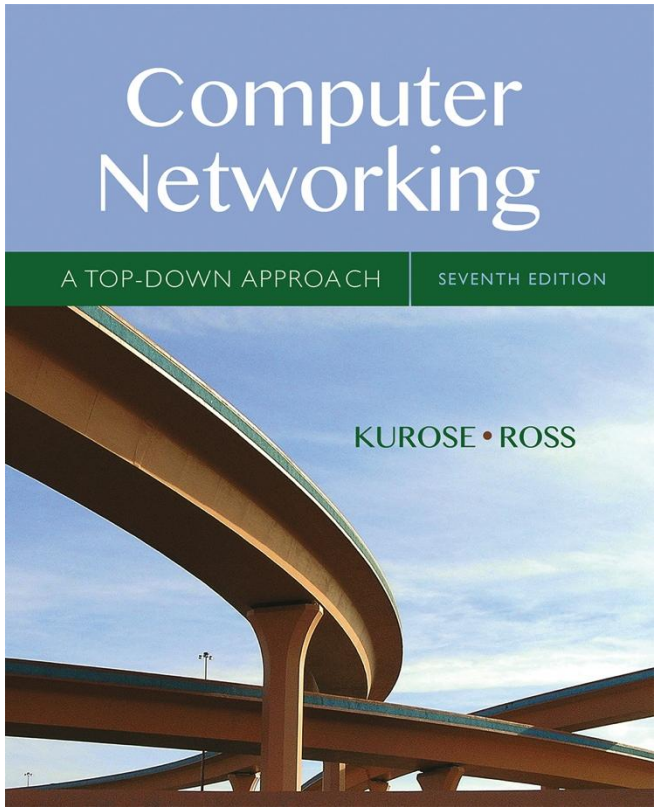
Chapter 6
The Link Layer and LANs



A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- *If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)*
- *If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.*

Thanks and enjoy! JFK/KWR

Computer Networking:
A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Chapter 6: The Data Link Layer

Our goals:

- 理解数据链路层服务原理:
  - 差错检测和纠正
  - 共享广播信道: 链路接入
  - 链路层编址
- 链路层实现
  - 以太网
  - 虚拟局域网

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization: MPLS

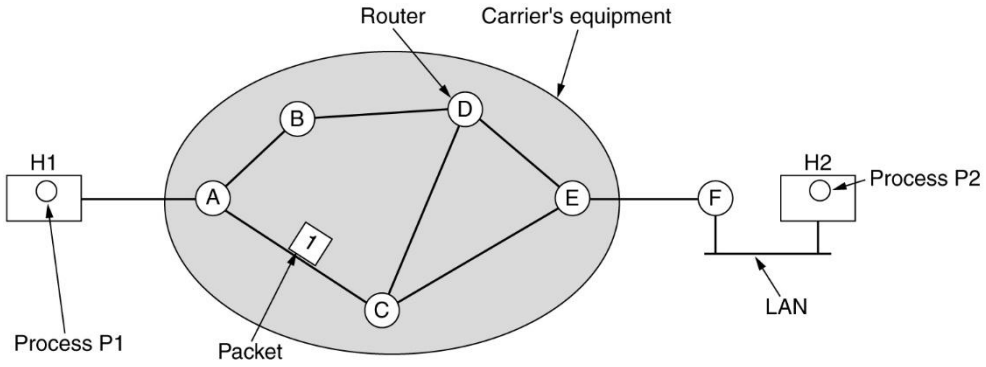
6.6 data center networking

6.7 a day in the life of a web request

网络层、链路层和物理层

网络层：

- 选路：路由器确定去往目的节点的下一跳
- 转发：在路由器内部将数据报从输入端口转移到输出端口



链路层：

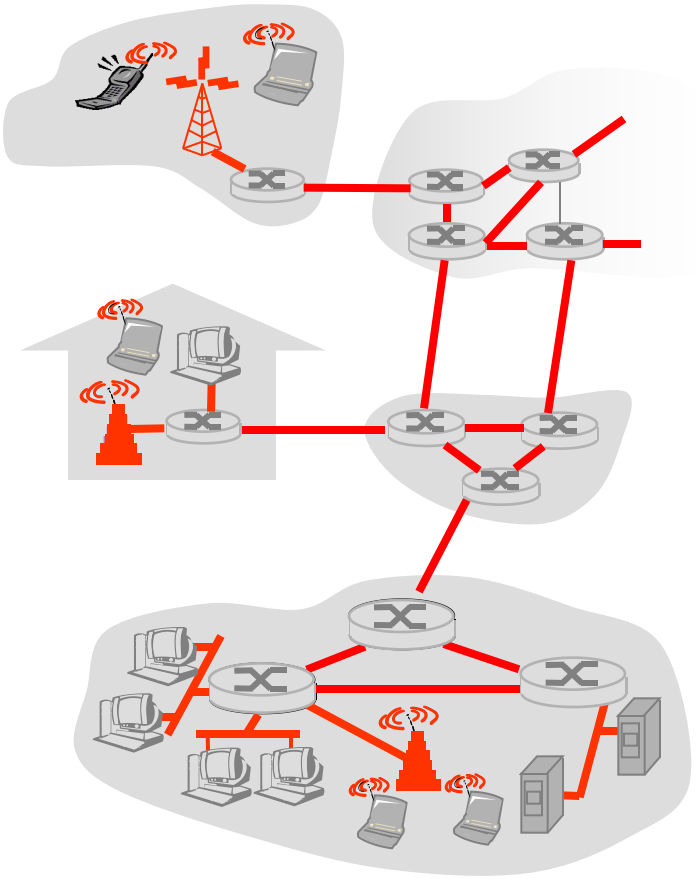
- 将数据报从一个节点传输到相邻的下一个节点

物理层：

- 多种类型的传输媒体
- 传输原始比特流（无结构）
- 容易产生传输错误

一些术语

- 节点：主机和路由器统称为节点
- 链路：连接相邻节点的通信信道
  - 有线链路
  - 无线链路
  - 局域网
- 帧：链路层分组称为帧



链路层服务

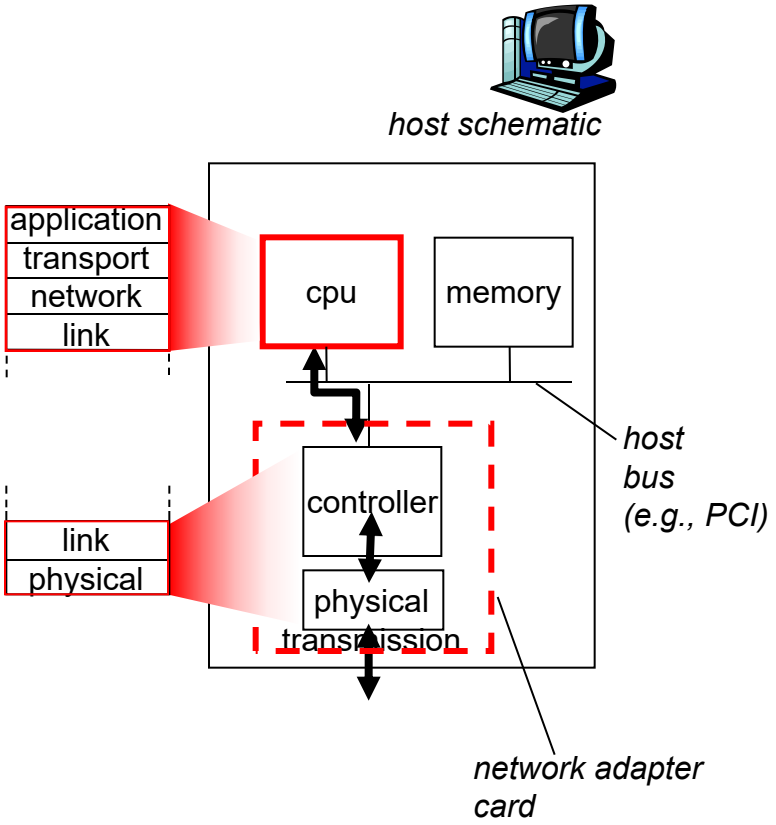
- 组帧（基本服务）
  - 发送：将数据报封装到帧中（加上帧头和帧尾）
  - 接收：从原始比特流中提取出完整的帧
- 链路接入（广播链路需要）
  - 在广播信道上协调各个节点的发送行为
- 差错检测（基本服务）
  - 检测传输错误
- 差错纠正（有些提供）
  - 检测并纠正传输错误（不使用重传）

链路层服务（续）

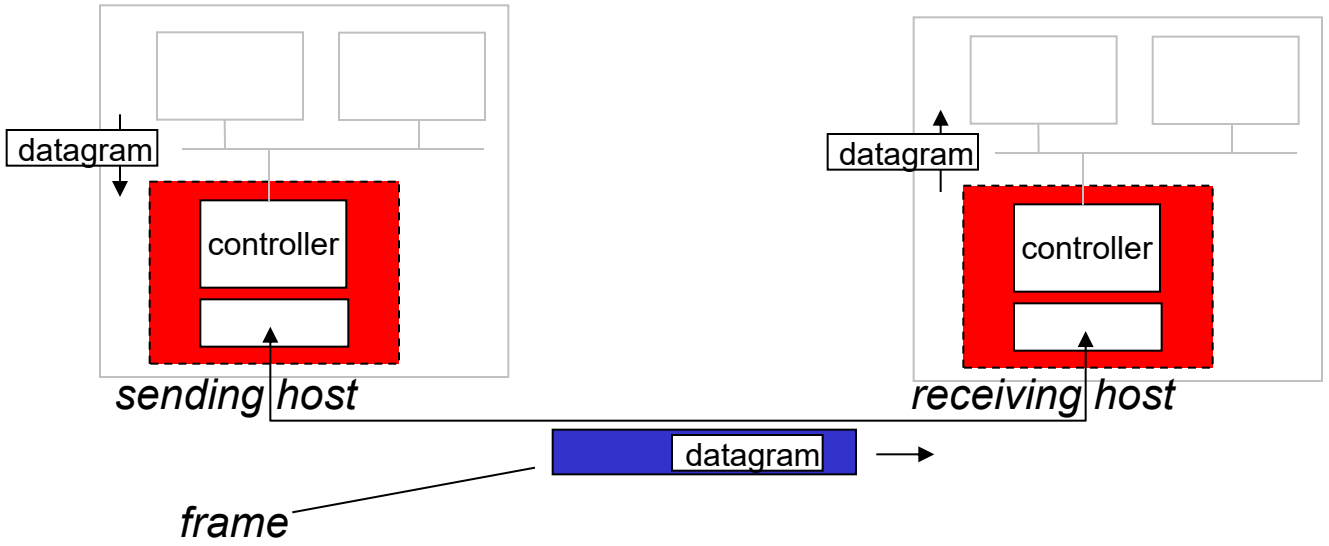
- 可靠交付（部分协议提供）
  - 通过确认、重传等机制确保接收节点正确收到每一个帧（停-等、GBN、SR）
  - 低误码率链路（如光纤、某些双绞线）上很少使用，高误码率链路（如无线链路）应当使用
- 流量控制:
  - 调节发送速度，避免接收节点缓存溢出
  - 提供可靠交付的链路层协议，不需要专门的流量控制
  - 不提供可靠交付的链路层协议，需要流量控制机制
- 半双工和全双工：
  - 半双工通信时，提供收/发转换

链路层在哪儿实现？

- 路由器：链路层在线卡（line card）中实现
- 主机：链路层主体部分在网络适配器（网卡）中实现
- 线卡/网络适配器连接物理媒体，还实现物理层的功能
- 链路层由硬件和软件实现：
  - 网卡中的控制器芯片：组帧、链路接入、检错、可靠交付、流量控制等
  - 主机上的链路层软件：与网络层接口，激活控制器硬件、响应控制器中断等



网络适配器之间的通信



- 发送侧:
  - 将数据报封装到帧中
  - 生成校验比特
  - （可选）执行可靠传输和流量控制
- 接收侧：
  - 提取帧，检测传输错误
  - （可选）执行可靠传输和流量控制
  - 解封装数据报，交给上层协议

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization: MPLS

6.6 data center networking

6.7 a day in the life of a web request

检错和纠错

- 传输出错的类型
  - 单个错：由随机的信道热噪声引起，一次只影响1位
  - 突发错：由瞬间的脉冲噪声引起，一次影响许多位，使用突发长度表示突发错影响的最大数据位数
- 差错控制编码的类型
  - 检错码：只能检测出传输错误的编码，不能确定出错位置，通常与反馈重传机制结合进行差错恢复
  - 纠错码：能够确定错误位置并自行纠正的编码

如何检测与纠正错误？

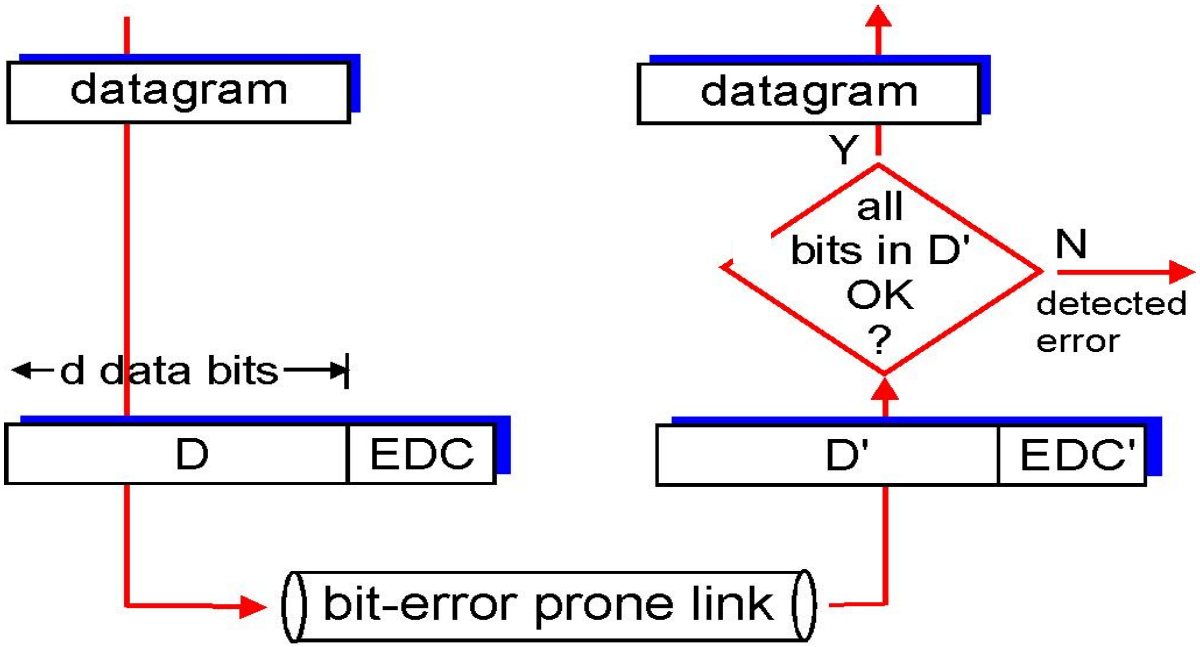
- 码字（codeword）：由 m 比特的数据加上 r 比特的冗余位（校验位）构成
- 有效编码集：由 2^m 个符合编码规则的码字组成
- 检错：若收到的码字为无效码字，判定出现传输错误
- 海明距离（Hamming Distance）：两个码字的对应位取值不同的位数（比如，100和101的海明距离为1）
- 纠错：将收到的无效码字纠正到距其最近的有效码字
- 检错码与纠错码的能力都是有限的！

编码集的检错与纠错能力

- 编码集的海明距离：编码集中任意两个有效码字的海明距离的最小值
- 检错能力：为检测出所有d比特错误，编码集的海明距离至少应为d+1
- 纠错能力：为纠正所有d比特错误，编码集的海明距离至少应为2d+1

差错检测的实施

- 发送端对要保护的数据D（包括帧头字段）生成校验位EDC，添加在帧头（尾）中
- 接收端对收到的数据D'计算校验位，与收到的校验位EDC'比较，不同则判定有错



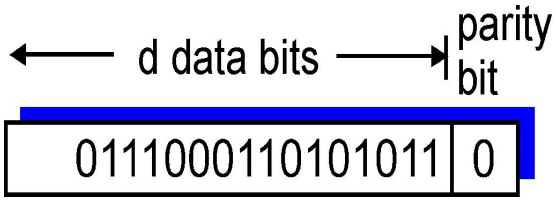
奇偶校验

单比特奇偶校验:

可检测奇数个比特错误

检错率为50%

编码集海明距离为2

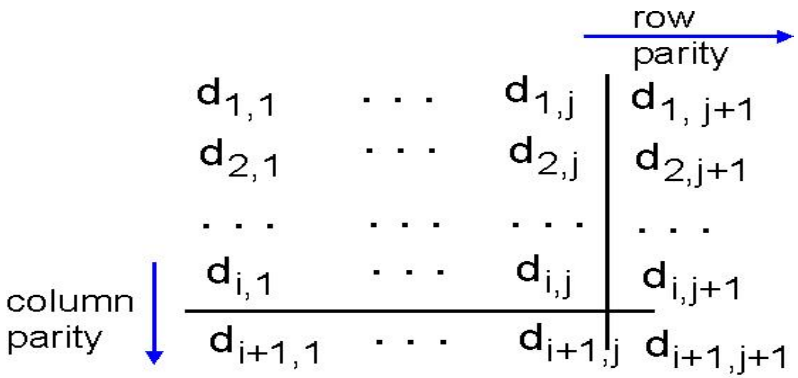


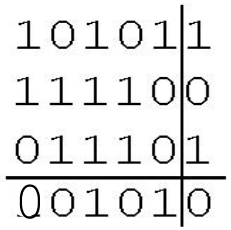
二维奇偶校验:

