

计算机网络: 自顶向下方法

肖周芳

计算机学院 1教606

计算机网络: 自顶向下方法 (第7版)

J.F.Kurose, K.W.Ross 著, 陈鸣译, 机械工业出版社, 2018.

Computer Networking: A Top-Down Approach (Sixth Edition)

J.F.Kurose, K.W.Ross, 2017.

本PPT改编自英文版教材附带的PPT。

Introduction 1-1

Chapter 5: The Data Link Layer

目标:

- ❑ 理解链路层服务的原理:
 - 错误检测, 纠正
 - 共享广播信道: 多路访问
 - 链路层寻址
 - 局域网技术: **Ethernet VLANs**

- ❑ 不同链路层技术的实施

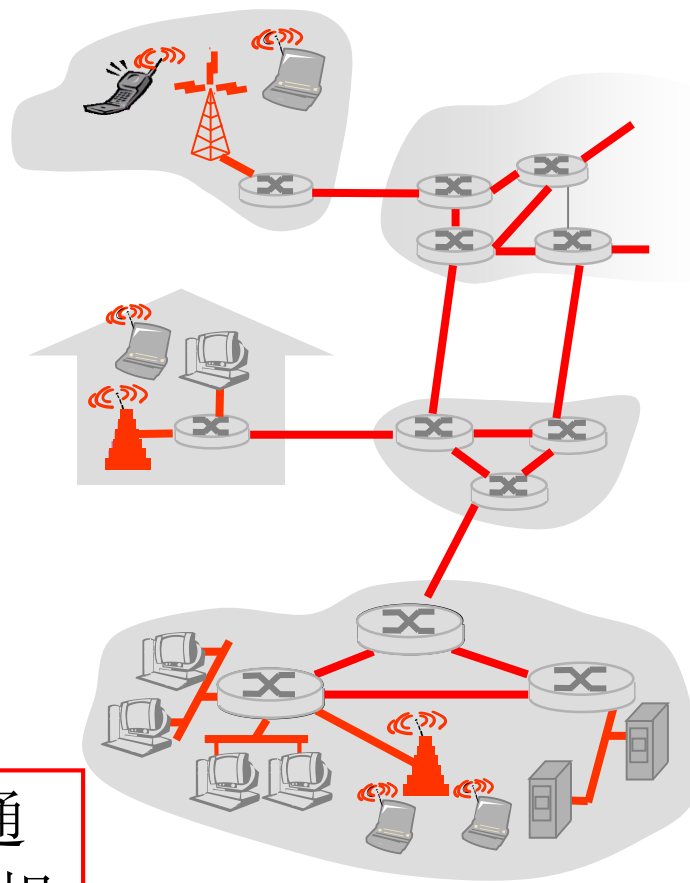
Chapter 5: Link Layer

- ❑ 5.1 概述和服务
- ❑ 5.2 检错和纠错
- ❑ 5.3 多路访问协议
- ❑ 5.4 链路层地址
- ❑ 5.5 以太网
- ❑ 5.6 链路层交换机
- ❑ 5.7 PPP
- ❑ 5.8 链路虚拟化: MPLS

Link Layer: 简介

一些专有名词:

- ❑ 主机和路由是节点 (nodes)
- ❑ 通信路径中连接相邻节点的通信信道称为链路 (links)
 - wired links
 - wireless links
 - LANs
- ❑ layer-2 数据单元 帧(frame), 封装上层数据报



data-link layer : 网络层数据报通过路径中的单段链路、一个节点向相邻节点进行传输

Link Layer: 背景

- ❑ 数据报在路径的不同链路上可能由不同链路层协议所承载:
 - e.g., 第一段链路Ethernet
第二段链路frame relay,
第三段链路802.11
- ❑ 每个链路层协议提供不同的服务
 - e.g., 有些链路层, 可能不提供rdt 传输

传输类比

- ❑ 上海到北京的一个行程
 - limo: 虹桥机场
 - plane: 虹桥到首都机场
 - train: 首都机场到市区
- ❑ 旅行者 = datagram
- ❑ 运输区段 = communication link
- ❑ 运输模式 = link layer protocol
- ❑ 旅行代理 = routing algorithm

链路层服务

❑ 成帧, 链路接入:

- 封装数据报成帧, 添加头部, 尾部
- 共享介质的访问
- 帧头部使用“**MAC**”地址标识源、目的地址
 - 与**IP** 地址不同!

❑ 相邻节点的可靠交付

- 已经讨论过这个问题(chapter 3)!
- 低比特差错的链路较少使用(fiber, some twisted pair)
- **wireless links**: 高比特差错
 - Q: 为什么需要端到端、节点到节点的可靠性?

链路层服务(more)

□ 流控:

- 发送和接收节点的同步问题

□ 差错检测:

- 信号衰减、噪音引起的比特差错.
- 接收端检测错误:
 - 丢弃数据帧或示意发送方重传

□ 差错纠正:

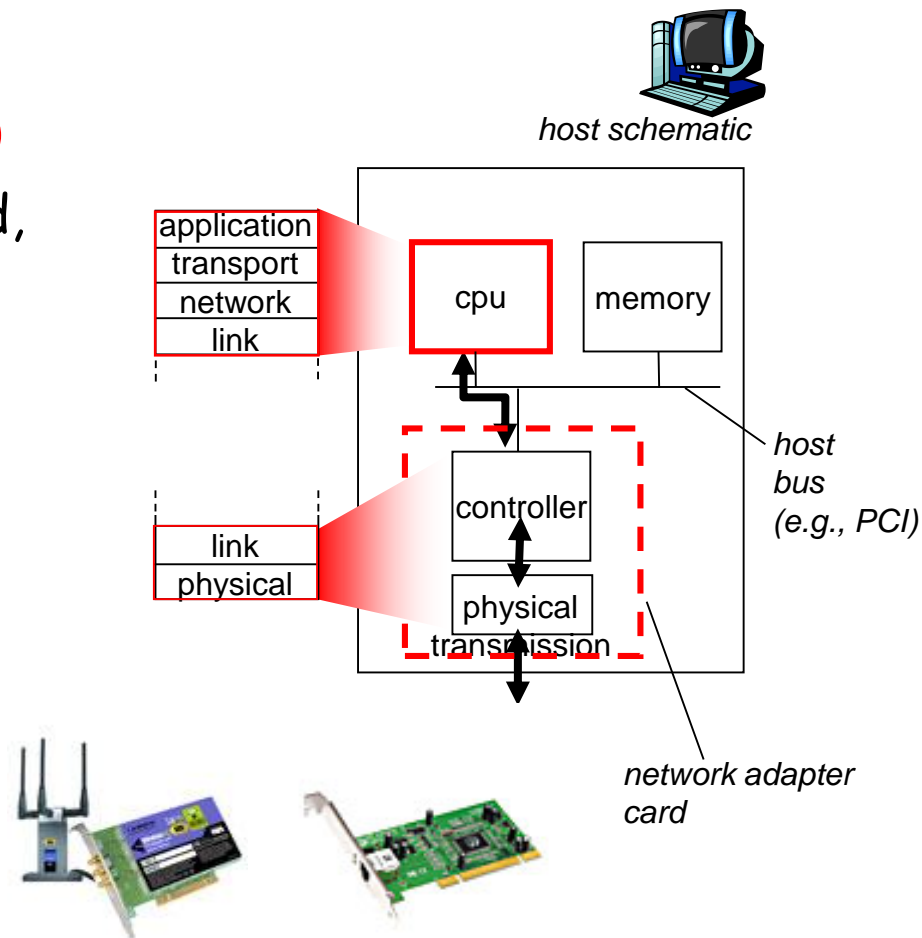
- 接收端检测错误并且纠正比特错误

□ 半双工和全双工

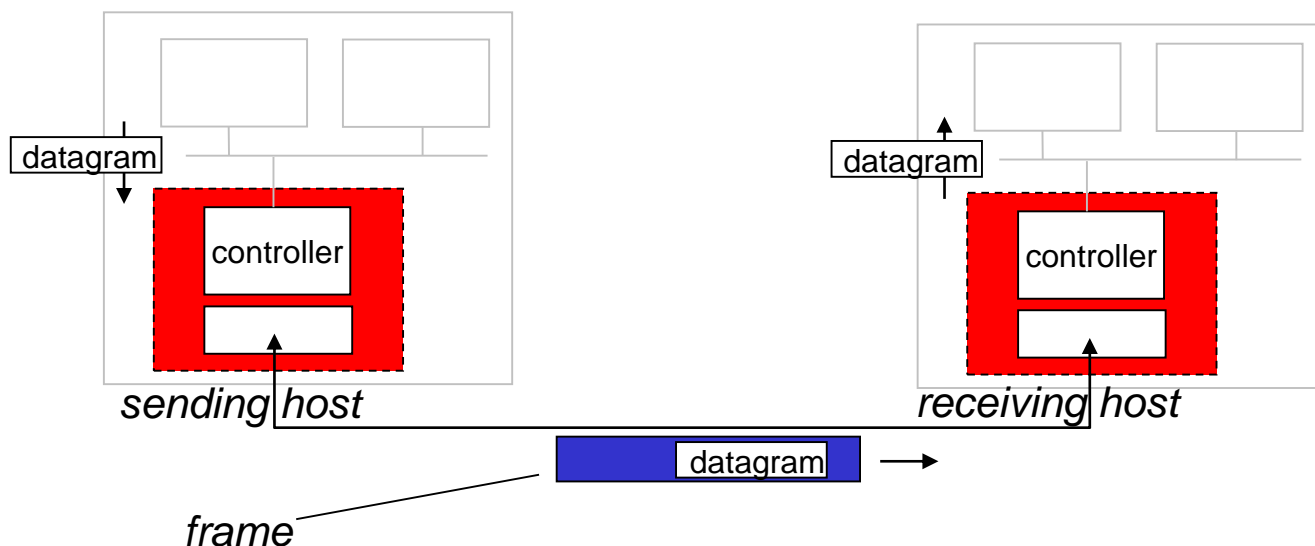
- 半双工, 链路两端的节点都可以发送数据, 但是不能同时发送

链路层在何处实现？

- ❑ 在每一台主机、路由器的
- ❑ 网络适配器 “adaptor”
(NIC network interface card)
 - Ethernet card, PCMCIA card, 802.11 card
 - 实现链路层, 物理层
- ❑ 与主机系统总线连接
- ❑ 由硬件和软件组成



适配器通信



□ 发送端:

- 封装datagram 为 frame
- 增加差错检测, rdt, 流量控制位, etc.

□ 接收端

- 检查差错检测, rdt, 流量控制位, etc
- 提取datagram, 交付给接收端的上层协议

Chapter 5: Link Layer

- 5.1 概述和服务
- 5.2 检错和纠错
- 5.3 多路访问协议
- 5.4 链路层地址
- 5.5 以太网
- 5.6 链路层交换机
- 5.7 PPP
- 5.8 链路虚拟化: MPLS

差错检测

差错检测和纠错比特:

EDC= Error Detection and Correction bits (redundancy)

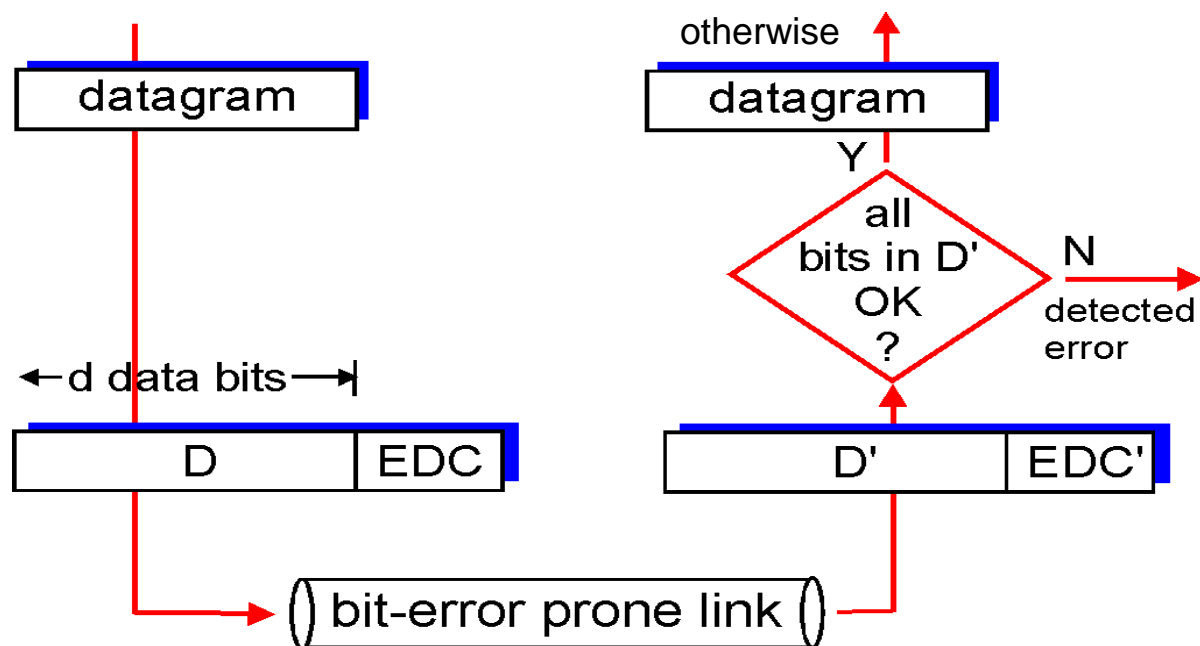
数据: **D** = 差错检测保护的数据, 可能包括头部字段

差错检测不能保证**100%**可靠!

