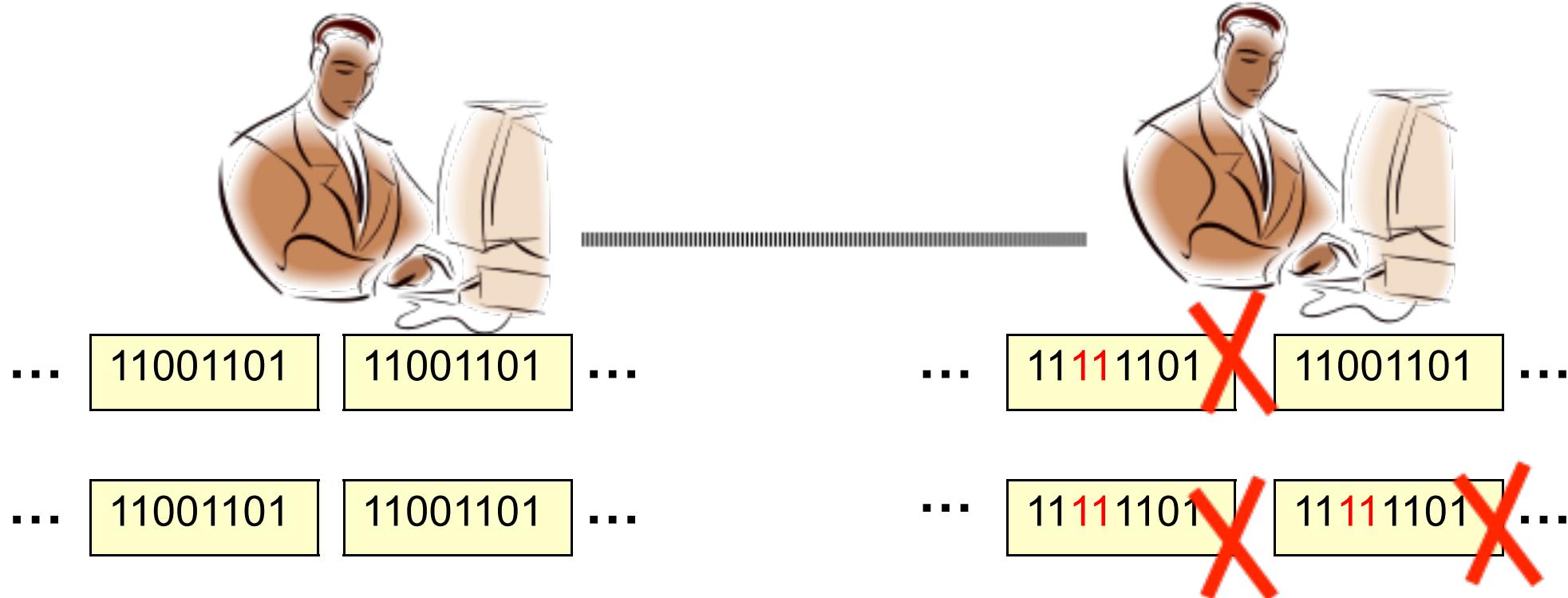


直连网络的研究问题



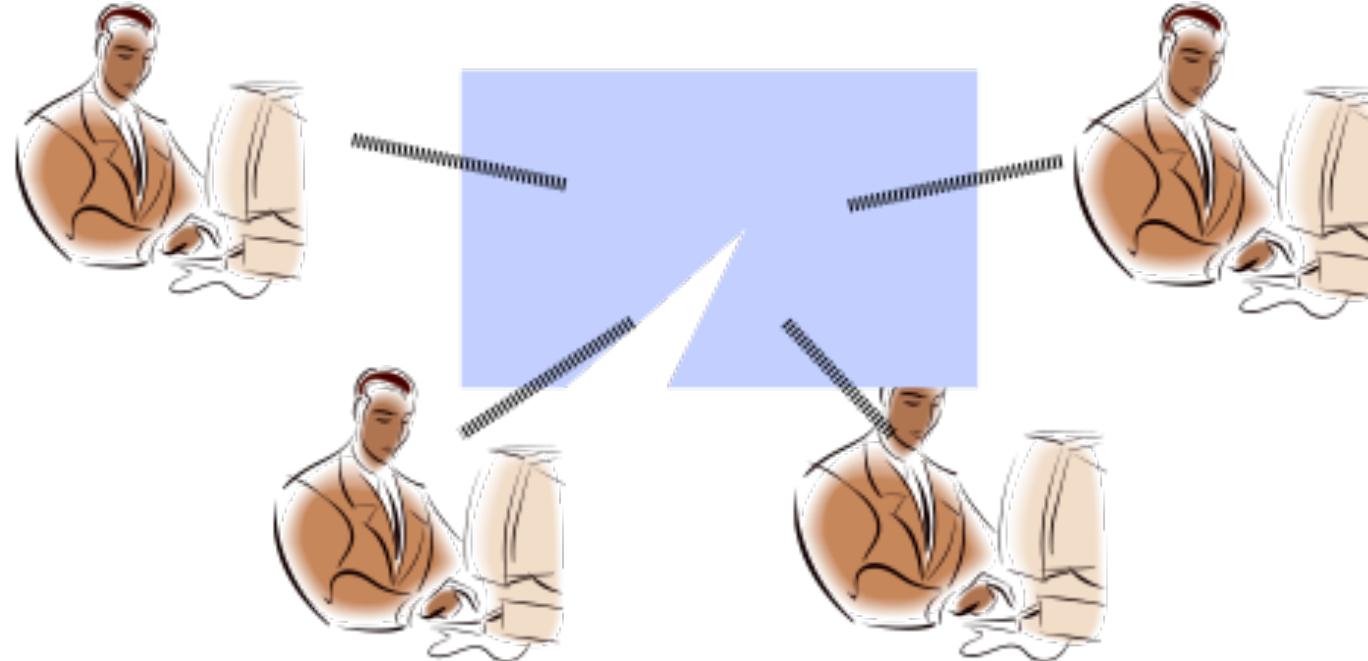
- 编码
 - 对发送到电缆或光纤上的比特进行编码，使其能被接收主机所理解
- 帧定界
 - 把物理链路上传输的比特序列描述为完整的消息，以便传送到端节点

直连网络的研究问题



- 差错检测
 - 检测帧传送过程中可能出现的错误，并采取相应的动作
- 可靠传输
 - 保证链路在帧不时可能出现错误情况下的可靠性

直连网络的研究问题



- 通信链路共享（介质访问控制）
 - 如果链路静态共享，很容易处理
 - 如果链路动态共享，如何控制多个主机的访问顺序？



提纲

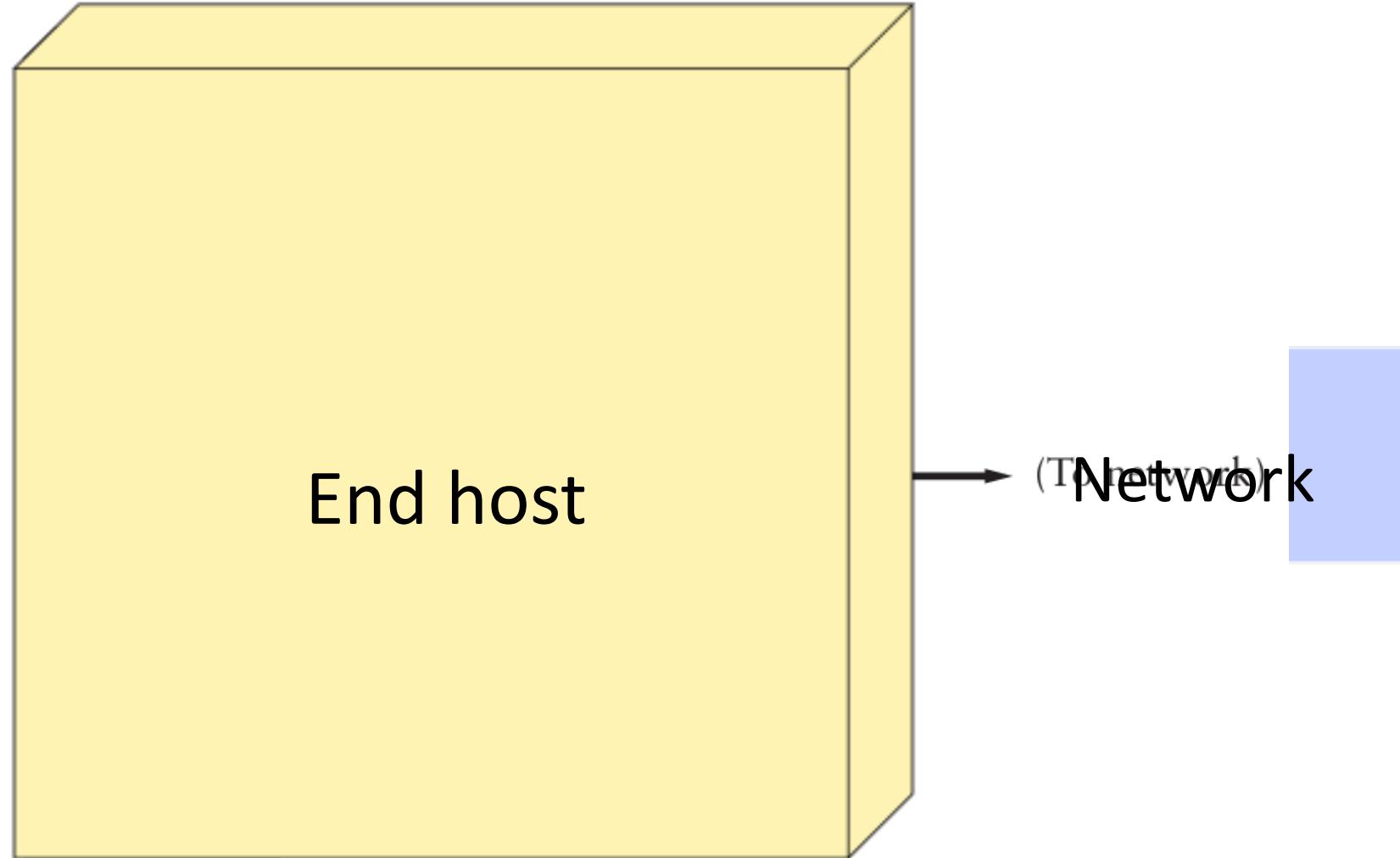
- 引言
- 核心问题: 连接到网络
- 网络硬件
 - 节点
 - 链路
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

点到点链路的
四个基本问题

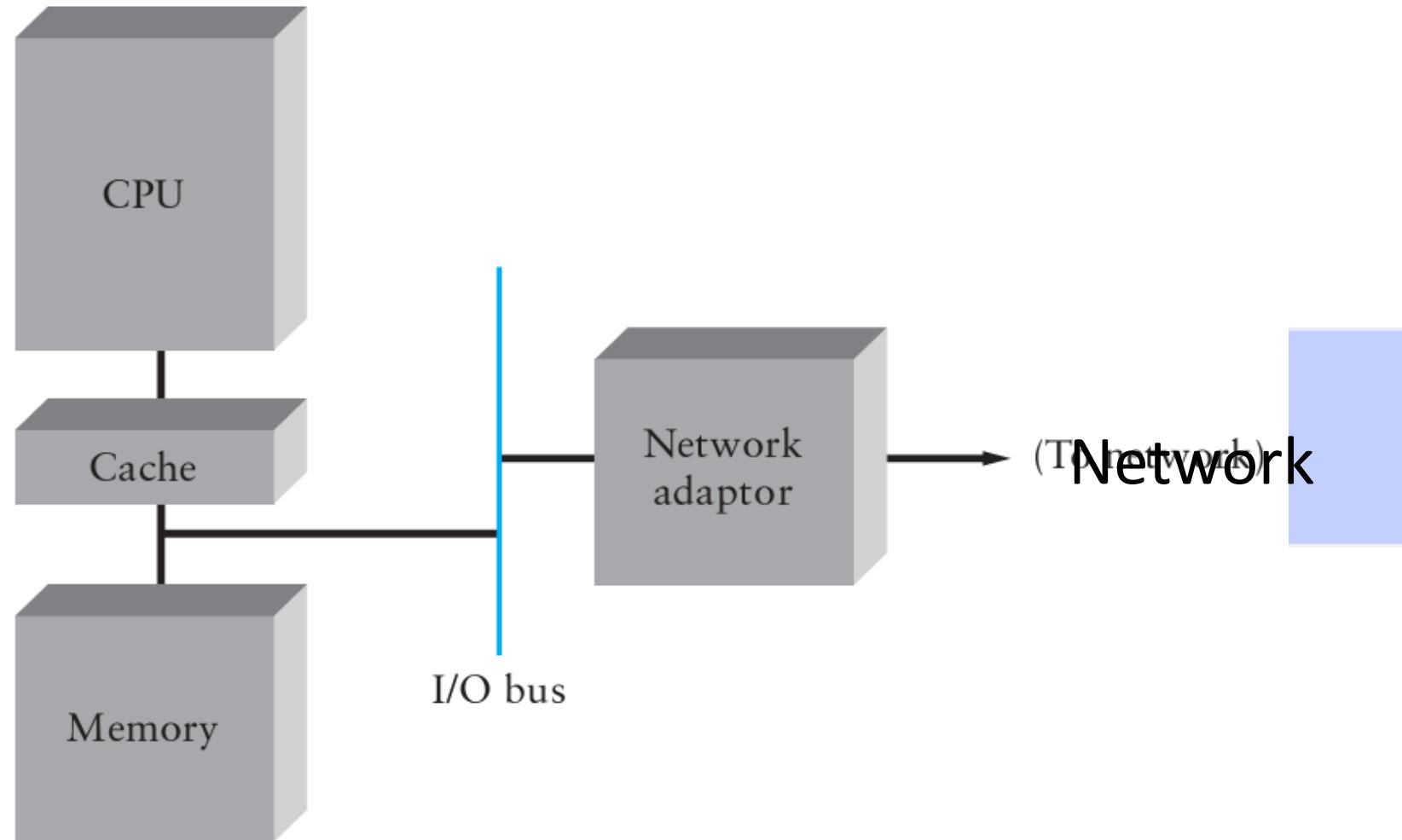
共享链路
的问题



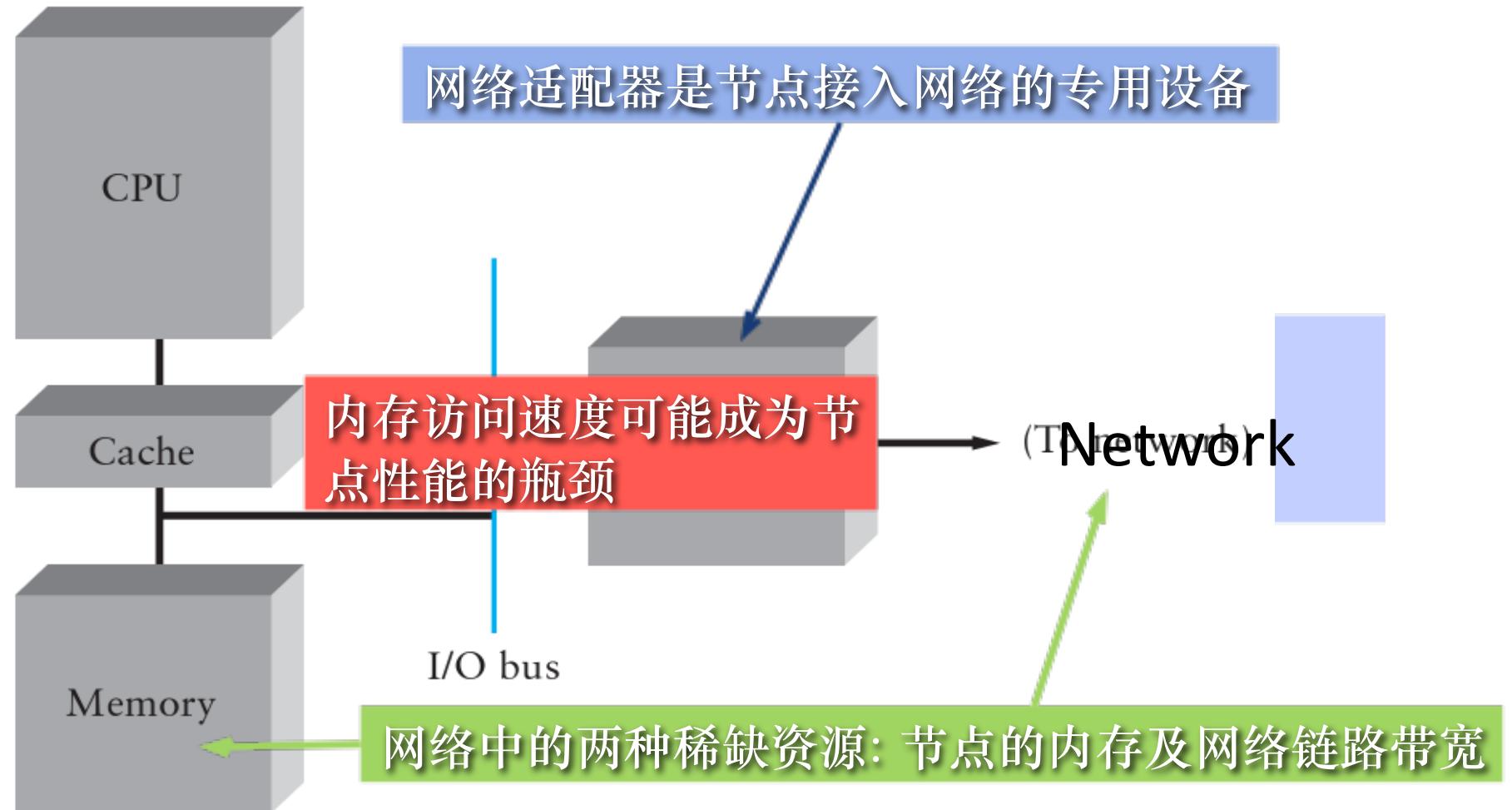
网络节点



网络节点



网络节点



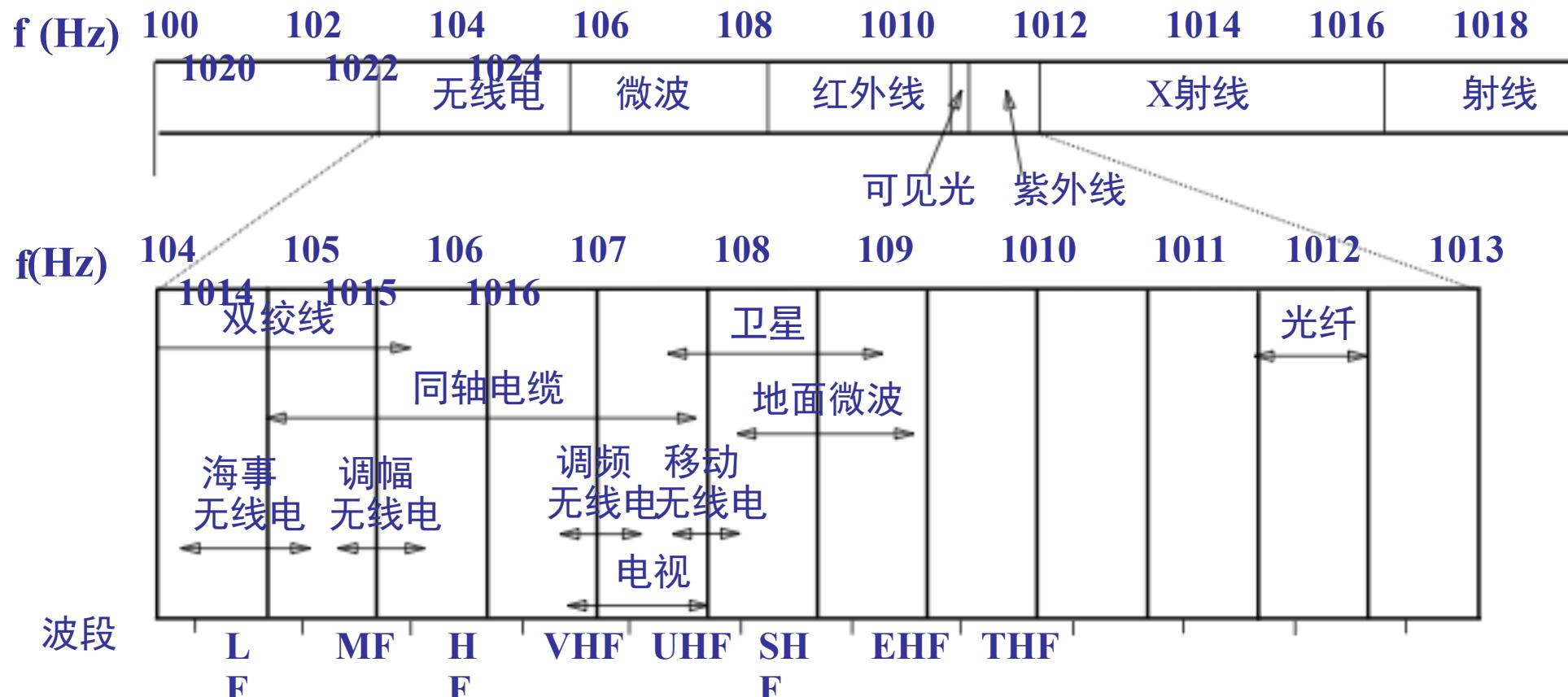


网络适配器

- 功能
 - 进行串行/并行转换
 - 对数据进行缓存
 - 设备驱动程序（数据链路层协议）
- 接口特性
 - 机械特性:接口所用接线器的形状和尺寸、引线数目和排列、固定和锁定装置等等.
 - 电气特性:在接口电缆的各条线上出现的电压的范围.
 - 功能特性:某条线上出现的某一电平的电压表示何种意义.
 - 规程特性:对于不同功能的各种可能事件的出现顺序.

链路

- 物理传输媒质
- 导向型媒质: 信号在固态媒质上传播, 例如同轴电缆, 光纤, 双绞线
- 非导向型媒质: 信号自由传播, 例如电磁波



物理媒质：有线

双绞线(TP)

- 两根绝缘的铜线
- 3类: 传统电话线, 10 Mbps 以太网
- 5类: 100Mbps 以太网



同轴电缆

- 全双工基带:
单信道
传统以太网
- 宽带:
多信道
混合光纤同轴网(HFC)



光纤

- 采用玻璃纤维传递光脉冲, 一个脉冲代表一个比特
- 高速
- 低误码率
- 远距离中继转发





有线链路容量

Cable	Typical Bandwidths	Distances
Category 5 twisted pair	10–100 Mbps	100 m
Thin-net coax	10–100 Mbps	200 m
Thick-net coax	10–100 Mbps	500 m
Multimode fiber	100 Mbps	2 km
Single-mode fiber	100–2400 Mbps	40 km



商业链路

- 租用线路
- 通常指通过电话网络的一个逻辑连接
- 价格昂贵
- 最后一公里链路
 - 传统电话服务 (POTS) 和综合业务数字网 (ISDN) : 网络接入网络
 - 数字用户线 (xDSL) 和电缆调制解调器: 用户接入网络

Service	Bandwidth
DS1	1.544 Mbps
DS3	44.736 Mbps
STS-1	51.840 Mbps
STS-3	155.250 Mbps
STS-12	622.080 Mbps
STS-48	2.488320 Gbps
STS-192	9.953280 Gbps

Service	Bandwidth
POTS	28.8–56 Kbps
ISDN	64–128 Kbps
xDSL	16 Kbps–55.2 Mbps
CATV	20–40 Mbps



物理媒质：无线

- 通过电磁波携带信号
 - 不存在物理的“线路”
 - 双向
 - 易受环境影响：
 - 反射
 - 障碍物
 - 干扰
- 无线链路类型：
- 地面微波
 - 可达到 45 Mbps
 - 局域网(例如, Wifi)
 - 11 Mbps, 54 Mbps
 - 较大区域 (例如, 蜂窝网)
 - 3G 蜂窝网: ~ 1 Mbps
 - 卫星网
 - 45 Mbps
 - 270 ms 端到端时延



链路容量

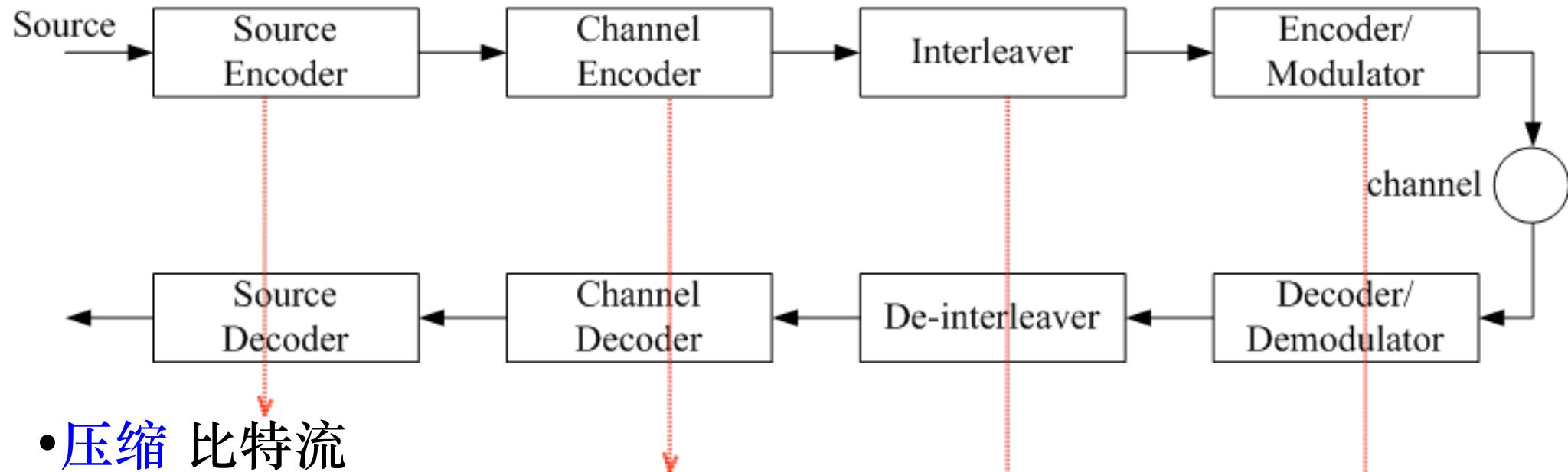
- 通信信道
- 收发机之间传送信息的传输媒质
- 用于分析物理传输媒质的特性
- 信道容量

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

其中

- C: 信道容量(bps);
- B: 信道的频带宽度(Hz);
- S/N: 信噪比(SNR).

典型的通信系统



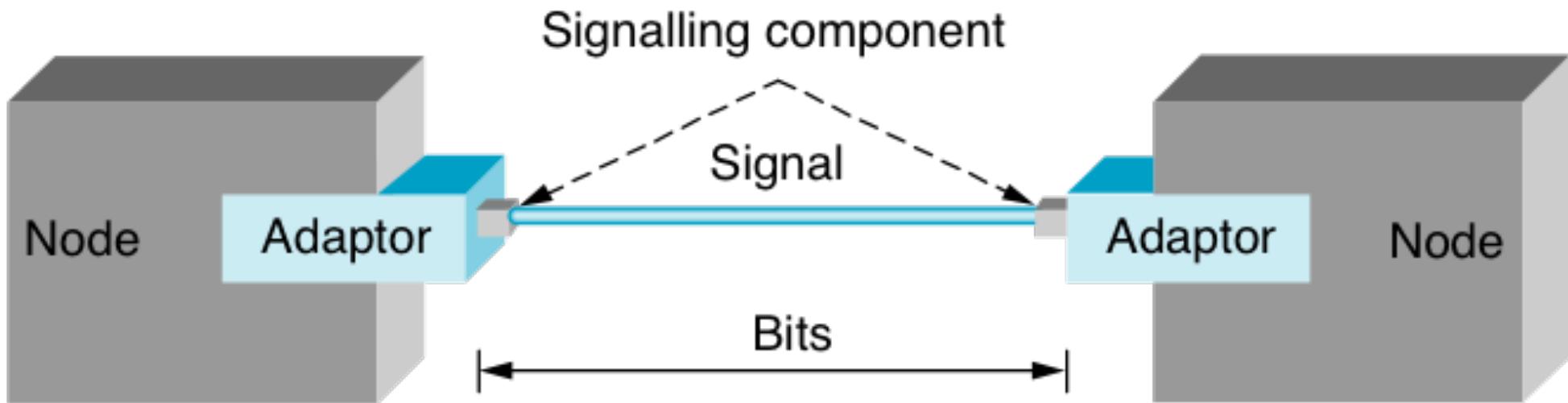
- **压缩** 比特流
- 加入**冗余**以便接收方能够检错或纠错 (比特对应于码字)
- **Scrambles** the bits of consecutive codewords to spread burst errors
- 将比特流转换为**模拟信号**进行调制并通过传输媒质传输



提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

比特和信号



- 比特 编码/解码 信号

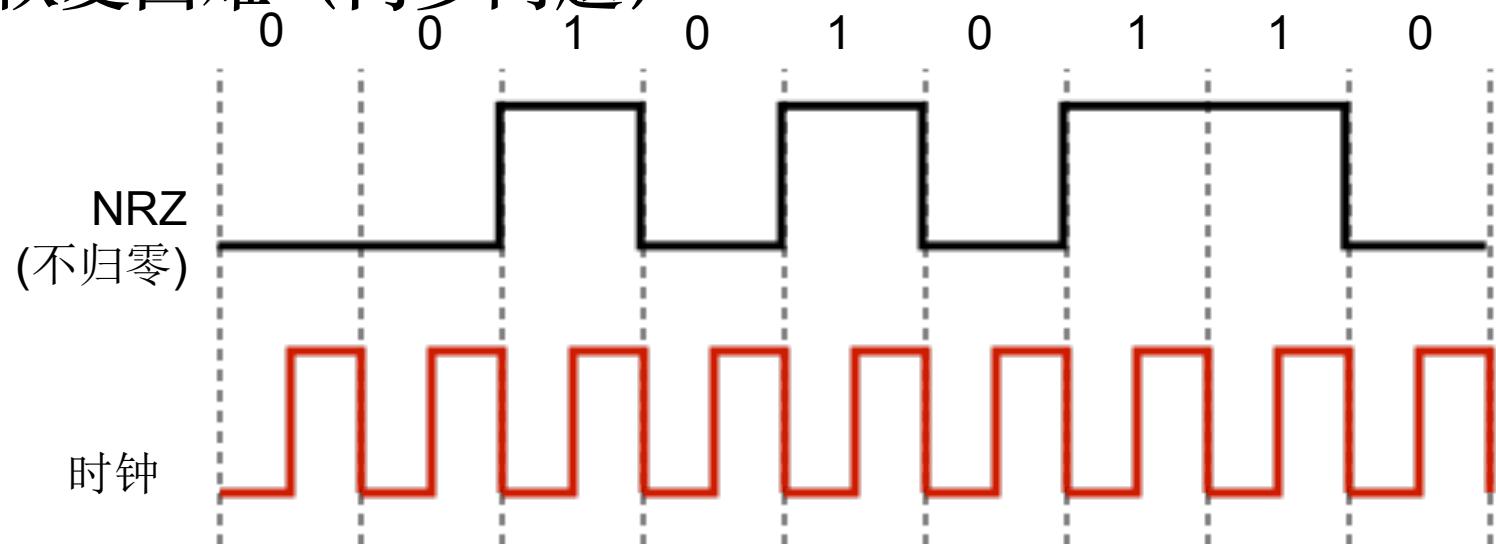


假设

- 假设处理离散信号(忽略调制细节), 高信号和低信号, 对应编码为1和0
- 收发双方同步, 即, 存在一个时钟进行信号采样
- 如果信号的幅值和持续时间足够大, 接收机可以识别出发送的信号.

