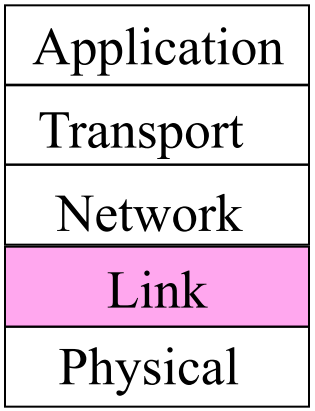
第三章 数据链路层

林海

Lin.hai@whu.edu.cn

数据链路层

- 针对直接相连的两个节点之间的通信
  - 点对点信道
  - 广播信道
- 解决什么问题
  - 成帧
  - 差错检测和纠正
  - 广播信道的接入（MAC）



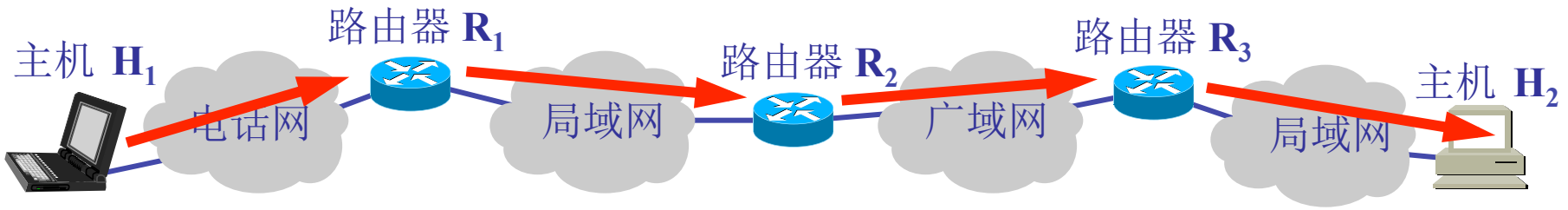
数据链路层使用的信道

数据链路层使用的信道主要有以下两种类型：

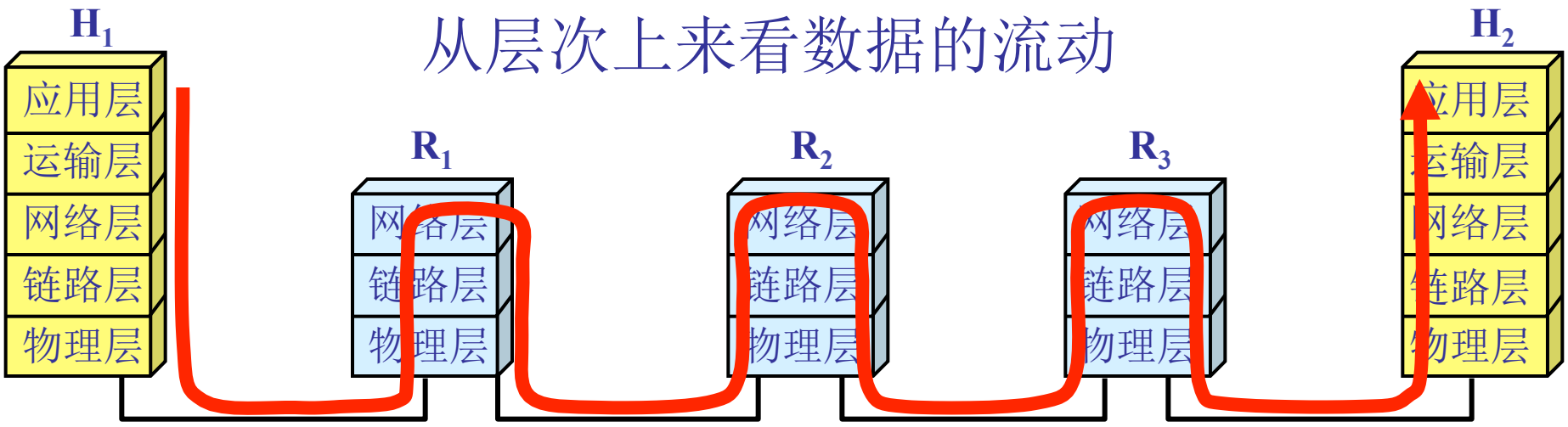
- 点对点信道。这种信道使用一对一的点对点通信方式。
- 广播信道。这种信道使用一对多的广播通信方式，因此过程比较复杂。广播信道上连接的主机很多，因此必须使用专用的共享信道协议来协调这些主机的数据发送。

数据链路层的简单模型

主机 H_1 向 H_2 发送数据

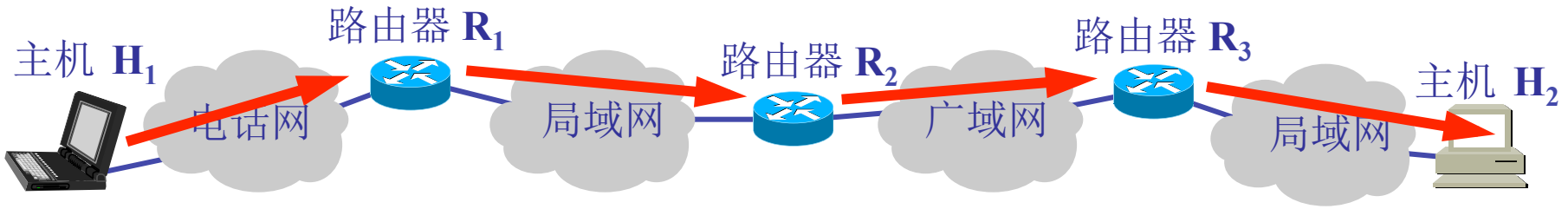


从层次上来看数据的流动

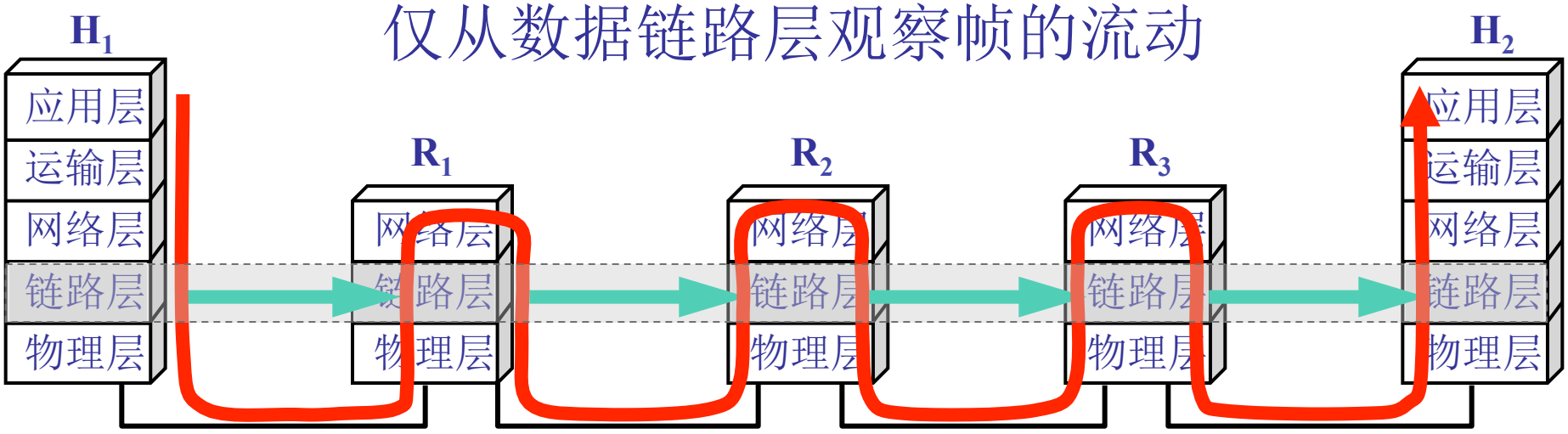


数据链路层的简单模型（续）

主机 H_1 向 H_2 发送数据



仅从数据链路层观察帧的流动



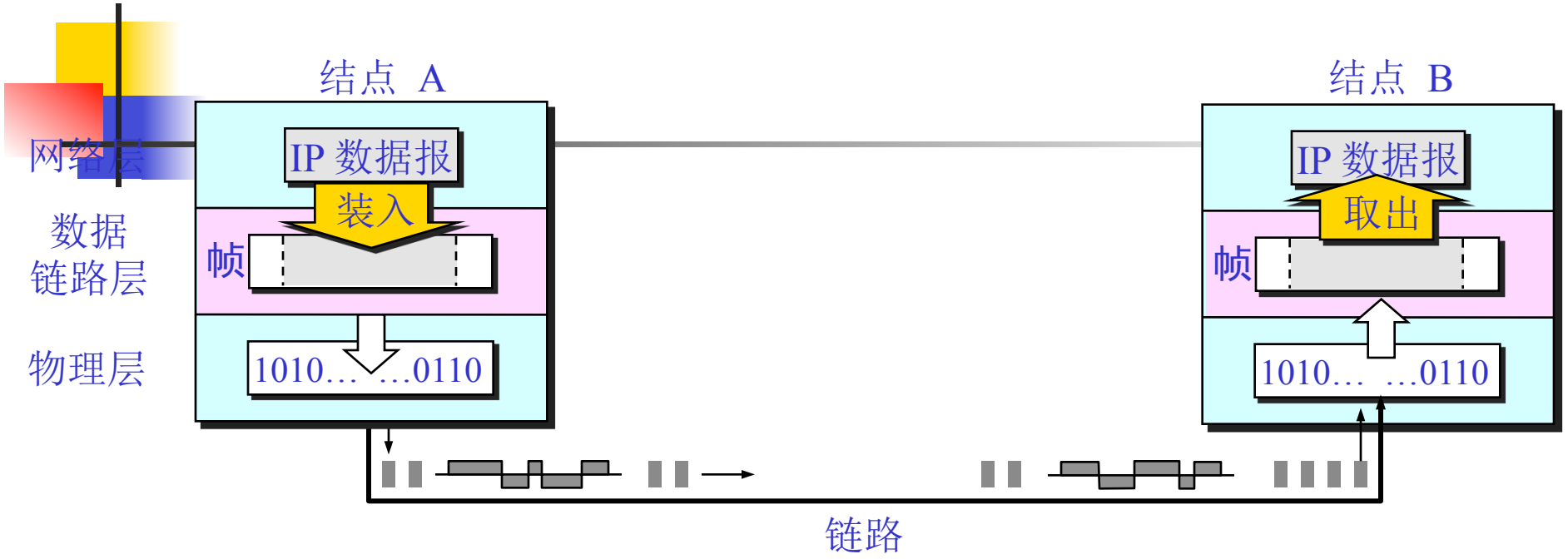
使用点对点信道的数据链路层

- 链路(link)是一条无源的点到点的物理线路段，中间没有任何其他的交换结点。
  - 一条链路只是一条通路的一个组成部分。
- 数据链路(data link) 除了物理线路外，还必须有通信协议来控制这些数据的传输。若把实现这些协议的硬件和软件加到链路上，就构成了数据链路。
  - 现在最常用的方法是使用适配器（即网卡）来实现这些协议的硬件和软件。
  - 一般的适配器都包括了数据链路层和物理层这两层的功能。

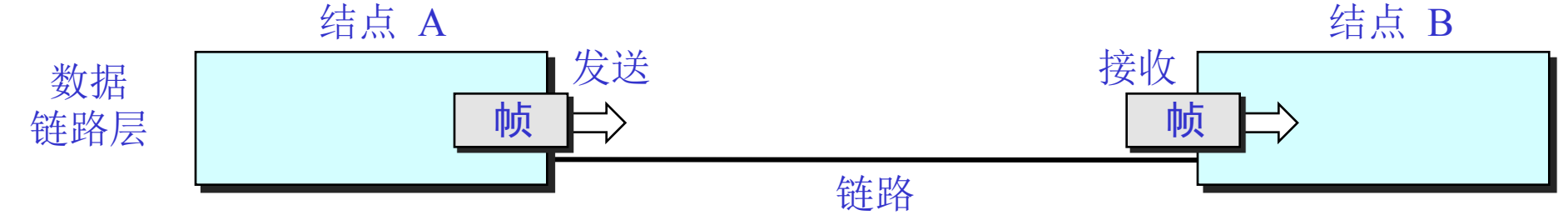
3.1 成帧和差错控制

- 3.1.1 成帧
- 3.1.2 差错控制

数据链路层传送的是帧



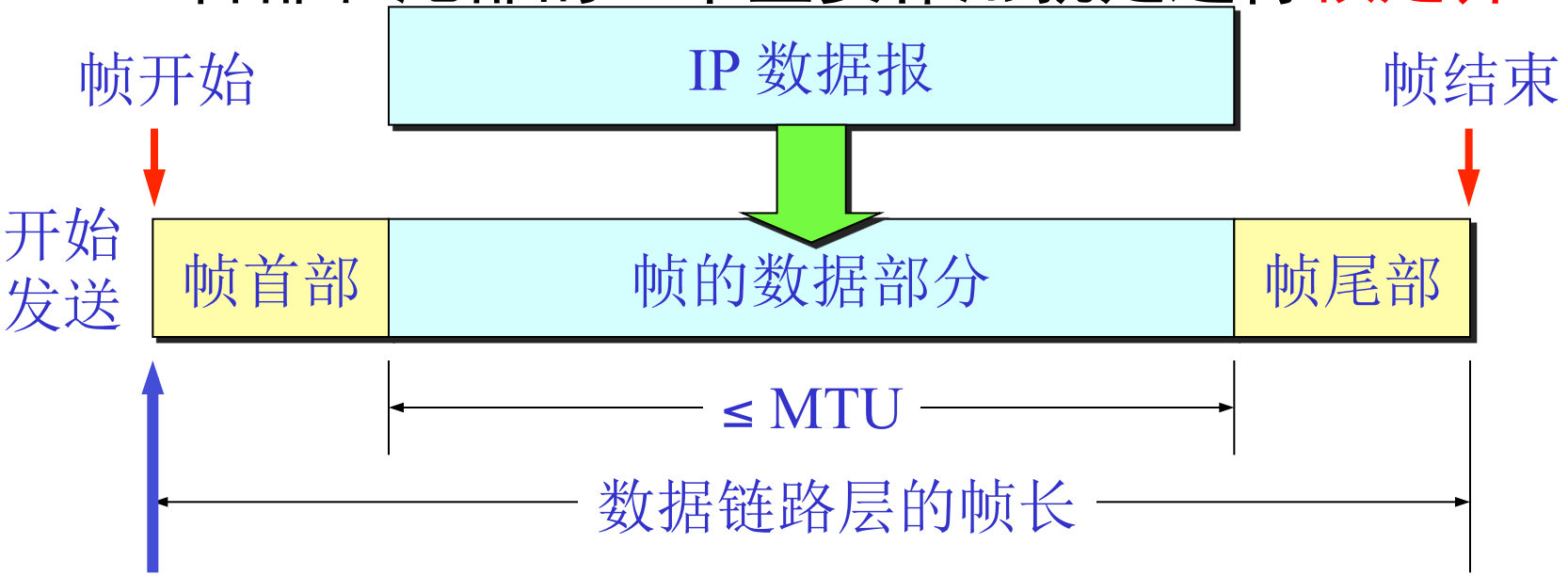
(a)



(b)

1. 封装成帧

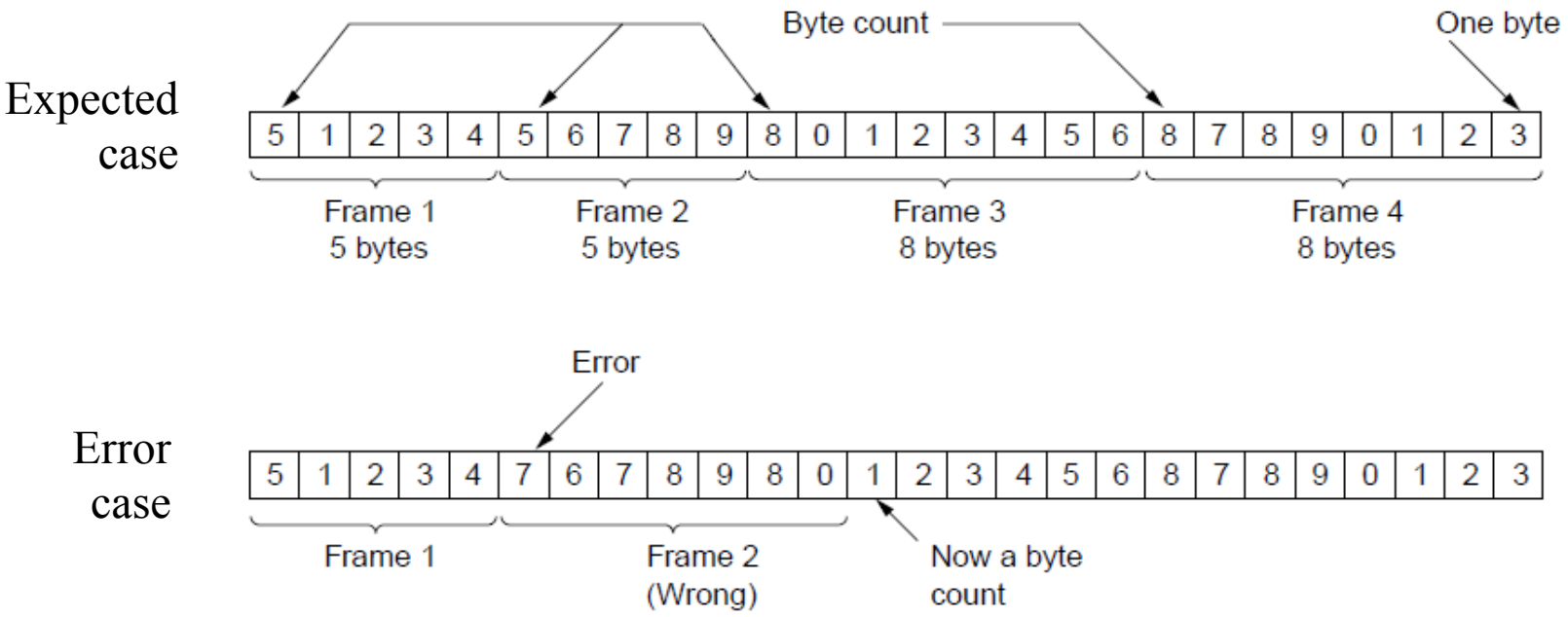
- 封装成帧(framing)就是在一段数据的前后分别添加首部和尾部，然后就构成了一个帧。确定帧的界限。
- 首部和尾部的一个重要作用就是进行帧定界。