- 协议可能不能检测某些差错, 但很少!
- 较大的 **EDC** 检测和纠错能力较强

3种差错检测技术

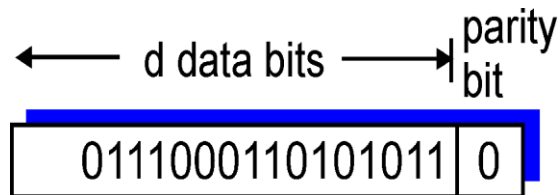
- 奇偶校验
- 校验和方法
- 循环冗余检测



奇偶校验

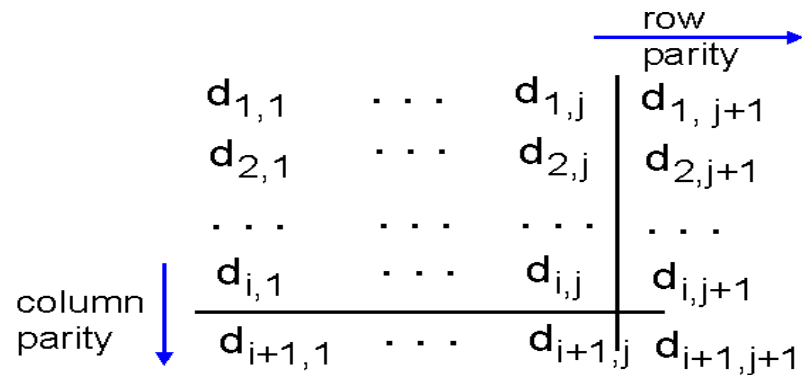
1比特奇偶校验:

Detect single bit errors



二维奇偶校验:

Detect and correct single bit errors



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

no errors

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

parity error

correctable
single bit error

校验和 (checksum, review)

Goal: 在传输的数据报中检测比特错误(e.g., flipped bits)
用于传输层，也用于链路层

发送端:

- ❑ 将segment 内容作为顺序的 16-bit 整数
- ❑ checksum: 将所有整数加起来，用得到的和作为差错检测比特
- ❑ 将 checksum 一块发送

接收端:

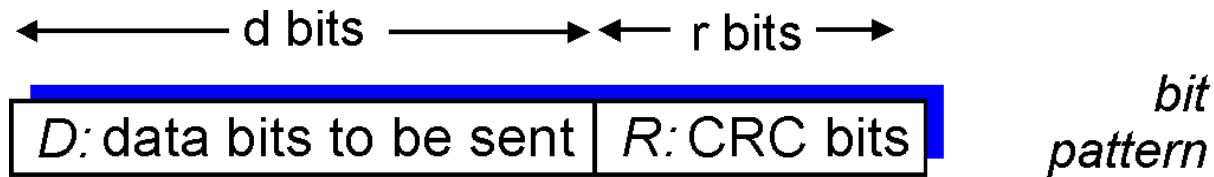
- ❑ 计算接收到数据段的 checksum
- ❑ 检查是否出错:
 - NO - 检测到错误
 - YES - 没检测到错误

But maybe errors nonetheless?

循环冗余检测

CRC: Cyclic Redundancy Check

- ❑ 有效数据, **D**, 使用二进制编码
- ❑ 选择 **r+1** 比特模式作为生成多项式 (generator), **G**
- ❑ goal: 选择 **r** CRC附加比特位, **R**, 得到 **d+r** 模式
 - $\langle D, R \rangle$ 恰好能被 G (modulo 2) 整除
 - 接收方知道 G , 将 $\langle D, R \rangle$ 除 G . 如果余数为非零: 则出错!
 - 能够检测少于 **r+1** 位错
- ❑ 被广泛采用 (Ethernet, 802.11 WiFi, ATM)



$$D * 2^r \text{ XOR } R$$

mathematical formula

CRC Example

目标:

$$D \cdot 2^r \text{ XOR } R = nG$$

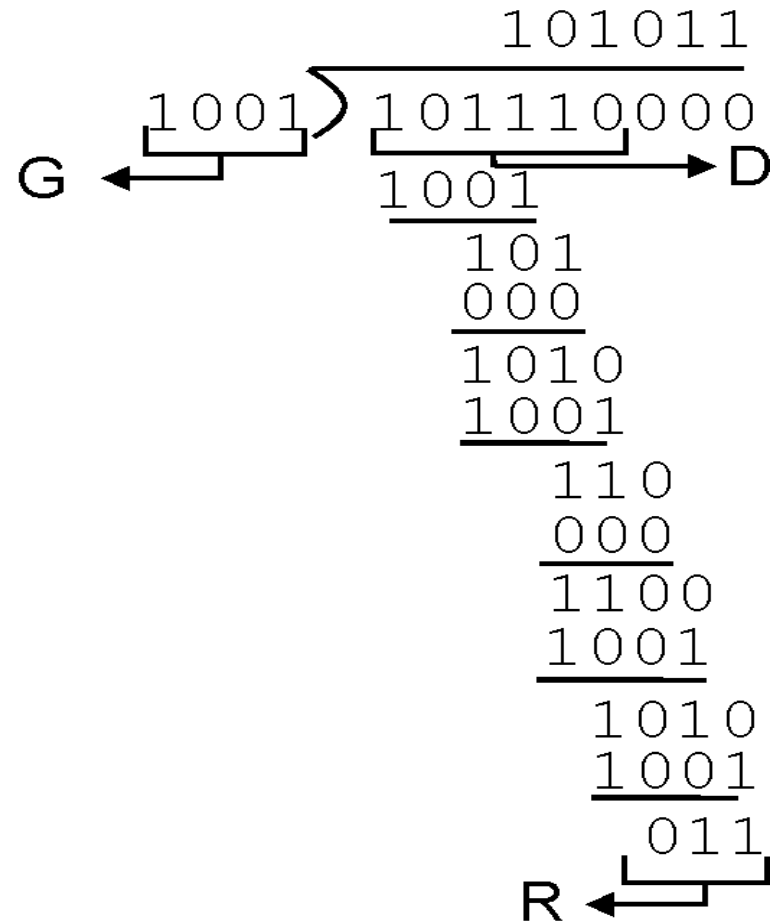
等同于:

$$D \cdot 2^r = nG \text{ XOR } R$$

等同于:

$D \cdot 2^r$ 被 G 除, 余数为 R

$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$



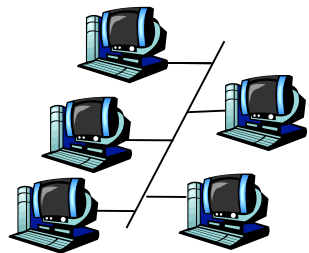
Chapter 5: Link Layer

- 5.1 概述和服务
- 5.2 检错和纠错
- 5.3 多路访问协议
- 5.4 链路层地址
- 5.5 以太网
- 5.6 链路层交换机
- 5.7 PPP
- 5.8 链路虚拟化: MPLS

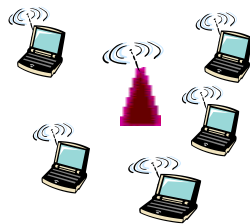
多路访问协议

两种类型链路:

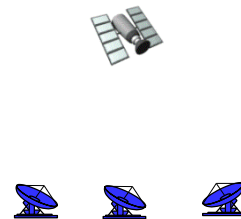
- ❑ 点对点: (point-to-point)
 - 拨号网络的PPP
 - Ethernet 交换机和主机之间的点对点链路
- ❑ 广播链路: (broadcast, 共享的链路或介质)
 - 共享式以太网
 - 802.11 无线网络



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



humans at a
cocktail party
(shared air, acoustical)

多路访问协议

- ❑ 共享广播信道的访问
- ❑ 两个或更多节点的同时传输会产生干涉
 - 冲突(**collision**): 节点同时接到多个帧

多路访问协议(*multiple access protocol*)

- ❑ 节点共享信道的分布式算法,
i.e., 决定了哪个节点、什么时候能够传输
- ❑ 关于信道共享的通信必须使用信道本身进行通信!
没有其它信道可以用于节点之间协调

理想的多路访问协议

速率为 R bps的广播信道

1. 一个节点传输时, 能够以速率 R 传输.
2. 当 M 个节点传输时, 每个节点以速率 R/M 传输
3. 完全的分布式:
 - 没有中心节点参与协调
 - 没有时钟、时隙(clocks, slots)同步
4. 实现简单

多路访问协议：分类

三种类型：

- ❑ 信道划分协议：(Channel Partitioning)
 - 把信道划分成小片(time slots, frequency, code)
 - 各个节点分配给分片后的信道
- ❑ 随机接入协议：(Random Access)
 - 信道不分片, 允许冲突
 - 能够从冲突中恢复
- ❑ 轮流协议：("Taking turns")
 - 节点轮流使用信道

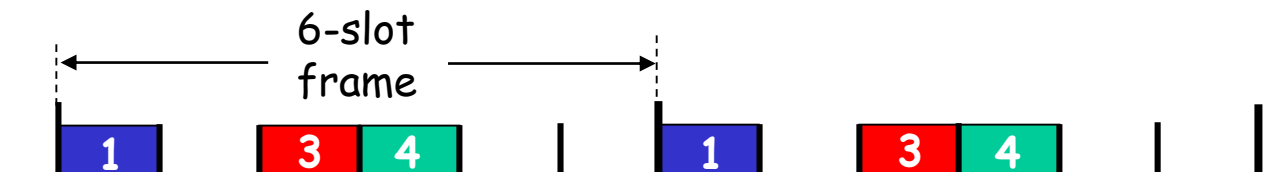
信道划分多路访问协议: TDMA

时分多路复用

TDMA(time division multiple access)

- ❑ 节点轮流访问信道
- ❑ 每个节点分到固定长度的时隙(slot, length = pkt 传输时间)
- ❑ 没有使用的时隙被浪费!
- ❑ example:

6-station LAN, 1,3,4 传输, slots 2,5,6 被浪费

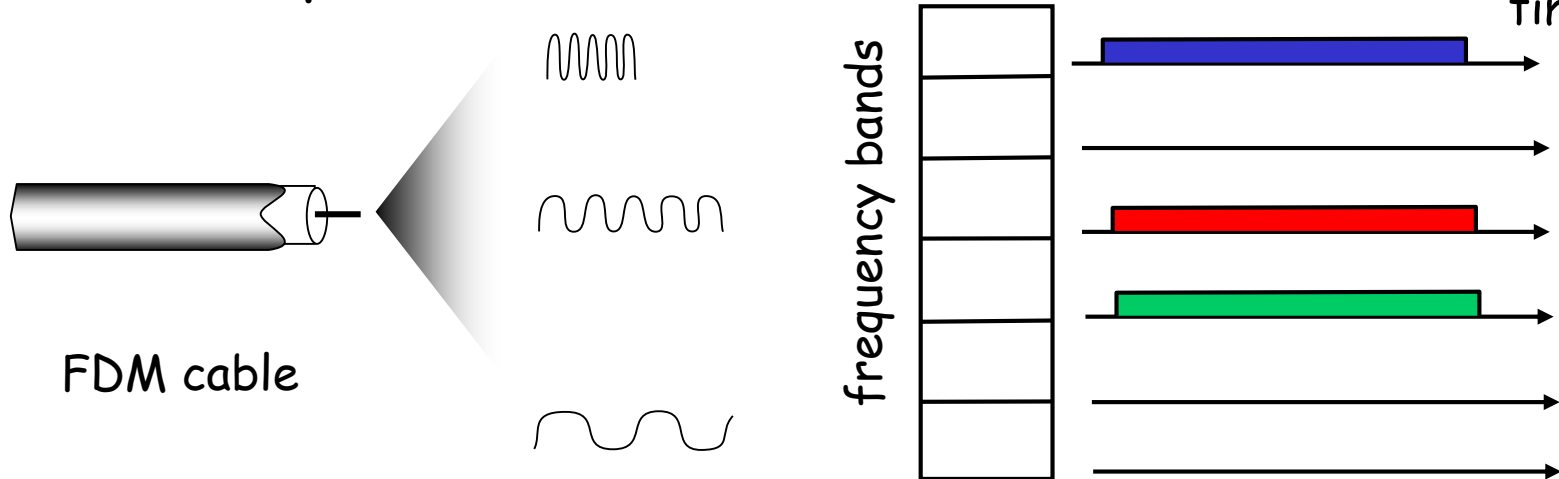


信道划分多路访问协议: FDMA

频分多路复用

FDMA(frequency division multiple access)