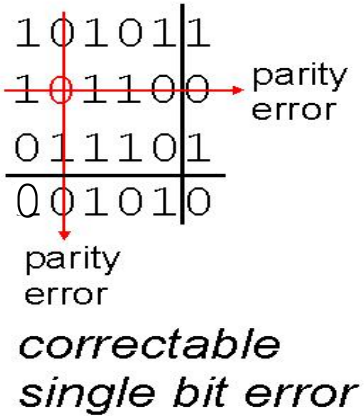
可检测2比特错和纠正单比特错

编码集海明距离为3

有利于检测突发错误







循环冗余校验（CRC）

- CRC是一种多项式编码，它将一个位串看成是某个一元多项式的系数，如1011看成是一元多项式$X^3 + X + 1$的系数
- 信息多项式M(x)：由m个信息比特为系数构成的多项式
- 冗余多项式R(x)：由r个冗余比特为系数构成的多项式
- 码多项式T(x)：在m个信息比特后加上r个冗余比特构成的码字所对应的多项式，表达式为$T(x) = x^r \cdot M(x) + R(x)$
- 生成多项式G(x)：双方确定用来计算R(x)的一个多项式
- 编码方法：$R(x) = x^r \cdot M(x) \div G(x)$ 的余式（减法运算定义为异或操作）
- 检验方法：若$T(x) \div G(x)$的余式为0，判定传输正确
- CRC码检错能力极强，可用硬件实现，是链路层上应用最广泛的检错码

CRC举例

例1：取$G(X) = X^3 + 1$，对信息比特101110计算CRC码。

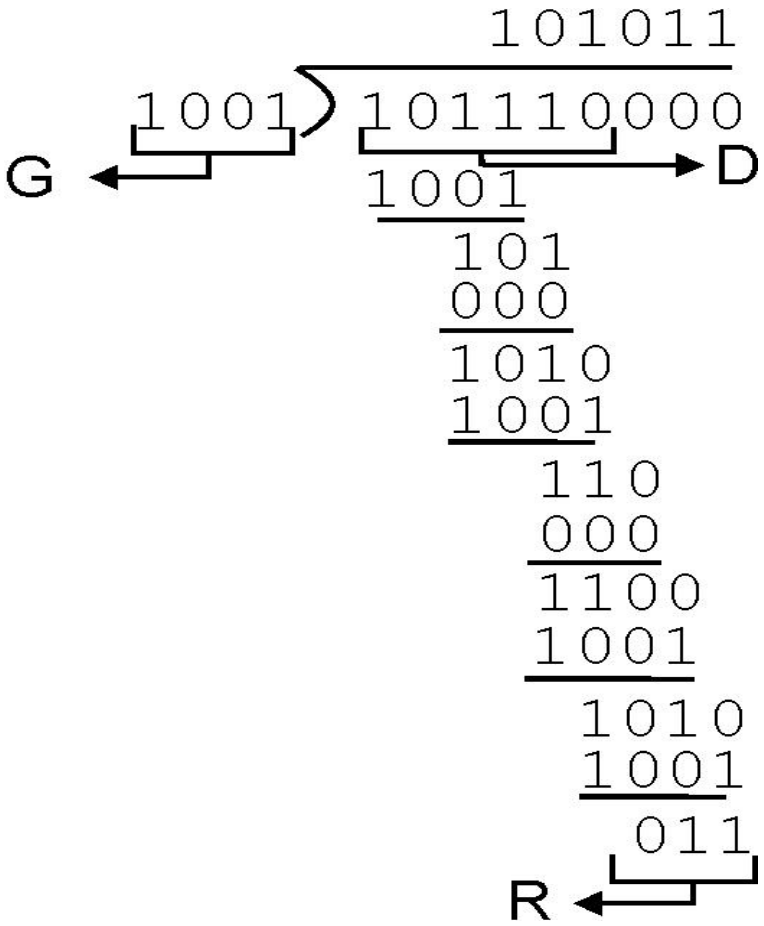
解答：

- 101110000 ÷1001的余式为R=011 (CRC code)
- 码字：101110011

例2：取$G(X) = X^3 + 1$，接收端收到比特串1001001，问是否有错？

解答：

- 1001001÷1001的余式为001（不为0），有传输错误



要求理解的知识点

- 检错和纠错的一般性原理
- 二维奇偶校验、循环冗余码
- 为什么链路层使用CRC，而其上各层使用checksum？（思考）

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

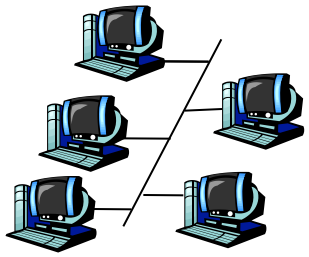
5.5 link virtualization

5.6 data center networking

5.7 a day in the life of a web request

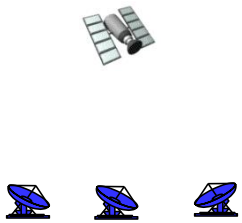
链路的两种类型

- 点到点链路：
  - 仅连接了一个发送方和一个接收方的链路
  - 一条全双工链路可以看成是由两条单工链路组成
- 广播链路：
  - 连接了许多节点的单一共享链路，任何一个节点发送的数据可被链路上的其它节点接收到



共享的电缆
（如早期以太网）

共享的无线射频
（如802.11 WiFi）



共享的无线射频
（如卫星）

humans at a
cocktail party
(shared air, acoustical)

多址接入（Multiple Access）

- 冲突（collision）
  - 在广播链路上，若两个或多个节点同时发送，发送的信号会发生干扰，导致接收失败
- 多址接入协议
  - 规定节点共享信道（谁可以发送）的方法
  - 多址接入协议也称媒体接入控制（Medium Access Control，MAC）协议

理想的多址接入协议

在速率为R bps的广播信道上

1. 当只有一个节点发送时，它应能以速率R发送（信道利用率高）
2. 当有M个节点发送时，每个节点应能以 R/M的平均速率发送（公平性好、信道利用率高）
3. 协议是无中心的（分散式）：
   - 不需要一个特殊的节点来协调发送（健壮性好）
   - 不需要时钟或时隙同步（不需要额外的机制）
4. 简单（实现和运行开销小）

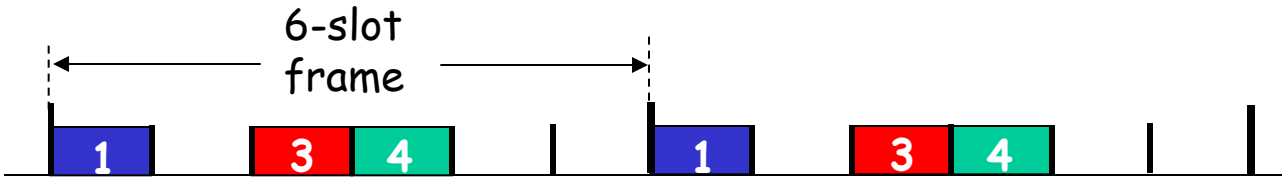
MAC协议的分类

- 信道划分
  - 将信道划分为若干子信道，每个节点固定分配一个子信道，不会发生冲突
  - 关注公平性，轻负载时信道利用率不高
- 随机接入（竞争）
  - 不划分信道，每个节点自行决定何时发送，出现冲突后设法解决
  - 轻负载时信道利用率高，重负载时冲突严重
- 轮流使用信道
  - 不划分信道，有数据的节点轮流发送，不会出现冲突
  - 信道利用率高，公平性好，但需引入额外机制

（1）信道划分的MAC协议

TDMA: 时分多址

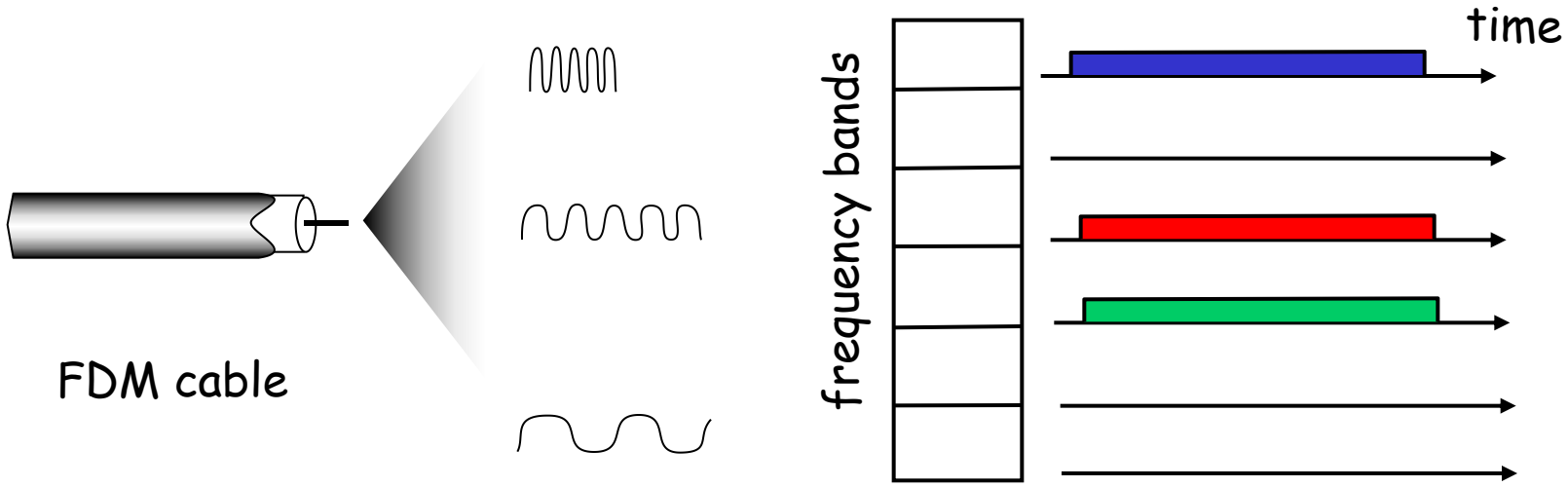
- 将信道的使用时间划分成帧，每个节点在帧中被分配一个固定长度的时间片，每个时间片可以发送一个分组
- 节点只能在分配给自己的时间片内发送
- 若节点不发送，其时间片轮空



信道划分的MAC协议

FDMA: 频分多址

- 将信道频谱划分为若干子频带
- 每个节点被分配一个固定的子频带
- 若节点不发送，其子频带空闲



信道划分的MAC协议

CDMA: 码分多址

- 将每个比特时间进一步划分为m个微时隙（称chip）
- 每个节点被分配一个惟一的m比特码序列（称chip code）
- 发送方编码：发送“1”=发送chip code；发送“0”=发送chip code的反码
- 信号叠加：多个节点发送的信号在信道中线性相加
- 接收方解码：用发送方的chip code与信道中收到的混合信号计算内积，恢复出原数据
- 前提条件：任意两个chip code必须是相互正交的

- **CDMA允许所有节点同时使用整个信道！**

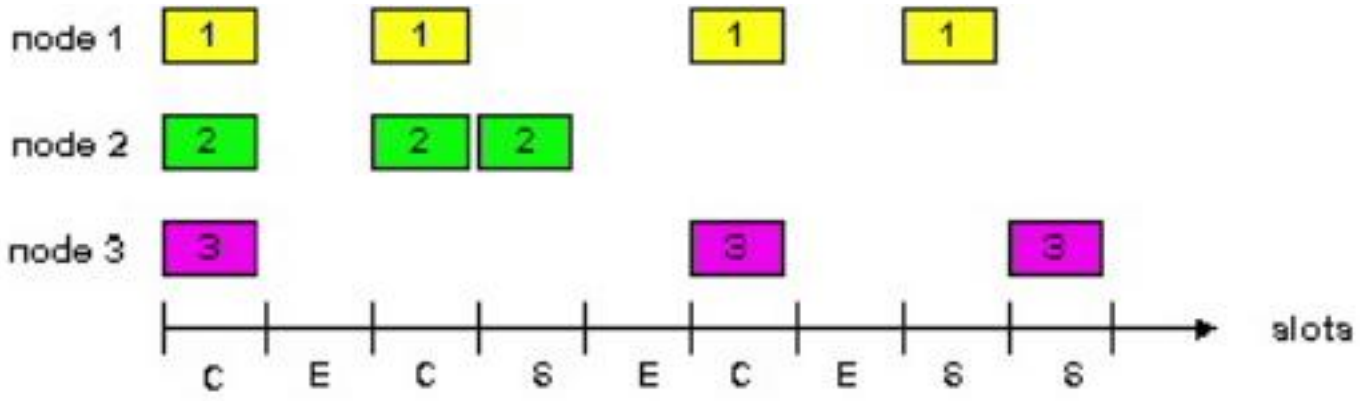
（2）随机接入的MAC协议

- 随机接入的基本思想：
  - 当节点有数据要发送时，以信道速率R发送，发送前不需要协调
  - 随机接入MAC协议规定如何检测冲突，以及如何从冲突中恢复
- 随机接入MAC协议的例子：
  - 发送前不监听信道：ALOHA家族
  - 发送前监听信道：CSMA家族

时分（Slotted） ALOHA

假设:

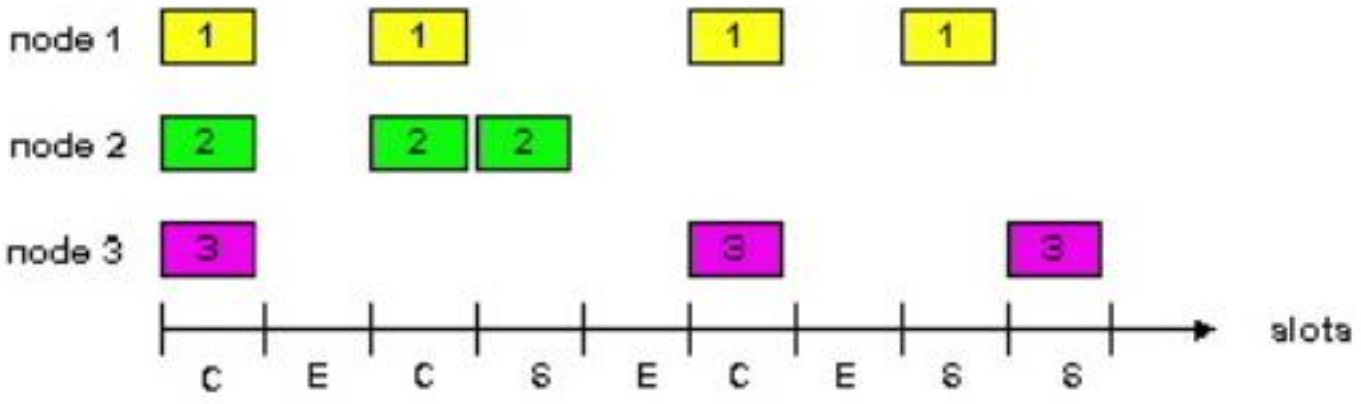
- 所有帧长度相同
- 时间被划分为等长的时隙，每个时隙传一帧
- 节点只能在时隙开始时发送
- 节点是时钟同步的（知道时隙何时开始）
- 所有节点可在时隙结束前检测到是否有冲突发生



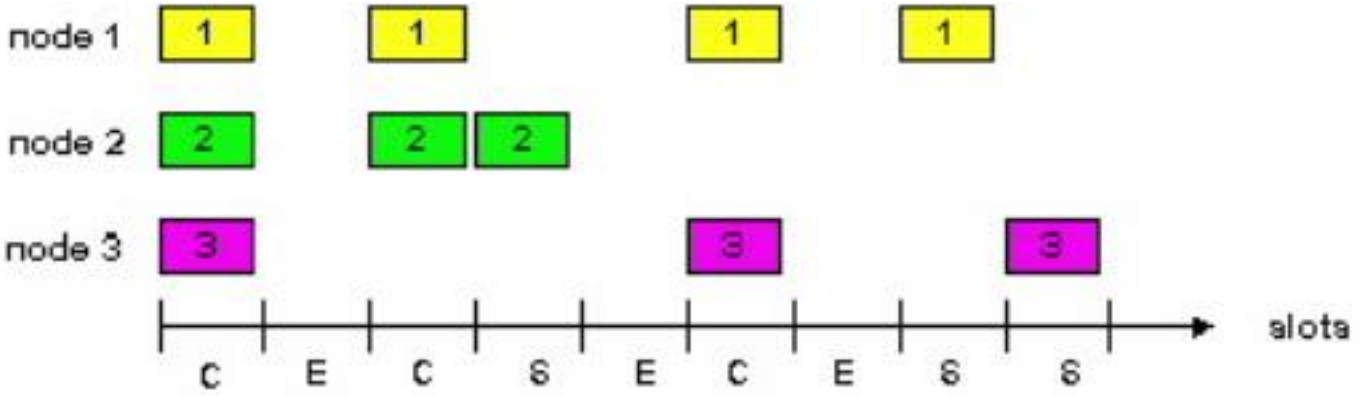
时分ALOHA

操作:

- 节点从上层收到数据后，在下一个时隙发送
- 若时隙结束前未检测到冲突，节点可在下一个时隙发送新的帧
- 若检测到冲突，节点在随后的每一个时隙中以概率P重传，直至发送成功



时分ALOHA



优点

- 单个活跃节点可以信道速率连续发送
- 分散式：节点自行决定什么时候发送
- 简单

缺点

- 发生冲突的时隙被浪费了
- 由于概率重传，有些时隙被闲置
- 需要时钟同步

时分Aloha的效率

效率：当网络中存在大量活跃节点时，长期运行过程中成功时隙所占的比例

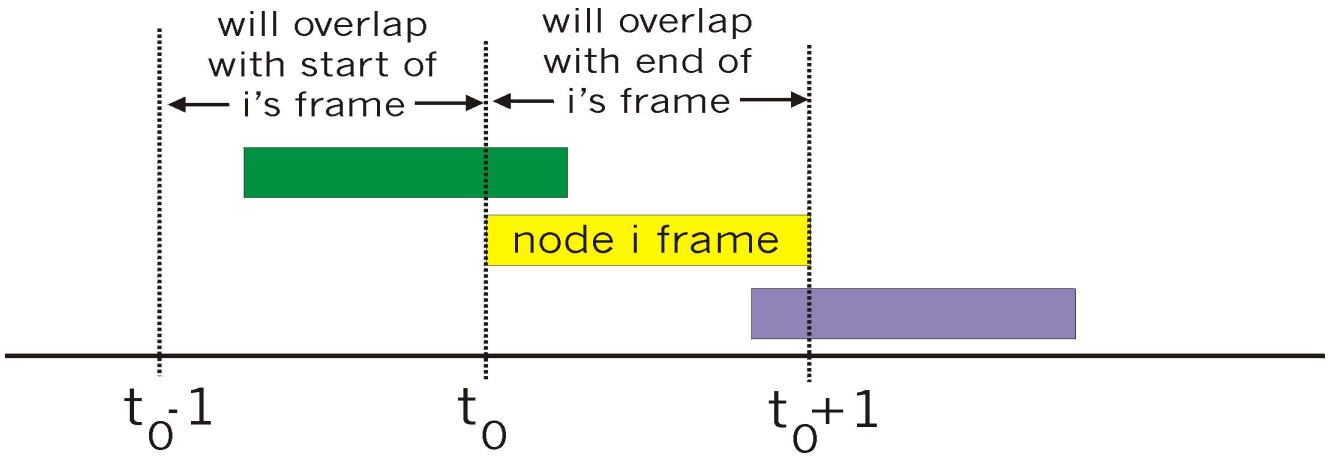
- 假设: 有N个活跃节点，每个节点在每个时隙开始时以概率P发送
- 给定节点在一个时隙中发送成功的概率 = $p(1-p)^{N-1}$
- 给定时隙中有节点发送成功的概率 = $Np(1-p)^{N-1}$

- 最大效率:
  - 找到令$Np(1-p)^{N-1}$最大的概率p^*
  - 代入$Np^*(1-p^*)^{N-1}$，并令N趋向于无穷，得到:
  - 最大效率 = $1/e = 0.37$

最佳情况: 信道用于有效传输的时间仅为37%!

