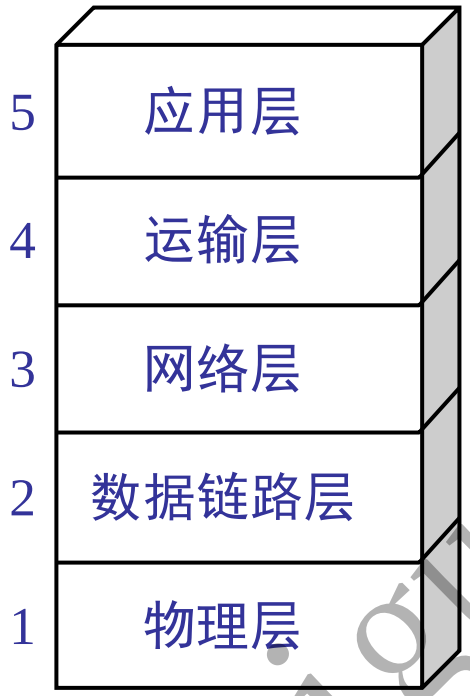
# 计算机网络

顾 军

计算机学院

jgu@cumt.edu.cn

# 专题3：数据帧怎么到达目的结点




- 应用层(application layer)
- 运输层(transport layer)
- 网络层(network layer)
- 数据链路层(data link layer)
- 物理层(physical layer)

# Q5：数据链路层向上提供什么服务？

- 在数据链路层使用CRC检验，能够实现无比特差错的传输，但还不是可靠传输。
- 所谓“可靠传输”就是：数据链路层的发送端发送什么，在接收端就收到什么。
- 传输差错可分为两大类：
  - 最基本的比特差错，可以通过CRC检错
  - 收到的帧并没有出现比特差错，但却出现了帧丢失、帧重复或帧失序（后发送的帧反而先到达接收端）

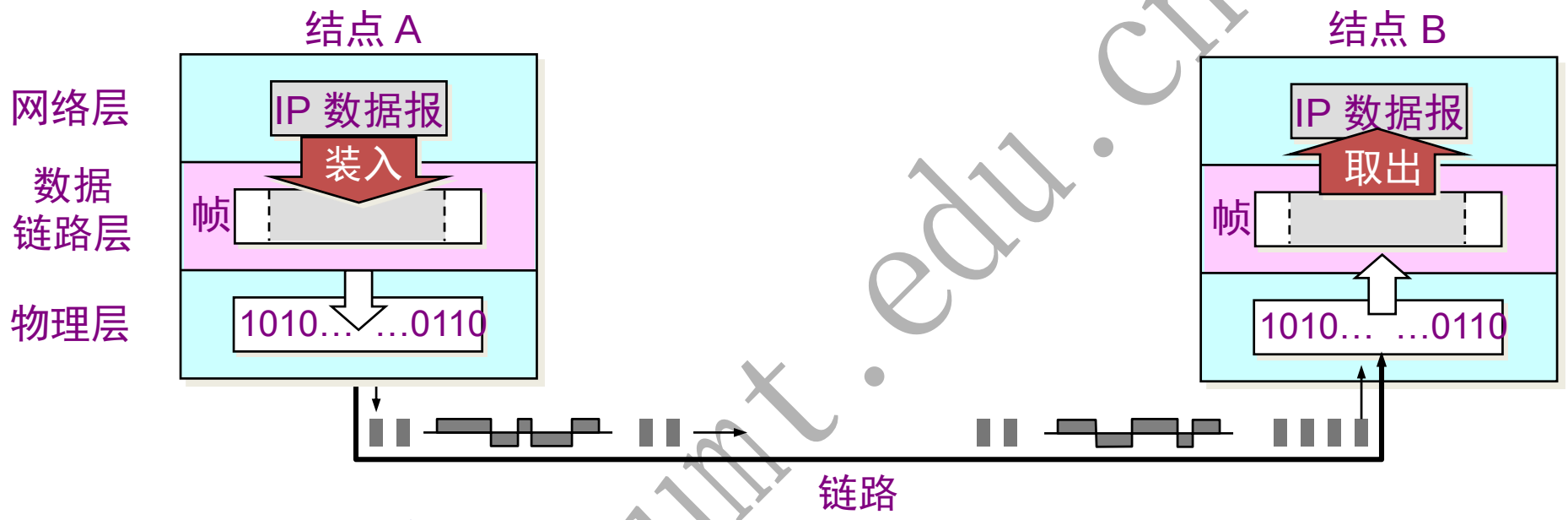
# 过去对数据链路层的设计

- 过去OSI的观点：必须让数据链路层向上提供可靠传输。
- 因此，在CRC检错的基础上，增加了帧编号、确认和重传机制。
  - 收到正确的帧就要向发送端发送确认。发送端在一定的期限内若没有收到对方的确认，就认为出错了，因而就进行重传，直到收到对方的确认为止。
- 随着通信线路的质量大大提高，由通信链路质量不好引起差错的概率已经大大降低。

# 现在对数据链路层的设计

- 现在互联网采取区别对待的方法：
  - 对于通信质量良好的有线传输链路，数据链路层不使用确认和重传机制，即不要求数据链路层向上提供可靠传输的服务。
    - 如果在数据链路层传输数据时出现了差错并且需要改正，那么就由上层协议（如TCP协议）来完成。
  - 对于通信质量较差的无线传输链路，数据链路层协议使用确认和重传机制，因而数据链路层可以向上提供可靠传输的服务。
- 实践证明，这样做可以提高通信效率。

# Q6：点对点协议——PPP？

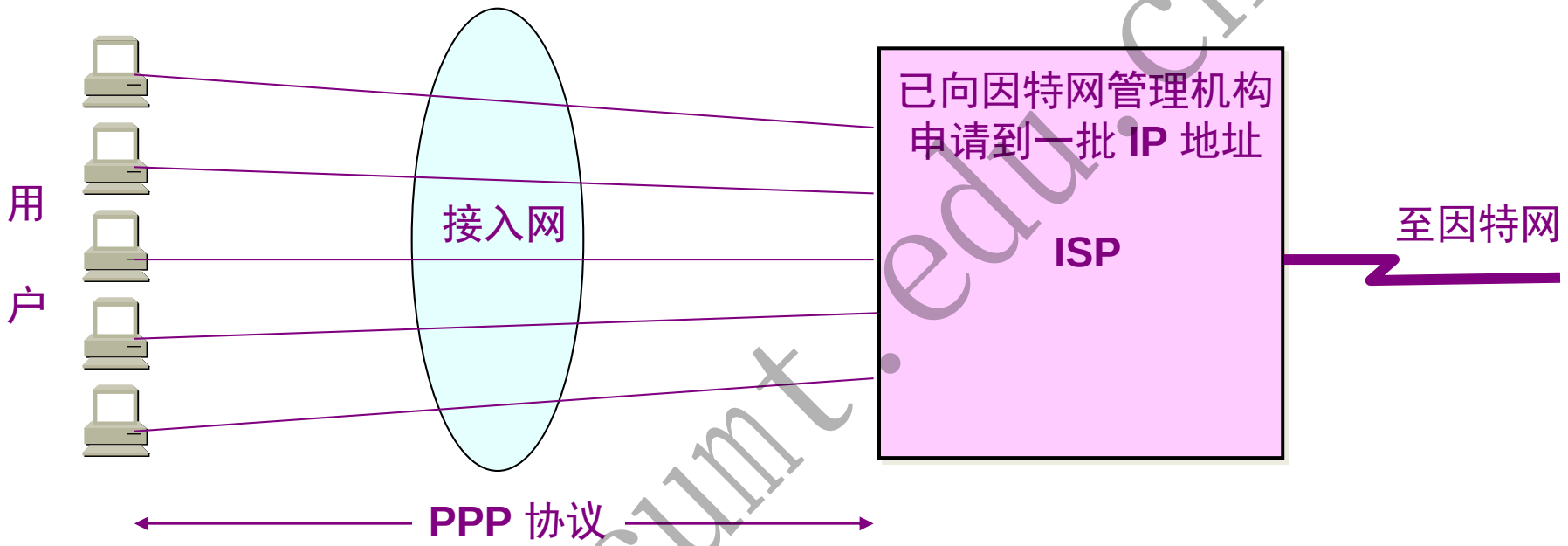



- 通信时的主要步骤如下：
  - 结点A的数据链路层把网络层交下的IP数据报添加首部和尾部封装成帧。
  - 结点A把封装好的帧发送给结点B的数据链路层。
  - 若结点B的数据链路层收到的帧无差错，则从收到的帧中提取出IP数据报上交给上面的网络层；否则丢弃这个帧。

# PPP 协议

- 点对点协议 PPP (Point-to-Point Protocol)，是现在全世界使用得最多的数据链路层协议。
- 1992 年制订了 PPP 协议。
- 经过 1993 年和 1994 年的修订，现在的 PPP 协议已成为因特网的正式标准[RFC 1661]。

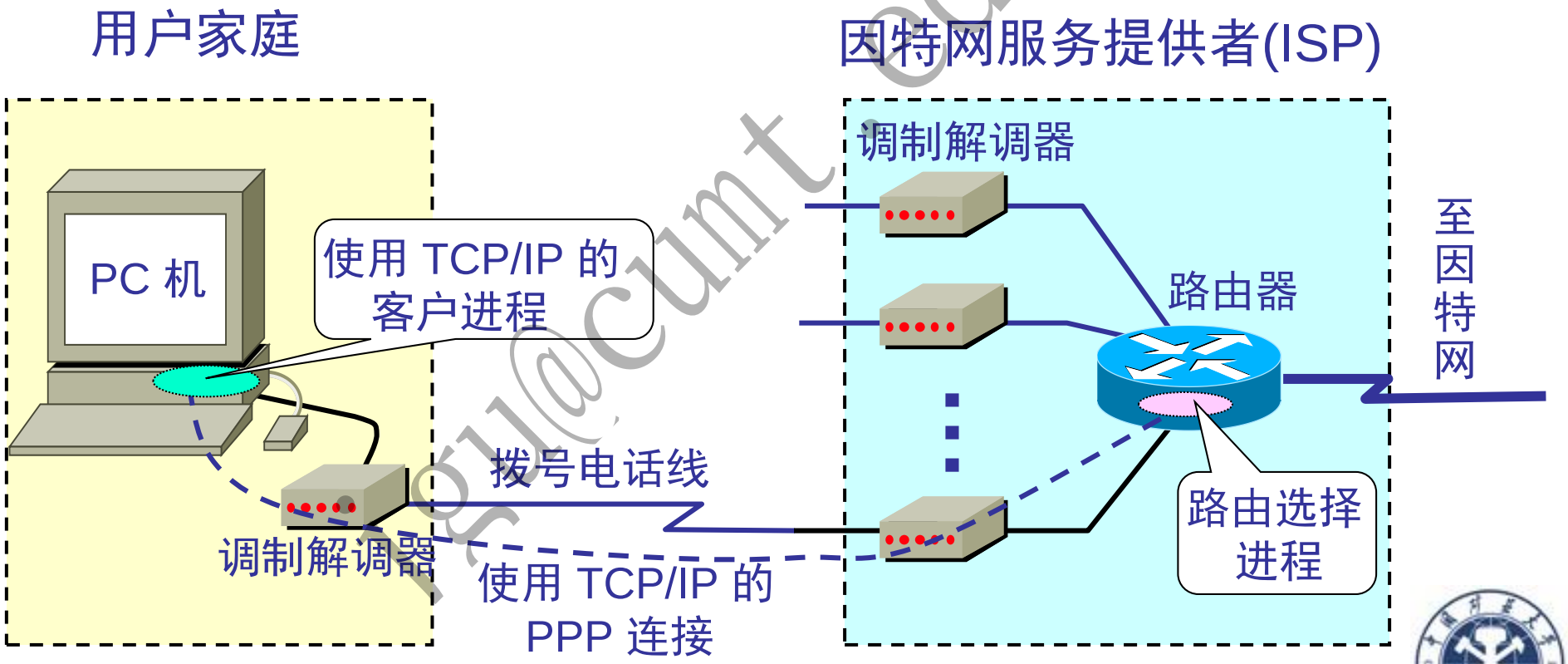
# 用户到 ISP 的链路使用 PPP 协议




- 1999年公布了PPP over Ethernet (PPPoE)，[RFC 2516]，把PPP帧再封装在以太网帧中。
- 宽带上网时由于数据传输速率较高，因此可以让多个连接在以太网上的用户共享一条到ISP的宽带链路。

