Ch6.Link layer and LANs

6.0 Background

目标

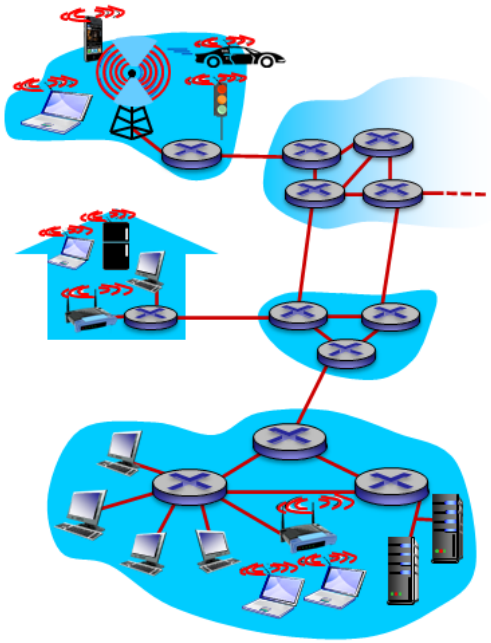
- error detection, correction
- sharing a broadcast channel: multiple access
- link layer addressing
- local area networks: Ethernet, VLANs

各种链路层技术的实例化、实现

6.1 Introduction

术语

- hosts和routers: nodes
- 沿着通信路径连接相邻节点的通信信道: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, 封装了datagram



data-link layer:将datagram从一个node传输到链路上物理上相邻的node

Link Layer：相邻的两个节点之间的数据传输

内容

- 不同的链路协议在不同的链路上传输datagram:
  - 例如: Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- 每种链路协议提供不同的服务
  - Link上可能也可能不提供rdt

即使提供了rdt，仍然需要在端到端上进行校验

提供服务

核心功能

- framing
- link access

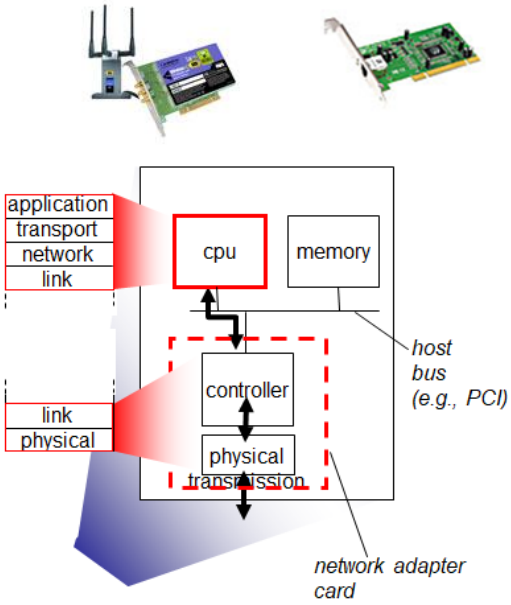
- *framing, link access* 包装，链路访问:
  - 将datagram封装到帧frame中，添加报头、尾部
  - 如果是共享设备，如何考虑信道占有
  - 在frame报头中用来识别源、目的的MAC地址
    - 与IP不同！
- *reliable delivery between adjacent nodes*
  - 很少用于低误码链路(光纤，一些双绞线)
  - 无线连接wireless link:高误码率
    - *Q:* why both link-level and end-end reliability?
    - *A:* 为了增强链路可靠性，减少端到端检验错误后要求重传的麻烦

- *flow control:*
  - 在相邻的发送和接收节点之间进行停顿
- *error detection:*
  - 由信号衰减、噪声引起的误差
  - 接收方检测到错误的存在:
    - 重传或丢弃frame
- *error correction:*
  - 接收方在不求助于重传的基础上辨别并纠正比特错误
- *half-duplex and full-duplex*
  - 在半双工的情况下，链路两端的节点可以进行传输，但不能同时进行

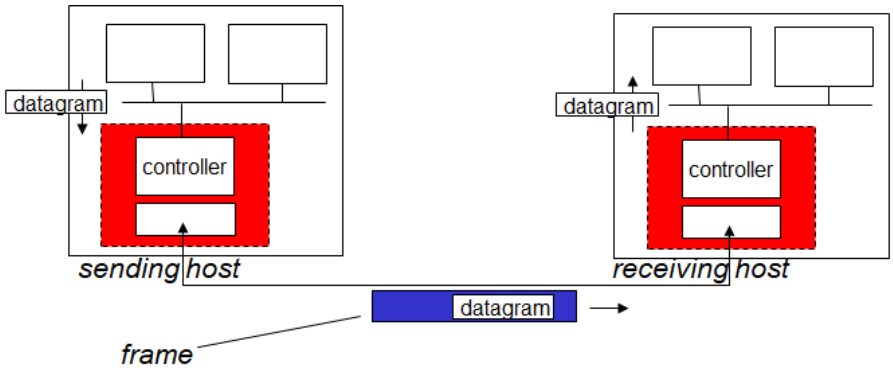
Link Layer实现位置

- 在每一个host
- 链路层在“适配器”(又名网络接口卡 network interface card NIC)或芯片chip上实现
  - 以太网卡，802.11卡;以太网芯片
  - 实现链路，物理层
- 连接到主机的系统总线
- 硬件、软件、固件的组合

- 一部分在操作系统中实现
- 一部分在网卡（硬件）中实现



Link Layer 数据传输交流（在适配器Adapter实现）



- 发端:
  - 把datagram封装成frame
  - :增加错误检查位，rdt，流量控制等
- 收端
  - 查找错误、rdt、流量控制等
  - 提取datagram，传递到接收侧的上层

6.2 error detection, correction

错误检查

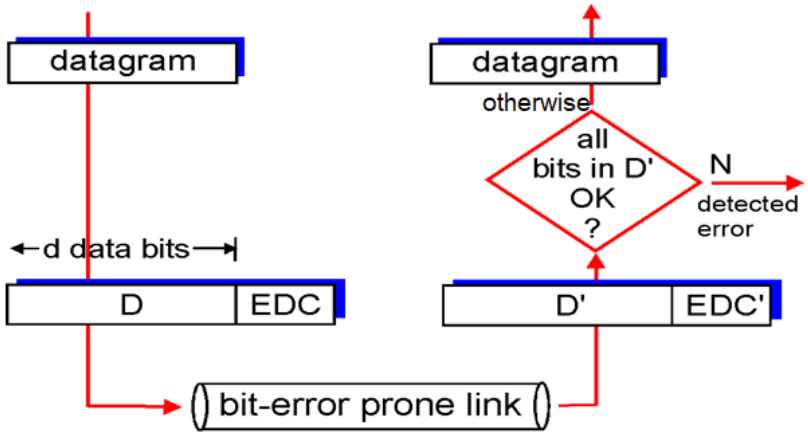
EDC = Error Detection and Correction bits 错误检测和校正位 (redundancy)

D = 受错误检查保护的数据，可能包括报头字段

• Error detection not 100% reliable!

•协议可能会漏掉一些错误，但很少

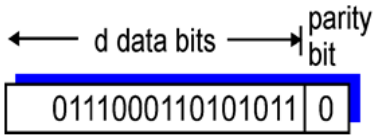
•更长的EDC字段产生更好的检测和校正



奇偶校验

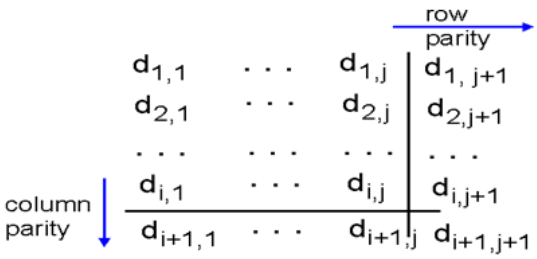
一维奇偶校验:

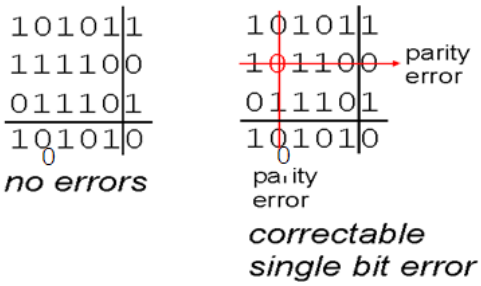
- *检测单比特错误*



二维位奇偶校验:

- 检测和纠正单比特错误





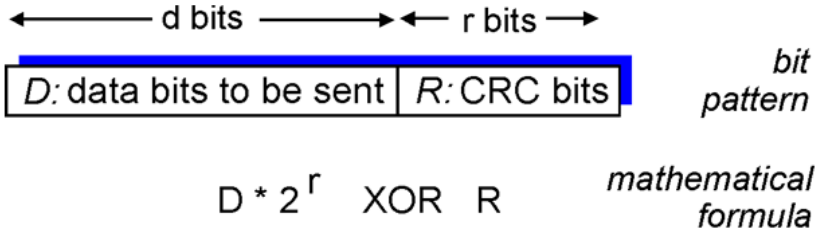
一维奇偶校验只能知道这一串数据是否有错，不能纠正

二维奇偶校验只能检测一个bit的错误，如果有两个bit出错则无法正确检查

Check sum（用于传输层）

循环冗赘核对 Cyclic redundancy check （CRC）

- 更强大的错误检测编码
- 将数据D看作一个二进制数
- 选择r+1 bit 从生成器，G
- goal: 选择 r CRC bits, R, such that
  - <D,R> 被G整除
  - 接收方知道G, 用G除以<D,R>
  - 如果非零余数:检测到错误!
  - 能检测到小于r+1比特的所有突发错误bit
- 广泛应用于实践(以太网、802.11 WiFi、ATM)



示例

want:

$D \cdot 2^r$ XOR R = nG

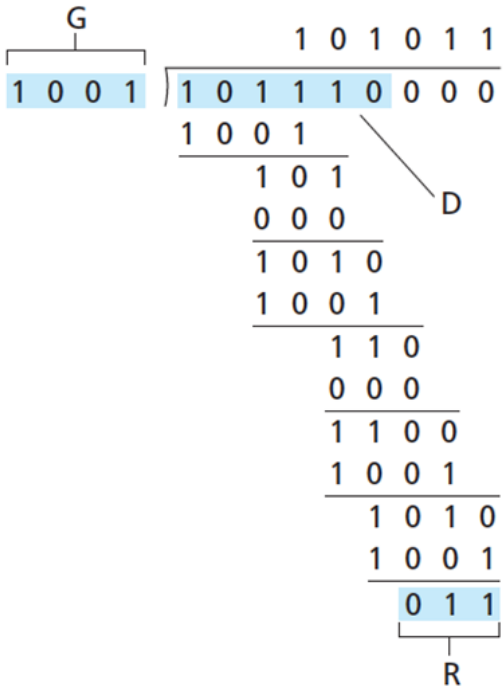
equivalently:

$D \cdot 2^r$ = nG XOR R

equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R to satisfy:

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



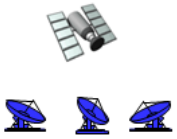
6.3 multiple access protocols

两种link类型

- 点对点
  - PPP用于拨号接入
  - 以太网交换机、主机之间的点对点链路
- 广播(共用电线或媒体)
  - 老式的的以太网
  - 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)

多路存取协议（MAC）介绍

- 单共享广播信道
- 节点同时进行两次或两次以上的传输:干扰
  - *collision*如果节点同时接收到两个或多个信号

*multiple access protocol*多址协议

- 确定节点如何共享信道的分布式算法，即确定节点何时可以传输
- 关于频道共享的沟通必须使用频道本身!
  - 没有带外协调通道

给定：广播信道速率为R bps

希望:

1.当一个节点想要传输时，它可以以R的速率发送

2.当M个节点需要发送时，每个节点的平均发送速率为R/M

3. 完全去中心化:
   - 没有特殊的节点来协调传输
   - 没有同步的时钟，slot

4. 简单

MAC的三种类型

三大类:

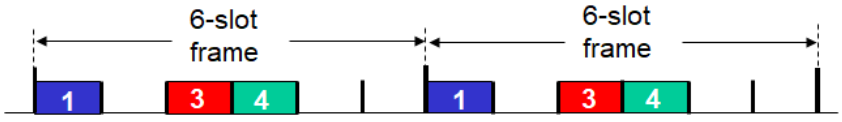
- *channel partitioning*
  - 将信道分成更小的“片段”(时隙、频率、代码)
  - 将块分配给节点单独使用
- *random access*
  - 信道未分割，允许碰撞
  - 从碰撞中“恢复”
- *“taking turns”*
  - 节点是轮流的，但是要发送的节点可能需要更长的时间

Channel Partitioning

TDMA

TDMA: time division multiple access时分多址

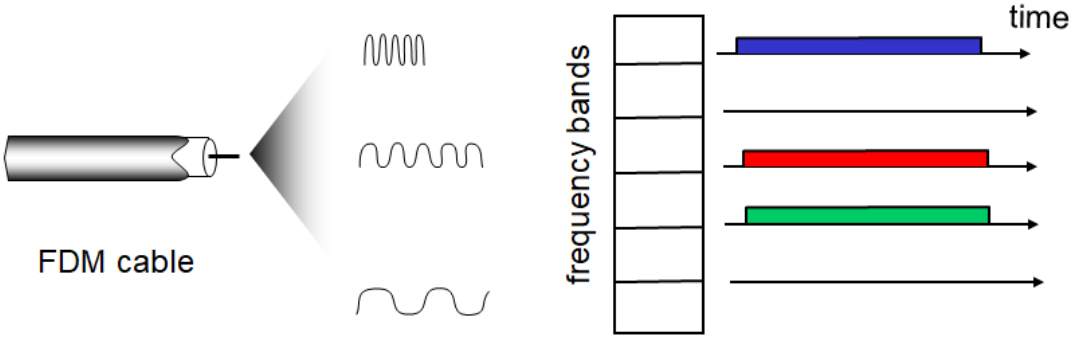
- “轮流”进入通道
- 在每一轮中每个站都有固定长度的slot (length = packet transmission time)
- 未使用的slot空闲
- 例: 6-station LAN, 1,3,4 有包要发，slots 2,5,6 空闲



FDMA

FDMA: frequency division multiple access频分多址

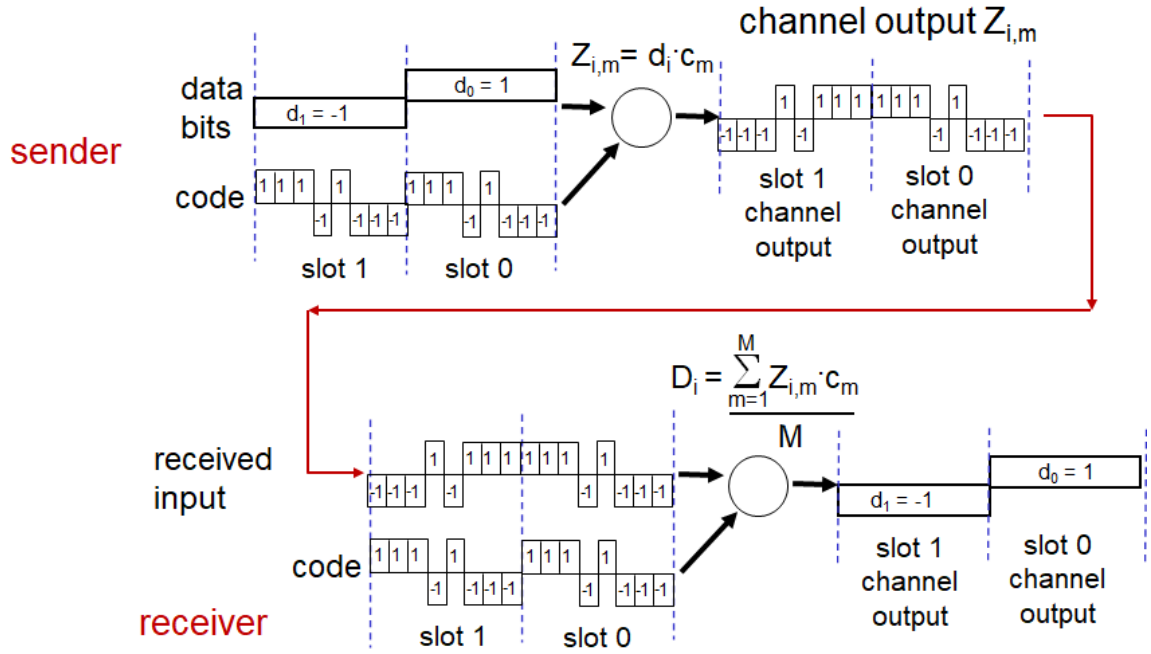
- 信道频谱分为频带
- 每个站被分配固定的频带
- 频带中未使用的传输时间空闲
- 例如: 6-station LAN, 1,3,4 有包要发，频带 2,5,6 空闲