纯ALOHA

- 基本思想：
  - 不需时钟同步，任何节点有数据发送就可以立即发送
  - 节点通过监听信道判断本次传输是否成功
  - 若不成功，立即以概率P重传，以概率（1-P）等待一个帧时后再决定。（帧时：发送一帧的时间，假设帧长度相同）
- 发生冲突的情形：
  - 在时刻t_0发送的帧与在 (t_0-1,t_0+1) 时段内发送的其它帧冲突



纯Aloha的效率

P(给定节点发送成功) = P(节点发送) .

P(无其它节点在$(t_0-1,t_0]$内发送) .

P(无其它节点在$[t_0, t_{0+1})$内发送)

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

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

求出令节点发送成功概率$Np \cdot (1-p)^{2(N-1)}$最大的p*，并令N -> infty:

最大效率 = 1/(2e) = 0.18

载波侦听多址接入（CSMA）

- 发送前监听信道（carrier sensing）：
  - 信道空闲：发送整个帧
  - 信道忙：推迟发送
- 冲突仍可能发生：
  - 由于存在传输延迟，节点可能没有监听到其它节点正在发送
  - 即使忽略传输延迟，当两个（或多个）节点同时发现信道由忙变为空闲、并都决定立即发送时，仍会发生冲突

CSMA/CD (Collision Detection)

- 若在发送的过程中检测到冲突，怎么办？
  - 继续发送余下的部分（浪费带宽）
  - 停止发送余下的部分
- CSMA/CD的基本思想：
  - 在发送的过程中检测冲突（发生冲突时信号较强）
  - 检测到冲突后，立即停止发送剩余的部分
  - 立即启动冲突解决的过程

以太网MAC协议

- 早期以太网采用CSMA/CD协议:
  1. 网卡从网络层接收数据报，构造以太帧
  2. 若网卡监听到信道空闲，立即发送帧；若信道忙，坚持监听直至发现信道空闲，然后发送帧
  3. 若网卡发送完整个帧而没有检测到冲突，认为发送成功！
  4. 若网卡在传输过程中检测到冲突，立即停止发送帧，并发送一个阻塞信号（加强冲突）

以太网MA协议（续）

5. 网卡进入指数回退阶段，选择一个等待时间：
   - 第一次冲突后：从{0,1}中选择K，延迟K· 512比特时间
   - 第二次冲突后：从{0,1,2,3}中选择K，……
   - 第三次冲突后：从{0,1,2,3,4,5,6,7}中选择K，……
   - ……
   - 第10次冲突后，从{0,1,2,3,4,…,1023}中选择K，……
6. 返回Step 2

   注：512比特是一个最小以太帧的长度

- 指数回退的目的是根据网络负载调整重传时间：
  - 负载越重（冲突次数越多），等待时间的选择范围越大，再次发生冲突的可能性越小

CSMA/CD的效率

- T_{prop} = 以太网中任意两个节点之间传播延迟的最大值
- t_{trans} = 最长帧的传输时间

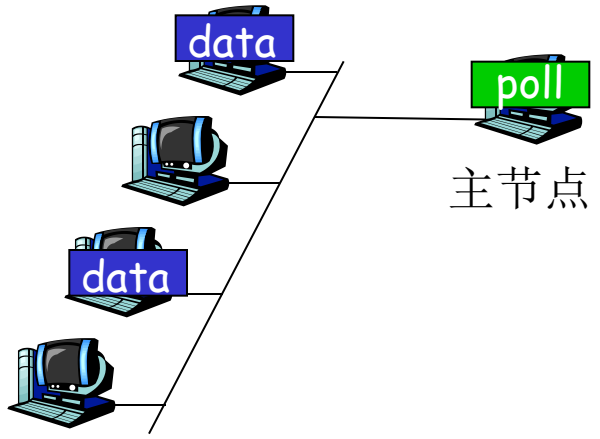
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- 在以下情况下，以太网的效率趋近于1：
  - t_{prop} 趋近于 0，或
  - t_{trans} 趋向于无穷
- 结论：应控制以太网的规模

（3）轮流MAC协议

轮询

- 主节点轮流“邀请”从节点发送，邀请到的从节点允许发送
- 缺点:
  - 引入轮询延迟
  - 单点失效（主节点）

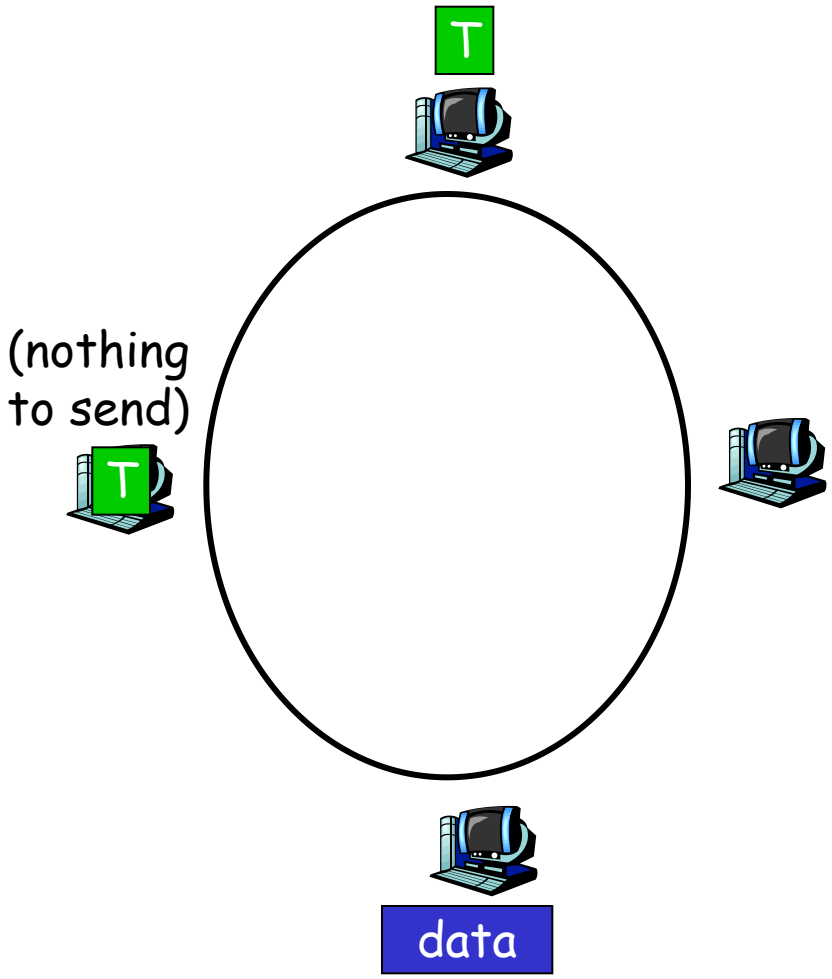


轮流MAC协议

令牌传递

- 网络中有一个令牌，按预定顺序在节点间传递
- 获得令牌的节点可以发送
- 发送完数据后释放令牌

- 缺点：
  - *令牌传递延迟*
  - *单点失效（令牌）*



MAC协议小结

- 按照时间、频率、编码划分信道：
  - 时分多址、频分多址、码分多址
- 随机接入：
  - 纯ALOHA, S-ALOHA（ALOHA网络）
  - CSMA/CD（早期以太网）
  - CSMA/CA（802.11）（第7章）
- 轮流访问：
  - 中心节点轮询（蓝牙）
  - 令牌传递（FDDI、IBM令牌环、令牌总线）

MAC协议比较

信道划分MAC协议：

- 重负载下高效：没有冲突，节点公平使用信道
- 轻负载下低效：即使只有一个活跃节点也只能使用1/N的带宽

随机接入MAC协议：

- 轻负载时高效：单个活跃节点可以使用整个信道
- 重负载时低效：频繁发生冲突，信道使用效率低

轮流协议（试图权衡以上两者）：

- 按需使用信道（避免轻负载下固定分配信道的低效）
- 消除竞争（避免重负载下的发送冲突）

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization: MPLS

6.6 data center networking

6.7 a day in the life of a web request

局域网、城域网和广域网

- 局域网LAN（Local Area Network）
  - 将小范围内的计算机及外设连接起来的网络，范围在几公里以内，通常为个人或机构所有
- 城域网MAN（Metropolitan Area Network ）
  - 通常覆盖一个城市的范围（几十公里），要能支持数据、音频和视频在内的综合业务，服务质量好，支持用户数量多
- 广域网WAN（Wide Area Network）
  - 通常覆盖一个国家或一个洲（一百公里以上），规模和容量可任意扩大

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization

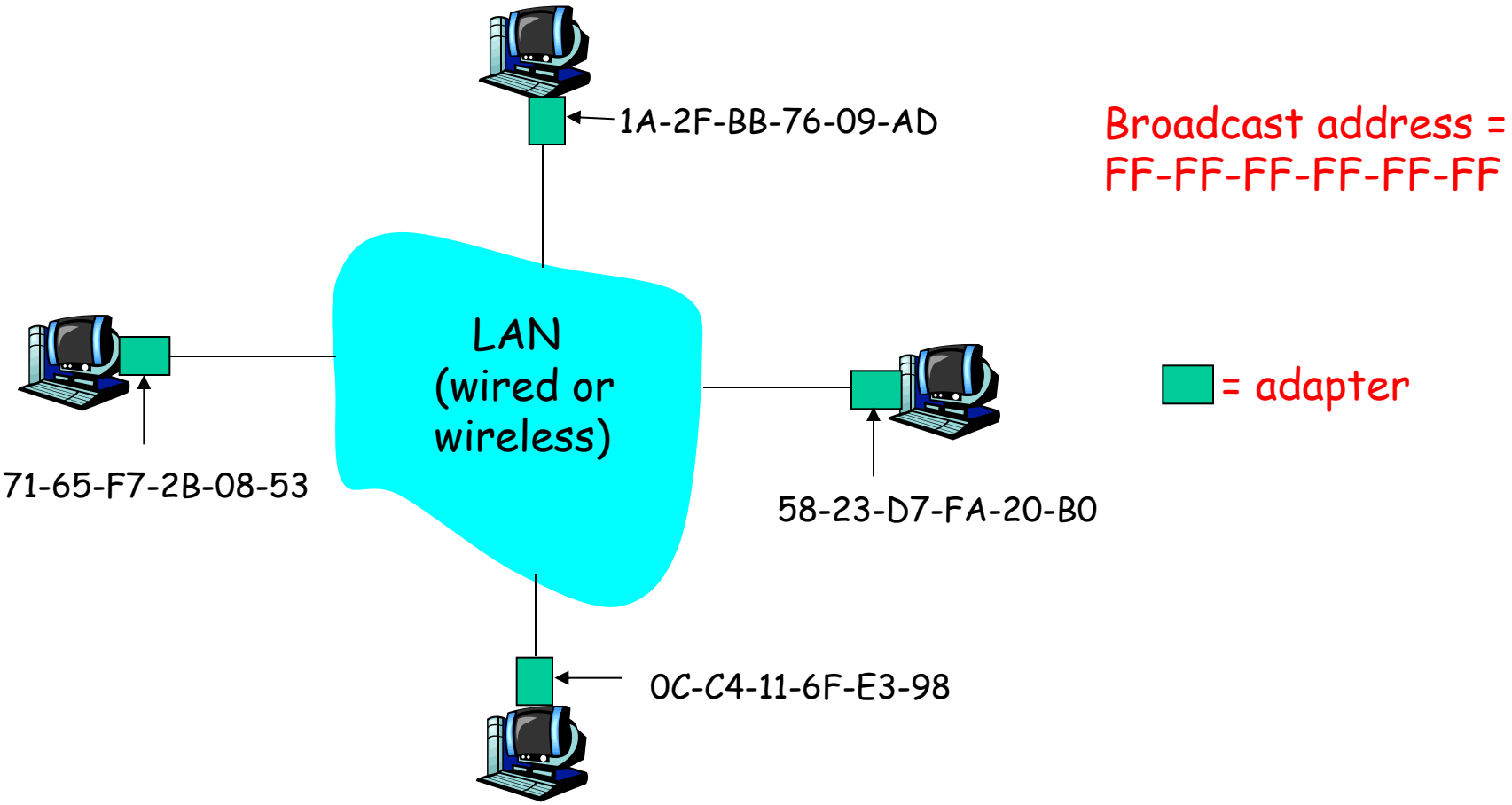
5.6 data center networking

5.7 a day in the life of a web request

链路层编址

- 早期的局域网多采用广播信道，节点如何判断收到的帧是给自己的？
- 每一块网络适配器（网卡）固定分配一个地址，称为物理地址、硬件地址、链路层地址、MAC地址等
- MAC地址长6个字节，一般用由“：”或“-”分隔的6个十六进制数表示
- MAC地址由IEEE负责分配，每块适配器的地址是全球唯一的：
  - 网卡生产商向IEEE购买一块MAC地址空间（前3字节）
  - 生产商确保生产的每一块网卡有不同的MAC地址
  - MAC地址固化在网卡的ROM中
  - 现在用软件改变网卡的MAC地址也是可能的

每个适配器有一个MAC地址



MAC地址类型

- 帧的目的MAC地址有三种类型：
  - 单播地址：适配器的MAC地址，地址最高比特为0
  - 多播地址：标识一个多播组的逻辑地址，地址最高比特为1
  - 广播地址：ff:ff:ff:ff:ff:ff
- 网络适配器仅将发送给本节点的帧交给主机：
  - 目的地址为适配器MAC地址的单播帧
  - 所有广播帧
  - 指定接收的多播帧
- 若将适配器设置成混收模式，适配器将收到的所有帧交给主机

MAC地址和IP地址

- 世界上先有MAC地址，后有IP地址
- 在TCP/IP（互联网）出现之前，只使用MAC地址在单个的物理网络中寻址
- 为什么有了MAC地址，还需要IP地址？
  - MAC地址是扁平结构的，无法在因特网范围内快速确定接口的位置
  - IP地址是有结构的，可以在因特网范围内快速确定网络接口的位置
- IP地址与MAC地址没有固定的关联关系：
  - MAC地址与网卡绑定，与节点在哪个子网无关
  - IP地址与所在子网有关，与网卡没有关系

如何将数据报发送到下一跳？

- 当发送节点A、接收节点B位于同一个物理网络上时，数据报可从A直接交付给B：
  - A的网络层将数据报、B的MAC地址交给数据链路层
  - A的数据链路层将数据报封装在一个链路层帧中，帧的目的地址=B的MAC地址
  - B的适配器收到帧，根据目的MAC地址判断是发给本机的，取出数据报交给网络层
- 问题：
  - A的网络层如何得知B的MAC地址？

地址解析（Address Resolution）

- 问题：已知IP地址，如何得到对应的MAC地址？
- 静态映射IP地址-MAC地址的缺点：
  - 主机每次使用的IP地址可能不同（DHCP）
  - 主机可能更换网卡
- 地址解析协议（ARP）用于动态获得IP地址-MAC地址映射，其基本思想是：
  - 若节点A希望获得节点B的MAC地址，节点A广播B的IP地址（地址解析请求）
  - 节点B用自己的MAC地址进行响应