# 用户拨号入网的示意图

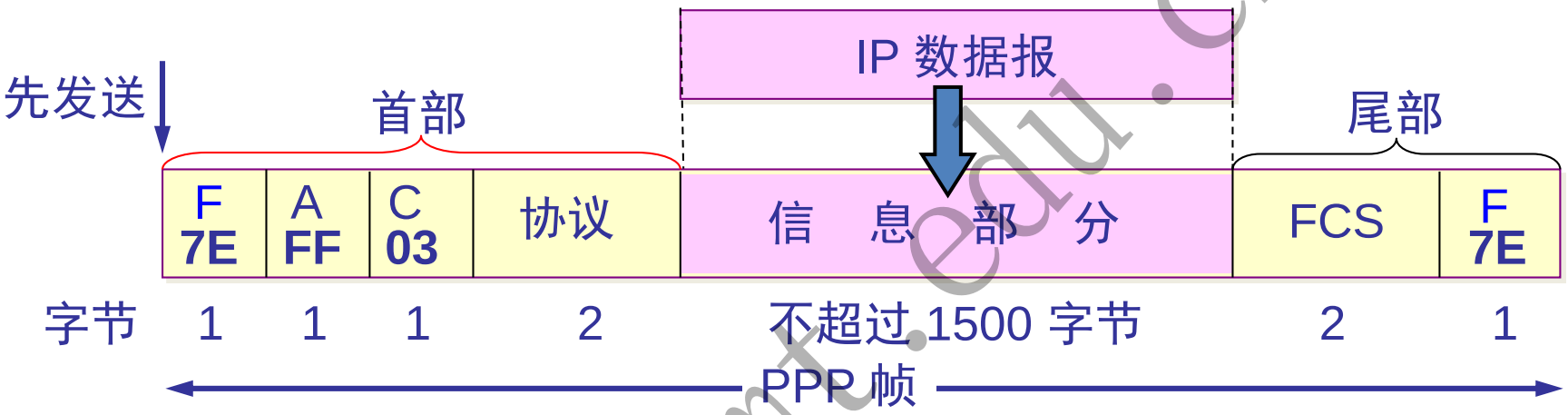
- 只有一个用户利用拨号电话线或ADSL进行宽度上网，也是使用PPP协议。



# PPP 协议的三个组成部分

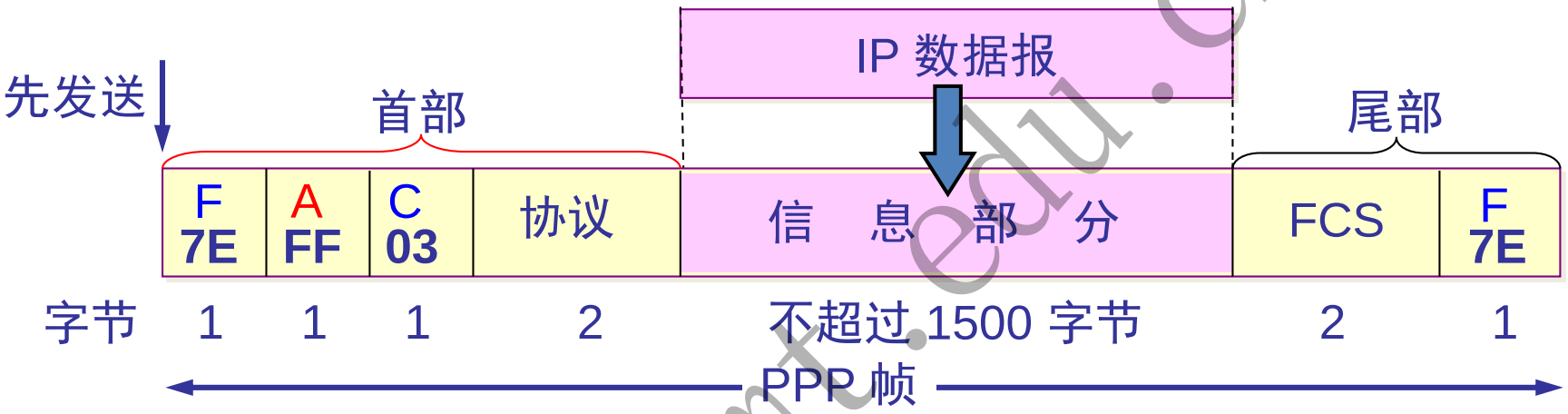
- 一个将 IP 数据报封装到串行链路的方法
  - 既支持异步链路（无奇偶检验的8比特数据）
  - 也支持面向比特的同步链路
- 链路控制协议 LCP (Link Control Protocol)
  - 用来建立、配置和测试数据链路连接
  - 通信的双方可以协商一些选项
  - RFC1661中定义了11种类型的LCP分组
- 一套网络控制协议 NCP (Network Control Protocol)
  - 其中的每一个协议支持不同的网络层协议，如IP、OSI的网络层、DECnet、Novell网的IPX，以及AppleTalk等

# PPP 协议的帧格式



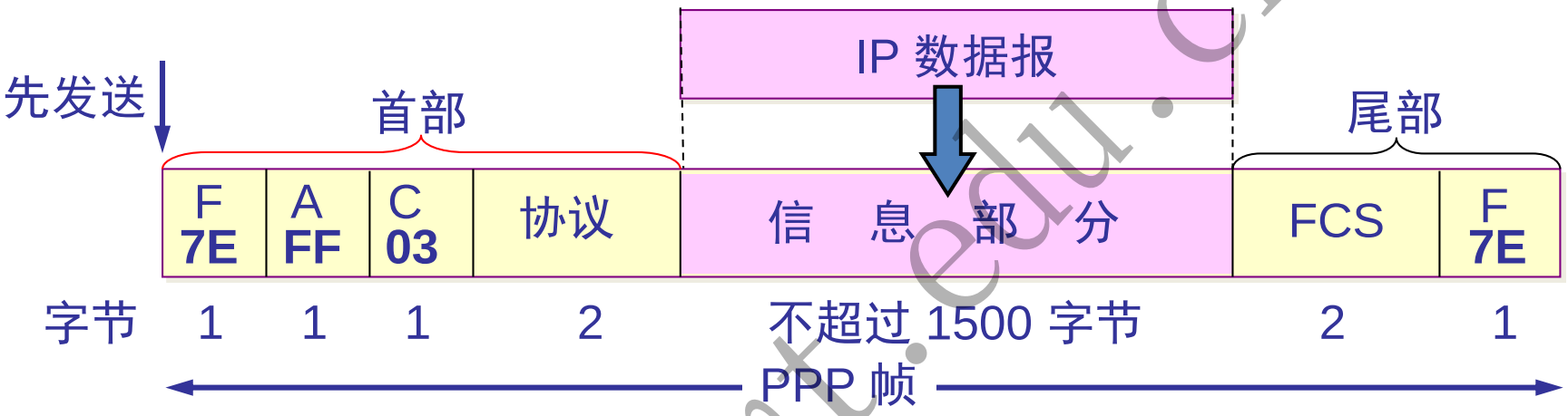

- 标志字段（Flag） F = 0x7E（01111110），表示一个帧的开始或结束，也就是PPP帧的定界符。
- 连续两帧之间只需要用一个标志字段。
- 如果出现连续两个标志字段，就表示这是一个空帧，应当丢弃。
- FCS使用的是CRC序列

# PPP 协议的帧格式



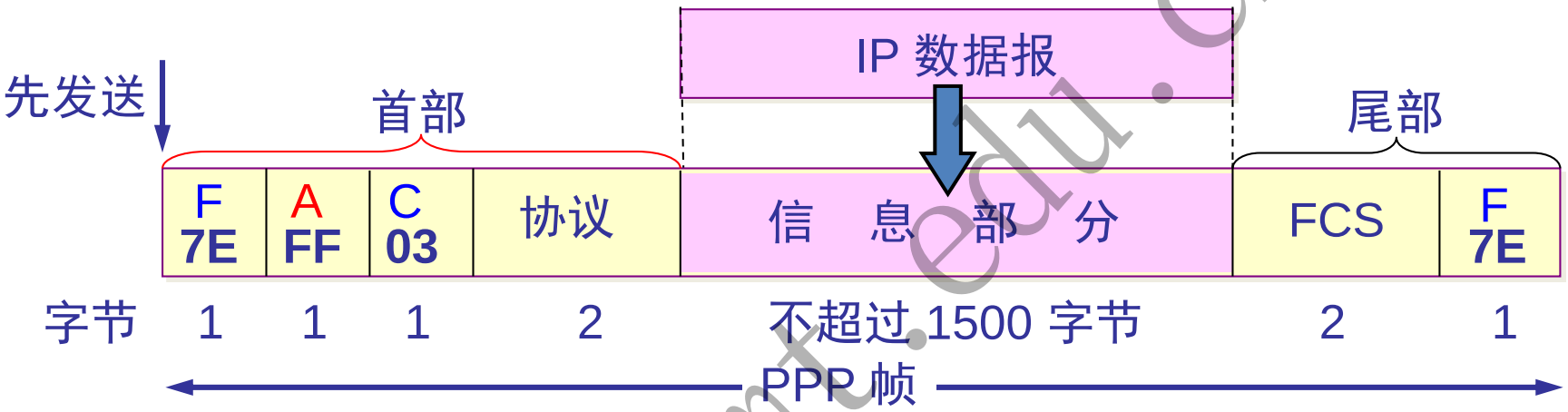

- 地址字段 A 规定为 0xFF
- 控制字段 C 通常置为 0x03
- 最初曾考虑以后对字段A和C进行其它定义，但至今也没有给出
- 所以，地址字段和控制字段实际上对PPP帧并不起作用

# PPP 协议的帧格式




- 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 链路控制协议LCP的数据。
  - 若为 0x8021，则表示这是网络层的控制数据。

# PPP 协议的帧格式




- PPP 是面向字节的，所有 PPP 帧的长度都是整数个字节。
- 信息部分的长度是可变的，不超过1500字节，即MTU的值是1500字节
- 如果高层协议发送的分组过长并超过MTU的数值，PPP就要丢弃这样的帧，并返回差错。