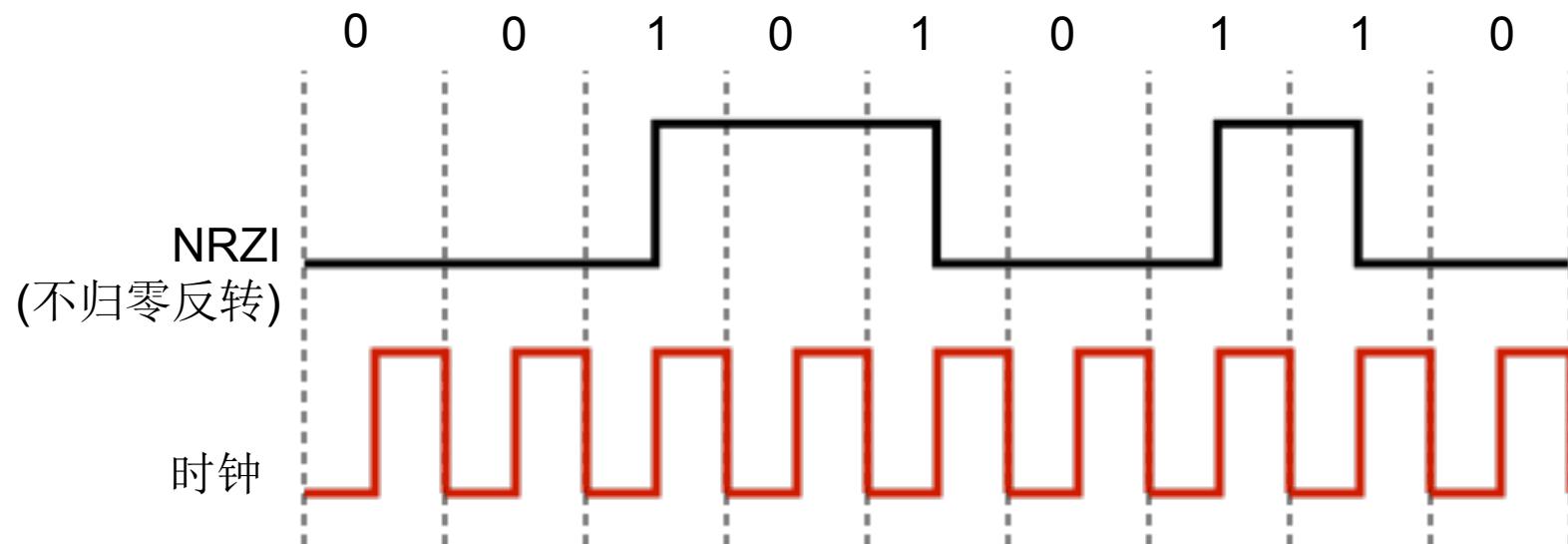
不归零 NRZ (Non-Return to Zero)

- 编码
 - $1 \rightarrow$ 高信号;
 - $0 \rightarrow$ 低信号
- 问题: 连续的1s 或 0s
 - 连续的 0s可能被误认为没有信号
 - 连续的 1s可能导致基线漂移
 - 时钟恢复困难 (同步问题)



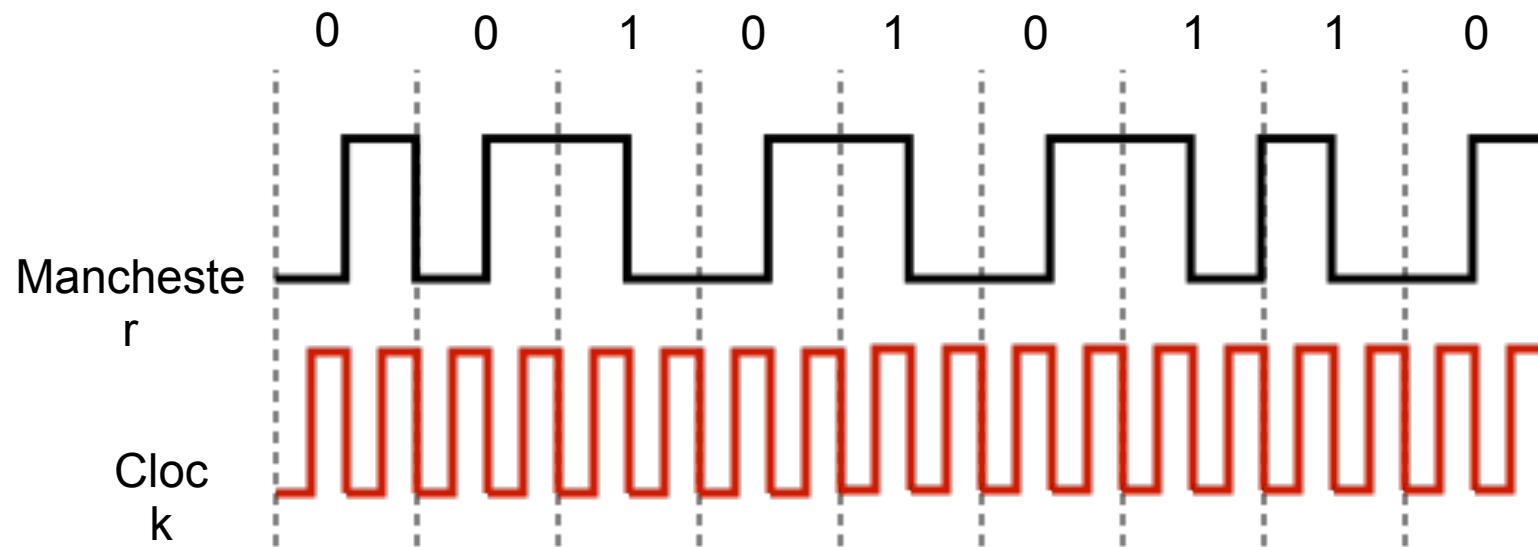
不归零反转 (NRZI)

- 编码
- $1 \rightarrow$ 信号跳变
- $0 \rightarrow$ 信号保持
- 可以解决持续1's的问题, 未能解决连续0's的问题

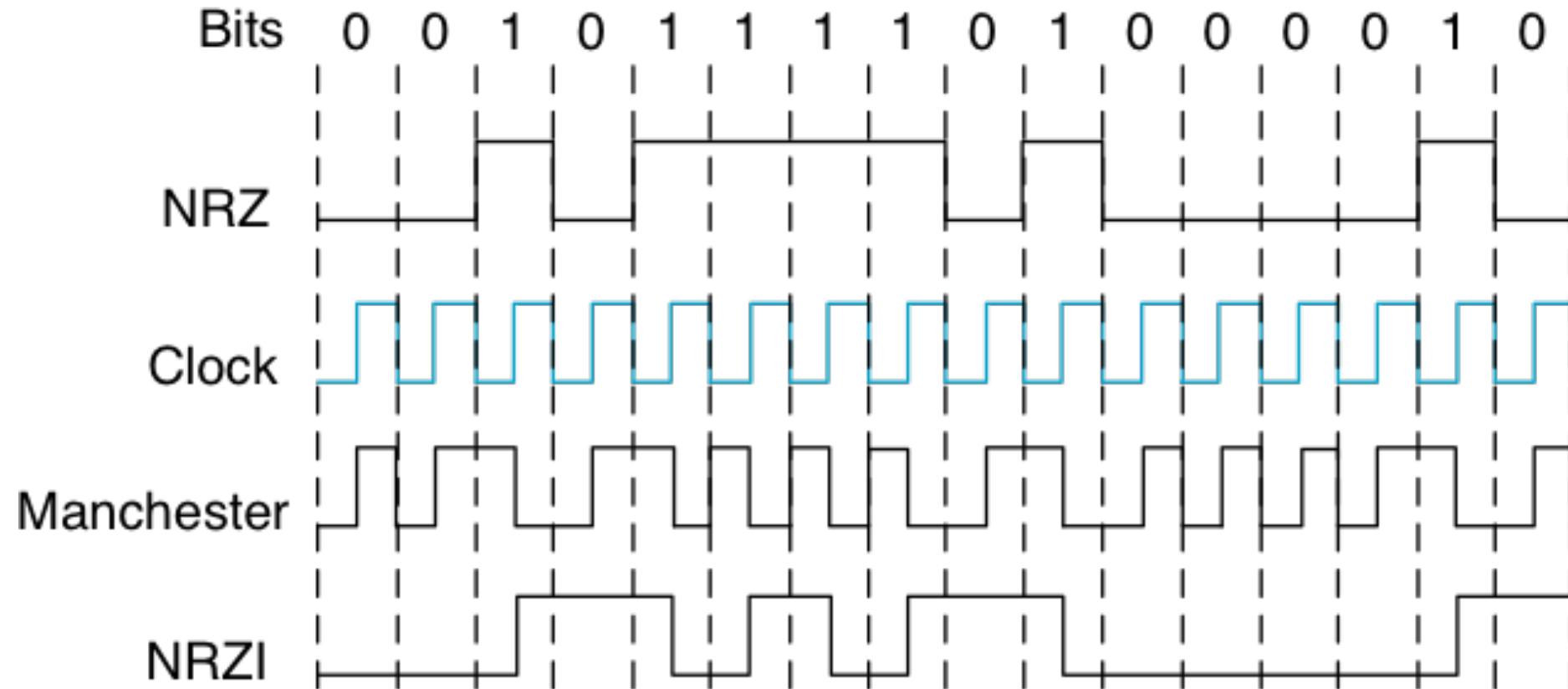


曼彻斯特编码

- 编码
 - $1 \rightarrow$ 高到低跳变
 - $0 \rightarrow$ 低到高跳变
- 有效恢复时钟
- 缺点: 信号跳变速率翻倍
 - 比特率是信号跳变速率 (波特率) 的一半
- 编码效率: 50%



图示说明





4-bit/5-bit

- 目标: 解决曼彻斯特编码的低效问题, 同时避免持续的低信号
- 解决方案:
- 用5个比特对4个比特的数据进行编码, 其中每个代码(5个比特)中最多有1个前导0, 且末端最多有2个0
- 采用NRZI对5比特的代码进行编码
- 编码效率: 80%

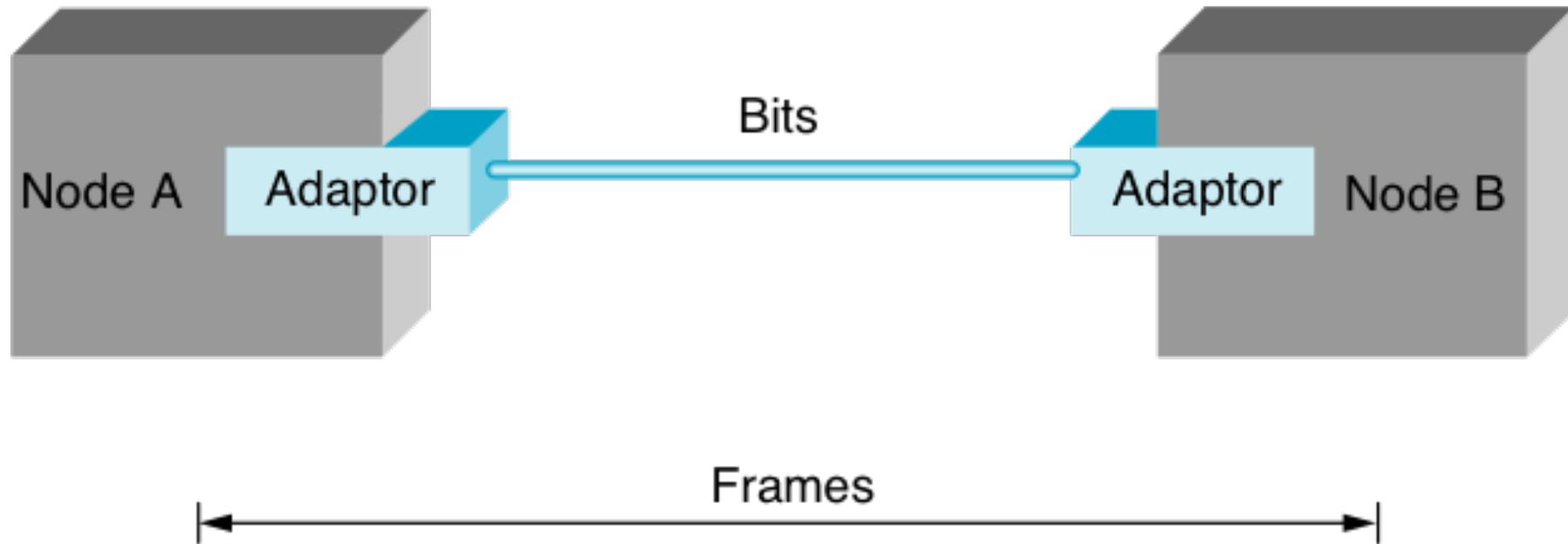
4-bit	5-bit	4-bit	5-bit
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101



提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

帧定界问题



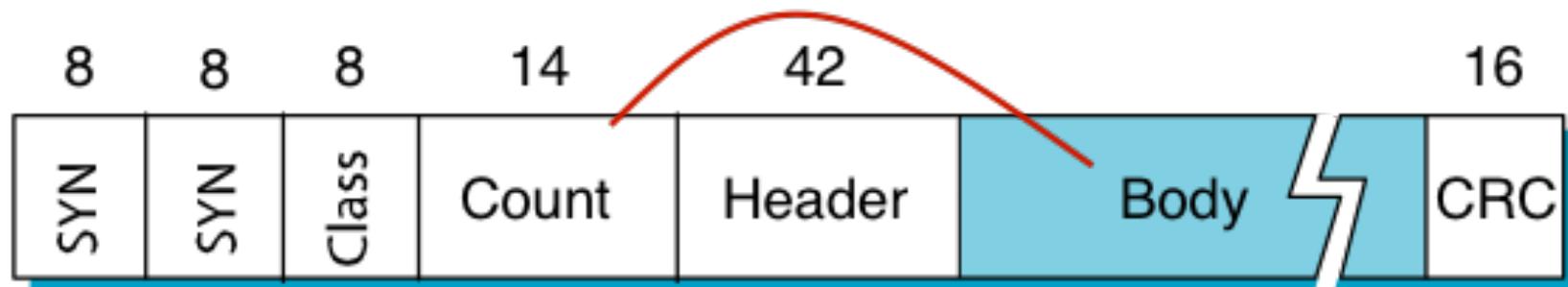
- 两个节点之间的数据传输以块为单位(帧)
- 能够识别数据帧的开始和结束



面向字节的协议

- 面向字节
- 把每一帧看做一个字节(字符)集合
- 两种方法
- 字符计数法
- 起止标记法

字节计数法



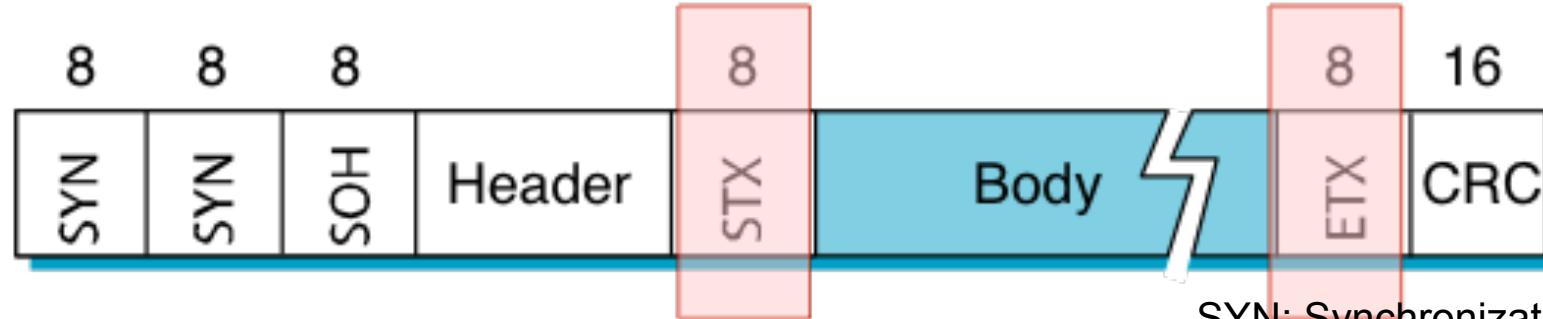
Count: Specifies # of bytes in the body

- (a)

5	1	2	3	4	5	6	7	8	9	8	0	1	2	3	4	5	6	8	7	8	9	0	1	2	3
Frame 1					Frame 2					Frame 3					Frame 4										
5 characters					5 characters					8 characters					8 characters										
- (b)

5	1	2	3	4	7	6	7	8	9	8	0	1	2	3	4	5	6	8	7	8	9	0	1	2	3
Frame 1					Frame 2 (Wrong)																				

起止标记法

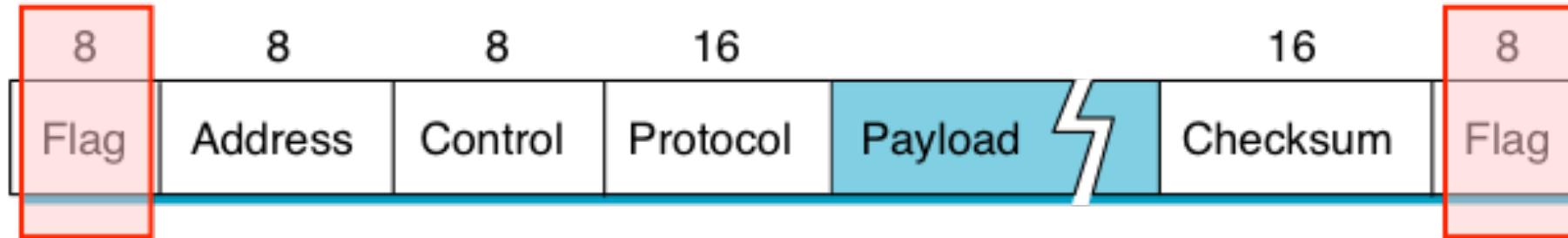


SYN: Synchronization character
 SOH: Start of header
 STX, ETX: Start of text, End of text
 CRC: Cyclic redundancy check

- BISYNC (二进制同步通信)
- IBM在1960s末期开发
- 特点
 - 起止字符: 开始和结束字符
 - STX (正文开始符), ETX (正文结束符)
- 问题
 - ETX字符可能出现在数据帧的数据部分
- 解决方案- 字符填充
 - 在数据部分的ETX前填充DLE (数据链路转义)
 - 数据部分的DLE前也重复填充DLE



起止标记法



- PPP (Point-to-Point Protocol)
- RFC 1547 (Requirements for an Internet Standard Point-to-Point Protocol, December 1993) provides historical information about the need for PPP and its development.
- RFC 1661 (The Point-to-Point Protocol, July 1994)
- 特点
 - STX和ETX: 0x7E
 - 数据帧的有效载荷字段的长度可以协商, 缺省值为1500字节
 - LCP (链路控制协议与NCP (网络控制协议)协商链路及网络参数



面向比特的协议

- 面向比特的协议
- 把数据帧看做比特的集合



- HDLC (高级链路控制规程)
- 起止比特位串:
 - 01111110
- 问题: 数据字段可能出现01111110
- 解决方法- 零比特填充



HDLC (高级链路控制规程)

- 发送端, 每连续发送5个1s
- 插入一个比特0

- 接收端, 每连续收到5个1s
- 如果后续比特为0: 删除0比特
- 如果后续比特为10: 帧结束
- 如果后续比特为11: 出错

- 比特填充的特点
- 可能连续两次接收失败
- 帧的长度由帧中有效载荷中传送的数据决定



提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结



比特错误

- 问题
 - 电磁干扰和热噪声可能导致比特错误
 - 数据帧有时会发生比特错误
- 解决方案
 - 差错检测
 - 接收方可以通过编码方式检测到差错
 - 差错纠正, 通常有两种方法:
 - 接收方通知发送方重发消息
 - 接收方重新构造消息

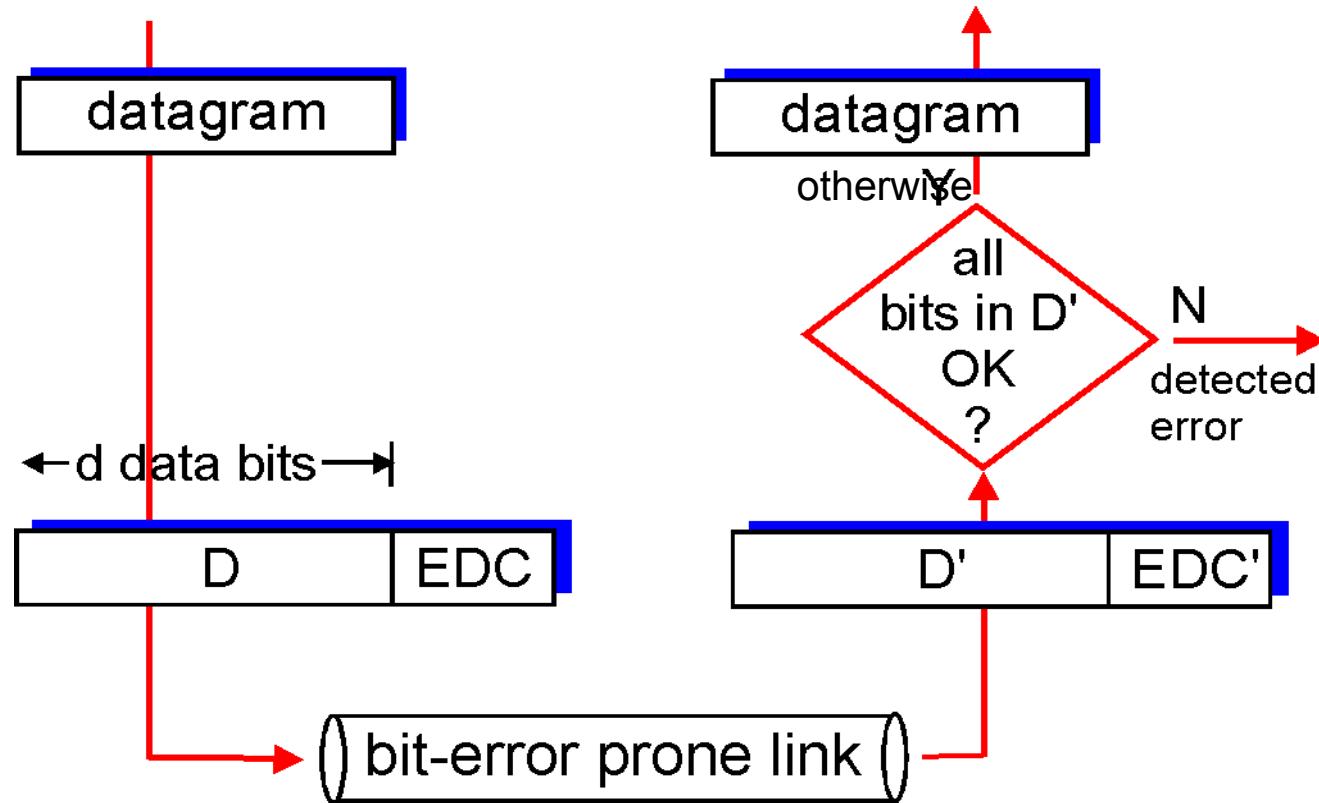


差错检测

- 差错检测通用方法（链路层或高层使用）
- 二维奇偶校验(数据链路层协议)
- 循环冗余校验, CRC (数据链路层协议)
- Internet校验和算法(传输层协议)

成帧原则	链路层协议	检错及纠错
面向字节	BISYNC	ASCII char: 二维奇偶校验 EBCDIC char: CRC
	PPP	CRC (名字称为校验和)
	DDCMP	CRC
面向比特	HDLC	CRC

差错检测的基本原理



EDC/ECC = 检错和纠错的比特(冗余填充)

D = 受差错保护的数据部分, 包括首部字段



差错检测的基本原理

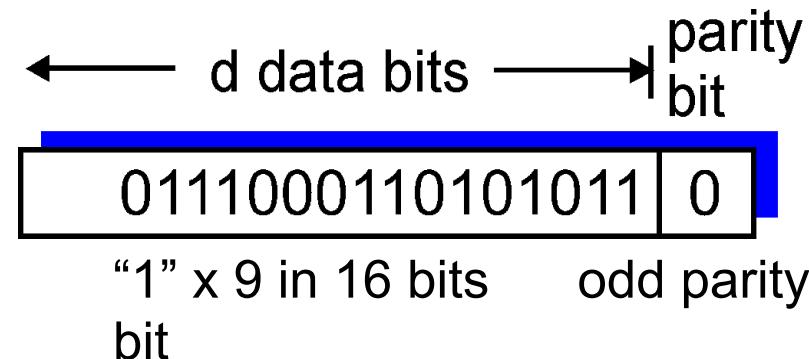
- 主要目标
- 冗余度
 - n = 数据长度, k = EDC长度
 - k 远小于 n
- 检错概率
- 概率最大化
- 差错检测并不一定100% 可靠!
 - 协议可能出现错误(极少数)
 - EDC字段越长检错效果越好



二维奇偶校验

基本的单比特奇偶校验

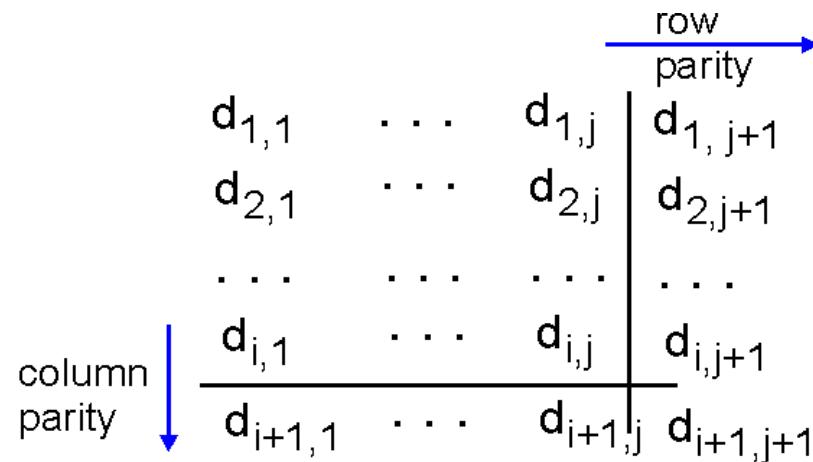
- 单比特教研:
- 检测 单比特错误



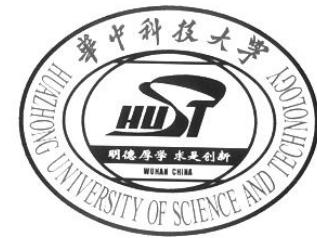
- 奇数校验根据字节中1的个数置位第8比特使得8个比特中1的个数为奇数
- 偶数校验根据字节中1的个数置位第8比特使得8个比特中1的个数为偶数.

二维奇偶校验

- 可以检测并纠正单比特错误



101011	101011	
111100	101100	parity error
011101	011101	
001010	001010	
no errors		
		correctable single bit error



互联网校验和算法

- 目标
 - 检测传输分组中的错误(e.g., flipped bits)
 - 注意: 仅在传输层使用
- 发送方:
 - 将数据分段为16比特整数的序列
 - 校验和: 对分段反码求和, 结果取反
 - 将结果写入IP分组的校验和字段
- 接收方:
 - 计算接收数据的校验和
 - 将计算结果于校验和字段的内容进行比较:
 - 不一致- 存在错误
 - 一致 - 正确



互联网校验和算法

- Internet 校验和算法是一种弱检测算法, 为什么IP协议中采用该算法 ?
- 易于软件实现
- ARPANET网的经验表明, 这种形式的校验和已经足够
- 网络中大部分的差错已由链路层更强的差错检测算法(例如CRC)所识别.



循环冗余校验

- 原理
- 以一个称为有限域的数学分支为依据
- 数学表述
- 待发送数据
- n -比特的数据看做 $n-1$ 次的多项式 $M(x)$
例如: 对于 10011010, $M(x) = x^7 + x^4 + x^3 + x^1$
- 检错码
- 表示为 $(n+k)$ 次的多项式 $P(x)$
- 基于 k 次的除数多项式 $C(x)$ 进行计算
例如: $C(x) = x^3 + x^2 + 1$, 其中 $k=3$



循环冗余校验

- 错误条件
- 接收方判断 $P(x)$ 是否可以整除 $C(x)$
- 如何计算 $P(x)$
- $M(x)$ 乘以 x^k 得到 $T(x)$
- $T(x)$ 除以 $C(x)$
- $T(x)$ 减去余数 (简单的XOR操作)



模2运算

- 类似于二进制运算, 但不存在借位

- 示例:

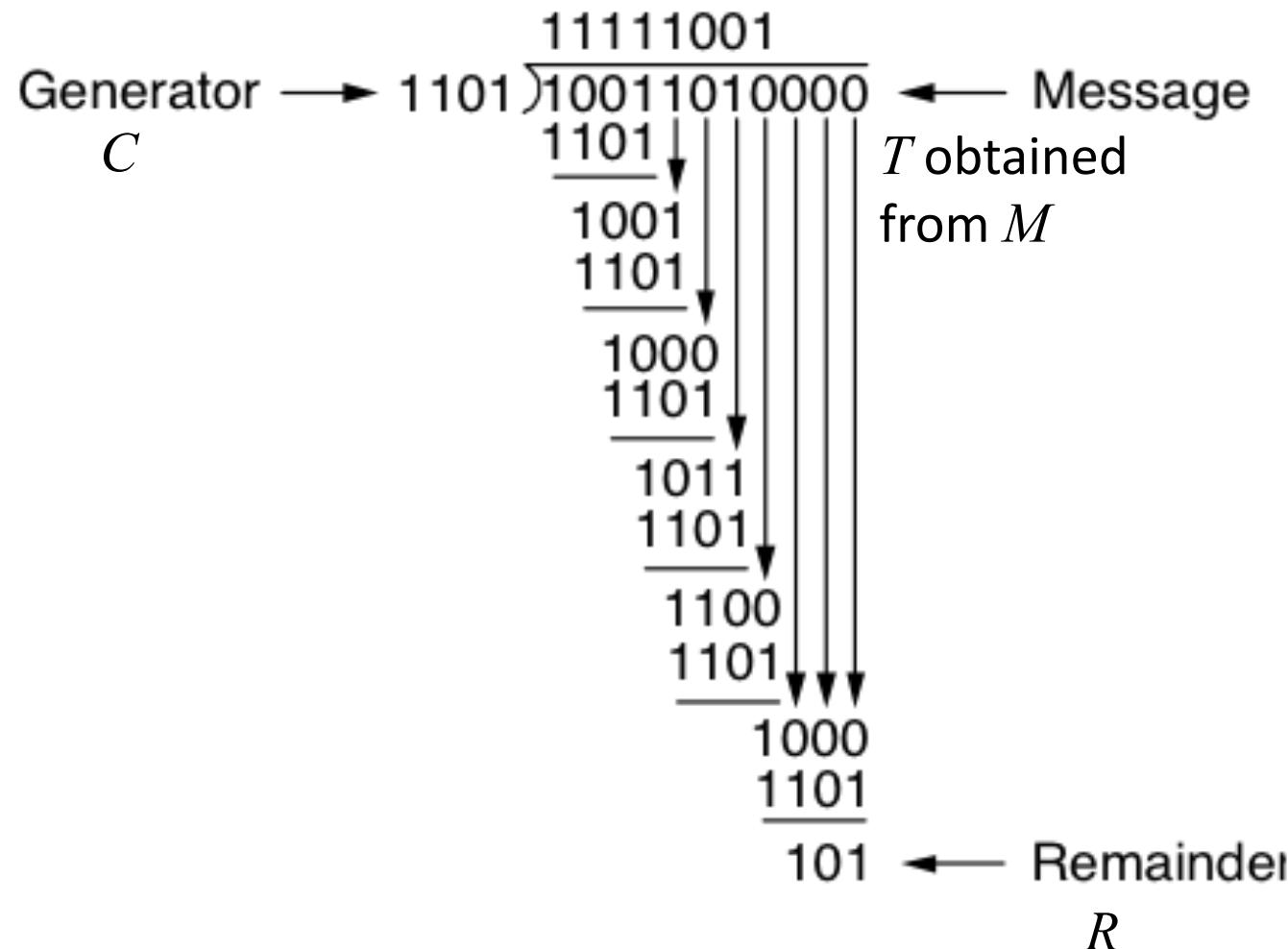
$$\begin{array}{r} 101 \\ + \\ \hline 010 \\ 111 \end{array} \quad \begin{array}{r} 101 \\ + \\ \hline 001 \\ 100 \end{array} \quad \begin{array}{r} 1011 + \\ 0111 \\ \hline 1100 \end{array}$$

$$\begin{array}{r} 101 \\ - \\ \hline 010 \\ 111 \end{array} \quad \begin{array}{r} 101 \\ - \\ \hline 001 \\ 100 \end{array} \quad \begin{array}{r} 1011 - \\ 0111 \\ \hline 1100 \end{array}$$

- 模2运算中的加法和减法等同于XOR操作

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

循环冗余校验



$$M = 10011010, C(x) = x^3 + x^2 + 1$$

$$T(x) = 10011010 | 000$$

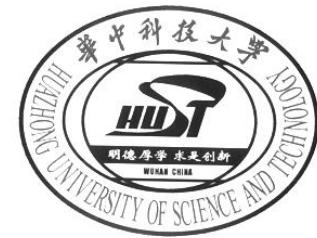
$$R(x) = 101$$

$$P(x) = M \text{ XOR } R(x)$$

$$= 1001\ 1010\ 000 \text{ XOR } 101$$

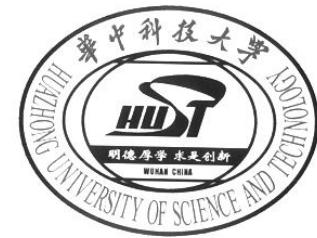
$$= 1001\ 1010\ 101$$

$$P(x) = 10011010 | 101$$



循环冗余校验

- 选择除数多项式
- 不同的 $C(x)$ 可以检测出特定的比特错误
- 合理的选择除数则可以：
 - 检测出所有1 & 2-比特的错误
 - 任意奇数个错误
 - 任何小于 k 比特的连续比特错误序列
 - 部分大于 k 比特的连续比特错误序列
- 硬件实现
- 使用一个 k 比特移位寄存器和若干个XOR门



循环冗余校验

- 常用的CRC 多项式

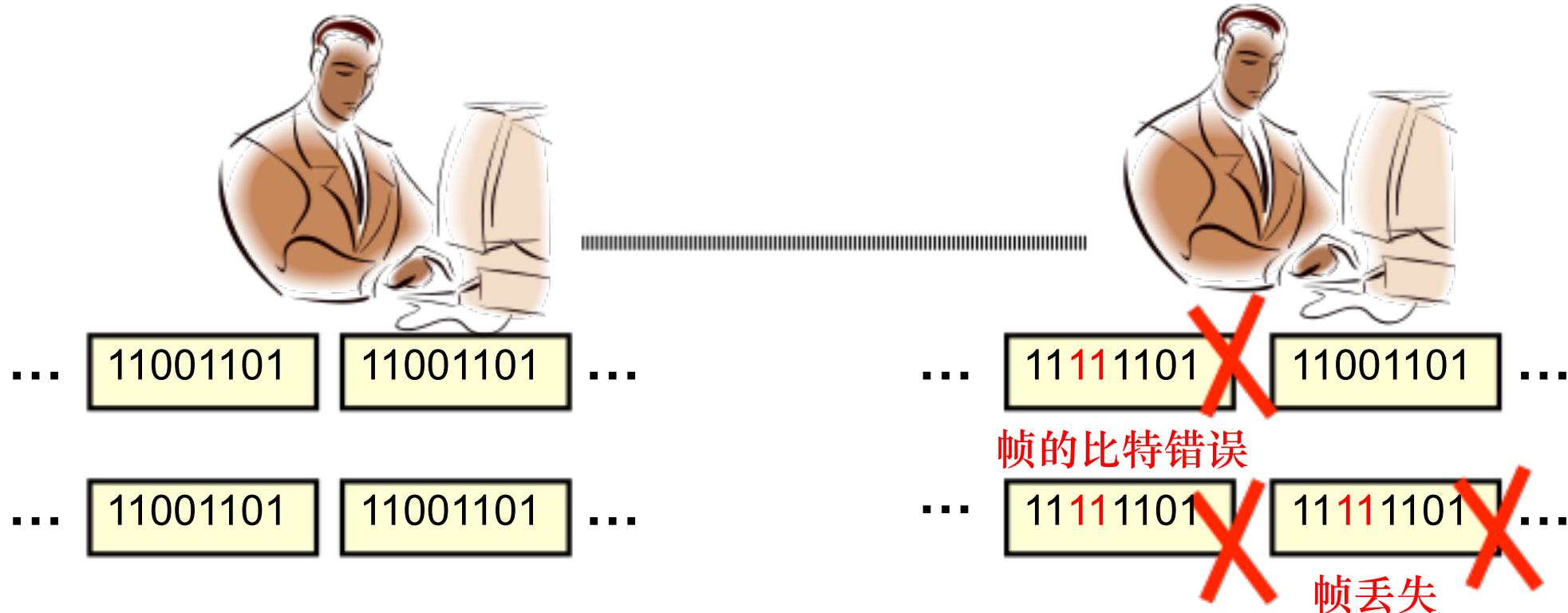
CRC	$C(x)$	<u>Link Protocol</u>
CRC-8	$x^8 + x^2 + x^1 + 1$	ATM
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^1 + 1$	ATM
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + 1$	
CRC-16	$x^{16} + x^{15} + x^2 + 1$	
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$	HDL
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+ x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	ATM, Ethernet, 802.5



提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
 - 停止等待协议
 - 连续ARQ协议
 - 滑动窗口协议
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

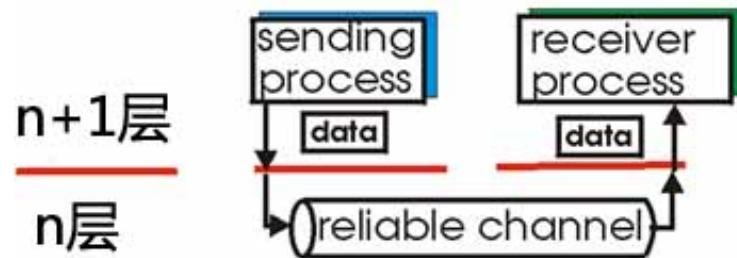
直连网络存在的基本问题



- 可靠传输
- 在可能发生传输错误的物理链路上建立一条可靠的数据链路

可靠数据传输的原理

- 在应用层，传输层和链路层中非常重要
- 计算机网络中最重要的十大研究专题之一

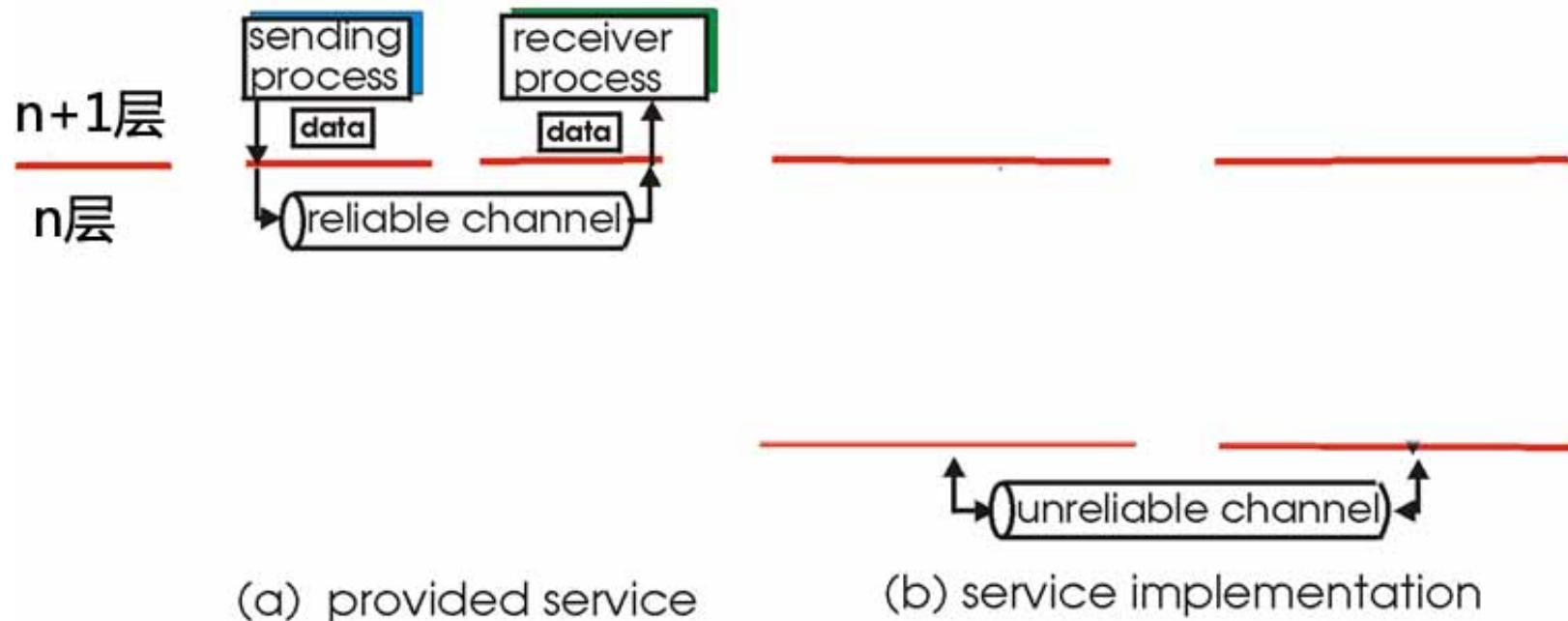


(a) provided service

- 不可靠通道的特性决定了可靠数据传输协议（rdt）的复杂度

可靠数据传输的原理

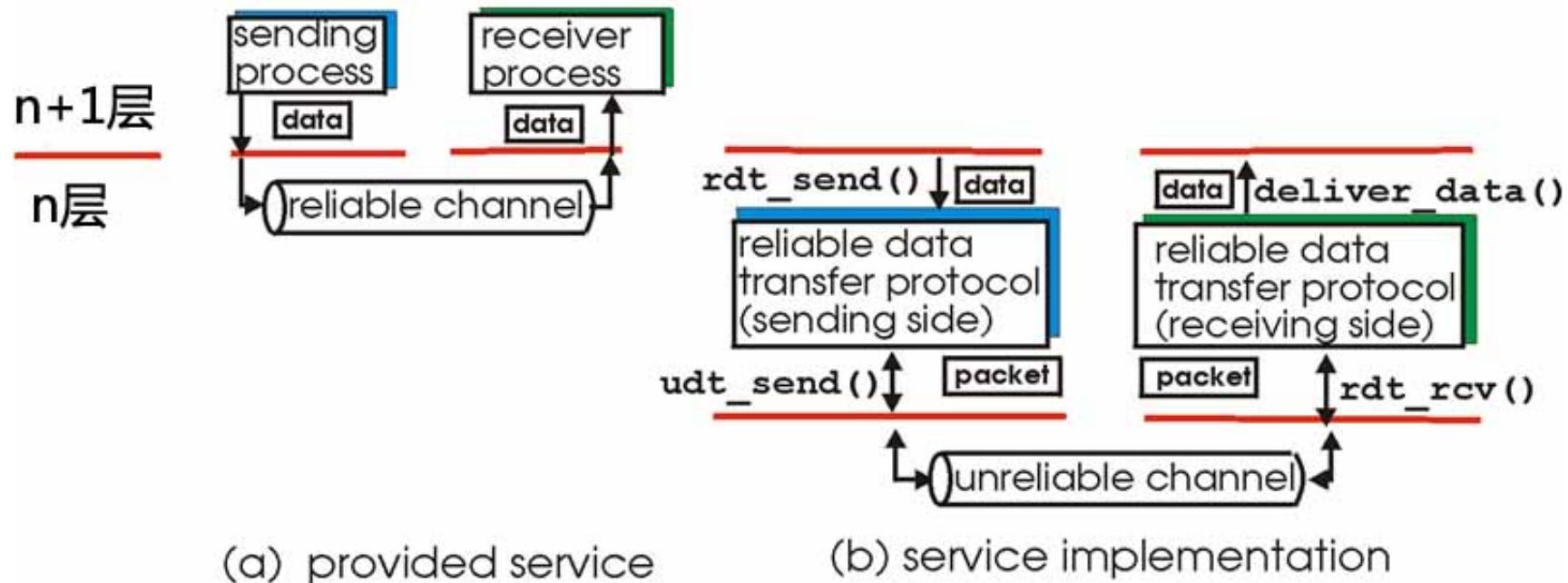
- 在应用层，传输层和链路层中非常重要
- 计算机网络中最重要的十大研究专题之一