ARP报文格式

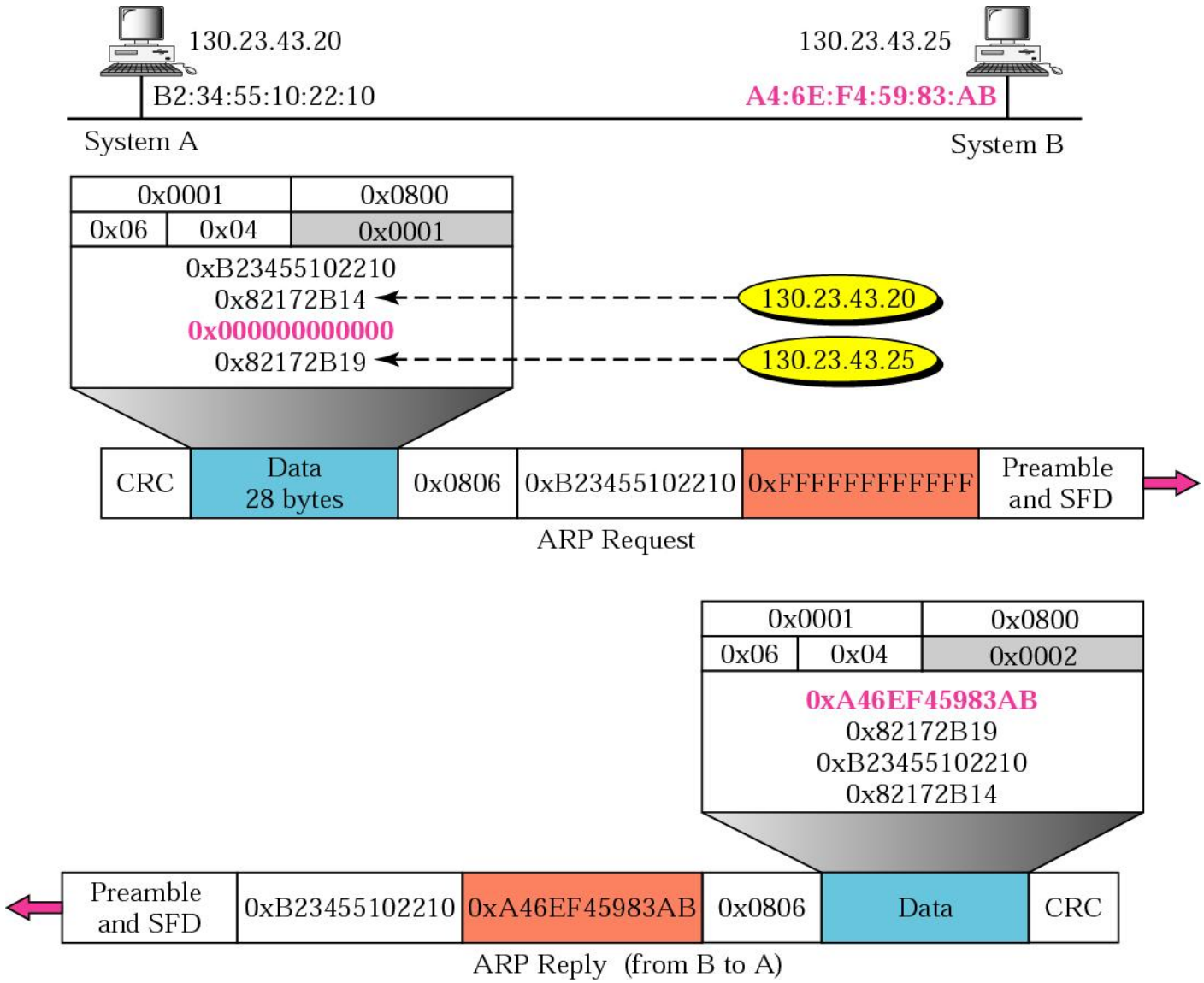
- 硬件类型：硬件接口类型。对于以太网，该值为“1”。
- 协议类型：高层协议地址类型。对于IP地址，该值为0800_{16}。
- 操作：ARP请求为1，ARP响应为2
- 在以太网上，ARP报文封装在以太帧中传输

地址解析的过程

A想知道B的MAC地址：

- A构造一个ARP请求，在发送方字段填入自己的MAC地址和IP地址，在目标字段填入B的IP地址
- A将ARP请求封装在广播帧中发送
- 每个收到ARP请求的节点用目标IP地址与自己的IP地址比较，地址相符的节点进行响应（B响应）
- B构造一个ARP响应，交换发送方与目标字段内容，在发送方硬件地址字段填入自己的MAC地址，修改操作字段为2
- B将ARP响应封装在单播帧（目的地址为A的MAC地址）中发送

IP地址为130.23.43.20、物理地址为oxB23455102210的主机，要获得IP地址为130.23.43.25的主机的MAC地址



改进ARP的措施：ARP缓存

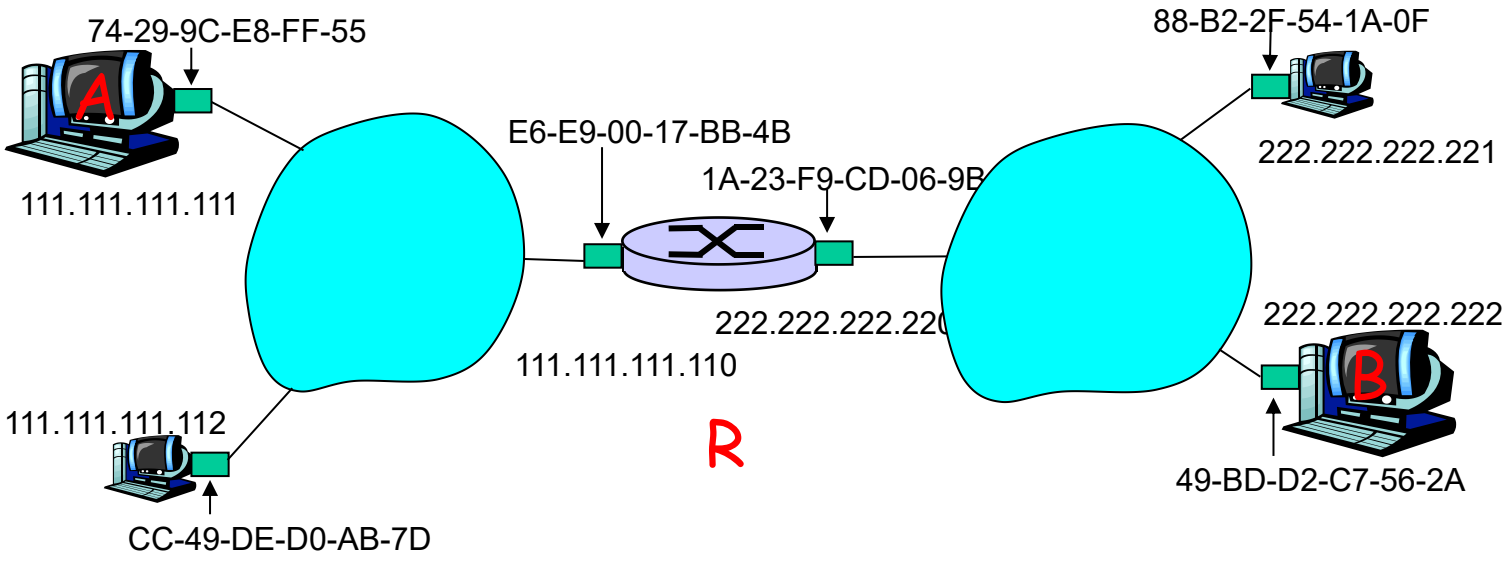
- 每个节点在内存中维护一个地址映射（绑定）表，称ARP缓存
- 每次发送数据报前先查询ARP缓存，若找不到则发送ARP请求，并在收到ARP响应后将地址映射缓存起来
- ARP缓存中的信息，在超时（一般为15～20分钟）后删除

改进ARP的措施：主动学习

- 从ARP请求中获取地址绑定信息：
  - 每个节点可以收到全部的ARP请求报文，可将发送节点的地址映射缓存到自己的ARP表中
- 节点在启动时自动广播自己的地址映射：
  - 节点A在启动时主动广播一个ARP请求，在目标字段内填入自己的IP地址
  - 收到ARP请求的节点将A的地址映射缓存起来
  - 若A收到ARP响应，报告IP地址重复错误