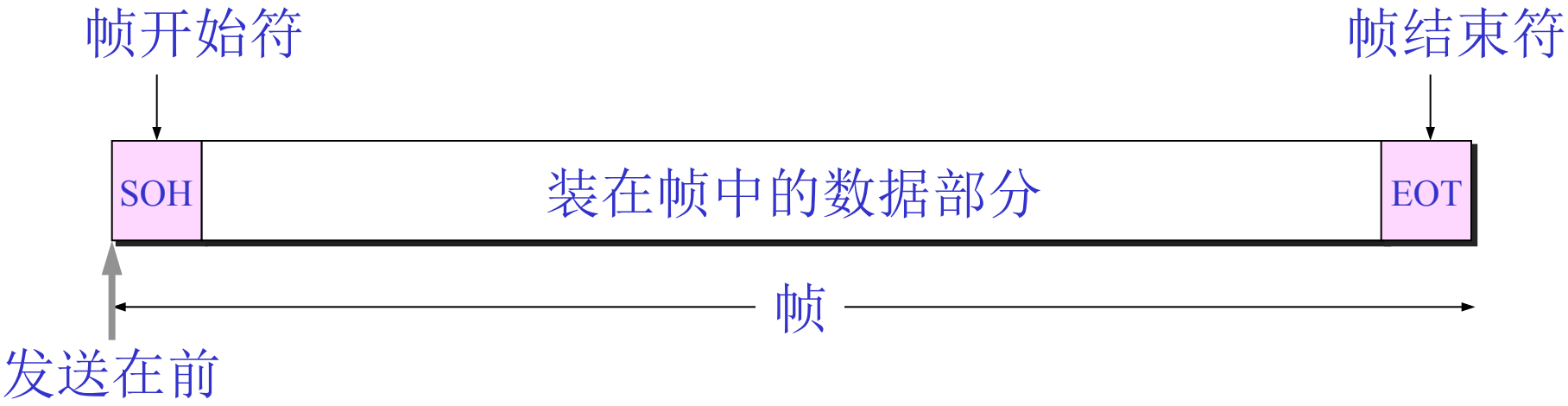
成帧 – 字节计数

帧里有多少字节的数据，就在帧前面添加这个计数值

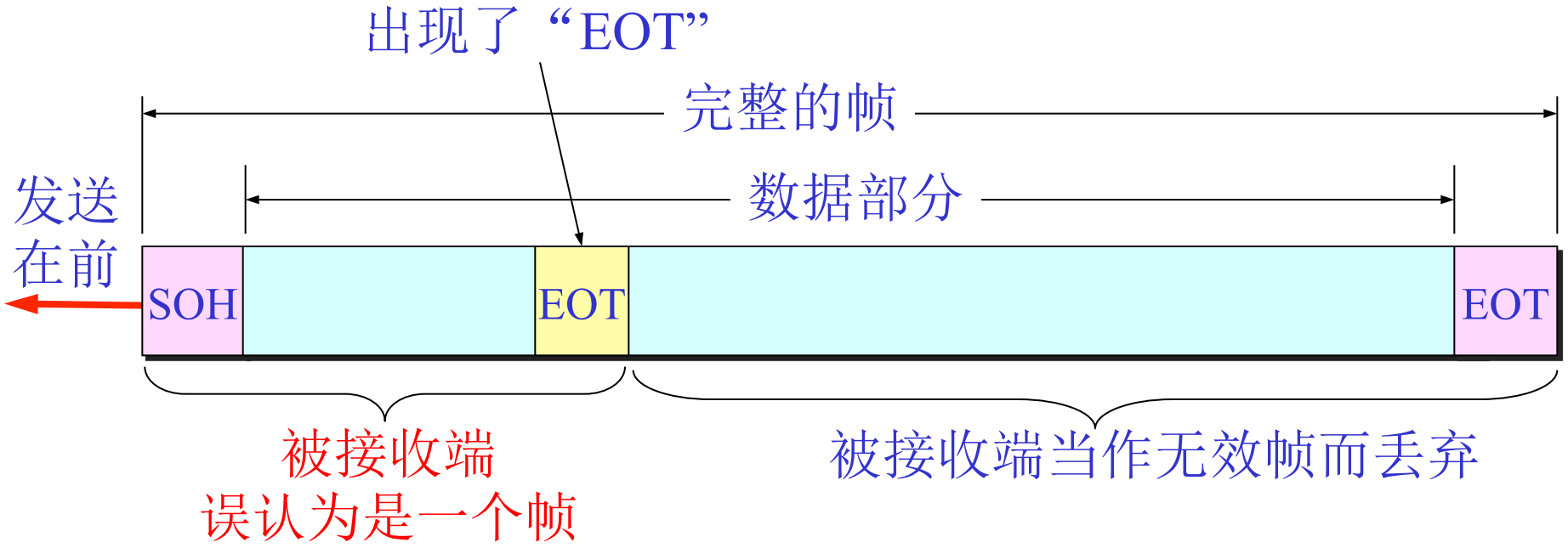
- 简单，但出错了就很难恢复



用控制字符进行帧定界的方法举例



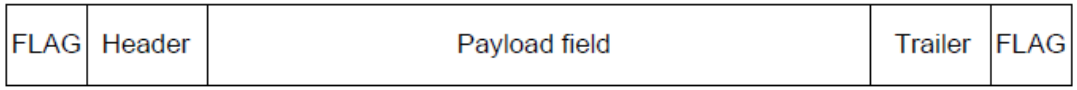
2. 透明传输



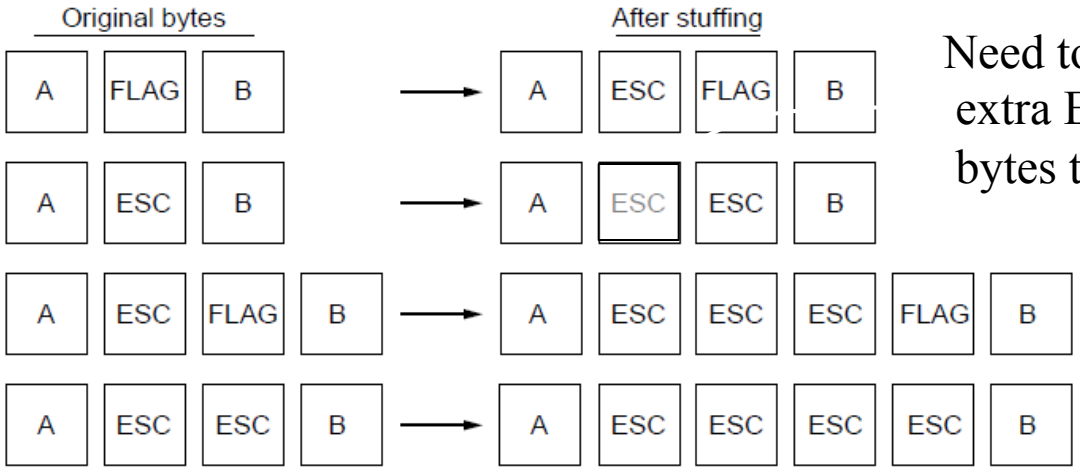
如果数据里也有这个‘flag’怎么来区分：**透明传输**

字节填充解决透明传输

Frame format

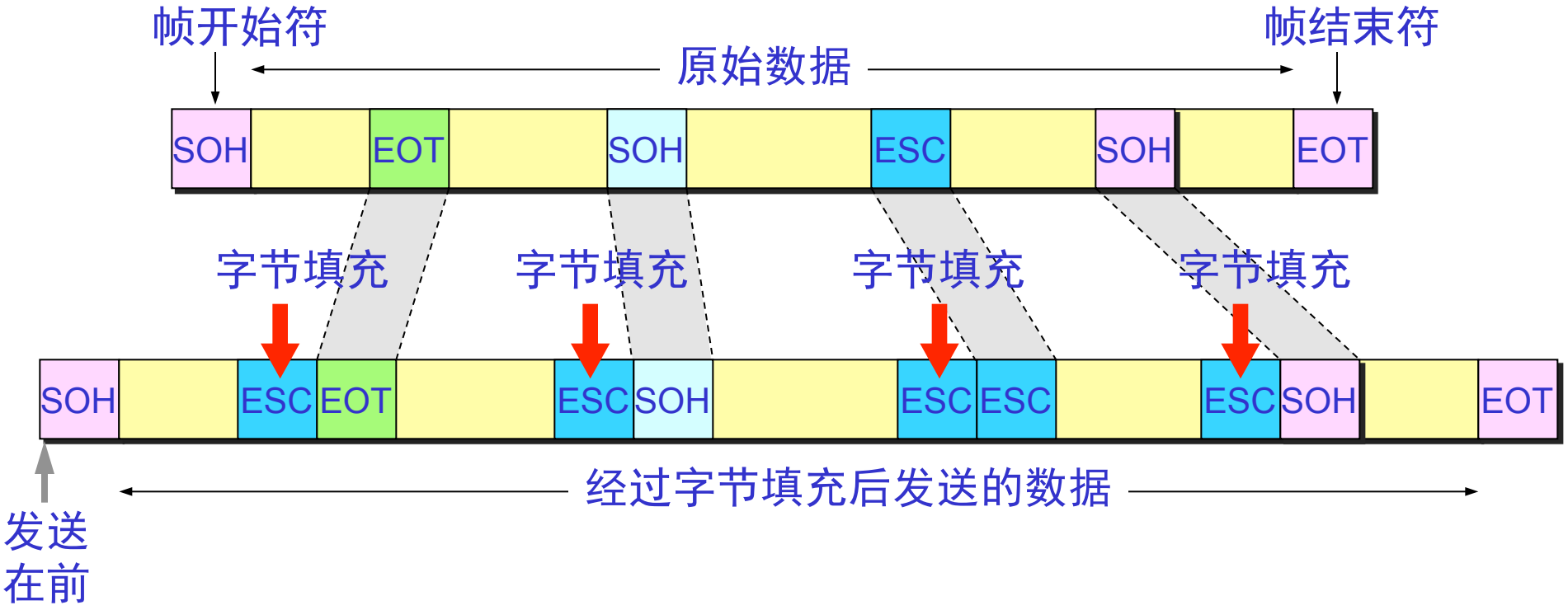


Stuffing examples



Need to escape extra ESCAPE bytes too!

用字节填充法解决透明传输的问题

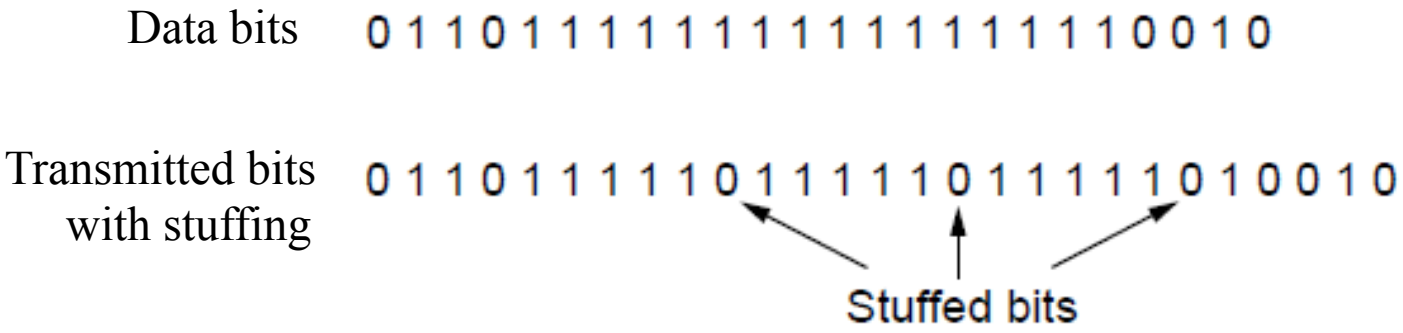


字节填充增加额外的开销，能不能降低开销？降低填充的比特

成帧 – 比特填充

为了将数据里的flag和标志flag区别出来，还可以在比特的层次进行填充:

- Five Frame flag has six consecutive 1s (01111110)
- How about if the data has six consecutive 1:
  - On transmit, after five 1s in the data, a 0 is added
  - On receive, a 0 after 1s is deleted



例子

如原始数据为: 01111110111110111110.
那么经过比特填充后输出数据是什么？假设 flag 是 01111110

答：The output is
011111010111101111100

2. 差错控制

- 在传输过程中可能会产生比特差错：1 可能会变成 0 而 0 也可能变成 1。
- 为了保证数据传输的可靠性，在计算机网络传输数据时，必须采用各种差错检测措施。
  - 错误检测：接收方发现错误
  - 错误纠正：发现错误并纠正

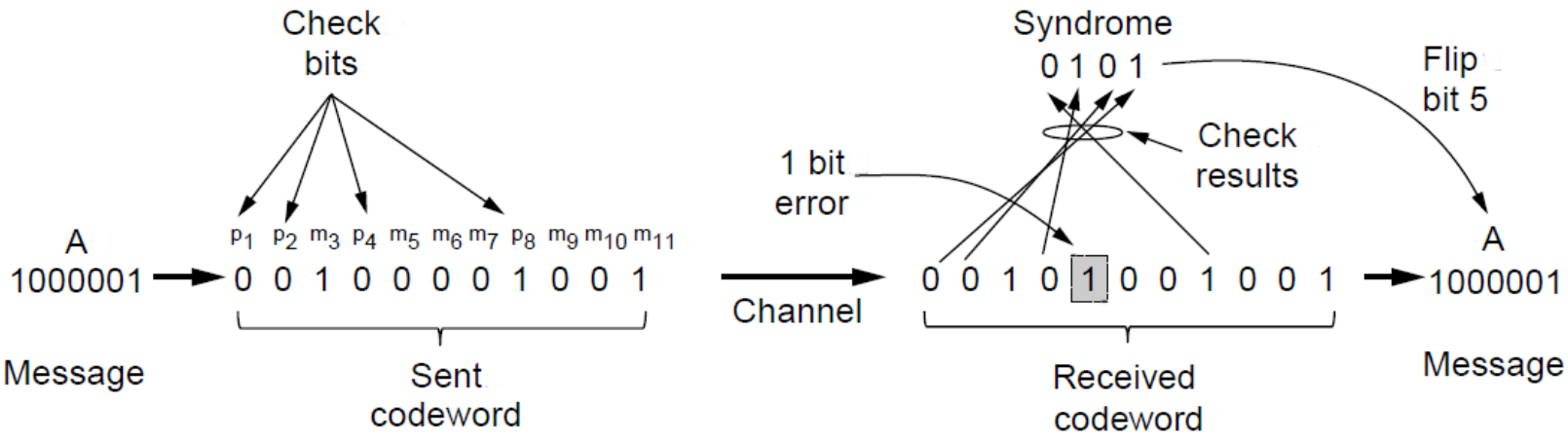
纠错码– Hamming code

他是如何工作的？先来看一个例子

- The original bits are 1000001, with hamming code, the check bits are inserted into the position which are powers of 2:

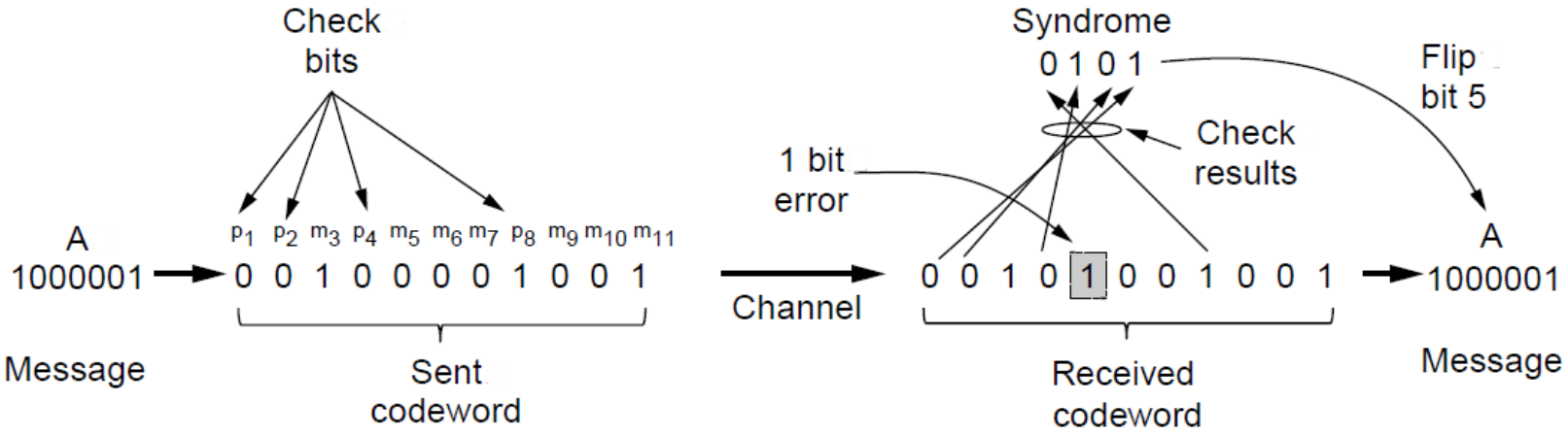
  AB1C000D001

where A=1+3+5+7+9+11; B=2+3+6+7+10+11
C=4+5+6+7; D=8+9+10+11;



继续...

- After the receiver receives the data, it calculate the checksum again, but the result should be inversed, see below
- If the result is 0, no problem, if not, we can deduce the error from the result. For example 0101 means 4+1=5, the bit 5 is error.



(11, 7) Hamming code adds 4 check bits and can correct 1 error

例子

接收方收到一个12位的海明码0xA47。假设最多只有一位错误发生，试问该码的原始值是多少？

Solution: oxA47=1010 0100 0111

A=1+3+5+7+9+11=1;
B=2+3+6+7+10+11=0

C=4+5+6+7+12=0; D=8+9+10+11+12=1;

DCBA=1001=9

So, the original data is 1010 0100 1111=oxA4F

海明距离

问题是前面的例子中，可以纠错几位错误？如有两位错误，可以吗？

海明距离：两个码字中对应位不相同的个数，如10001001、10110001的海明距离为3

对一个编码来说，编码中所有合法字符中两个具有最小海明距离，称为整个编码的海明距离.如10001001、10110001、01110011海明距离为3

当海明距离为n时，任何n-1个错误都不可能把一个合法的字符变成另外一个合法的字符，所以要检测d个错误，需要一个海明距离为d+1的编码方案

当海明距离为n时，任何（n-1）/2个错误后形成的错误码都不可能偏离原来的的码而离另外一个码更近，所有要纠错d个错误，就需要一个海明距离为2d+1的编码方案

错误检测

错误纠正需要添加大量的附加码，如:

- To correct error in the case of 7 bits, Hamming code requires 4 additional bits, so 4/11 bandwidth is useless
- In some case, we need not to correct the error, but to detect the error.