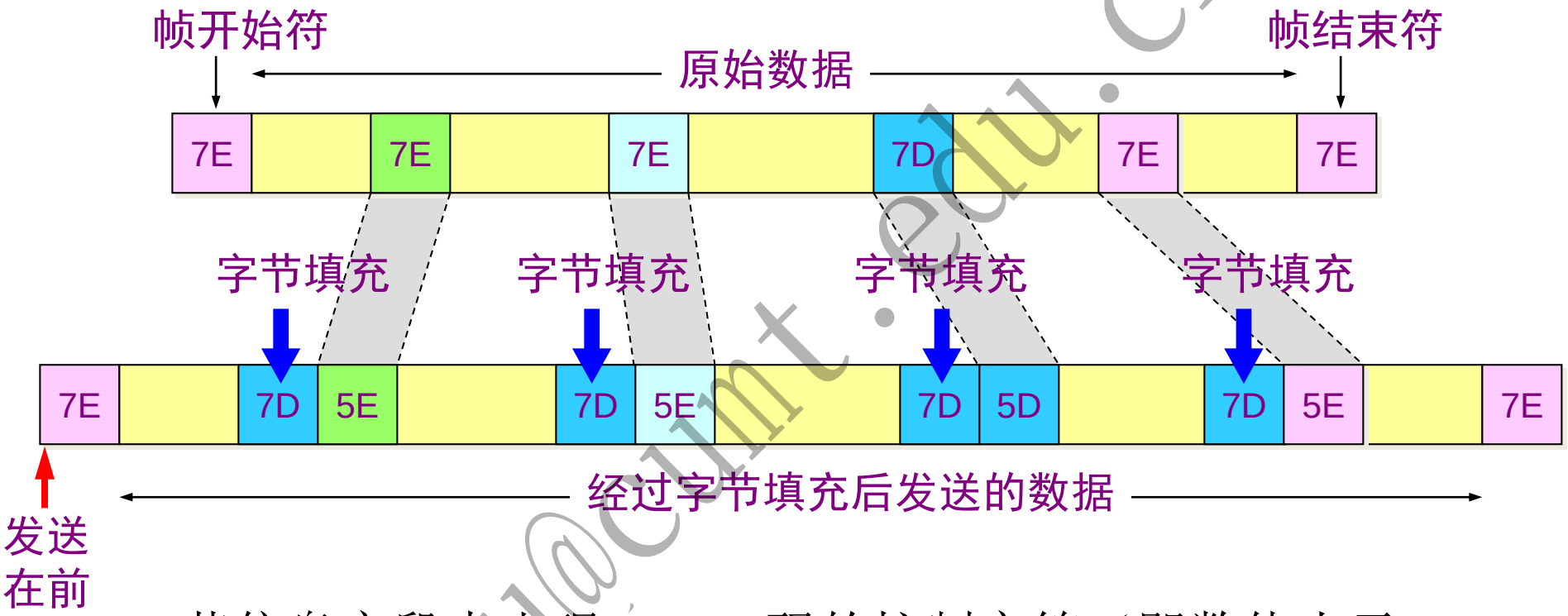
# 透明传输问题

- 当 PPP 用在异步传输时，就使用一种特殊的字符填充法。
- 当 PPP 用在同步传输链路时，协议规定采用硬件来完成比特填充（和 HDLC 的做法一样）。

# 异步传输时的透明传输问题

- 当 PPP 用在异步传输时，就使用一种特殊的字符填充法，即引入了一个转义序列。
  - 转义序列包括一个转义字符 7D，后面是原来的值与 0x20 异或的结果，即7E 转义为7D 5E。而发送 7D 时则转义为 7D 5D。
  - 同样，这种转义方式还保护控制字符，比如，XOFF (Transmit Off) 是底层驱动用来中断串行传输的，为了避免引起混淆，将 0x14 用转义序列 7D 34发送。
  - 默认的，0x00 到 0x1F 之间的所有值都要转义，不过，经过协商后可以去除部分需要转义的值。

# 字符填充法



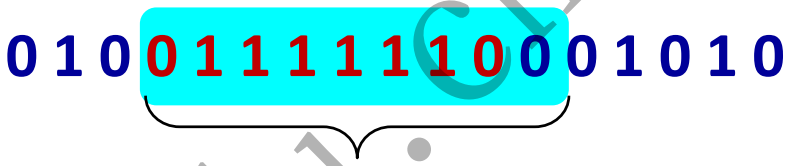

若信息字段中出现 ASCII 码的控制字符（即数值小于 0x20 的字符），则在该字符前面要加入一个 0x7D 字节，同时将该字符的编码加以改变（与 0x20 异或）。

# 同步传输时的透明传输问题

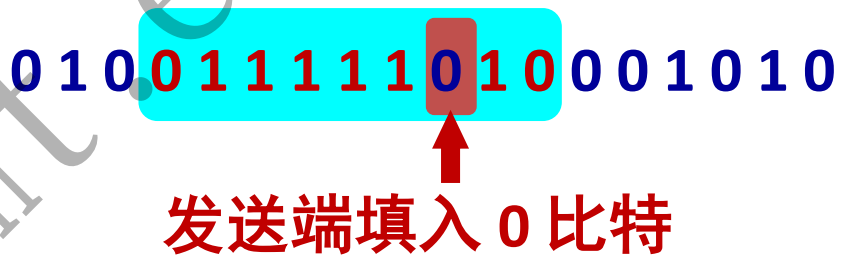
- 当 PPP 用在同步传输链路时，协议规定采用硬件来完成比特填充（和 HDLC 的做法一样）。
- PPP 协议用在 SONET/SDH 链路时，使用同步传输（一连串的比特连续传送）。这时 PPP 协议采用零比特填充方法来实现透明传输。
  - 在发送端，只要发现有 5 个连续 1，则立即填入一个 0。
  - 接收端对帧中的比特流进行扫描。每当发现 5 个连续1时，就把这 5 个连续 1 后的一个 0 删除。