- 信道带宽按频率分段
- 每个节点分到固定的频段
- 没有使用的频段被浪费！
- example: 6-station LAN, 1,3,4 传输, 频段2,5,6 闲置



随机接入协议

Random Access Protocols

- ❑ 当节点有数据发送的时候
 - 以信道速率 R 发送数据.
 - 节点之间没有事先的协调
- ❑ 两个或更多的节点传输 → 冲突 ("collision"),
- ❑ 随机接入 **MAC** 协议定义了:
 - 如何检测冲突
 - 如何从冲突中恢复 (e.g., 通过延期的重传)
- ❑ 随机接入 **MAC** 协议例子:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

时隙ALOHA

Slotted ALOHA

Assumptions:

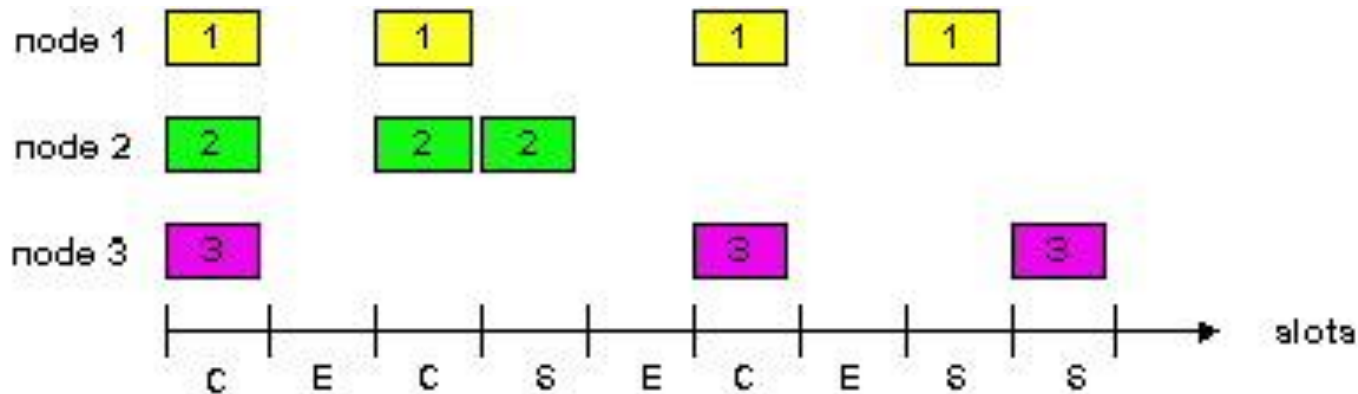
- 所有帧恰好为 L bits长
- 时隙被划分为长度为 L/R
- 节点只在时隙开始时传输帧
- 节点是同步的
- 如果两个或多个帧碰撞发生, 则所有节点在时隙结束前检测碰撞

Operation:

- 如果节点发送一个新帧, 在下一个时隙开始时发送
 - *如没有冲突:*
节点成功发送该帧
 - *如果有冲突:*
节点在时隙结束前检测碰撞, 以概率 p 在后续的每个时隙开始时重传, 直到发送成功

时隙ALOHA

Slotted ALOHA



优点:

- ❑ 只有一个活跃节点时, 高效
- ❑ 高度分布: 每个节点单独检测碰撞并决定何时重传
- ❑ 简单

缺点:

- ❑ 有冲突, 浪费时隙
- ❑ 空闲时隙
- ❑ 成功时隙: 刚好有一个节点传输的时隙
- ❑ 时钟同步问题

时隙ALOHA效率

Efficiency :

当有大量的节点要发送数据时，成功时隙占总时隙的份额

- ❑ 假设: N 节点中每一个时隙以概率 p 传输数据报
- ❑ 任意一个节点成功传送的概率 = $p(1-p)^{N-1}$
- ❑ N 个活跃节点都成功传送的概率 = $Np(1-p)^{N-1}$

❑ 效率最大化:

找到 p^* 使 $Np(1-p)^{N-1}$ 最大

❑ 取 $Np^*(1-p^*)^{N-1}$

N 趋于无穷极限:

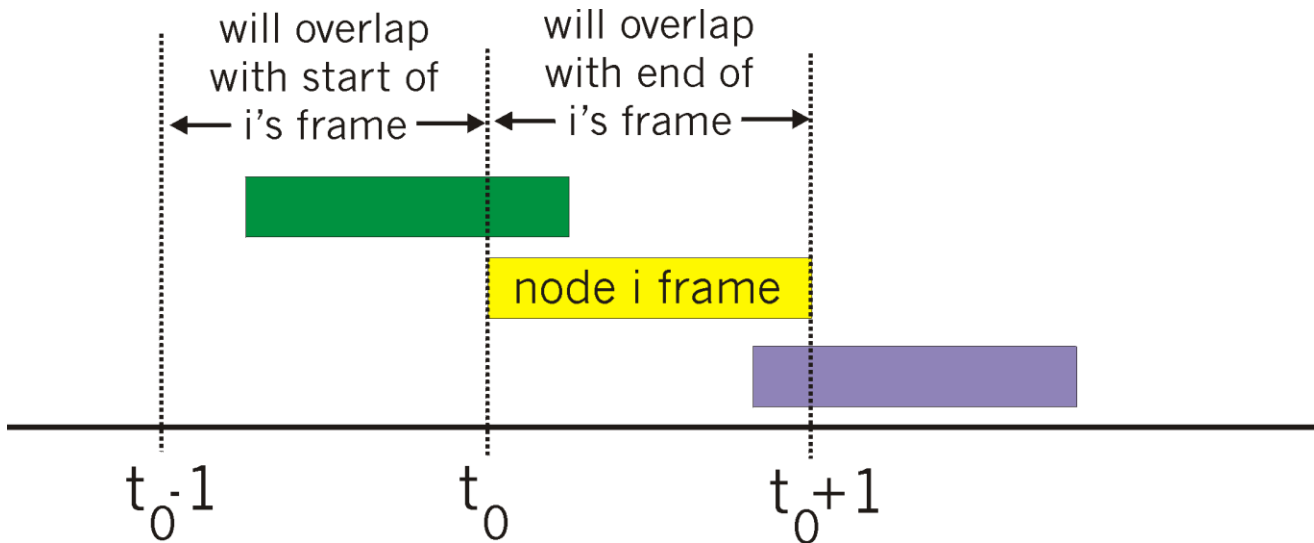
最大效率 = $1/e = .37$

At best:

信道利用率仅为 37%!

纯ALOHA(unslopped)

- ❑ unslotted Aloha:
- ❑ 简单, 不需要同步, 当有数据帧到达时马上传输
 - 同步传输
- ❑ 冲突概率增加:
 - 在 t_0 发送的帧会与 $[t_0-1, t_0+1]$ 时段内发送的帧冲突



纯ALOHA效率

$P(\text{成功发送的概率}) = P(\text{节点传输概率}) \cdot$

$P(\text{在 } [t_0-1, t_0] \text{ 没有节点传输的概率}).$

$P(\text{在 } [t_0, t_0+1] \text{ 没有节点传输的概率})$

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

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

... 取 $N \rightarrow \infty$ 时极限...

$$= 1/(2e) = .18$$

比slotted Aloha更差!

载波侦听多路访问

CSMA (Carrier Sense Multiple Access)

CSMA: 节点中传输前先侦听信道:

- ❑ 如果侦听到信道是空闲的，则传输整个帧
- ❑ 如果侦听到信道忙，则等待一段时间

CSMA 冲突

冲突仍然会发生:

传播的延迟意味着双方都不能完全检测到冲突

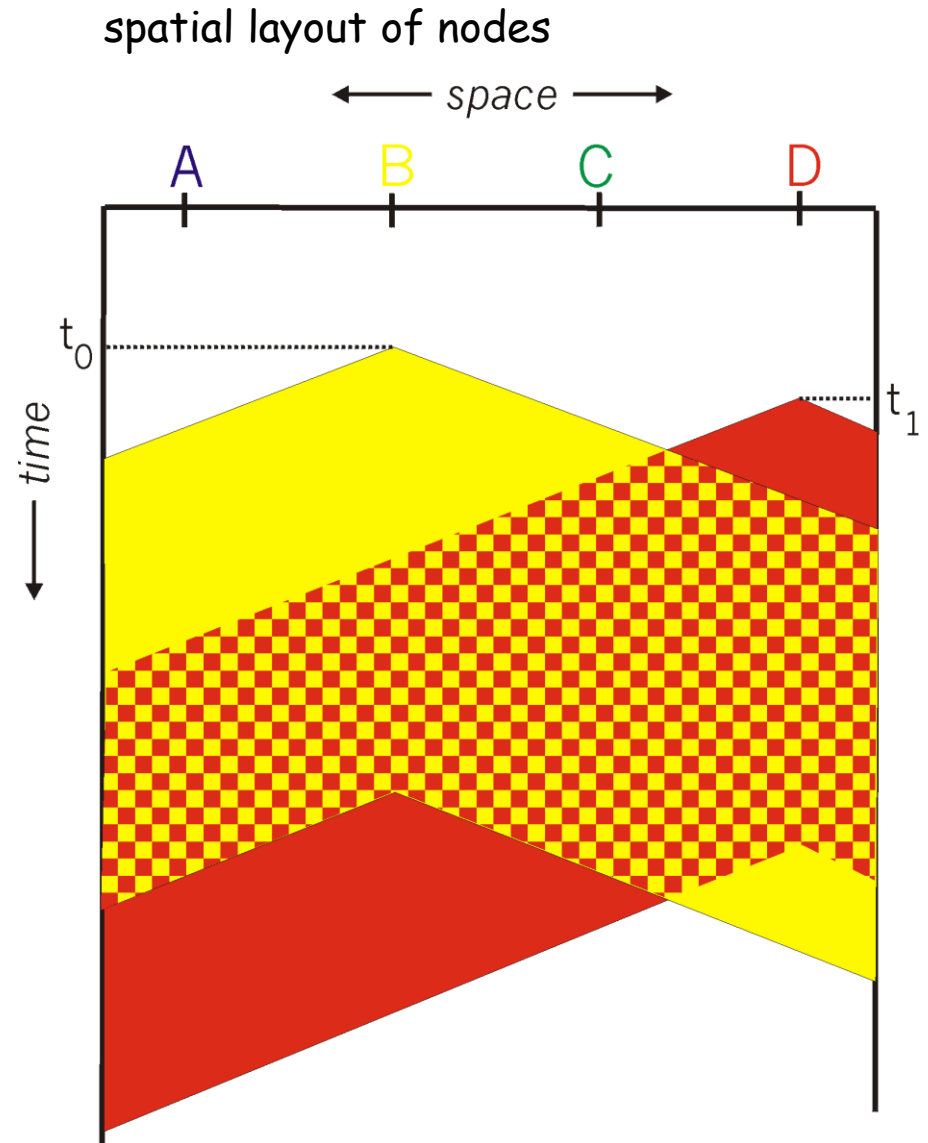
传播时延越长?

冲突:

整个数据帧的传输时间被浪费

note:

距离& 传播延迟决定了冲突的概率



CSMA/CD

CSMA/CD: 载波侦听、冲突检测

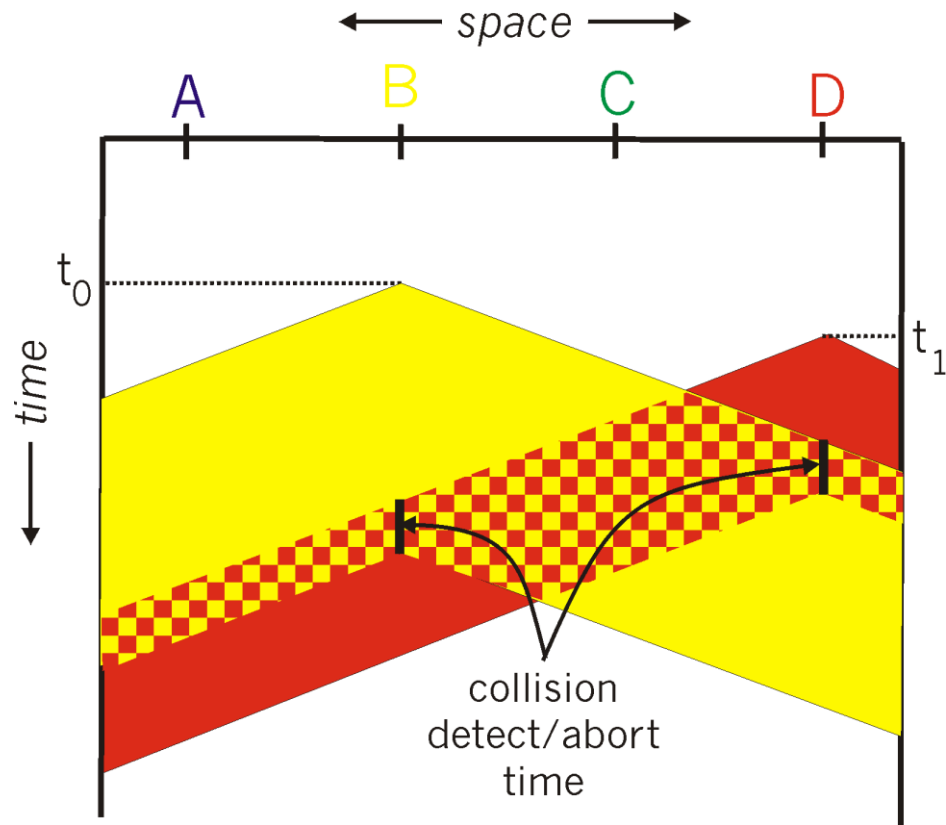
- 检测到冲突
- 立即停止传输，减少信道浪费

□ 冲突检测:

- 在有线**LAN**中易于实现:
通过测量信号强度, 与发送、接收的信号比较
- 在无线**LAN**中难以实现:
接收的信号可能发生衰减、反射等

CSMA/CD

检测到冲突后停止发送



轮流 MAC 协议

轮询(Polling)

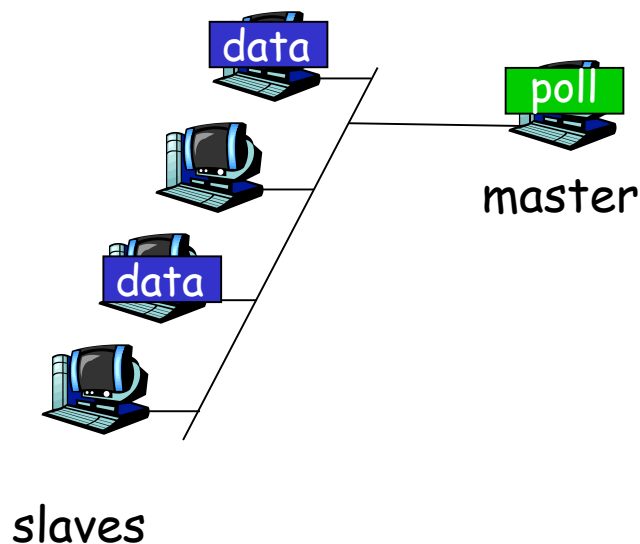
❑ 主节点:

主节点询问从节点是否发送数据

❑ 用于工业控制网络

❑ 存在的问题:

- 轮询的开销
- 单点错误(master)

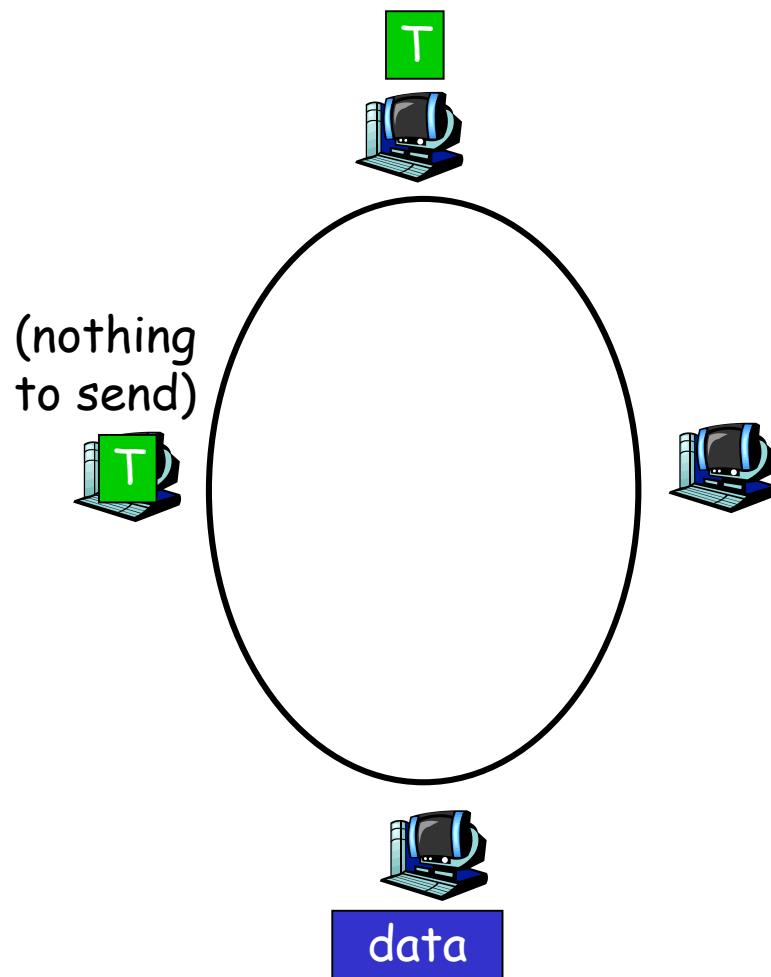


轮流 MAC 协议

令牌传递协议

(Token passing)

- ❑ 控制令牌在节点中顺序传递
- ❑ token message
- ❑ 存在的问题:
 - 令牌开销
 - 延迟
 - 单点错误(token)



MAC 协议小结

- ❑ 信道分区(*channel partitioning*)
 - 按time, frequency 或 code
 - TDMA, FDMA, CDMA
- ❑ 随机访问(*random access*)
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - CSMA/CD 用于Ethernet
 - CSMA/CA 用于802.11(Wifi)
- ❑ 轮流协议(*taking turns*)
 - 主节点轮询, 令牌传递
 - Bluetooth, FDDI, IBM Token Ring

Chapter 5: Link Layer

- 5.1 概述和服务
- 5.2 检错和纠错
- 5.3 多路访问协议
- 5.4 链路层地址
- 5.5 以太网
- 5.6 链路层交换机
- 5.7 PPP
- 5.8 链路虚拟化:MPLS

MAC 地址和 ARP

□ 32-bit IP 地址:

- 网络层地址
- 用于数据报寻址, IP子网

□ MAC (LAN, physical, Ethernet) 地址:

- 功能:

在同一个子网, 从一个接口卡传送帧到另外一个接口卡

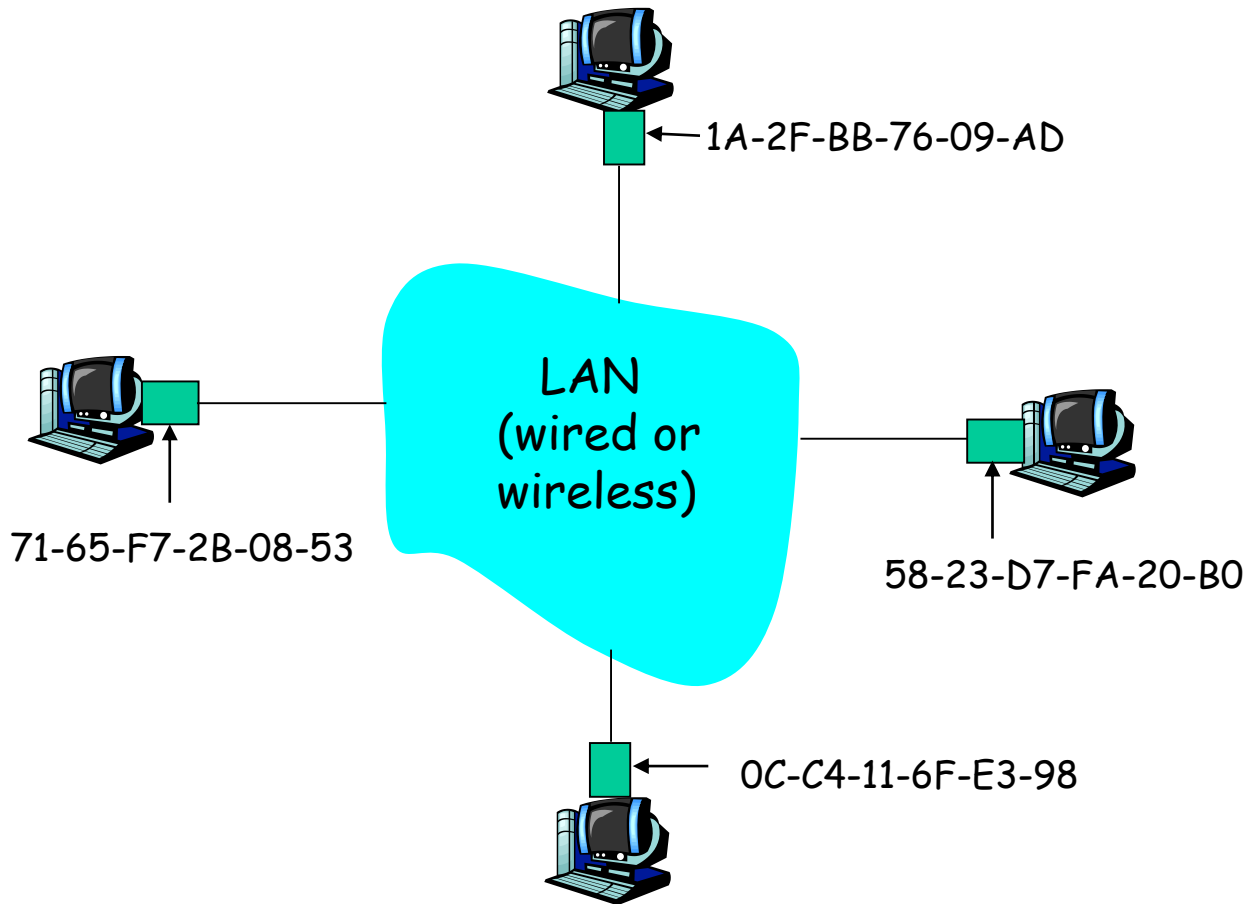
- 48 bit MAC 地址
 - 固定于NIC ROM, 有时软件可修改

MAC 地址(more)

- ❑ **MAC** 地址由 **IEEE**管理分配
- ❑ 制造商购买部分**MAC**地址空间
- ❑ analogy:
 - (a) **MAC address**: 类似身份证号码
 - (b) **IP address**: 类似邮政编码
- ❑ **MAC** 扁平地址结构 → **portability**
 - 能够从一个**LAN**移动 接口卡 到另外一个**LAN**
- ❑ **IP** 层次地址结构 → **NOT portable**
 - 地址依赖于所在的**IP** 子网

MAC 地址和 ARP

LAN 上每块接口卡都有一个全球唯一 LAN 地址



Broadcast address =
FF-FF-FF-FF-FF-FF

■ = adapter

地址解析协议

ARP: Address Resolution Protocol

Question: 如何从B的IP地址知道B的MAC地址?

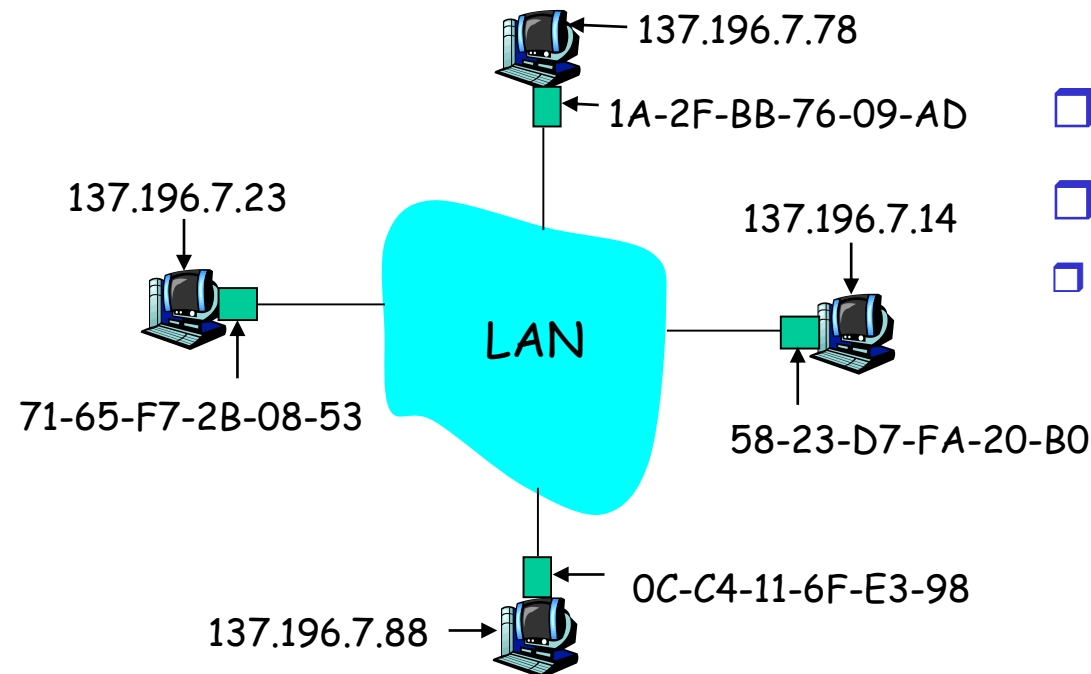
□ LAN上每个IP节点 (host, router) 都有一个**ARP 表**