检错码– 奇偶校验 (1)

最简单的检错码是添加一位数据，从而是整个数据所有位的和为奇数或者偶数（1的个数为奇数或者偶数）：

- Ex: 1110000 → 11100001 (偶数)
- Detection checks if the sum is wrong (an error)
- 海明距离为2

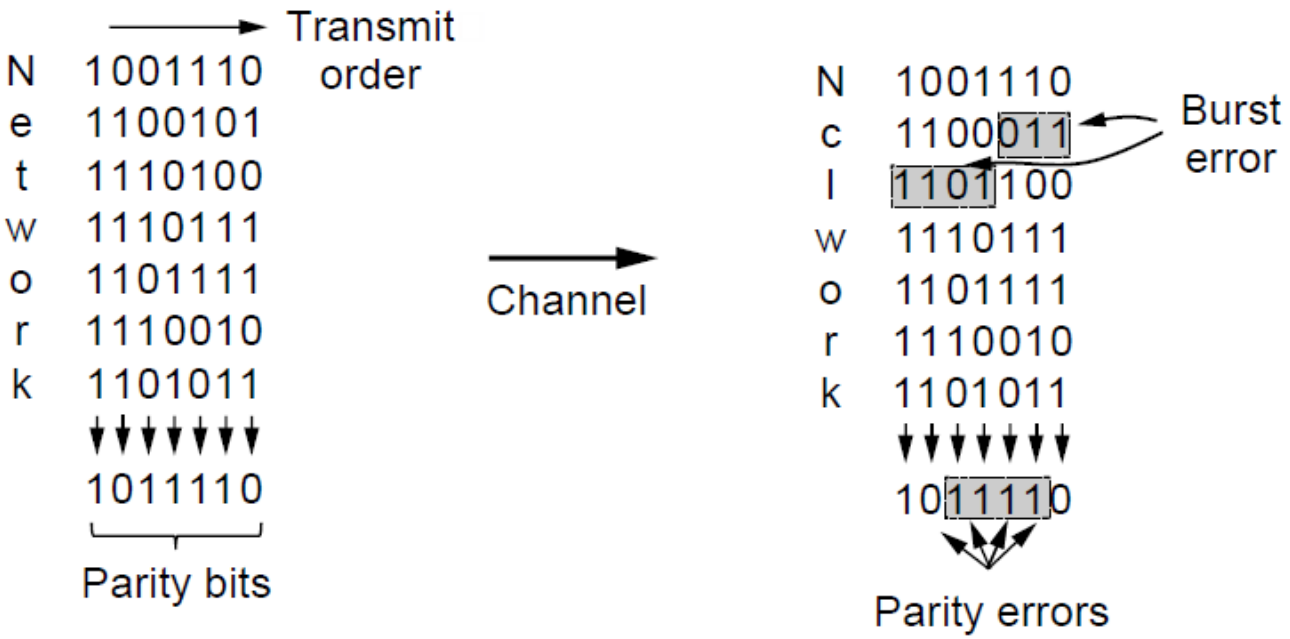
Simple way to detect an *odd* number of errors

- Ex: 1 error, 11100101; detected, sum is wrong
- Ex: 3 errors, 11011001; detected sum is wrong
- Ex: 2 errors, 1110110 ; *not detected*, sum is right!
- Error can also be in the parity bit itself
- Random error is detected with probability ½
- The probability is low, can we improve it?

检错码– 奇偶校验 (2)

交错校验

- Each parity sum is made over column bits
- N burst errors will not cause it to fail
- Of course, one bit error can also be detect.
- For long burst error (>=n+1), the detection probability for a column is ½, so the detection probability for whole block (n column) is $1-(½)^n$
- The additional bandwidth used is 1/8 which is much less than 4/11 in hamming code case



检错码– 校验和

另一个简单的检错方式是将数据做和运算，再将结果作为检错码。当接收方收到数据后也通过和运算来检测错误 (Checksums).

- 如: if we put N bit for checksum, we need to do the modulo 2N sum of the words
- Internet 16位补码校验和

属性:

- Improved error detection over parity bits
- Vulnerable to systematic errors, e.g., added zeros

Internet 补码校验和 (4-bit)

如: 假设数据1001 1100 1010 0011使用4位Internet校验和. 校验码是多少?

4位的补码和运算和模2^4 和运算是一样的，不同的是将高位的溢出加到低位去:

0011 + 1010 = 1101

1101 + 1100 = 1001 + 1 = 1010

1010 + 1001 = 0011 + 1 = 0100.

So, the Internet checksum is 0100.

检错码–CRC (1)

循环冗余检错码(CRC) 的基本思想是通过除法运算来进行检错

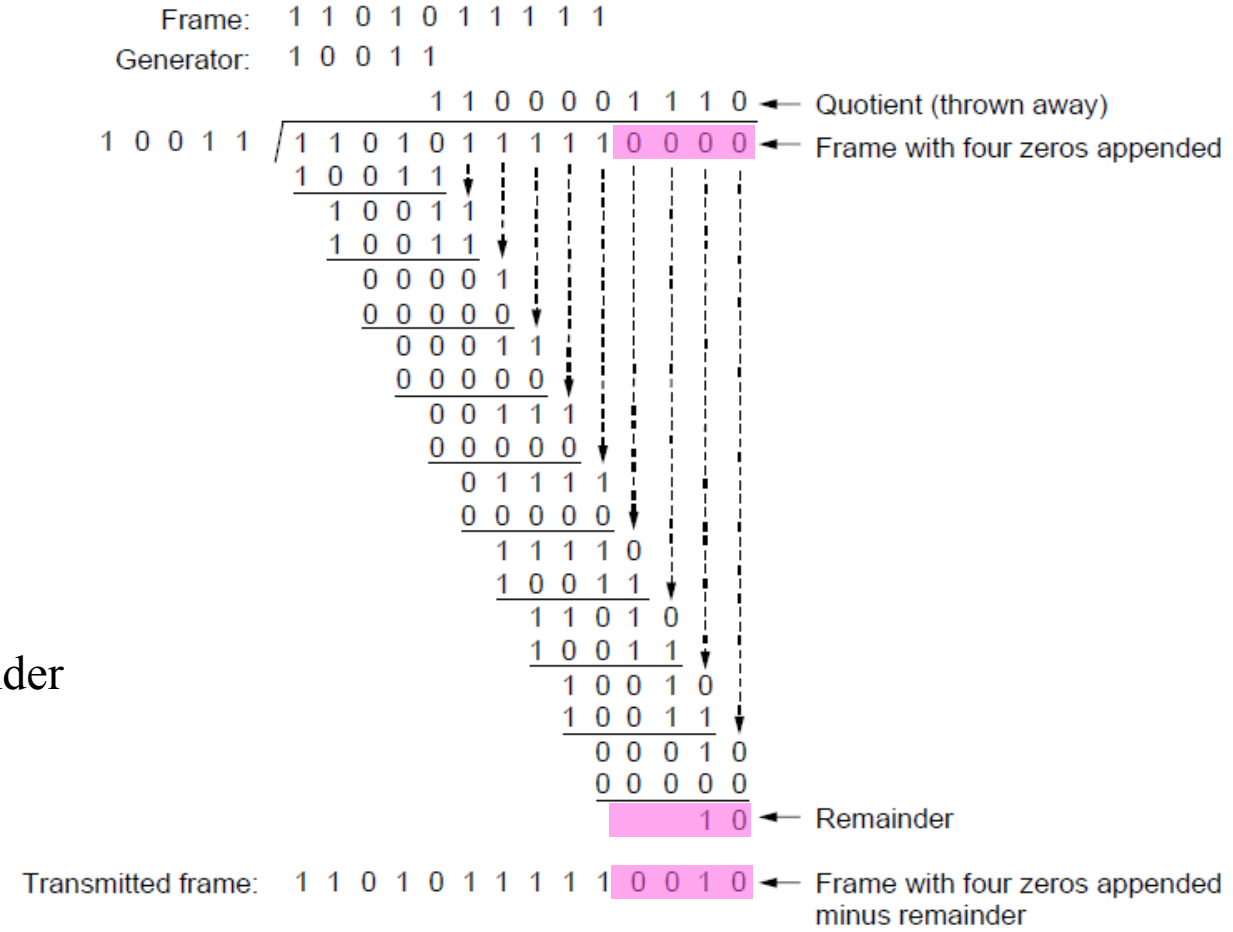
- 双方预先商定一个生成多项式（Generator polynomial，除数），
- 流程
  1. 假设G(x)的为r比特。在数据的低位端加上（r-1）个0。
  2. 生成的数据除以G(x) (模2除)
  3. 得到余数后，将前面生成的数据减去余数(模2减). 结果为最终的数据（包括校验位）

检错码–CRC(2)

- Example:

Start by adding 0s to frame and try dividing

Offset by any reminder to make it evenly divisible



检错码–CRC (3)

- 接收方如何做?
- 接收方将接收到的数据除以G(x),如果能够整除，则认为传输正确，否则，传输错误.
- 比校验和具有更大的检错能力:
  - 能够检测所有的1位、2位、奇数个位错误、长度小于等于r的突发错误
  - Not vulnerable to systematic errors

例子

- CRC传输10011101，G(x)为1001.试问实际传输的比特是什么？假设第三个比特在传输过程中变反了，这个错误能够被检测出来吗？
- 答：G（x）=1001，10011101000 除以1001余数为100.
- 所以实际传输的为：10011101100。
- 第三位错误后为：10111101100，除以1001余数为100，不能整除，所以错误被检测

例子

- 链路层的错误控制有哪两种方式？有以下两种情况：1. 链接并不是可靠的（即经常会发生错误）；2. 链接是可靠的（错误很少发生）。请问哪种方式更适合情况1，哪种方式更适合情况2？
- 答：Error control has two strategies: error detection and error correction.
- Error correction更适合情况1，error detection 更适合情况2
-

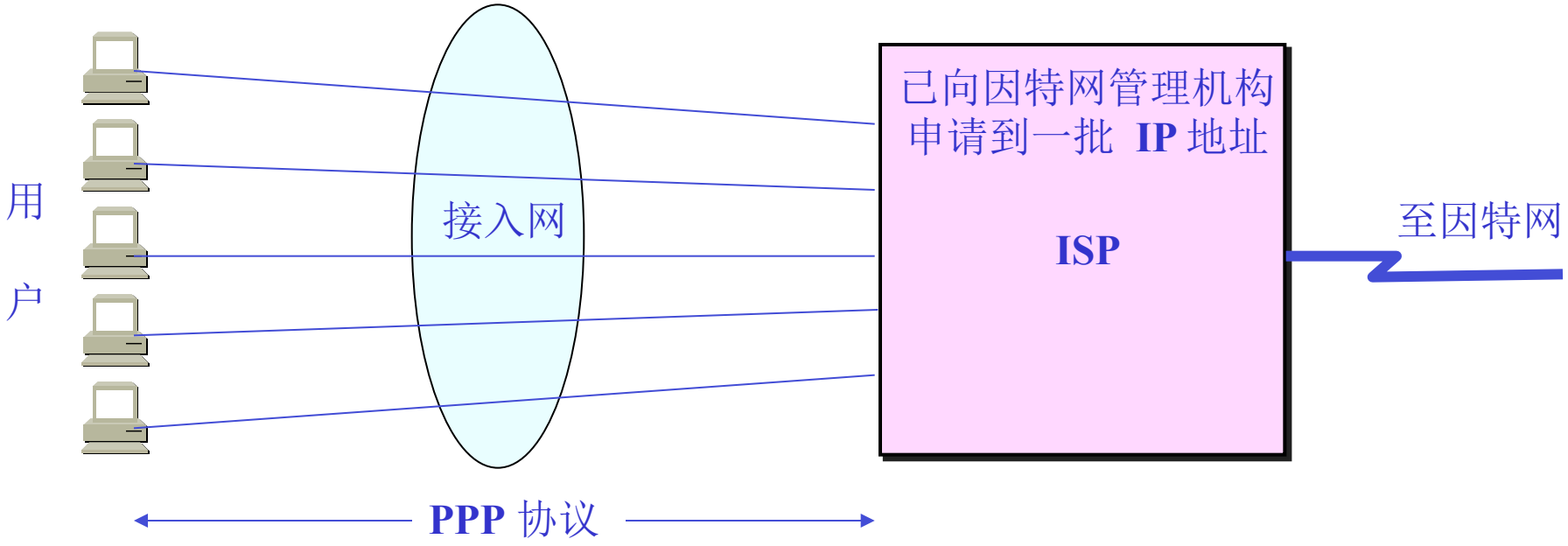
3.2 点对点协议 PPP

- 3.2.1 PPP 协议的特点
- 3.2.2 PPP 协议的帧格式
- 3.2.3 PPP 协议的工作状态

3.2.1 PPP 协议的特点

- 对于点对点的链路，目前使用得最广泛的数据链路层协议是点对点协议 PPP (Point-to-Point Protocol)。
- 用户使用拨号电话线接入互联网时，用户计算机和 ISP 进行通信时所使用的数据链路层协议就是 PPP 协议。
- PPP 协议在1994年就已成为互联网的正式标准。

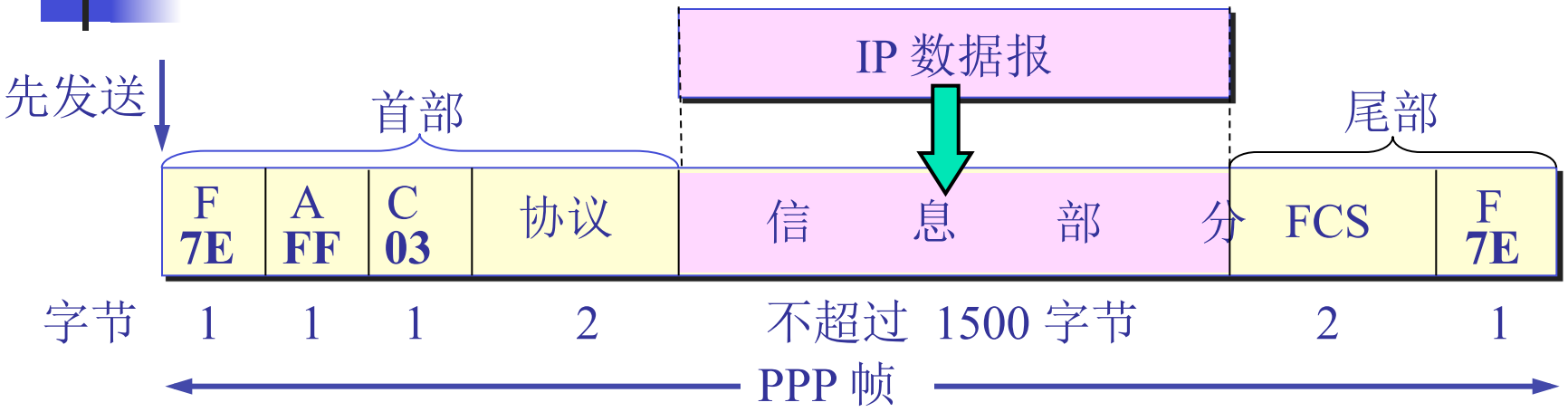
用户到 ISP 的链路使用 PPP 协议



1. PPP 协议主要完成什么功能？

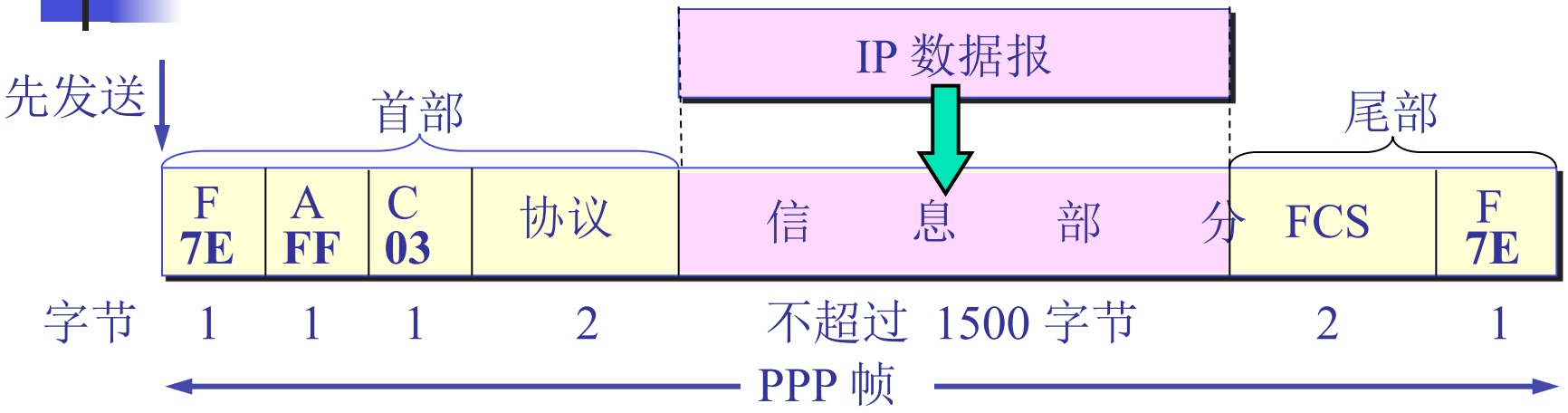
- 成帧
- 链路控制协议（LCP）：启动链路、测试链路、协商参数等
- 网络控制协议（NCP）：针对不同的网络层，都有一个不同的网络控制协议

PPP 协议的帧格式



- 标志字段 F = 0x7E （符号“0x”表示后面的字符是用十六进制表示。十六进制的 7E 的二进制表示是 01111110）。
- 地址字段 A 只置为 0xFF。地址字段实际上并不起作用。
- 控制字段 C 通常置为 0x03 （无编号的帧）。
- PPP 是面向字节的，所有的 PPP 帧的长度都是整数字节。

PPP 协议的帧格式



- PPP 有一个 2 个字节的协议字段。
  - 当协议字段为 0x0021 时，PPP 帧的信息字段就是IP 数据报。
  - 若为 0xC021, 则信息字段是 PPP 链路控制数据。
  - 若为 0x8021，则表示这是网络控制数据。

如何解决透明传输问题：字符填充

- 将信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列 (0x7D, 0x5E)。
- 若信息字段中出现一个 0x7D 的字节，则将其转变成为 2 字节序列 (0x7D, 0x5D)。

好处是什么？

零比特填充

- PPP 协议用在 SONET/SDH 链路时，是使用同步传输（一连串的比特连续传送）。这时 PPP 协议采用零比特填充方法来实现透明传输。
- 在发送端，只要发现有 5 个连续 1，则立即填入一个 0。接收端对帧中的比特流进行扫描。每当发现 5 个连续1时，就把这 5 个连续 1 后的一个 0 删除，