# 零比特的填充与删除

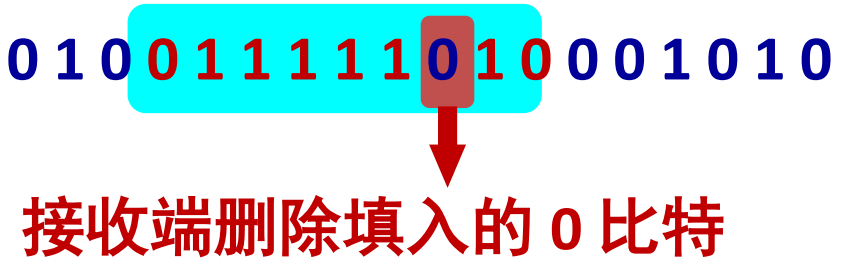
信息字段中出现了和标志字段 F 完全一样的 8 比特组合




发送端在 5 个连 1 之后填入 0 比特再发送出去



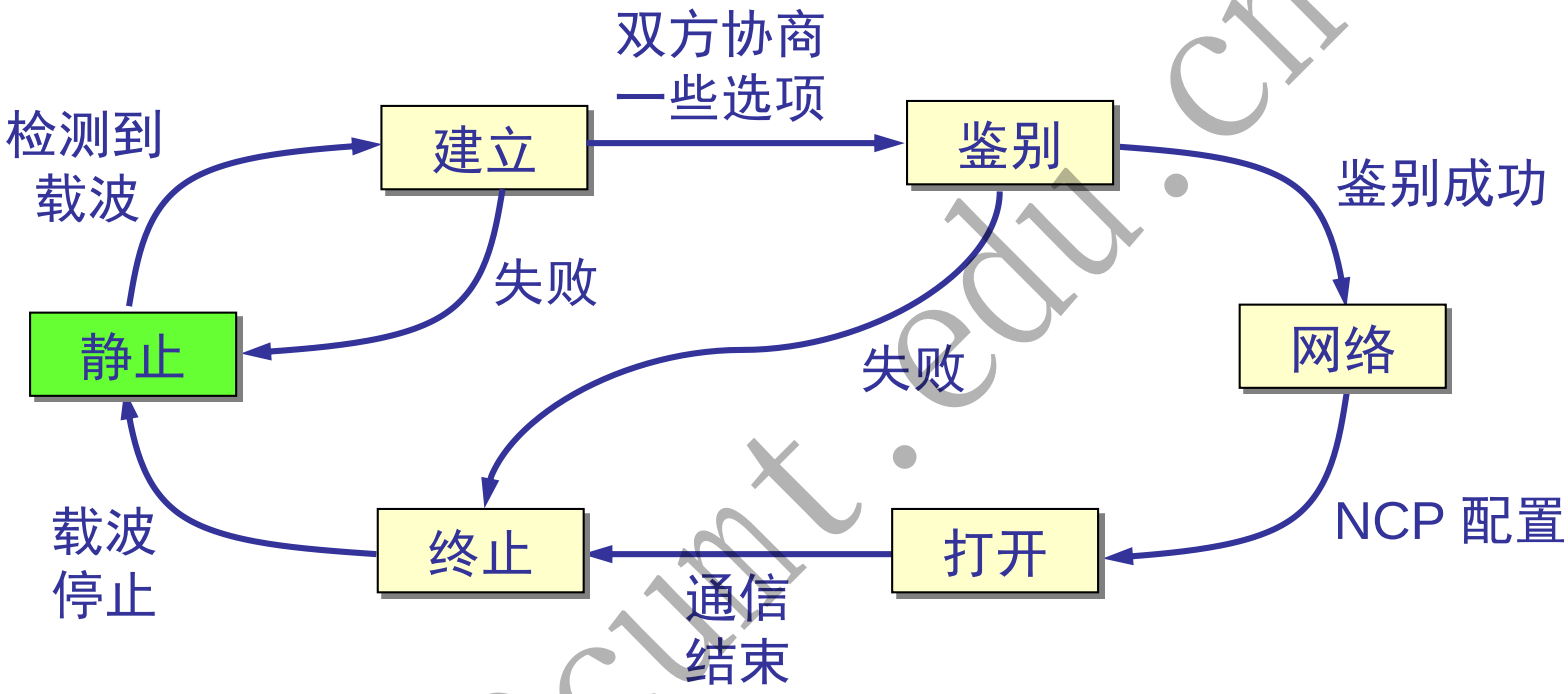

接收端把 5 个连 1 之后的 0 比特删除



# PPP不使用序号和确认机制

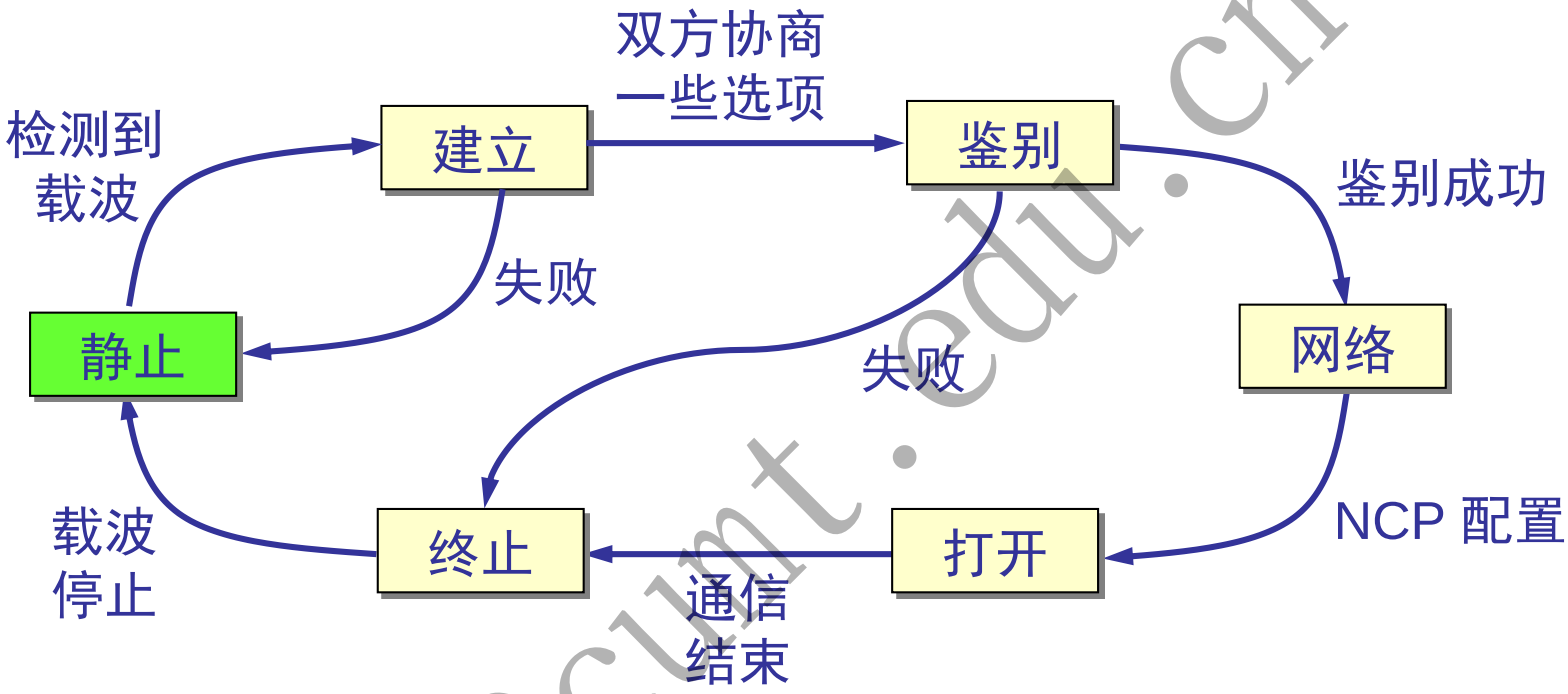
- PPP 协议之所以不使用序号和确认机制是出于以下的考虑：
  - 在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理。
  - 在因特网环境下，PPP 的信息字段放入的数据是 IP 数据报。数据链路层的可靠传输并不能够保证网络层的传输也是可靠的。
  - TCP/IP协议族中，可靠传输由运输层协议TCP负责
  - 帧检验序列 FCS 字段可保证无差错接受。

# PPP 协议的工作状态



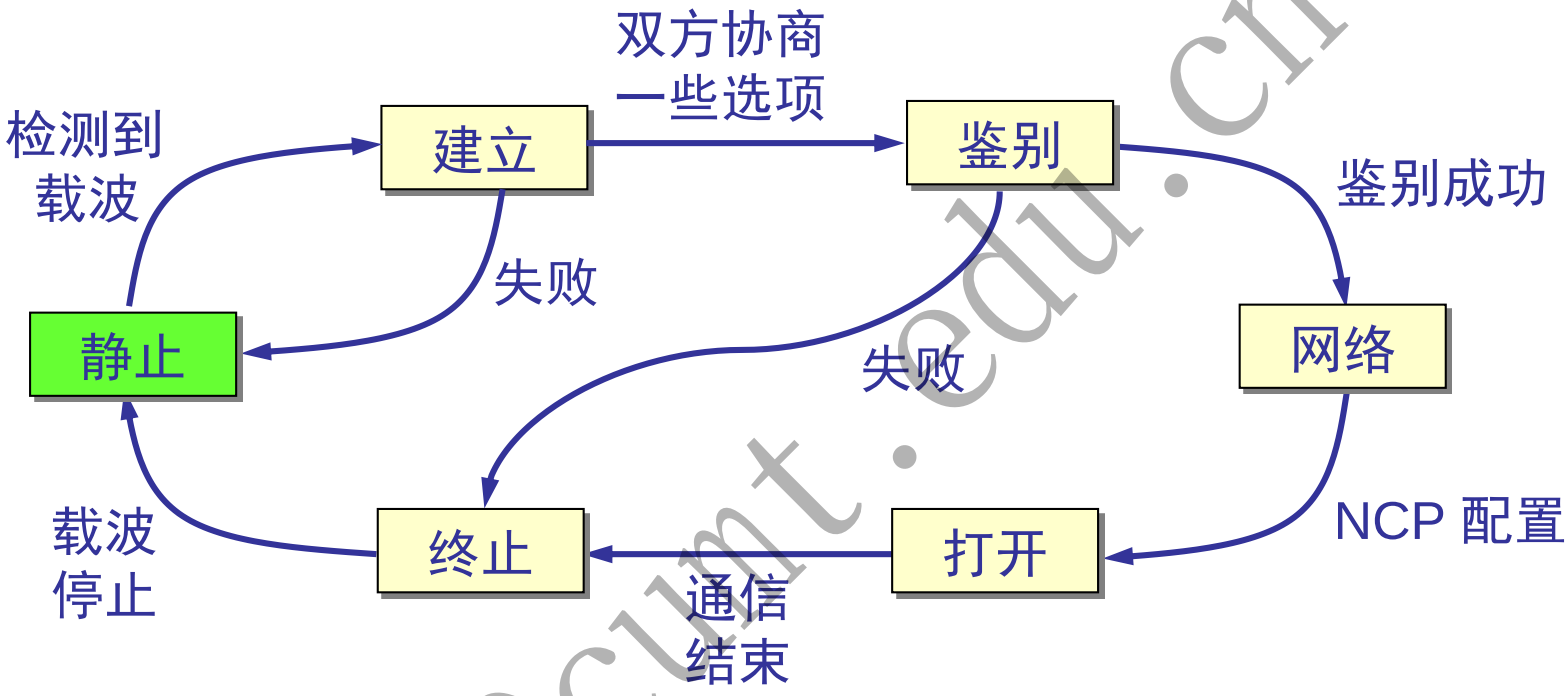

- 当用户拨号接入 ISP 时，路由器的调制解调器对拨号做出确认，并建立一条物理连接。
- PC 机向路由器发送一系列的 LCP 分组（封装成多个 PPP 帧），即配置请求帧(Configure-Request)，其协议字段置为 LCP对应的代码，用来协商一些配置选项。

# PPP 协议的工作状态



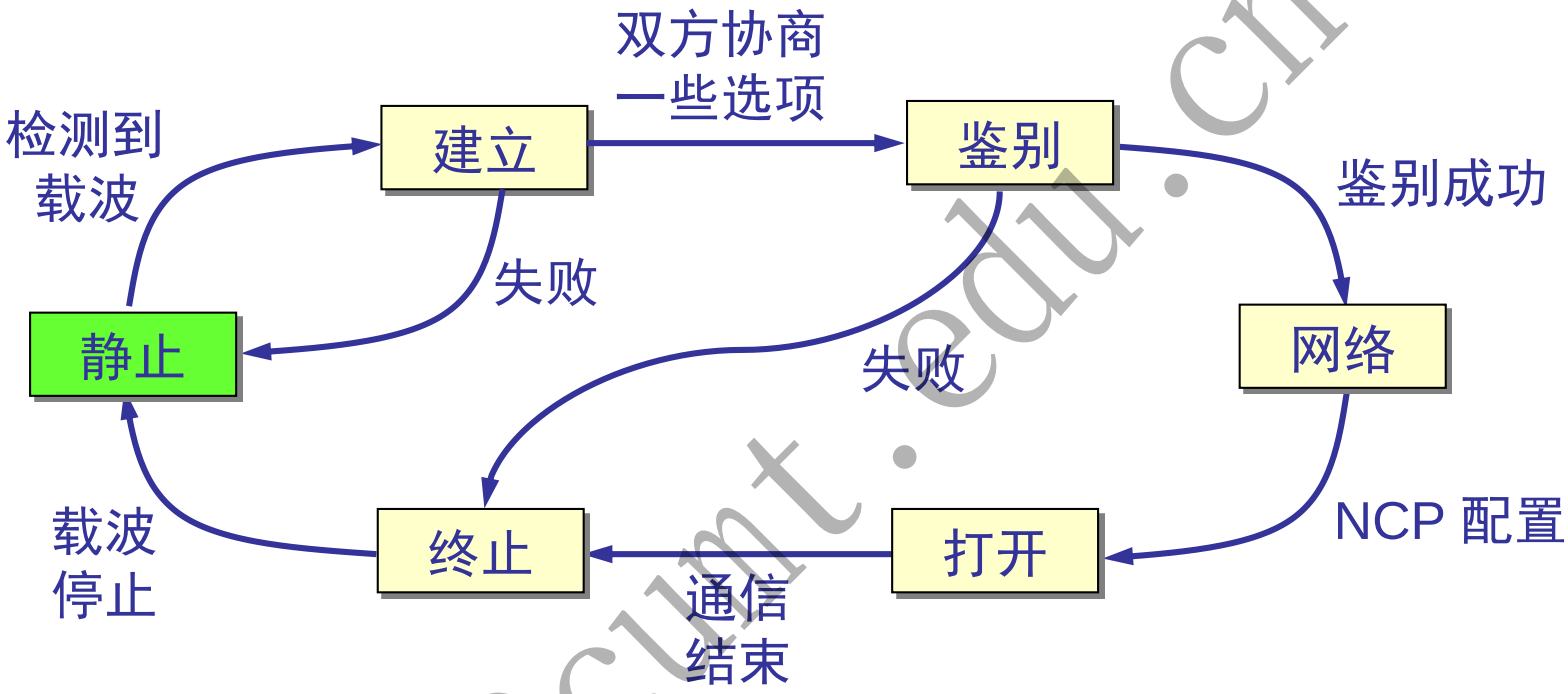

- 协商结束后双方就建立了LCP链路，接着就进入“鉴别”状态。
- PPP 中支持的鉴别协议包括口令鉴别协议（PAP，Password Authentication Protocol）和挑战握手鉴别协议（CHAP，Challenge-Handshake Authentication Protocol）。