□ ARP 表:

□ IP/MAC 地址的映射关系

□ < IP address; MAC address; TTL >

○ TTL (Time To Live):
映射的过期时间
(typically 20 min)

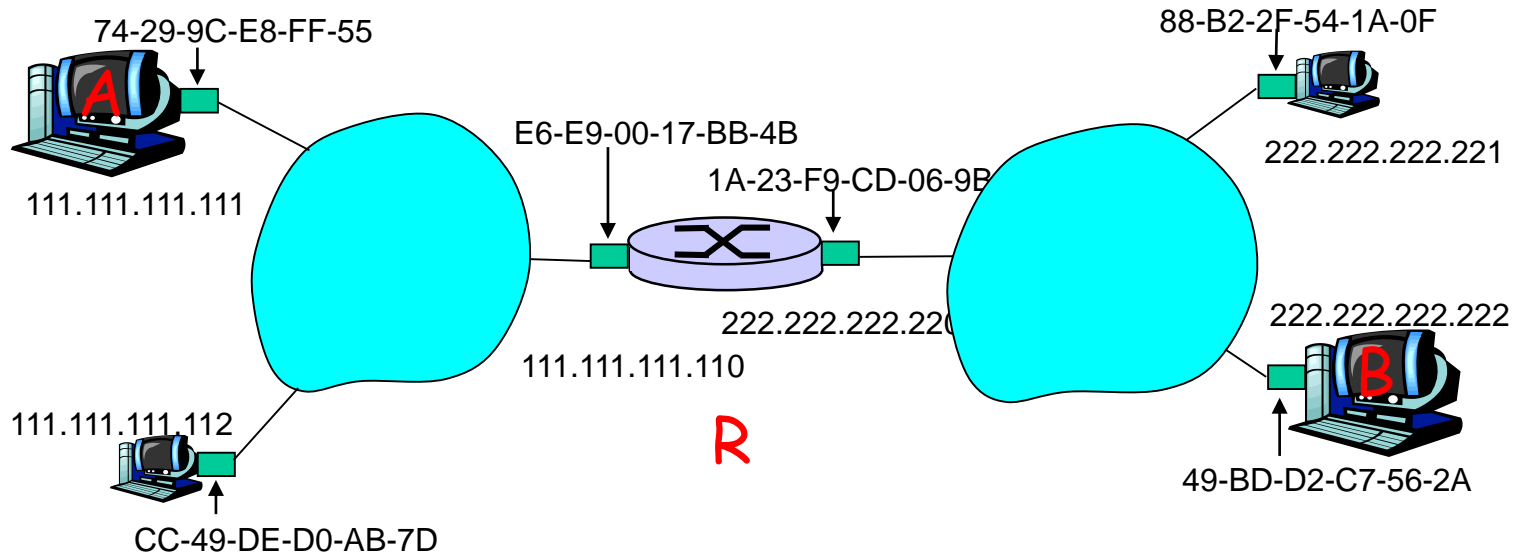


ARP 协议: 同一个 LAN

- A 希望发送数据报给 B, 但 B 的 **MAC** 地址没有在 A 的 **ARP** 表中.
- 1. A 广播 **ARP** 查询数据报, 包含 B 的 **IP** 地址
 - 目标 **MAC** 地址 =
FF-FF-FF-FF-FF-FF
 - **LAN** 中所有主机都收到 **ARP** 查询
- 2. B 接到 **ARP** 查询数据报, 使用自己的 (B) **MAC** 地址构造数据报响应 A
 - 发送到 A 的帧以 A 的 **MAC** 地址为目标 (**unicast**)
- 3. A 在 **ARP** 表中缓存 **IP-to-MAC** 地址对, 直到该信息过时
 - A 和 B 之间若没有数据传输在一段时间后删除该条目
- **ARP** 是即插即用
- “**plug-and-play**”:
 - 生成 **ARP** 表不需要设置

ARP 协议:不同 LAN

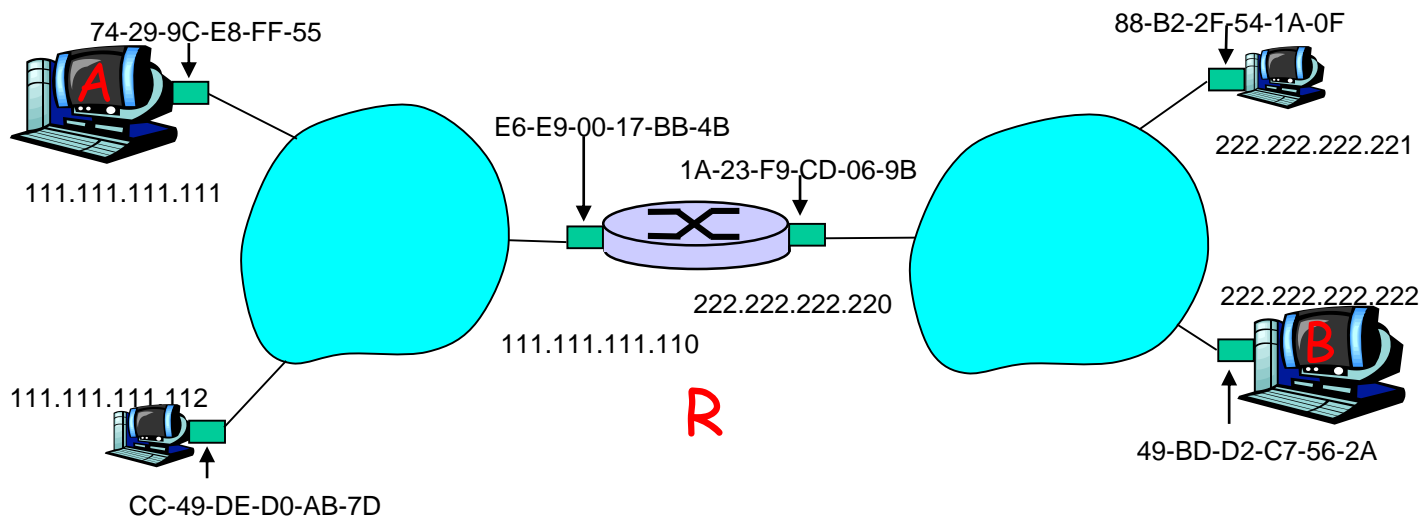
假设:从A发送数据报到B经过R
A 知道 B 的 IP 地址



- R的两个接口卡分别与LAN相连
两个接口卡中都存在相应的ARP 表,

- ❑ A 生成 IP 数据报: 源地址为 A, 目的地址为 B
- ❑ A 使用 ARP 找到 R 的 MAC 地址, 对应于 111.111.111.110
- ❑ A 生成链路层帧, 包含 R 的 MAC 地址为目的地址, 帧包含了 A-to-B IP 数据报
- ❑ A's NIC 发送帧
- ❑ R's NIC 接收帧
- ❑ R 解开 IP 从 Ethernet 帧中, 发现它的目的地是 B
- ❑ R 使用 ARP 找到的 MAC 地址
- ❑ R 生成链路层帧, 帧包含了 A-to-B IP 数据报

非常重要的知识点!



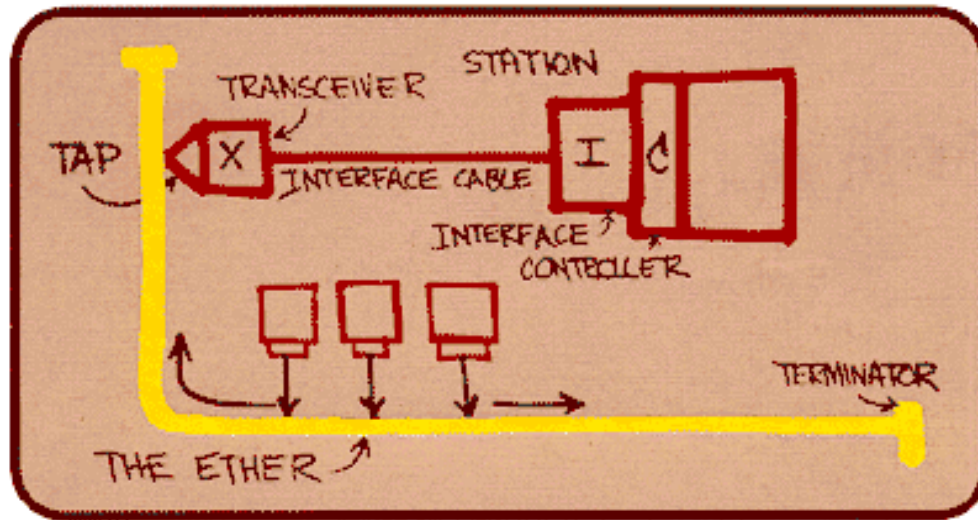
Chapter 5: Link Layer

- ❑ 5.1 概述和服务
- ❑ 5.2 检错和纠错
- ❑ 5.3 多路访问协议
- ❑ 5.4 链路层地址
- ❑ 5.5 以太网
- ❑ 5.6 链路层交换机
- ❑ 5.7 PPP
- ❑ 5.8 链路虚拟化: MPLS

Ethernet

完全占有有线局域网市场

- ❑ NIC便宜
- ❑ 最早使用的 LAN 技术
- ❑ 简单，相对于token LANs 和 ATM
- ❑ 跟上了发展速度： 10 Mbps - 10 Gbps

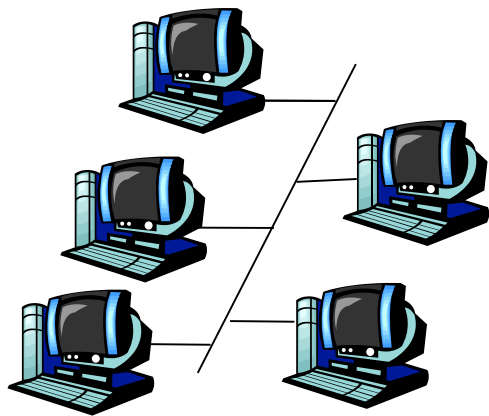


Metcalfe's Ethernet sketch

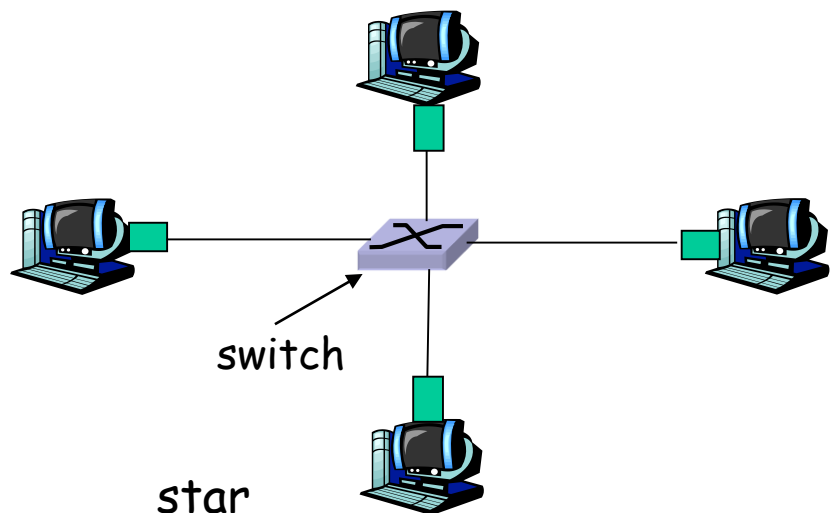
星型拓扑

Star topology

- ❑ 总线拓扑流行于上世纪 90s
 - 所有节点位于同一个冲突域
- ❑ 目前星型拓扑已是标准
 - 交换机(*switch*) 位于中心
 - 使用以太网协议(Ethernet protocol), 但是消除了冲突域



bus: coaxial cable

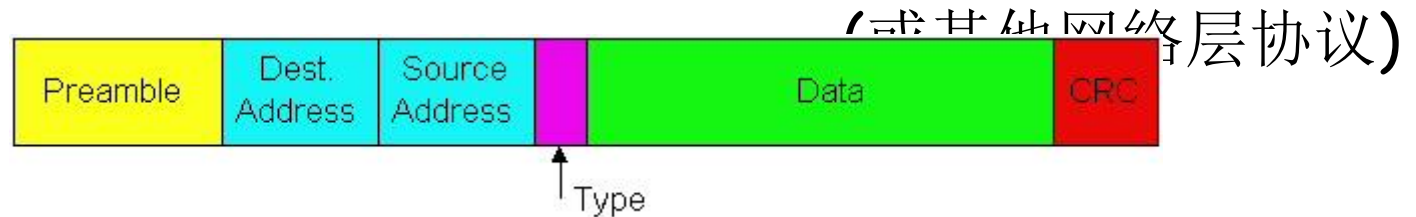


star

以太网帧结构

Ethernet Frame Structure

发送数据的接口卡在 **Ethernet** 帧中封装 IP 数据报



前同步码 **Preamble**:

- 7 个字节 10101010 后面跟着一个字节 10101011
- 用于同步 receiver, sender 时钟频率 (clock rates)