零比特填充

信息字段中出现了和标志字段 F 完全一样的 8 比特组合

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

会被误认为是标志字段 F

发送端在 5 个连 1 之后填入 0 比特再发送出去

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

发送端填入 0 比特

在接收端把 5 个连 1 之后的 0 比特删除

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

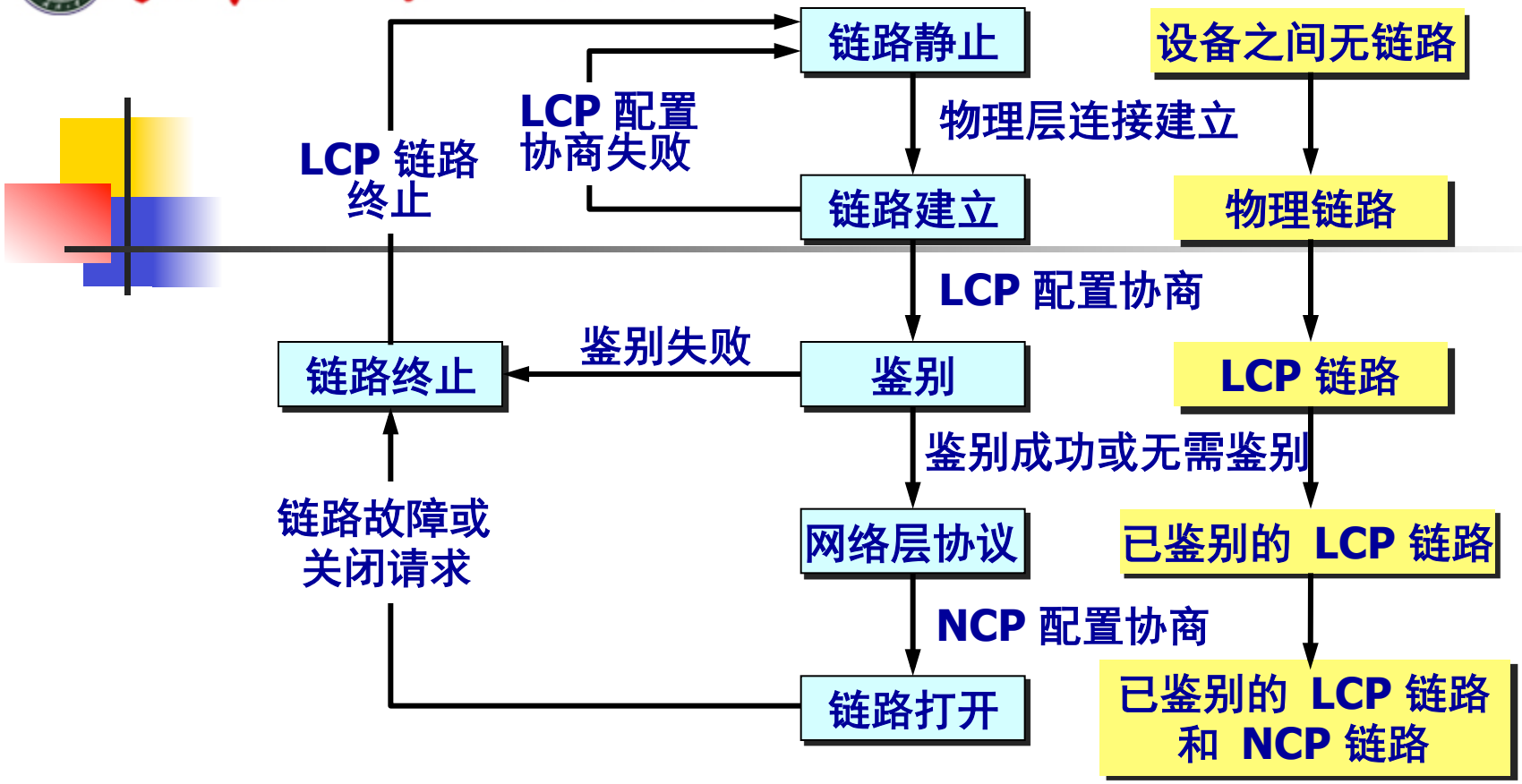
接收端删除填入的 0 比特

不提供使用序号和确认的可靠传输

- PPP 协议之所以不使用序号和确认机制是出于以下的考虑：
  - 在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理。
  - 没有冲突。
  - 帧检验序列 FCS 字段可保证无差错接受。

3.2.3 PPP 协议的工作状态

- 链路初始状态为"链路静止（DEAD）"，当检测到载波信号，并建立一条物理连接，并转移到“链路建立（ESTABLISHMENT）”状态。
- 双方交换LCP信息进行协商（如协议采用几个字节，检错采用几个字节，要不要地址字段和控制字段等）。协商完毕进入"鉴别（AUTHENTICATE）"状态
- 双方进行验证，成功后进入“网络层协议（NETWORK）”状态，并进行网络层配置（通过交换网络层的网络控制信令）。
- 最后进入"链路打开（OPEN）"状态进行数据传输
- 完成后，进入"链路终止（TERMINATE）"状态，回收资源，释放物理层连接。进入"链路静止（DEAD）"

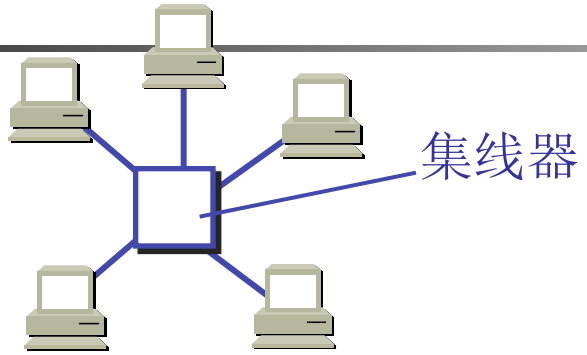


(1) 链路建立阶段：PPP通信双方用链路控制协议交换配置信息（如最大传输单元），一旦配置信息交换成功，链路即宣告建立。
(2) 链路认证阶段。密码验证协议（PAP）和握手鉴权协议(CHAP)。
(3) 网络层控制协议阶段：协商压缩方式，检验IP地址（代替ARP，如检验对方IP是否合法，是否和自己冲突）
(4) 链路终止阶段：： 载波丢失、认证失败、链路质量失败、空闲周期定时器期满或管理员关闭链路等

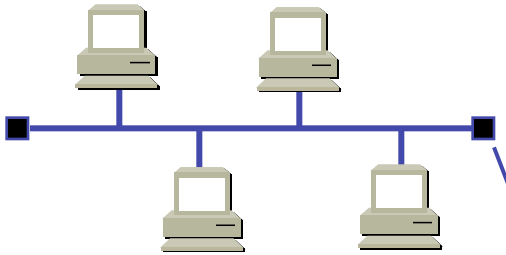
3.3 使用广播信道的数据链路层

- 多个用户共享信道
  - 怎么知道数据发送给谁？
  - 如果很多用户同时发数据的话，会发送冲突了，如何避免冲突？
- 多信道主要是以太网(局域网)采用的，以太网是一个什么样的网络?

局域网的拓扑

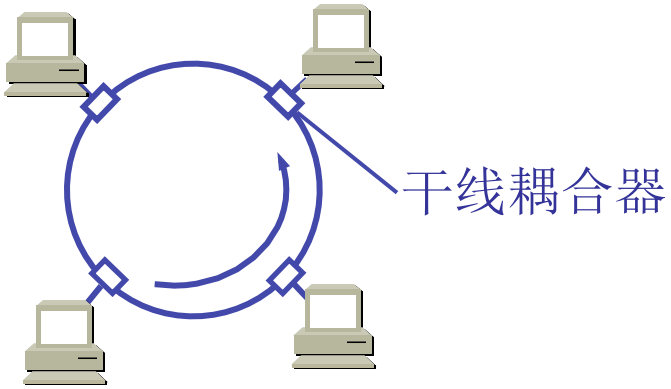


星形网

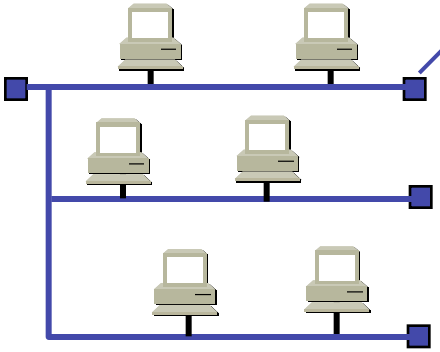


总线网

匹配电阻



环形网

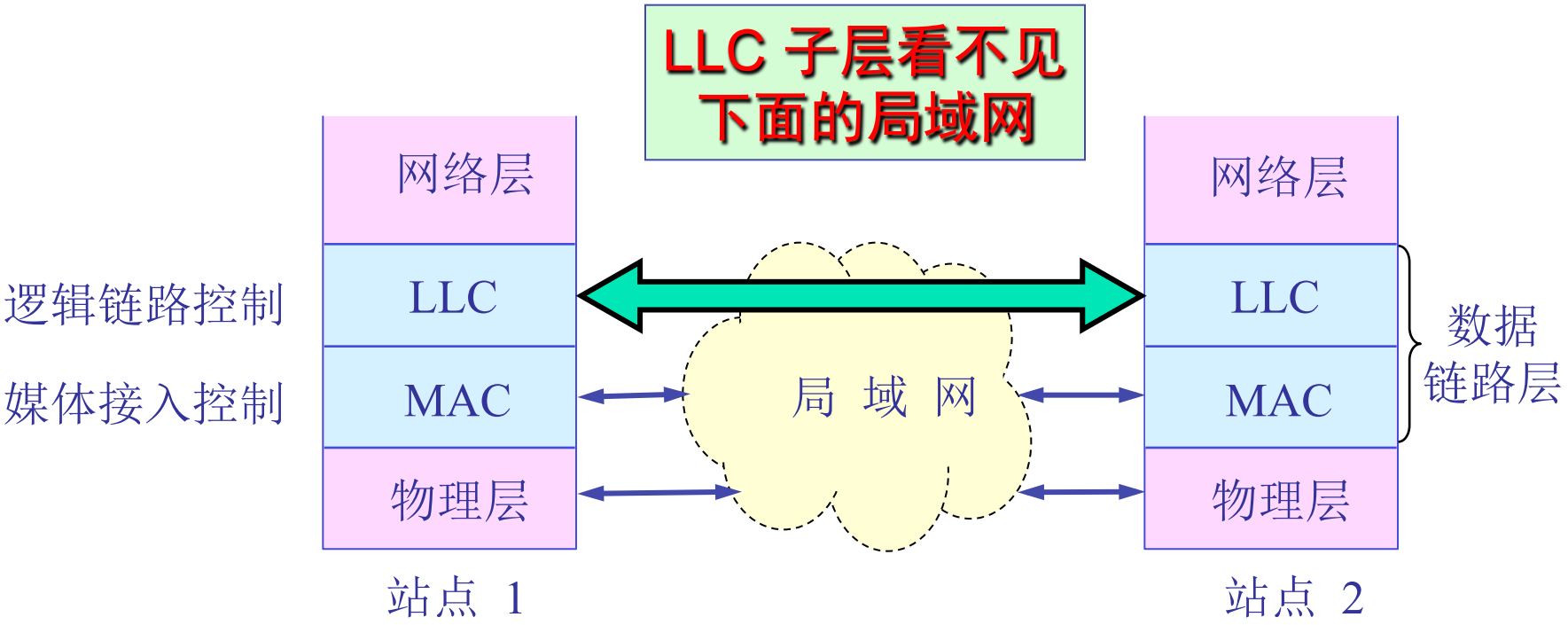


树形网

数据链路层的两个子层

- 为了使数据链路层能更好地适应多种局域网标准，IEEE 802 委员会就将局域网的数据链路层拆成两个子层：
  - 逻辑链路控制 LLC (Logical Link Control)子层；
  - 媒体接入控制 MAC (Medium Access Control)子层。
- 与接入到传输媒体有关的内容都放在 MAC 子层，而 LLC 子层则与传输媒体无关。
- 不管采用何种协议的局域网，对 LLC 子层来说都是透明的。

局域网对 LLC 子层是透明的



以太网（广播信道）的MAC 层

- 协议（地址，帧格式等）
- CSMA/CD

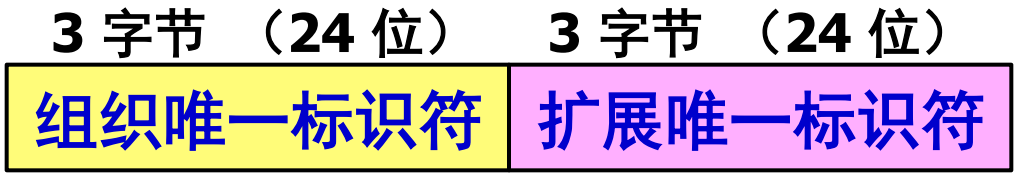
1. MAC 层的硬件地址

- 在局域网中，硬件地址又称为物理地址，或 MAC 地址。

请注意，如果连接在局域网上的主机或路由器安装有多个适配器（网卡），那么这样的主机或路由器就有多个“地址”。更准确些说，这种 48 位“地址”应当是某个接口的标识符。

48 位的 MAC 地址

- IEEE 802 标准规定 MAC 地址字段可采用 6 字节 (48位) 或 2 字节 (16 位) 这两种中的一种。
- IEEE 的注册管理机构 RA 负责向厂家分配地址字段 6 个字节中的前三个字节 (即高位 24 位) ，称为组织唯一标识符。
- 地址字段 6 个字节中的后三个字节 (即低位 24 位) 由厂家自行指派，称为扩展唯一标识符，必须保证生产出的适配器没有重复地址。



48 位的 MAC 地址

48 位的 MAC 地址

- 一个地址块可以生成 2^{24} 个不同的地址。这种 48 位地址称为 MAC-48，它的通用名称是 EUI-48。
- 生产适配器时，6 字节的 MAC 地址已被固化在适配器的 ROM，因此，MAC 地址也叫做硬件地址 (hardware address)或物理地址。
- “MAC地址”实际上就是适配器地址或适配器标识符 EUI-48。

适配器检查 MAC 地址

- 适配器从网络上每收到一个 MAC 帧就首先用硬件检查 MAC 帧中的 MAC 地址.
  - 如果是发往本站的帧则收下，然后再进行其他的处理。
  - 否则就将此帧丢弃，不再进行其他的处理。
- “发往本站的帧”包括以下三种帧：
  - 单播(unicast)帧（一对一）
  - 广播(broadcast)帧（一对全体）
  - 多播(multicast)帧（一对多）

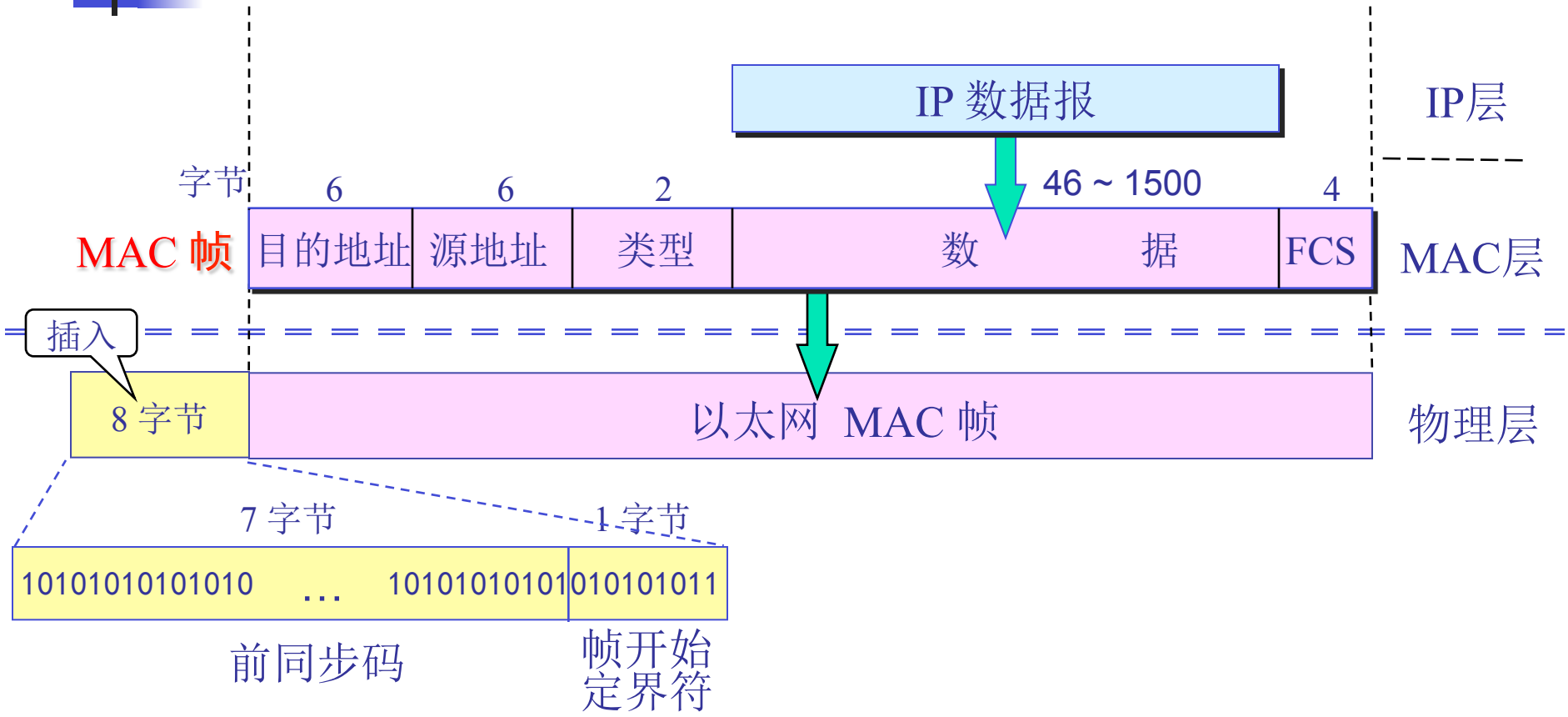
单站地址，组地址，广播地址

- IEEE 规定地址字段的第一字节的最低位为 I/G 位。I/G 表示 Individual / Group。
  - 当 I/G位 = 0 时，地址字段表示一个单站地址。
  - 当 I/G位 = 1 时，表示组地址，用来进行多播（以前曾译为组播）。此时，IEEE 只分配地址字段前三个字节中的 23 位。
  - 当 I/G 位分别为 0 和 1 时，一个地址块可分别生成 2^{24} 个单个站地址和 2^{24} 个组地址。
  - 所有 48 位都为 1 时，为广播地址。只能作为目的地址使用。