# PPP 协议的工作状态



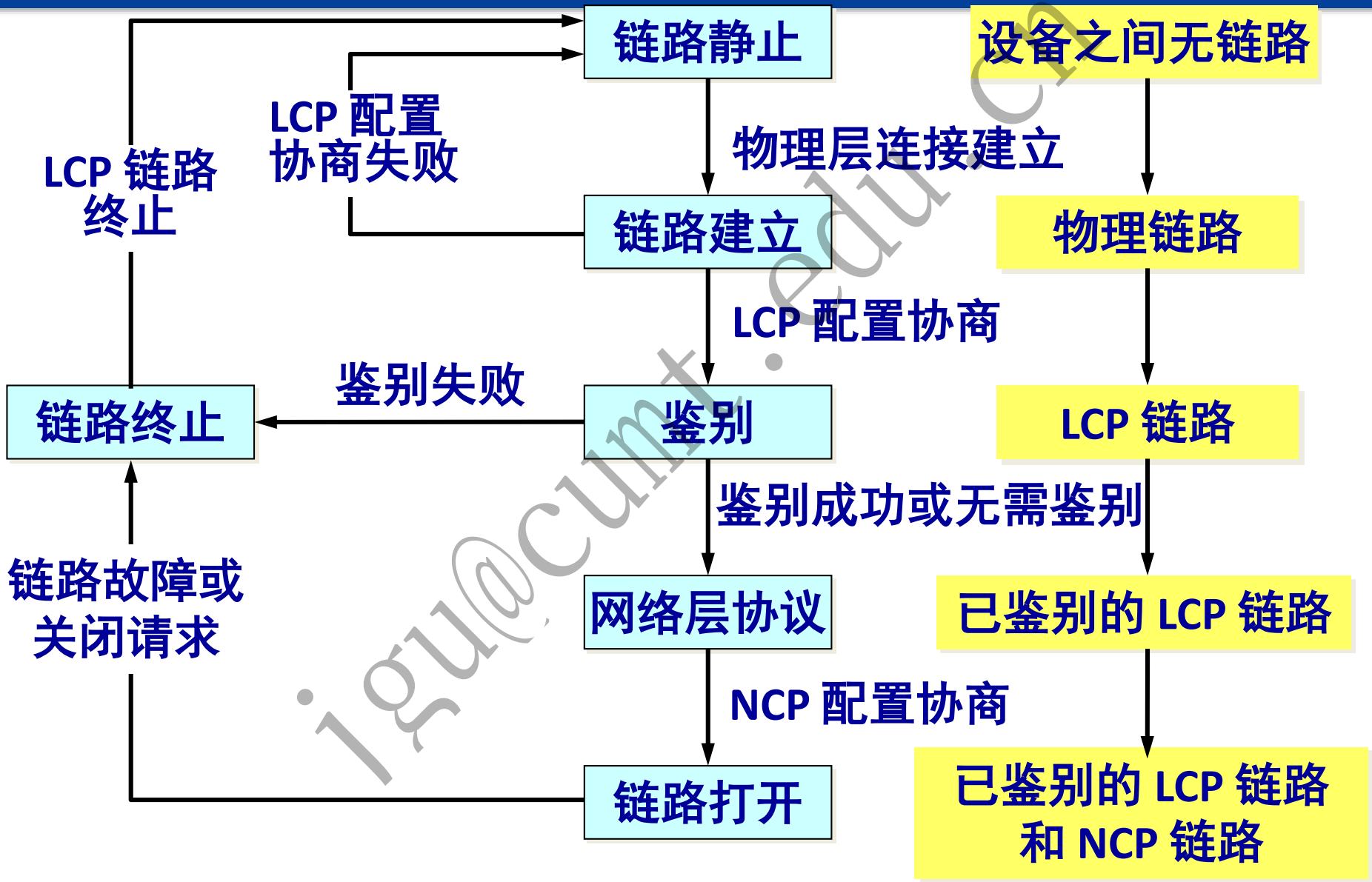

- 如果鉴别成功，则进入网络层协商阶段，否则直接进入终止阶段。
- 如果运行的是IP协议，则使用NCP中支持IP的协议——IP控制协议IPCP为PPP链路的每一端配置IP协议模块（如分配IP地址）。IPCP分组也封装成PPP帧在PPP链路上传送。

# PPP 协议的工作状态




- 当网络层配置完毕后，链路就进入可进行数据通信的“链路打开”状态。链路的两个PPP端点可以彼此向对方发送分组。
- 通信完毕后，NCP 释放网络层连接，收回原来分配出去的 IP 地址。接着，LCP 释放数据链路层连接。最后释放的是物理层的连接。

# PPP 协议的状态图



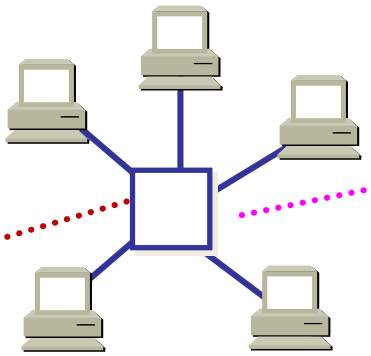
# Q7：什么是局域网？

- 局域网为一个单位所拥有，且地理范围和站点数目均有限。
  - 具有广播功能，从一个站点可很方便地访问全网。
  - 局域网上的主机可共享连接在局域网上的各种硬件和软件资源。
  - 便于系统的扩展和逐渐地演变，各设备的位置可灵活调整和改变。
  - 提高了系统的可靠性、可用性和残存性。

# 局域网拓扑

物理拓扑 → 连接方式

逻辑拓扑 → 工作方式



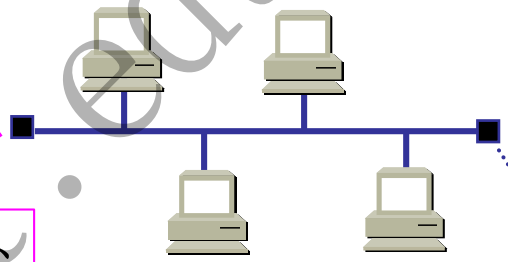
集线器

交换机

?

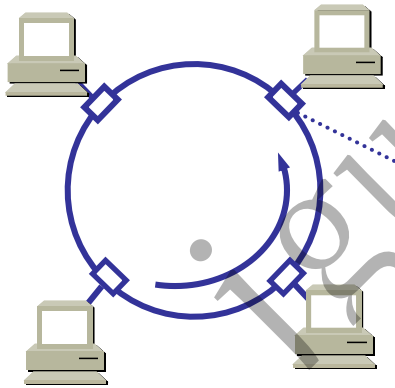
星形网

不同的物理拓扑
相同的逻辑拓扑



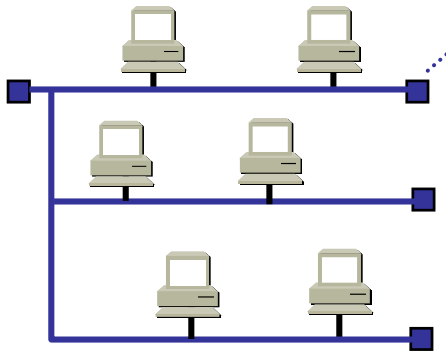
总线网

匹配电阻



干线耦合器

环形网

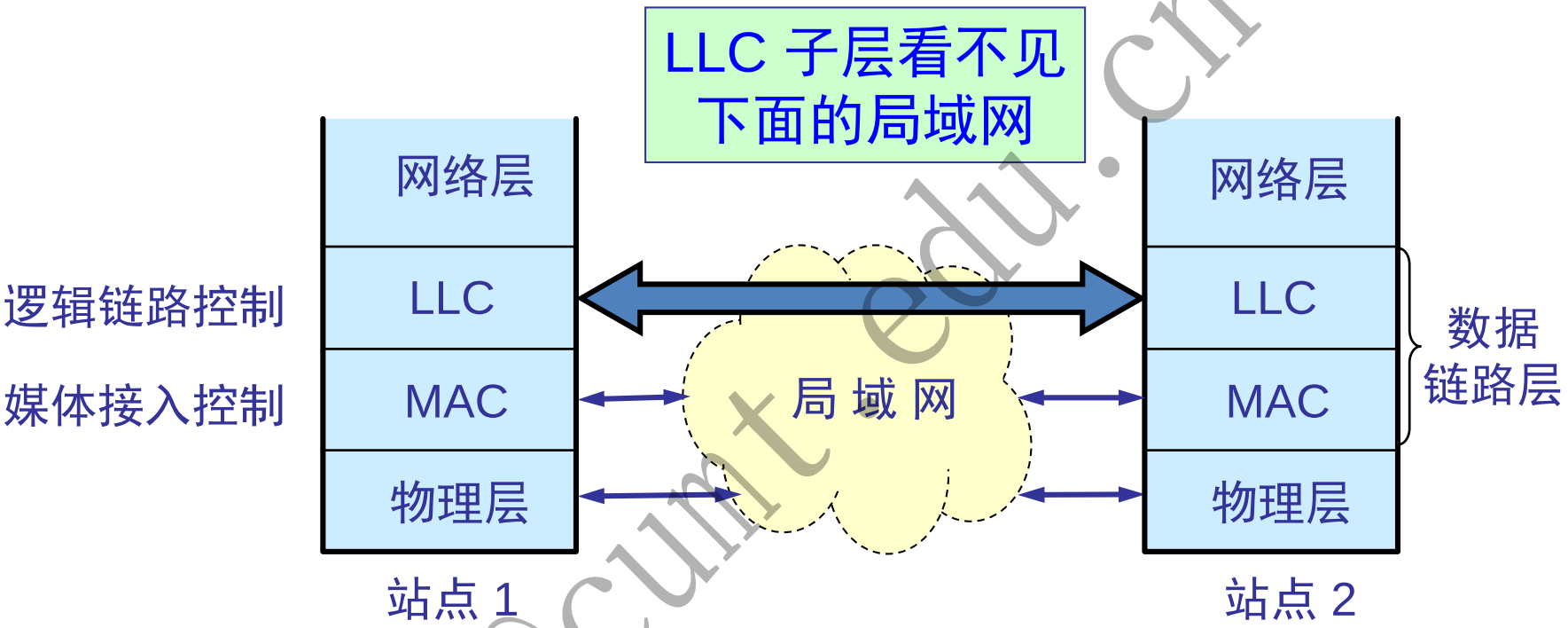


树形网

# 局域网体系结构

- 局域网发展初期，采用不同网络拓扑和媒体接入技术的各种类型网络相继出现。
- 由于有关厂商在商业上的激烈竞争，IEEE 802委员会未能“一统江湖”形成一个“最佳的”局域网标准，而是被迫“求同存异”，设计了一个能更好地适应多种局域网标准的数据链路层。
- 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。

# 局域网对 LLC 子层是透明的




- 与接入到传输媒体有关的内容都放在MAC子层；
- 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。

## OSI参考模型

| |
|---|
| 应　用　层 |
| 表　示　层 |
| 会　话　层 |
| 传　输　层 |
| 网　络　层 |
| 数据链路层 |
| 物　理　层 |

## IEEE 802模型

| |
|---|
| 逻辑链路控制子层 |
| 介质访问控制子层 |
| 物　理　层 |

- 20世纪90年代后，以太网在局域网市场中取得了垄断地位。
- TCP/IP经常使用的局域网是 DIX Ethernet V2 而不是 802.3 标准中的几种局域网，因此 LLC（即 802.2 标准）的作用已经不大了。
- 现在，很多厂商生产的适配器上就仅装有 MAC 协议而没有 LLC 协议。