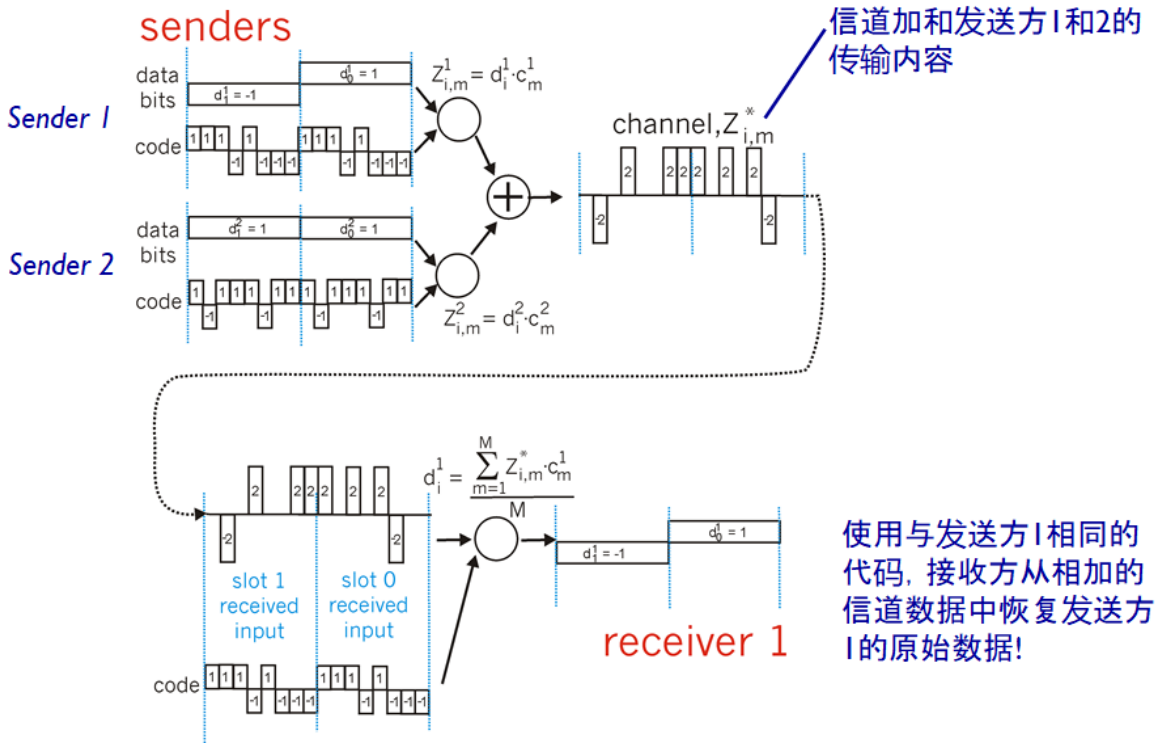
CDMA

- 分配给每个用户的唯一“代码”;例如，代码集分区
  - 所有用户共享相同的频率，但每个用户都有自己的“chipping”序列(即代码)来编码数据
  - 允许多个用户“共存”并以最小的干扰(如果代码是“正交的”)同时传输
- *encoded signal* = (original data) X (chipping sequence)
- *decoding*: coded signal and chipping sequence的内积

编解码思路



两个发送方的CDMA运用思路



Random access MAC protocol

- 当节点有数据包要发送时
  - 以全信道数据速率R传输
  - 节点之间不存在先验协调
- 两个或多个传输节点会导致“碰撞”
- random access MAC protocol 说明:
  - 如何检测碰撞
  - 如何从碰撞中恢复(例如，通过延迟重传)
- random access MAC protocols的示例:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

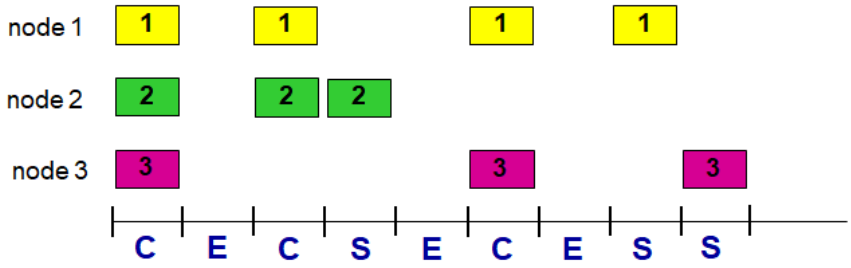
Slotted ALHOA

假设:

- 所有frame大小相同
- 时间分成相等大小的slot的时间(传输一个frame的时间)
- 节点开始只在slot起点开始传输
- 节点同步
- 如果slot中有2个或2个以上的节点传输数据，则所有节点都检测冲突

操作:

- 当一个节点获得新的frame时，在下一个slot传输
  - *如果没有collision:*节点可以在下一个slot发送新frame
  - *如果collision:* 节点用prob在每个后续slot中重新传输frame。p直到成功



优点:

- 单个有源节点可以在全信道速率下连续传输
- 高度分散:只有节点中的slot需要同步
- 简单

缺点:

- collisions, 浪费slots
- 空闲slots
- 节点需要在少于传输一个包的时间内检测到碰撞
- 如何满足时钟同步

效率:长期能够成功传输的slot的部分(有很多节点且节点与很多frame要发)

- 假设: N个节点有许多frame要发送，每个节点在slot中传输的概率为p
- 给定节点在slot中成功传输的概率= $p(1-p)^{N-1}$
- 任何节点都有可能成功的概率= $Np(1-p)^{N-1}$

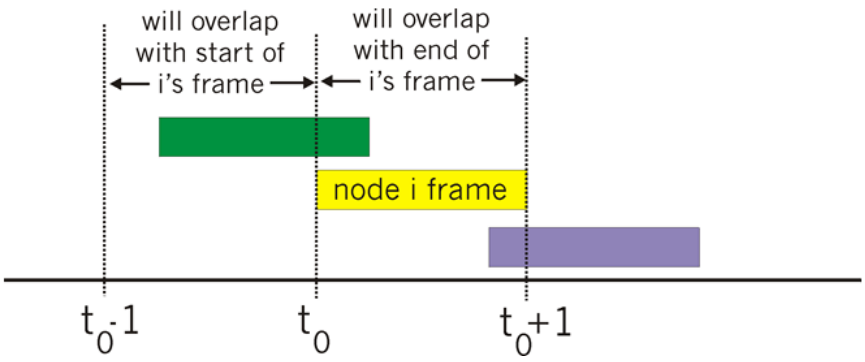
- 最大效率:找到最大化的p*使得$Np(1-p)^{N-1}$最大
- 对于多个节点，当N趋近于无穷时，取$Np^*(1-p^*)^{N-1}$的极限
- *max efficiency = 1/e = 0.37*

最好效果:频道使用的有用传输时间为37%！

!

Unslotted（pure） ALHOA

- unslotted Aloha:更简单，没有同步
- 当frame到达时
  - 立刻传输
- 碰撞概率增加:
  - 在t_0发送的frame与其他在$[t_0-1,t_0+1]$发送的frame发生冲突



P(给定节点成功) = P(节点传输)·P(没有节点在$[t_0-1,t_0]$传输)·P(没有节点在$[t_0,t_0+1]$传输)

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

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

…选择最优p，然后让$n \to \infty$

$= 1/(2e) = 0.18$

比slotted Aloha更差

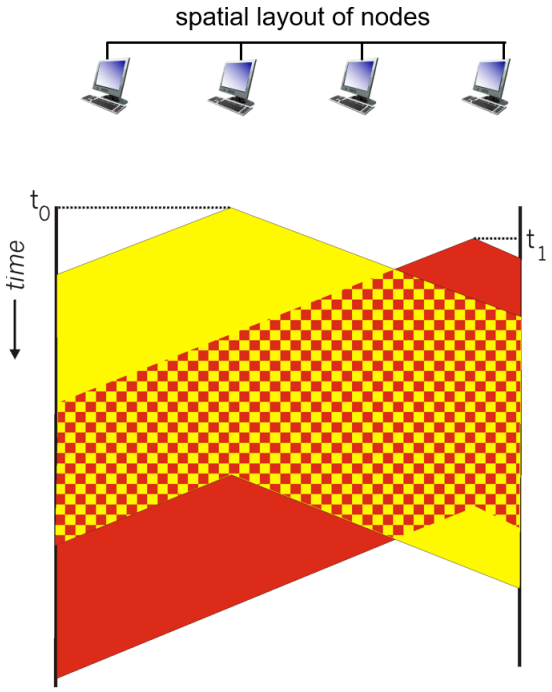
CSMA (carrier sense multiple access)

CSMA: 在传输前先监听:

- 如果感知到信道空闲:传输整个frame
- 如果信道感知忙:延迟传输

- 不打断别人!