数据报如何从源主机到达目的主机

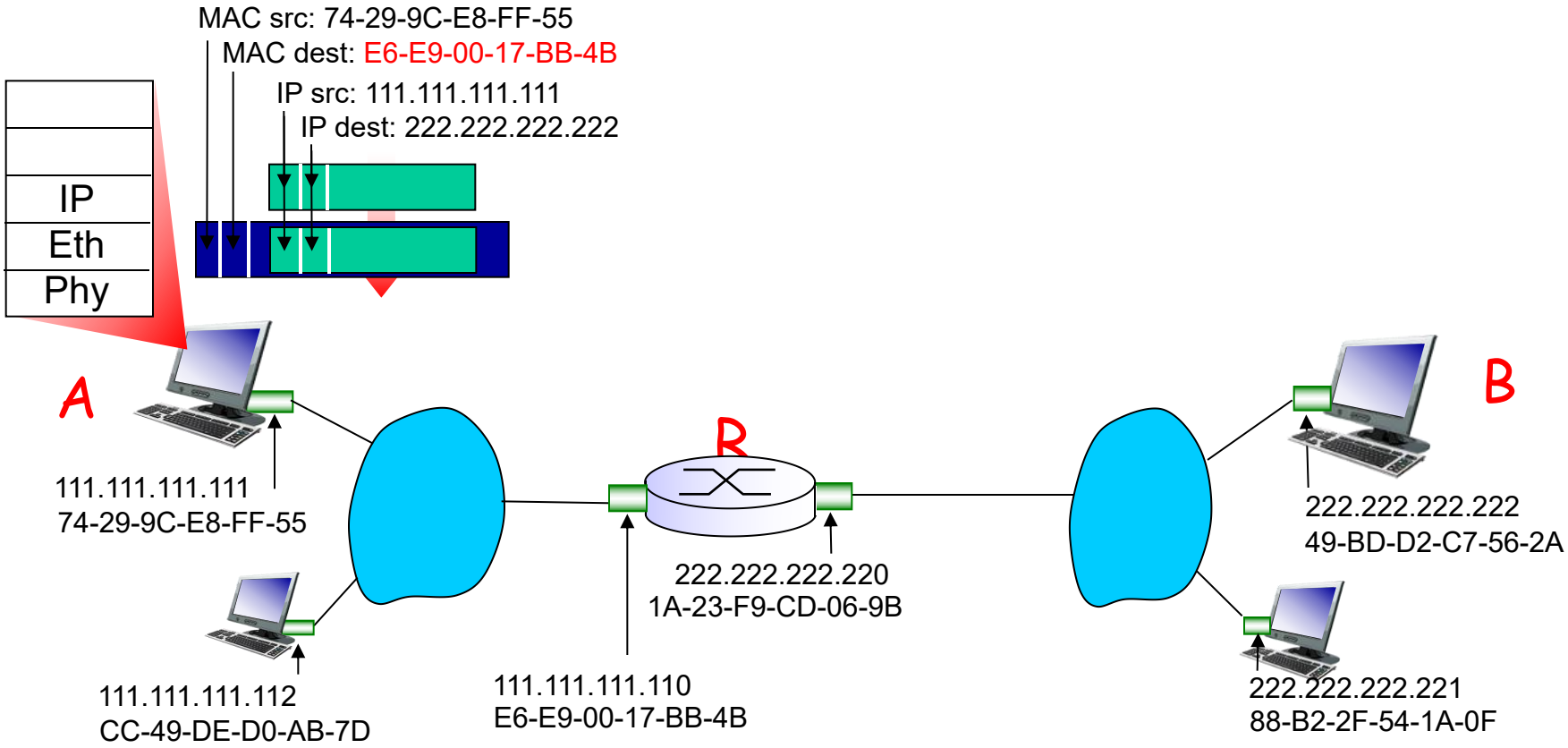
数据报从A经过R到达B：



- 根据转发表，A知道下一跳为111.111.111.110（R-1）
- 根据转发表，R知道B从其端口R-2直接可达

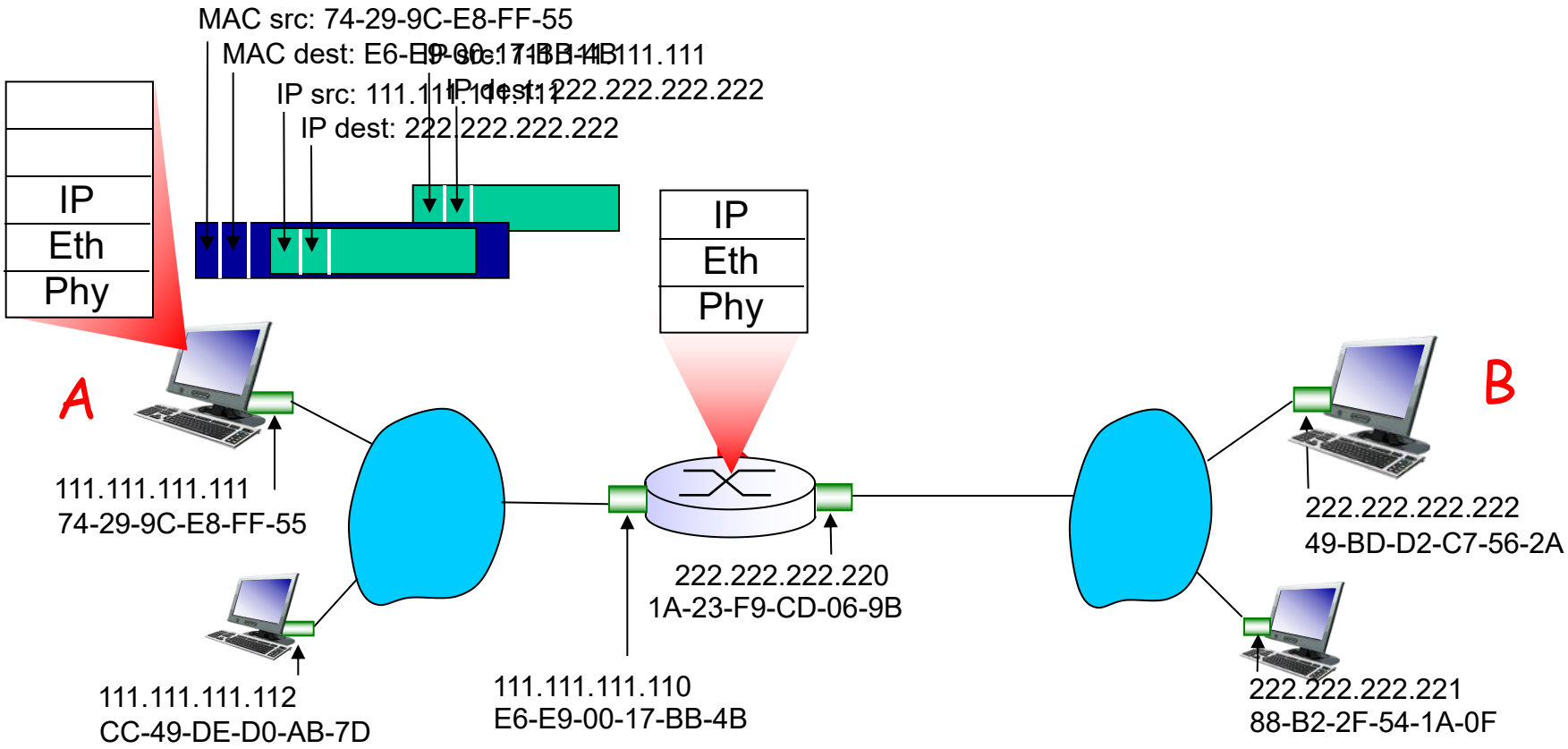
数据报如何从源主机到达目的主机

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram



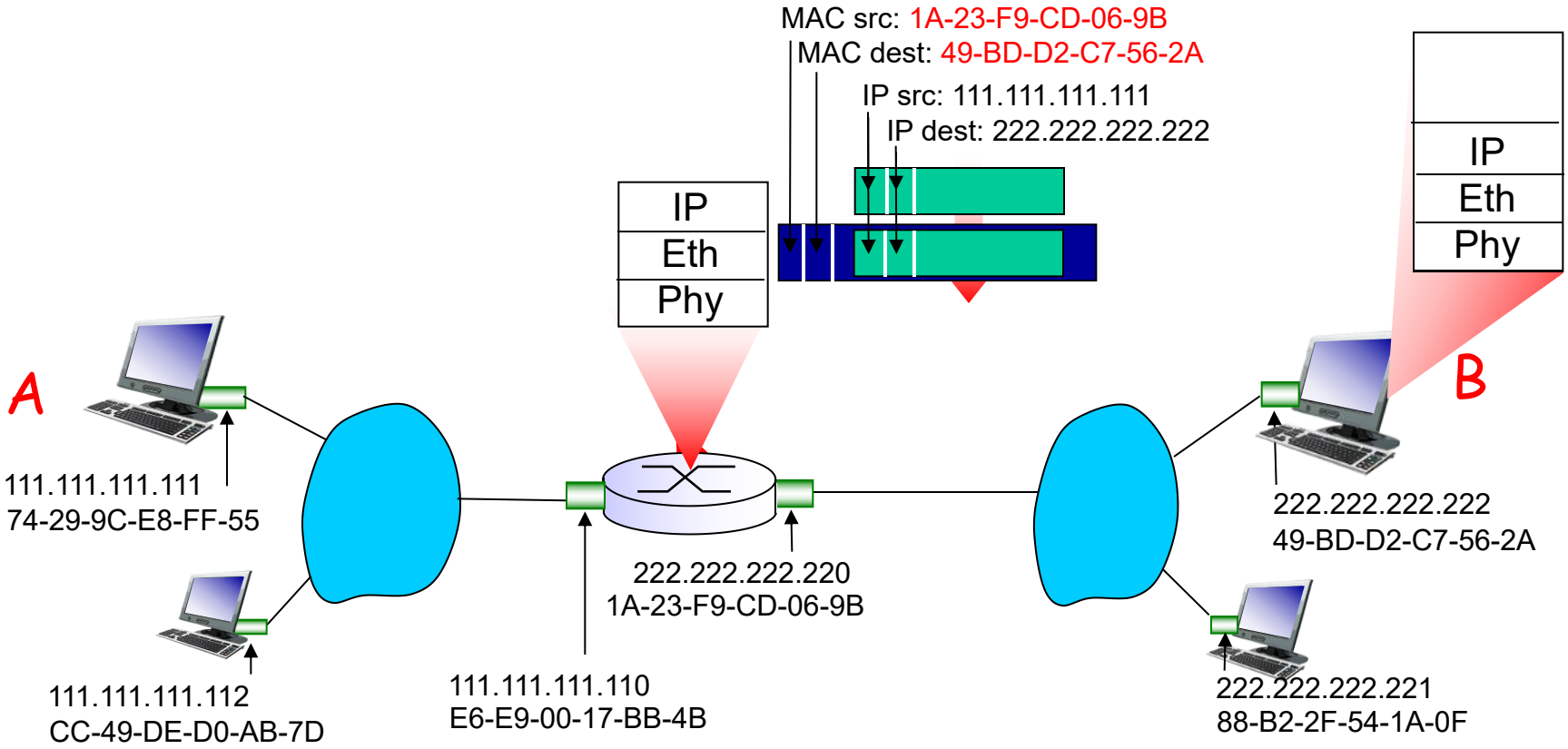
数据报如何从源主机到达目的主机

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



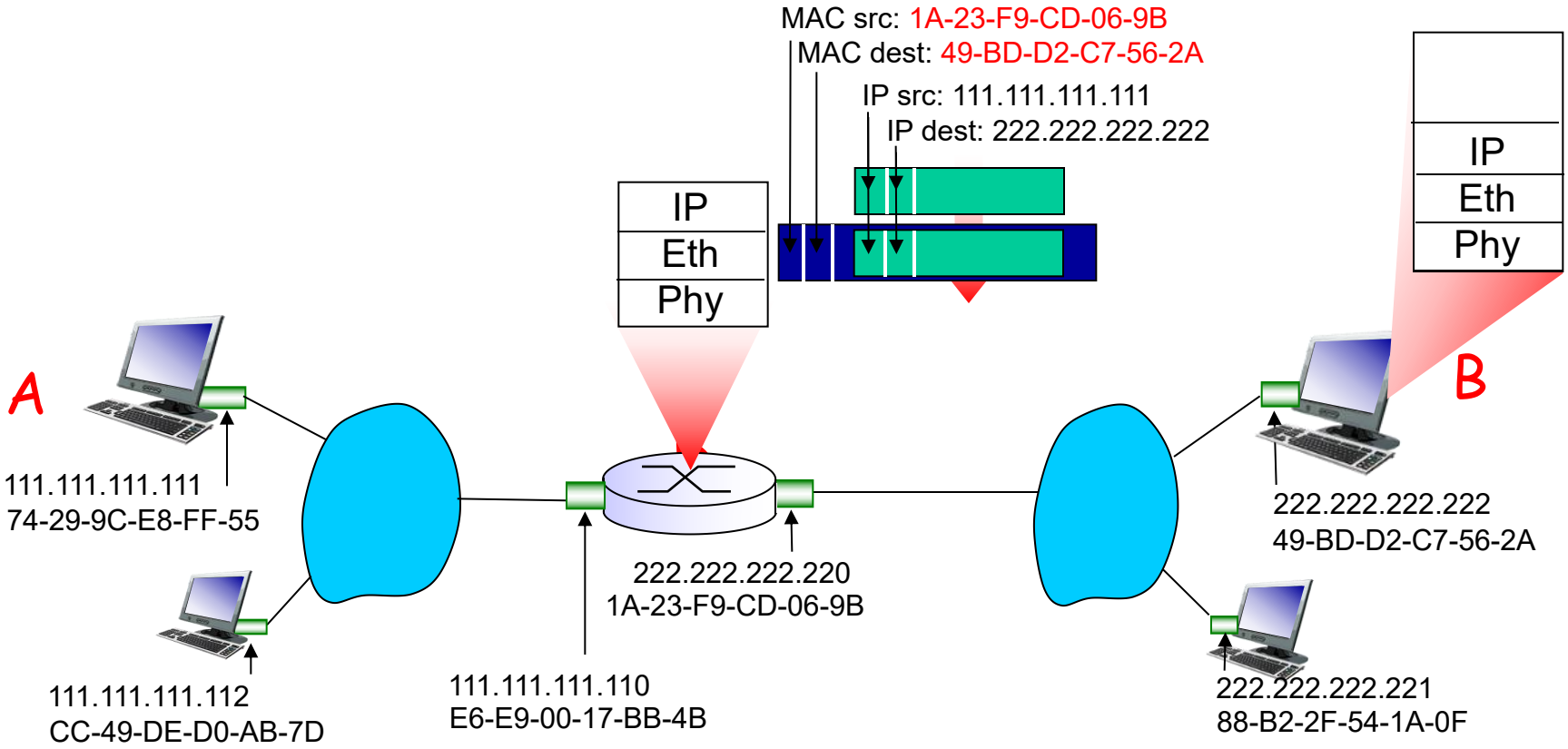
数据报如何从源主机到达目的主机

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



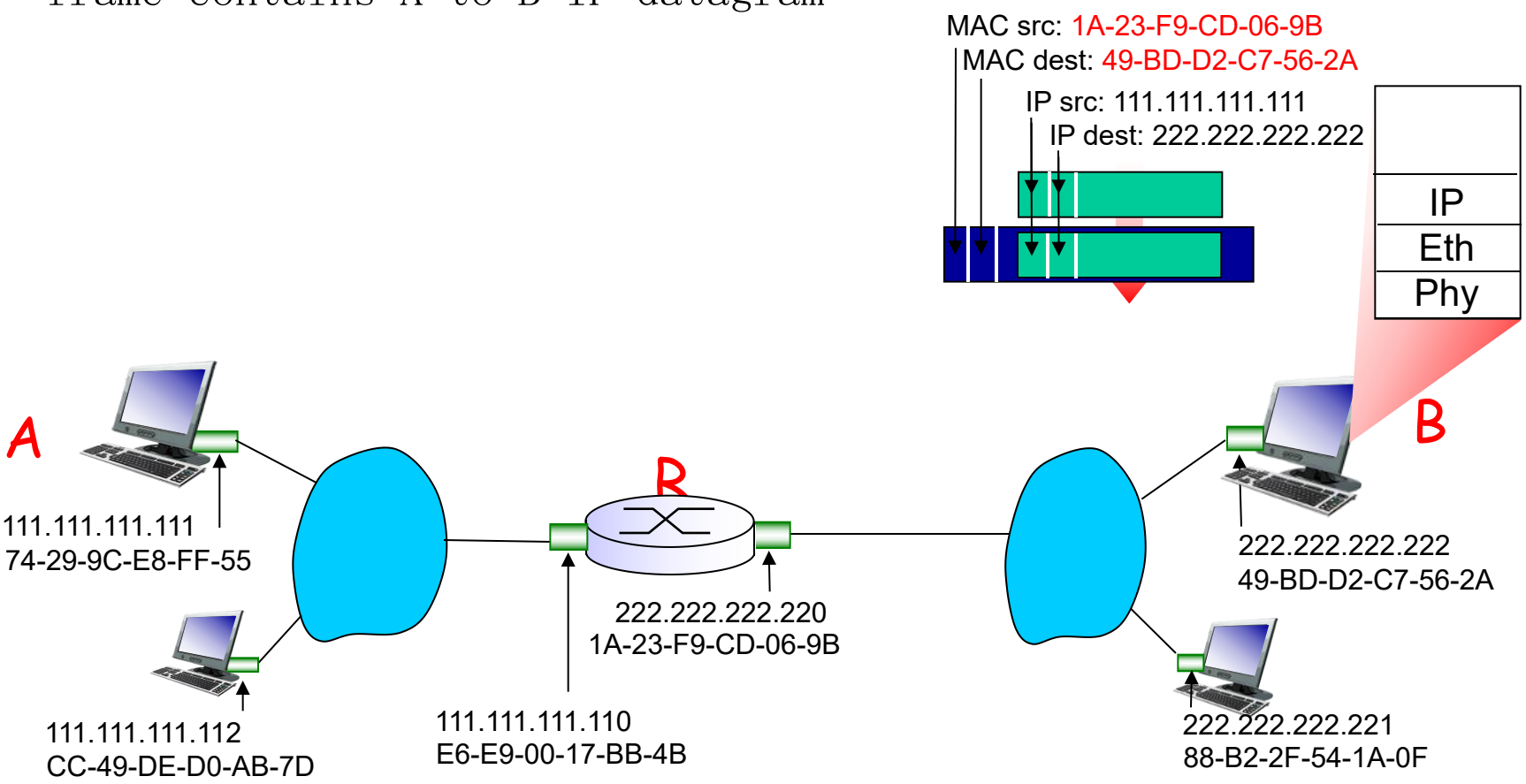
数据报如何从源主机到达目的主机

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



数据报如何从源主机到达目的主机

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



** Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/*

重要的知识点

- 为什么有了MAC地址，还需要IP地址？
- 地址解析：
  - ARP过程，ARP缓存
- 分组逐跳转发的过程：
  - 仔细梳理源主机、路由器、目的主机上分别进行了什么操作，分组是如何逐跳地从源主机经路由器到达目的主机的

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization

5.6 data center networking

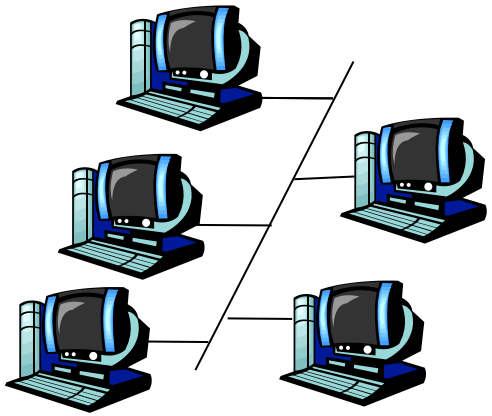
5.7 a day in the life of a web request

以太网

- 第一个广泛应用的局域网技术，也是目前占主导地位的有线局域网技术
- 与其它的局域网技术相比，技术简单、成本低
- 为提高速率，以太网技术不断演化和发展
- 速率持续提高：10 Mbps -> 100Mbps -> 1Gbps -> 10 Gbps -> 40Gbps -> 100Gbps -> …

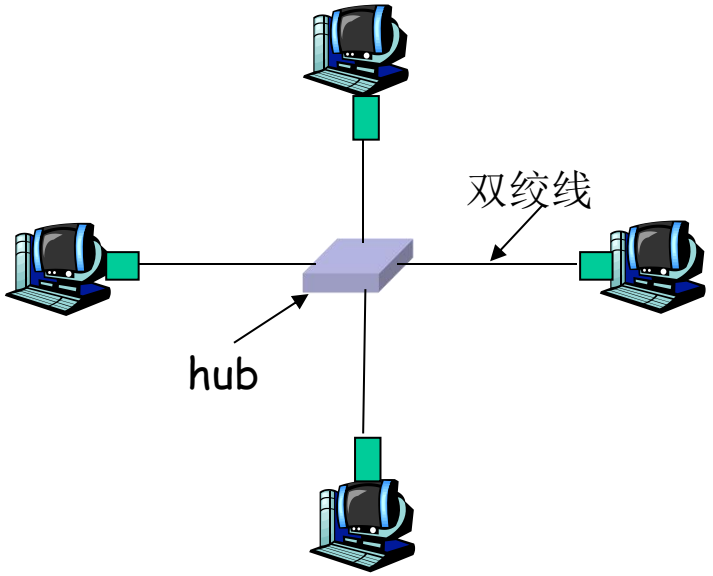
总线拓扑：共享式以太网

- 总线（1970s中期）：
  - 以同轴电缆作为共享传输媒体（总线）
  - 所有节点通过特殊接口连接到这条总线上



总线: 同轴电缆

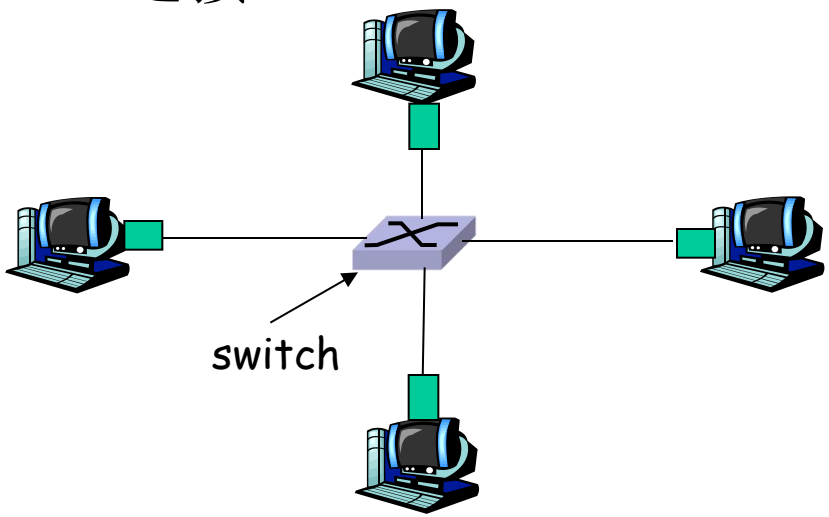
- 集线器（1990s后期）：
  - 一个物理层中继器，从一个端口进入的物理信号（光，电），放大后立即从其它端口输出
  - 集线器相当于共享电缆



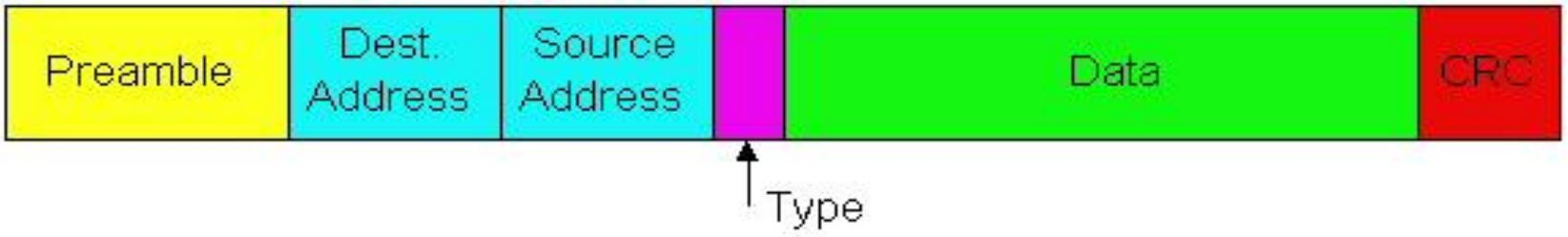
星型拓扑：交换式以太网

- 交换机（21世纪早期）：
  - 主机通过双绞线或光纤连接到交换机
  - 主机与交换机之间为全双工链路
  - 交换机在端口之间存储转发帧（链路层设备）
- **交换式以太网不会产生冲突，不需使用CSMA/CD协议！**

- 星型拓扑：
  - 各节点仅与中心节点直接通信，各节点之间不直接通信
  - 不同于基于hub的星型连接



以太帧结构



- Preamble（前导码）：
  - 7个10101010字节，后跟一个10101011字节，用于在发送方和接收方之间建立时钟同步
  - 一般不计入以太帧的长度
- Dest.Address/Src Address：目的/源MAC地址
- Type（2字节）：指出Data所属的高层协议（如IP、ARP等），每个协议有一个编号
- Data：46～1500字节，不足46字节填充至46字节
- CRC（4字节）：对dest addr.、src addr.、type和data四个字段计算得到的CRC码

无连接、不可靠的数据传输

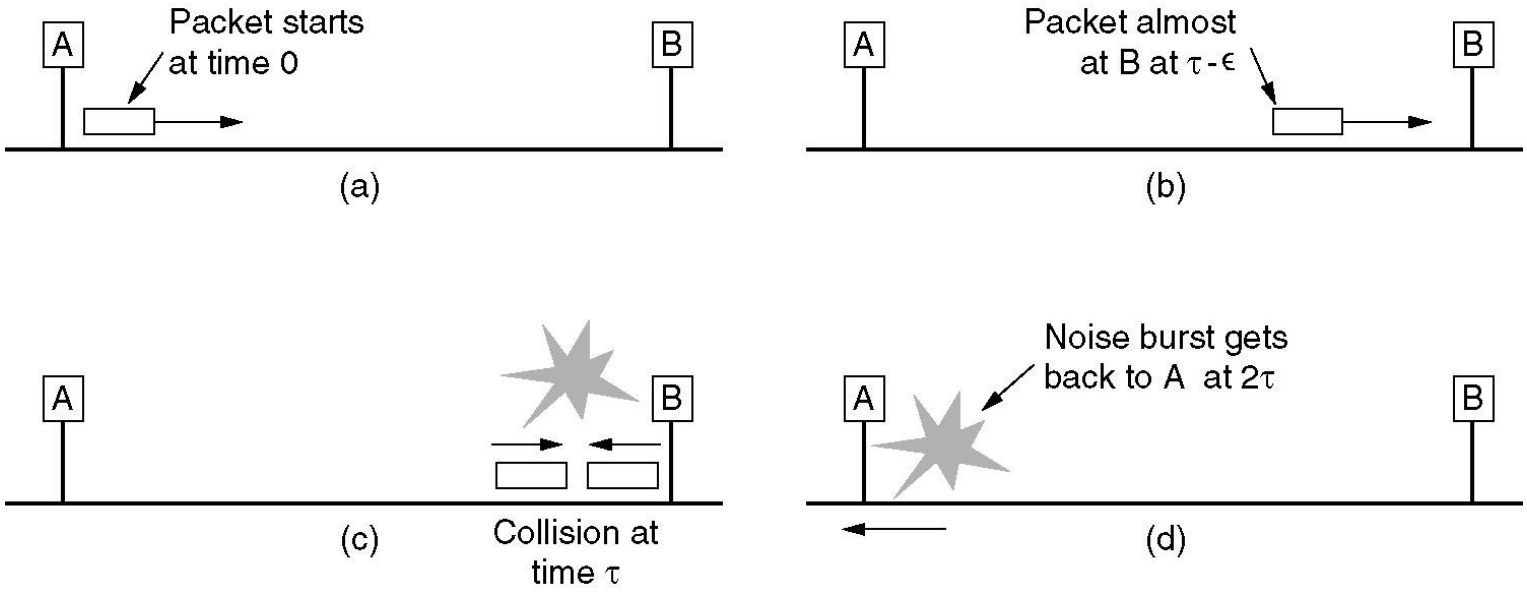
- 无连接:
  - 发送方网卡与接收方网卡之间没有握手
- 不可靠:
  - 接收方网卡不发送确认
  - 接收方网卡丢弃CRC错误的帧
  - 依靠上层协议（TCP或应用）进行错误恢复

为什么有最小帧长的要求？

- CSMA/CD协议规定，发送方仅在发送的过程中检测冲突；为保证在发送结束前检测到冲突，帧的发送时间必须足够长：
  - 节点检测冲突需要时间
  - 假设信号在相距最远的两个适配器之间的往返延迟为2τ，则帧的发送时间不应小于2τ，即帧的最小长度≧链路速率×2τ

- 为什么最小帧长为64字节（不包括前导码）：
  - 根据早期以太网的最大直径（2500米）和数据速率（10Mbps）计算得到

检测冲突需要时间



Collision detection can take as long as 2τ.