802.10 可互操作的局域网安全

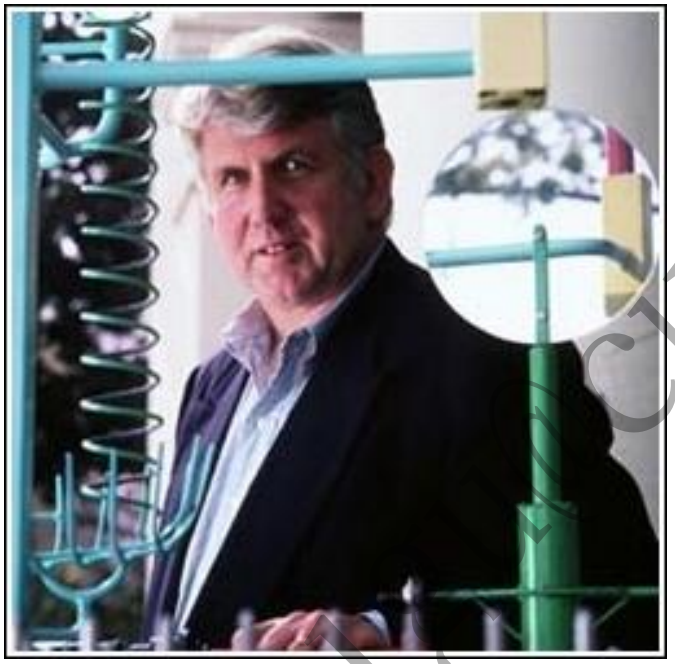
802.1 体系结构与网络互连

802.2 逻辑链路控制子层

| 802.3 CSMA/CD | 802.4 Token Bus | 802.5 Token Ring | 802.6 城域网 | 802.9 语音与数据综合局域网 | 802.11 无线局域网 |
|---|---|---|---|---|---|

# Q8：什么是以太网？

- 70年代末，涌现出了数十种局域网技术，以太网正是其中的一员。

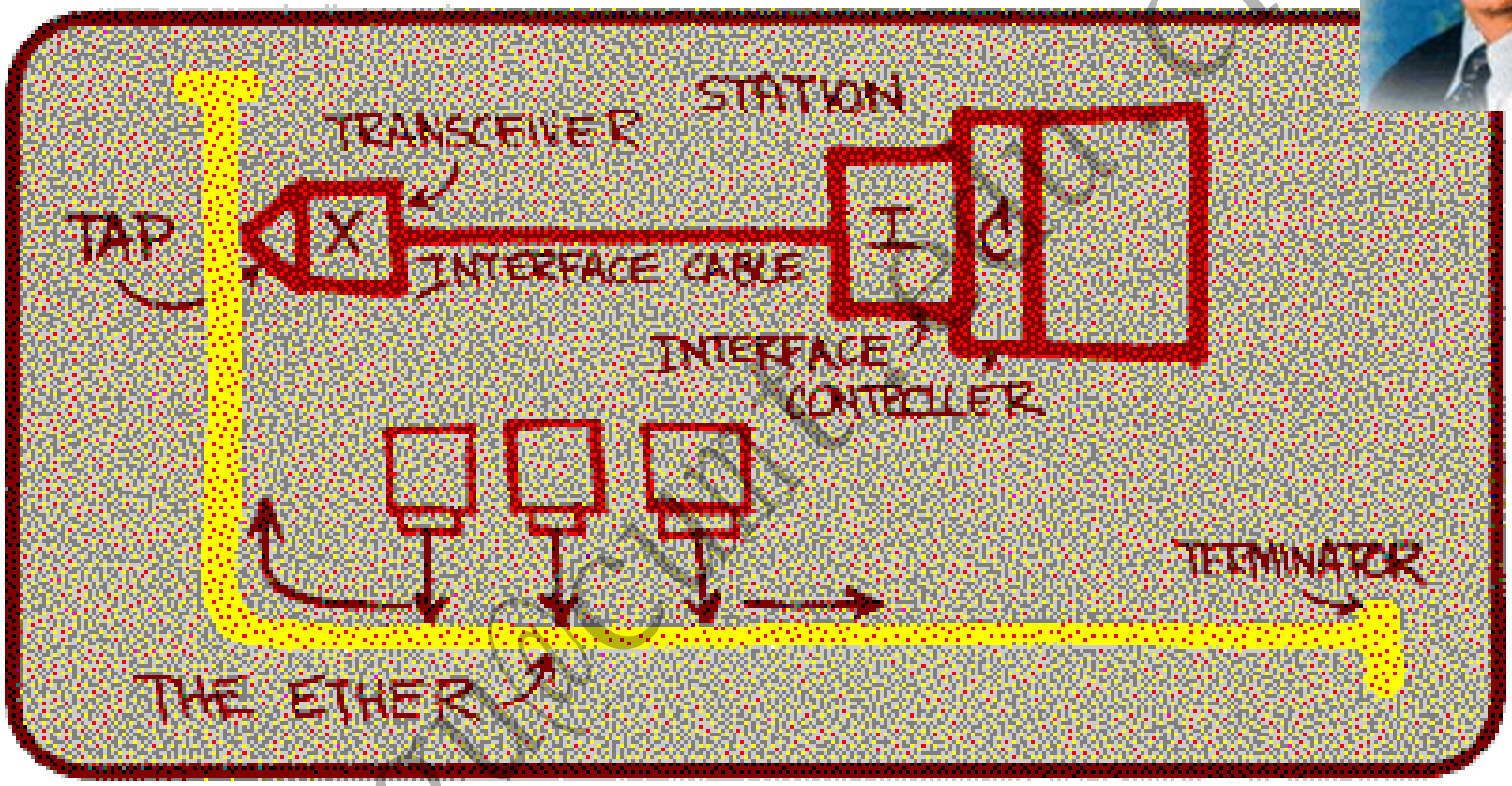


- 鲍勃·麦特卡夫(Bob Metcalfe)既是以太网之父，也是3Com公司创始人，还是一位广受欢迎的专栏作家，一位见多识广的博学者，还以发明著名的网络界第一定律著称。
- 麦特卡夫定律：网络价值同网络用户数量的平方成正比，即N 个联结能创造 N 的2次方效益。

# 以太网的实验原型

- 以太网是在1972年开创的，Bob Metcalfe被Xerox雇佣为网络专家，Palo Alto研究中心(PARC)的第一个任务是把Palo Alto的计算机连接到ARPANet。
  - 1972年底，Bob Metcalfe设计了一套网络，把Alto计算机连接起来。因为该网络是以ALOHA系统（一种无线电网络系统）为基础的，而又连接了众多的ALTO计算机，所以Metcalfe把它命名为ALTO ALOHA网络。
- 1973年5月ALTO ALOHA网络开始运行，Metcalfe正式改名为以太网（Ethernet）。
  - 这就是最初的以太网试验原型，该网络运行的速率为2.94Mbps，网络运行的介质为粗同轴电缆。

# 以太网架构草图




以太网之父鲍勃 麦特卡夫于1973年绘制的以太网草图

# 以太网的诞生

- 1976年6月， Metcalfe和Boggs发表了题为“以太网：局域网的分布型信息包交换”的著名论文。
- 1977年底，Metcalfe和他的三位合作者获得了“具有冲突检测的多点数据通信系统”的专利，多点传输系统被称为CSMA／CD(带冲突检测的载波侦听多路访问)。
- 从此，以太网就正式诞生了。

# DIX以太网标准

- 1979年，Digital Equipment Corporation（DEC）、Intel公司与Xerox公司联盟，促进了以太网标准化。
- 1980年9月30日，DEC、Intel和Xerox公布了著名的以太网蓝皮书——“以太网，一种局域网：数据链路层和物理层规范，1.0版”，也称为DIX(取三家公司名字的第一个字母而组成的)版以太网1.0规范。
- 最初的实验型以太网工作在2.94Mbps，而DIX规范定义的以太网工作在10Mbps。
- 1982年，DIX联盟发布了以太网的第二个版本，即Ethernet II。

# IEEE以太网标准

- DIX联盟虽已推出以太网规范，但还不是国际公认的标准，所以在1981年6月，IEEE 802工程决定组成802.3分委员会，以产生基于DIX工作成果的国际公认标准。
- 1983年6月IEEE 802.3工作组发布了第一个关于以太网技术的IEEE标准，即IEEE 10BASE5。
- 1984年美国联邦政府以 FIPS PUB107的名字采纳802.3标准。
- 1989年ISO以标准号 IS88023采纳802.3以太网标准，至此，IEEE标准802.3正式得到国际上的认可。

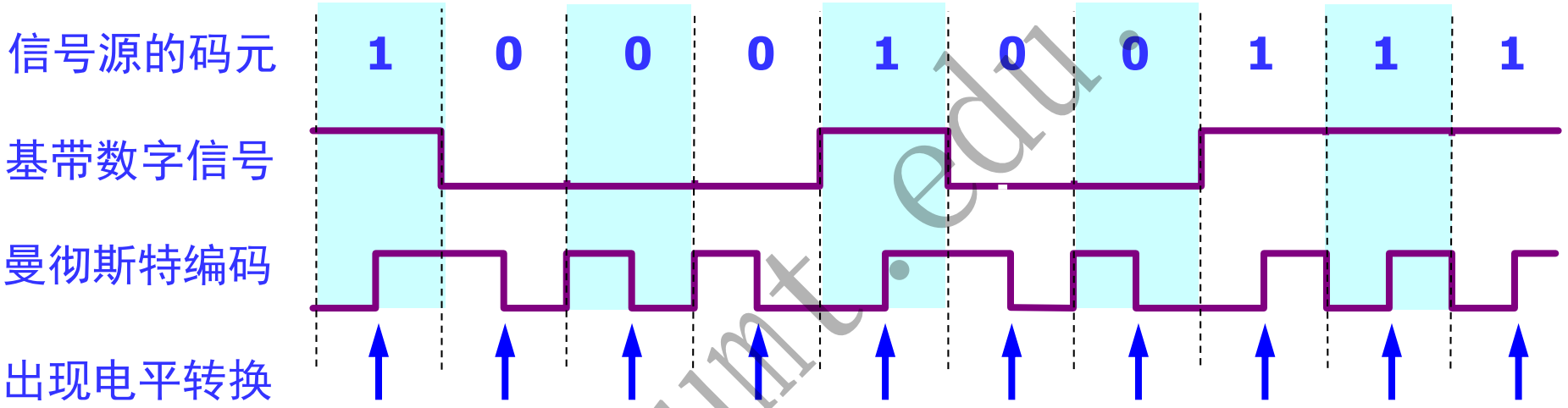
# 以太网的两个标准

- DIX Ethernet V2 标准与 IEEE 802.3 标准只有很小的差别，因此可以将 802.3 局域网简称为“以太网”。
- 严格说来，“以太网”应当是指符合 DIX Ethernet V2 标准的局域网。

# Q9：以太网如何简化通信的实现？

- 一是采用较为灵活的无连接的工作方式
  - 不必先建立连接就可以直接发送数据，对发送的数据帧既不进行编号，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。
  - 差错的纠正由高层来决定。
    - ✓ 以太网提供尽力而为的不可靠交付服务，当目的站收到有差错的数据帧时就丢弃此帧，其他什么也不做。
    - ✓ 如果高层发现丢失了一些数据而进行重传时，以太网并不知道这是一个重传的帧，而是当作一个新的数据帧来发送。