以太网帧结构

❑ 地址(Addresses): 6 bytes

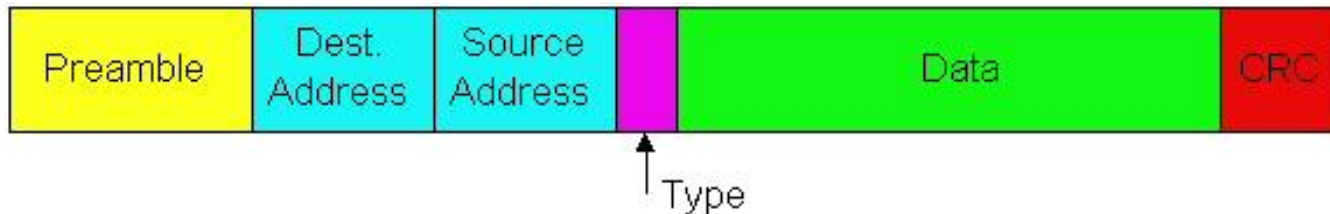
- 当适配器收到一个以太网帧，目的地址是它的**MAC**地址或者是广播地址，都将该帧的数据字段的内容传递给网络层
- 如果它收到了具有任何其他**MAC**地址的帧，则丢弃之。

❑ 类型(Type): 2 bytes

- 允许以太网复用多种网络层协议(e.g., Novell IPX)

❑ 循环冗余校验(CRC): 4 bytes

- 使得接收适配器检测帧中是否引入了差错



以太网：不可靠，无连接服务

Ethernet: Unreliable, connectionless

- ❑ 无连接(connectionless):
 - 发送和接收者之间没有握手过程
- ❑ 不可靠(unreliable):
 - 接收NIC 不发送 acks 或 nacks 给发送 NIC
 - 由于丢弃了以太网帧而可能存在间隙 (丢失了某些数据报)
 - TCP会察觉到这些间隙
- ❑ Ethernet的 MAC 协议: unslotted CSMA/CD

Ethernet CSMA/CD 算法

1. **NIC** 从网络层接收数据报, 准备以太网帧
2. 如果 **NIC** 侦听到信道空闲, 它开始传输该帧。
如果 **NIC** 侦听到信道忙, 它需等待到侦听不到信号, 然后开始传输该帧
3. 如果 **NIC** 传输了整个帧, 而没有监测到来自其他 **NIC** 的信号, 就完成了该帧的传输!
4. 如果 **NIC** 在传输中监测到来自其他适配器的信号, 中止传输, 而发送一个阻塞信号
5. 中止, **NIC** 进入指数后退 (**exponential backoff**):
在 m 次 碰撞后,
NIC 随机选定 $K \in \{0, 1, 2, \dots, 2^m - 1\}$.
NIC 等待 $K \cdot 512$ 比特时间, 退回到第二步

Ethernet CSMA/CD

阻塞信号(Jam Signal):

- 确保所有其他的传输中的适配器都意识到此次碰撞;
- 48 bits

比特时间(Bit time):

- 10 Mbps Ethernet 0.1 microsec ;
- $K=1023$,
等待时间约 50 msec

指数后退(Exponential Backoff):

- *Goal:*
- 评估当前负载后调整重传的时机
 - 负载较重时,
随机等待时间较长
- 第一次冲突:
从{0,1} 选择 K ;
延时 $K \cdot 512 \text{ bit}$ 传输时间
- 第二次冲突:
从{0,1,2,3} 选择 K ...
- ...
- 第十次冲突:
从{0,1,2,3,4,...,1023} 选择 K ...

CSMA/CD 效率

- T_{prop} = LAN中两节点间传播的最大时间
- t_{trans} = 传输一个最大以太网帧的时间

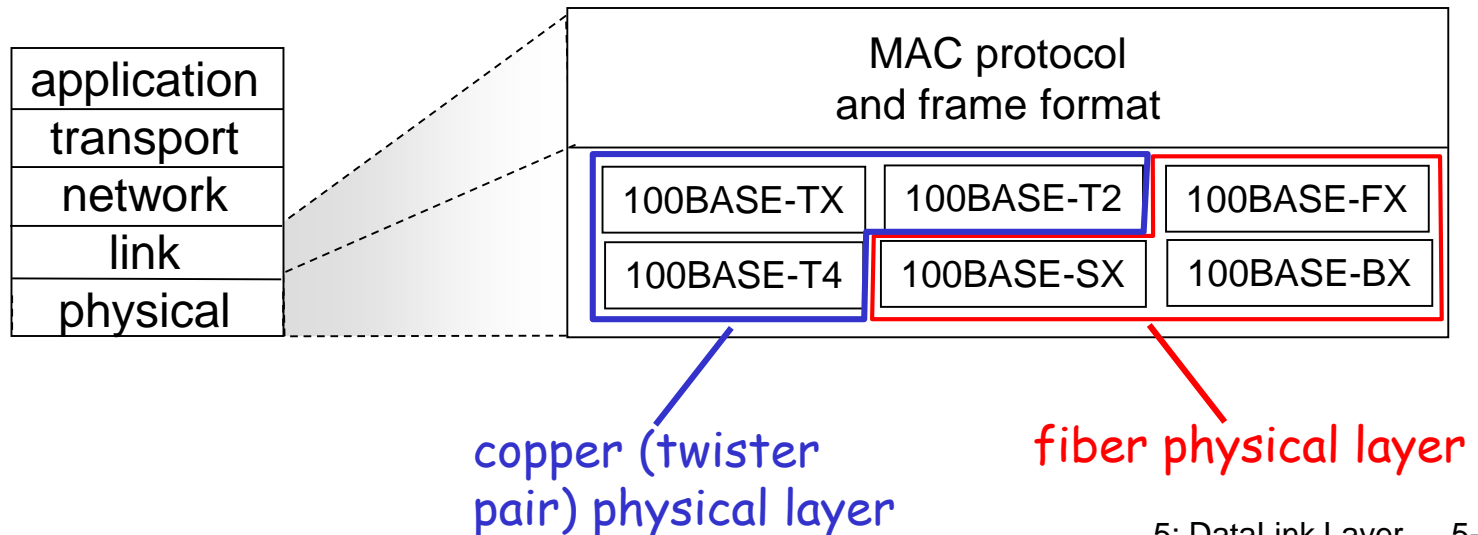
$$\text{efficiency} = \frac{1}{1 + 5t_{\text{prop}}/t_{\text{trans}}}$$

- 效率趋向于1
 - 当 t_{prop} 趋向于0
 - 当 t_{trans} 趋向于无穷
- 性能优于 **ALOHA**: 并且简单, 便宜, 分布式!

802.3 以太网标准: Link & Physical Layers

□ 有许多不同的以太网标准

- 相同的 **MAC** 协议和帧结构
- 不同速度: 10 Mbps, 100 Mbps, 1Gbps, 10G bps
- 不同物理介质: 光纤, 铜缆



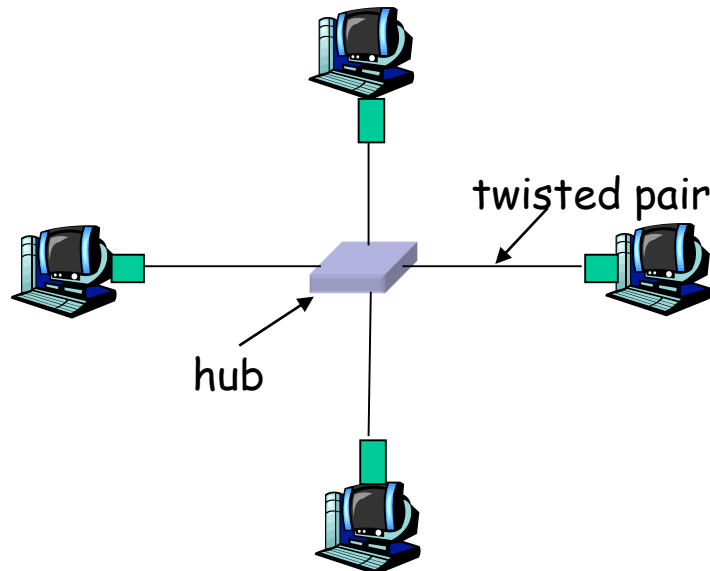
Chapter 5: Link Layer

- 5.1 概述和服务
- 5.2 检错和纠错
- 5.3 多路访问协议
- 5.4 链路层地址
- 5.5 以太网
- 5.6 链路层交换机
- 5.7 PPP
- 5.8 链路虚拟化: MPLS

Hubs

... 物理层("dumb") 中继器:

- bits 从一个接口进来, 从相同速度**所有**其它接口出去
- 与Hub相连所有节点位于同一个冲突域
- 没有帧缓冲
- Hub没有 CSMA/CD : 主机 NICs 检测冲突



交换机Switch

❑ 链路层设备:

- 存储, 转发以太网帧
- 检查进来帧的 **MAC** 地址, 选择性的(**selectively**) 转发帧到一个或多个出去的接口

❑ 透明性(**transparent**)

- 主机意识不到交换机的存在

❑ 即插即用和自学习(**plug-and-play, self-learning**)

- 交换机基本不需要配置

Switch: 允许多个节点同时传输

- ❑ 主机与 switch 有专用连接
- ❑ switch 缓冲数据报
- ❑ Ethernet 协议用于每一个连接

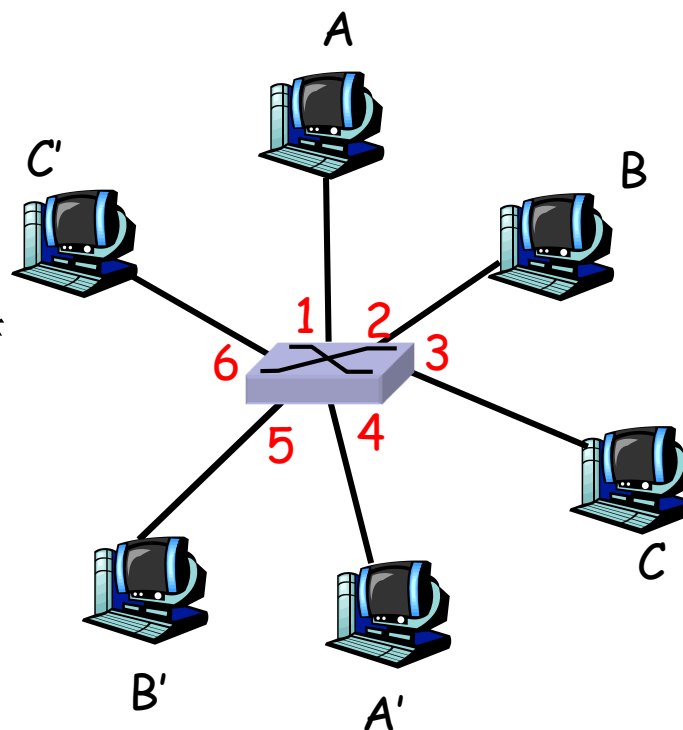
但是**没有冲突**; 全双工通信

- 每个链路有自己的冲突域

❑ *switching:*

A-to-A' and B-to-B' 同时传输,
没有冲突

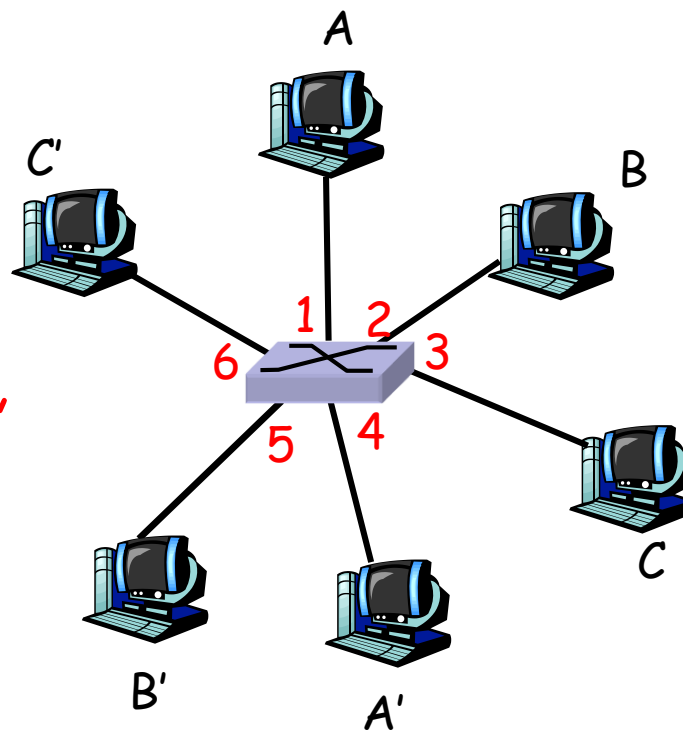
- 与 Hub 完全不一样



*switch with six interfaces
(1,2,3,4,5,6)*

Switch Table

- **Q:** switch 如何知道
A' 在接口4可达,
B' 在接口5 可达?
- **A:** switch 有一个 **switch table**,
其中每个条目包含:
 - (主机**MAC** 地址,
 - 主机的可达端口,
 - 时间戳)
- 看起来像路由表!
- **Q:** **switch table**如何建立和维护?
 - 有点像路由表?