- 碰撞仍然可能发生: 传播延迟意味着两个节点可能无法听到对方的传输
- collision: 整个数据包传输时间浪费
  - 距离和传播延迟是决定碰撞概率的重要因素

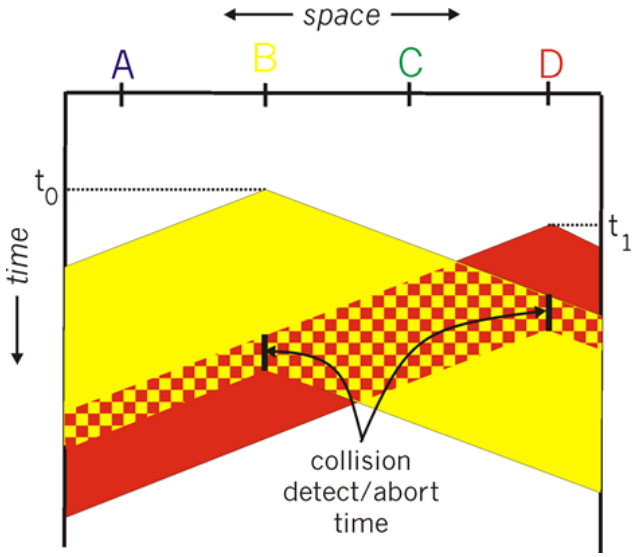


CSMA/CD（collision detection）

CSMA/CD: 在CSMA也具有感知和延迟

- 在短时间内检测到碰撞
- 碰撞的话传输中止，减少信道浪费

- 碰撞检测:
  - 有线局域网易于检测:测量信号强度，比较传输信号和接收信号
  - 无线局域网难于检测:接收的信号强度超过了本地传输强度



以太网CSMA/CD算法

1. NIC从网络层接收datagram，创建frame
2. 如果NIC检测到通道空闲，启动frame的传输。如果NIC检测到通道忙，等待通道空闲，然后传输。
3. 如果NIC传输整个frame的时候没有检测到另一个传输，NIC完成了一个frame的传输
4. 如果NIC在传输过程中检测到其他传输，则中止传输并发送干扰信号
5. 终止后，NIC进入*binary (指数) backoff*:
   - 在第m次碰撞后，NIC选择一个在区间 {0, 1, 2, …, 2^m-1} *中的随即数*K. NIC希望K·512 bit时间后，回到Step 2
   - 更多的碰撞，更长的backoff

- T_{prop} = LAN中2个节点之间的最大prop延迟
- t_{trans} =发送最大的frame的时间

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

- 效率趋于1
  - 当t_{prop} 趋于0
  - 当t_{trans} 趋于无穷
- 性能优于ALOHA:且简单、便宜、分散

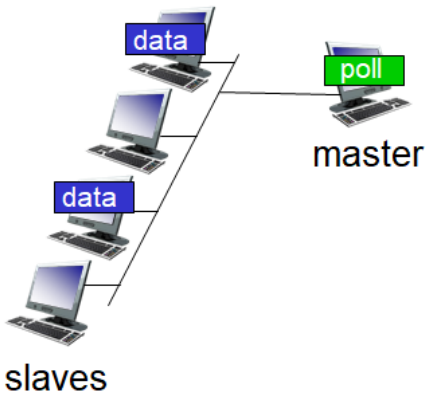
Taking turns

polling

前面100个都不传的话，很浪费时间

polling给予:

- 主节点“邀请”从节点依次进行传输
- 通常与“哑”从设备一起使用
- 关注:
  - polling的开销
  - 潜在因素
  - 单点故障(主设备)

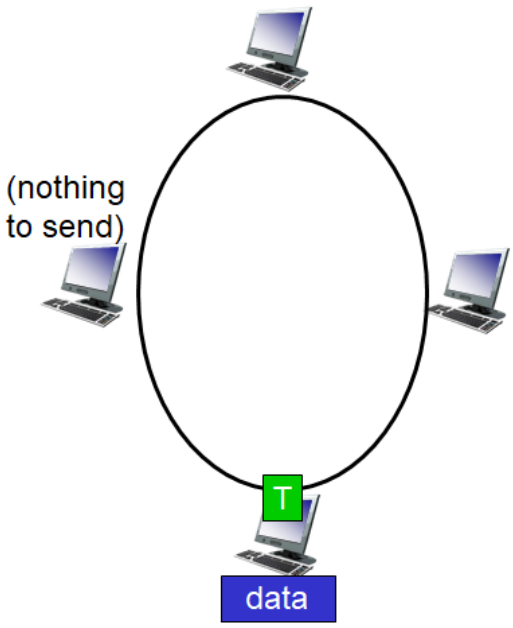


token passing

开始大家随机生成一个包（在某一个node上）
后续谁有这个包谁才可以发

token passing令牌传递:

- 控制token按顺序从一个节点传递到下一个节点。
- token信息
- 关注:
  - token的开销
  - 潜在因素
  - 单点故障(token)



总结

- *channel partitioning*,
  - TDMA,FDMA,CDMA
- *random access* (动态),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - 负载感知:有些技术容易(有线)，有些技术难(无线)
  - 以太网使用CSMA/CD
  - 802.11（WiFi）使用CSMA/CA
- *taking turns*
  - 从主设备分配，token传递
  - Bluetooth, FDDI，token ring

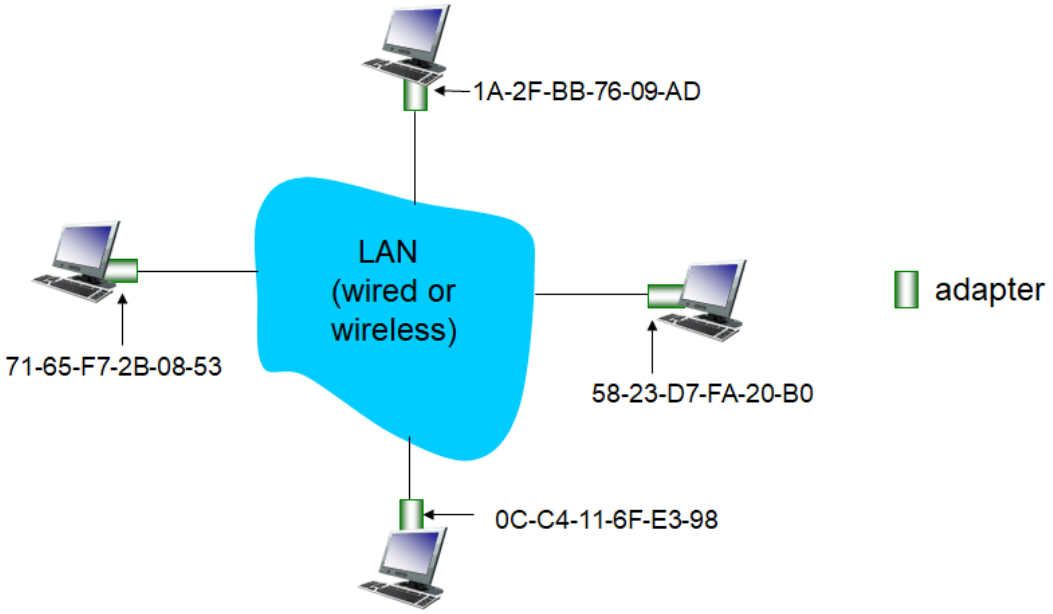
6.4 局域网LANs

MAC地址介绍 48bit

- 32-bit IP地址:
  - 网络层地址的接口
  - 用于layer 3网络层转发
- MAC (or LAN or physical or Ethernet) 地址:
  - 功能: 使用“本地”从一个接口获得frame到另一个物理连接的接口(在ip地址的意义类似)
  - 48 bit MAC address (对于大多数LANs) 烧录在NIC ROM, 有时也可软件设置
  - 例如: 1A-2F-BB-76-09-AD

十六进制(以16为基数)表示法
(每个“数字”代表4位)

局域网LAN上的每个适配器都有唯一的*LAN*地址



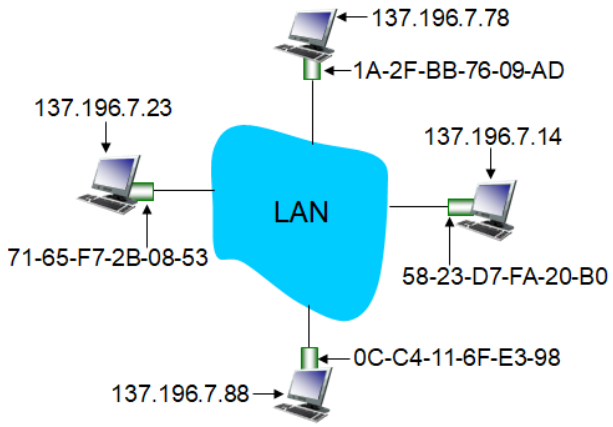
它是永远不会改变的

- 由IEEE管理的MAC地址分配
- 制造商购买部分MAC地址空间(以确保唯一性)
- 类比:
  - MAC address: 身份证号码
  - IP address: 邮寄地址
- MAC flat address ➜ 可移植性
  - 可以移动局域网卡LAN card 从一个局域网到另一个局域网
- IP分级地址不可移植
  - 地址取决于节点所连接的IP子网

地址分辨协议ARP: address resolution protocol

我知道了目的地的IP地址，如何找到接口的MAC地址?