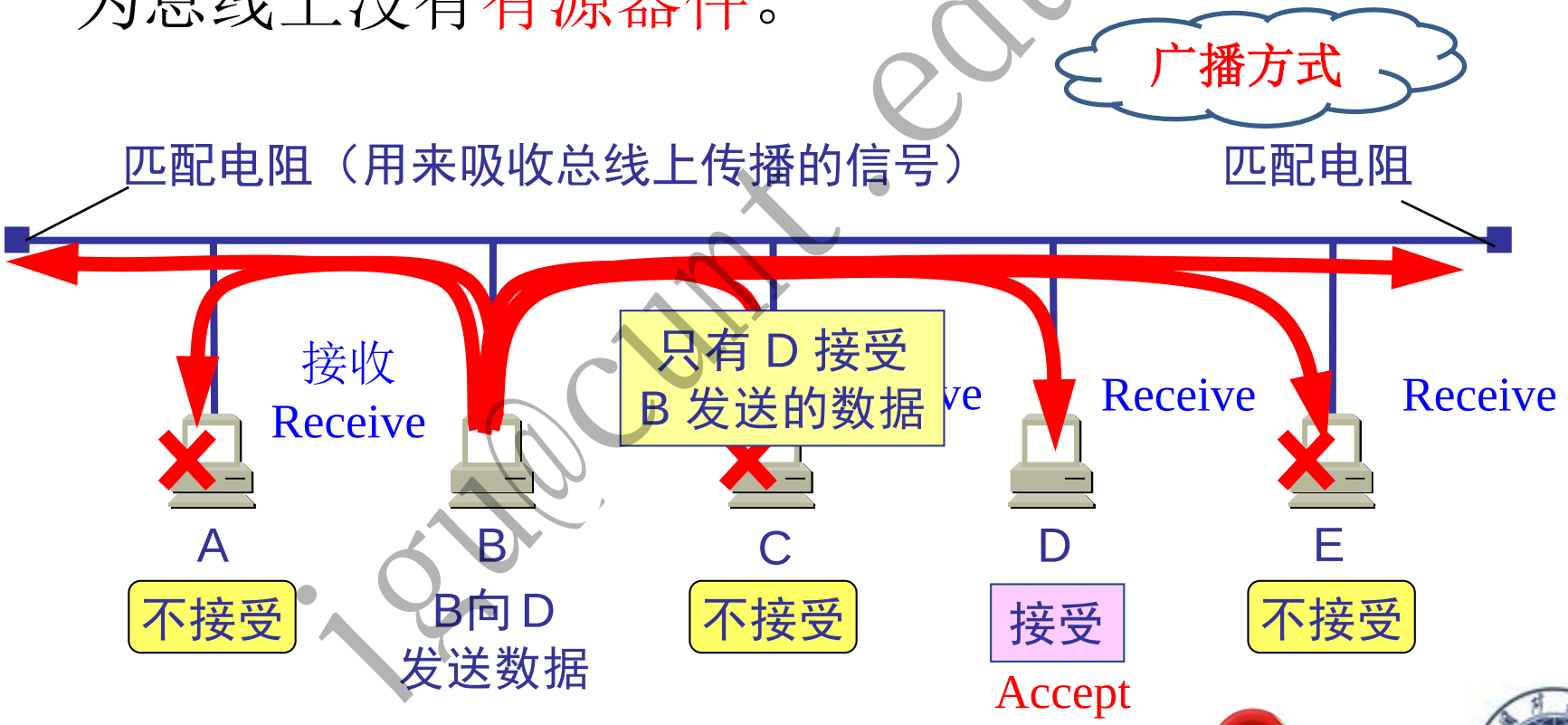
## 二是采用曼彻斯特(Manchester)编码




- 曼彻斯特编码是一种自同步编码方法，接收方利用包含有同步信号的特殊编码从信号自身提取同步信号来锁定自己的时钟脉冲频率，达到同步目的。
- 所占频带宽度比原始基带信号增加一倍，即经过曼彻斯特编码器之后，原来的信号源的每一个码元都变成了两个码元。
- 10Mb/s数据率的以太网的码元速率为每秒20兆个码元。