- 不可靠通道的特性决定了可靠数据传输协议（rdt）的复杂度

可靠数据传输的原理

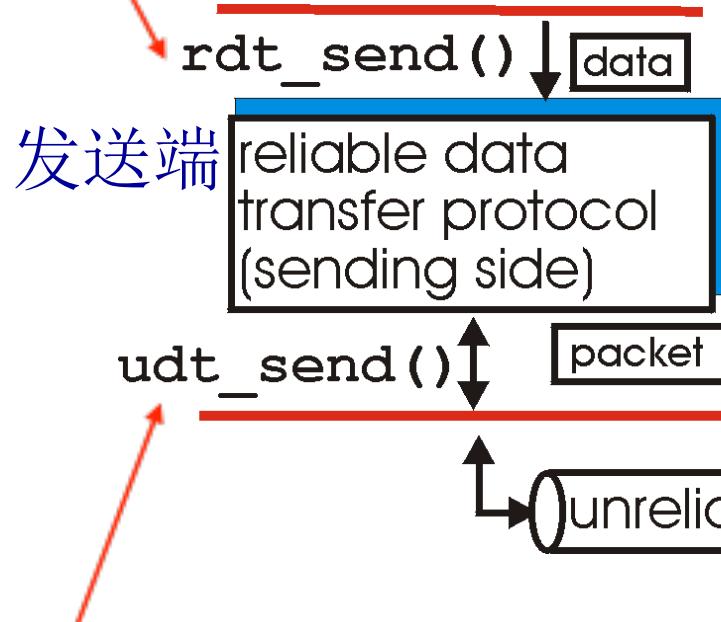
- 在应用层，传输层和链路层中非常重要
- 计算机网络中最重要的十大研究专题之一



- 不可靠通道的特性决定了可靠数据传输协议（rdt）的复杂度

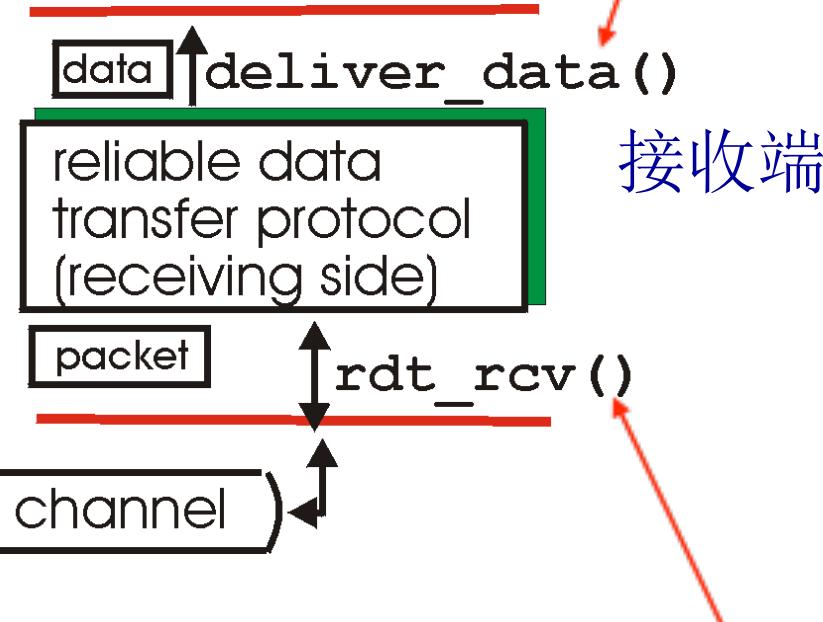
可靠数据传输的原理

rdt_send() : called from above, (e.g., by app.). Passed data to deliver to receiver upper layer

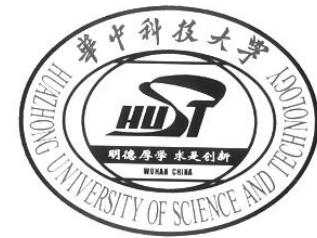


udt_send() : called by rdt, to transfer packet over unreliable channel to receiver

deliver_data() : called by rdt to deliver data to upper



rdt_rcv() : called when packet arrives on rcv-side of channel



可靠传输

- 问题
 - 数据帧可能出现比特错误
 - 数据帧整个被破坏(帧丢失)
- 基本解决方法
 - **ACK (确认):** 接收方根据差错检测的结果向发送方发送一个控制帧进行确认
 - ACK = 正确接收, NAK = 帧错误
 - **超时:** 如果发送方在一定的时间范围内未收到来自接收方的确认, 则**重传**数据帧
 - **帧序号:** 识别数据帧

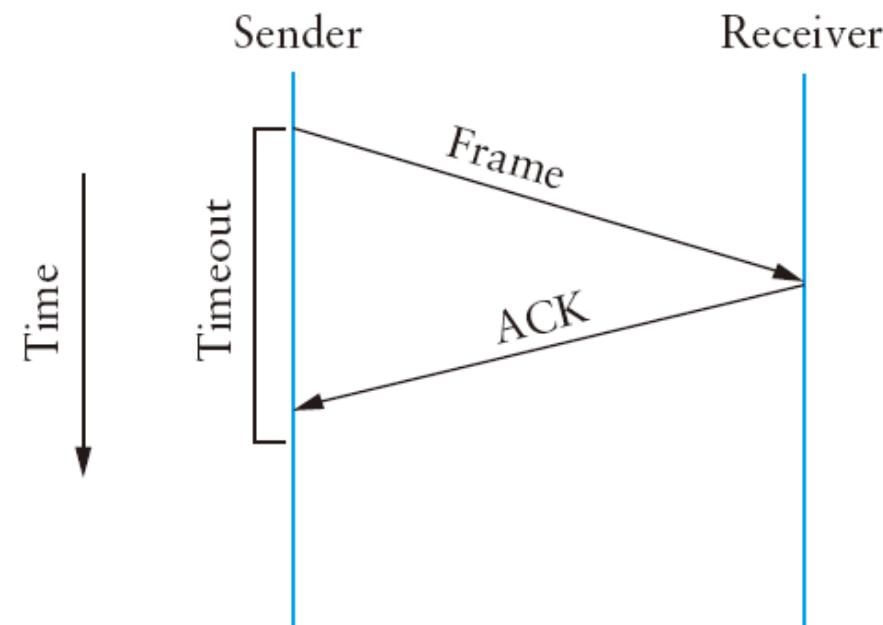


自动请求重传 (ARQ)

- ARQ
- 采用确认和超时定时器的可靠传输机制
- 链路层假设
 - 串行通信信道：传输过程中不存在帧的乱序
 - 所有数据帧的传播时延相同
- 接收方
 - 对数据帧进行差错检测
 - 对正确帧进行确认，丢失错误帧
 - 丢弃禁止接收的数据帧
- 发送方
 - 发送原始数据帧
 - 对错谬帧和丢失帧进行重传

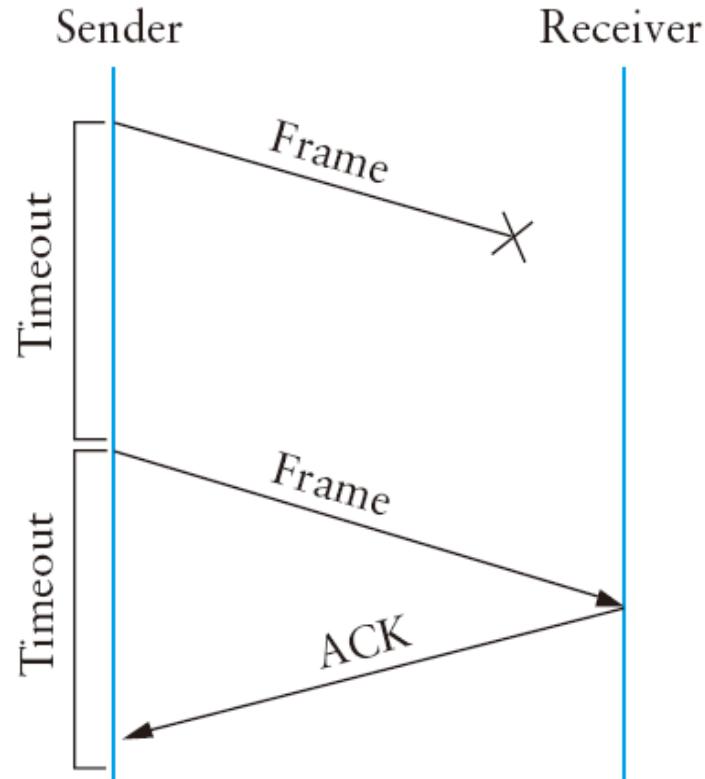
停止等待协议

- 最简单的ARQ机制
- 每发送完一个数据帧，发送方在继续发送下一个数据帧之前必须等待确认
- 如果在一定的时间范围内，发送方未收到确认(ACK)，则发送定时器超时激发，发送方重传原始数据帧.

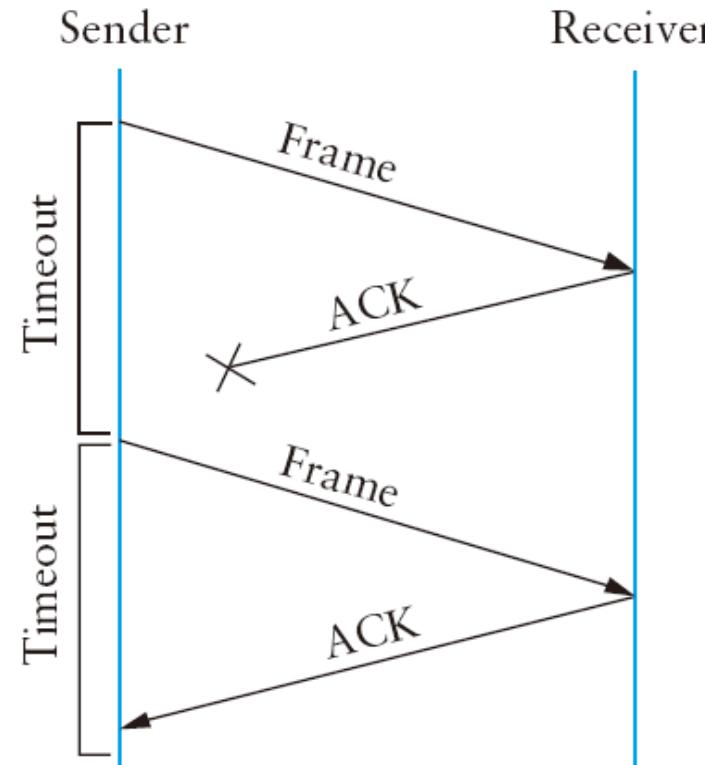


停止等待协议

- 两种不同的帧丢失情况

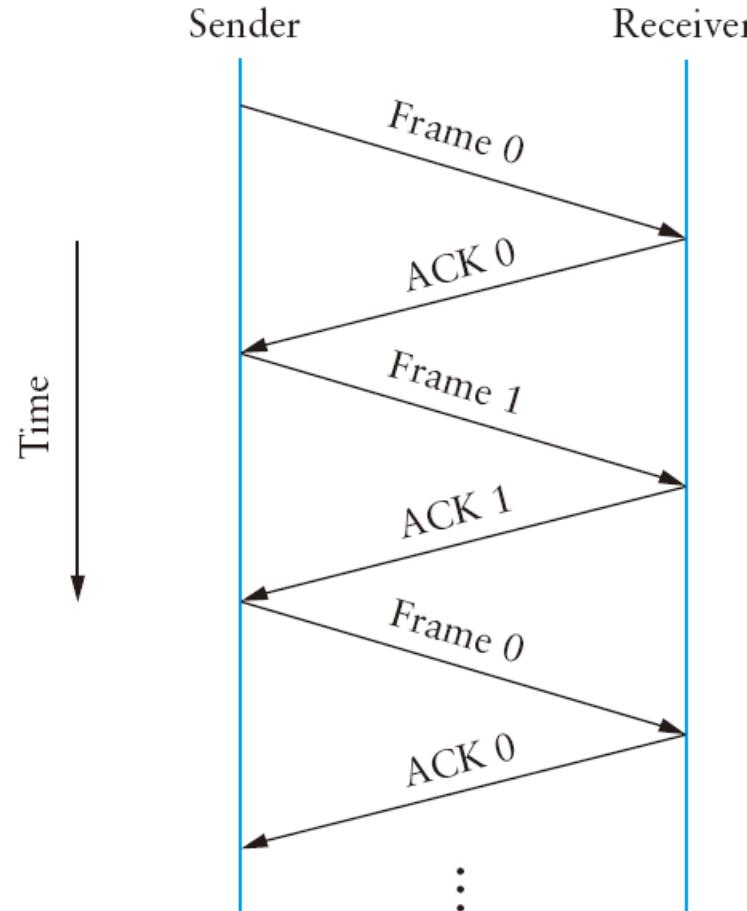
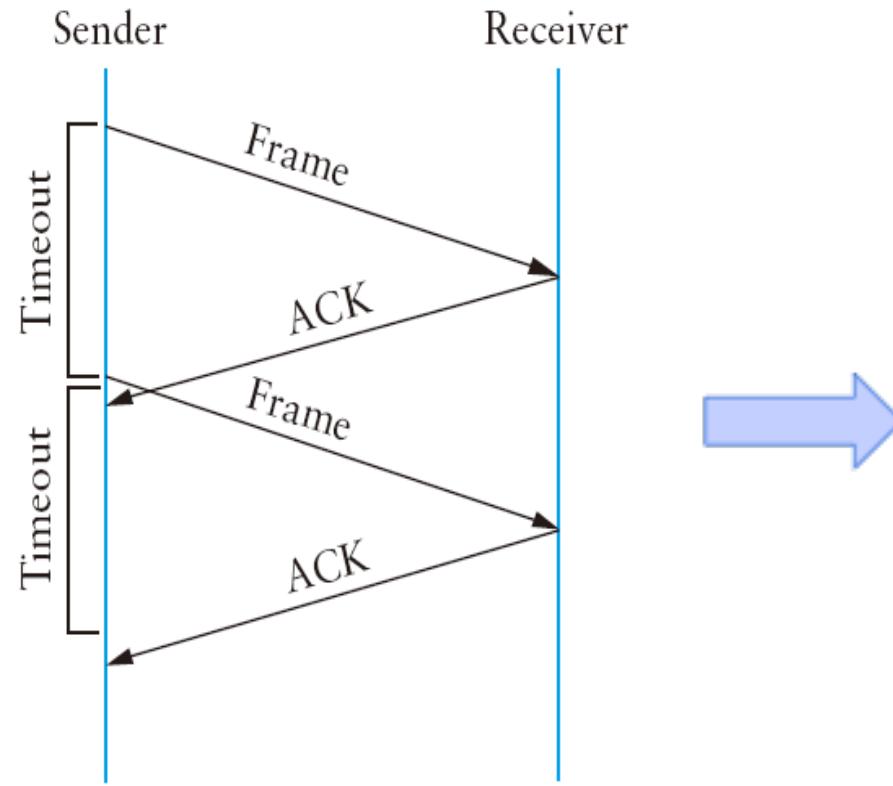


数据帧丢失



ACK丢失

停止等待协议



- 问题: 超时重传
- 解决方案: 帧序号 (SeqNum)
- 避免确认丢失导致的重复数据帧
- 在有线传输媒质中, 只需1比特



停止等待协议

- 缺点
 - 链路带宽利用率较低
 - 示例
 - 链路带宽为2 Mbps, RTT为45 ms, 数据帧大小为1.5KB
 - 每一个RTT内, 发送方仅能发送一个数据帧
 - 吞吐量为 $1500 \times 8 / 0.045 = 266.7 \text{ kbps}$
- 如何改进
 - 充分利用“管道”流水传送!

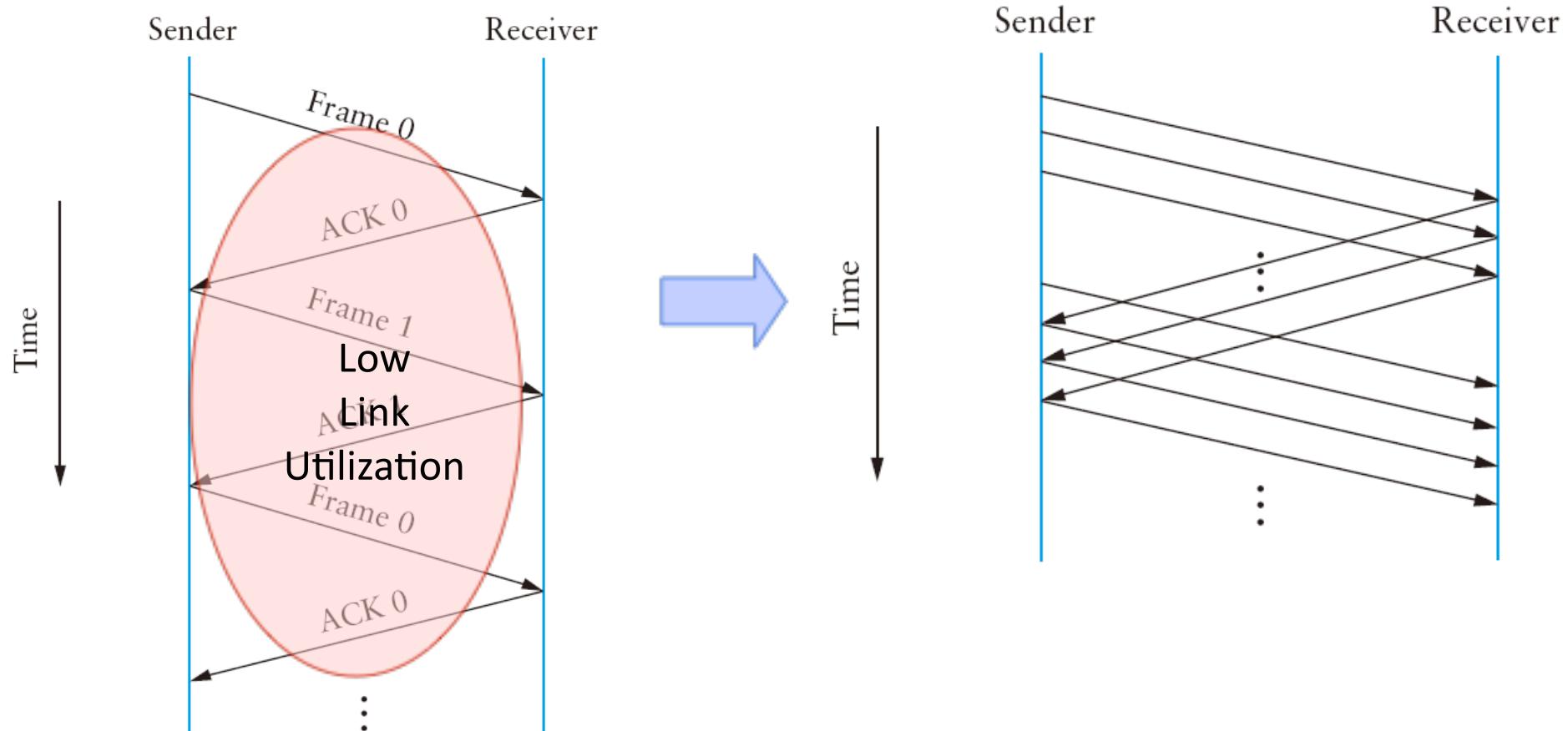


提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
 - 停止等待协议
 - 连续ARQ协议
 - 滑动窗口协议
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结



连续ARQ协议

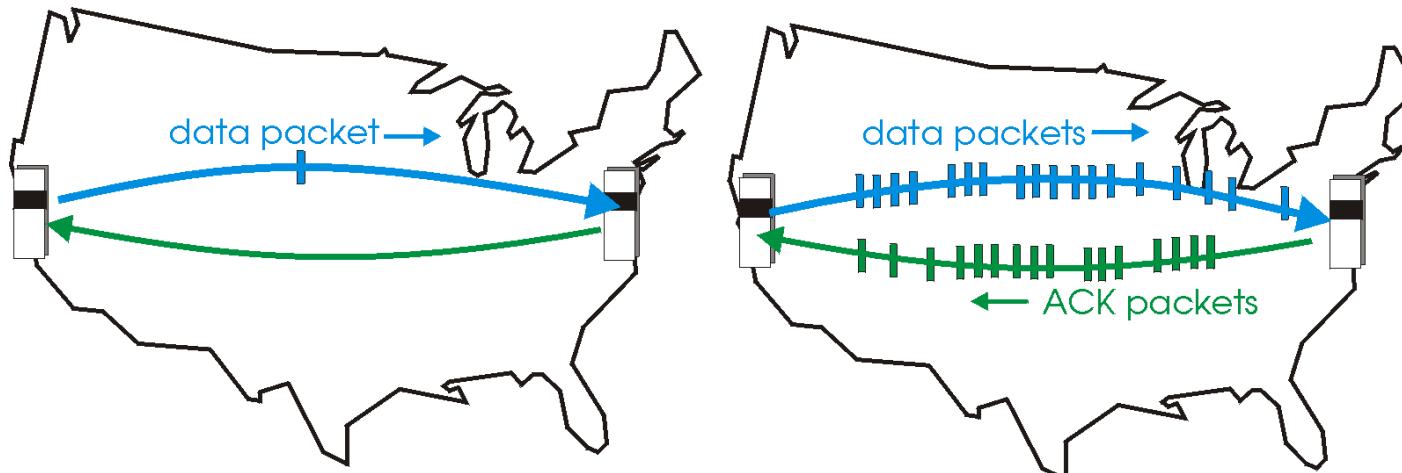


- 问题: 链路带宽利用率较低
- 解决方法:
- 采取流水线设计, 允许未收到确认, 连续发送数据帧

流水线协议

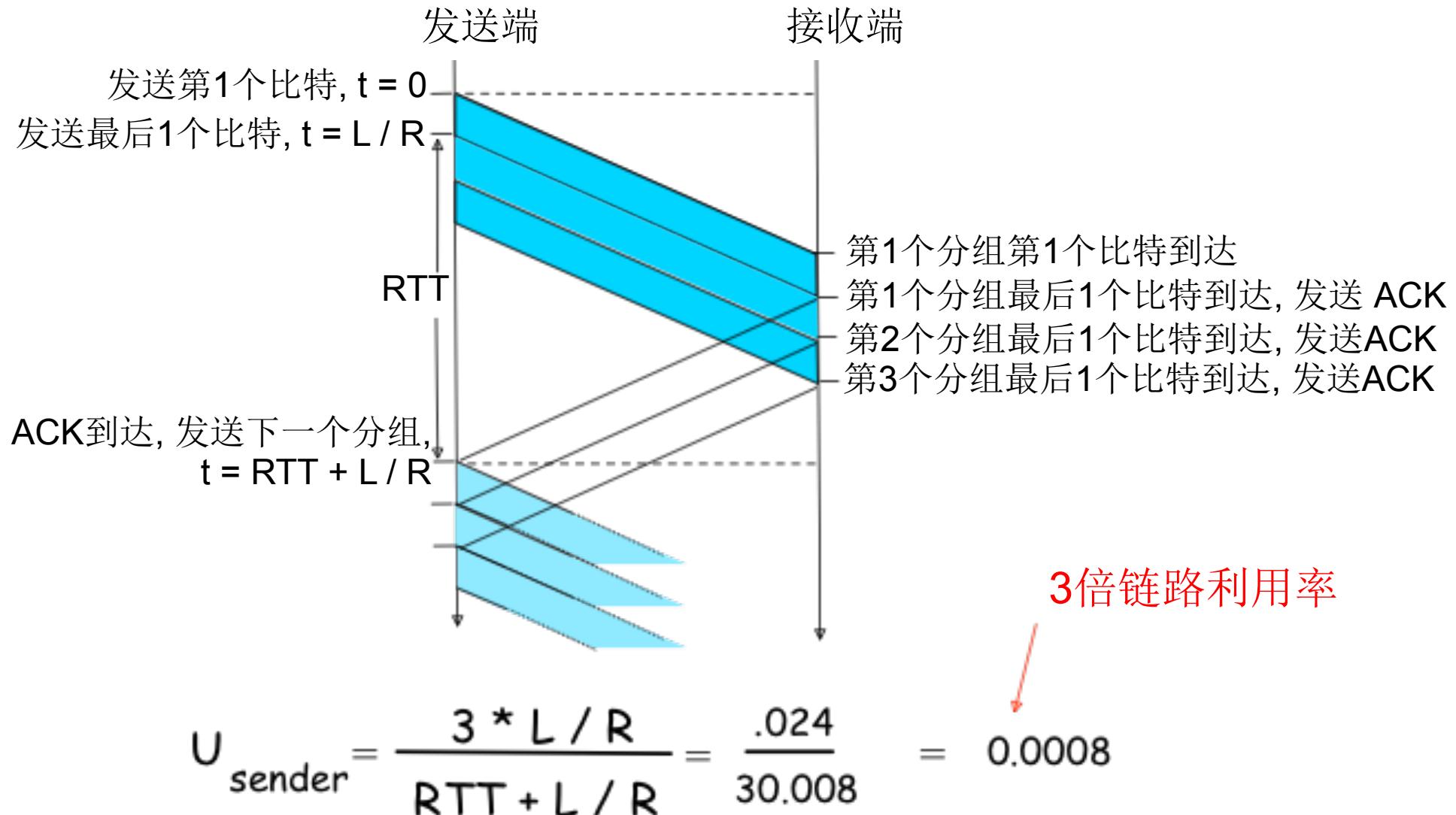
流水线: 允许发送端未收到确认，连续发送多个“传输中”的数据帧

- 必须增加序列号的允许范围
- 发送端和接收端需要缓存



- 流水线协议的两种基本形式
 - Go-Back-N
 - 选择性重传

流水线: 增加链路利用率



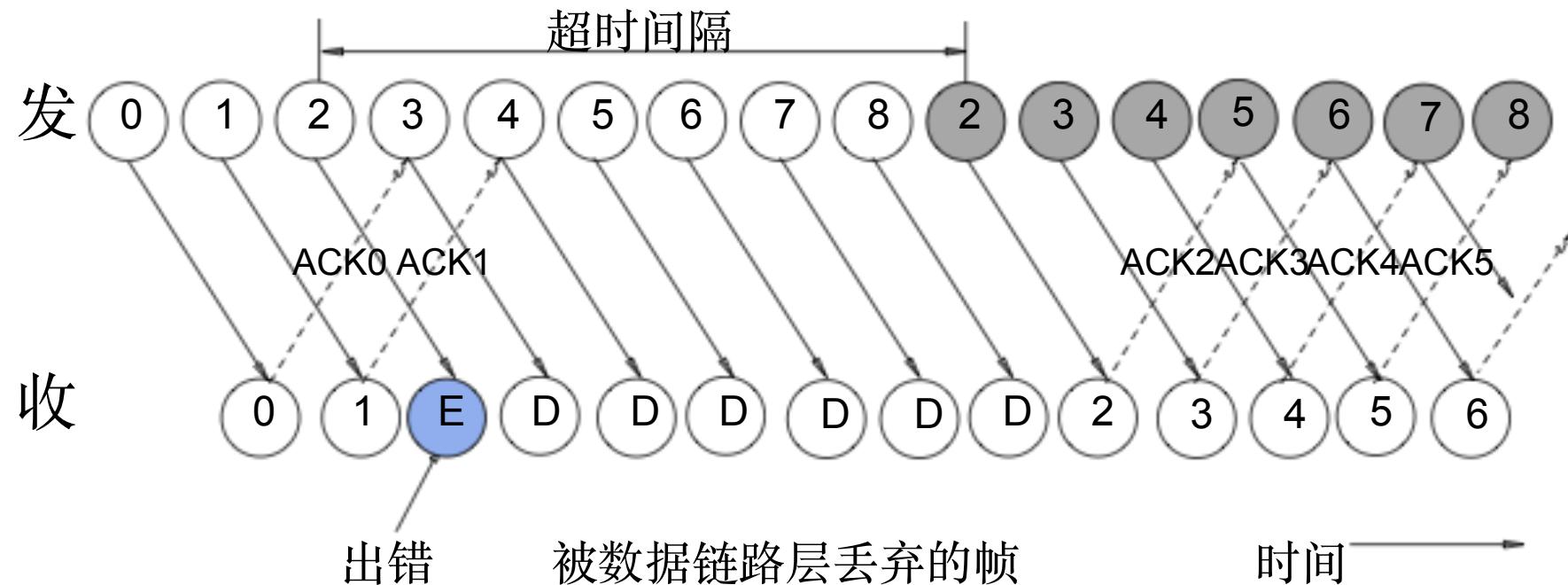


连续ARQ协议的两种策略

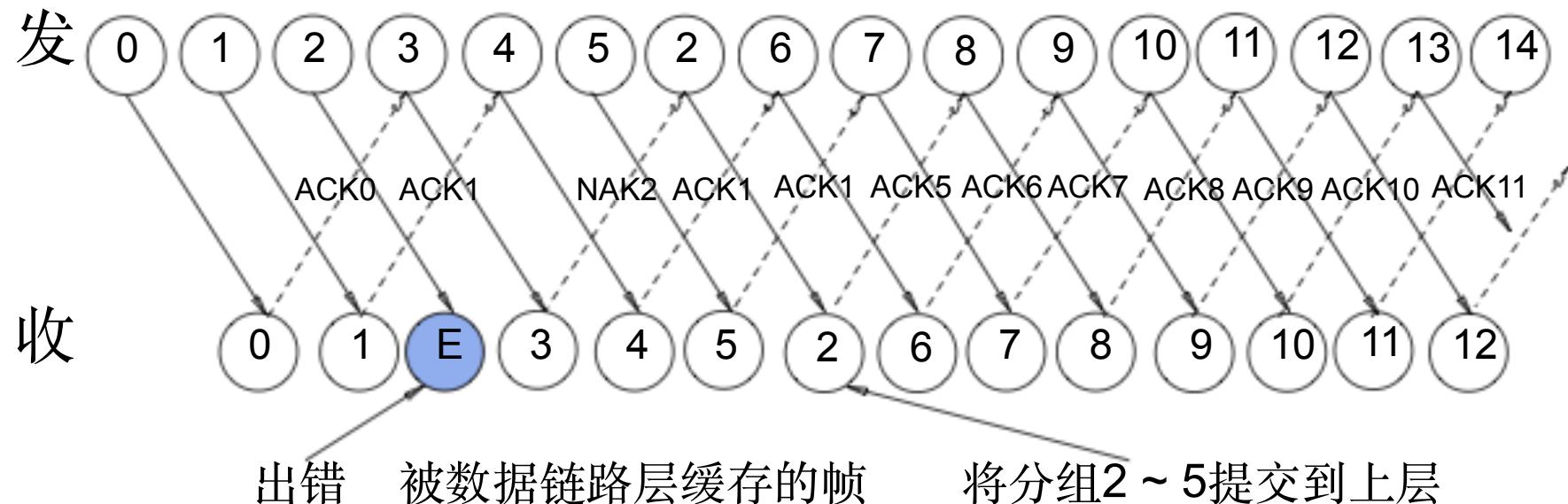
- Go-Back-N
 - 一次性发送N个数据帧；
 - 如果第k个帧丢失，对[k, k+N-1]范围内的所有帧重传。
 - 优点：接收方不需要缓存接收到的乱序帧，确认简单
 - 缺点：正确帧也可能被重传，效率较低

- 选择性重传
 - 一次性发送N个数据帧；
 - 如果第k个帧丢失，仅重传第k个帧；
 - 接收方对每一个帧进行确认。
 - 优点：链路利用率较高
 - 缺点：接收方更复杂

Go-Back-N策略



选择重传策略





提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
 - 停止等待协议
 - 连续ARQ协议
 - 滑动窗口协议
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结



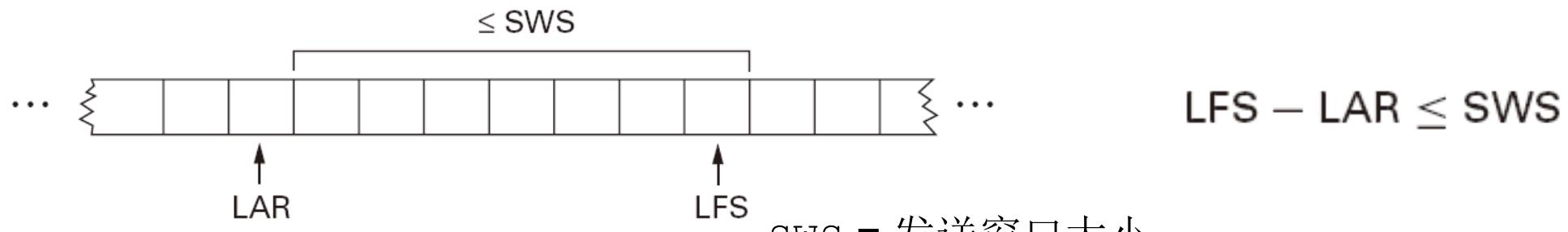


滑动窗口协议

- 引入**滑动窗口**对收发行为进行控制
- 发送方
 - 发送窗口大小: 发送方在未收到确认前能够发送的数据帧的最大个数
 - 发送方在未收到确认前最多可以发送多个数据帧 (受限于**发送窗口大小**)
 - 对未确认的数据帧缓存
- 接收方
 - 接收窗口大小: 接收方所能接收的乱序期望数据帧的最大个数
 - 接收方通过ACK告知发送方下一次期望其传送的数据帧编号, 避免每次收到数据帧都发送确认