Question: 知道接口的IP地址，如何确定接口的MAC地址?



在同一个LAN下的ARP配置

ARP表: 局域网中的每个IP节点(主机、路由器)都有一个表
- IP/MAC 地址
- 部分LAN节点的映射关系
- < IP address; MAC address; TTL>
- TTL (Time To Live):地址映射被忘记的时间(通常为20分钟)

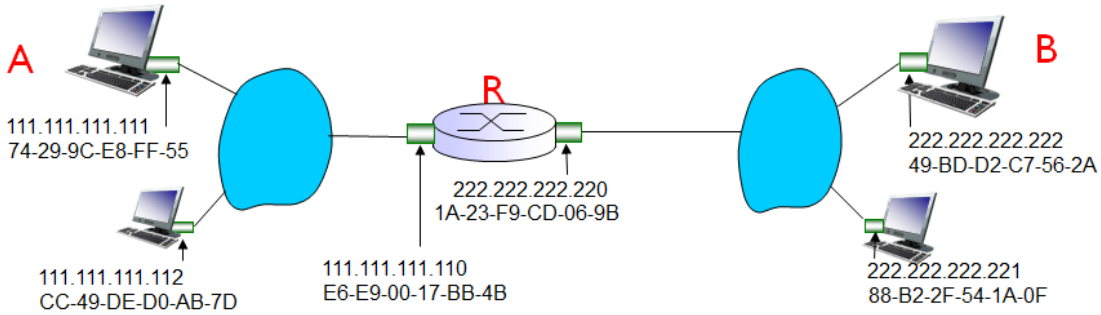
- A想把datagramm发送给B
  - B的MAC地址不在A的ARP表中
- A 广播 ARP 查询报文，包含B的IP地址
  - 目的地 MAC 地址 = FF-FF-FF-FF-FF-FF
  - 局域网内所有节点都接收到ARP查询
- B收到ARP报文，用自己的MAC地址回复A
  - 发送到A的MAC地址的frame(单播)

- A在ARP表中缓存(保存)IP-to-MAC地址对，直到信息老化(超时)。
  - 软状态:信息超时(消失)，除非刷新
- ARP 是“plug-and-play”即插即用的:
  - 节点在没有网络管理员干预的情况下创建它们的ARP表

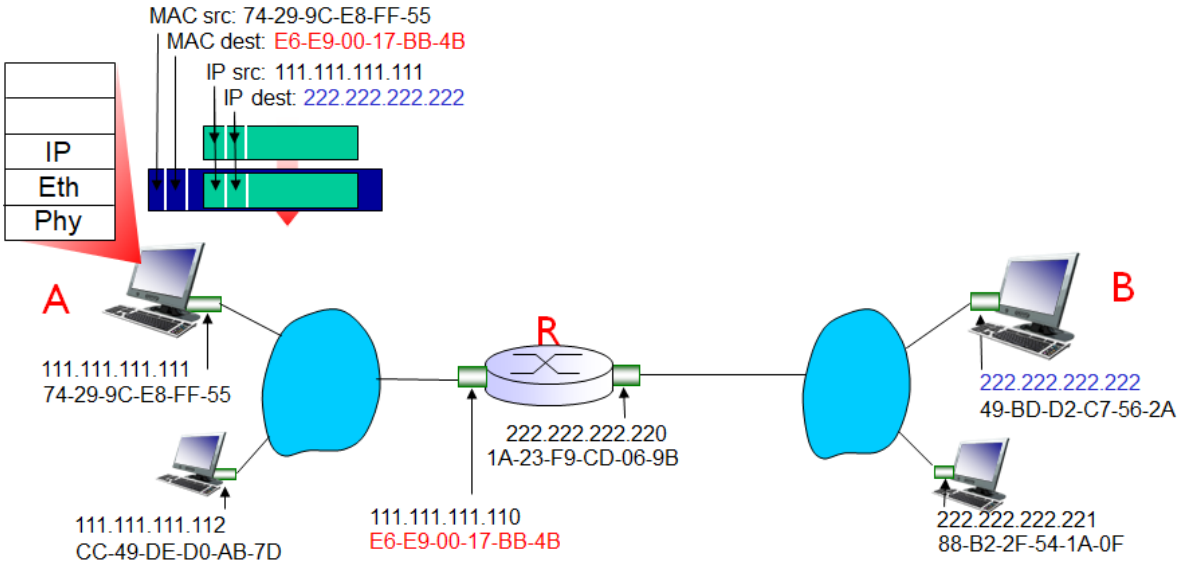
在不同LAN下的寻址

通过R将datagram从A发送到B

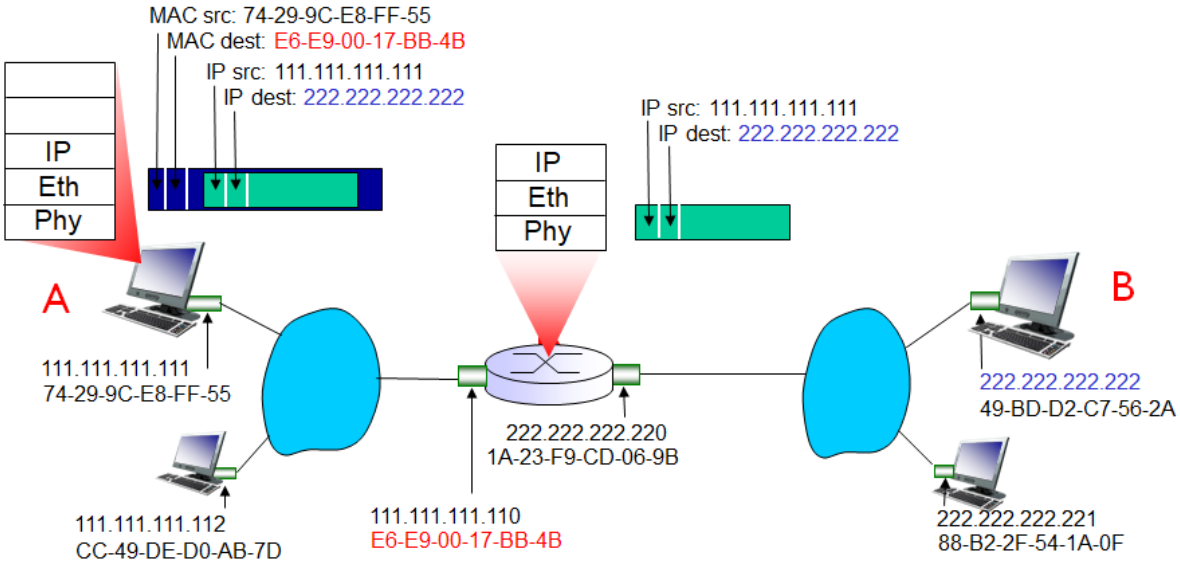
- 专注于IP(datagram)和MAC层(farme)的寻址
- 假设A知道B的IP地址
- 假设已知第一跳路由器R的IP地址（默认网关）
- 假设A知道R的MAC地址（ARP表里有了）