全球管理与本地管理

- IEEE 把地址字段第一字节的最低第 2 位规定为 G/L 位，表示 Global / Local。
- 当 G/L位 = 0 时，是全球管理（保证在全球没有相同的地址），厂商向IEEE购买的 OUI 都属于全球管理。
- 当 G/L位 = 1 时， 是本地管理，这时用户可任意分配网络上的地址。

以太网的 MAC 帧格式


IP 数据报
IP层
字节
6
6
2
46 ~ 1500
4
MAC 帧
目的地址
源地址
类型
数
据
FCS
MAC层
插入
8 字节
以太网 MAC 帧
物理层
7 字节
1 字节
10101010101010 ... 10101010101010101011
前同步码
帧开始
定界符

例子

下图为某一数据包，其中加黑部分为以太网头，请分别指出源地址、目的地址、类型

```
0000   00 06 82 00 39 e4 08 18  1a 78 f2 2a 81 00 00 01
0010   08 00 45 00 00 c8 83 8a  40 00 80 11 07 71 0a 01
0020   2d 0d 0a 01 2e 1b 0f a8  b8 0c 00 b4 05 af 80 00
0030   c9 e6 80 50 36 00 61 98  75 29 66 76 f0 7f 69 75
0040   e5 de ea 78 6f 7e f6 f7  7d 75 fe ee ef 7b 76 f3
```

```
▼ Ethernet II, Src: Zte_78:f2:2a (08:18:1a:78:f2:2a), Dst: Convedia_00:39:e4 (00:06:82:00:39:e4)
  ▶ Destination: Convedia_00:39:e4 (00:06:82:00:39:e4)
  ▶ Source: Zte_78:f2:2a (08:18:1a:78:f2:2a)
    Type: 802.1Q Virtual LAN (0x8100)
```

无效的 MAC 帧

- 数据字段的长度与长度字段的值不一致；
- 帧的长度不是整数个字节；
- 用收到的帧检验序列 FCS 查出有差错；
- 数据字段的长度不在 64 ~ 1500 字节之间。
  - 64为最小长度（参考后面）
  - 1500为最大长度

对于检查出的无效 MAC 帧就简单地丢弃。以太网不负责重传丢弃的帧。

IEEE 802.3 MAC 帧格式

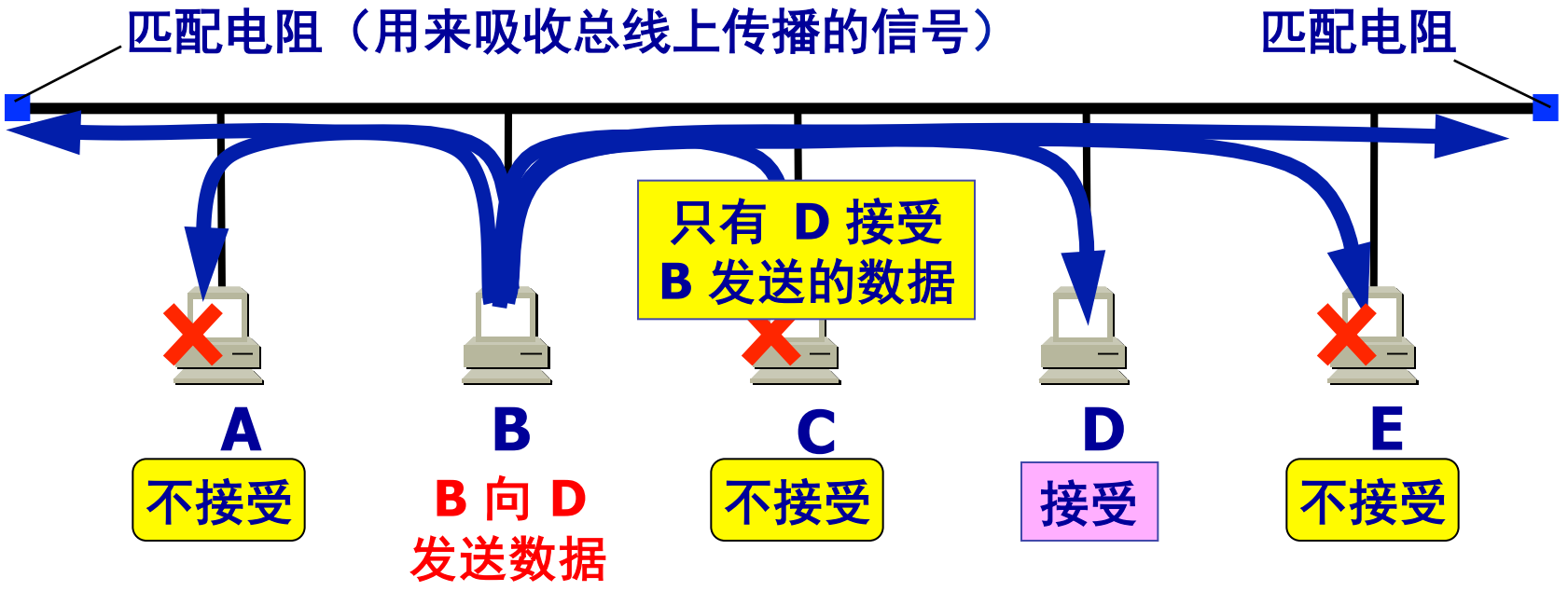
与以太网V2 MAC 帧格式相似，区别在于：

- (1) IEEE 802.3 规定的 MAC 帧的第三个字段是“长度 / 类型”。
  - 当这个字段值大于 0x0600 时（相当于十进制的 1536），就表示“类型”。这样的帧和以太网 V2 MAC 帧完全一样。
  - 当这个字段值小于 0x0600 时才表示“长度”。

现在市场上流行的都是以太网V2 的 MAC 帧，但大家也常常把它称为 IEEE 802.3 标准的 MAC 帧。

以太网的广播方式发送

- 最初的以太网是将许多计算机都连接到一根总线上。当初认为这样的连接方法既简单又可靠，因为总线上没有有源器件。



以太网的广播方式发送

- 总线上的每一个工作的计算机都能检测到 B 发送的数据信号。
- 由于只有计算机 D 的地址与数据帧首部写入的地址一致，因此只有 D 才接收这个数据帧。
- 其他所有的计算机（A, C 和 E）都检测到不是发送给它们的数据帧，因此就丢弃这个数据帧而不能够收下来。
- 具有广播特性的总线上实现了一对一的通信。

为了通信的简便
以太网采取了两种重要的措施

- 采用较为灵活的无连接的工作方式，即不必先建立连接就可以直接发送数据。
- 以太网对发送的数据帧不进行编号，也不要求对方发回确认。
  - 这样做的理由是局域网信道的质量很好，因信道质量产生差错的概率是很小的。

CSMA/CD协议

- CSMA/CD 含义：载波监听多点接入 / 碰撞检测 (Carrier Sense Multiple Access with Collision Detection) 。
- “多点接入”表示许多计算机以多点接入的方式连接在一根总线上。
- “载波监听”是指每一个站在发送数据之前先要检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不要发送数据，以免发生碰撞。

碰撞检测（CD）

- “碰撞检测”就是计算机边发送数据边检测信道上的信号电压大小。
- 当几个站同时在总线上发送数据时，总线上的信号电压摆动值将会增大（互相叠加）。
- 当一个站检测到的信号电压摆动值超过一定的门限值时，就认为总线上至少有两个站同时在发送数据，表明产生了碰撞。
- 所谓“碰撞”就是发生了冲突。因此“碰撞检测”也称为“冲突检测”。

碰撞检测（CD）

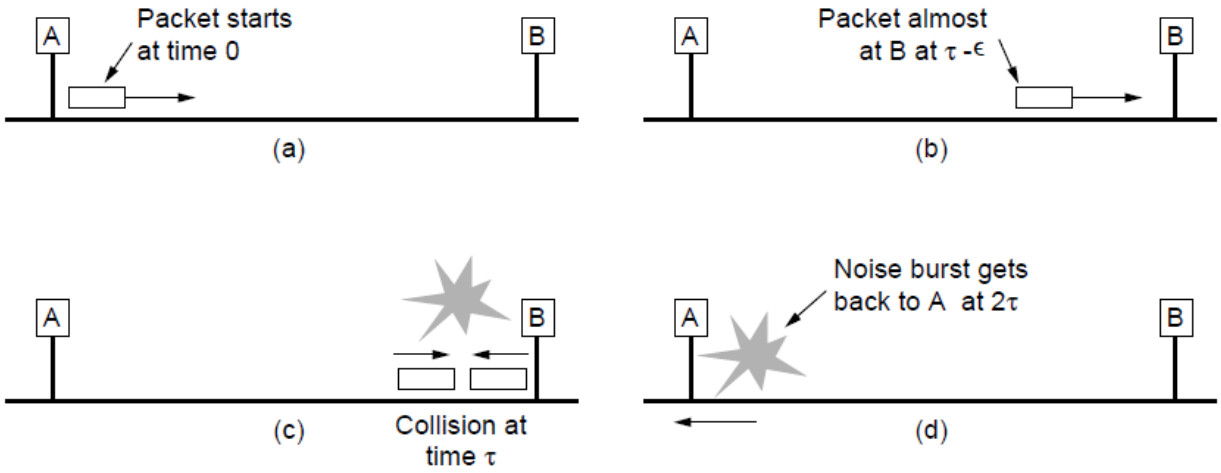
- 发送端同监听信号来监测冲突
  - 监听多长时间?
- 数据的发送时间都需要监听
  - 数据的发送时间由带宽和数据长度决定
  - 以太网的带宽固定
- 为了能够监测到冲突，对数据的长度由要求
  - 这就是为什么数据的最小长度为64的原因

CSMA/CD

为什么至少要64bytes？帧必须要64bytes，才有可能完成Collisions Detection:

- 以太网的长度最长为2500m（规定），
- τ 是信号传播时间, 对于2500m的以太网，τ 大概是25usec.
- 为了检测到冲突, 一个站需要25*2=50usec.
- 对于经典以太网(10Mbps), 50usec*10Mbps=500bits, we let it to be 512bits, i.e. 64bytes

Note: contention slot is equal to 2τ=51.2us



争用期

- 最先发送数据帧的站，在发送数据帧后至多经过时间 2τ（两倍的端到端往返时延）就可知道发送的数据帧是否遭受了碰撞。
- 以太网的端到端往返时延 2τ 称为争用期，或碰撞窗口。
- 经过争用期这段时间还没有检测到碰撞，才能肯定这次发送不会发生碰撞。
- 发送数据的两种方式
  - 以太网在发送数据时，若前 64 字节没有发生冲突，则后续的数据就不会发生冲
  - 发送端需要发送一个64的数据来"抓住"信道

CSMA/CD

例子:

1000m 长以太网**,** 带宽为**10mbps**，使用 **CSMA/CD,** 信号传播速度**200m/usec.** 帧的最小长度为多少**?**

解:

round-trip propagation time=10usec

So the minimum length=10usec*10mbps=100bit

假设帧长为**256bits,** 包括**32**位开销（头**, checksum** 和其他开销）**.** 并假设站会用一个争用期时间来“抓住”信道在发送数据或者**ACK**之前，**ACK**为**32bits.** 那么实际的有效数据发送率为多少？

Solution: 1. Transmitter seizes cable (10 μsec)
2. Transmit data (25.6 μsec)
3. Delay for last bit to get to the end (5.0 μsec) 4. Receiver seizes cable (10 μsec)
5. Acknowledgement sent (3.2 μsec)
6. Delay for last bit to get to the end (5.0 μsec)

The sum is 58.8usec, in this period, 224 data bits are sent, so effective data rate=224/58.8=3.8Mbps

二进制指数类型退避算法 (truncated binary exponential type)

- 发生碰撞的站在停止发送数据后，要推迟（退避）一个随机时间才能再发送数据。
  - 基本退避时间取为争用期 2τ。
  - 从整数集合 [0, 1, … , $(2^k - 1)$] 中随机地取出一个数，记为 r。重传所需的时延就是 r 倍的基本退避时间。
  - 参数 k 按下面的公式计算：

    k = Min[重传次数, 10]

  - 当 $k \le 10$ 时，参数 k 等于重传次数。
  - 当重传达 16 次仍不能成功时即丢弃该帧，并向高层报告。

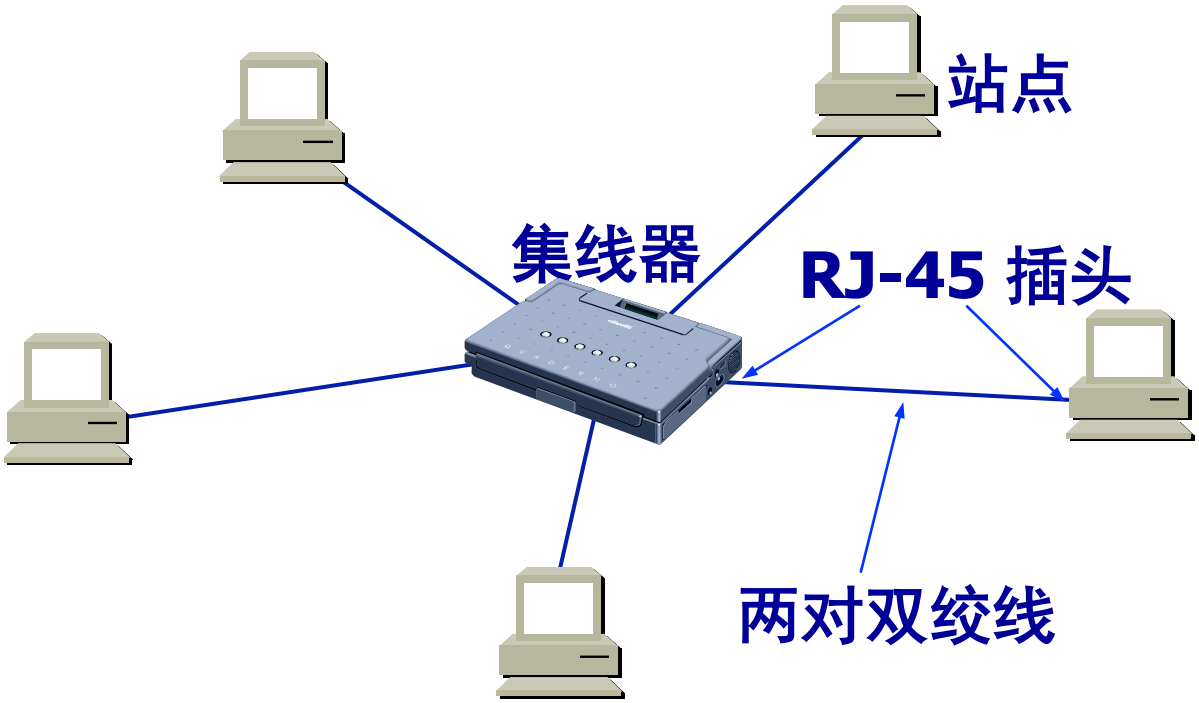
CSMA/CD协议的要点

- (1) 准备发送。但在发送之前，必须先检测信道。
- (2) 检测信道。若检测到信道忙，则应不停地检测，一直等待信道转为空闲。若检测到信道空闲，并在 96 比特时间内信道保持空闲（保证了帧间最小间隔），就发送这个帧。
- (3) 检查碰撞。在发送过程中仍不停地检测信道，即网络适配器要边发送边监听。这里只有两种可能性：
  - ①发送成功：在争用期内一直未检测到碰撞。这个帧肯定能够发送成功。发送完毕后，其他什么也不做。然后回到 (1)。
  - ②发送失败：在争用期内检测到碰撞。这时立即停止发送数据，并按规定发送人为干扰信号。适配器接着就执行指数退避算法，等待 *r* 倍 512 比特时间后，返回到步骤 (2)，继续检测信道。但若重传达 16 次仍不能成功，则停止重传而向上报错。

3.3.3 使用集线器的星形拓扑

- 传统以太网最初是使用粗同轴电缆，后来演进到使用比较便宜的细同轴电缆，最后发展为使用更便宜和更灵活的双绞线。
- 采用双绞线的以太网采用星形拓扑，在星形的中心则增加了一种可靠性非常高的设备，叫做集线器 (hub)。

使用集线器的双绞线以太网



集线器的一些特点

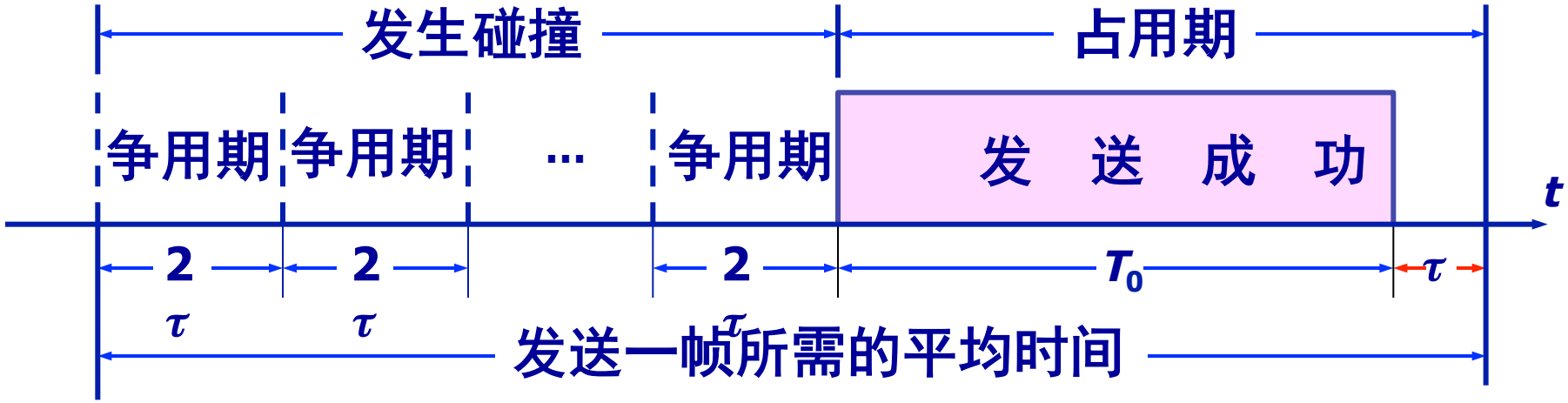
- (1) 集线器是使用电子器件来模拟实际电缆线的工作，因此整个系统仍然像一个传统的以太网那样运行。
- (2) 使用集线器的以太网在逻辑上仍是一个总线网，各工作站使用的还是 CSMA/CD 协议，并共享逻辑上的总线。
- (3) 集线器很像一个多接口的转发器，工作在物理层。
- (4) 集线器采用了专门的芯片，进行自适应串音回波抵消，减少了近端串音。