802.3以太网标准：链路层 & 物理层

- 历史上出现过许多不同的以太网技术：
  - 链路层相同：MAC协议，帧格式，帧处理
  - 物理层不同：
    - 传输媒体：光纤，同轴电缆，双绞线
    - 数据速率：如10Mbps, 100 Mbps, 1Gbps, …
    - 物理层编码方式
- 所有这些以太网技术由IEEE 802.3工作组标准化，形成IEEE 802.3标准族

10Mbps以太网（早期以太网）

- 10Base-5：
  - 基带同轴电缆（粗），每段电缆最大长度500米
- 10Base-2：
  - 基带同轴电缆（细），每段电缆最大长度约200米
- 10Base-T
  - 3类双绞线和集线器，双绞线最大长度100米
- 10Base-F
  - 多模光纤和集线器，光纤最大长度2000米

100Mbps以太网（快速以太网）

仅能使用光纤/双绞线，以及集线器/交换机

- 100Base-TX（可使用集线器或交换机）：
  - 5类双绞线（2对），不超过100米
- 100Base-T4 （可使用集线器或交换机）：
  - 3类双绞线（4对），不超过100米
- 100Base-FX（只能使用交换机）：
  - 多模光纤（2条），不超过2000米

千兆、万兆以太网

使用交换机，并增加了对流量控制的支持

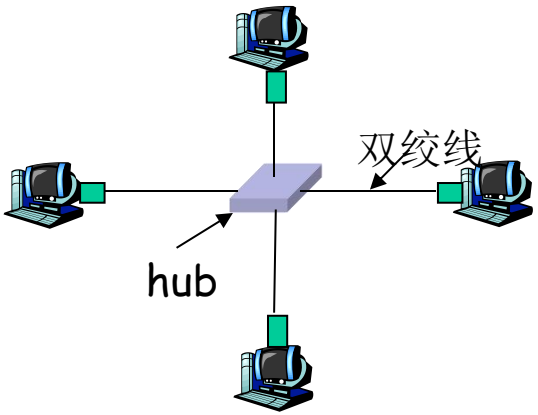
- 1000Base-SX：
  - 多模光纤，不超过550米
- 1000Base-LX：
  - 单模或多模光纤，不超过5000米
- 1000Base-CX（很少用）：
  - 2对屏蔽双绞线，不超过25米
- 1000Base-T：
  - 4对5类双绞线，不超过100米
- 10GBase-T：
  - 只使用光纤，长距离用单模光纤，短距离用多模光纤

DIX以太帧与802.3帧

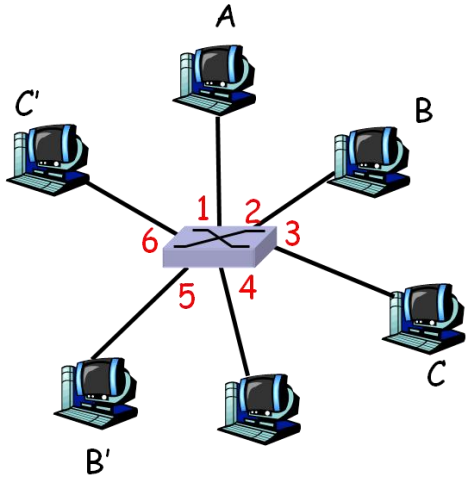
- 最早提出的以太帧称为DIX（DEC-Intel-Xerox）以太帧：
  - type：指出处理data域的协议实体
- 符合IEEE 802.3标准的帧（802.3帧）：
  - length：替代DIX帧中的type域，指出data的长度
- 这两种格式都可使用，当type/length的值大于1500时解释为type，否则解释为length

讨论：共享式以太网和交换式以太网

- 共享式以太网：
  - 集线器的所有端口位于同一个冲突域
  - 任一时刻最多只允许一个主机发送
  - 网络规模（节点数量）与网络性能的矛盾无法解决



- 交换式以太网：
  - 交换机的每个端口为一个冲突域
  - 多对端口可以同时通信
  - 网络的集合带宽=各个端口的带宽之和
  - 从根本上解决了网络规模与网络性能的矛盾



交换式以太网的最小帧长及规模

- 交换式以太网不再使用CSMA/CD协议，理论上不再需要限制帧的最小长度。但为了向后兼容，帧格式及最小帧长度的限制仍然保持不变
- 由于交换式以太网不再使用CSMA/CD协议，网络直径不再受到信号最大往返时间的限制
- 交换式以太网的MAC层除了帧格式保持不变外，其它都和共享式以太网不同了

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

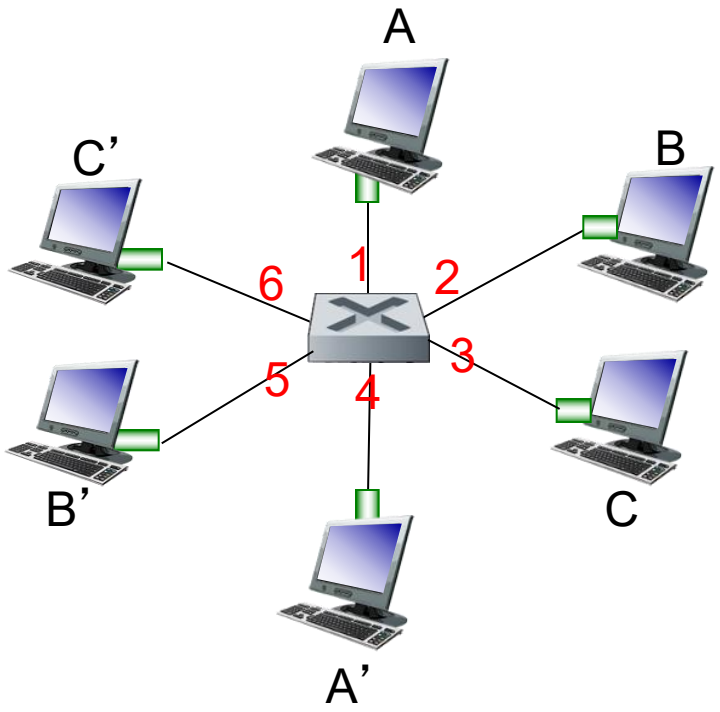
5.5 link virtualization

5.6 data center networking

5.7 a day in the life of a web request

以太网交换机

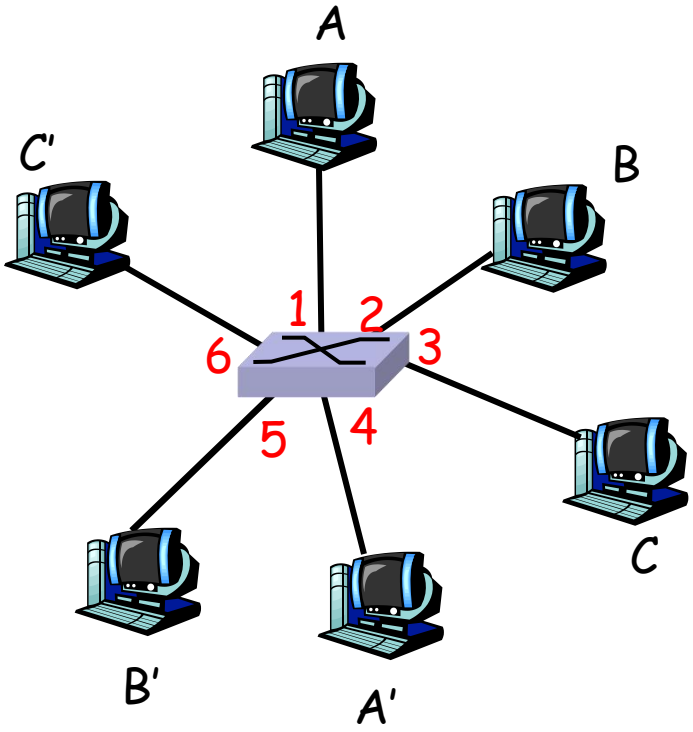
- 链路层设备：
  - 存储-转发帧：检查输入帧的MAC地址，有选择地将帧转发到一条或多条输出链路
- 对主机透明
  - 交换机没有MAC地址，主机不需要了解有关交换机的任何信息，感觉不到交换机的存在
- 即插即用，自主学习
  - 交换机不需要配置



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

交换机如何转发？

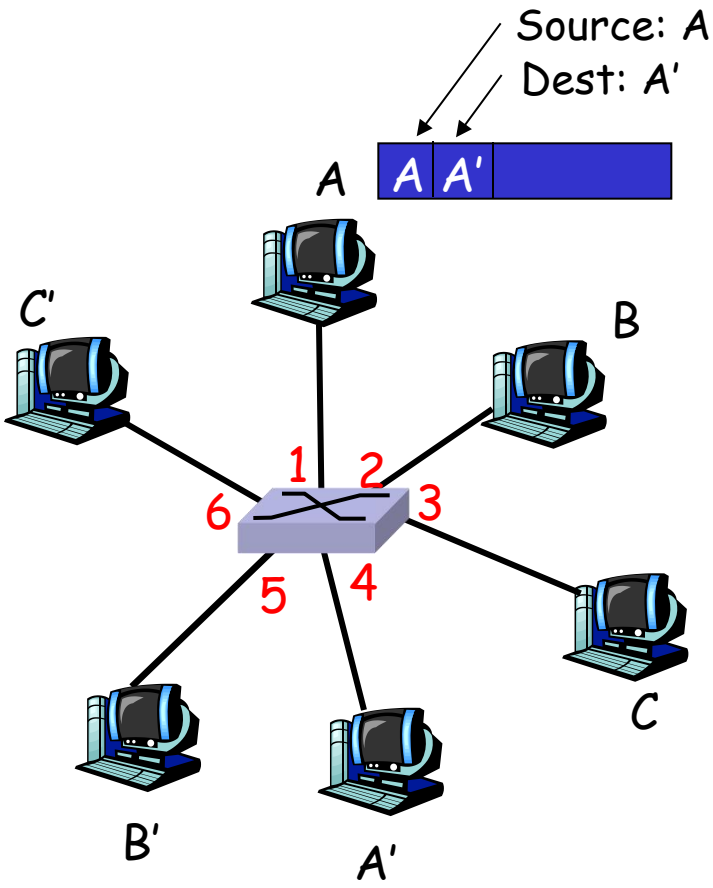
- Q: 交换机如何知道A'通过端口4可达，而B'通过端口5可达？
- A: 每个交换机内部有一张转发表，每个表项记录以下信息：
  - MAC地址，去往该MAC地址的端口
- Q: 转发表是如何建立和维护的？



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

自主学习

- 交换机自主学习“哪个主机通过哪个端口可达”：
  - 当一个帧到达时，交换机从源MAC地址了解到发送节点，从帧到来的端口了解到发送节点的位置（从该端口可达）
  - 在转发表中记录发送节点和可达端口



MAC addr	interface	TTL
A	1	60

转发表
（初始为空）

帧的过滤和转发

当帧到来时：

1. 记录帧的到来端口（自学习）
2. 用帧的目的MAC地址查找转发表
3. **if** 找到目的MAC地址　//已知节点
   then {
   if 目的地址所在端口=帧的到来端口
   then 丢弃帧　//过滤不需要转发的帧
   else 转发帧到表项指定的端口　//按转发表转发帧
   }
   else 扩散帧　//未知节点，采用扩散法转发

向输入端口以外的所有端口转发

交换机收到帧的处理过程

- 用帧的目的地址查找转发表（转发决策）：
  - 若目的地址所在端口 = 帧的进入端口，丢弃帧
  - 若目的地址所在端口 ≠ 帧的进入端口，转发帧
  - 若目的地址不在转发表中，扩散帧
- 用帧的源地址查找转发表（更新转发表）：
  - 若找到地址，更新相应表项
  - 若没有找到该地址，添加源地址和进入端口到转发表，设置表项的生存期为最大值

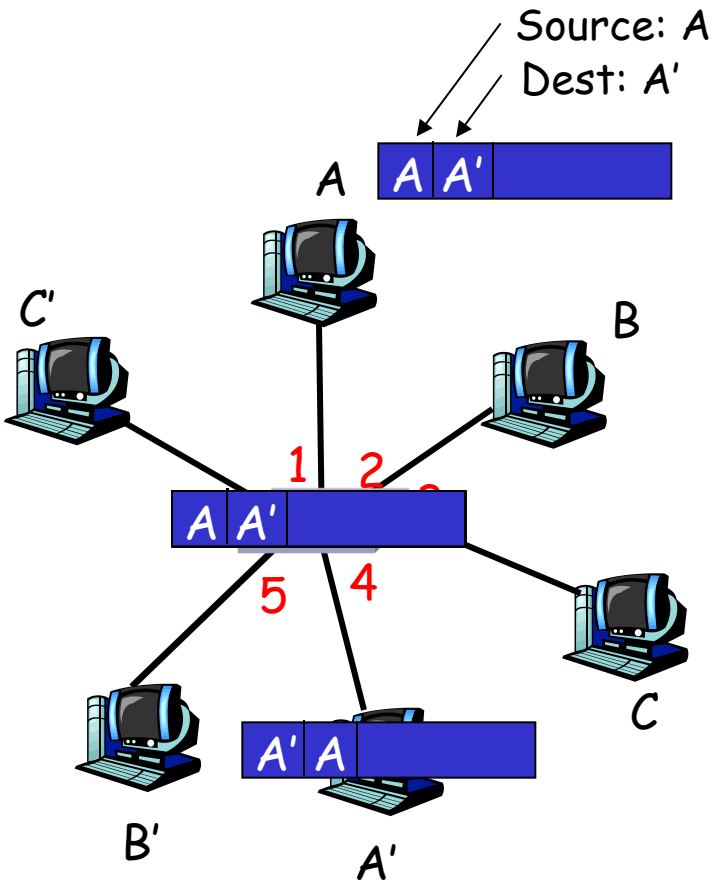
举例

- 目的地址未知：

  扩散

- 目的地址*A*已知：

  按照转发表转发

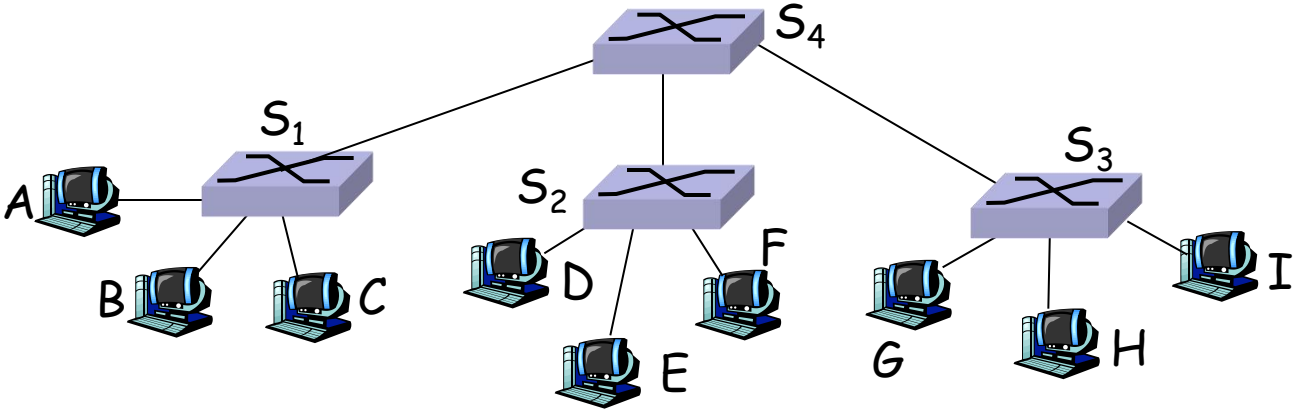


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

转发表
（初始为空）

级联交换机

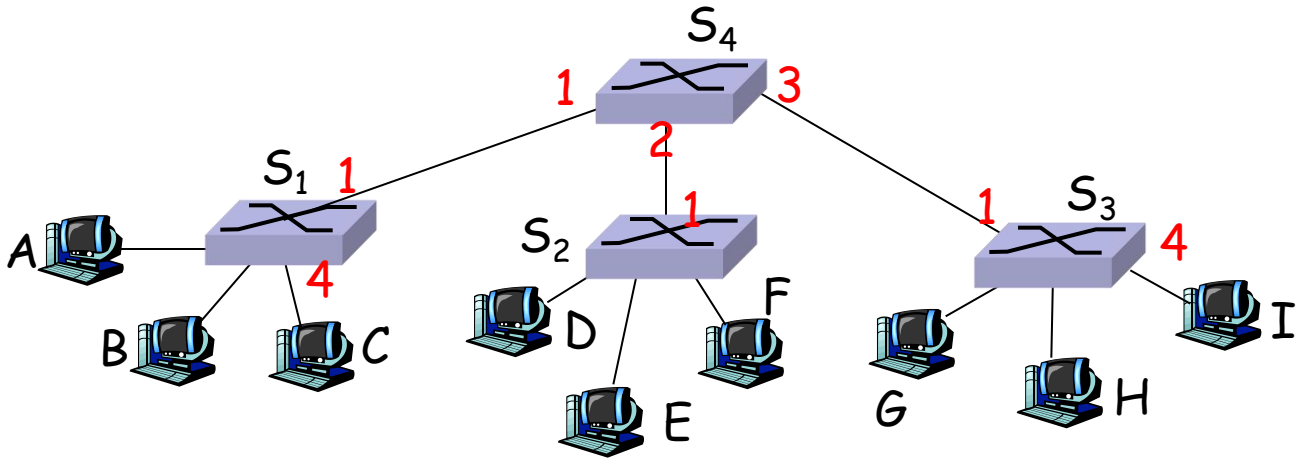
- 多个交换机也可以级联在一起，将更多或更大范围内的节点连接到一个网段中



- Q: 数据包要从A发往F，交换机S_1如何知道应转发给S_4，而S_4如何知道应转发给S_2？
- A: 通过自主学习！（与单交换机情形相同）

举例

假设C发送一个帧给I，I响应C：

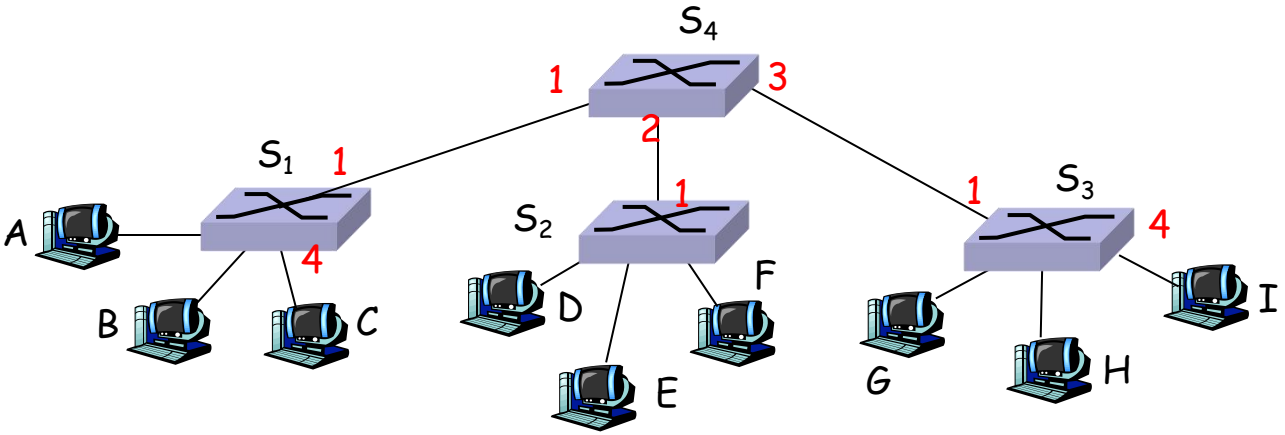


- Q: 给出 S_1, S_2, S_3, S_4 中的转发表和包转发决策

解答

1、C发送一个帧给I

2、I发送一个帧给C



S1

MAC addr	interface	action
C	4	扩散
I	1	转发

S2

MAC addr	interface	action
C	1	扩散

S3

MAC addr	interface	action
C	1	扩散
I	4	转发

S4

MAC addr	interface	action
C	1	扩散
I	3	转发

有环网络和生成树算法

- 实际的网络是有环网络：
  - 实际网络不采用树状结构（可靠性不高），而是存在冗余链路，即网络中存在环
  - 在有环的网络中扩散帧会造成冗余传输，且不能终止
- 解决方法（构造网络的生成树）：
  - 在有环的物理网络上建立一个无环的网络拓扑（生成树），正常情况下只使用无环拓扑转发帧
- IEEE 802.1D标准化了构造生成树的分布式算法：
  - 首先选举具有最小序列号的交换机作为生成树的根
  - 按照根到各个交换机的最短路径构造生成树
  - 只有位于生成树中的交换机可以在属于生成树的边上发送
  - 当有节点或链路发生故障时，重新计算一个无环拓扑