滑动窗口协议

- Sender side

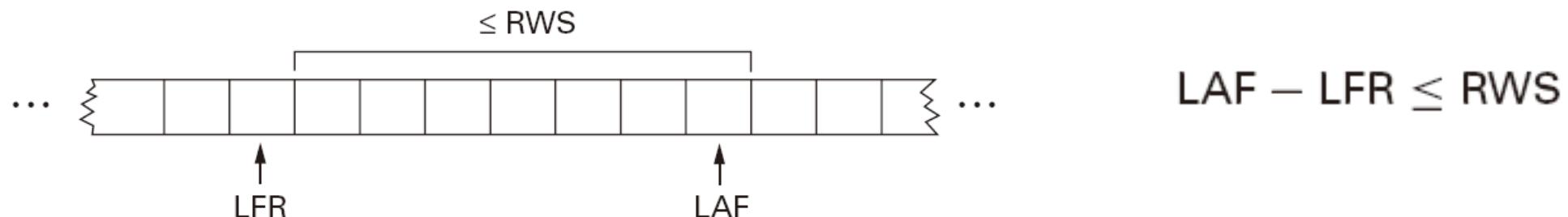


SWS = 发送窗口大小

LAR = 最后收到的ACK中的SeqNum

LFS = 最后发送的数据帧的SeqNum

- Receiver side



RWS = 接收窗口大小

LAF = 允许接收的数据帧的最大SeqNum

LFR = 最后接收的数据帧的SeqNum

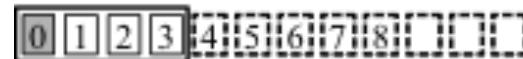


滑动窗口协议

- 接收方: 接收数据帧
 - 如果一个数据帧到达(帧序号为SeqNum)
 - 如果 $\text{SeqNum} \leq \text{LFR}$ 或 $\text{SeqNum} > \text{LAF}$,
数据帧落在接收窗口外, 则丢弃该数据帧.
 - 如果 $\text{LFR} < \text{SeqNum} \leq \text{LAF}$,
数据帧落在接收窗口内, 则接收.
- 接收方: 回复ACK
 - $\text{SeqNumToAck} = \text{未确认数据帧的最大SeqNum}$
 - 接收方进确认 SeqNumToAck 之前的数据帧, 及时更大序号的数据已接收.
 - 设置 $\text{LFR} = \text{SeqNumToAck}$, 调整 $\text{LAF} = \text{LFR} + \text{RWS}$

滑动窗口ARQ

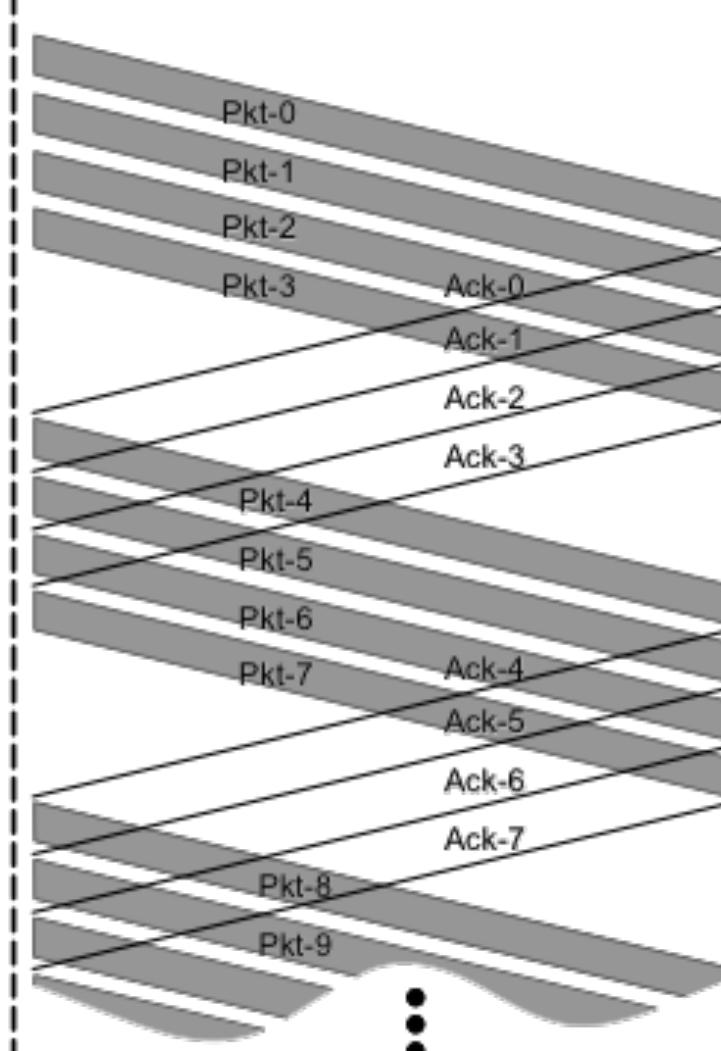
Sender
window N = 4



Key:

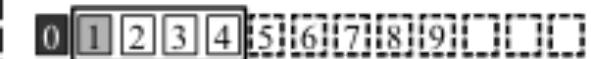
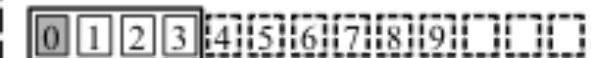
- Already ACK'd
- Sent, not yet ACK'd
- Allowed to send
- NOT allowed to send

Sender



Receiver

Receiver
window W = 4

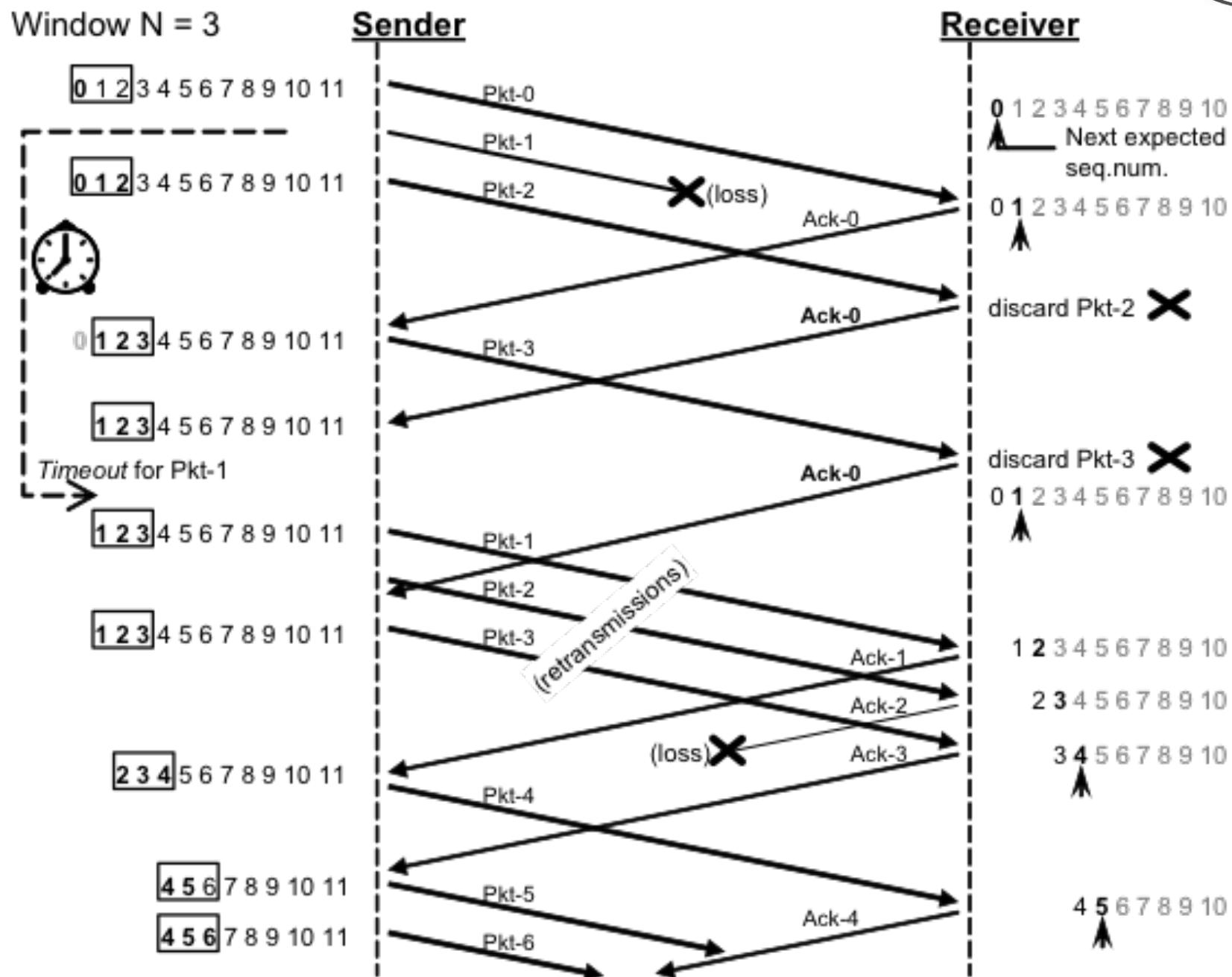


Key:

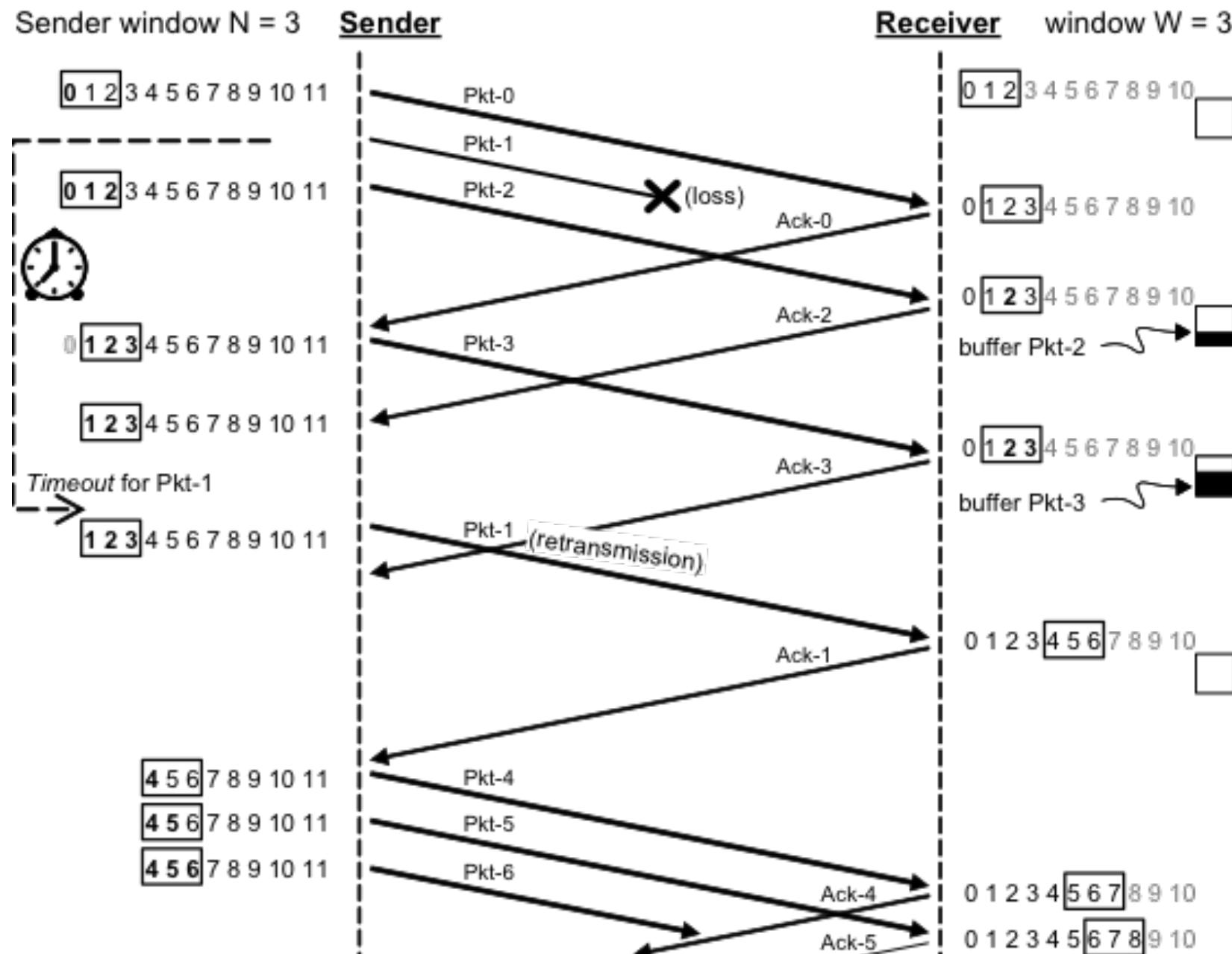
- Received in-order & ACK'd
- Expected, not yet received
- Acceptable to receive
- NOT acceptable

>>Demo>>

Go-back-N ARQ 协议



选择重传ARQ协议

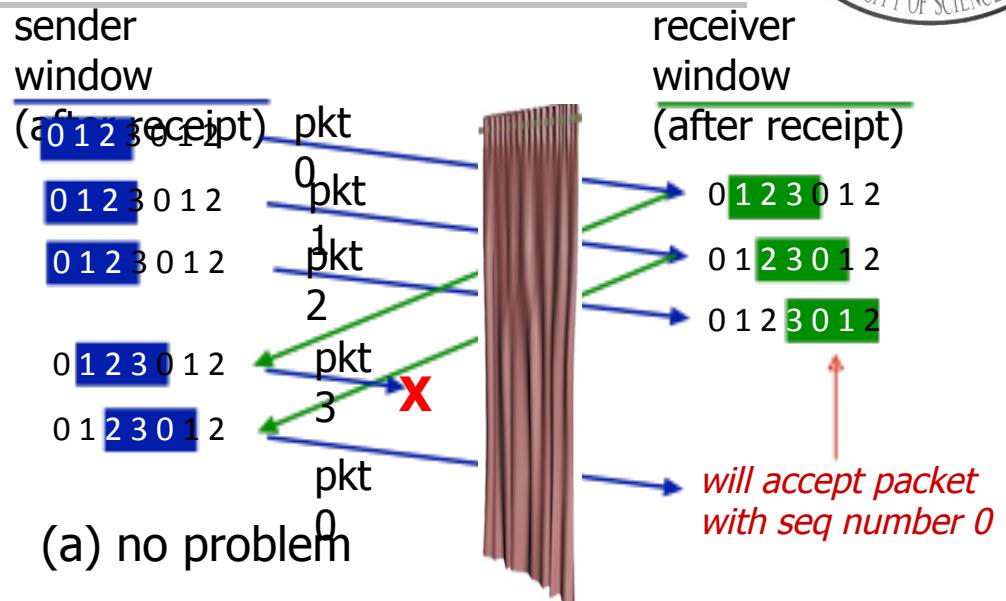


选择重传ARQ协议

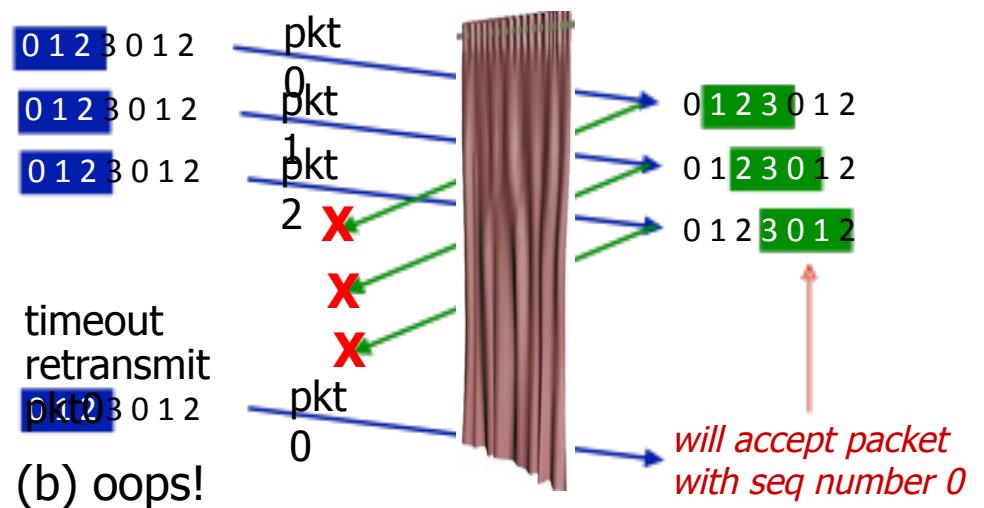
实际案例：

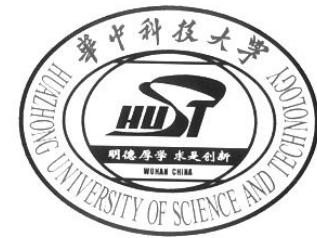
- 帧序号字段k=2
- seq #'s: 0, 1, 2, 3
- 窗口大小=3
- 接收端无法区分右边的两个场景！
- 在(a)中，接收端错误的将重复发送的数据视为新到数据

Q: 序列号大小和窗口大小应满足什么关系？



*receiver can't see sender side.
receiver behavior identical in both cases!
something's (very) wrong!*





有限范围的帧序号

- 发送方
- 对数据帧编号
- 帧中包含 k 比特 的帧序号字段
- 帧序号循环使用
- 帧序号的有效取值范围: $0 \sim 2^k - 1$

- 可以通过帧序号的周期区别相同序号的数据帧
- 如果接收窗口大小 = 1
 - 发送窗口大小 $\leq 2^k - 1$
- 如果接收窗口 = 发送窗口大小
 - 发送窗口大小 $\leq 2^k - 1$



小结: 滑动窗口协议

- 滑动窗口协议
不仅:
 - 保证帧在物理链路上的可靠传输
- 而且:
 - 保证帧传送的顺序
 - 通过帧序号和滑动窗口, 数据链路层协议可以将数据帧按发送顺序地交付给高层协议
- 支持流量控制, 接收方通过反馈机制可以压制发送方的速率
- 同步发送方的发送(帧)速率和接收方的接收(帧)速率



小结: 可靠传输

- 可靠传输的核心机制

机制	用途
CRC	差错检测
ACK	成功接收数据帧的确认(携带ACK序号)
定时器	检测发送方超时时间(帧丢失)
帧序号	识别不同的帧(避免帧丢失导致的重复帧)
滑动窗口	控制数据帧的收发 流量控制(保证传输的顺序, 控制传输速率)

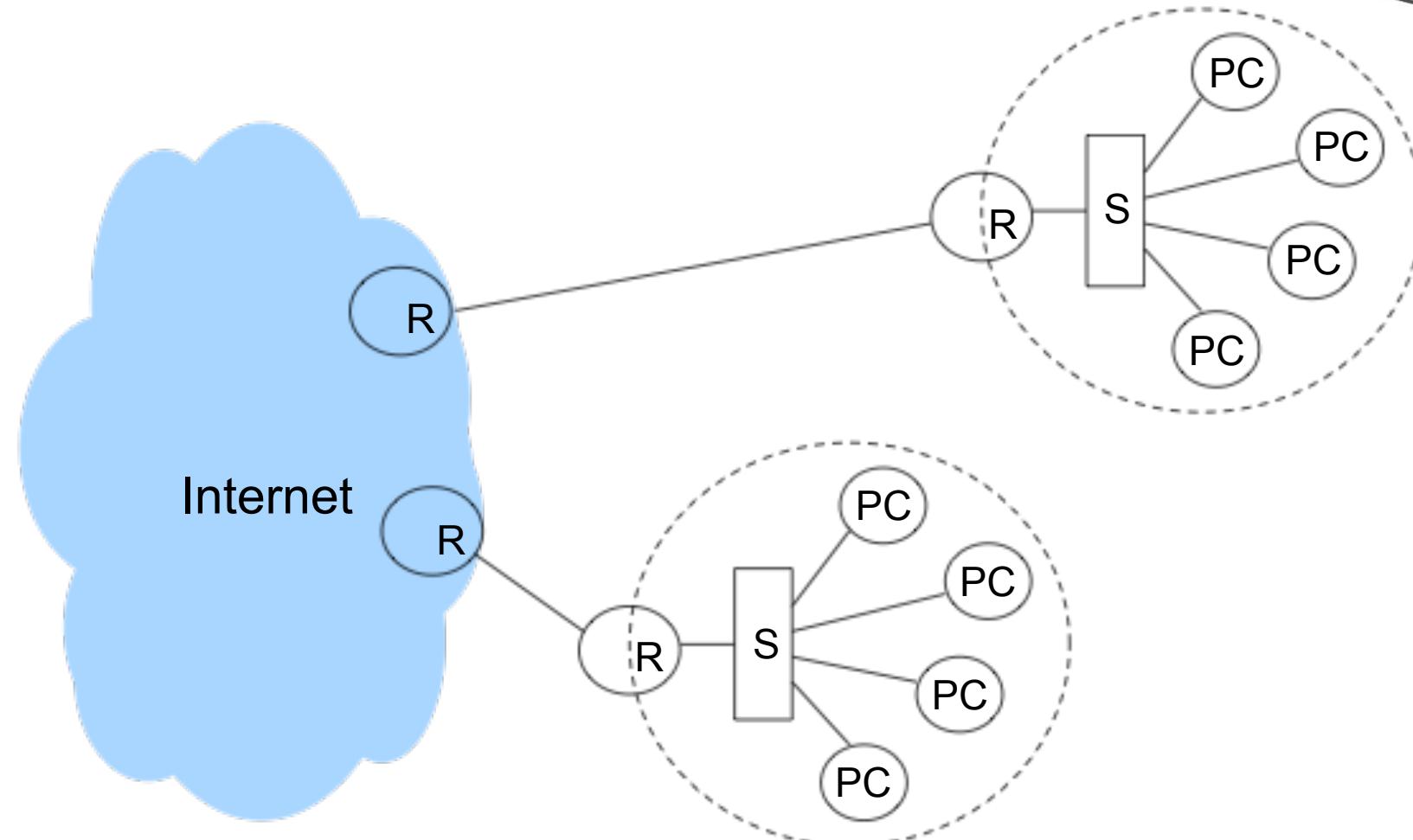


小结: 可靠传输

- 协议涉及的关键问题
 - 编码
 - 帧定界
 - 差错检测
 - 可靠传输**
- 媒质接入控制
- 协议举例

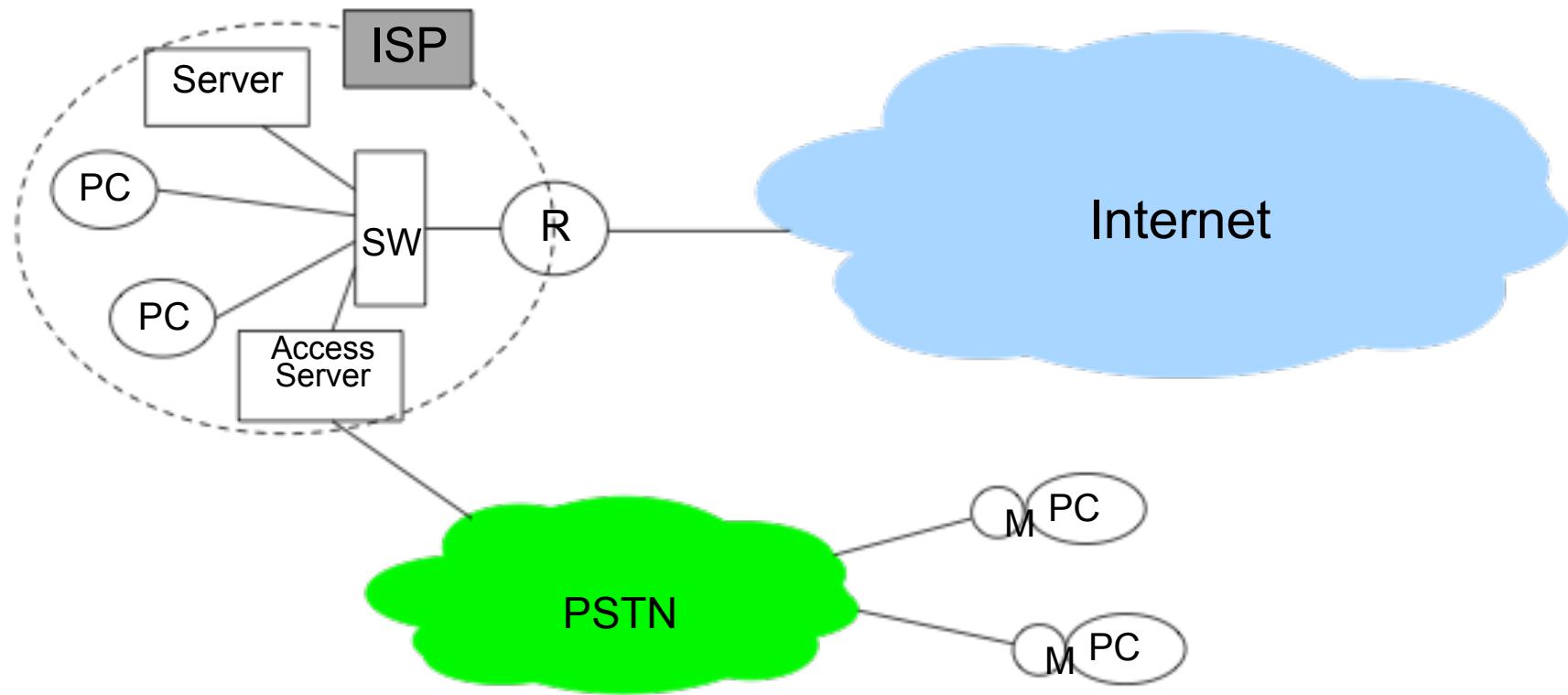
组件	协议			
	BISYNC	DDCMP	PPP	HDLC
可靠传输	停止等待		连续ARQ	
差错检测	二维奇偶校验		CRC	
帧定界	面向字节			面向比特
物理链路特点	点到点			

互联网的点到点链路层协议



- 路由器之间可能是一根点对点的链路(HDLC)

互联网的点到点链路层协议



- 家庭用户通过拨号与远程ISP联接后，采用点对点方式通信(PPP)

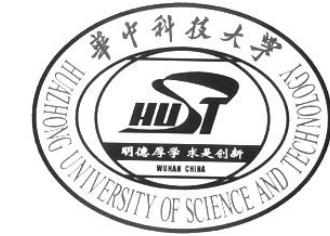


数据链路层协议示例

- HDLC (高级链路控制规程)

8bit	8	8	≥ 0	16	8
0111111 0	地址	控制	数据	校验和	0111111 0

帧标志序列	即01111110，作为帧的分隔标志，如线路空闲，则用标志序列填充，用位插入方法实现透明传输
地址域	在总线型多终端情况下，是终端的站号；在点对点的情况下，用来标志命令和响应
控制域	定义帧的类型、序号等和其它一些功能
数据域	用户数据，长度任意
校验和	CRC码，ISO和CCITT有相似的生成多项式



HDLC

- HDLC的帧有三种类型,不同的类型其控制域的定义不同

0	Seq	P/F	Next
1	0	Type	P/F
1	1	Type	P/F

- 信息帧 (I)
- 监控帧 (S)
- 无序号帧 (U)

P/F(Poll/Final)位

P 主机查询哪个终端要发送数据

F 终端发送数据的最后一帧用F

RR	1	0	0	0	P/F	捎带确认号
RNR	1	0	0	1	P/F	捎带确认号
REJ	1	0	1	0	P/F	捎带确认号
SREJ	1	0	1	1	P/F	捎带确认号



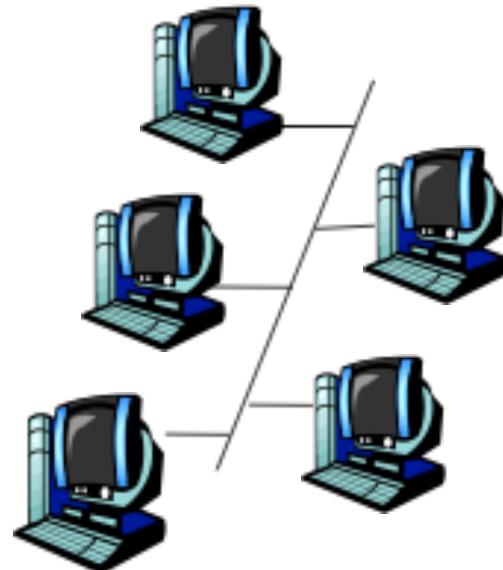
提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

多路访问

两种类型的“链路”：

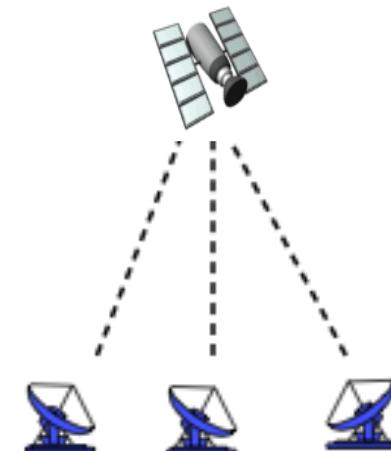
- 点到点链路
- 广播链路 (共享的有线/无线传输媒质)



共享的有线媒质(例如,
同轴电缆以太网)



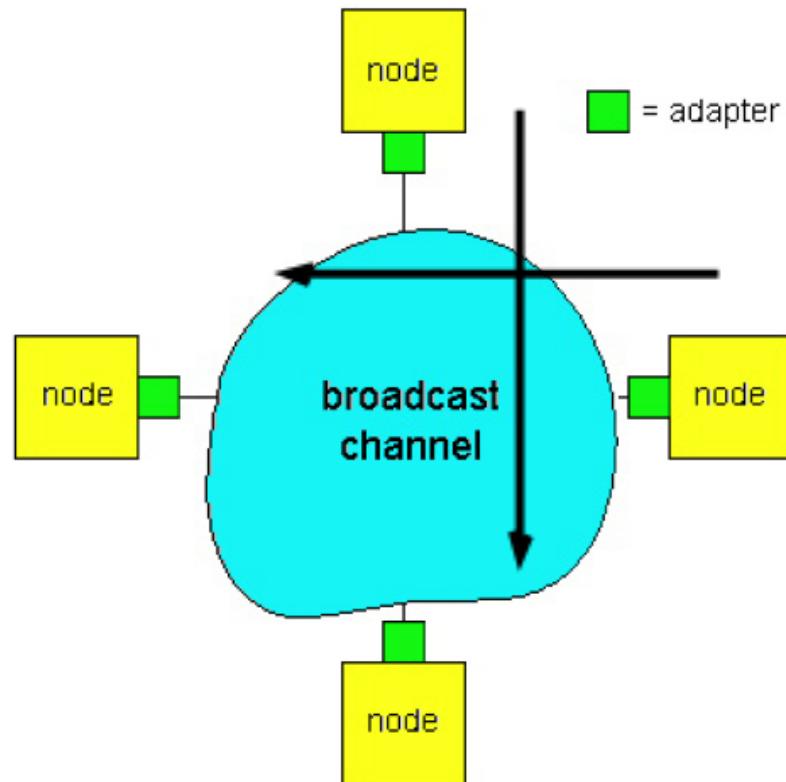
共享的无线媒质
(例如, 802.11 WiFi)



共享的无线媒质
(例如, 卫星通信)

多路访问

- 共享广播链路通信的基本问题
- 干扰: 如果两个或多个节点同时传输
- 冲突: 如果节点同时收到两个或多个信号



基本问题:

如何协调多个发送节点和多个节点的
接入一条共享的广播链路?



多路访问

- 基本解决方法
- 首先, 寻址
- 识别谁在说话
- 其次, 多路访问控制协议
 - MAC算法决定节点之间如何共享信道, i.e., 决定每一个节点什么时候可以发送数据
 - 分布式MAC协议简单, 应用广泛, i.e., WiFi
 - 信道共享信息的通信只能使用通信信道本身!
 - 不存在专门的信令信道



MAC 协议

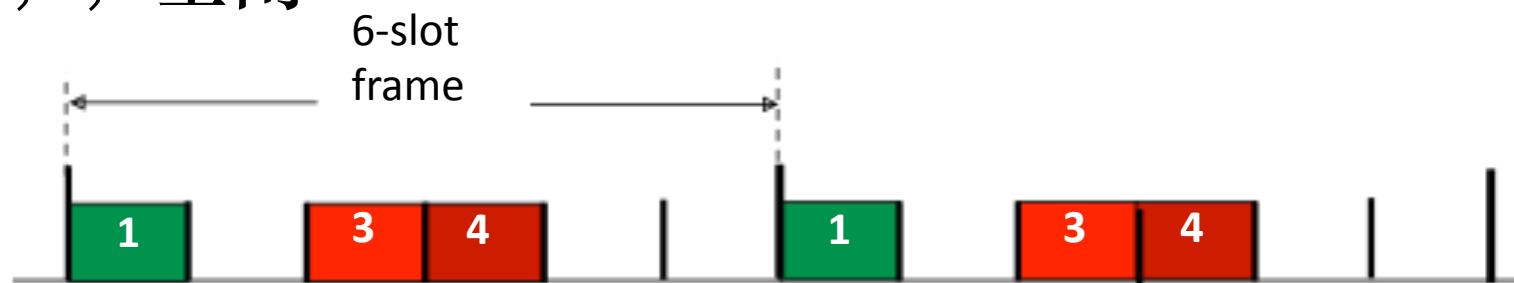
静态信道划分

- 将信道划分为较小的“分片”(时隙, 频率, 编码)
- 每一个分片被分配给某一节点专用
- 广泛应用于数字通信
- 例如: TDMA, CDMA, FDMA, OFDMA,
- 随机接入
- 轮转发送

信道划分: TDMA

TDMA: 时分多路复用

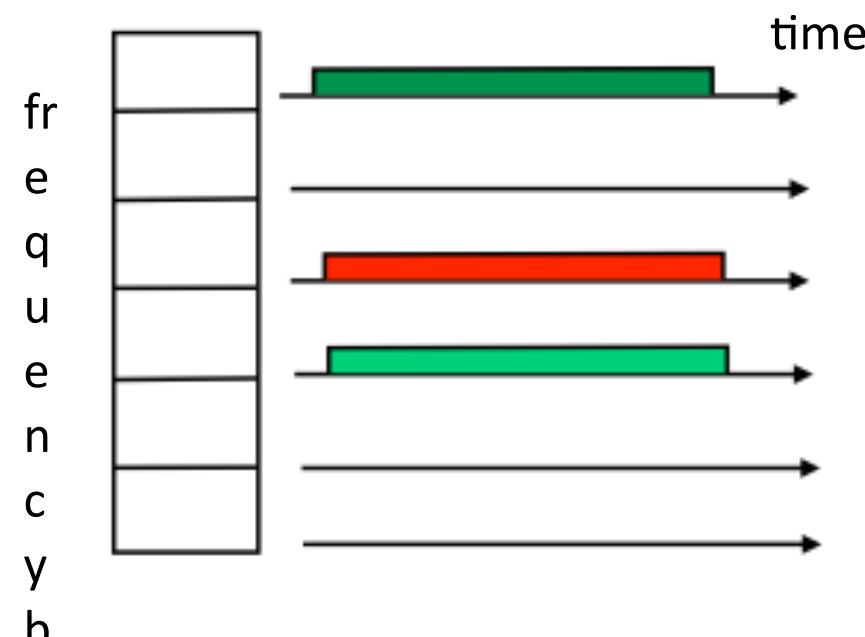
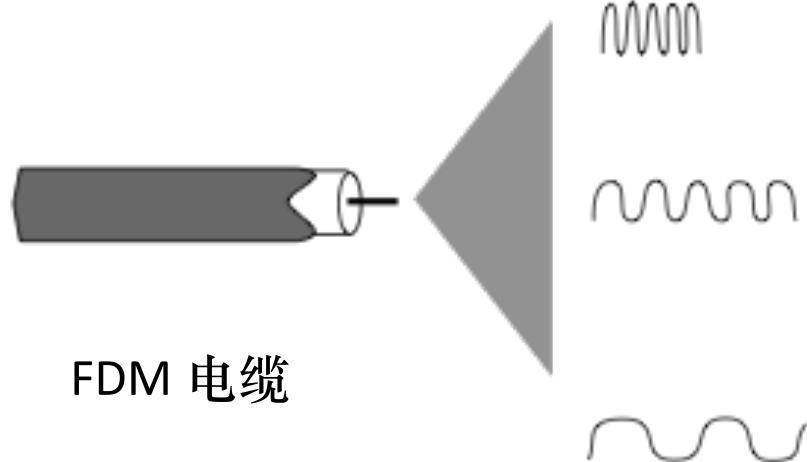
- “循环” 接入信道
- 在每次循环中每一个节点占用信道固定时长 (时隙, 长度 = 数据帧传输时延)
- 未使用的时隙保持空闲
- 示例: 6个站点的LAN, 时隙1,3,4 有帧发送, 时隙2,5,6空闲



信道划分: FDMA

FDMA: 频分多路复用

- 信道频谱划分为多个频带
- 每个站点分配固定的频带
- 频带未使用时为空闲
- 示例: 6个站点的LAN, 时隙1,3,4 有帧发送, 频带2,5,6空闲





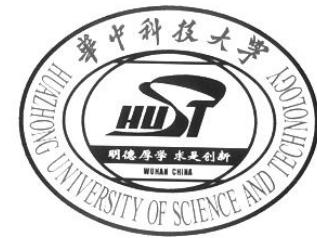
MAC 协议

- 静态信道划分

随机接入

- 不划分信道, 允许冲突发生
- “避免”冲突或冲突“恢复”
- 更适合于基于分组的数据通信
- 案例: Aloha, CSMA, ...