3.3.4 以太网的信道利用率

- 多个站在以太网上同时工作就可能会发生碰撞。
- 当发生碰撞时，信道资源实际上是被浪费了。因此，当扣除碰撞所造成的信道损失后，以太网总的信道利用率并不能达到 100%。
- 假设 τ 是以太网单程端到端传播时延。则争用期长度为 2τ，即端到端传播时延的两倍。检测到碰撞后不发送干扰信号。
- 设帧长为 L (bit)，数据发送速率为 C (bit/s)，则帧的发送时间为 $T_0 = L/C$ (s)。

以太网信道被占用的情况

- 一个站在发送帧时出现了碰撞。经过一个争用期 2τ 后，可能又出现了碰撞。这样经过若干个争用期后，一个站发送成功了。假定发送帧需要的时间是 T_0。



以太网信道被占用的情况

- 注意到，成功发送一个帧需要占用信道的时间是 $T_0 + \tau$，比这个帧的发送时间要多一个单程端到端时延τ。
- 这是因为当一个站发送完最后一个比特时，这个比特还要在以太网上传播。
- 在最极端的情况下，发送站在传输媒体的一端，而比特在媒体上传输到另一端所需的时间是 τ。

参数 a 与利用率

- 要提高以太网的信道利用率，就必须减小 τ 与 T_0 之比。
- 在以太网中定义了参数 a，它是以太网单程端到端时延 τ 与帧的发送时间 T_0 之比：

$$a = \tau / T_0$$

- **$\alpha \to 0$，表示一发生碰撞就立即可以检测出来， 并立即停止发送，因而信道利用率很高。**
- **α 越大，表明争用期所占的比例增大，每发生一次碰撞就浪费许多信道资源，使得信道利用率明显降低。**

对以太网参数 a 的要求

- 为提高利用率，以太网的参数a的值应当尽可能小些。
- 对以太网参数 a 的要求是：
  - 当数据率一定时，以太网的连线的长度受到限制，否则 τ 的数值会太大。
  - 以太网的帧长不能太短，否则 T_0 的值会太小，使 a 值太大。

信道利用率的最大值 S_{max}

- 在理想化的情况下，以太网上的各站发送数据都不会产生碰撞（这显然已经不是 CSMA/CD，而是需要使用一种特殊的调度方法），即总线一旦空闲就有某一个站立即发送数据。
- 发送一帧占用线路的时间是 $T_0 + \tau$，而帧本身的发送时间是 T_0。于是我们可计算出理想情况下的极限信道利用率 S_{max} 为：

$$S_{max} = \frac{T_0}{T_0 + \tau} = \frac{1}{1 + a}$$

- **只有当参数 *a* 远小于 1 才能得到尽可能高的极限信道利用率。**
- **据统计，当以太网的利用率达到 30%时就已经处于重载的情况。很多的网络容量被网上的碰撞消耗掉了。**

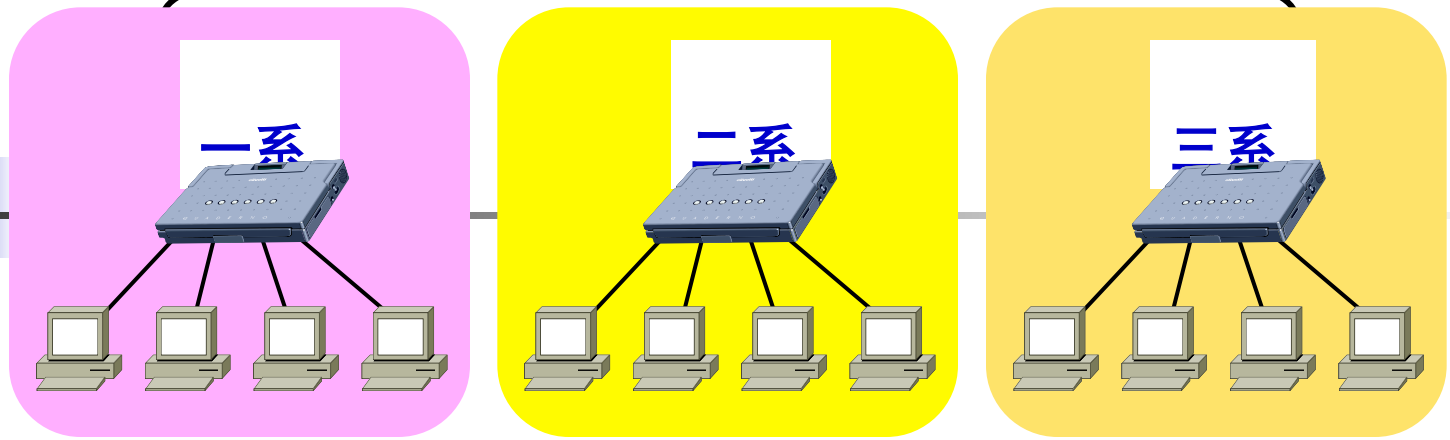
3.4 扩展的以太网

- 3.4.1 在物理层扩展以太网
- 3.4.2 在数据链路层扩展以太网
- 3.4.3 虚拟局域网

3.4.1 在物理层扩展以太网

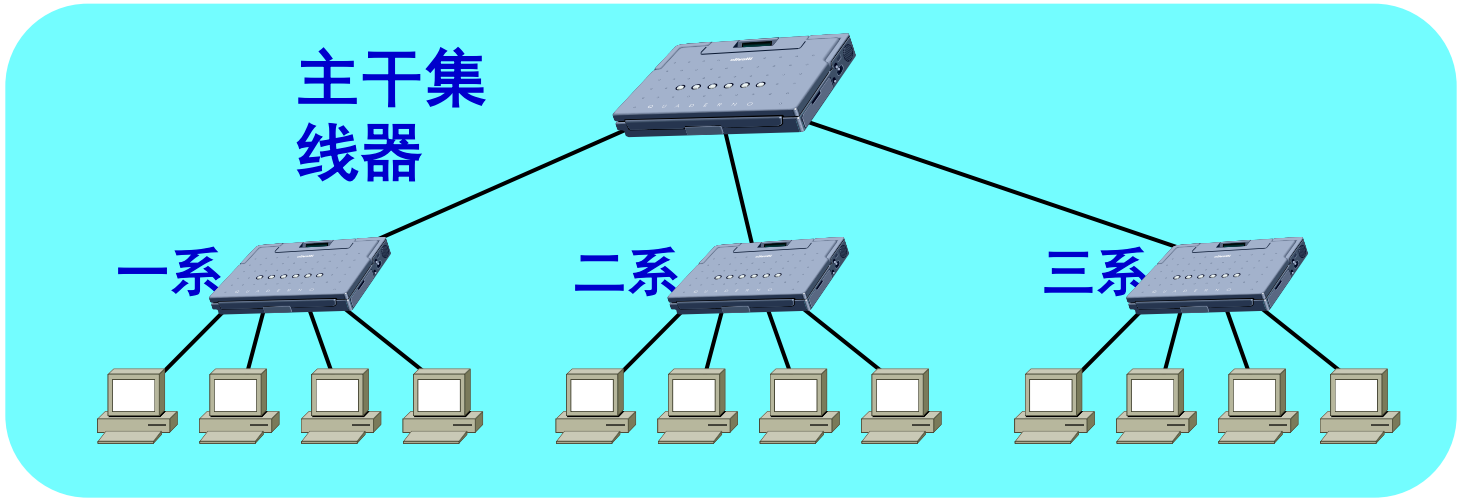
- 使用集线器扩展
  - 使用多个集线器可连成更大的、多级星形结构的以太网。
  - 例如，一个学院的三个系各有一个 10BASE-T 以太网，可通过一个主干集线器把各系的以太网连接起来，成为一个更大的以太网。

三个独立的碰撞域



三个独立的以太网

一个更大的碰撞域



一个扩展的以太网

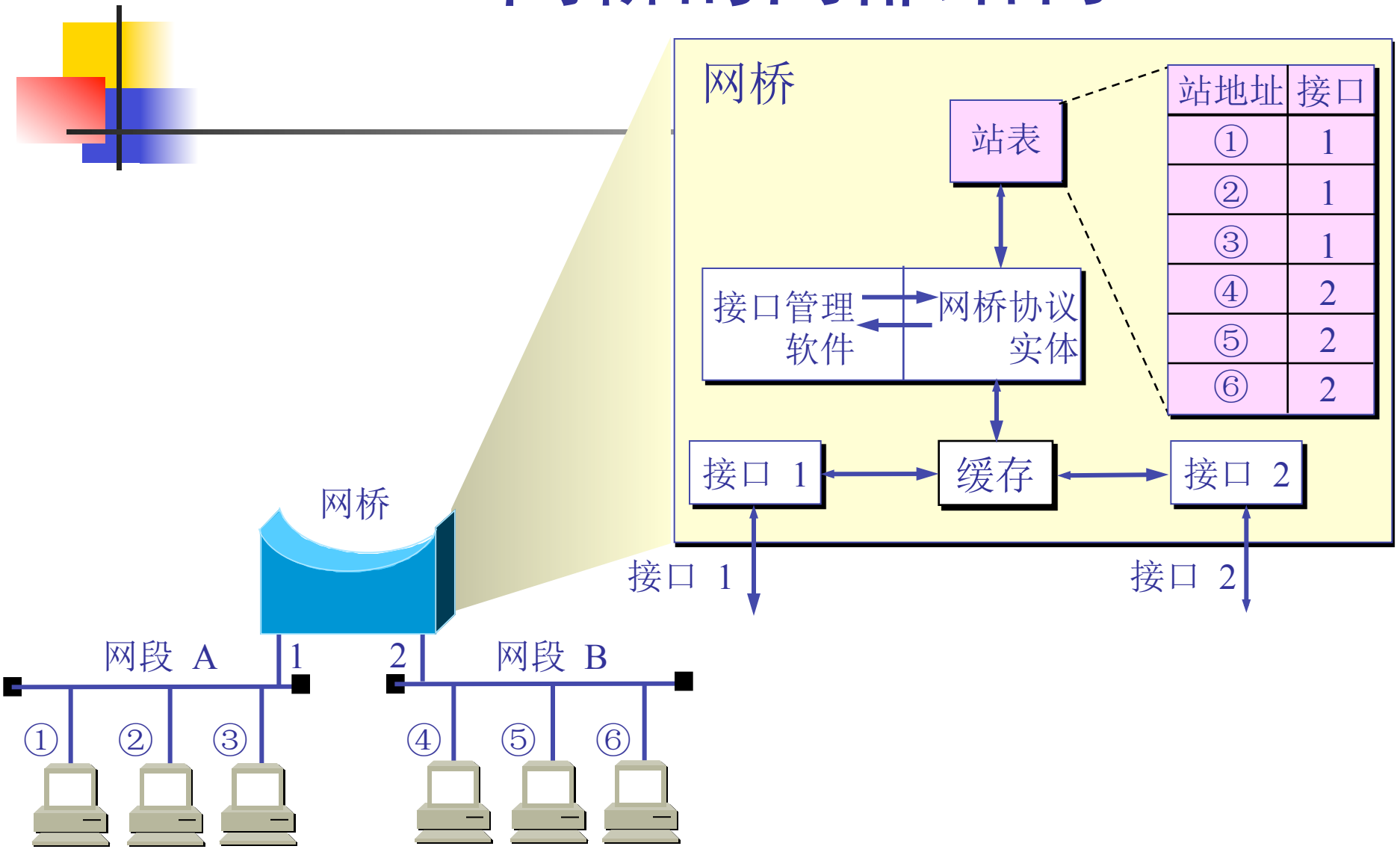
用集线器扩展以太网

- 优点
  - 使原来属于不同碰撞域的以太网上的计算机能够进行跨碰撞域的通信。
  - 扩大了以太网覆盖的地理范围。
- 缺点
  - 碰撞域增大了，但总的吞吐量并未提高。
  - 如果不同的碰撞域使用不同的数据率，那么就不能用集线器将它们互连起来。

3.5.2 在数据链路层扩展局域网

- 在数据链路层扩展局域网是使用网桥。
- 网桥工作在数据链路层，它根据 MAC 帧的目的地址对收到的帧进行转发。
- 网桥具有过滤帧的功能。当网桥收到一个帧时，并不是向所有的接口转发此帧，而是先检查此帧的目的 MAC 地址，然后再确定将该帧转发到哪一个接口

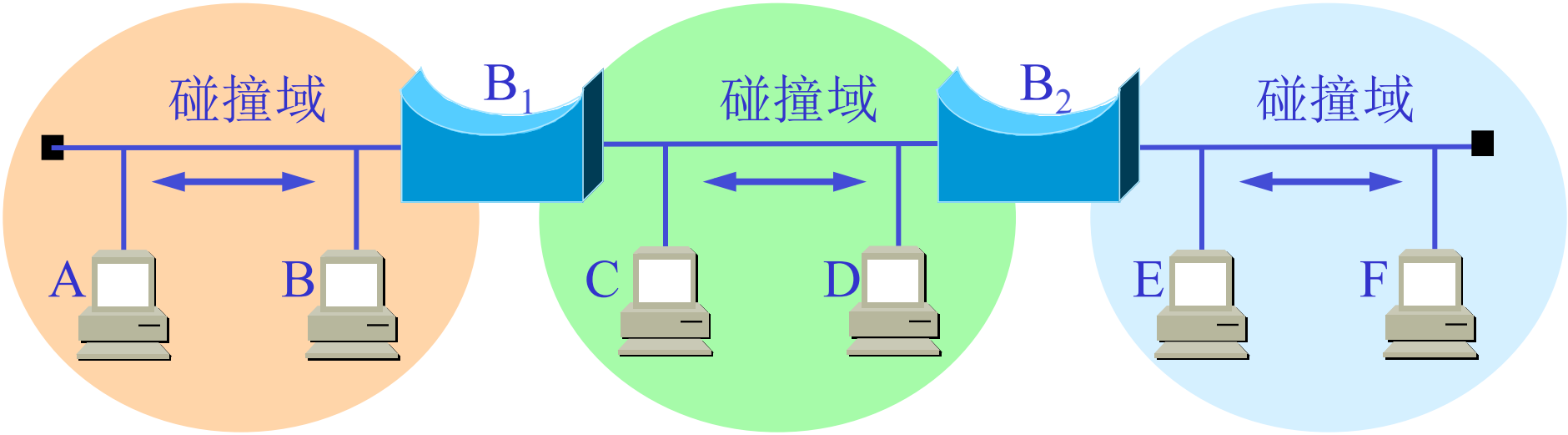
1. 网桥的内部结构


网桥
站表
站地址 接口
① 1
② 1
③ 1
④ 2
⑤ 2
⑥ 2
接口管理软件
网桥协议实体
接口 1
缓存
接口 2
接口 1
接口 2
网桥
网段 A
1
2
网段 B
①
②
③
④
⑤
⑥

使用网桥带来的好处

- 过滤通信量。
- 扩大了物理范围。
- 提高了可靠性。
- 可互连不同物理层、不同 MAC 子层和不同速率（如10 Mb/s 和 100 Mb/s 以太网）的局域网。

网桥使各网段成为隔离开的碰撞域



网桥和集线器（或转发器）不同

- 集线器在转发帧时，不对传输媒体进行检测。
- 网桥在转发帧之前必须执行 CSMA/CD 算法。
  - 若在发送过程中出现碰撞，就必须停止发送和进行退避。

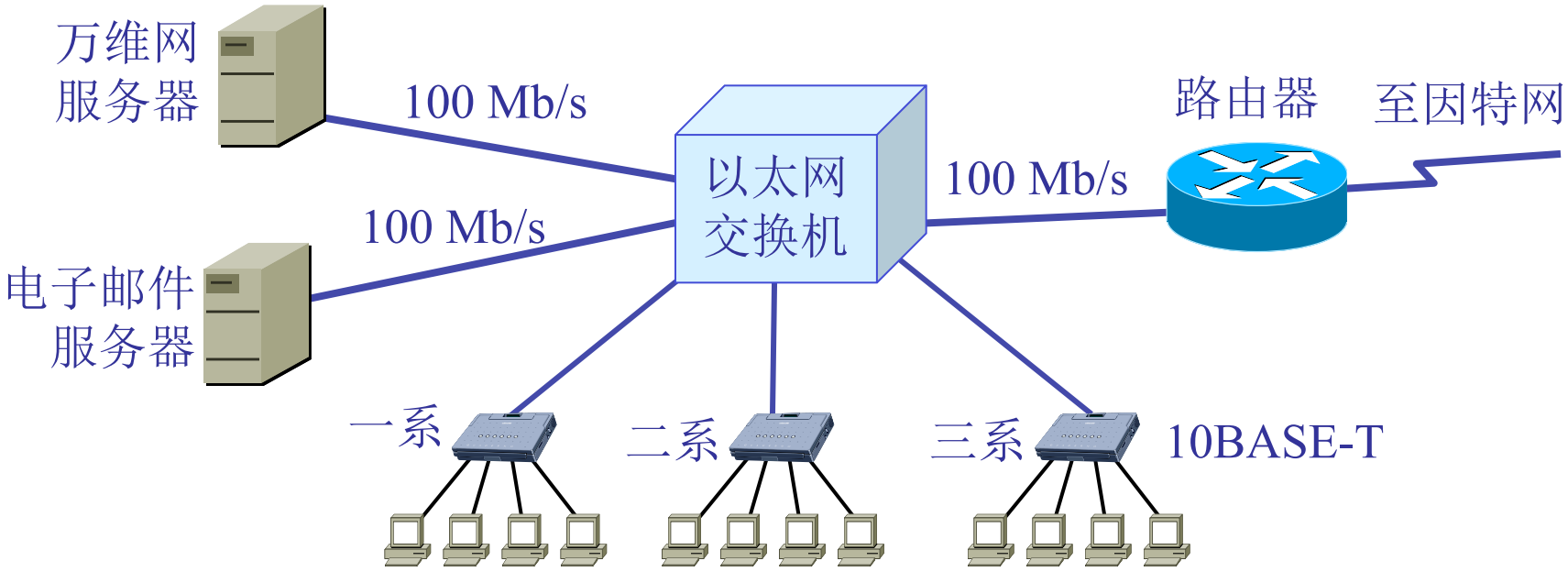
3.4.2 在数据链路层扩展以太网

- 扩展以太网更常用的方法是在数据链路层进行。
- 早期使用网桥（将两个网络连接起来），现在使用以太网交换机（多接口网桥）。

- 网桥工作在数据链路层。
- 它根据 MAC 帧的目的地址对收到的帧进行转发和过滤。
- 当网桥收到一个帧时，并不是向所有的接口转发此帧，而是先检查此帧的目的 MAC 地址，然后再确定将该帧转发到哪一个接口，或把它丢弃。

- 1990 年问世的交换式集线器（switching hub）可明显地提高以太网的性能。
- 交换式集线器常称为以太网交换机（switch）或第二层交换机（L2 switch），强调这种交换机工作在数据链路层。

用以太网交换机扩展局域网



以太网交换机的优点

- 用户独享带宽，增加了总容量。
  - 对于普通 10 Mbit/s 的共享式以太网，若共有 *N* 个用户，则每个用户占有的平均带宽只有总带宽 (10 Mbit/s)的 *N* 分之一。
  - 使用以太网交换机时，虽然在每个接口到主机的带宽还是 10 Mbit/s，但由于一个用户在通信时是独占而不是和其他网络用户共享传输媒体的带宽，因此对于拥有 *N* 个接口的交换机的总容量为 *N*×10 Mbit/s。
- 从共享总线以太网转到交换式以太网时，所有接入设备的软件和硬件、适配器等都不需要做任何改动。
- 以太网交换机一般都具有多种速率的接口，方便了各种不同情况的用户。

以太网交换机的自学习功能

交互机将不同的LAN连接起来，那么哪些设备可以连到交换机的端口上

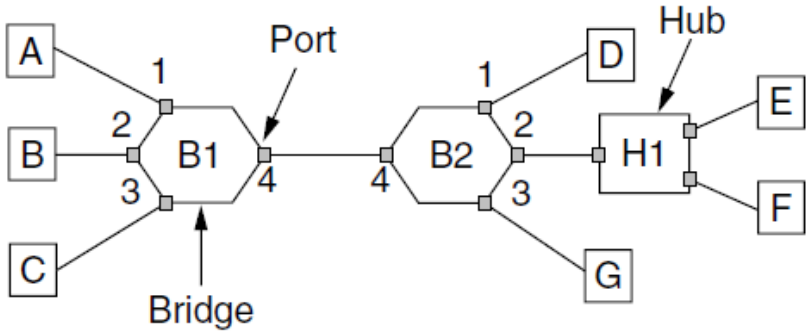
- Computers, bridges, and hubs connect to its ports

交换机如何工作? For example: A 发送帧到D

- 帧先到达port 1 of B1. B1发现目的地为D, 所以朝port 4发
- B2收到帧并发现目的地为D, 所以发往port 1
- 帧到达D

假设帧的目的地端口和接收端口是同一个端口怎么办?