交换机 vs. 路由器

- 交换机工作于链路层，根据MAC地址存储转发帧
- 路由器工作于网络层，根据IP地址存储转发数据报

- 交换机不能连接异构链路（即MAC协议不同的网络），因为交换机只是按原样转发帧
- 路由器可以连接异构链路，因为路由器需重新封装链路层帧

交换机 vs. 路由器

- 交换机不能阻断广播帧的传播：
  - 交换机只能学习到单播MAC地址，所有广播帧都会扩散发送
  - 通过交换机连接的所有主机在同一个广播域中
- 路由器可以阻断广播帧的传播：
  - 路由器根据IP地址转发包（看不到MAC地址）
  - 每个路由器端口是一个独立的广播域

- 冲突域：
  - 共享同一条广播链路的主机集合
  - 任何一个主机发送的帧（各种帧），可被冲突域中的其它主机接收到
- 广播域：
  - 广播帧能够到达的主机集合
  - 广播风暴：广播帧在网络中大量传播，消耗大量资源

三层交换机和路由器

- 路由器可分隔二层网络，但转发速度慢、成本高
- 三层交换机：
  - 具有部分路由功能、又有二层转发速度的交换机
  - 专为加快大型局域网内部的数据交换而设计
  - 但在安全、协议支持等方面不如专业路由器
- 机构网络中三层交换机和路由器的使用：
  - 三层交换机：通常用在机构网络的核心层，连接不同的子网或虚拟局域网（每个虚拟局域网是一个独立的子网）
  - 专业路由器：连接机构网络与外网

三层交换机为什么快？

- 路由器转发IP包的过程：
  1. 用目的IP地址查找转发表，获得下一跳IP地址及端口
  2. 利用ARP获得下一跳MAC地址
  3. 用下一跳MAC地址构造链路层帧，发送

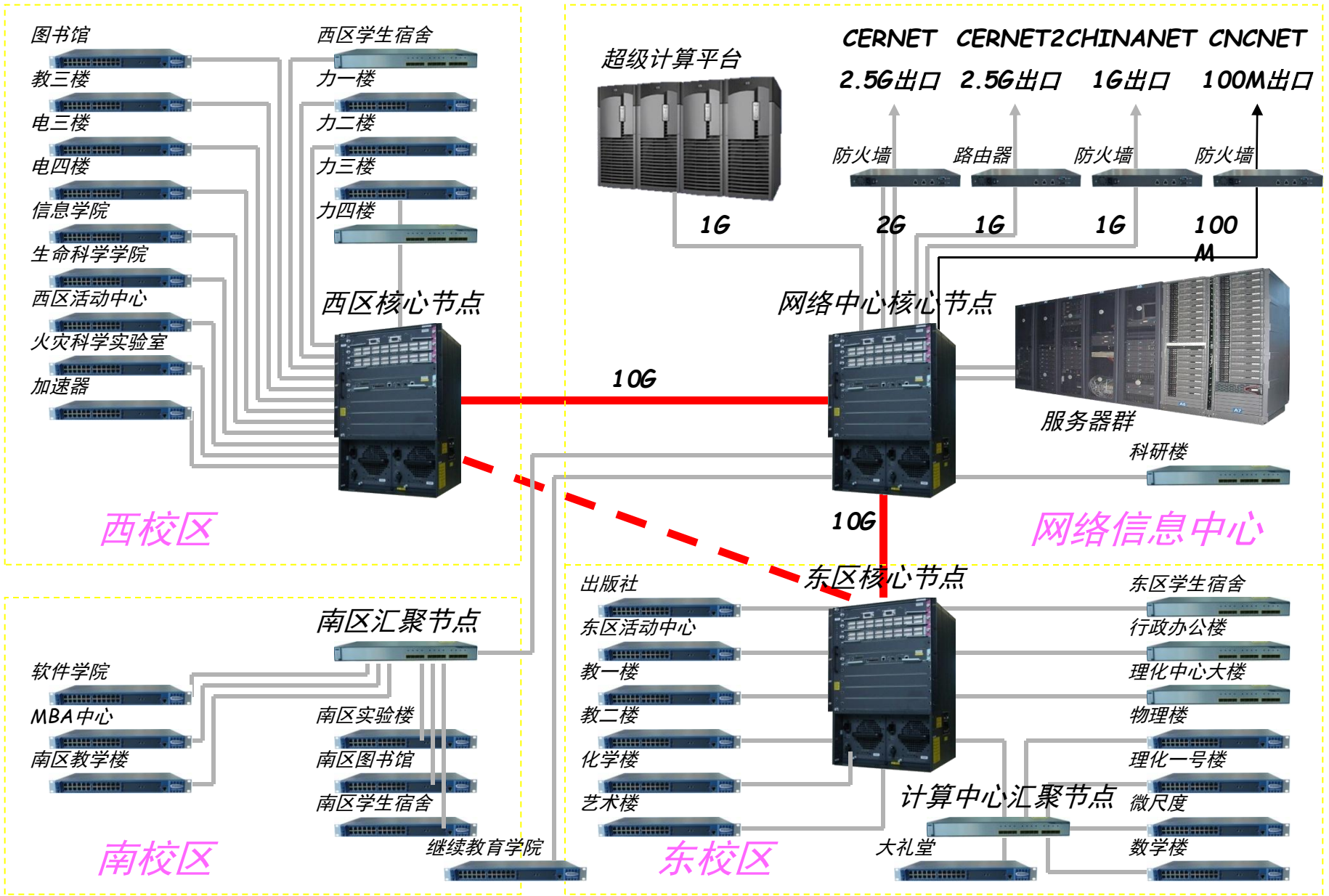
- 三层交换机转发IP包的过程：
  1. 将以上第1、第2步的结果缓存起来（以目的IP地址为索引，哈希表）
  2. 用目的IP地址查找缓存（一次精确匹配）：

     1）若命中，直接用下一跳MAC地址构造链路层帧，发送

     2）若未命中，执行以上第1、2、3步

- 三层交换机转发速度快的原因：
  - **一次选路，多次转发**

中国科学技术大学校园网络示意图（非最新）



Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization

5.6 data center networking

5.7 a day in the life of a web request

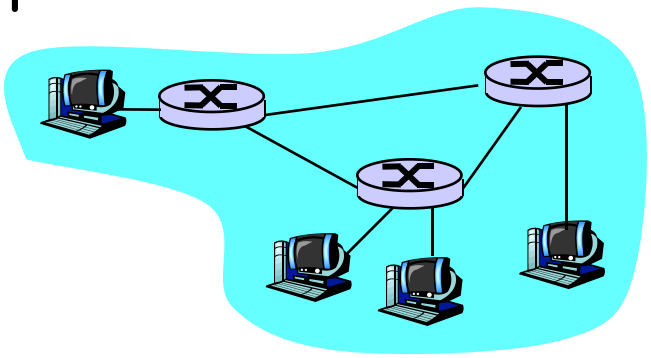
The Internet: virtualizing networks

1974: 多个不连通的网络

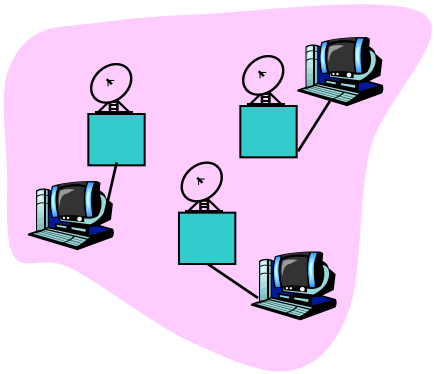
- ARPAnet
- data-over-cable networks
- packet satellite network (Aloha)
- packet radio network

它们在以下方面不同:

- *编址方法*
- *包格式*
- *差错恢复*
- *选路*



ARPAnet



satellite net

"A Protocol for Packet Network Intercommunication", V. Cerf, R. Kahn, IEEE Transactions on Communications, May, 1974, pp. 637-648.

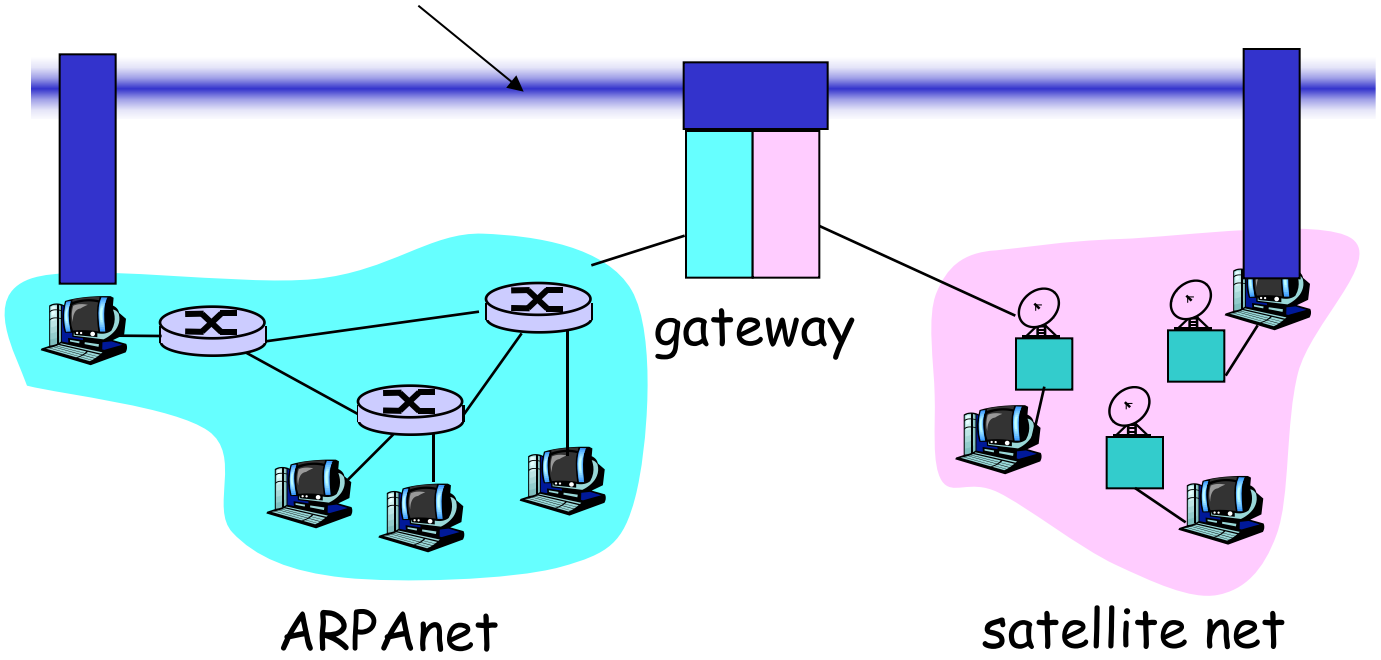
The Internet: virtualizing networks

在物理网络上增加一个逻辑层次（IP层）

- 在逻辑层上统一编址、统一包格式
- 互联在一起的网络看起来像一个网络
- network of networks

用网关连接不同的物理网络:

- 在逻辑层上选路到下一个网关
- 将IP包封装在本地网络帧中，发送到下一个网关



Cerf & Kahn's Internetwork Architecture

- 两级编址: IP网络，物理网络
- IP层提供统一的网络视图：地址，包格式
- 底层可以是任意的物理网络：
  - cable
  - satellite
  - 56K telephone modem
  - today: ATM, MPLS
  - …
- 物理网络对于IP层是不可见的，对于IP来说物理网络只是一条虚拟链路而已！

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

64 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization

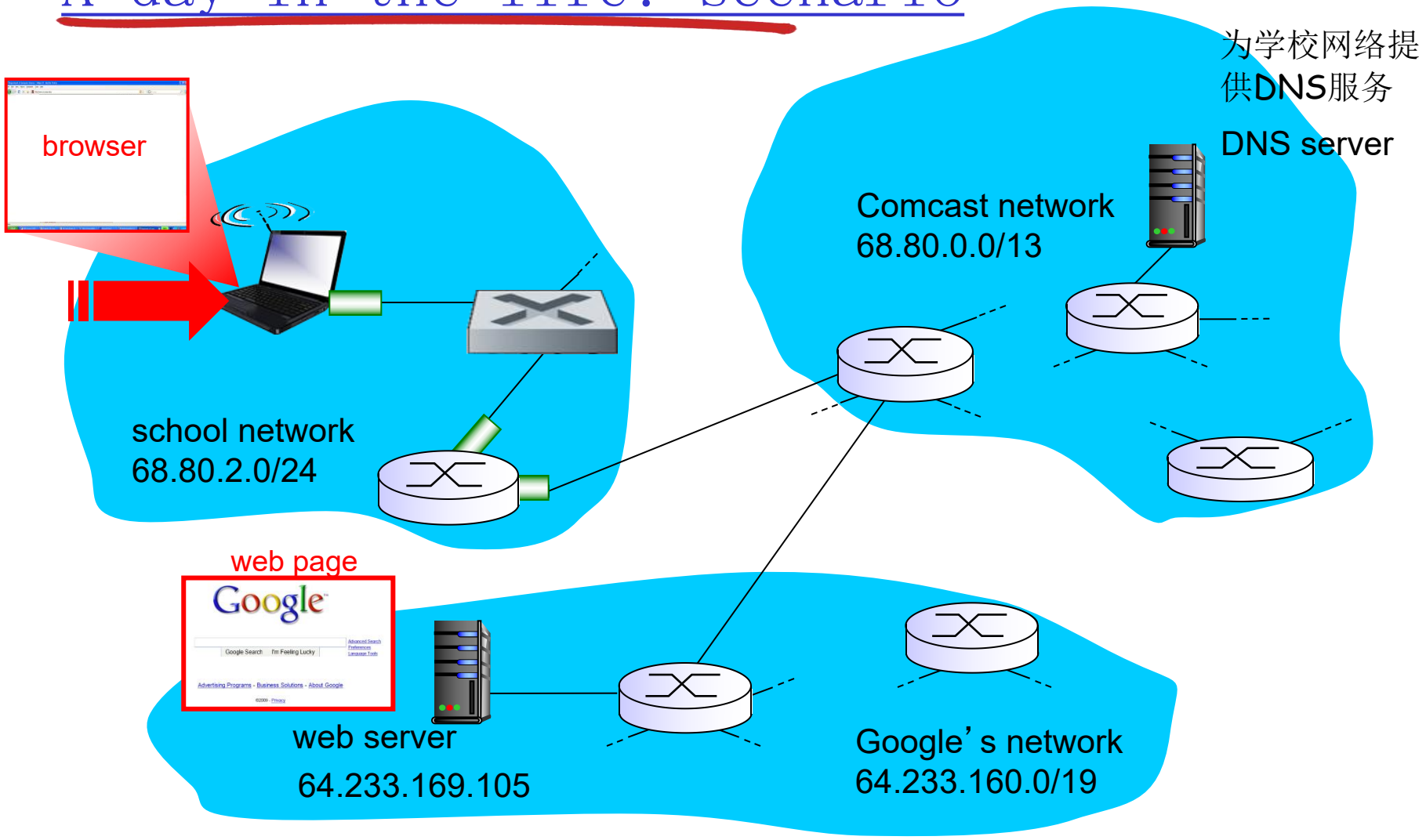
6.6 data center networking

6.7 a day in the life of a web request

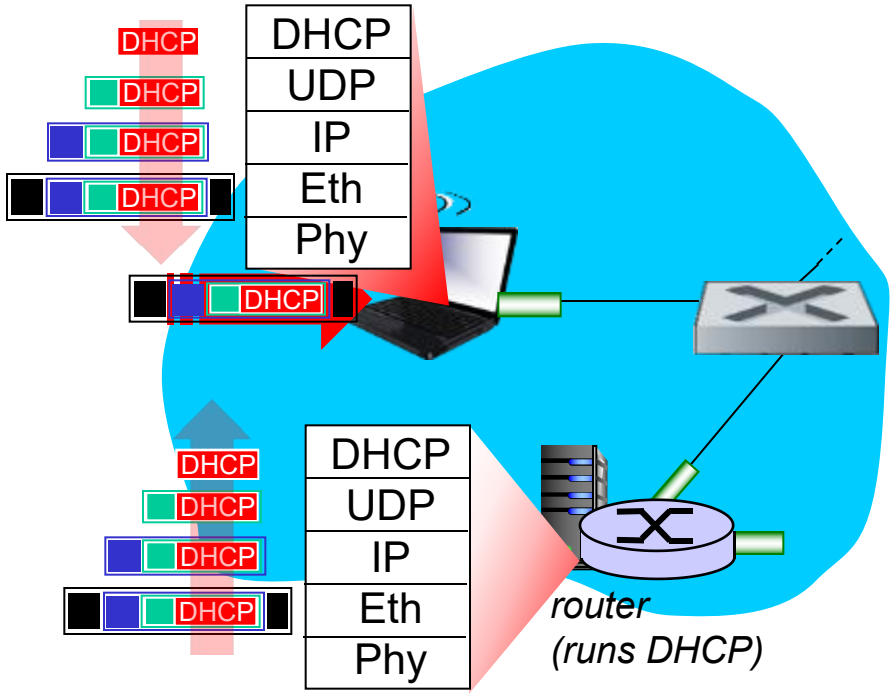
Synthesis: a day in the life of a web request