- 轮转发送

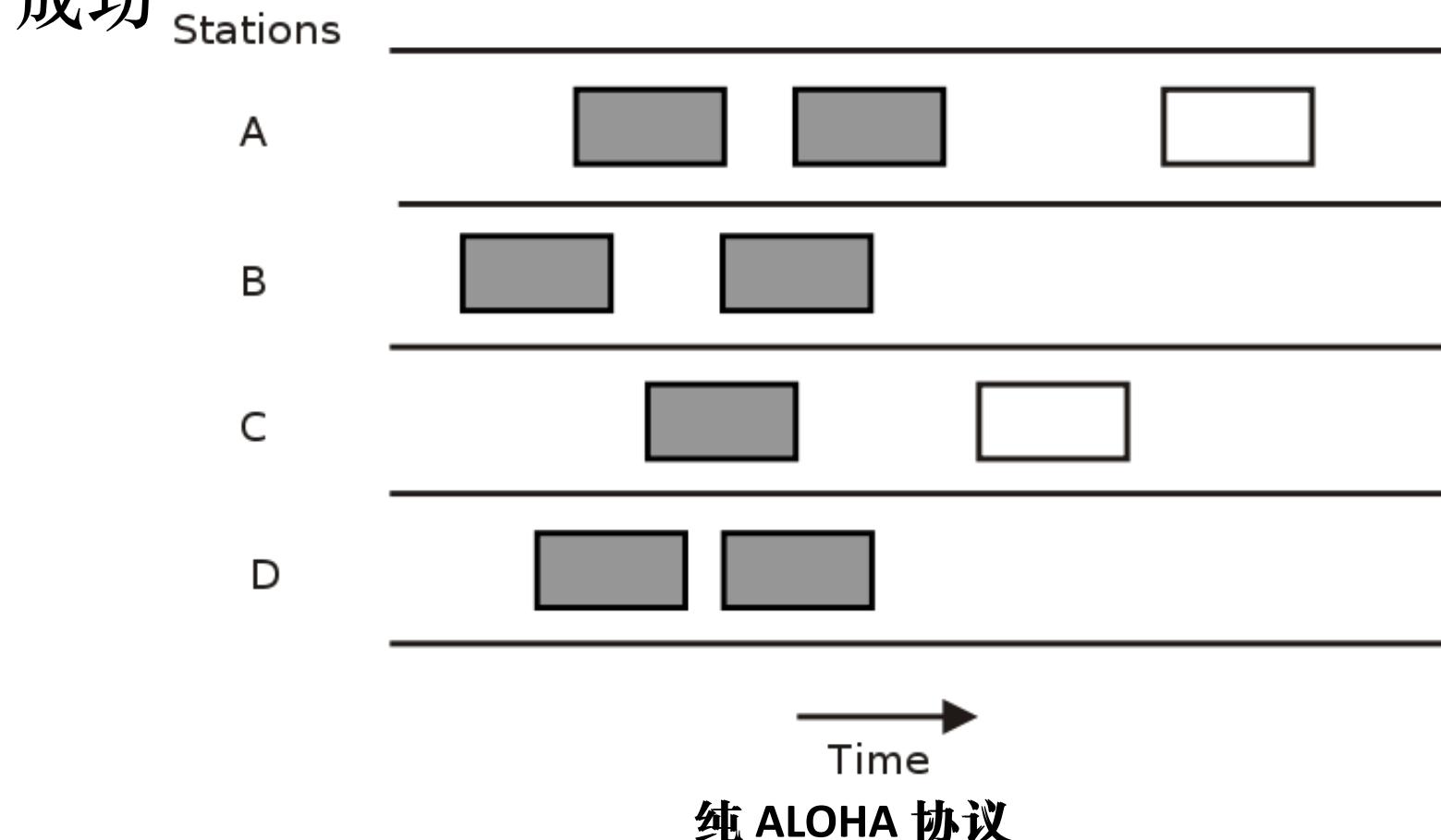


随机接入方式

- 当节点有数据发送时
- 以信道带宽R发送数据
- 节点之间不进行事前的协调
- 两个或多个节点同时发送数据 → “冲突”，
- 随机接入MAC协议：
 - 如何检测冲突
 - 发生冲突后如何恢复 (例如, 延迟重传)
- 随机接入MAC协议的实例：
 - ALOHA, 时隙ALOHA
 - CSMA, CSMA/CD, CSMA/CA

随机接入: 纯 ALOHA

- 纯Aloha: 简单
- 一旦数据帧到达, 立即发送
- 如果发生冲突, 节点等待随机时间后重发数据直到发送成功

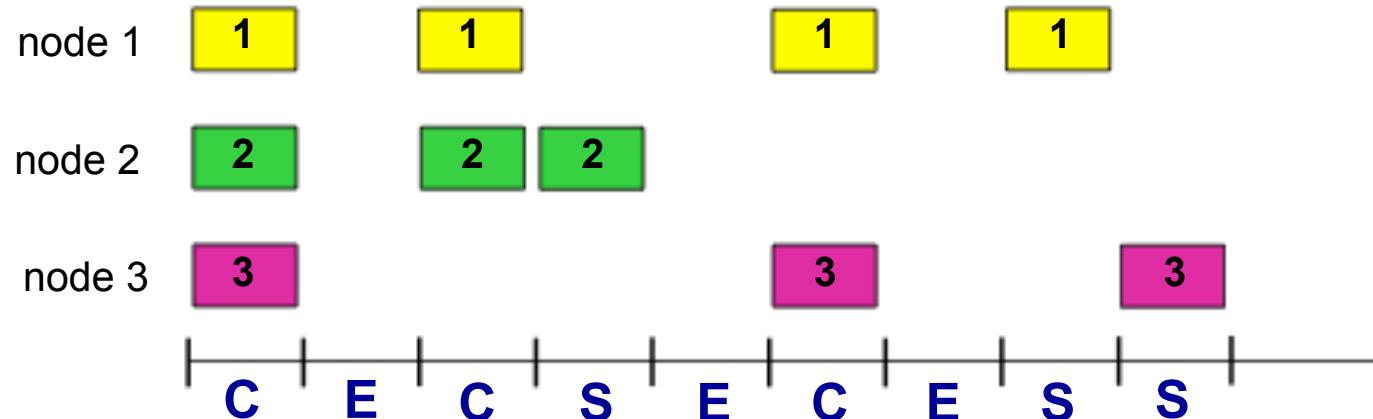




随机接入: 时隙ALOHA

- 假设:
 - 所有的数据帧长度相同
 - 时间轴被划分为等时长的时隙 (时长为发送1个数据帧的时延)
 - 节点只能在时隙到达时开始发送数据
 - 节点之间同步
 - 如果2个或多个节点在同一时隙内发送数据, 所有节点均能检测到冲突的发生
- 操作:
 - 当节点获得新的数据帧, 等待下一时隙到达开始发送
 - 如果不存在冲突: 节点可以在下一时隙到达发送新的数据帧
 - 如果发生冲突: 节点以概率 p 在每一个后续时隙内发送数据直到发送成功

随机接入: 时隙ALOHA



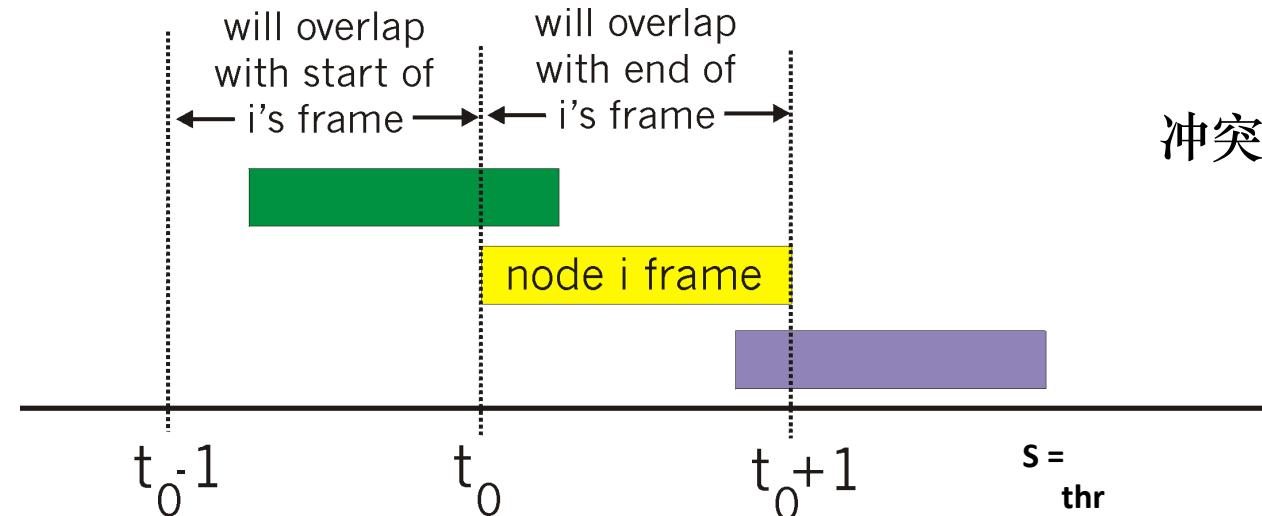
优点

- 一个活跃节点可以持续以全速(信道带宽)发送数据帧
- 高度去中心化: 不需要中央调度
- 简单

缺点

- 存在冲突, 浪费时隙
- 存在空闲时隙
- 节点可能花费较长的时间进行冲突监测与退避重传
- 要求时钟同步

随机接入: 纯 ALOHA 的性能



冲突概率增加:

T0时刻发送的数据帧会与
[t0-1,t0+1]时间范围内发送的
其他数据帧发生冲突

Prob(节点成功发送)

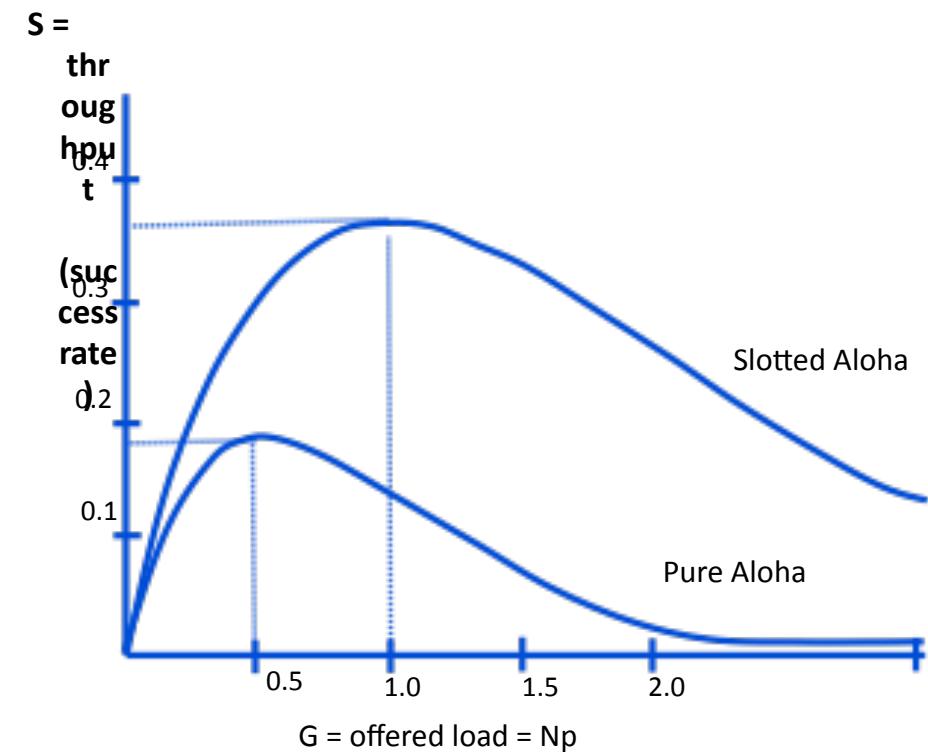
= Prob(节点发送) ×

Prob([t0-1,t0]时间范围内无其他节点发送) ×

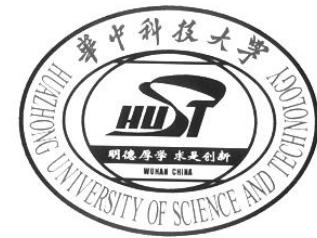
Prob([t0,t0+1]时间范围内无其他节点发送)

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$



... 选择最优的参数p , 设置n -> infinity ...



随机接入: 时隙 Aloha 的性能

效率: 成功发送的时隙数在整个发送过程所占的比例
(多个节点, 所有节点均有数据帧待发送)

- 假设: N 个节点有数据帧待发送, 并以概率 p 发送
- 一个时隙内特定节点成功发送的概率= $p(1-p)^{N-1}$
- 所有节点成功发送的概率= $Np(1-p)^{N-1}$
- 最大效率: 寻找 p^* 使得 $Np(1-p)^{N-1}$ 最大化
- 对多个节点而言, 当 N 取无穷大时, $Np^*(1-p^*)^{N-1}$ 达到极限值, 从而得到 效率最大值= $1/e = 0.37$

最大信道利用率: 信道只有37%的时间被用于有效数据帧传输!



随机接入: CSMA

- CSMA (载波监听多路访问)
 - 发送前监听信道
 - 如果信道空闲, 发送整个数据帧
 - 如果信道忙, 延迟发送
- 类比人类行为: 不打扰其他人!
 - 说话之前先听是否有人说话 → 载波监听
 - 如果有其他人同时说话, 停止讲话 → 冲突检测
- 问题
 - 如果所有的节点均执行载波监听, 为什么还会发生冲突?

随机接入: CSMA 冲突

冲突依然会发生:

传播时延导致两个节点无法监听到对方的发送

冲突:

整个数据帧传送时间被浪费

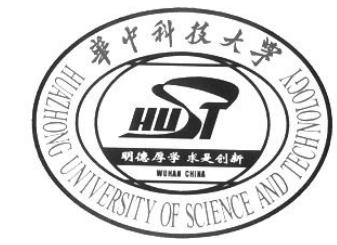
注意:

距离和传播时延对冲突检测概率的影响

该问题的解决方案(Ethernet 和 WiFi 中采用的CSMA的变种协议)将在后续内容中说明

节点的空间布局





随机接入: CSMA 冲突

冲突依然会发生:

传播时延导致两个节点无法监听到对方的发送

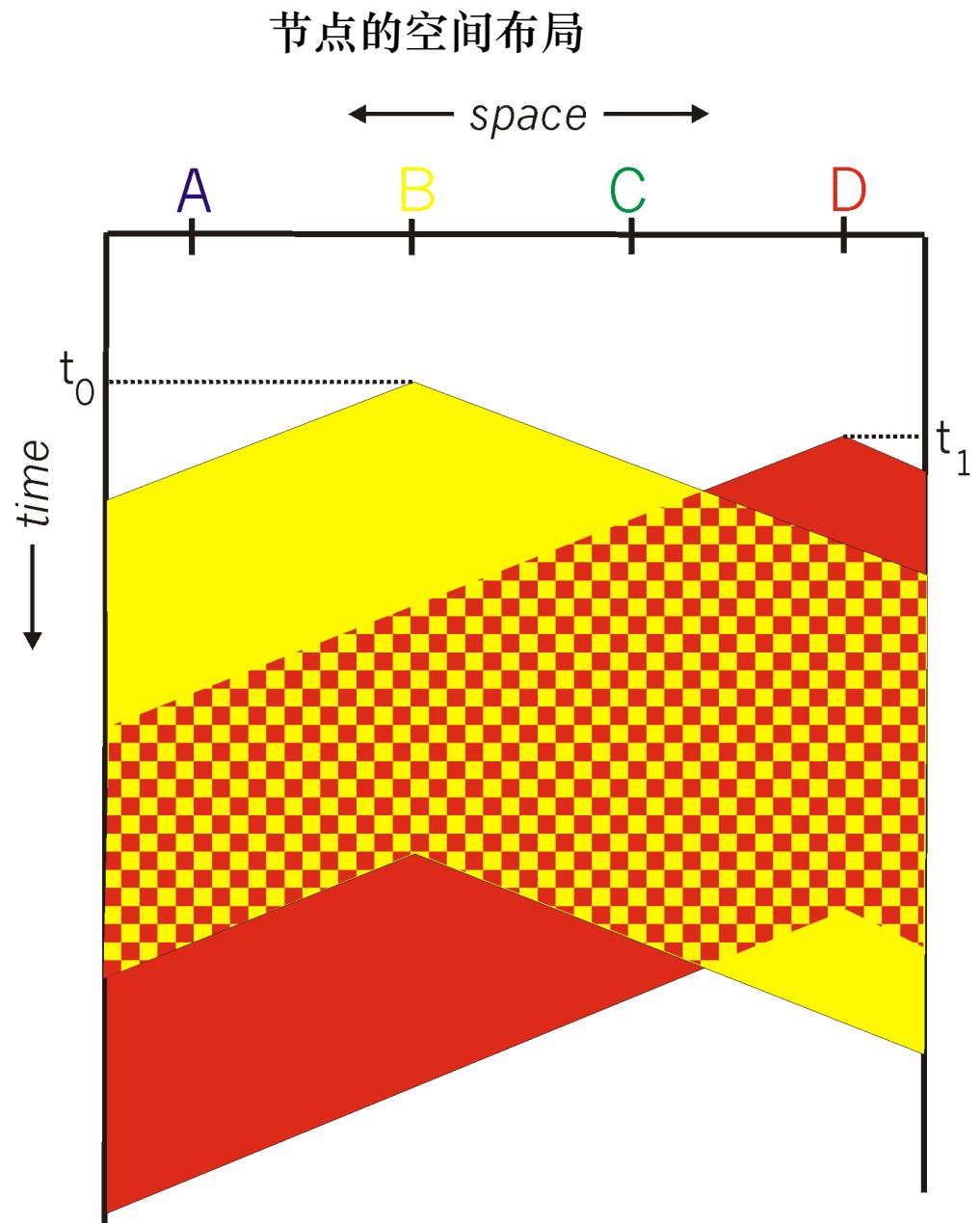
冲突:

整个数据帧传送时间被浪费

注意:

距离和传播时延对冲突检测概率的影响

该问题的解决方案(Ethernet 和 WiFi 中采用的CSMA的变种协议)将在后续内容中说明





MAC 协议

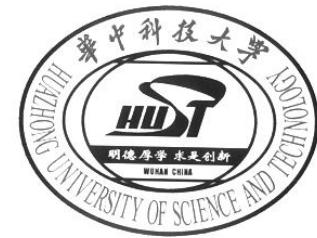
- 静态信道划分
- 随机接入

■ 轮转发送

- 节点轮流发送

- 待发送数据量大的节点占用信道更长时间

- 无冲突



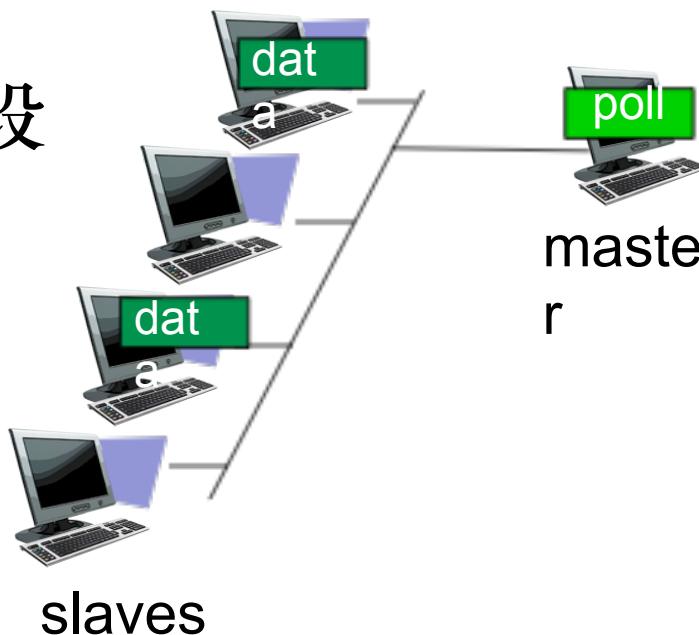
轮转发送MAC 协议

- 信道静态分配MAC协议:
 - 高负载情况下可有效且公平的共享信道
 - 低负载时效率低下: 信道接入时延(TDMA), 只有一个节点发送数据时只有 $1/N$ 的带宽被利用(FDMA)!
- 随机接入MAC协议
 - 低负载情况下效率较高: 单个节点可以利用整个信道资源
 - 高负载: 冲突开销很大
- 轮转发送协议
 - 不同负载条件下, 希望取得链路效率与通信开销的平衡

轮转发送MAC协议

轮询:

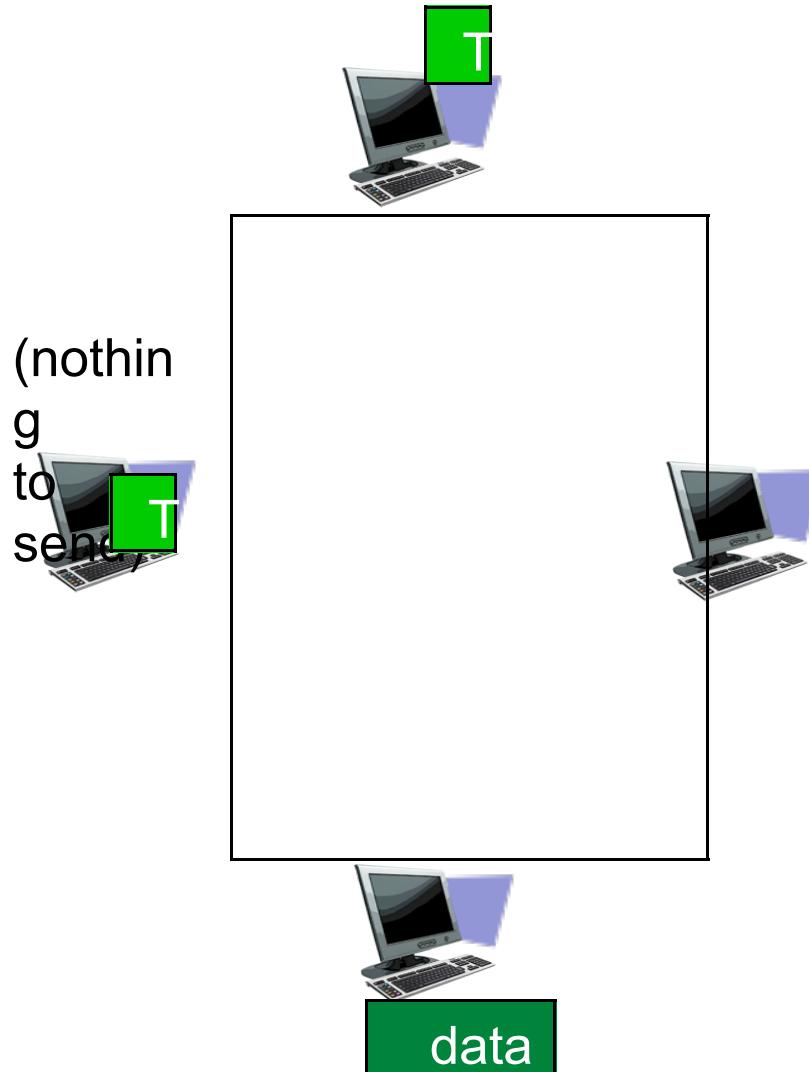
- 主节点“邀请”从节点轮流发送数据
- 典型应用于“dumb”从设备
- 关注点:
 - 轮询开销
 - 时延
 - 主节点失效



轮转发送MAC 协议

令牌传递：

- 控制令牌在节点之间依序传递。
- 令牌帧
- 关注点：
 - 令牌开销
 - 时延
 - 令牌丢失





小结:多路访问控制问题

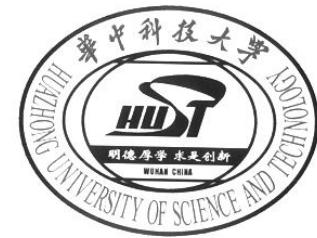
- 寻址
- 识别谁说话
- MAC(介质访问控制)协议的设计
- 信道静态分配
 - 时间, 频率 或 编码
 - 时分, 频分, 码分
- 随机接入(动态),
 - ALOHA, S-ALOHA, CSMA
- 轮转发送
 - 中央受控轮询, 令牌传递



提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结



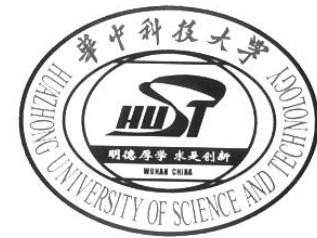


802.x 标准背景介绍

- IEEE, 电气和电子工程师学会(IEEE, I-Triple-E)
- IEEE 802 标准委员会
- 成立于Aug., 1979, 起初主要集中与局域网标准研究, 目前已扩展到LAN/MAN领域

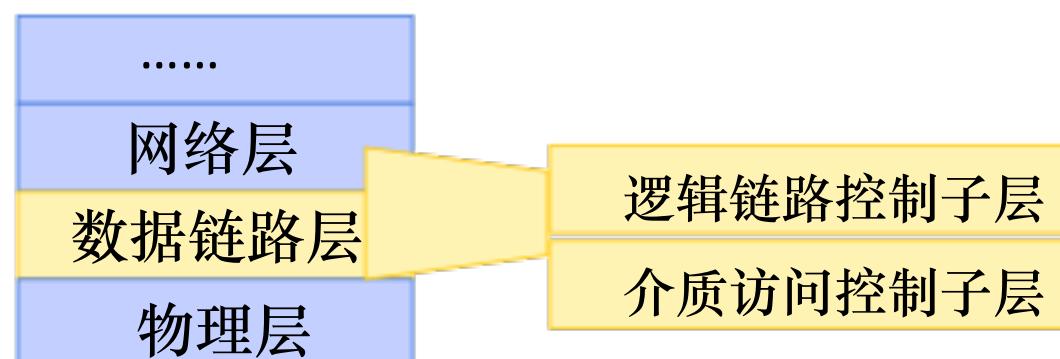


Standards Project Authorization	
1. <u>August 25, 1979</u> Date of Request	Project No <u>802</u> Approved: <u>3/13/80</u> Date For Standards Committee Use Only
2. <input checked="" type="checkbox"/> New Standard <input type="checkbox"/> Revision of _____ Standard No. _____	<input type="checkbox"/> Reaffirmation of _____ <input type="checkbox"/> Withdrawal of _____ Standard No. _____
3. Project Title: <u>Local network for Computer Interconnection.</u>	
4. Scope and Purpose of Proposed Standard: The proposed standard will apply to Data Processing devices which need to communicate with each other at a moderate data rate (1 M bit/sec) and within a local area (physical data path up to 4 km). The purpose of the proposed standard is to provide compatibility between devices of different manufacture so that the hardware and software customization necessary for effective data communication is minimized or eliminated.	
5. Sponsor: Computer Standards Technical Committee	Computer Society Society



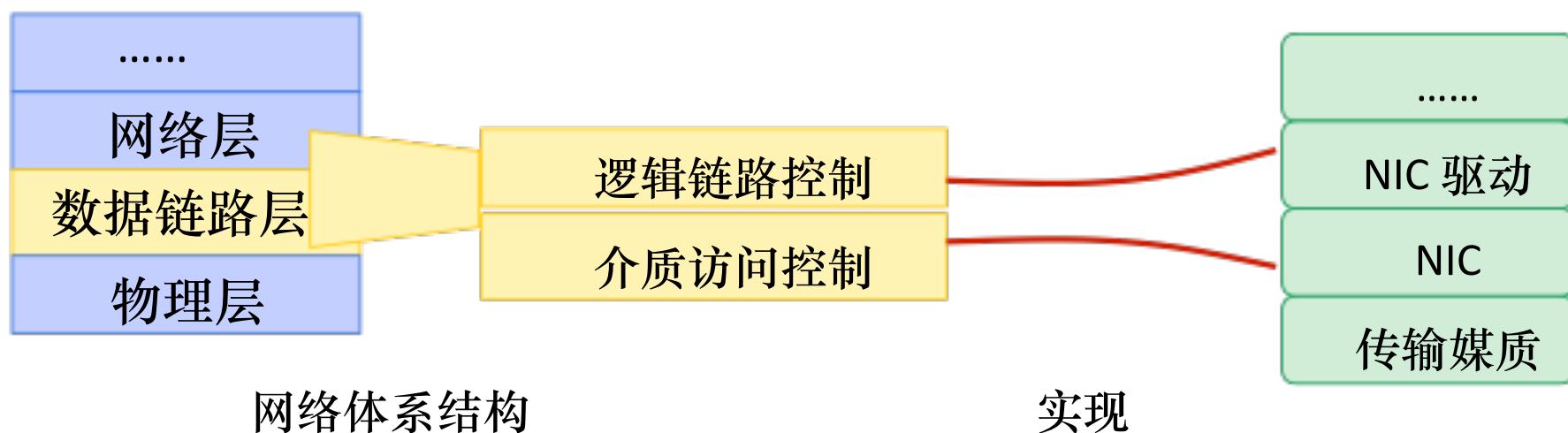
802.x 标准背景介绍

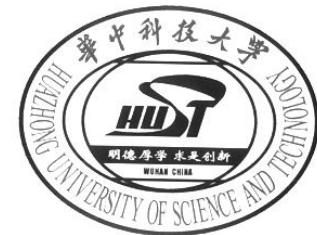
- 1980s, IEEE 802成立了多个不同的工作组分别研究不同的局域网技术:
- 802.3: 以太网, DIX (DEC, Intel, Xerox)
- 802.4: 令牌总线网, GM
- 802.5: 令牌环形网, IBM
- IEEE 802将数据链路层功能划分为两个子层
 - LLC (逻辑链路控制) 子层
 - MAC (介质访问控制) 子层



数据链路层的两个子层

- LLC (逻辑链路控制) 子层
 - 功能: 差错检测, 可靠传输
 - 驱动程序实现
- MAC (介质访问控制) 子层
 - 功能: 帧定界, 寻址, 多路访问控制
 - 与传输媒质特点和网络适配器的设计目前相关, NIC (网络接口卡) 硬件实现





802.x 标准近展(2010)

IEEE 802 ORGANIZATION

EXECUTIVE COMMITTEE (EC)

CHAIR
Paul Nikolich

WORKING GROUP/TAG CHAIRS

802.1
BRIDGING/ARCH
Tony Jeffree

802.3
CSMA/CD
David Law

802.11
WLAN
Bruce Kraemer

802.18 TAG
Radio Regulatory
Mike Lynch

802.15
WPAN
Bob Heile

802.16
BWA
Roger Marks

802.17
ResPackRing
John Lemon

802.19 TAG
Coexistence
Shellhammer

802.20
MBWA
Mark Klerer

802.21
Handoff
Vivek Gupta

802.22
WRAN
Wendong Hu

APPOINTED OFFICERS

1st VICE CHAIR
Mat Sherman

2nd VICE CHAIR
Pat Thaler

EXECUTIVE
SECY
Buzz Rigsbee

RECORDING
SECY
James Gilb

TREASURER
John
Hawkins

MEMBER
EMERITUS
Geoff
Thompson

HIBERNATION

802.2 LLC (Dave Carlson)
802.12 Demand Priority (Pat Thaler)

DISBANDED

802.4 Token Bus
802.7 Broadband TAG
802.9 ISLAM
802.14 CATV
802.6 DQDB
802.8 Fiber Optic TAG
802.10 Security
802.5 Token Ring

Emerg Svcs
ECSG
Geoff
Thompson



802.x 标准近展(2010)

IEEE 802 ORGANIZATION

802.2 (LLC): 不再被关注

802.4 (Token Bus):解散

802.5 (Token Ring):解散

802.6 (DQDB): 解散



802.3 (Ethernet): 幸免

802.11 (Wireless LAN):兴起

802.15 (Wireless PAN): 兴起



WORKING GROUP

802.11
BRIDGING
Tony Thaler

802.15
WP
Bob H

802.2
MBW
Mark K

RS

VICE CHAIR
Pat Thaler

CHAIR
SECY
James Gilb

MEMBER
EXTRUS
Geoff Thompson

Emerg Svcs
ECSG
Geoff
Thompson

HIBERNATION

802.2 LLC (Dave Carlson)
802.12 Demand Priority (Pat Thaler)

DISBANDED

802.4 Token Bus
802.7 Broadband TAG
802.9 ISLAN
802.14 CATV
802.6 DQDB
802.8 Fiber Optic TAG
802.10 Security
802.5 Token Ring

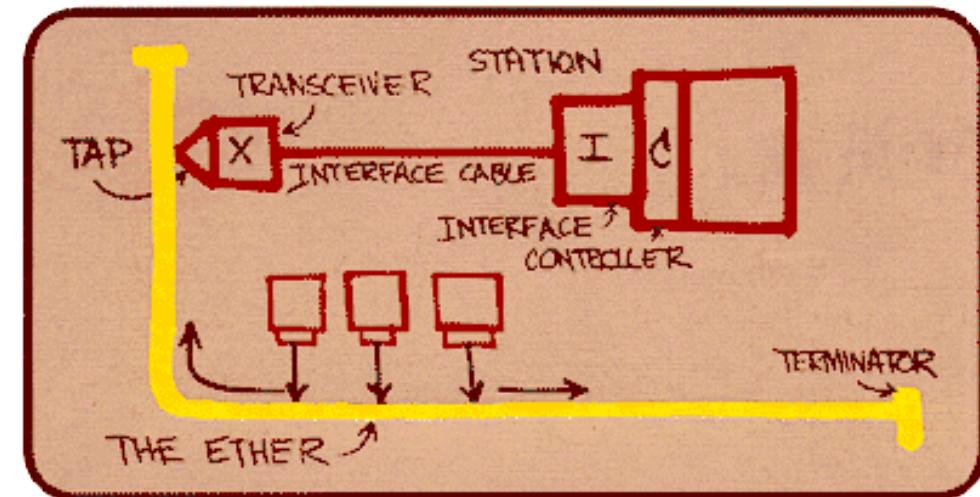
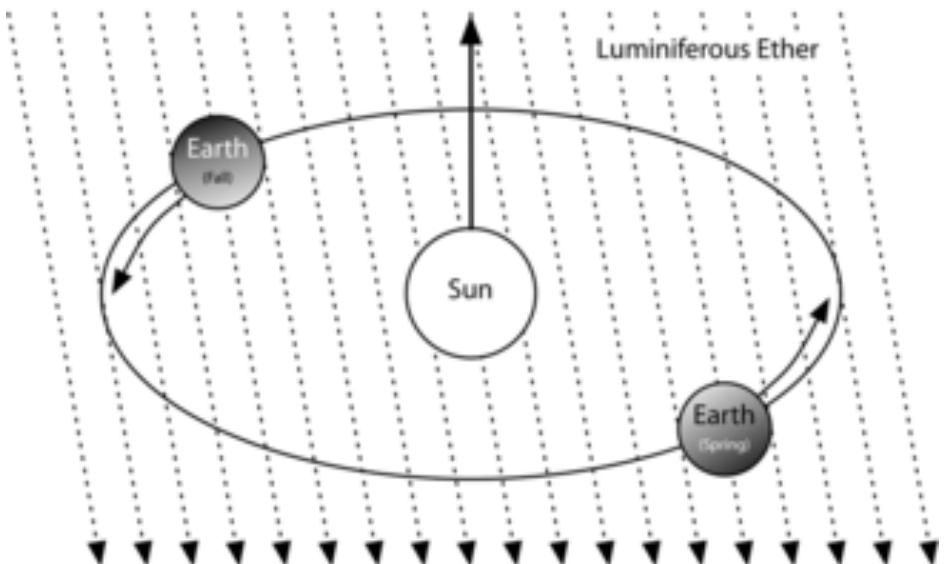