- Frame from E to F arrives at port 2 of B2, B2 just discards the frame



以太网交换机的自学习功能

Question: 交换机如何根据目的地来选取一个端口?

Backward learning algorithm picks the output port:

- 当收到一个帧时，将帧的原地址和这个接收端口联系起来(learning)，即在forwarding table记录数据
- 帧如果要发给某个地址，而这个地址已经有相关的端口，则发送
- 如果这个地址没有相关端口，则发往所有的其他端口

无需配置

- 有效时间字段可将一些过时的信息自动删除

交换表一开始是空的

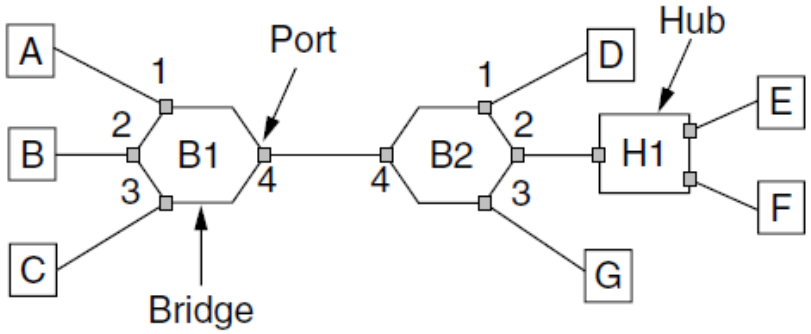
以太网交换机的自学习功能

假设forwarding table在开始时是空的，帧的发送顺序如下，请给出交换机以及hub的发送策略

1. C sends a frame to D; 2. E to F; 3. B to C; 4. F to E; 5. B to E; 6. C to F

Solution:

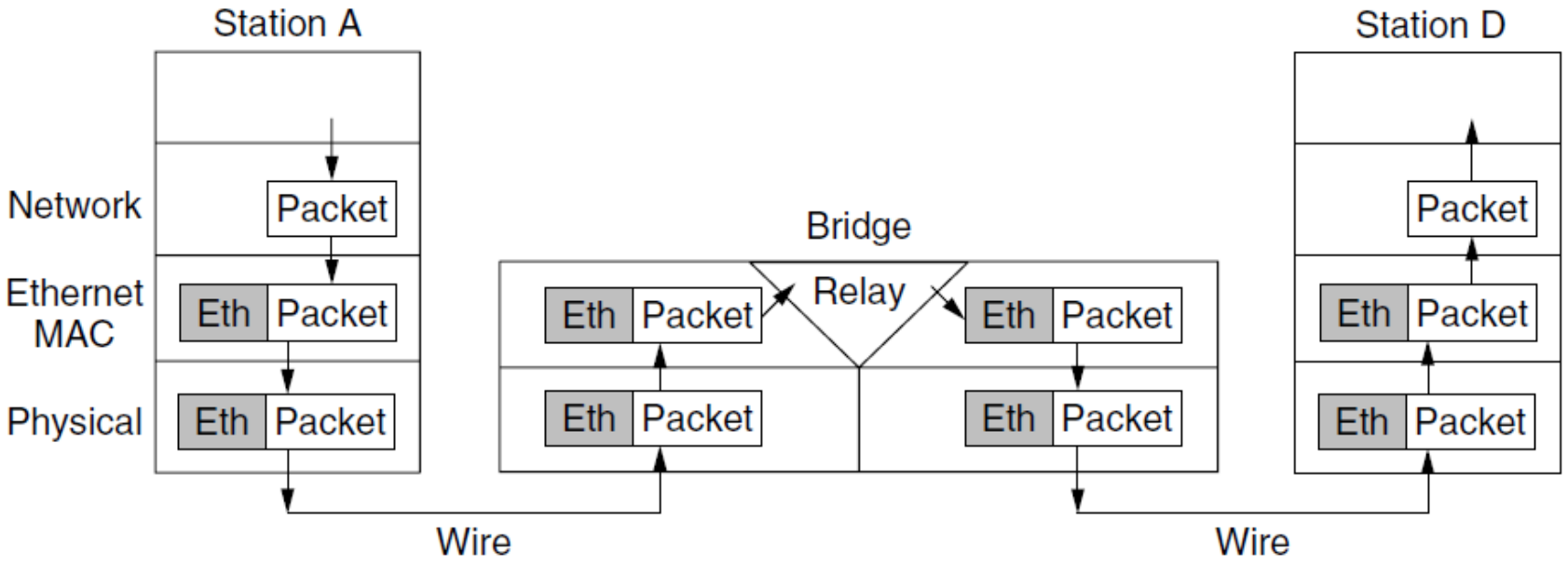
1. B1-> (1,2,4), B2->(1,2,3), H1->(E,F) 2. H1->(F, B2), B2->(1,3,4), B1->(1,2,3); 3. B1->(3); 4. H1->(E, B2), B2 discards; 5. B1->(4), B2->(2), H1->(E, F); 6. B1->(1,2,4), B2->(1,2,3), H1->(E,F)



以太网交换机

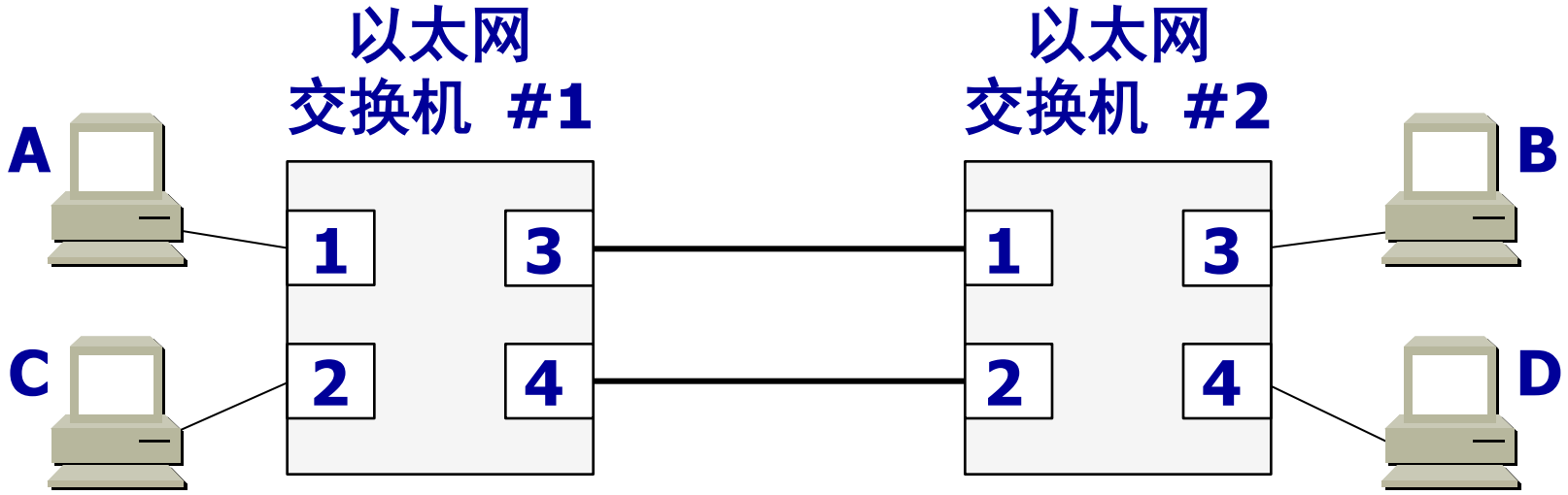
下图显示了一个数据从站A发送到站D的过程

- 交换机不对网络层操作
- 交换机属于链路层设备



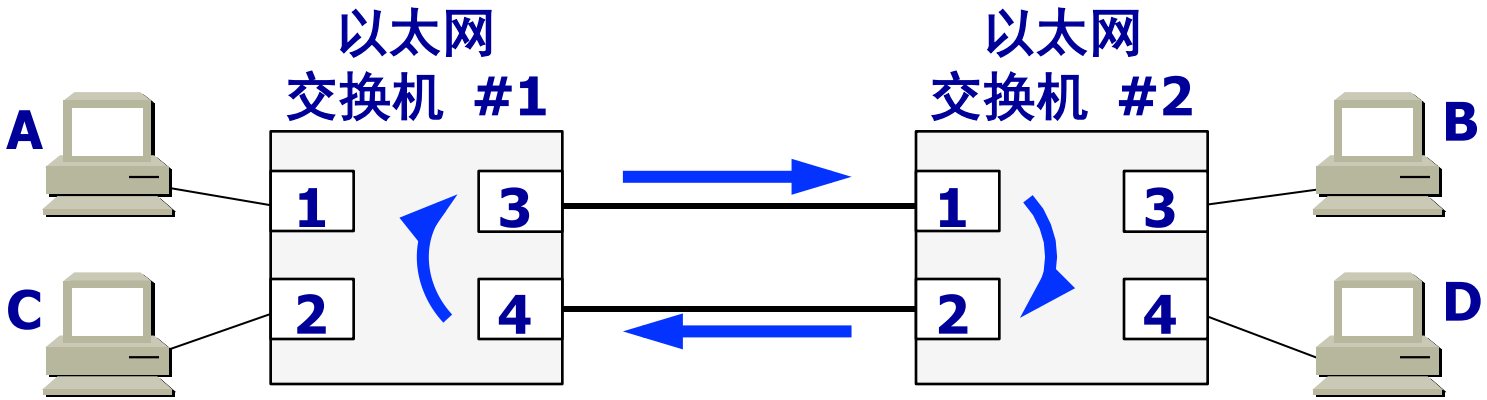
交换机使用了生成树协议

- 增加冗余链路时，自学习的过程就可能导致以太网帧在网络的某个环路中无限制地兜圈子。
- 如图，假定开始时，交换机 #1 和 #2 的交换表都是空的，主机 A 通过接口交换机 #1 向主机 B 发送一帧。



交换机使用了生成树协议

- 按交换机自学习和转发方法，该帧的某个走向如下：离开交换机 #1 的接口 3 → 交换机 #2 的接口 1 → 接口 2 → 交换机 #1 的接口 4 → 接口 3 → 交换机 #2 的接口 1 →……。这样就无限制地循环兜圈子下去，白白消耗了网络资源。



在两个交换机之间兜圈子的帧

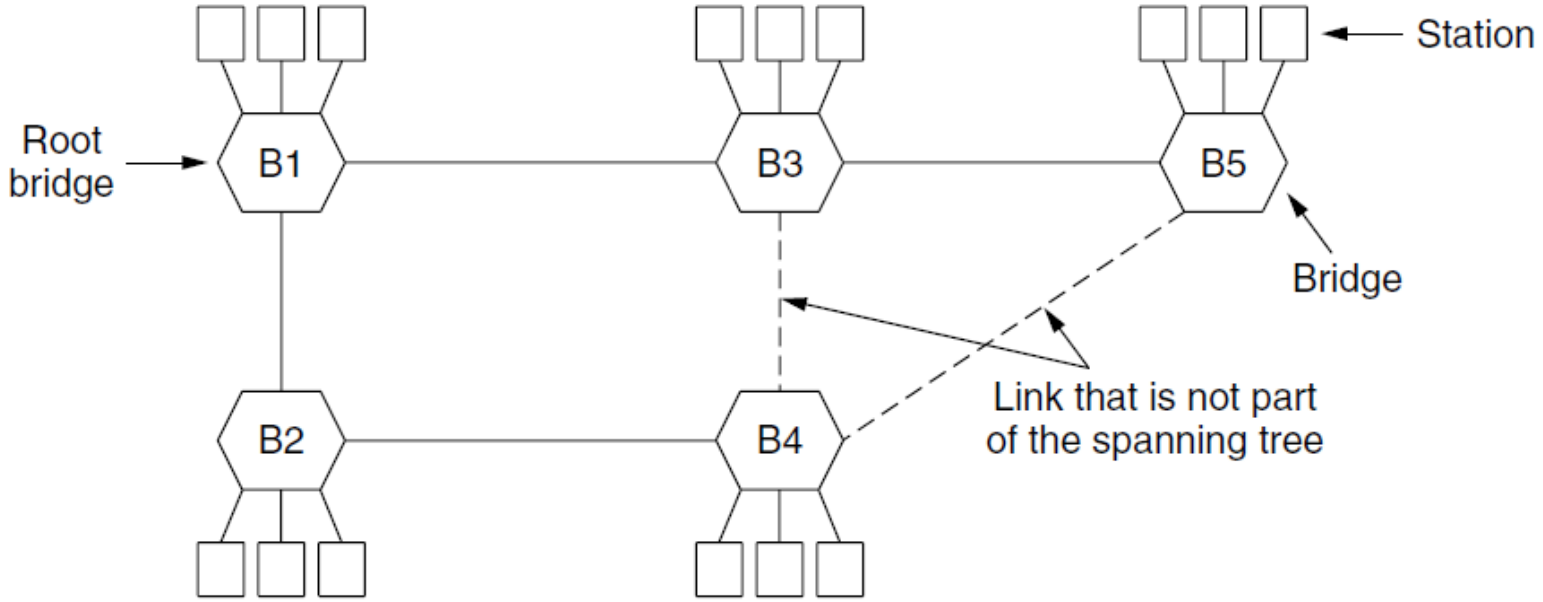
交换机使用了生成树协议

- IEEE 802.1D 标准制定了一个生成树协议 STP (Spanning Tree Protocol)。
- 其要点是：不改变网络的实际拓扑，但在逻辑上则切断某些链路，使得从一台主机到所有其他主机的路径是无环路的树状结构，从而消除了兜圈子现象。

交换机使用了生成树协议

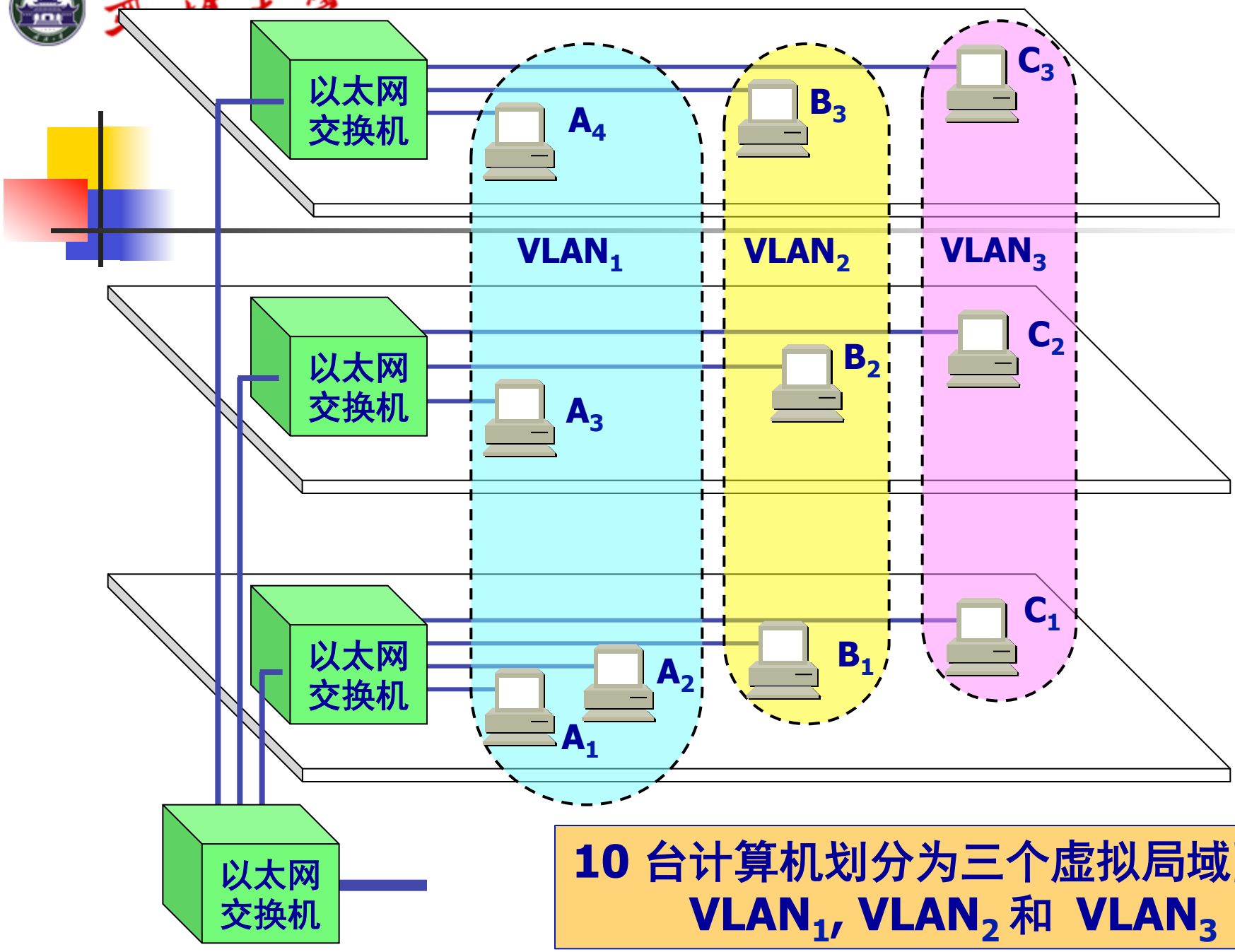
使用生成树后:

- B1 是根节点，虚线连接被去除掉（备用）
- B4 使用B2的链接 (比B3的链接跳数少)
- B5 使用B3 (比B4得链接跳数少)

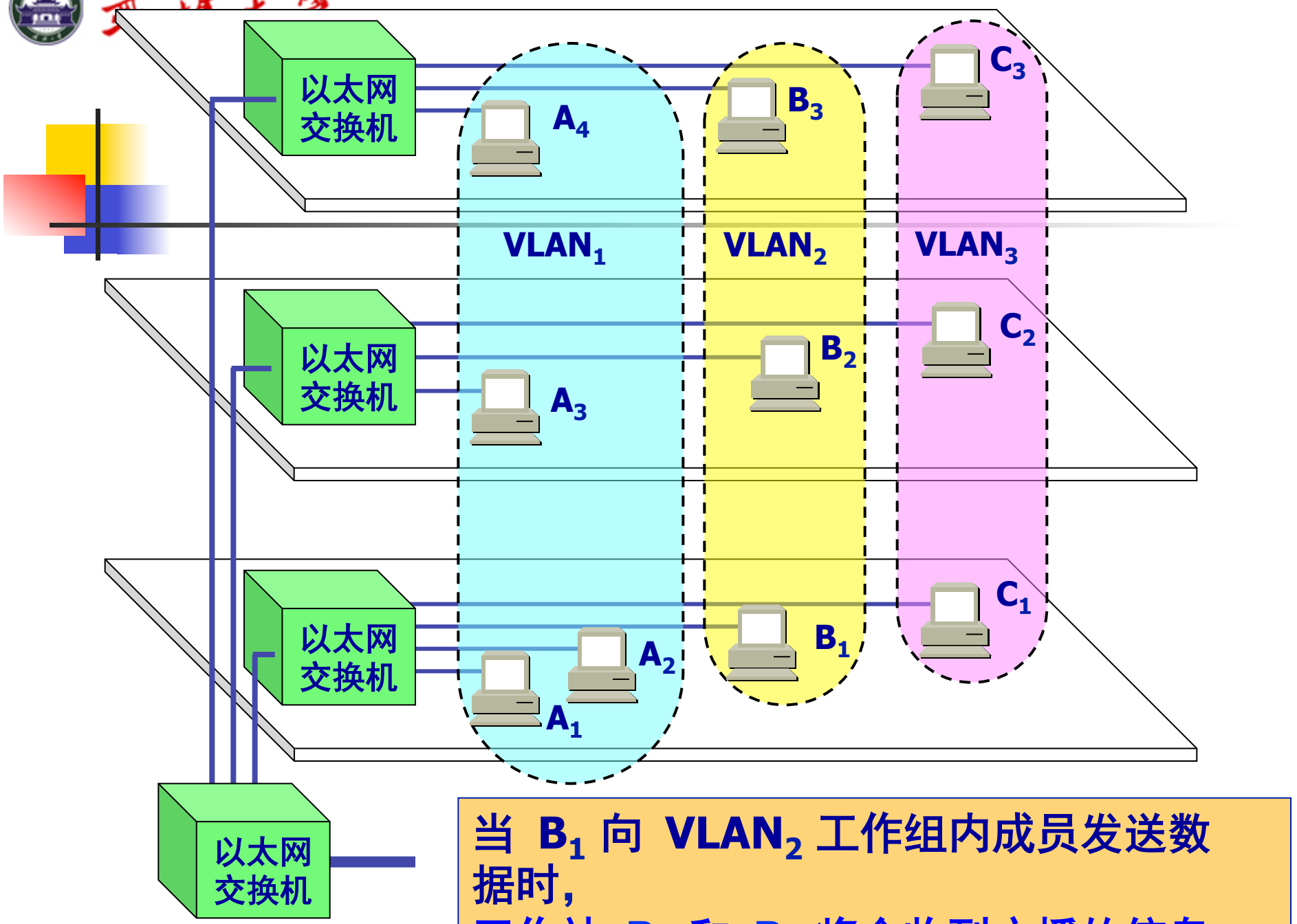


3.4.3 虚拟局域网

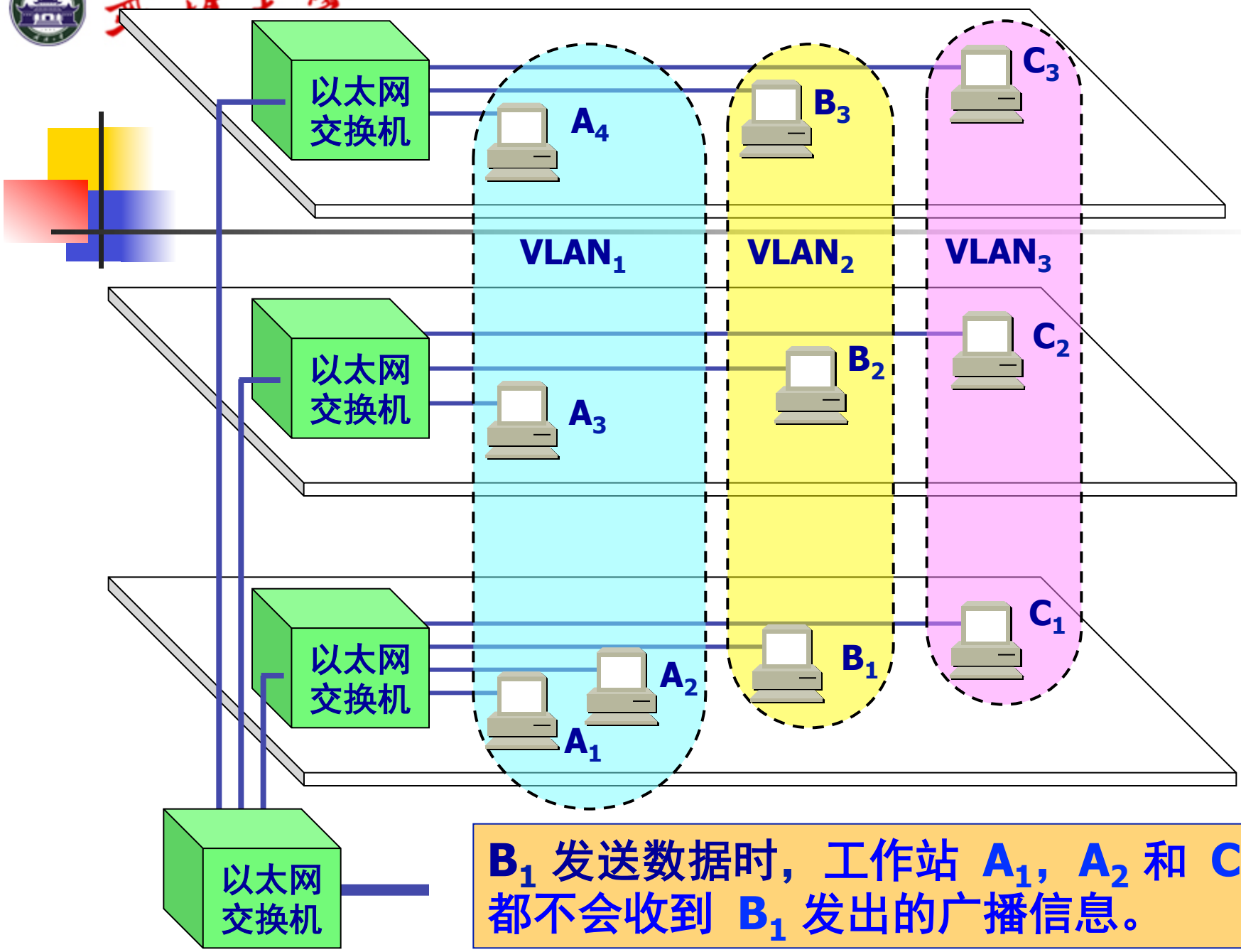
- 利用以太网交换机可以很方便地实现虚拟局域网 VLAN (Virtual LAN)。
- 虚拟局域网 VLAN 是由一些局域网网段构成的与物理位置无关的逻辑组，而这些网段具有某些共同的需求。每一个 VLAN 的帧都有一个明确的标识符，指明发送这个帧的计算机是属于哪一个 VLAN。
- 虚拟局域网其实只是局域网给用户提供的一种服务，而并不是一种新型局域网。
- 由于虚拟局域网是用户和网络资源的逻辑组合，因此可按照需要将有关设备和资源非常方便地重新组合，使用户从不同的服务器或数据库中存取所需的资源。



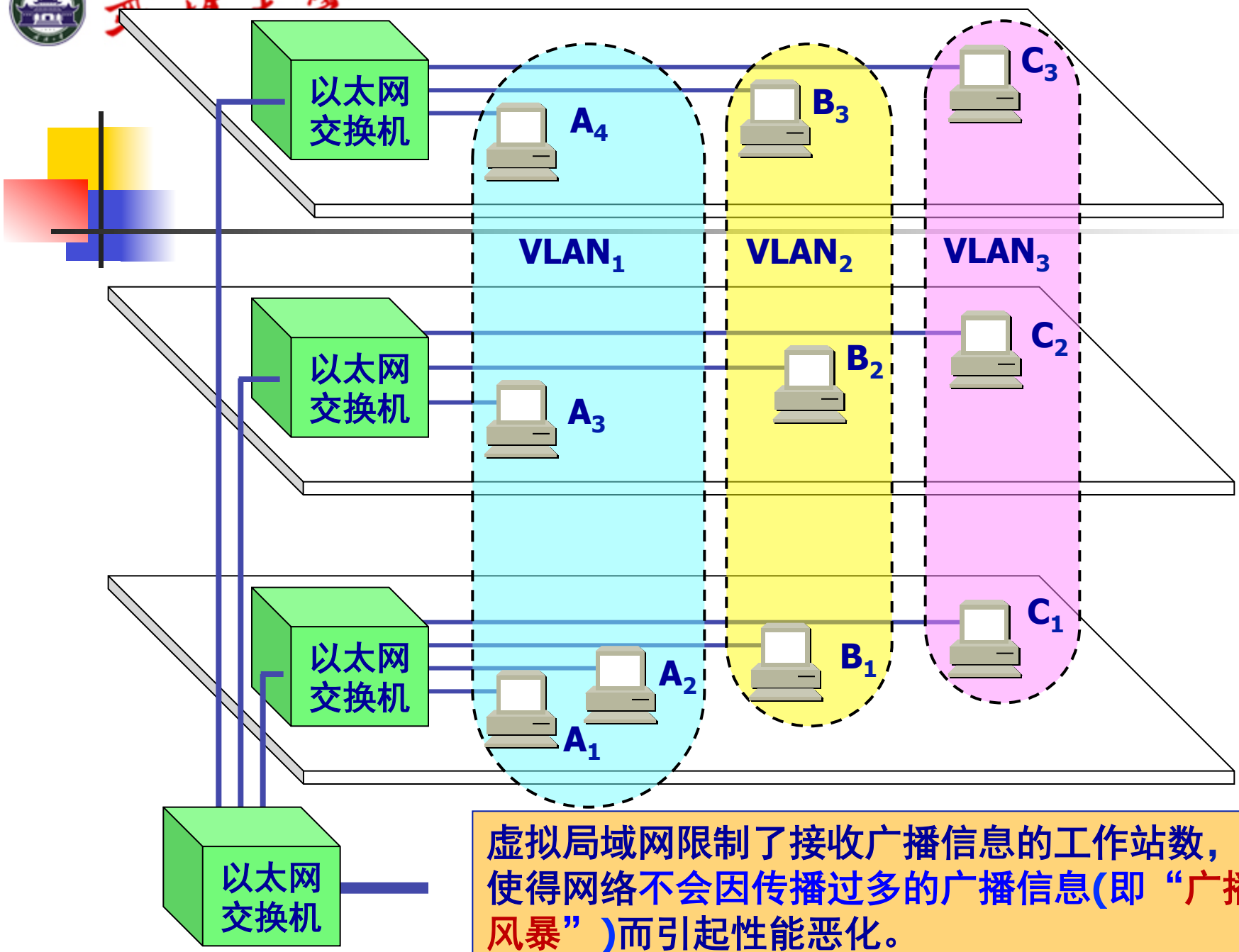
10 台计算机划分为三个虚拟局域网：
$VLAN_1$, $VLAN_2$ 和 $VLAN_3$



当 B_1 向 $VLAN_2$ 工作组内成员发送数据时，
工作站 B_2 和 B_3 将会收到广播的信息。



B_1 发送数据时，工作站 A_1，A_2 和 C_1 都不会收到 B_1 发出的广播信息。



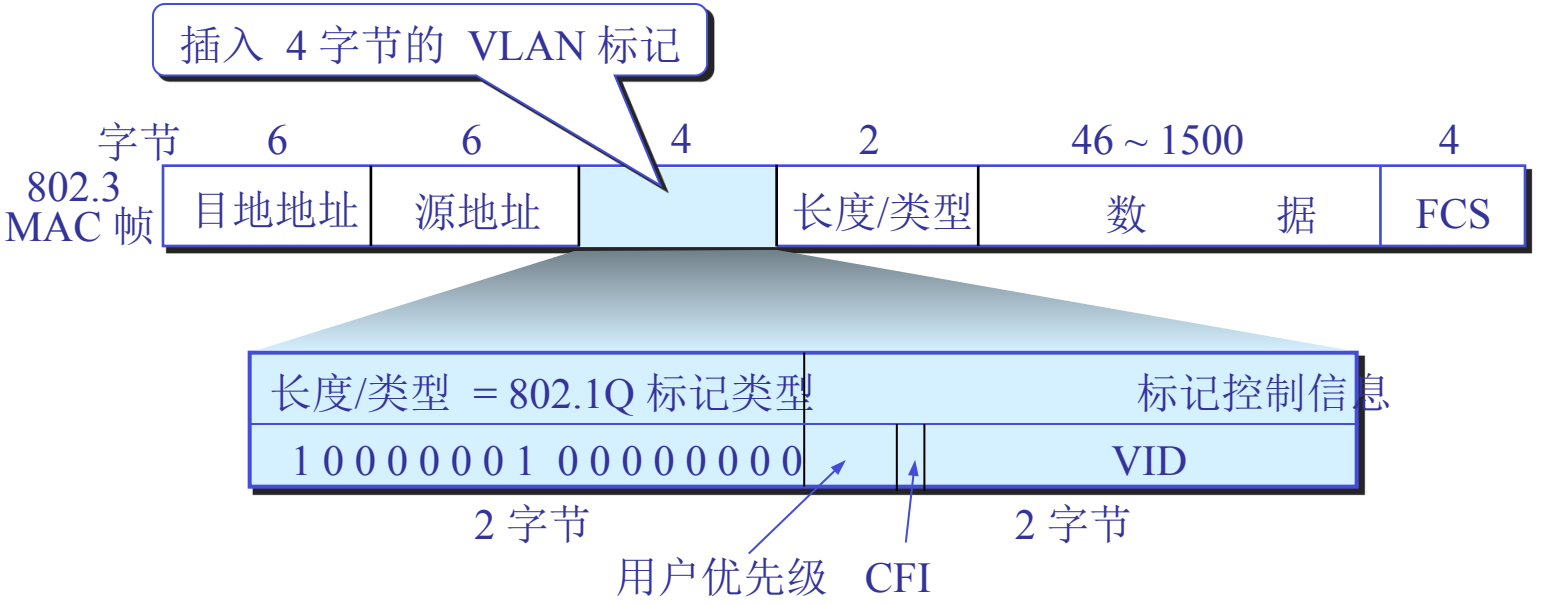
虚拟局域网限制了接收广播信息的工作站数，使得网络不会因传播过多的广播信息(即“广播风暴”)而引起性能恶化。

虚拟局域网使用的以太网帧格式

- IEEE 批准了 802.3ac 标准，该标准定义了以太网的帧格式的扩展，以支持虚拟局域网。
- 虚拟局域网协议允许在以太网的帧格式中插入一个4字节的标识符，称为 VLAN 标记（tag），用来指明发送该帧的计算机属于哪一个虚拟局域网。
- 插入 VLAN 标记得出的帧称为 802.1Q 帧或 带标记的以太网帧。

虚拟局域网使用的以太网帧格式

- 虚拟局域网协议允许在以太网的帧格式中插入一个 4 字节的标识符，称为 VLAN 标记(tag)，用来指明发送该帧的工作站属于哪一个虚拟局域网。



如何保证兼容性？

3.6 高速以太网

3.6.1 100BASE-T 以太网

- 速率达到或超过 100 Mb/s 的以太网称为高速以太网。
- 在双绞线上传送 100 Mb/s 基带信号的星型拓扑以太网，仍使用 IEEE 802.3 的CSMA/CD 协议。100BASE-T 以太网又称为快速以太网(Fast Ethernet)。

100BASE-T 以太网的特点

- 可在全双工方式下工作而无冲突发生。因此，不使用 CSMA/CD 协议。
- MAC 帧格式仍然是 802.3 标准规定的。
- 保持最短帧长不变，但将一个网段的最大电缆长度减小到 100 m。
- 帧间时间间隔从原来的 9.6 μs 改为现在的 0.96 μs。

3.6.2 吉比特以太网

- 允许在 1 Gb/s 下全双工和半双工两种方式工作。
- 使用 802.3 协议规定的帧格式。
- 在半双工方式下使用 CSMA/CD 协议（全双工方式不需要使用 CSMA/CD 协议）。
- 与 10BASE-T 和 100BASE-T 技术向后兼容。

吉比特以太网的物理层

- 1000BASE-X　基于光纤通道的物理层：
  - 1000BASE-SX　SX表示短波长
  - 1000BASE-LX　LX表示长波长
  - 1000BASE-CX　CX表示铜线
- 1000BASE-T
  - 使用 4对 5 类线 UTP

全双工方式

- 当吉比特以太网工作在全双工方式时（即通信双方可同时进行发送和接收数据），不使用载波延伸和分组突发。

作业

- P109：3-2，3-6，3-8，3-10，3-13，3-19，3-20，3-22，3-27，3-30，3-33，3-34