*switch with six interfaces
(1,2,3,4,5,6)*

自学习

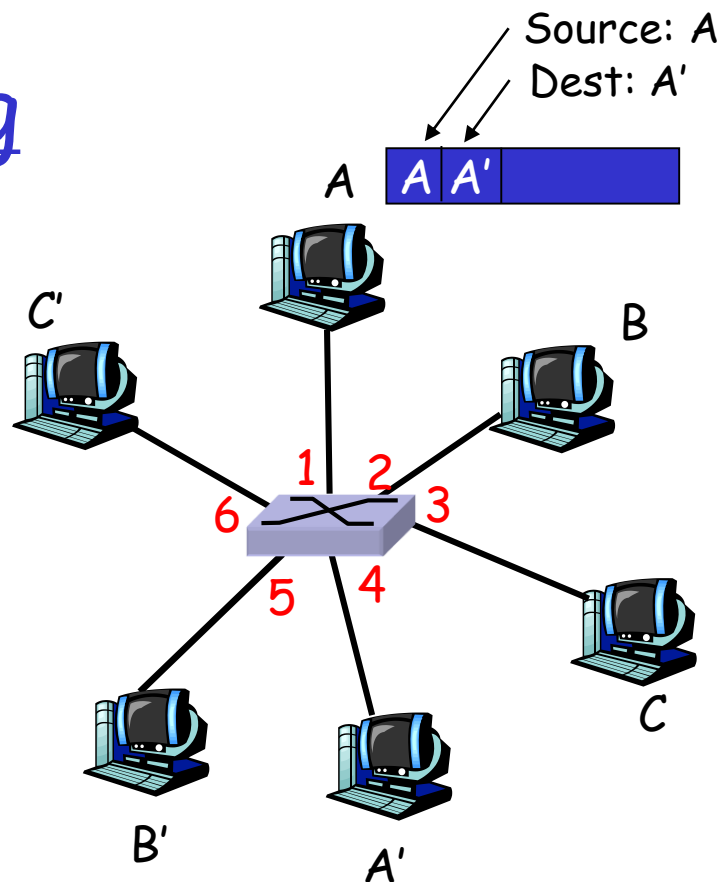
Switch: self-learning

□ switch 学习 哪个主机可以通过哪个端口到达

- 当一个帧到达交换机, 交换机“学习” 发送者位置: LAN 网段
- 在switch table记录 sender/location

MAC addr	interface	TTL
A	1	60

Switch table
(initially empty)



交换机: 帧过滤和转发

Switch: frame filtering/forwarding

When frame received:

1. 对接口 x 接收到的每个入帧, 交换机在其表中存储
2. 使用 **MAC** 目标地址进行索引
3. **if** 表中有个表项将接口 y 与**MAC**目标地址联系起来

then {

if $y=x$

then 丢弃该帧

else 转发到接口 y 相应网段

}

else 广播该帧



除接收到该帧端口外全部转发!

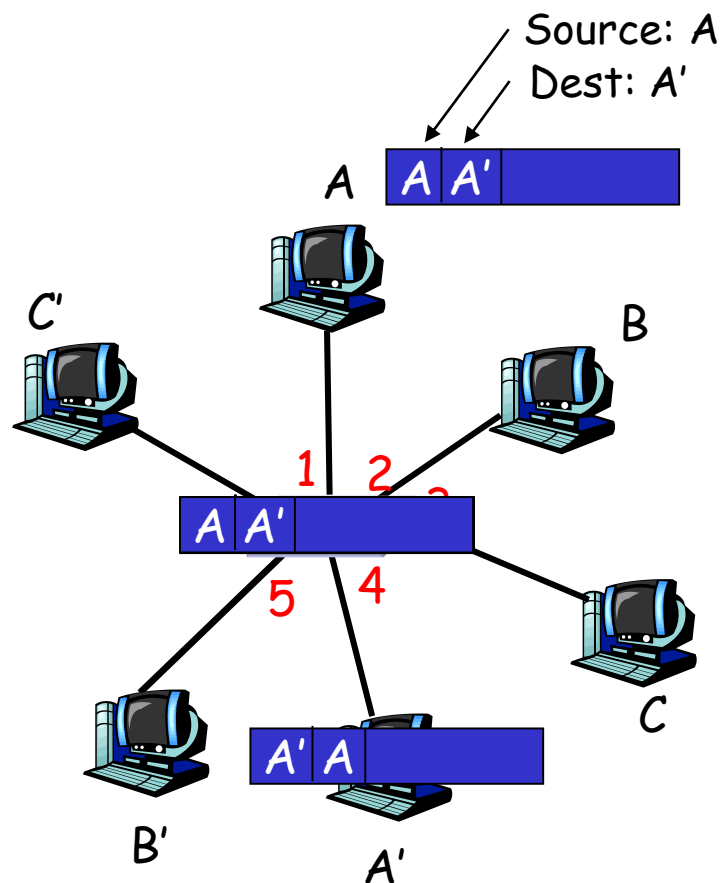
自学习, 前向转发例子:

❑ 目的地未知:

flood

❑ 目标 **A** 位置已知:

selective send

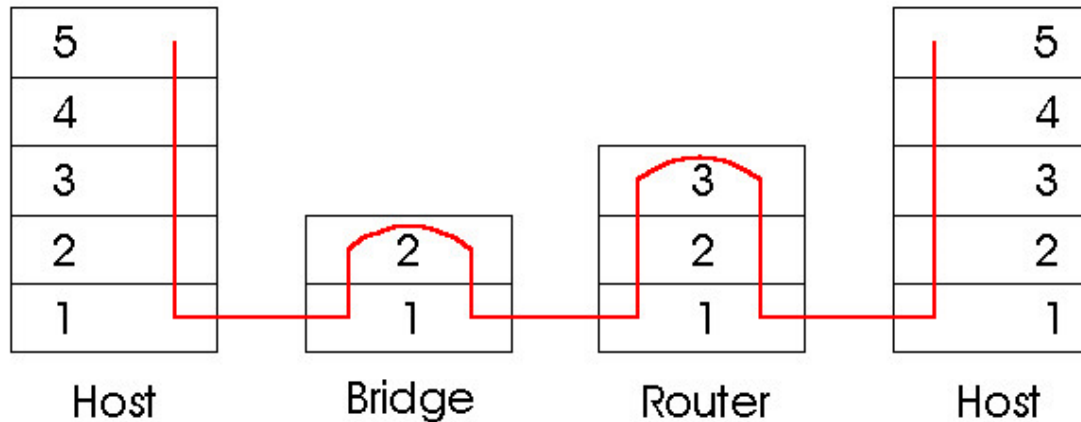


MAC addr	interface	TTL
A	1	60
A'	4	60

*Switch table
(initially empty)*

Switches vs. Routers

- ❑ 都是存储转发设备
 - **routers**: 网络层设备(检查网络层头部, **IP**地址)
 - **switches**: 链路层设备(检查链路层层头部, **MAC**地址)
- ❑ 路由器维护 **路由表**, 使用路由算法
- ❑ 交换机维护 **交换表**, 实施过滤, 学习算法



Chapter 5: Link Layer

- ❑ 5.1 概述和服务
- ❑ 5.2 检错和纠错
- ❑ 5.3 多路访问协议
- ❑ 5.4 链路层地址
- ❑ 5.5 以太网
- ❑ 5.6 链路层交换机
- ❑ 5.7 PPP
- ❑ 5.8 链路虚拟化: MPLS

点对点协议

Point to Point Data Link Control

- ❑ 只有一个发送者, 一个接收者, 一条链路:
比广播链路简单:
 - 没有介质访问控制
 - 没有寻址
 - e.g., 拨号网络, ISDN 线路
- ❑ 常用的 point-to-point DLC 协议:
 - PPP (point-to-point protocol)
 - HDLC (High level data link control)

PPP 设计要求[RFC 1557]

❑ 分组成帧:

- 链路层的发送方必须封装在PPP链路层帧中, 以便接收方接收
- 能够携带网络层的分组
- 能够向上多路分解

❑ 透明性:

- 不能对出现在网络层分组中的数据(首部或者数据)做任何限制

❑ 差错检测:

- 接收方必须能够检测到接收帧中的比特差错

❑ 连接的活性:

- 能够检测链路活性、发出信令:

❑ 网络层地址协商:

- 为网络层(例如IP)提供一个机制, 来获知或配置相互的网络层地址

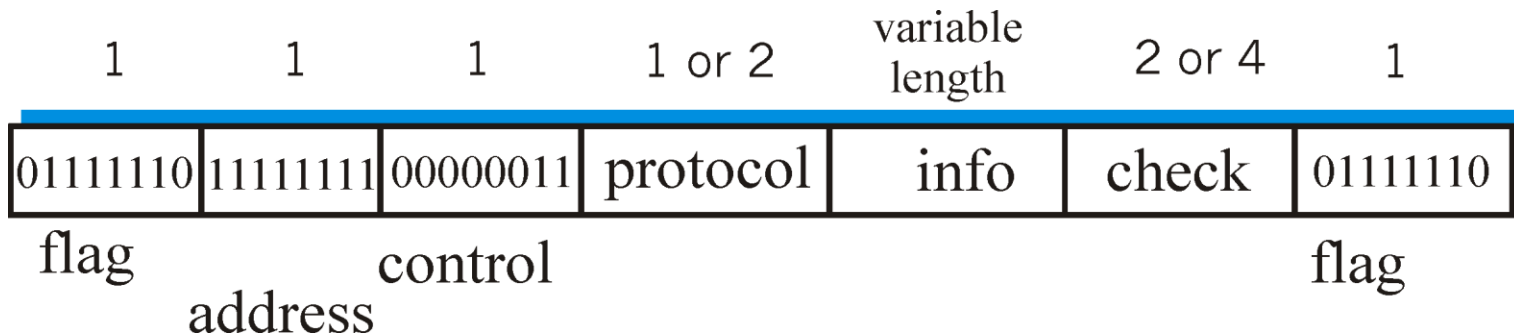
PPP 不要求

- ❑ 没有差错纠正
- ❑ 没有流量控制
- ❑ 不需要有序性
- ❑ 不需要支持多点链路

错误恢复, 流量控制, 数据顺序性保证
都由高层协议实现!

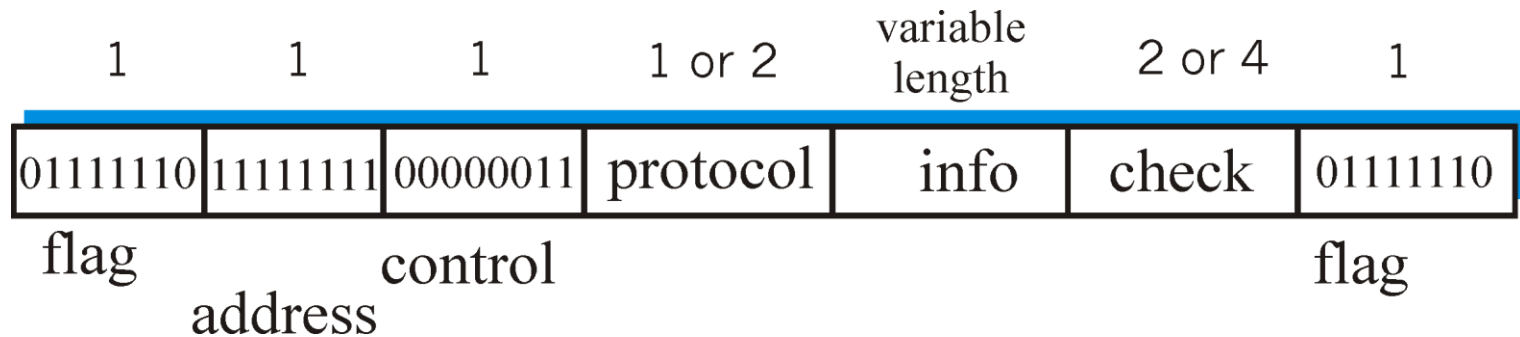
PPP数据格式

- ❑ 标志(Flag):
- ❑ 地址(Address):
- ❑ 控制(Control):
- ❑ 协议(Protocol):
 - 告诉PPP接收方所接收的数据所属的上层协议
 - (eg, PPP-LCP, IP, IPCP, etc)



PPP数据格式

- ❑ 信息(Info):
- ❑ 检验和(Check):



字节填充

Byte Stuffing

❑ “数据透明”

禁止上层协议发送包含标志字段比特模式的数据，
如 <01111110>

○ Q: 接收到<01111110> 怎么办?