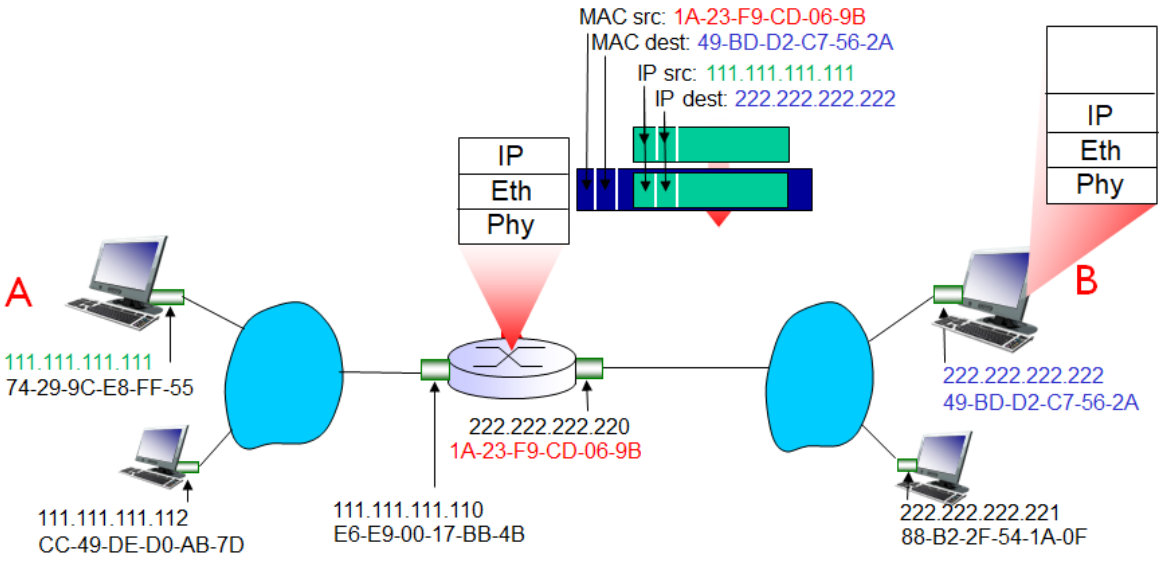
- A用IP源A和目的B创建IP datagram
- 以R的MAC地址作为目的地址创建链路层frame，frame包含A-to- B IP数据报



- 从A发送到R的frame
- frame在R接收，datagram删除，向上传递给IP



- R以源A、目的B转发datagram
- 以B的MAC地址作为目的地址创建链路层frame，frame包含A-to-B IP datagram



R知道B的MAC地址的原理是在同一个LAN下的ARP配置问题

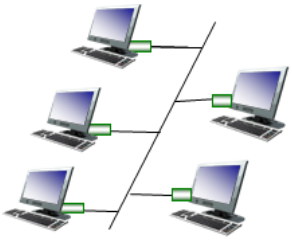
6.5 802.3 以太网（有线局域网）

以太网介绍

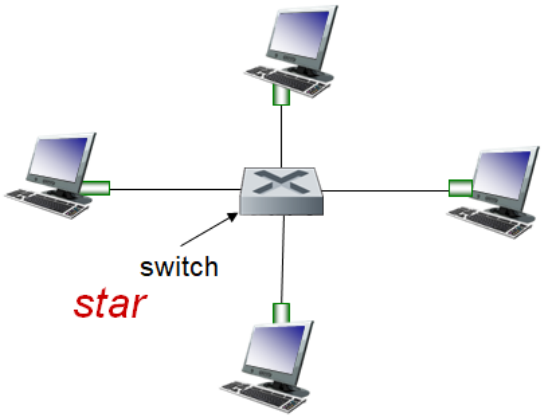
“主要”的有线局域网技术:

- 单芯片，多速度(如Broadcom BCM5761)
- 首先广泛应用的局域网技术
- 简单、廉价
- 10 Mbps – 10 Gbps

- *bus:* 一直流行到90年代中期
  - 所有节点在碰撞域内(可以相互碰撞)
- *star:* 盛行的今天
  - 中心有活动的 *switch*
  - 每个“spoke”运行一个(单独的)以太网协议(节点之间不会发生冲突)

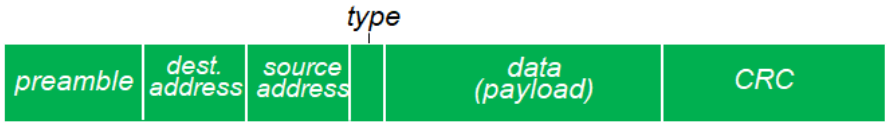


bus: coaxial cable



以太网报文字段

发送适配器把IP数据报封装成以太网形式的frame(或其他网络层协议包)



preamble:

- 带有10101010 pattern的7个字节，后跟一个带有10101011 pattern的字节
- 同步接收方、发送方时钟速率

- *addresses:* 6 byte源MAC地址、目的MAC地址
  - 如果适配器接收到与目标地址匹配的frame，或与广播地址(例如ARP包)匹配的帧，它将frame中的数据传递给网络层协议
  - 否则，适配器将丢弃该frame
- *type:* 表示更高层次的协议(主要是IP，但其他可能，如Novell IPX, AppleTalk)
- *CRC:* 接收方的循环冗余校验
  - 检测到错误:frame被丢弃

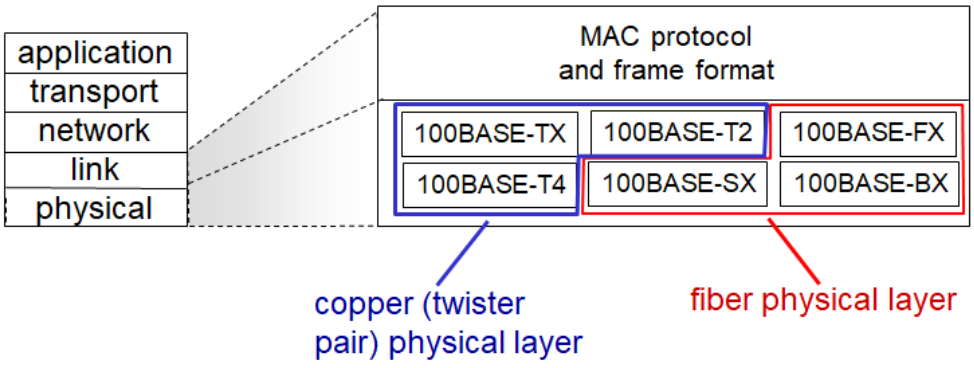
以太网:不可靠,无连接，CSMA/CD（Random Access)

- *connectionless:* 发送和接收网卡之间不握手
- *unreliable:* 接收网卡不会向发送网卡发送ACK或NACK
  - 只有当初始发送方使用更高层次的rdt(例如，TCP)时，丢弃frame中的数据才会恢复，否则丢弃的数据就会丢失
- Ethernet's MAC protocol: unslotted *CSMA/CD with binary backoff*

1. NIC从网络层接收datagram，创建frame
2. 如果NIC检测到通道空闲，启动frame的传输。如果NIC检测到通道忙，等待通道空闲，然后传输。
3. 如果NIC传输整个frame的时候没有检测到另一个传输，NIC完成了一个frame的传输
4. 如果NIC在传输过程中检测到其他传输，则中止传输并发送干扰信号
5. 终止后，NIC进入*binary (指数) backoff:*
   - 在第m次碰撞后，NIC选择一个在区间 *{0, 1, 2, …, 2^m-1}中的随即数*K. NIC希望K·512 bit时间后，回到Step 2
   - 更多的碰撞，更长的backoff

以太网标准

- *很多* 不同的以太网标准
  - 相同的MAC协议和frame格式
  - 不同的速率: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
  - 不同物理层介质:光纤、电缆



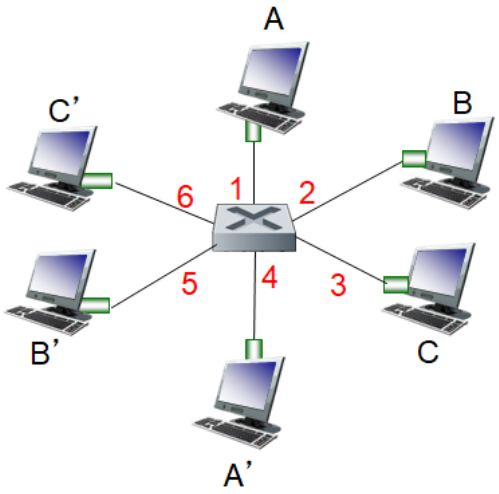
以太网交换机

介绍

- link-layer device: takes an *active* role
  - 存储、转发以太网frame
  - 检查入站frame的MAC地址，当frame在segment上被转发时，有选择地将帧转发到一个或多个出站链路，使用CSMA/CD访问segment
- *transparent*
  - 主机不知道交换机的存在
- *plug-and-play, self-learning*
  - 交换机不需要配置

多个同时传输

- 主机与交换机专用的，直接的连接
- 交换机缓存包
- 在每个传入链路上使用的以太网协议，但没有碰撞;全双工
  - 每个链接都是它自己的碰撞域
- *switching:* A-to-A’和B-to-B’可以同时传输，没有碰撞

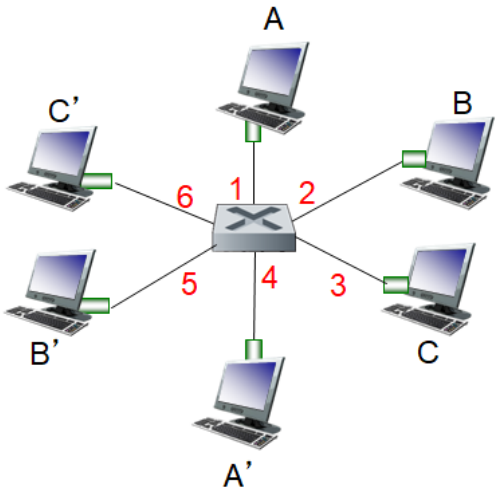


switch with six interfaces (1,2,3,4,5,6)

交换机查找表

Q: 交换机如何知道A '通过接口4可达，B '通过接口5可达?