- 我们已经从上到下走过了整个协议栈：
  - 应用层，传输层，网络层，数据链路层
- 现在将所有内容综合起来：
  - 目的：使用一个简单的场景来复习和理解相关的协议
  - 简单场景：将一台笔记本电脑连入校园网，请求网页www.google.com

A day in the life: scenario

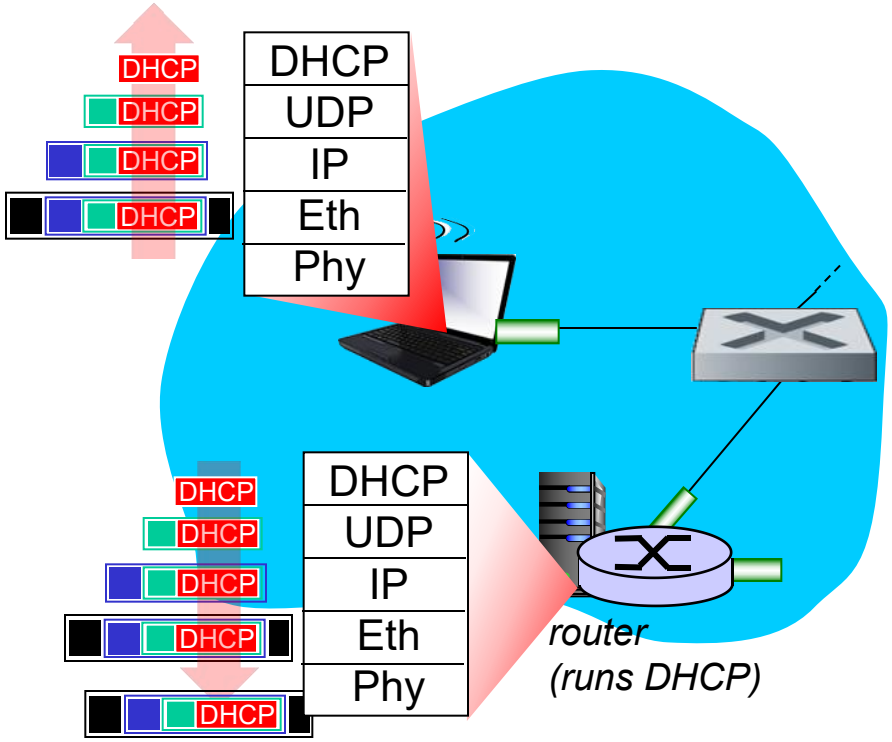


A day in the life… connecting to the Internet



- 笔记本电脑需要获取IP地址、第一跳路由器的IP地址、DNS服务器的IP地址：使用DHCP
  - DHCP请求被封装在UDP/IP/Ethernet中
  - 以太帧在局域网中广播(dest: FFFFFFFFFFFF)，被运行了DHCP服务器的路由器收到
  - 经过层层解封装，DHCP请求到达DHCP服务器

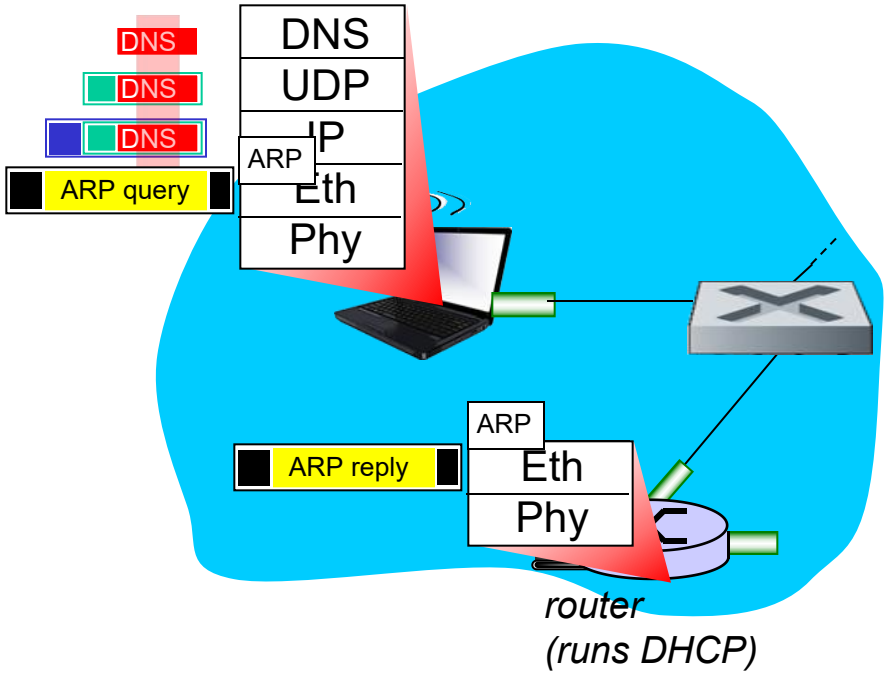
A day in the life… connecting to the Internet



- DHCP服务器构造DHCP ACK，包含客户的IP地址、第一跳路由器的IP地址、DNS服务器名字和IP地址
- DHCP ACK被封装，通过LAN转发 (**交换机自学习**)，到达笔记本
- 经过层层解封装，客户收到DHCP ACK

客户现在知道了IP地址、第一跳路由器的IP地址、DNS服务器的IP地址

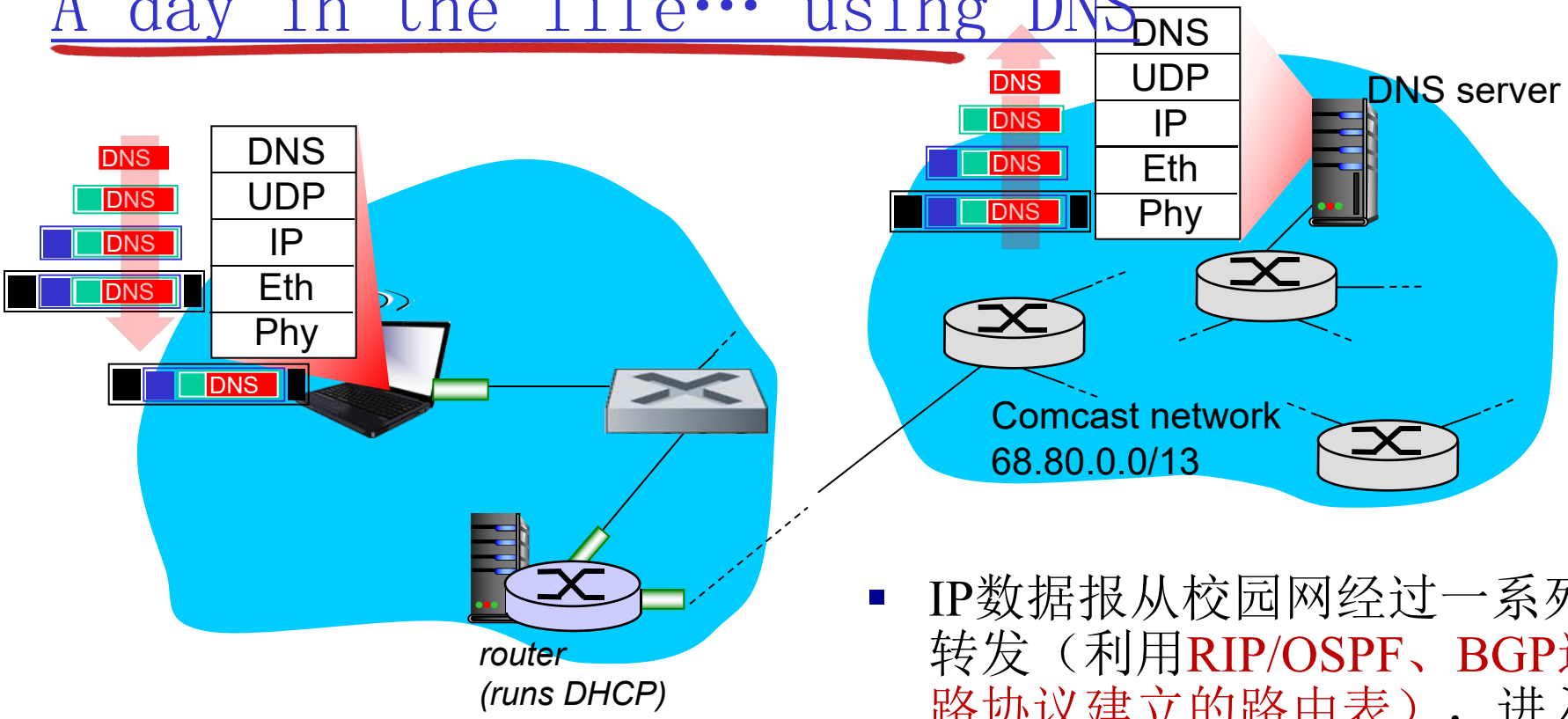
A day in the life… ARP (before DNS, before HTTP)



- 浏览器发送HTTP请求前，需要知道www.google.com对应的IP地址：使用DNS
  - 解析器构造DNS查询报文，封装到UDP/IP/Eth中；为发送帧，需要第一跳路由器接口的MAC地址：使用ARP
  - ARP查询报文被广播，到达路由器；路由器构造ARP响应报文，通过LAN转发，到达笔记本

客户现在知道了第一跳路由器对应接口的MAC地址

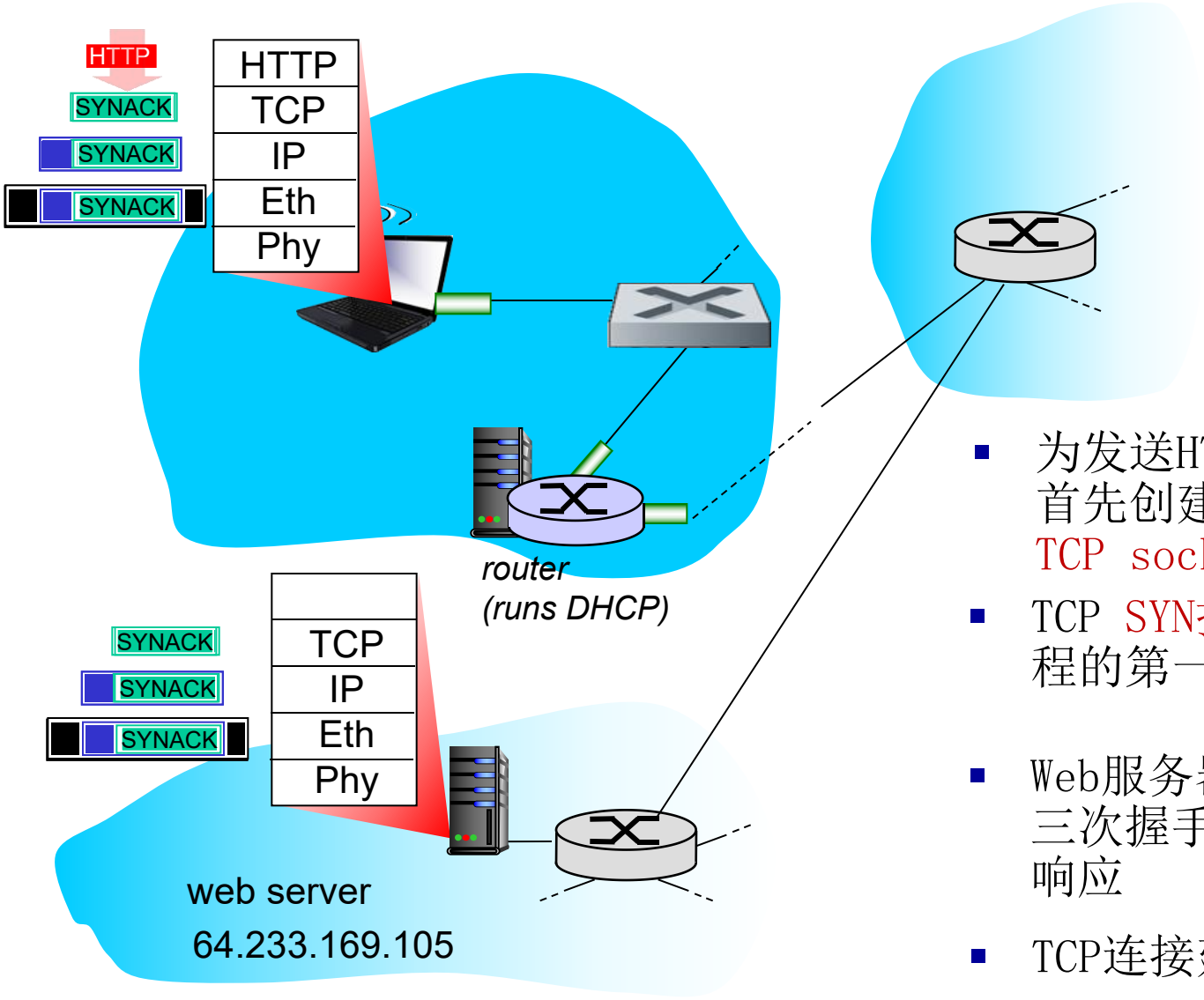
A day in the life… using DNS



- 携带了DNS查询报文的IP包经交换机转发，到达第一跳路由器

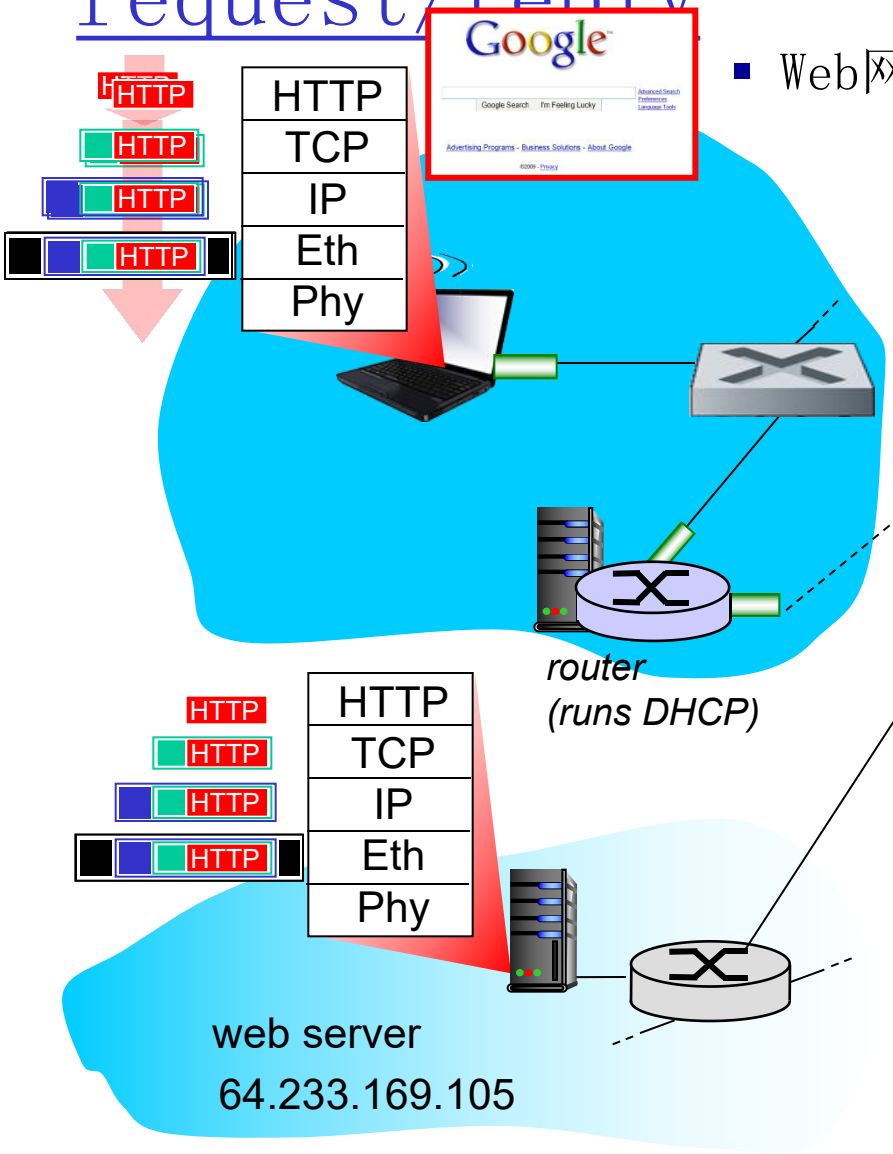
- IP数据报从校园网经过一系列转发（利用RIP/OSPF、BGP选路协议建立的路由表），进入Comcast网络, 到达 DNS服务器
- DNS查询报文经层层解封装到达DNS服务器程序
- DNS服务器用（缓存的）www.google.com对应的IP地址进行响应

A day in the life…TCP connection carrying HTTP



- 为发送HTTP请求，浏览器首先创建到web服务器的TCP socket
- TCP SYN报文段（三次握手过程的第一步）到达web服务器
- Web服务器用TCP SYNACK （三次握手过程的第2步）进行响应
- TCP连接建立

A day in the life… HTTP request/reply



- Web网页终于(!!!)在浏览器中显示

- HTTP request送入TCP套接字
- 携带HTTP请求的IP数据报被路由到www.google.com
- Web服务器用包含网页的HTTP reply 进行响应
- 包含HTTP reply的IP数据报被路由到浏览器

Chapter 6: Summary

- 数据链路层服务原理：
  - 差错检测与纠正
  - 共享广播信道：多址技术
  - 链路层编址，ARP
- 链路层技术实例：
  - 以太网
- 综合: a day in the life of a web request

作业

- 习题（11月28日）
  - 5，8，11，23，24，25，26
- 实验（12月4日）
  - Ethernet-ARP