❑ **Sender:** 在< 01111110>**数据**字节前添加额外的< 01111110>

❑ **Receiver:**

○ 两个 01111110 字节: 丢弃第一个, 接收第二个

○ 单个01111110: 标志字节

Byte Stuffing



PPP 控制协议

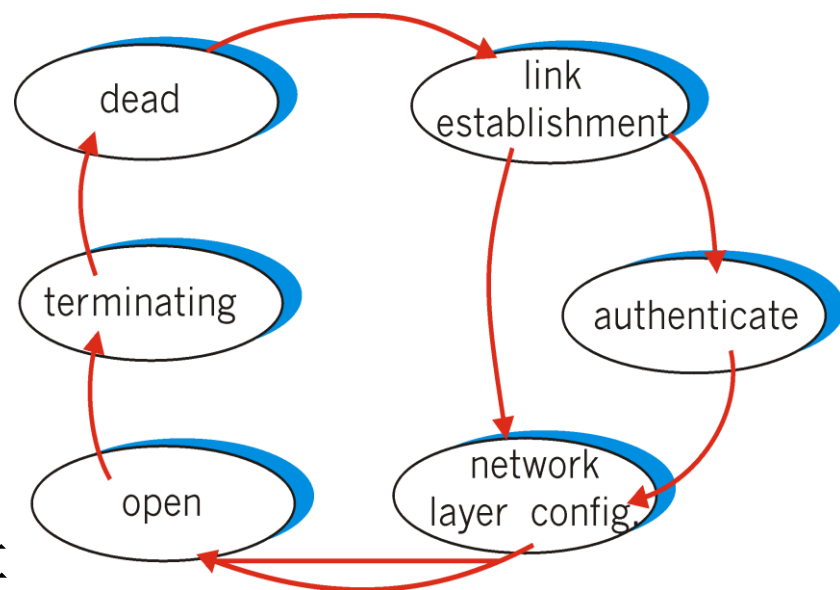
在交换网络层数据前, 链路层先要

❑ 配置PPP连接:

- 最大帧长度
- 认证(authentication)

❑ 学习配置网络层信息:

- IP: IPCP消息
(协议字段: 8021) 配置IP地址



Chapter 5: Link Layer

- ❑ 5.1 概述和服务
- ❑ 5.2 检错和纠错
- ❑ 5.3 多路访问协议
- ❑ 5.4 链路层地址
- ❑ 5.5 以太网
- ❑ 5.6 链路层交换机
- ❑ 5.7 PPP
- ❑ 5.8 链路虚拟化: MPLS

网络虚拟化

资源的虚拟化:

系统工程中的一种强大的管理策略!

- ❑ 计算机中的实例: virtual memory, virtual devices
 - Virtual machines: e.g., java
 - VMWare machines
- ❑ 层次的抽象:
 - 对下一层硬件具体实现不关心
 - 只对其进行抽象

Cerf & Kahn's Internet架构

What is virtualized?

- 两层寻址：网间网和本地网
- **IP**层使异质网络互联成为可能

下一层网络技术可以是：

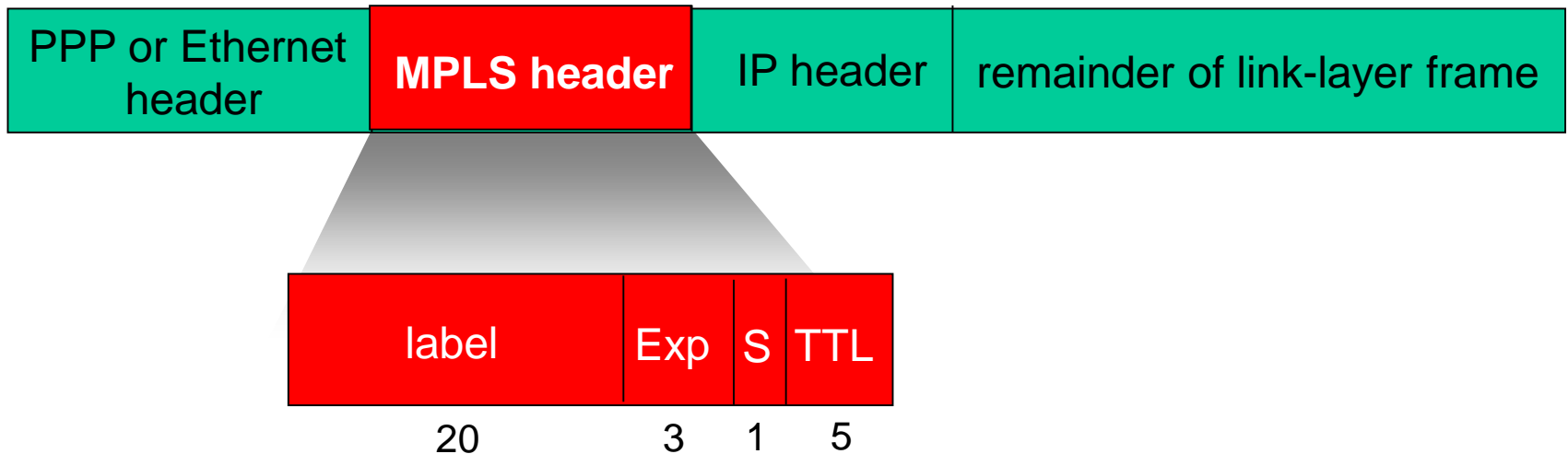
- 卫星
- 电话网网络
- 以太网
- **MPLS**

... 对**IP**来说它们都是不可见的，仅仅是一种链路层技术!

多协议标记交换

Multiprotocol label switching (MPLS)

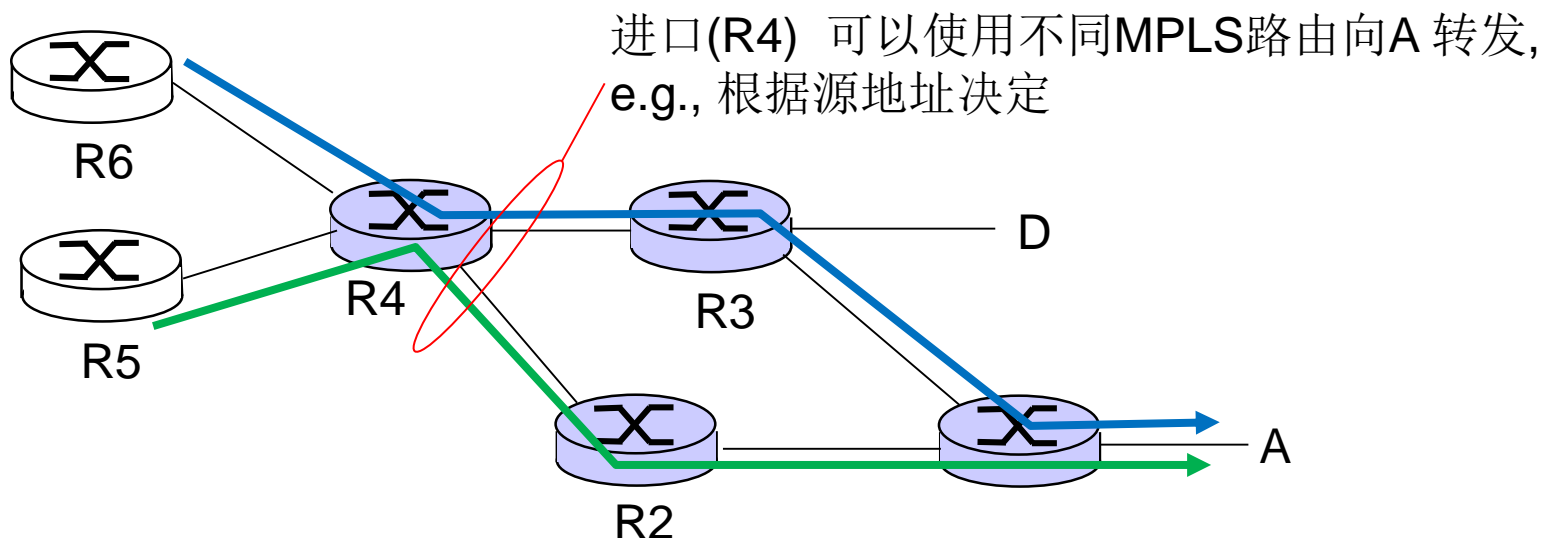
- ❑ 目标: 使用固定长度标签, 可以改善 **IP** 路由器的转发速度
 - 从 **Virtual Circuit (VC)** 策略中得到借鉴
 - **IP** 仍然保持**IP**地址!



MPLS 标签交换路由

- ❑ 标签交换路由
- ❑ 通过在转发表中查找**MPLS**标签来转发**MPLS**帧
 - **MPLS**转发表与**IP**转发表不同
- ❑ 前向转发通道的建立需要专门的信令协议
 - **RSVP-TE**
 - 前向转发路径可能不同于**IP** 转发路径!!
 - 可以使用 **MPLS** 进行流量工程
- ❑ 需要与普通**IP**路由并存

MPLS 与 IP 路径



□ IP 路由:

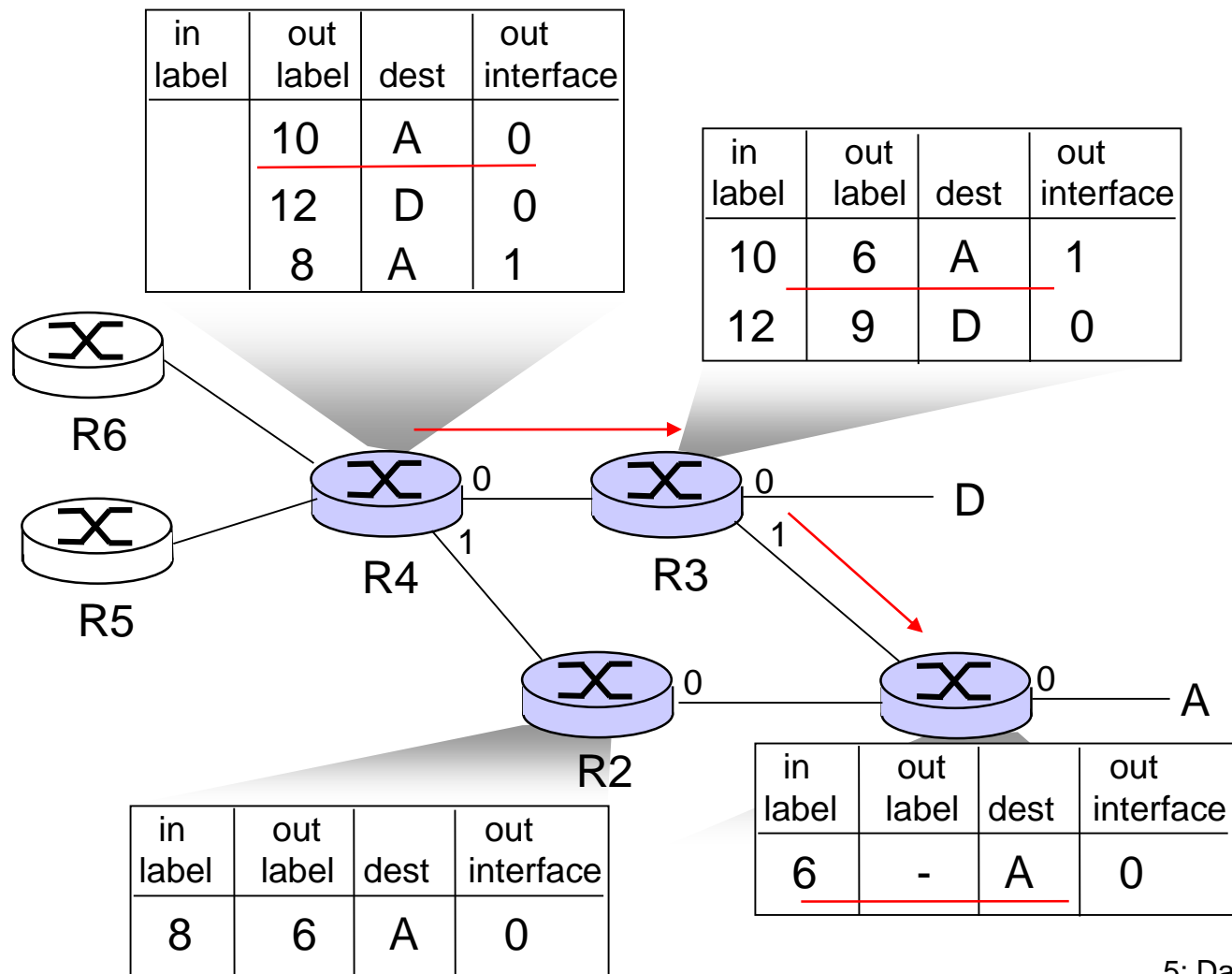
- 路径由目的地**IP**地址决定

□ MPLS 路由:

- 路径可以根据某种策略由源、目的**IP**地址决定
- 可以进行快速路由:
预先设置备份路由, 应对链路错误



MPLS 前向转发表



Chapter 5: 小结

- ❑ 链路层服务的原理:
 - error detection, correction
 - sharing a broadcast channel: multiple access
 - link layer addressing
- ❑ 不同链路层技术的实现
 - Ethernet
 - PPP
 - virtualized networks as a link layer: MPLS