提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
 - 物理属性
 - 接入控制协议
 - 实际应用
- 无线网络
- 总结

物理属性

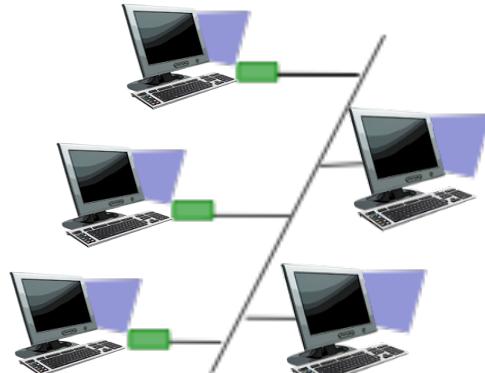
- Ether = ?
- 以太是希腊语，原意为上层的空气，指在天上的神所呼吸的空气
- 化学：醚
- 物理：以太
- 占据天体空间的物质。
- Ether in ``EtherNet''
- 用于描述共享媒质的某些共同特征



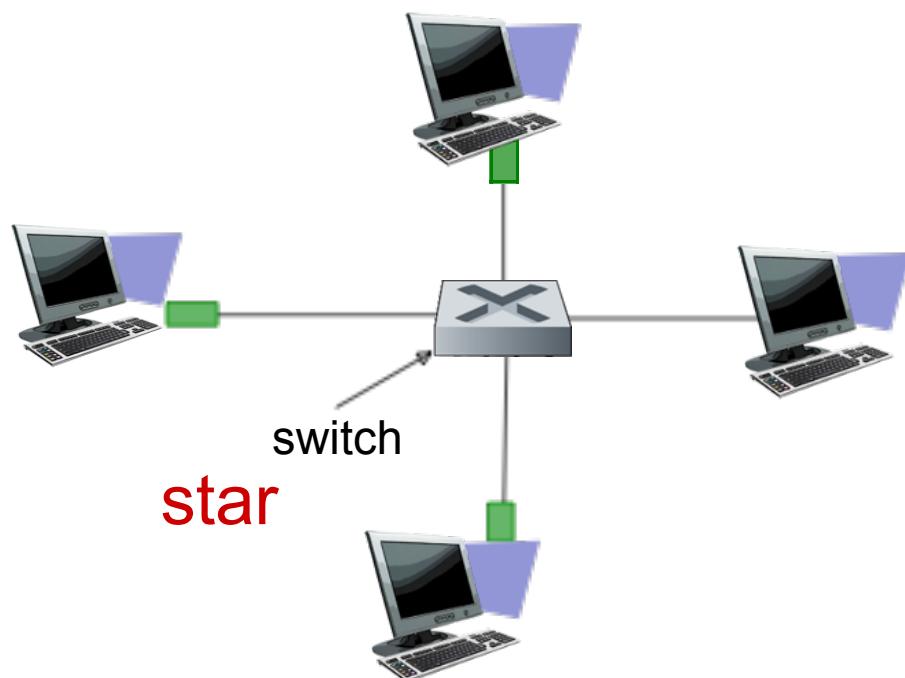
1973 年Bob Metcalfe提出的Ethernet 的草案

Ethernet: physical topology

- **bus:** popular through mid 90s
- all nodes in same collision domain (can collide with each other)
- **star:** prevails today
- active **switch** in center
- each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)



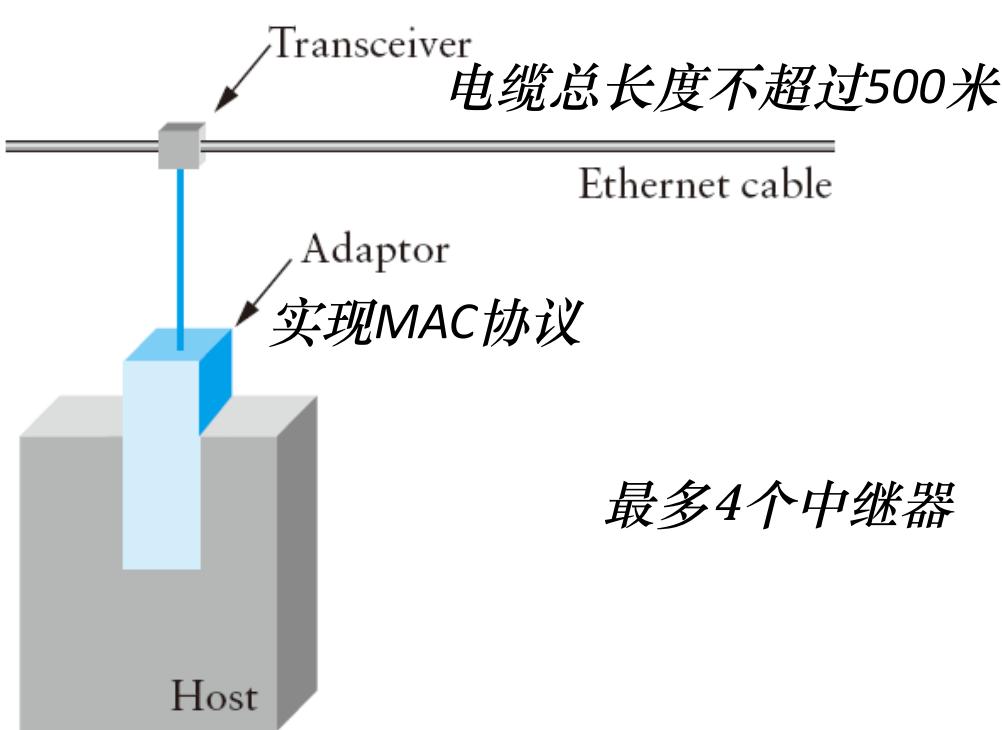
bus: coaxial cable



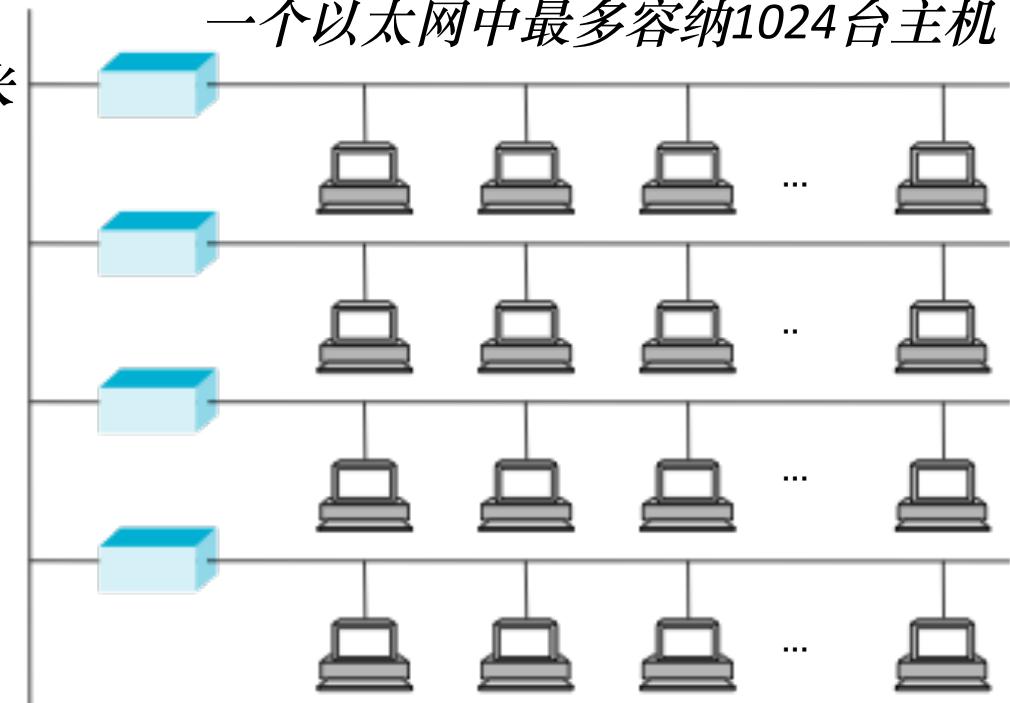
物理属性

- 早期Ethernet的设计采用总线拓扑结构

检测信道是否空闲，
收发信号

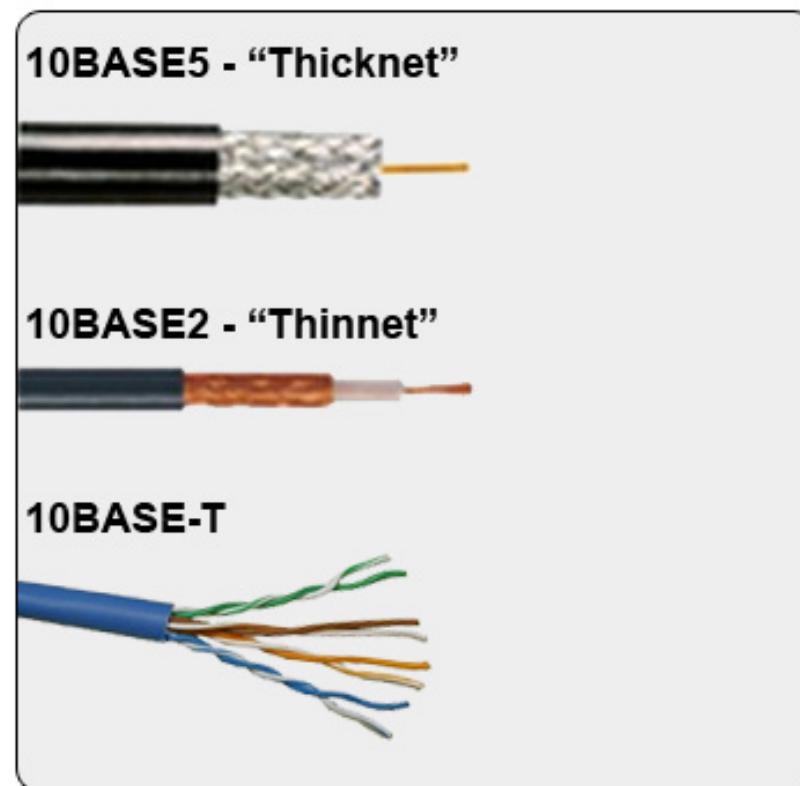


一个以太网中最多容纳1024台主机

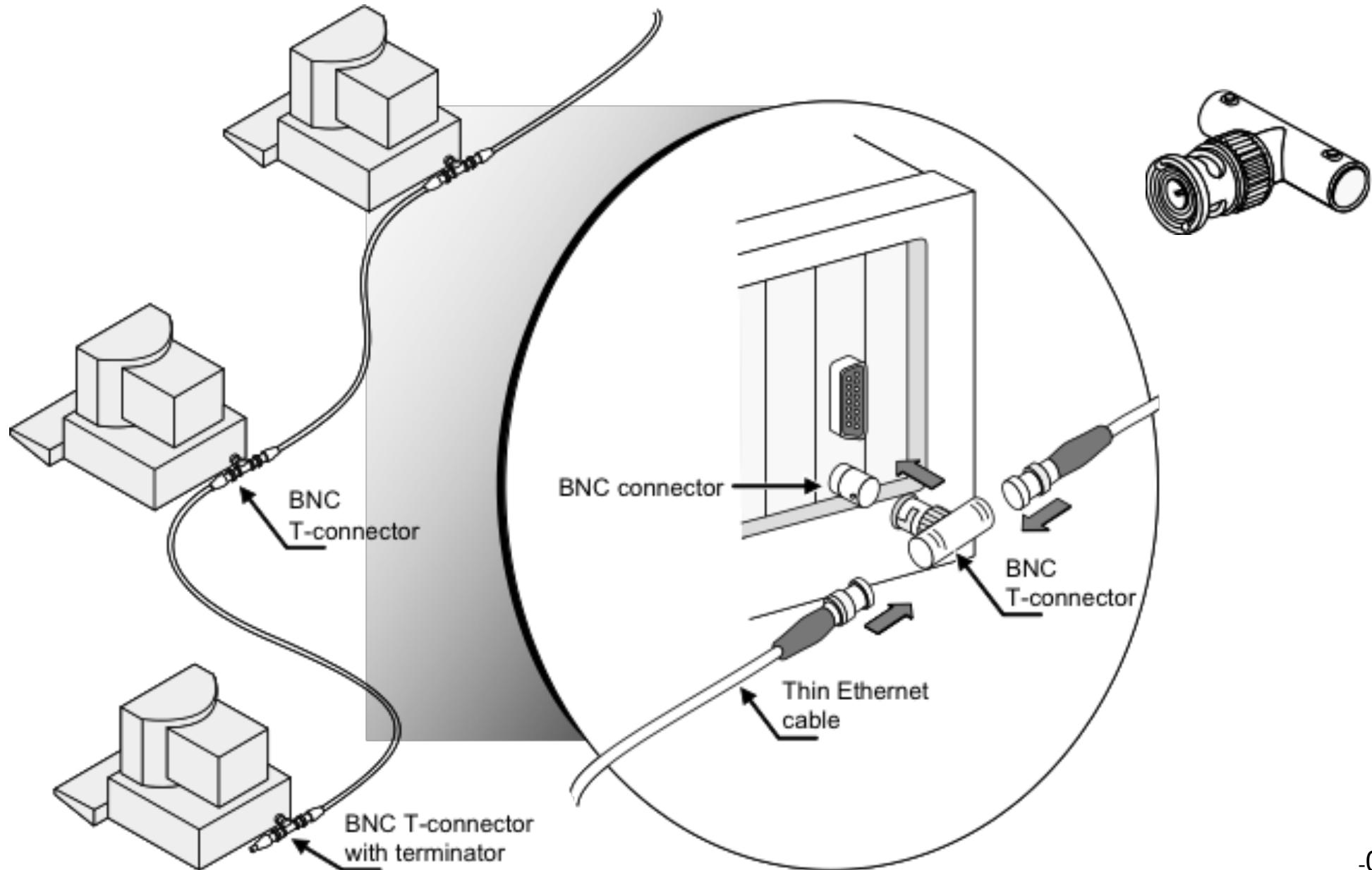


物理属性

- 收发器和传输媒质
- 10Base5 (粗缆)
- 10Base2 (细缆)
- 10BaseT/100BaseT (T:5类双绞线)

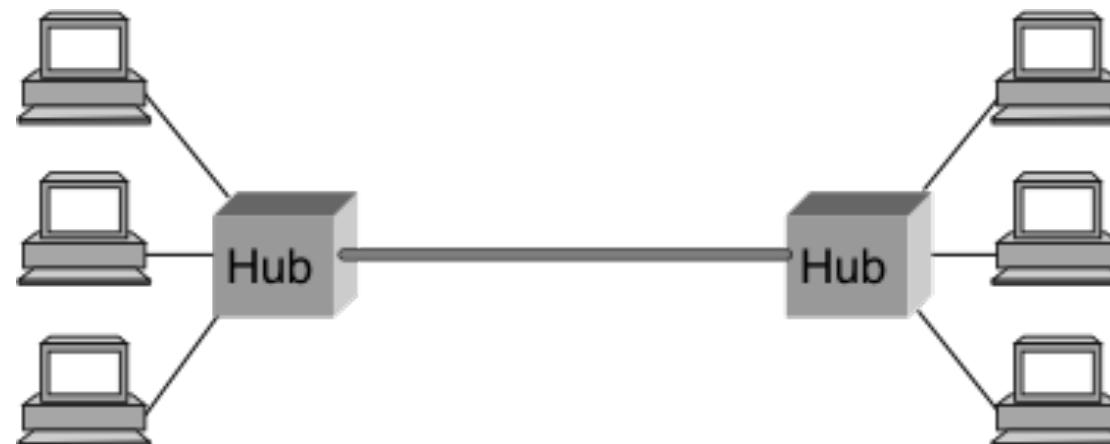


总线拓扑结构: 传统Ethernet



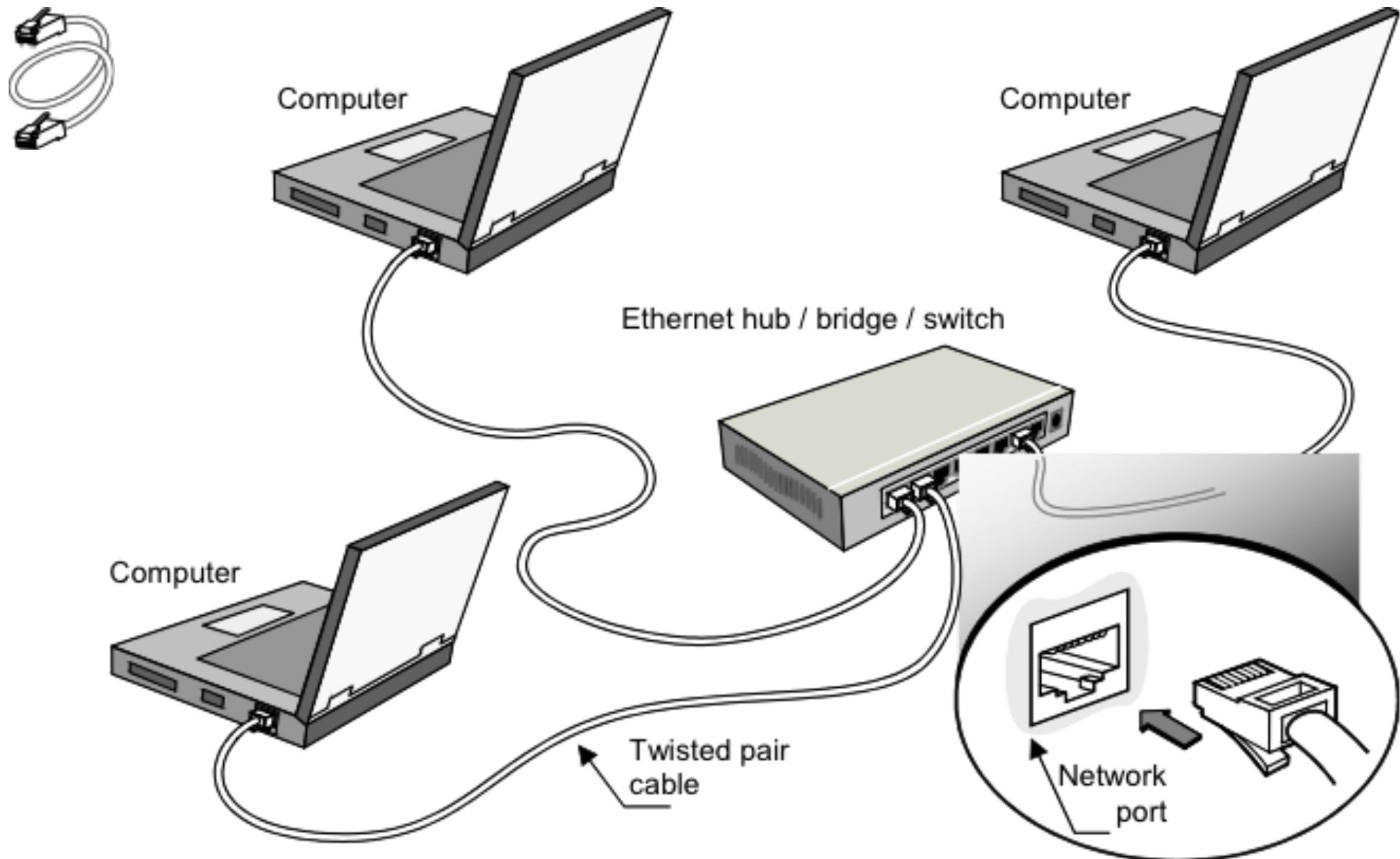
星型拓扑结构: 集线器/交换设备

- 现行的Ethernet, 常采用星型拓扑结构
- 在总线型拓扑结构中: 所有节点在同一个冲突域内, 相互之间可能产生冲突
- 星型拓扑结构
 - 基于集线器: 所有节点在一个冲突域内
 - 基于交换机: 每一个“spoke”执行独立的MAC协议, 节点之间无冲突

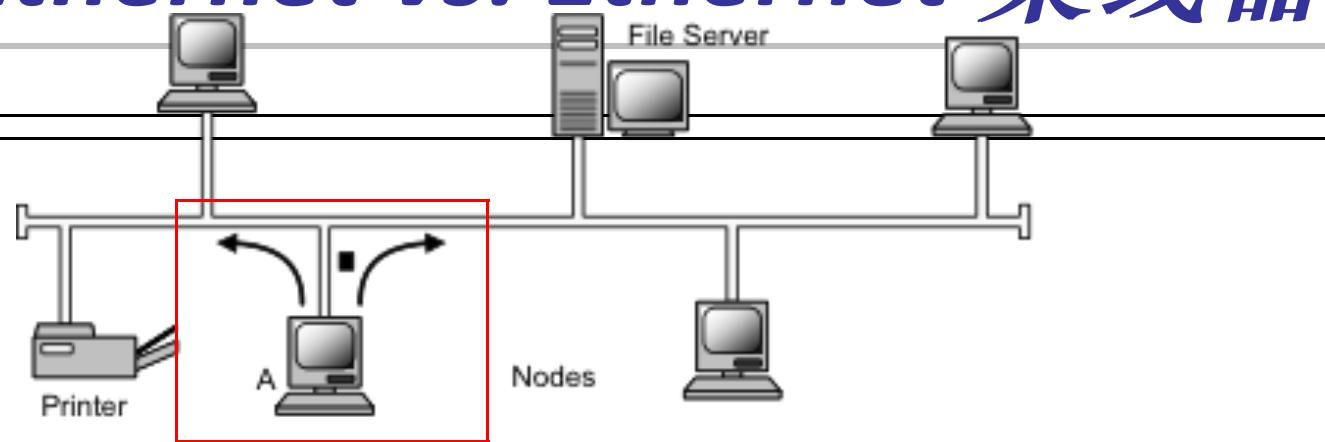


通过集线器和双绞线连接

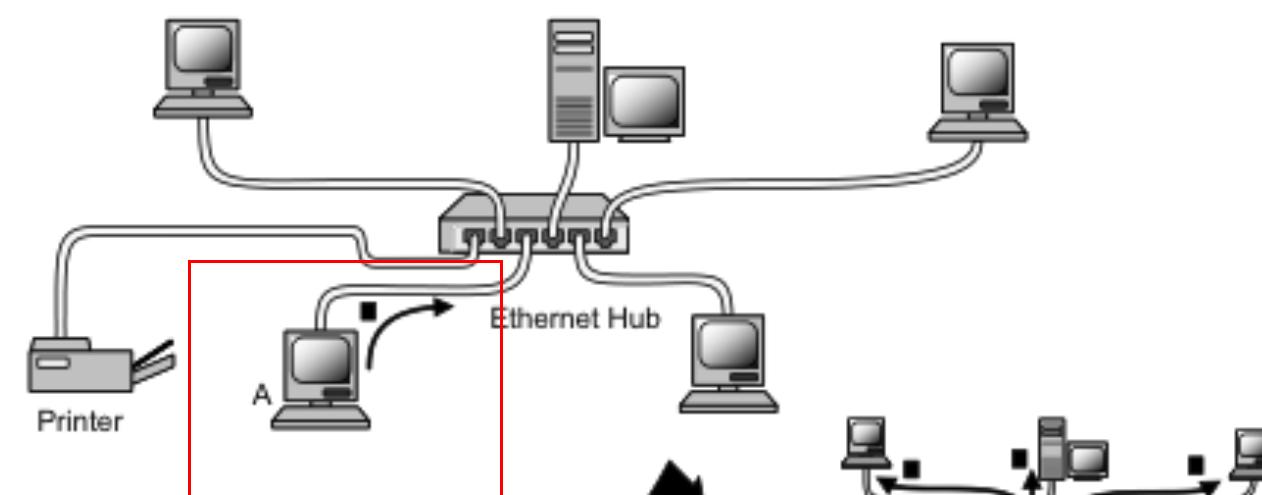
星型拓扑结构: 集线器/交换设备



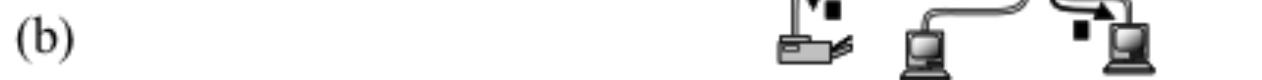
传统Ethernet vs. Ethernet 集线器



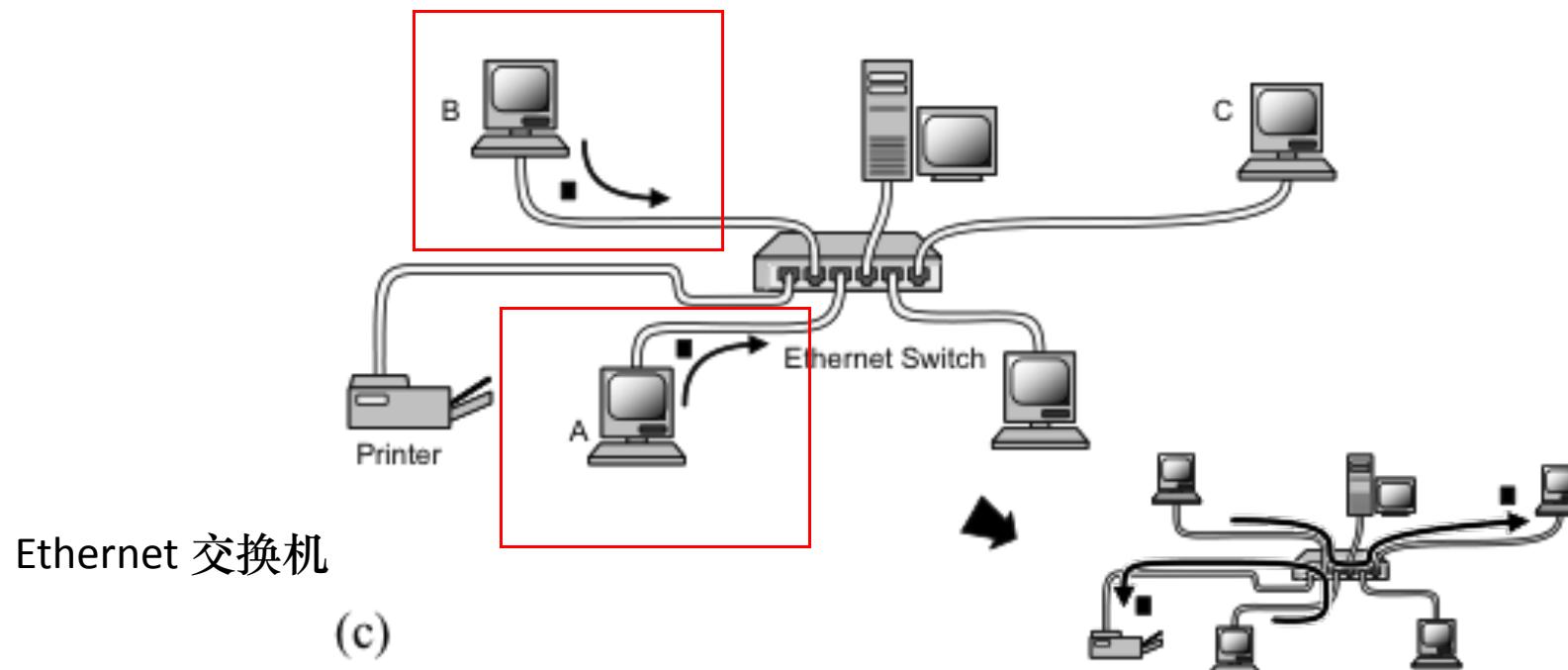
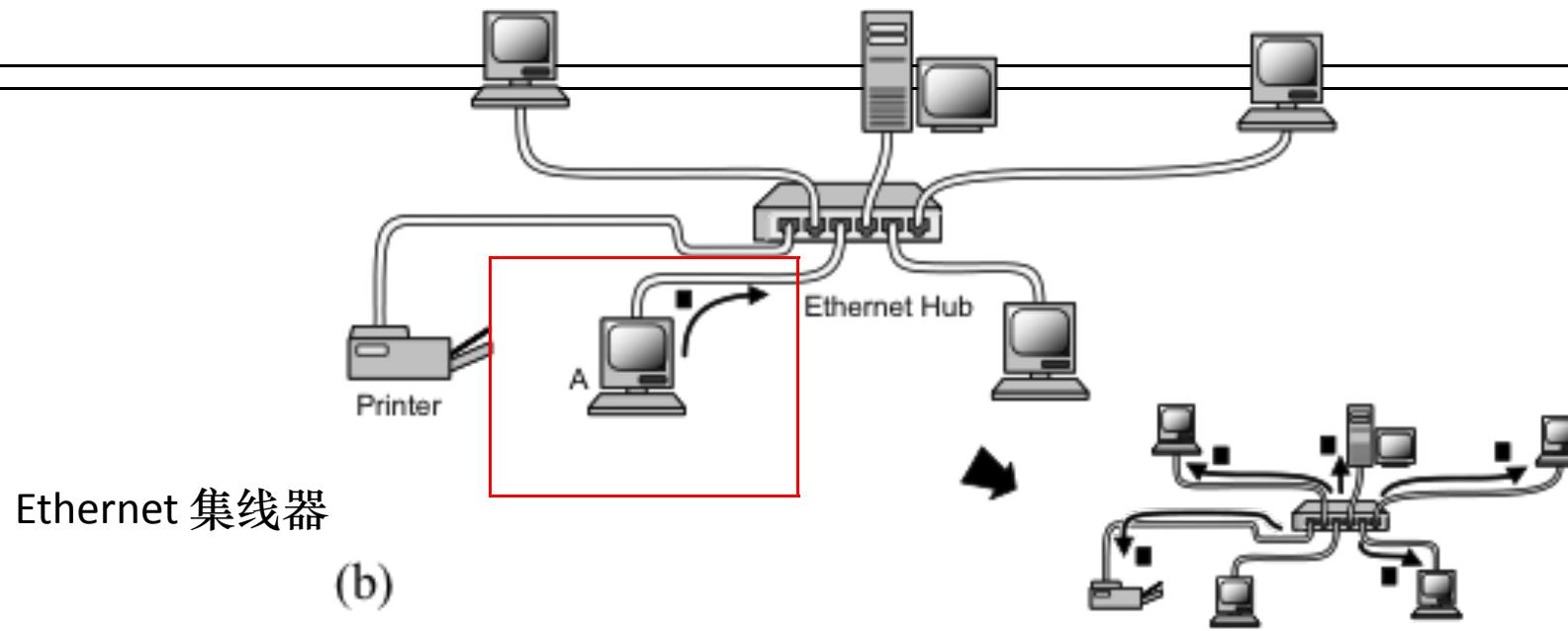
细缆Ethernet



Ethernet 集线器



Ethernet 集线器vs. Ethernet 交换机



Ethernet 集线器vs. Ethernet 交换机

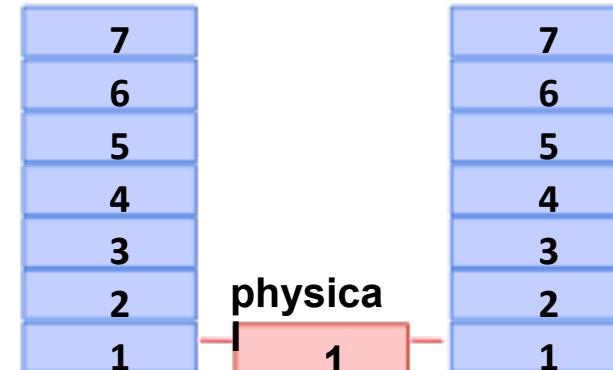
- Ethernet 集线器
- 层1设备
- 简单的信号中继器
- 为节点提供共享链路



100baseT hub



10baseT hub



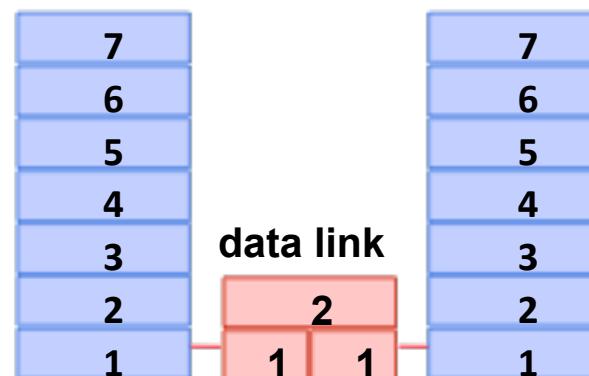
- Ethernet Switch
- 层2设备
- 识别数据帧的地址, 完成数据帧的存储转发
- 为节点提供独立的链接

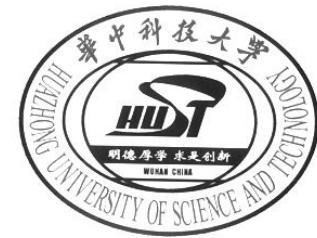


Ethernet Switch



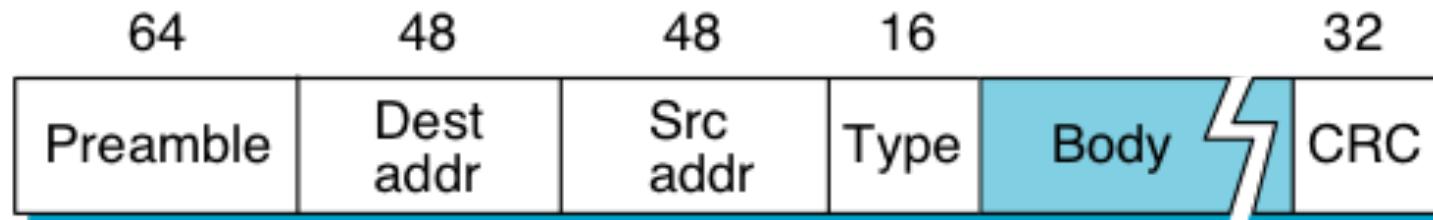
Ethernet/ATM Switch





介质访问控制协议

帧定界和差错检测:



- 帧定界
- 面向比特, 数据字段大小为46 – 1500字节
- 前导码: 同步, 7个连续的 10101010位串, 后续1个 10101011位串
- 寻址: 目的地地址和源地址, 占48 比特
- 差错检测
- CRC: 接收方检测
- 如果检测出错误, 丢弃数据帧



介质访问控制协议

寻址:

- 地址
 - 为每一个网络适配器分配一个全球唯一的48比特单播地址
 - 示例: 8:0:2B:E4:B1:2
 - 广播: 全1
 - 多播: 第一个比特为 1
- 地址操作
 - 网络适配器收到所有的数据帧, 但仅接收处理目的地址为本机地址的数据帧:
 - 目的地址为本机的单播地址
 - 广播地址
 - 多播地址(要求支持多播)
 - 混杂模式