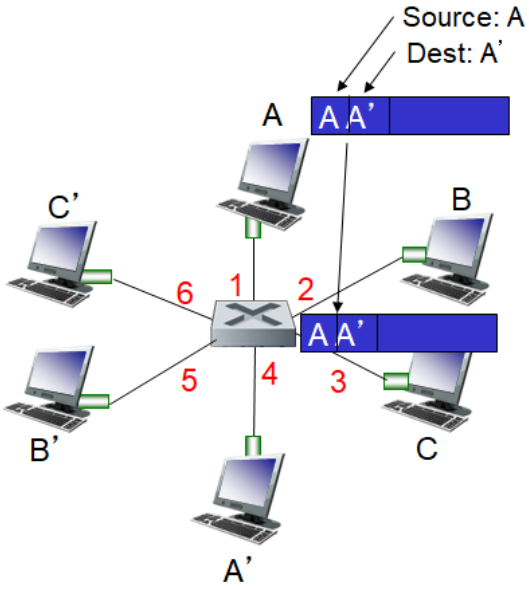
- A: 每个交换机都有一个交换机表switch table，每个条目包含：
  - (主机MAC地址，到达主机的接口，时间戳)
  - 看起来像一个路由表!



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

交换机自主学习

- 交换机自主学习哪些主机可以通过哪些接口到达
  - 当接收到frame时，交换机“学习”发送者的位置:传入局域网segment
  - 记录交换机表中的发送方/位置对



MAC addr	interface	TTL
A	*1*	*60*

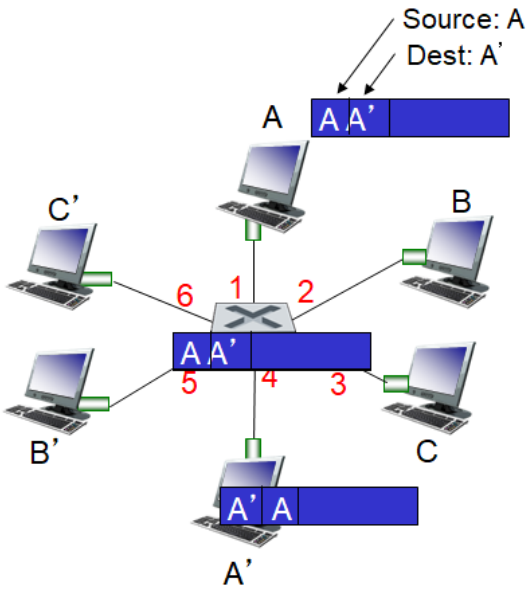
Switch table (initially empty)

当交换机接收到frame时:

1.记录传入链路，发送主机的MAC地址
2.找到包含MAC目的地址的交换机表项
3. if 找到目的地的入口
then {
if frame到达的segment上的目的地
then 丢掉该frame
else 按照入口指示的接口转发frame
}
else flood /*除了到达外，转发给所有接口*/

- frame目的地，A '，位置未知: *flood*
- 目的地的位置已知:

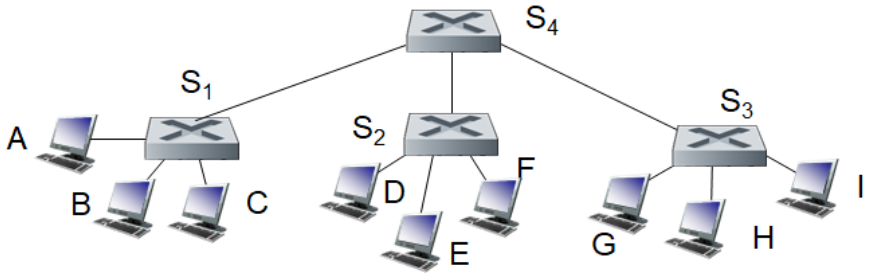
有选择地只转发给一个link



MAC addr	interface	TTL
A	*1*	*60*
A '	*4*	*60*

switch table (initially empty)

自学习交换机可以连接在一起:



Q: 从A发送到G - S1 如何知道通过S4和S3转发到G的帧?

A: 自学习! (与单交换机情况完全相同!)

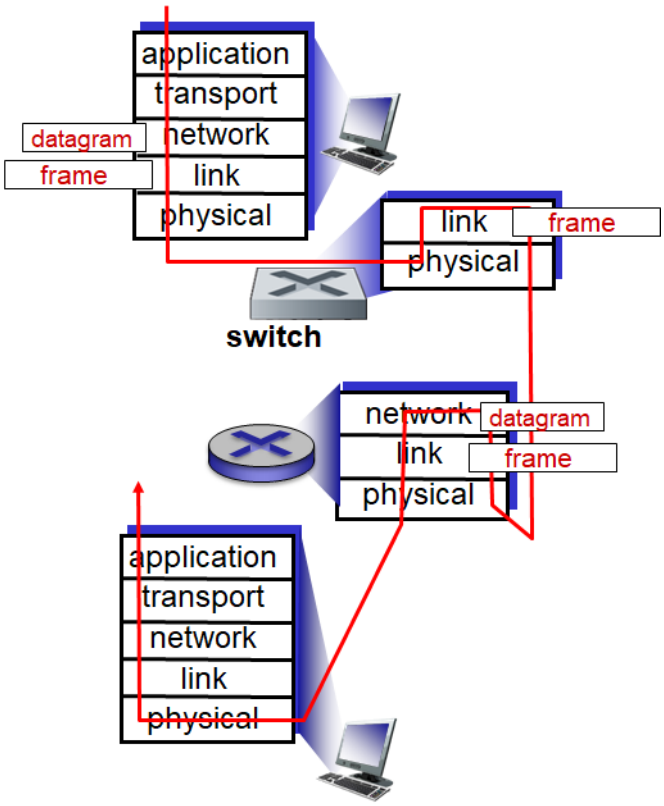
交换机和路由器

两者都是转发:

- *routers:* 网络层设备(检查网络层报头)
- *switches:*链路层设备(检查链路层报头)

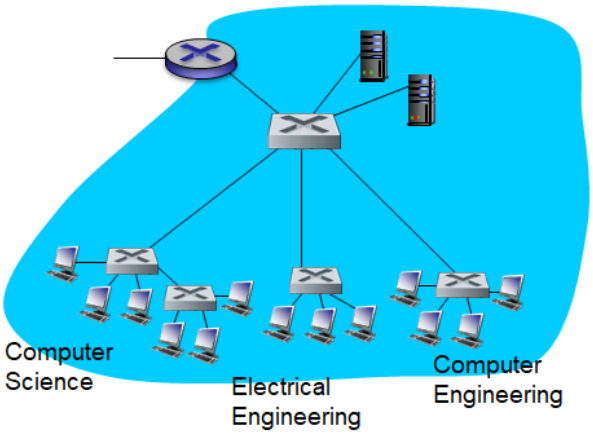
两者都有转发表:

- *routers:* 使用routing algorithm, IP地址计算转发表
- *switches:* 使用flooding、self learning、MAC地址学习转发表



6.6 802.1 虚拟局域网 VLAN

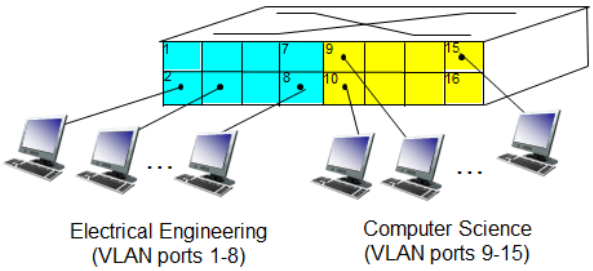
介绍



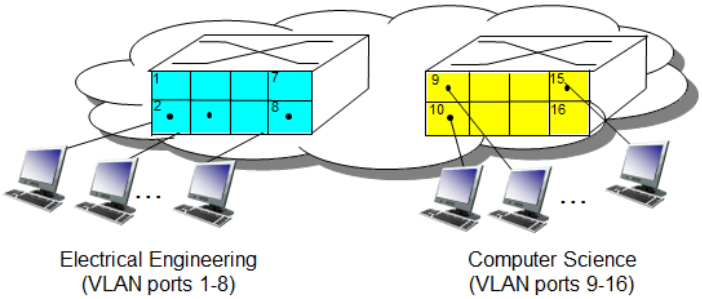
考虑:

- CS用户将办公室搬到EE, 但想要连接到CS交换机?
- 单一的广播域:
  - 所有的二层广播流量(ARP, DHCP, 未知的目的MAC地址位置)必须跨越整个局域网
  - 安全/隐私、效率问题

port-based VLAN: 交换机端口分组(通过交换机管理软件), 使单个物理交换机……



…作为多个虚拟交换机运行

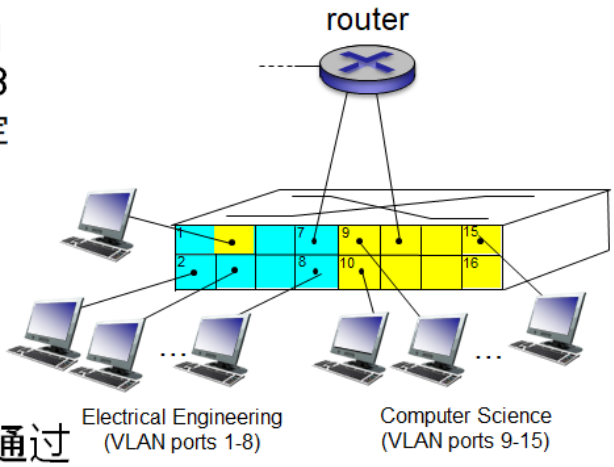


Virtual Local Area Network

支持VLAN能力的交换机可以配置为在单个物理LAN基础设施上定义多个虚拟LAN

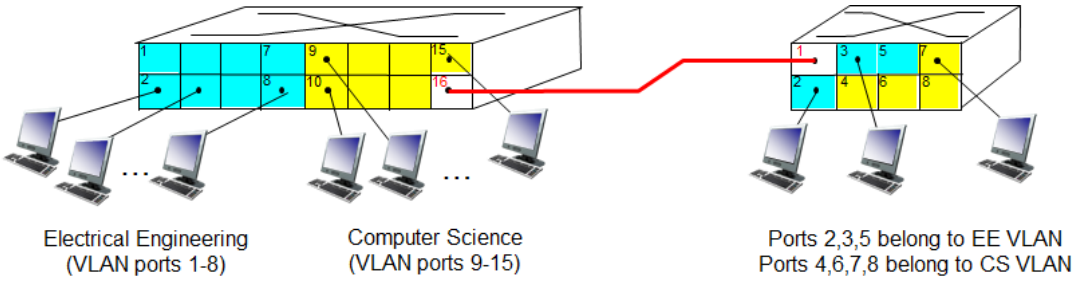
特点

- *traffic isolation:* 从端口1到端口8的帧只能到达端口1到端口8
  - 还可以根据端点的MAC地址定义VLAN，而不是交换机端口
- dynamic membership: 之间可以动态分配端口



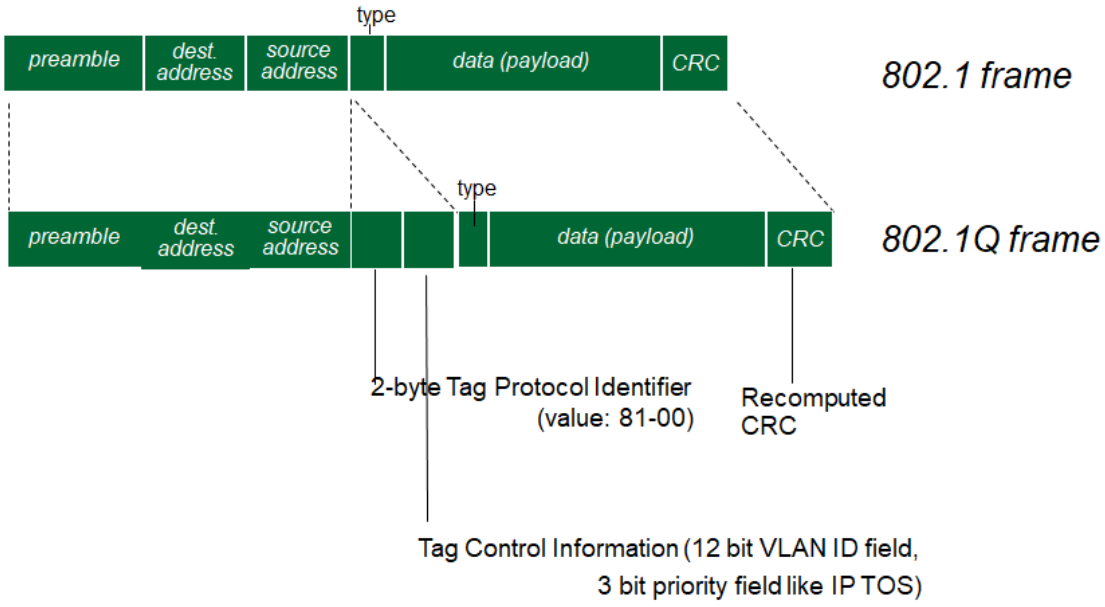
- forwarding between VLANS: 通过routing完成(就像使用单独的交换机一样)
  - 实际上，厂商销售的是交换机加路由器的组合

跨多交换机的VLAN



- *trunk port:* 在多个物理交换机上定义的VLAN之间carrier frame
  - VLAN内交换机之间转发的frame不能是普通的802.1frame(必须携带VLAN ID信息)
  - 802.1q协定中有额外的字段来指示通过trunk port转发

VLAN报文字段



6.7 data center networking

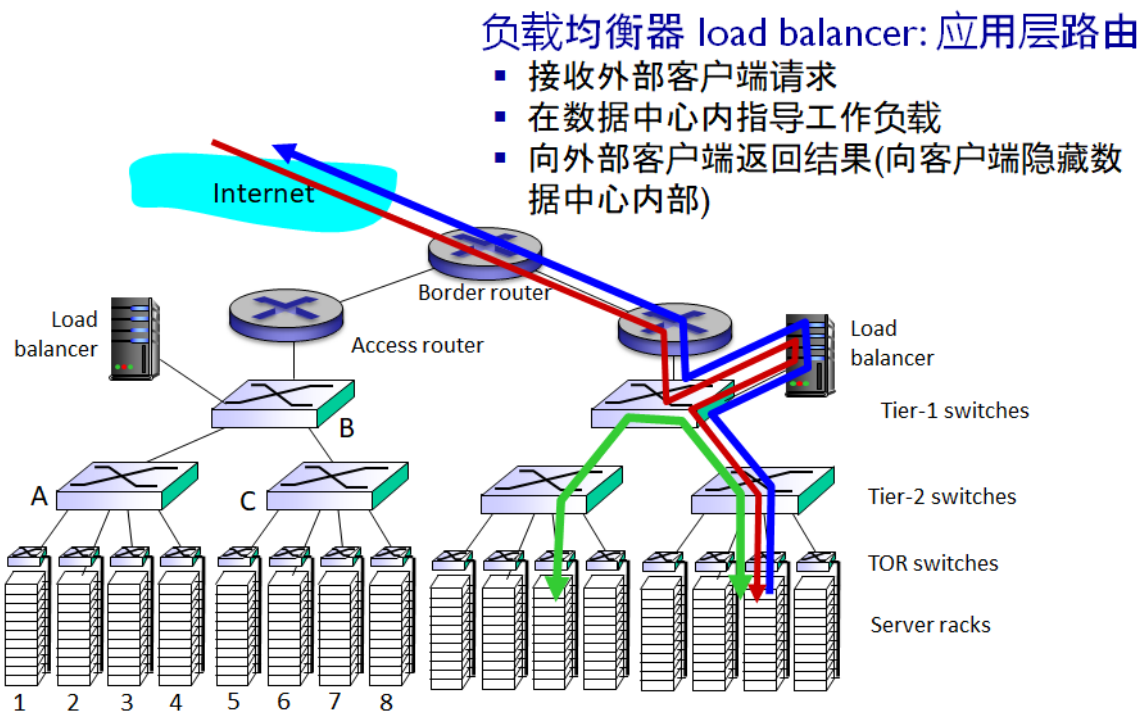
介绍

- 成千上万的主机，经常是紧密耦合的，在很近的距离:
  - e-business (e.g. Amazon)
  - content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (e.g., Google)

- 挑战:
  - 多个应用程序，每个应用程序为大量客户端服务
  - 理/平衡负载，避免处理、网络、数据瓶颈

Inside a 40-ft Microsoft container, Chicago data center

load balancer 应用层路由



- 交换机、机架间丰富互联(不仅仅是层级关系):
- 机架之间的吞吐量增加(可能有多条路由路径)
- 通过冗余增加可靠性