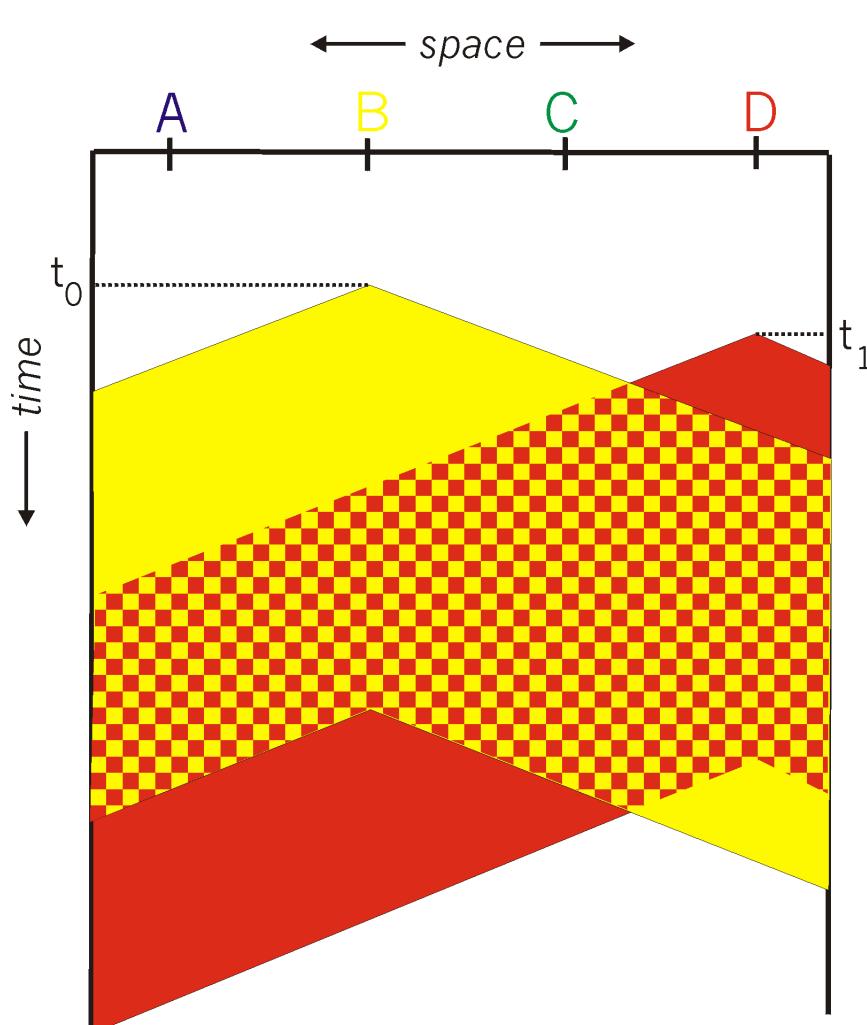


介质访问控制协议

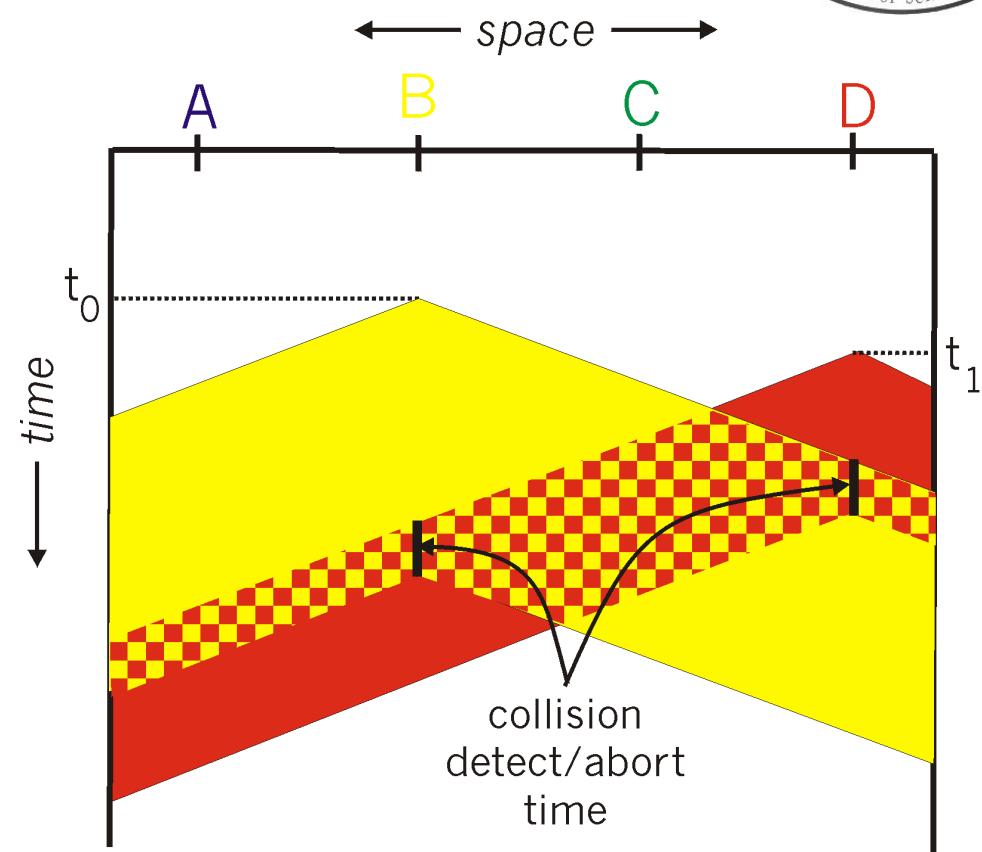
多路访问控制算法:

- 带有冲突检测的载波监听多路访问控制 (CSMA/CD)
- 载波监听
 - 所有节点可以区分信道是否空闲
 - 发送前监听信道, 且不影响现有通信
 - 监测是否已有其他人正在发送数据
 - ... 等待直到对方发送完毕
- 冲突监测
 - 节点可以检测数据帧发送过程是否发生冲突
 - 如果其他人同时开始发送, 则停止发送行为

CSMA/CD 冲突监测



Original CSMA



CSMA/CD



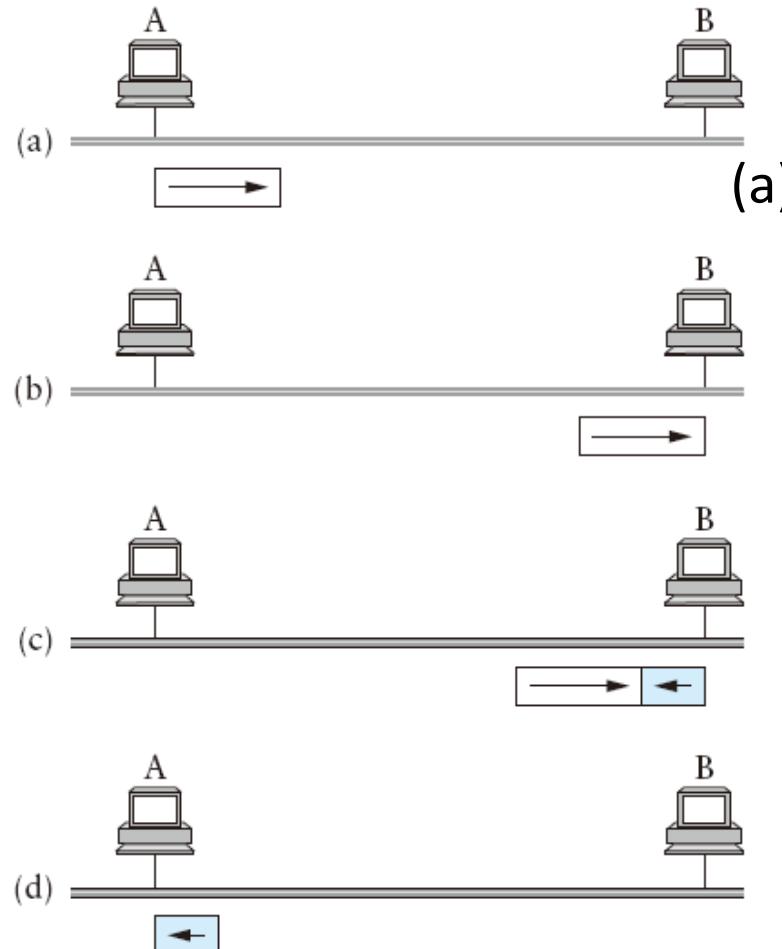
Ethernet CSMA/CD 协议

- 1. NIC 接收来自于网络层的数据, 构造数据帧
- 2. NIC 监听信道:如果发现信道空闲, 则开始发送数据帧. 如果信道忙, 则等待直到信道空闲再发送数据帧
- 3. 如果整个数据帧的发送过程未检测到其他数据发送过程, 则完成数据帧的发送!
- 4. 如果在数据帧发送过程中监测到其他数据发送过程, 则退出并发送**干扰序列 (jamming)**
 - 通知所有其他节点冲突的发生
- 5. 退出数据帧发送后, NIC执行**二进制退避算法**: 发生第 m 次冲突后, NIC 从中 $\{0,1,2,\dots,2m-1\}$ 随机选择 K . NIC 等待 $K \cdot 512$ 比特时延后, 重新执行第2步
 - 采用尝试重传来估计当前的负载
 - 第一次冲突: 从 $\{0,1\}$ 中选择 K ; 等待时延为 $K \cdot 512$ 比特传输时延
 - 第二次冲突:从 $\{0,1,2,3\}$ 中选择 K ...

改进CSMA随机退避机制 Xinghua Sun and Lin Dai, "Backoff Design for IEEE 802.11 DCF Networks: Fundamental Tradeoff and Design Criterion, " IEEE/ACM ToN, 2015

Ethernet CSMA/CD 协议

Why 512 bits?



最坏的情况:

(a) A 在 t 时刻发送数据帧;

(b) A 发送的数据帧在 $t + d$ 时
刻到达 B;

(c) B 在 $t + d$ 时刻开始发送数据
帧, 则会与A发送的数据帧
发生冲突;

(d) B的残缺帧在 $t + 2d$ 时刻才
能到达A.

主机A必须传输 $2d$ 的实现才能确保检测到所有的冲突.

考虑初始设计中, 最大覆盖范围的以太网(2500米)的往返时延
为51.2us.

在10Mbps的以太网中, 最短帧长为 $10 \text{ Mbps} \times 51.2 \text{ us} = 512 \text{ bits (64 B)}$

CSMA/CD 退避算法示例

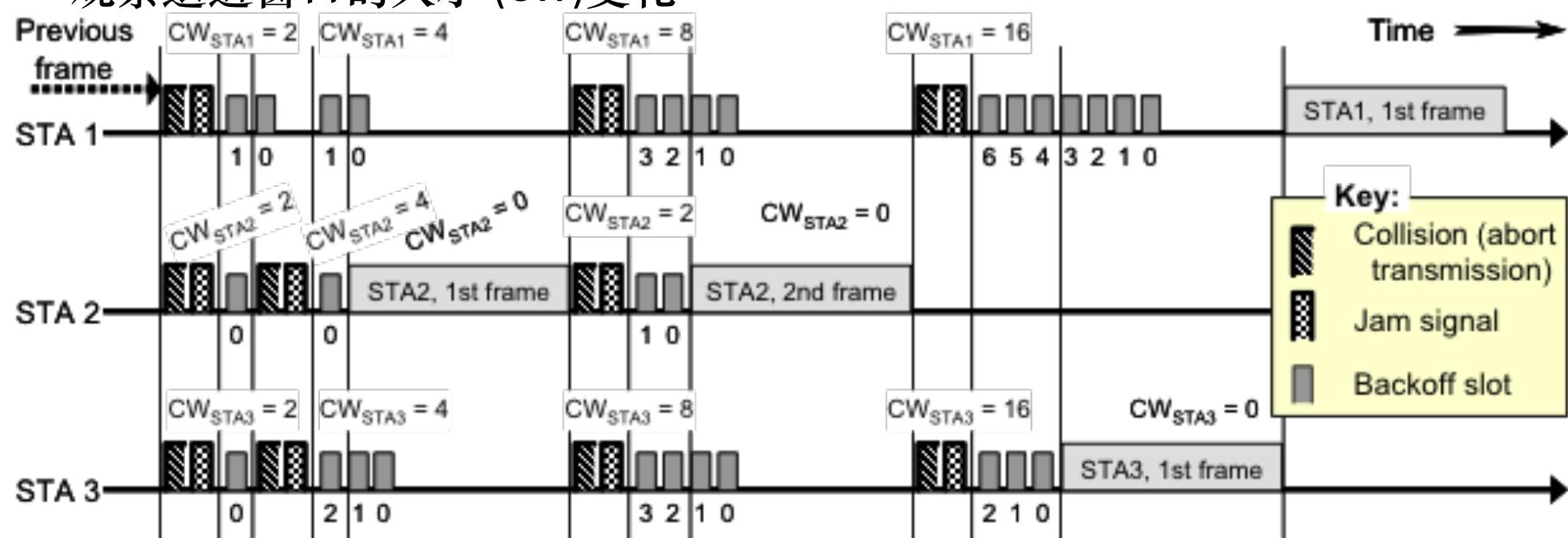
STA1和STA3有一个数据帧待发送, STA2有两个待发送数据帧
 假设所有数据帧长度相同

第一次冲突各站点的退避值 : STA1 = 1; STA2 = 0; STA3=0

第二次冲突各站点的退避值: STA1 = 1; STA2 = 0; STA3=2

第三次冲突各站点的退避值: STA1 = 3; STA2 = 1; STA3=3

观察退避窗口的大小 (CW)变化



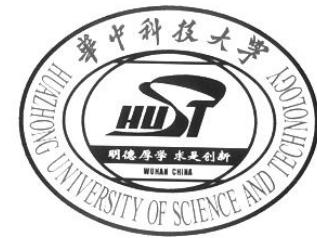


CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$\text{efficiency} = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to 1
 - as t_{prop} goes to 0
 - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!



Ethernet的实际应用

- “占主导地位”的有线局域网技术:
- NIC的价格: \$20
- 第一个广泛使用的局域网技术
- 通信带宽: 10 Mbps – 10 Gbps
- 实际规模
 - 大部分以太网上连接的主机数比200少, 远小于最大值 1024
 - 大部分以太网的覆盖范围远小于2500米, 往返时延接近 $5 \mu\text{s}$ 而不是 $51.2 \mu\text{s}$.
- 成功的原因
 - 以太网易于**维护和管理**, 易于新的节点加入.
 - **成本低廉**: 电缆价格便宜, 已成为每台主机标配网络适配器.

技术与经济：规模效应

Ethernet的最新进展

- 2002, 万兆以太网(10 Gigabit)
- 802.3ae, 10GbE
- 放弃支持双绞线UTP, 只支持光纤
- 放弃支持CSMA/CD, 只支持全双工
- 仍然兼容802.3数据帧结构

**成为城域网主要技术
以太网用来构建端到端链接!**



2010, 100Gbps 以太网标准

2014, 华为和Xilinx合作完成400Gbps交换机原型机

2013年华为成为全球最大电信设备供应商！

<http://www.miit.gov.cn/n1146290/n1146402/n1146455/c3230624/content.html>



Ethernet的成功之道

KISS

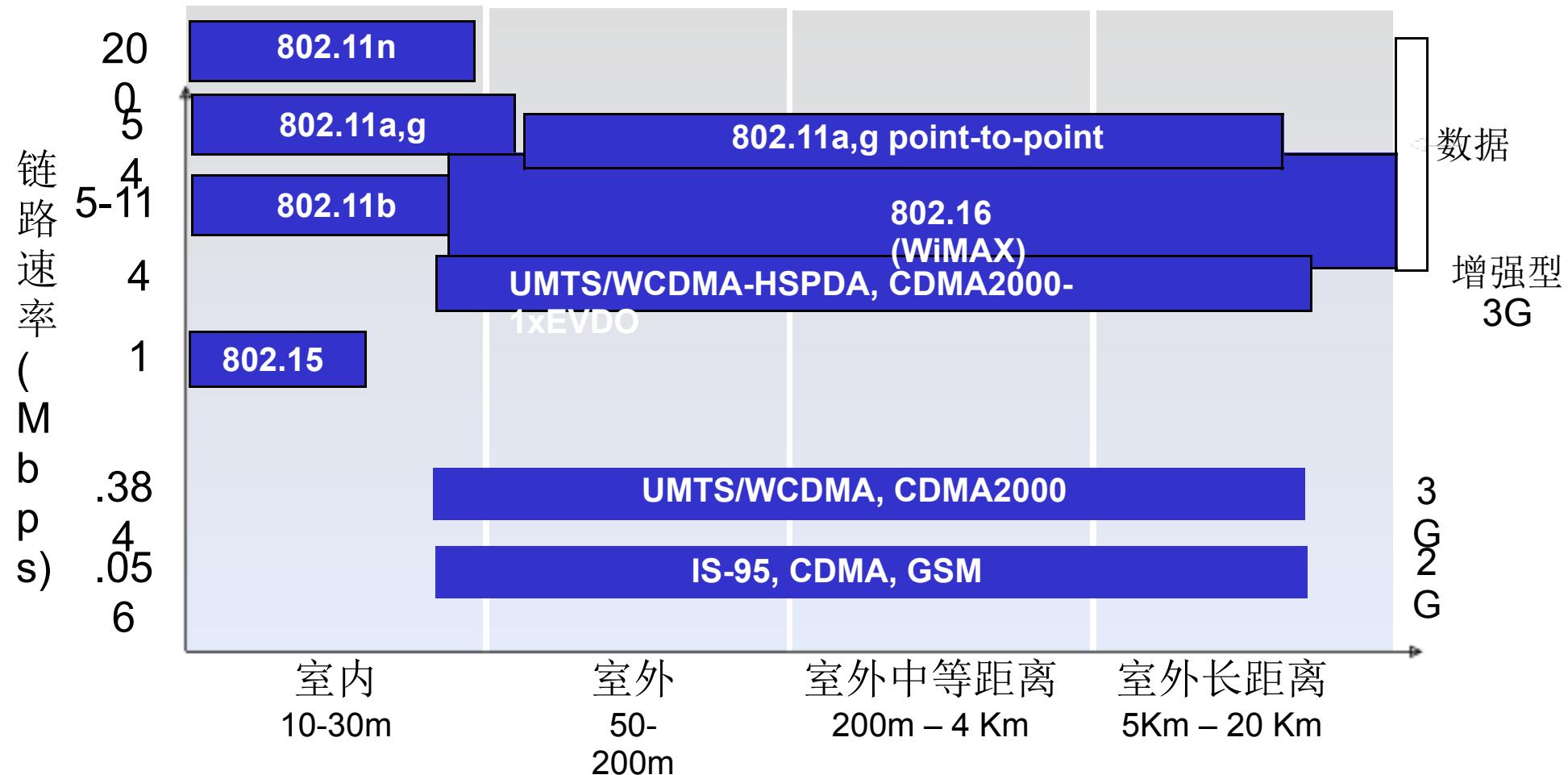
Keep It Simple, Stupid!

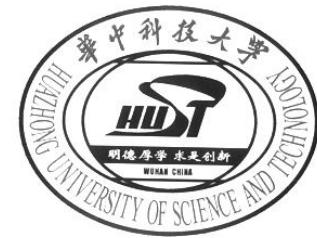


提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结

几种无线网络标准的链路特性





无线链路特性 (1)

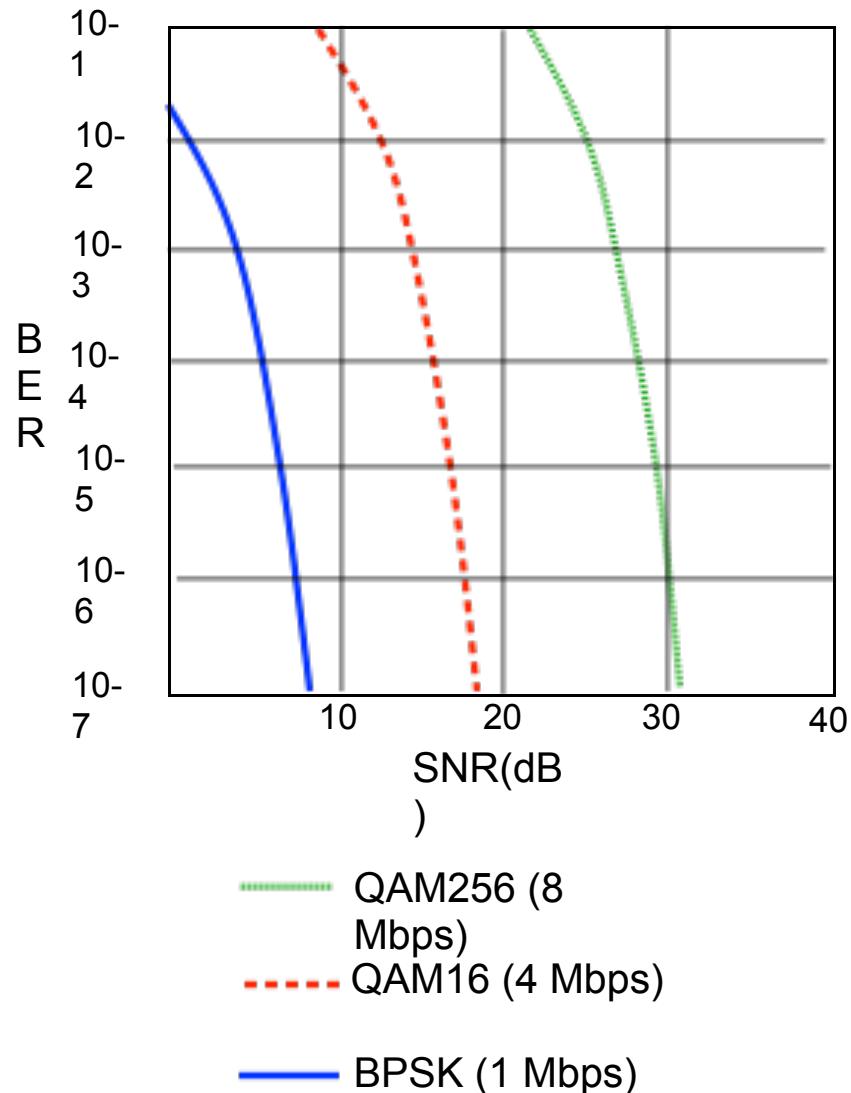
与有线链路的区别:

- **信号强度递减:** 电磁波在穿越物体时强度将减弱 (路径损耗)
- **存在来自于其他信号源的干扰:** 其他设备 (如手机) 也使用同一个无线频段(如2.4 GHz)发送信号; 环境中其他设备 (如电动机) 也能形成干扰
- **多径传播:** 电磁波的一部分受物体和地面反射, 在不同的时间到达接收端

以上这些因素使得无线链路中的通信 (即使是点到点的通信) 也变得更加困难

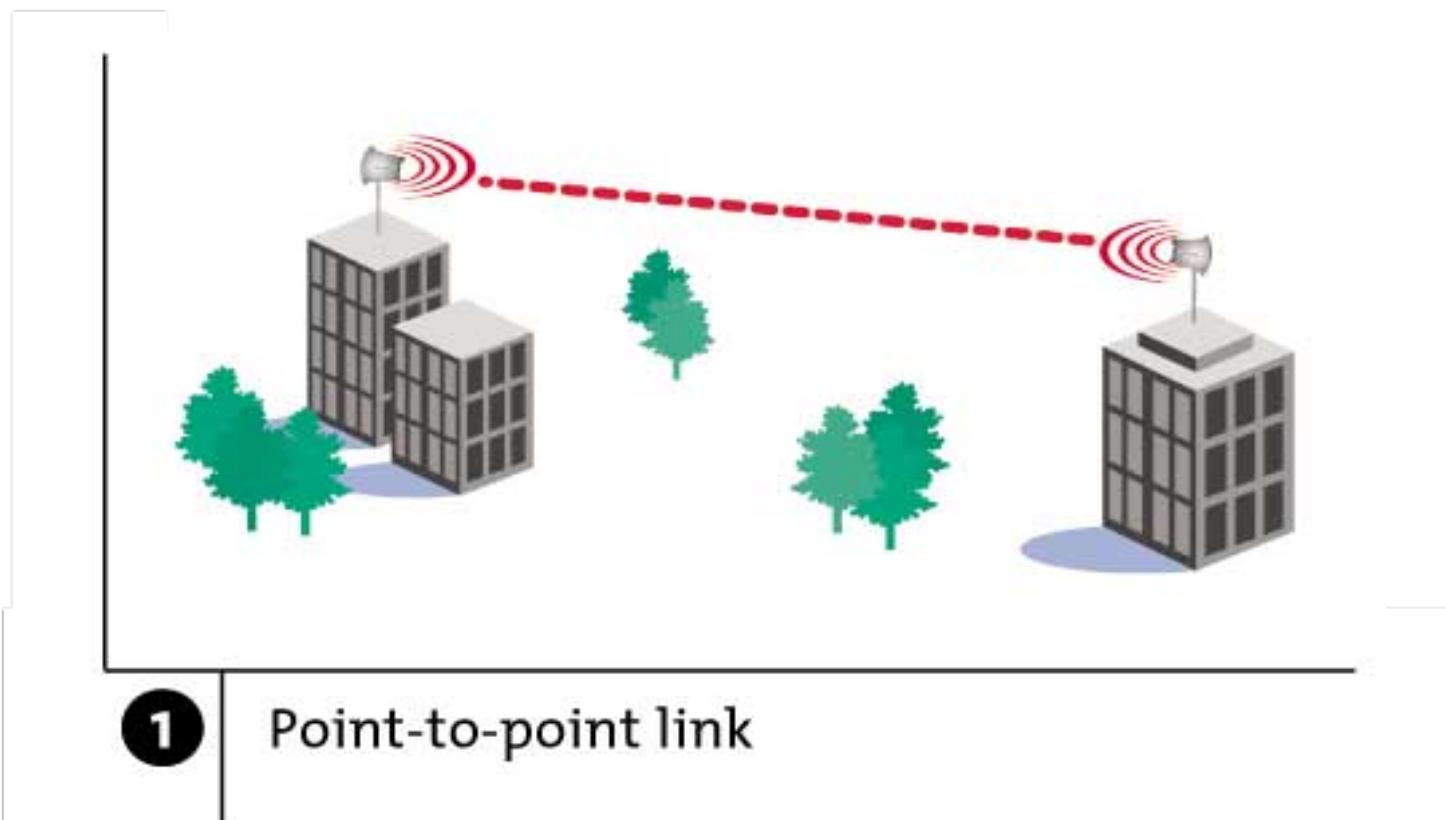
无线链路特性 (2)

- SNR: 信噪比
 信噪比越大，越容易将信号从噪声中识别出来
- 信噪比与比特差错率（BER）的关系
- 给定调制方案: 增加功率 -> 提高信噪比->减小比特差错率
- 给定信噪比: 选择符合BER需求和最高吞吐量的调制方法
 信噪比可能随着调制方法,发送速率的不同而不同



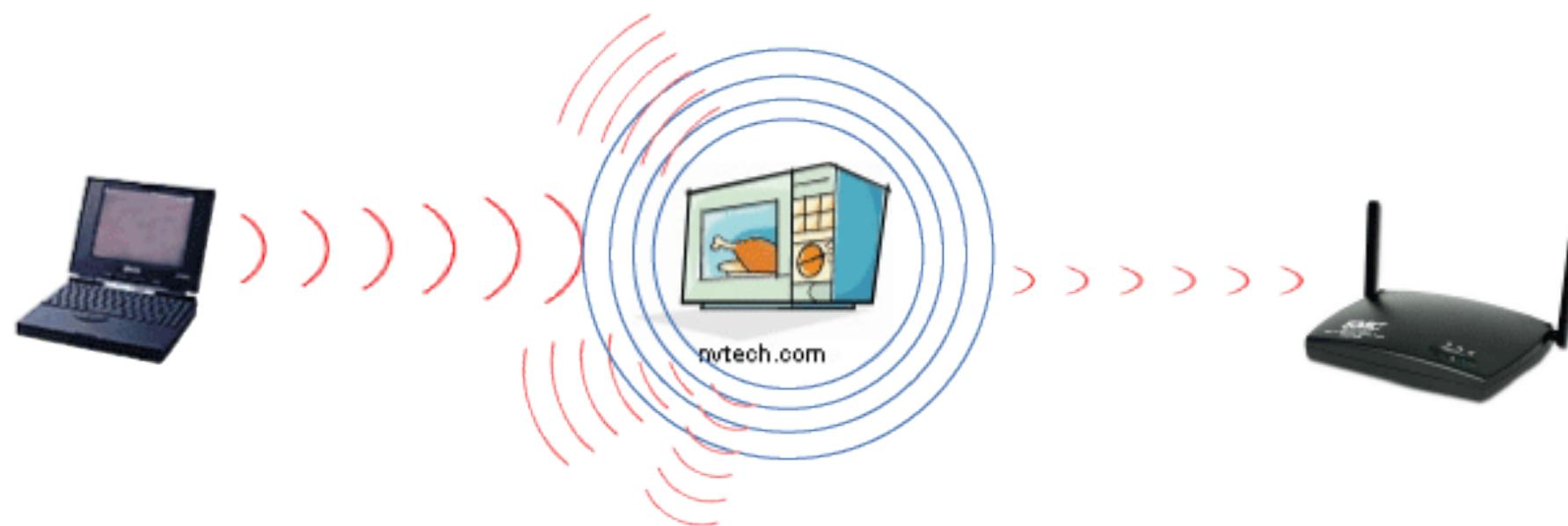
无线链路：高比特误码率

- 信号强度衰减
- 随着传播距离衰减
- 穿透其他物质时信号减弱



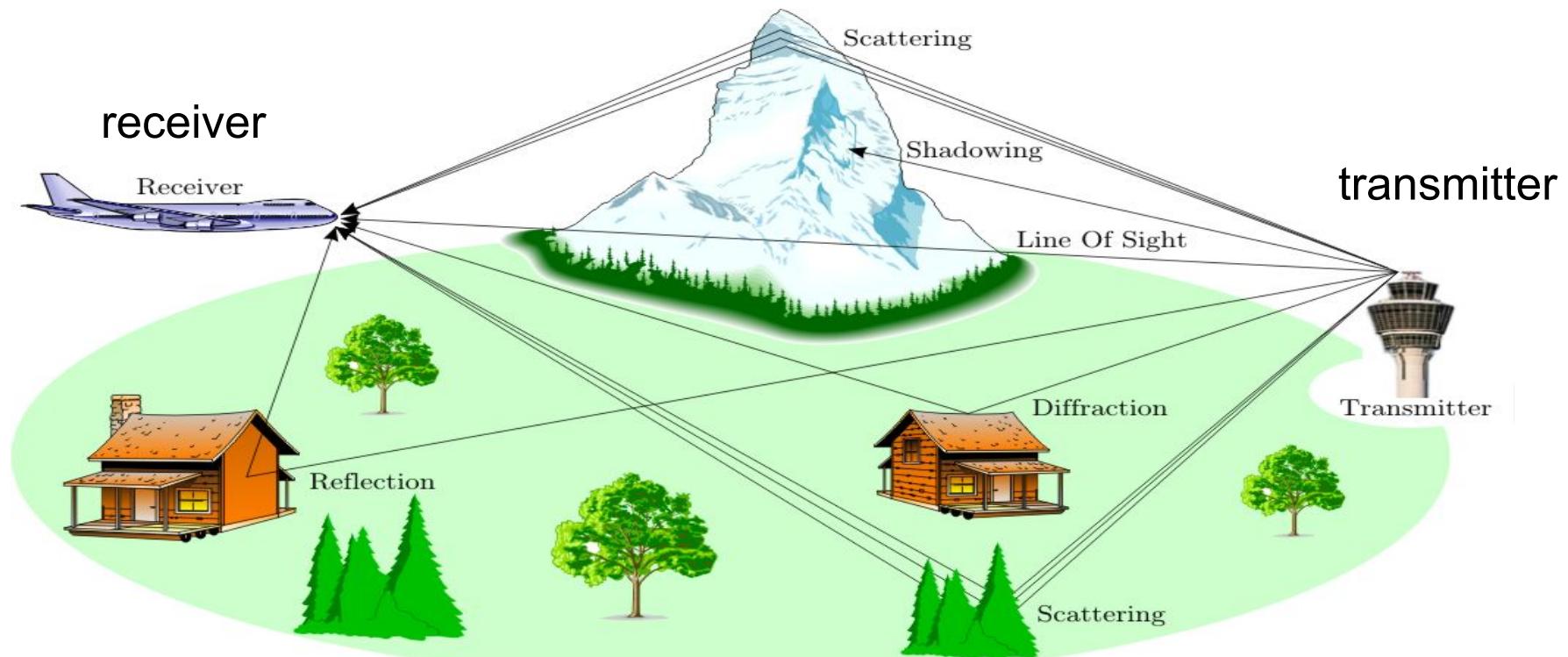
无线链路：高比特误码率

- 其他干扰源
- 同频发射源
 - E.g., 2.4 GHz 无绳电话会对802.11b WLAN产生干扰
- 电磁噪声(e.g., 微波炉)



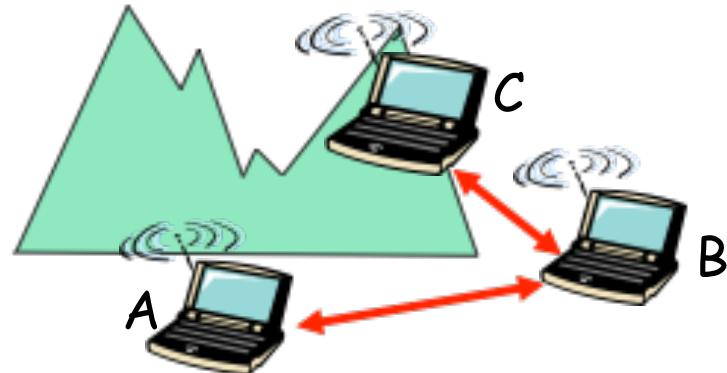
无线链路：高比特误码率

- 多径传播
- 电磁波的反射
- 产生多条不同长度的传播路径
- 在接收方产生模糊信号

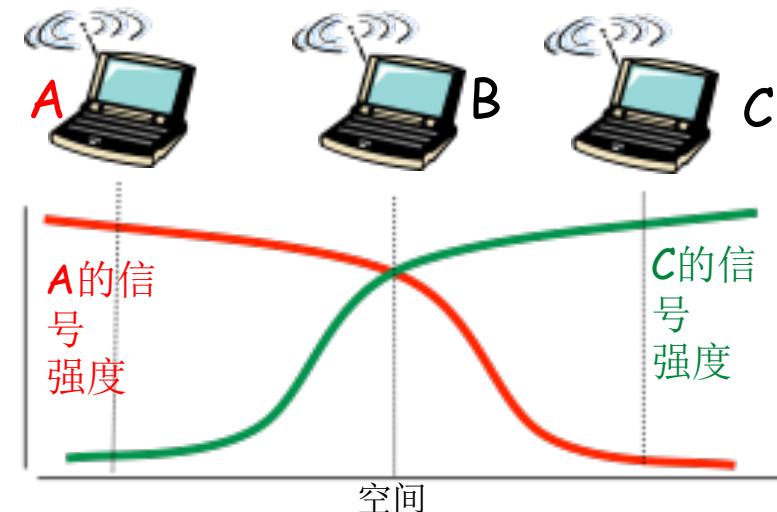


无线网络特性

- 除了多路接入访问以外，多个无线发送端和接收端带来更多的问题：

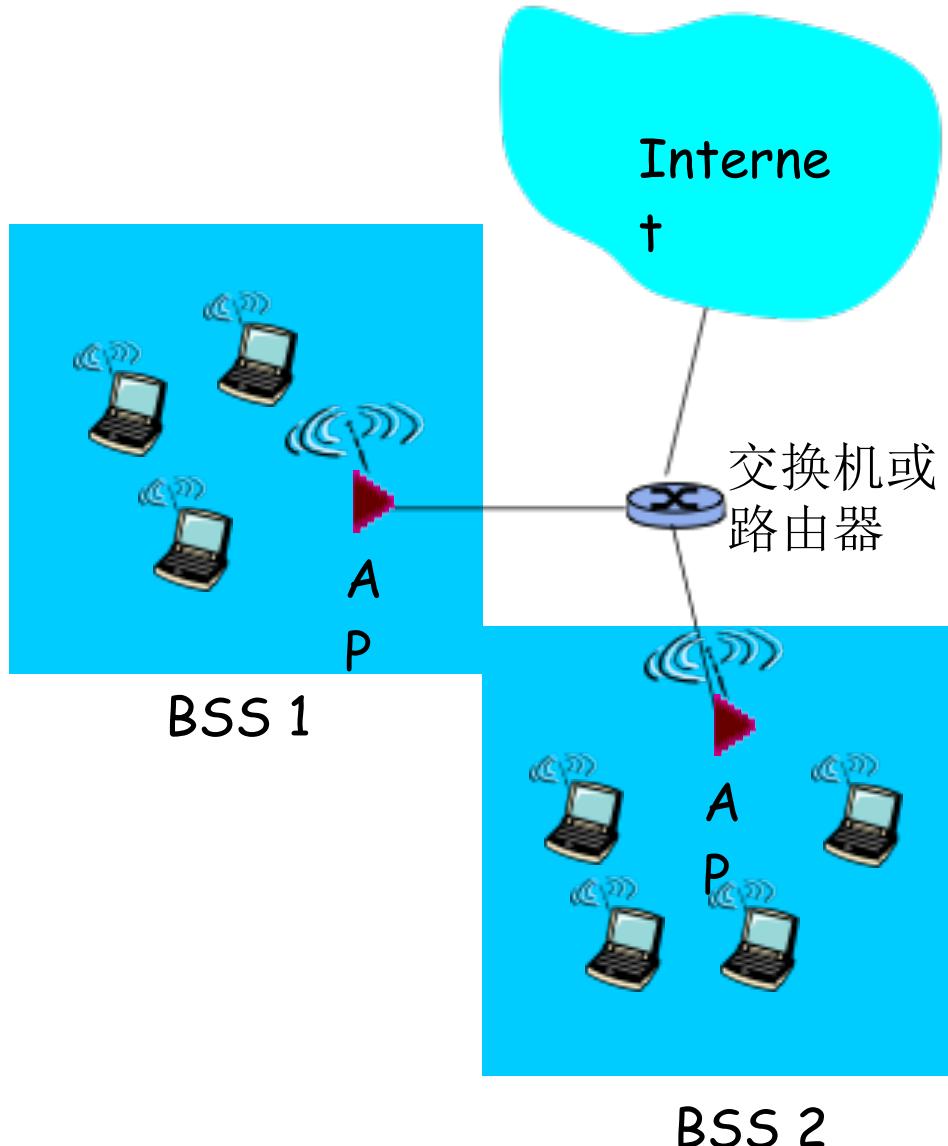


- 隐藏终端问题：
 - B, A能互相监测到对方
 - B, C能互相监测到对方
 - A, C之间不能互相监测到对方
 - 这意味着**A, C**不确定它们的传输会不会在目的地**B**发生干扰



- 信号衰减问题：
 - B, A能互相监测到对方
 - B, C能互相监测到对方
 - A, C不能监测到对方与自己的传输是否在目的地B形成干扰

802.11 局域网体系结构



- 无线主机与基站通信
- 基站 = 接入点 (AP)
- 基本服务集 (BSS)
 - 一个BBS通常包含:
 - “网络名”通过SSID识别
 - 无线主机
- 接入点 (AP): 基站
- 自组织 (ad hoc) 模式: 只有主机

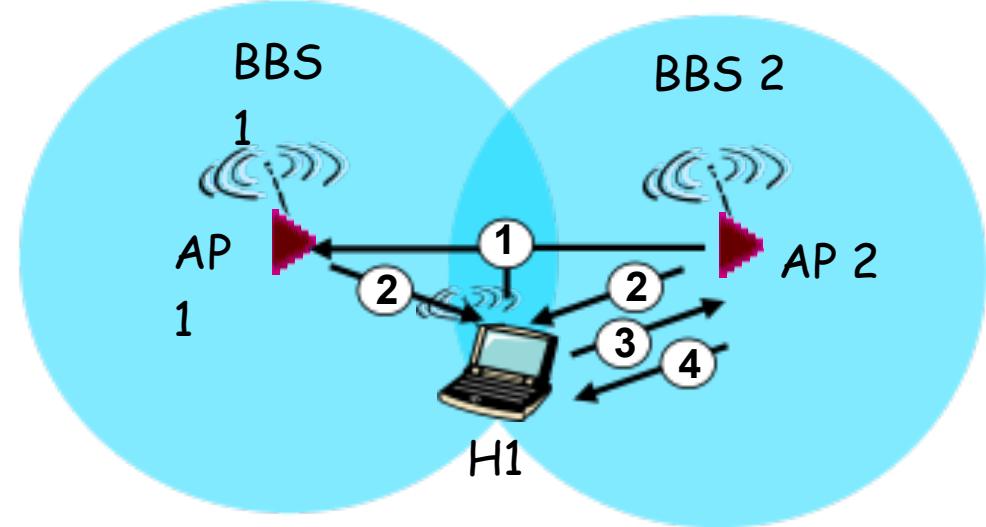
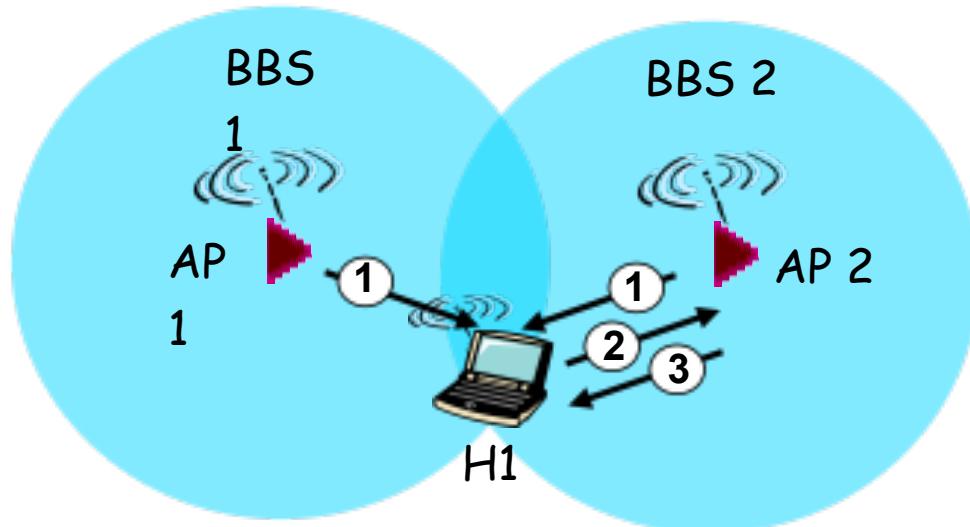
SSID: 服务集标识



802.11: 信道与关联

- 802.11b: 2.4GHz-2.485GHz 频段被分为11个部分重叠的信道
- AP管理员为AP选择工作频段
- 可能发生的冲突: 相邻AP有可能选择了相同的频段。
- 主机: 必须与一个AP关联
- 扫描信道, 监听信标帧 (beacon frames), 每个信标帧包含该AP的SSID和MAC地址
- 选择一个AP建立关联
- 可能需要认证
- 通常使用DHCP协议在AP的子网中获取IP

802.11: 被动/主动扫描



- **被动扫描:**

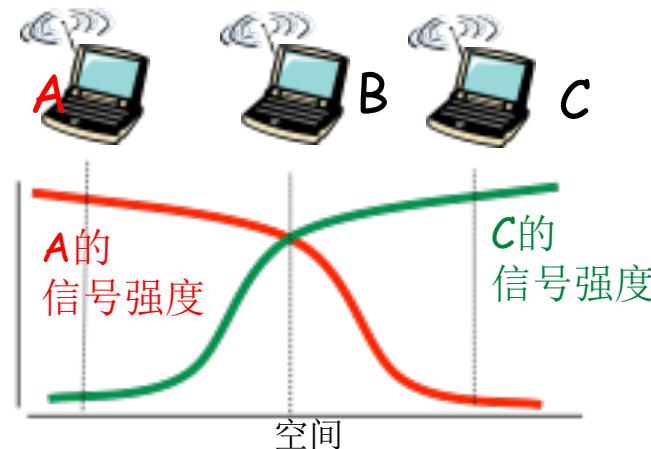
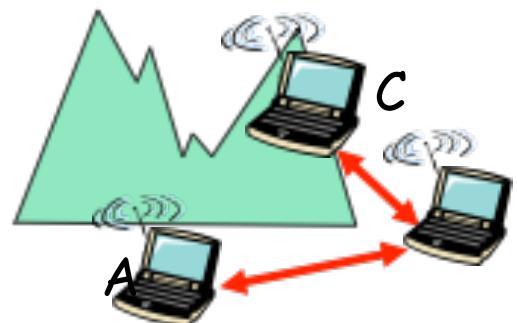
- 自AP发送信标帧
- H1向选择的AP发送关联请求帧
- 选择的AP向H1发送关联响应帧

- **主动扫描:**

- 自H1广播探测请求帧
- 自AP发送探测响应
- H1向选择的AP发送关联请求帧
- 选择的AP向H1发送关联响应帧

IEEE 802.11: 多路访问

- 避免碰撞: 2个以上的节点同时发送数据
- 802.11: CSMA – 传输数据前侦听
不与正在传输数据的其他节点碰撞
- 802.11: 没有碰撞检测!
在接收信号很弱的情况下（衰减）很难接收监测碰撞信号
- 所有碰撞都监测不到的情况: 隐藏终端问题，衰减问题
- 目标: 避免碰撞, 带碰撞避免的载波监听多路访问协议
(CSMA/CA)



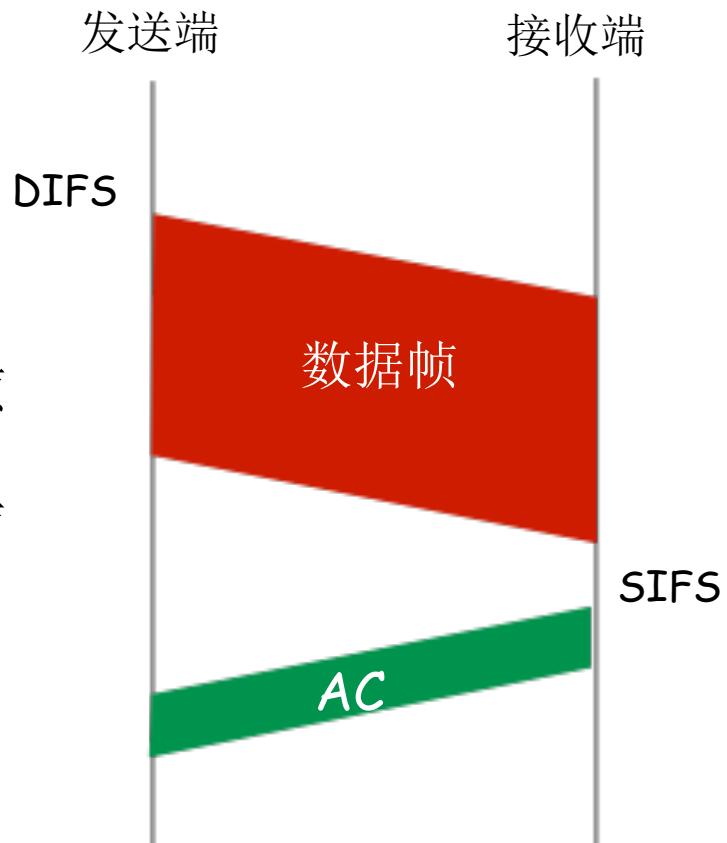


CA: 冲突避免, 而不是检测

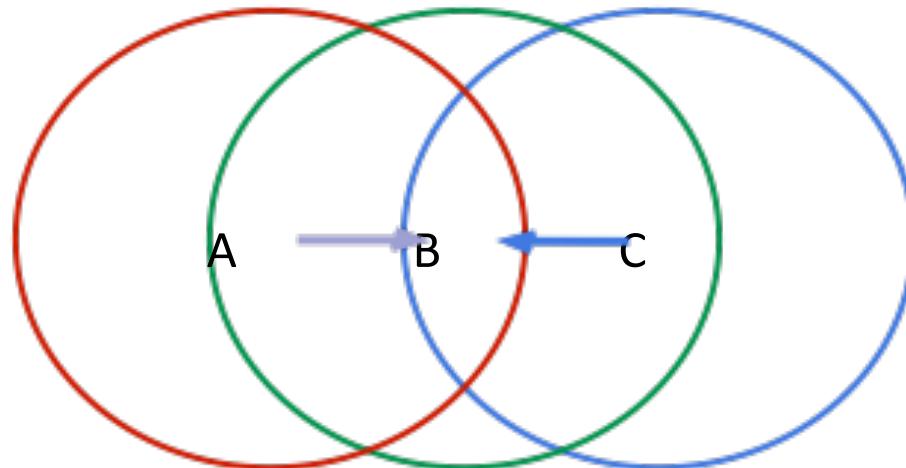
- 有线以太网的冲突检测
 - 站点在传输过程中监听信道
 - 如果存在其他站点发送数据则检测到冲突发生
 - 终止当前数据发送过程并尝试重传
- 问题 #1: 无法检测所有的冲突
 - 隐藏终端问题
 - 信道衰落
- 问题 #2: 难以同时收发
 - 接收信号强度明显弱于发送信号强度
 - 硬件限制
- 因此, 802.11 采用冲突避免机制, 而非冲突检测

IEEE 802.11 MAC 协议: CSMA/CA

- 802.11 发送端
 - 如果监听到信道空闲
 - 它将在一个被称作分布式帧间间隔 (DIFS) 的短时间段后发送该帧
 - 如果监听到信道正忙
 - 选取一个随机回退值计时
 - 当信道空闲时递减该值
 - 当计数值减为0时，该站点发送整个数据帧并等待确认
 - 如果未收到确认，增加回退值，重复第2步
- 802.11接收端
 - 如果数据帧接收成功
 - 在SIFS时间后返回确认信息（确认信息在隐藏终端问题中是必须的）



隐藏终端问题



- A 和 C 无法监听到对方, 同时向 B 发送数据
- 依赖于物理载波监听, 可能产生隐藏终端问题



虚拟载波监听

- 在发送数据帧之前交换控制信息
- 发送方 询问 “Request-to-Send” (RTS), 包括数据帧长度
- 接收方 响应 “Clear-to-Send” (CTS)
- 如果发送方收到 CTS, 则开始发送数据 (指定长度)
- 其他节点收到 CTS, 则认定信道在指定长度数据帧发送期间处于繁忙状态
- 如果多个节点同时检测到一个空闲链路并试图发送一个RTS, 那么他们的RTS帧将彼此冲突。当发送端在一段时间内没有收到CTS帧时, 节点知道发生冲突, 会等待一段随机时间后再试。



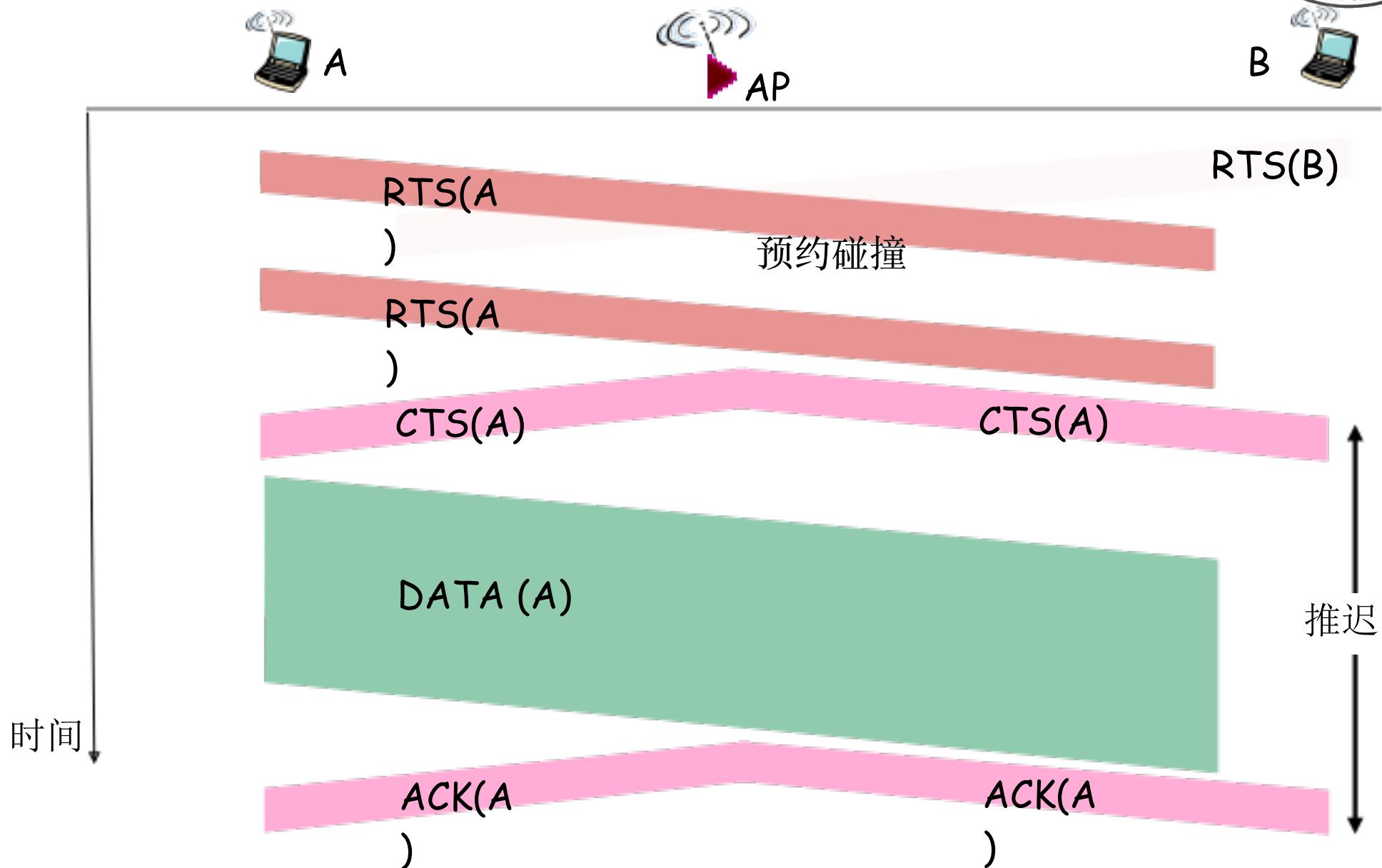
避免传输碰撞(Collision avoidance)

思路：允许发送端“预约”信道，优于随机访问，避免了长数据帧的碰撞

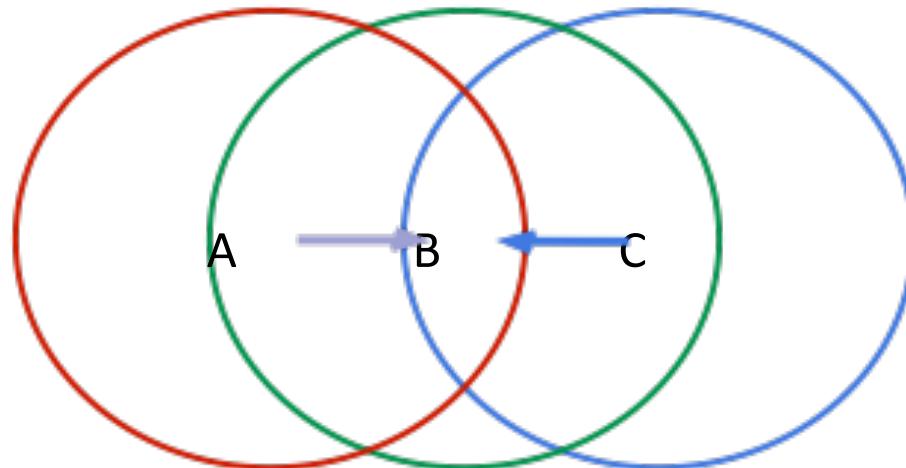
- 发送端先用CSMA方式向AP发送一个短请求发送(CTS) 帧
RTS帧有可能仍然会互相发生碰撞（但是它们很短）
- 当AP收到RTS后，它广播一个允许发送（CTS）帧作为响应
- RTS帧能够被所有的节点监听到
- 发送端发送数据帧
- 其他站点推迟发送

使用短预约帧可以完全避免数据帧碰撞！

使用RTS和CTS的避免信号碰撞

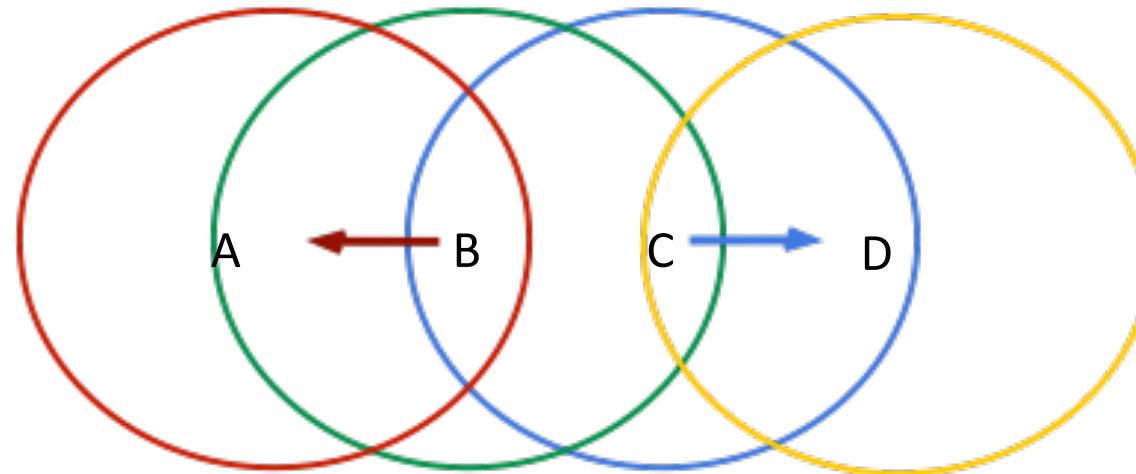


隐藏终端问题



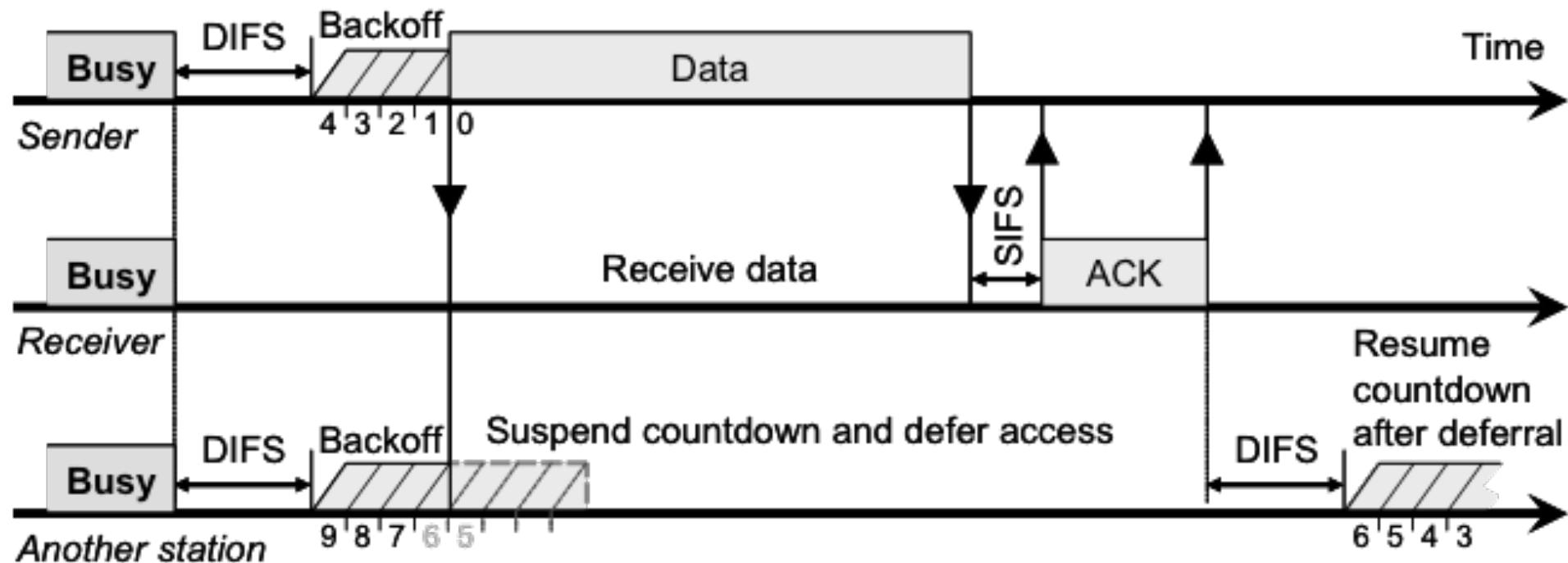
- A 和 C 无法监听到对方, 同时向 B 发送数据
- RTS/CTS 机制
- A和C首先发送RTS, B会收到第一个到达的RTS
- B 仅响应一个 CTS (假如, 响应 A的RTS)
- C 收到 CTS与本地地址**不**匹配, 则等待

暴露终端问题



- B向A发送数据, C同时向D发送数据
- 一旦 C 收到 B 的数据帧, 载波监听机制会禁止其向 D 发送数据, 即使不会产生干扰
- RTS/CTS机制
 - C 监听到 B 发送的RTS, 但未收到A发送的CTS
 - 因此 C 知道其数据发送过程不会对A造成干扰
 - C可以安全的发送数据至D

CSMA/CA 退避算法





802.11 帧: 地址



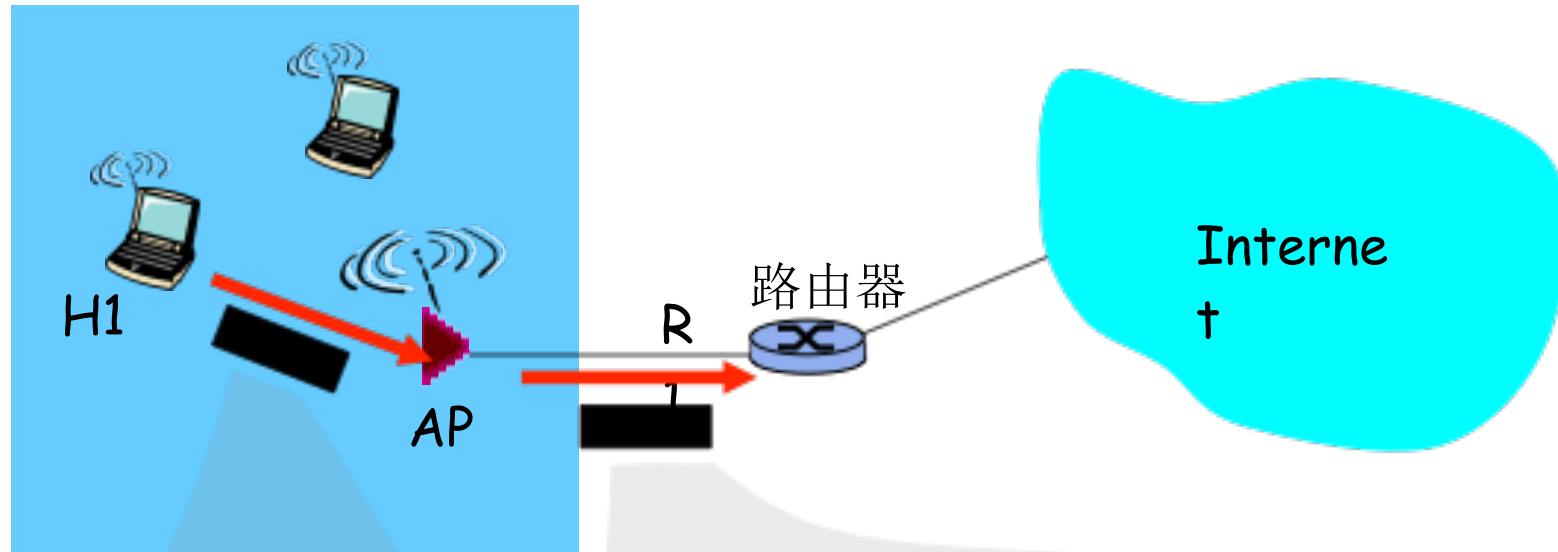
地址1: 要接收帧的无线站点的**MAC**地址

地址2: 传输帧的站点的**MAC**地址

地址3: 包含与**AP**相连的路由器的**MAC**地址

地址4: 只用在自组织网络模式

802.11 帧:地址



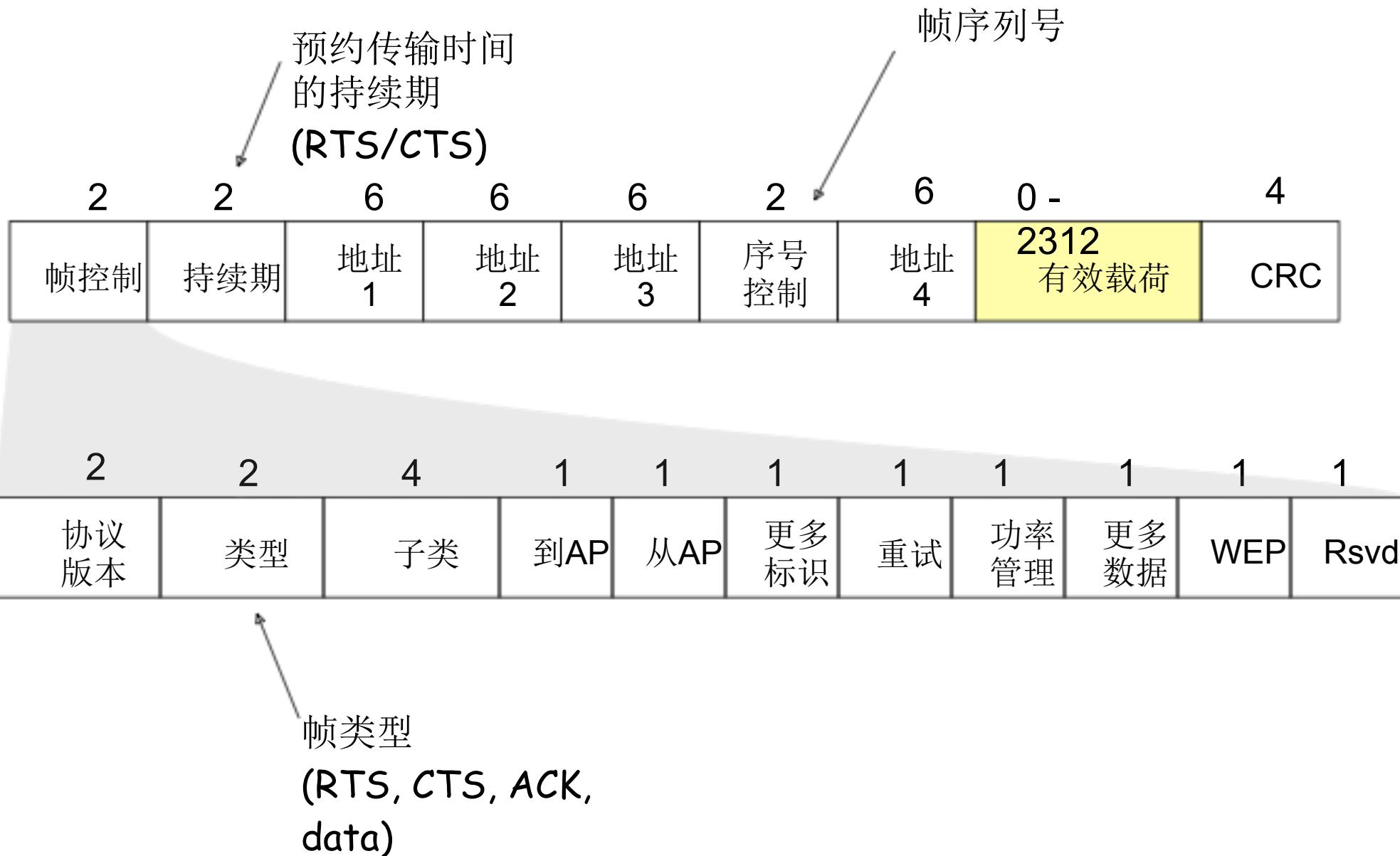
802.3 帧



802.11 帧



802.11 帧: 其他





提纲

- 引言
- 核心问题: 连接到网络
- 网络硬件
- 编码 (NRZ, NRZI, Manchester, 4B/5B)
- 组帧
- 差错检测
- 可靠传输
- 多路访问控制
- 以太网(802.3)
- 无线网络
- 总结





数据链路层组件和协议

- 协议设计组件
 - 编码
 - 帧定界
 - 差错监测
 - 可靠传输
 - 多路访问控制

子层	组件	协议			
		HDLC	PPP	Ethernet	WiFi
LLC	可靠传输	滑动窗口ARQ		--	
	差错检测		CRC		
MAC	多路访问控制	--	--	CSMA/CD	CSMA/CA
	帧定界	面向比特	面向字节	面向比特	面向比特
	寻址	8bit+	1byte	48bit	48bit
物理链路特征		点到点		广播	

谢谢！



华中科技大学
电子信息与通信学院
Email: itec@hust.edu.cn
网址: <http://itec.hust.edu.cn>



参考资料

- *Chapter 2 in L. L. Peterson and B. S. Davie, Computer Networking: A System Approach (5th edition), Morgan Kaufmann, 2012*
- *Chapter 5/6 in James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach (6th edition), Pearson Education Inc., 2012*
- 吴功宜, 计算机网络 (第3版), 清华大学出版社, 2011

附录

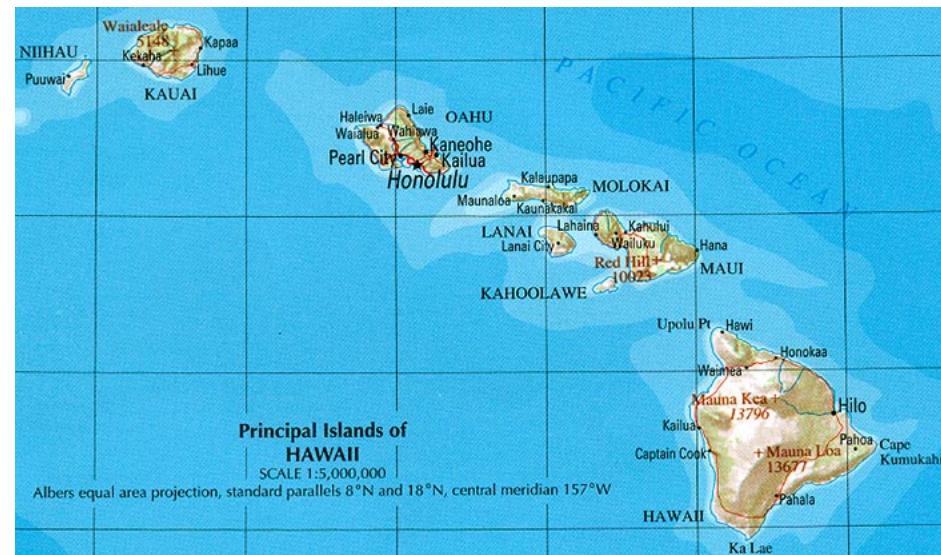


ALOHA的历史故事



Norman Abramson was a professor of engineering at Stanford, but was also an avid surfer. He joined the staff in 1970 and started working on a radio-based data communications system to connect the Hawaiian islands together

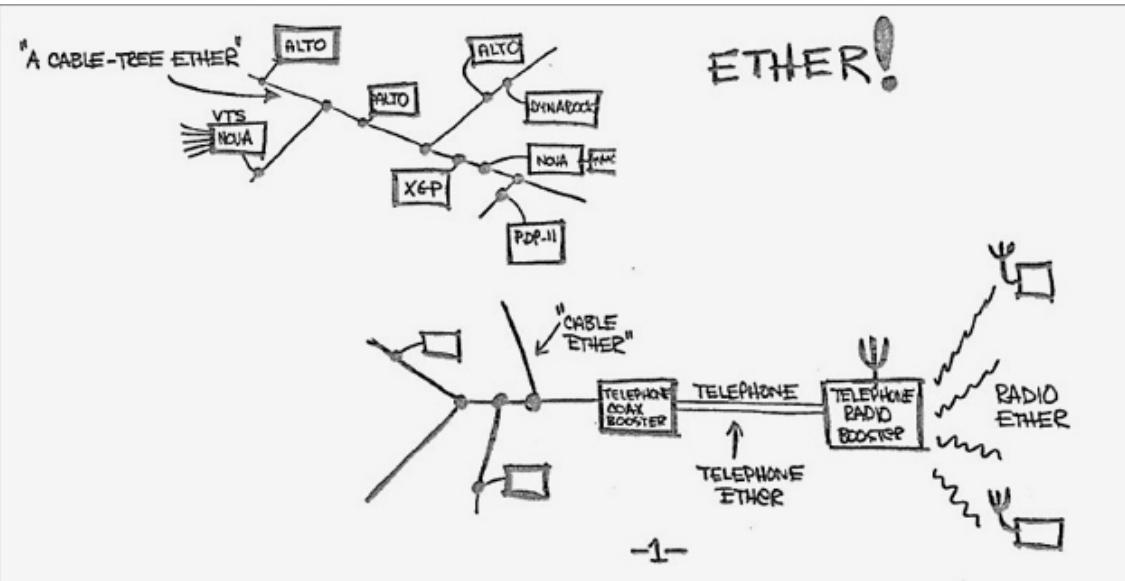
The idea was to use low-cost amateur radio-like systems to create a computer network linking the far-flung campuses of the University.



By late 1970 the system was already in use, the world's first wireless packet-switched network. Abramson connected ALOHAnet to the ARPANET on the mainland in 1972. It was the first time another network was connected to the ARPAnet.



Ethernet的历史故事



He and PARC colleague David Boggs published the concept in a 1976 paper, "Ethernet: Distributed Packet-Switching For LANs."



Metcalfe went on to found 3Com ("computers, communication, compatibility") in 1979
He obtained the United States National Medal of Technology

1973: Bob Metcalfe writes a memo outlining how to connect a new personal computers to a shared printer

Computer Systems

G. Bell, S. Fuller and D. Siewiorek, Editors

Ethernet: Distributed Packet Switching for Local Computer Networks

Robert M. Metcalfe and David R. Boggs
Xerox Palo Alto Research Center



隐藏终端问题