# Q10：总线结构以太网如何工作？

- 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。

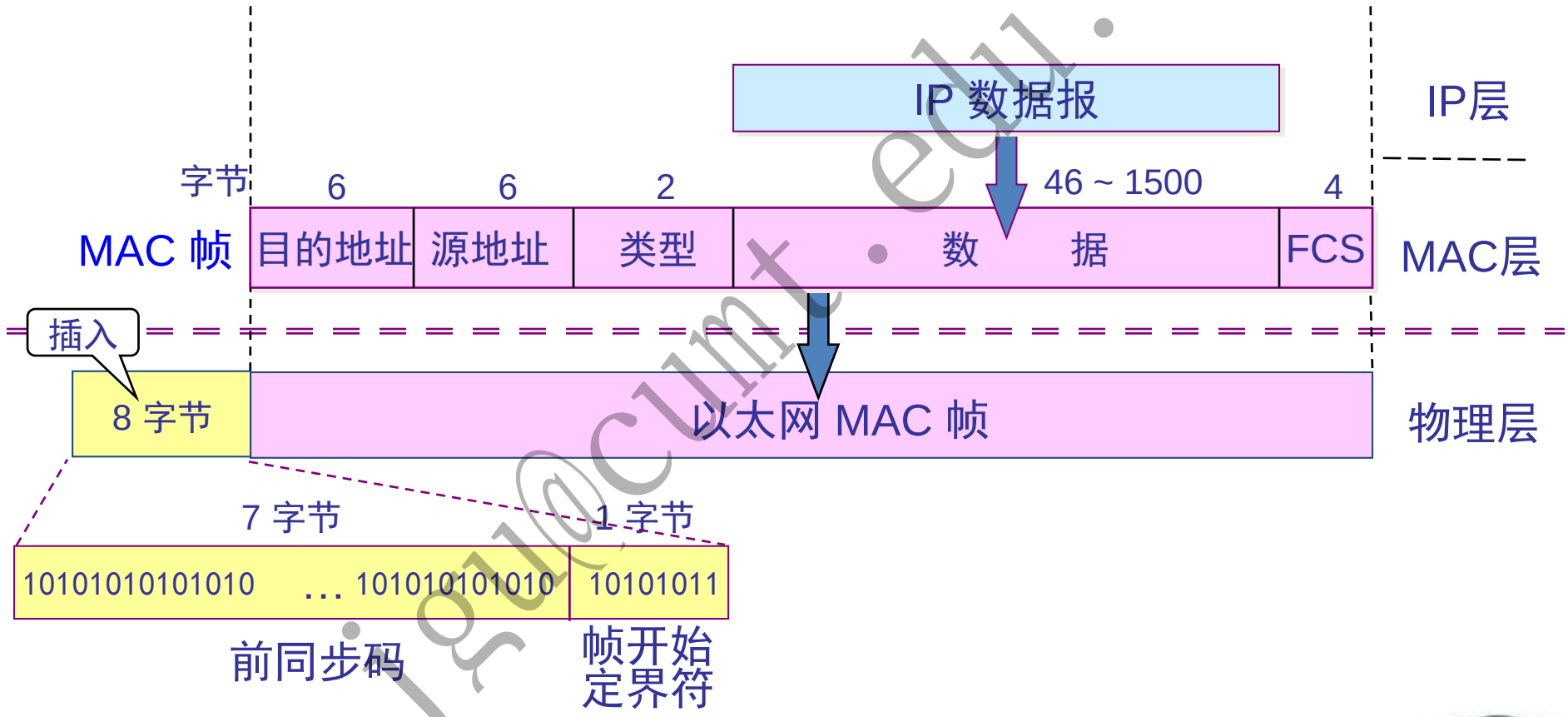



具有广播特性的总线上实现了一对一的通信

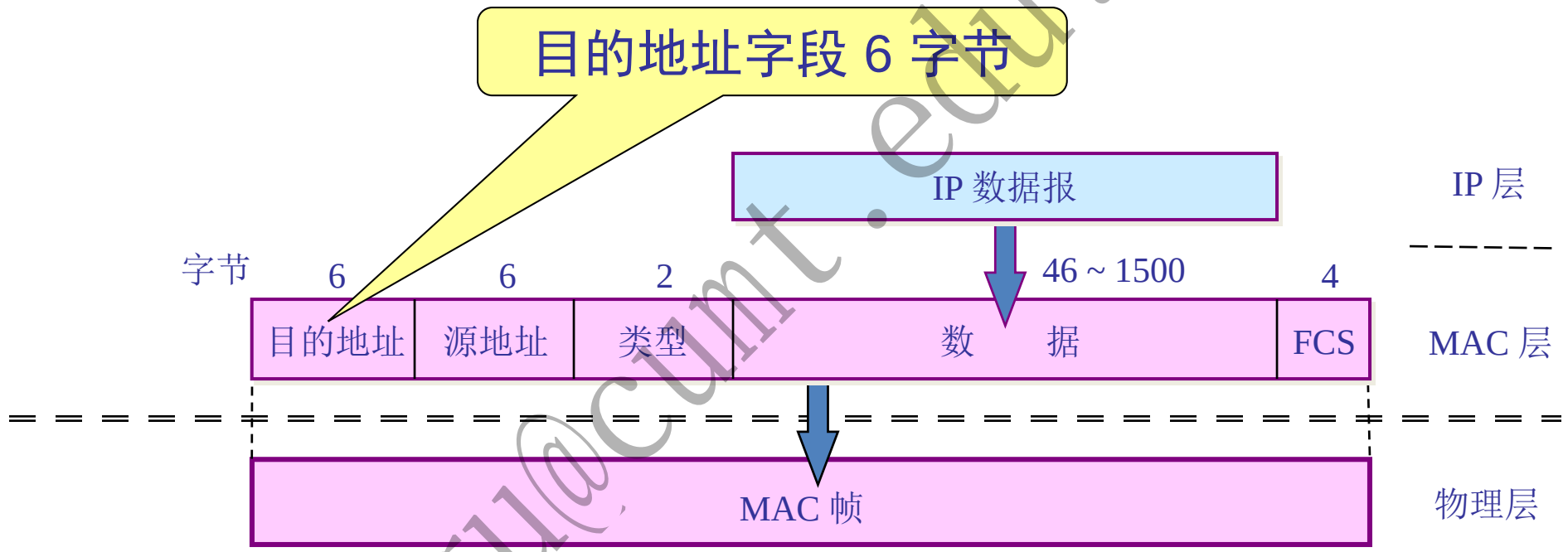
# 以太网的MAC帧

- 常用的以太网MAC帧格式有两种标准：
  - DIX Ethernet V2 标准（最常用的）
  - IEEE 的 802.3 标准

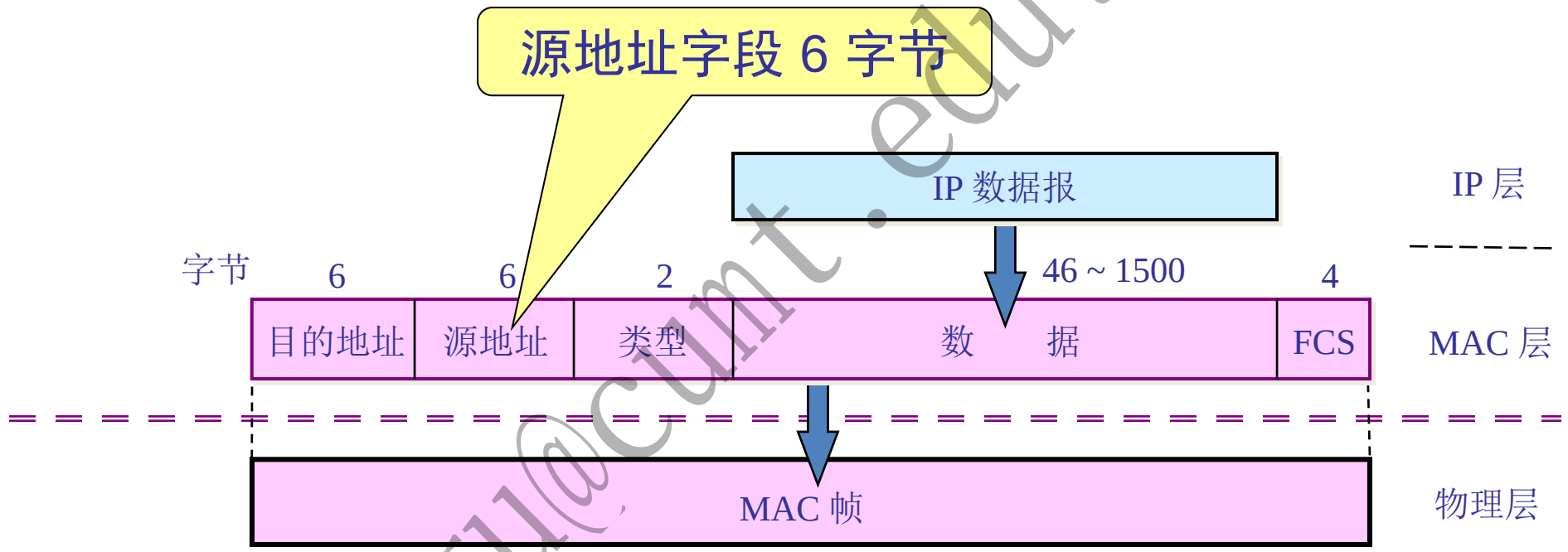
# 以太网V2的 MAC 帧格式



# 以太网 V2 的 MAC 帧格式

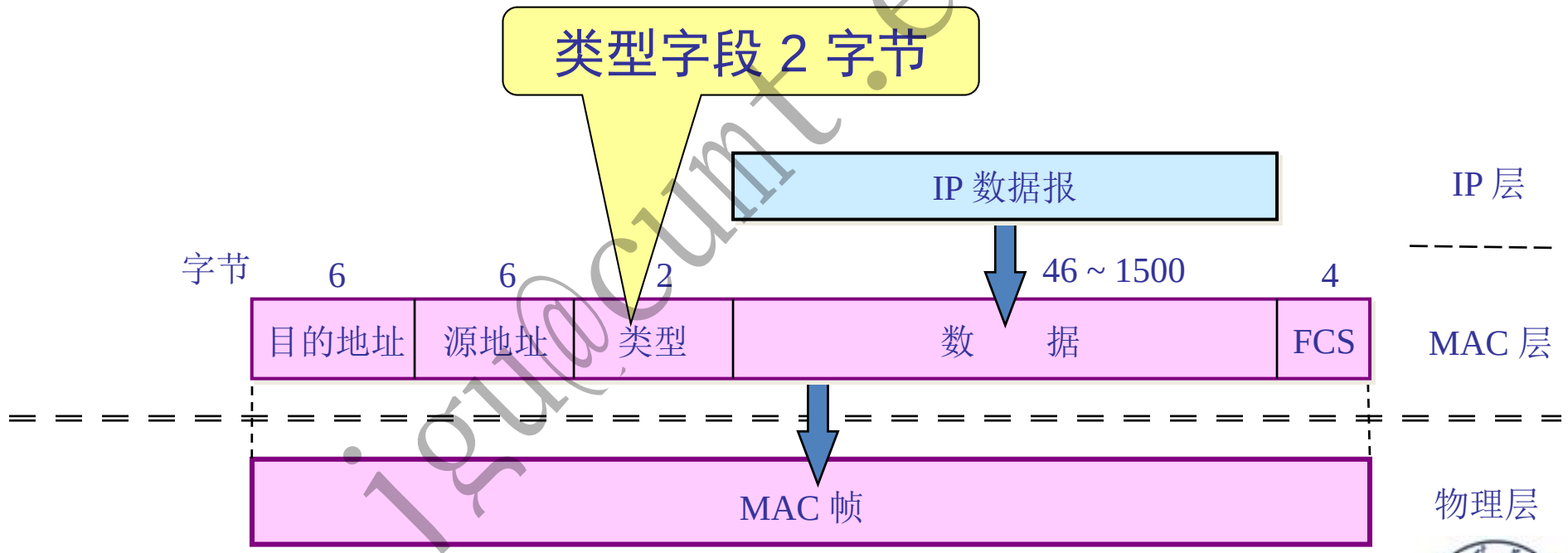


# 以太网 V2 的 MAC 帧格式



# 以太网 V2 的 MAC 帧格式

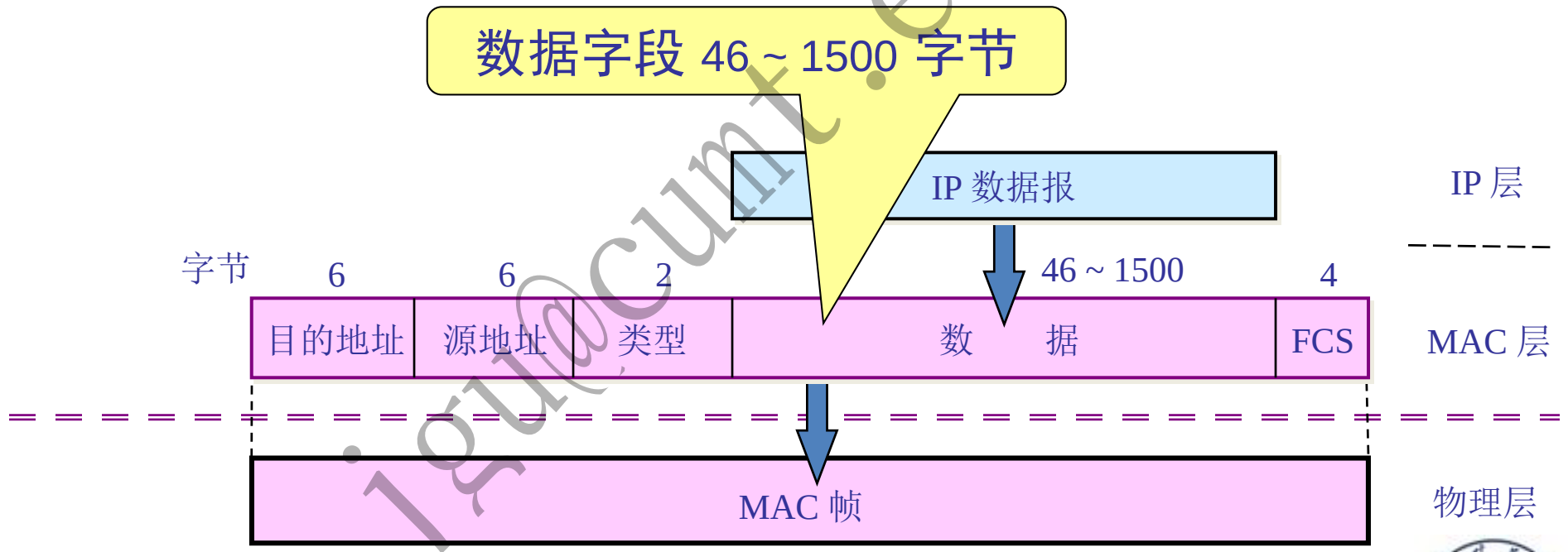
类型字段用来标志上一层使用的是什么协议，
以便把收到的 MAC 帧的数据上交给上一层的这个协议。



# 以太网 V2 的 MAC 帧格式

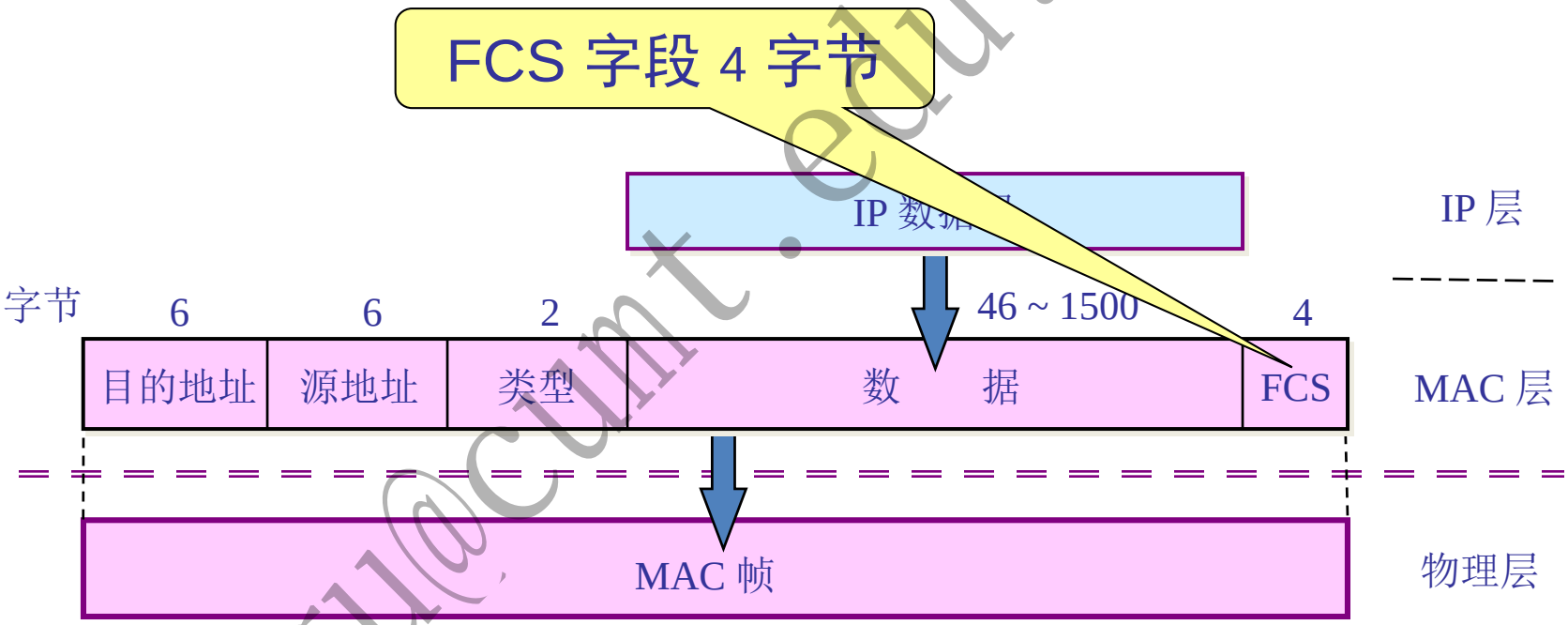
**数据字段的正式名称是 MAC 客户数据字段**

**最小长度 64 字节 – 18 字节的首部和尾部 = 数据字段的最小长度**



# 以太网 V2 的 MAC 帧格式

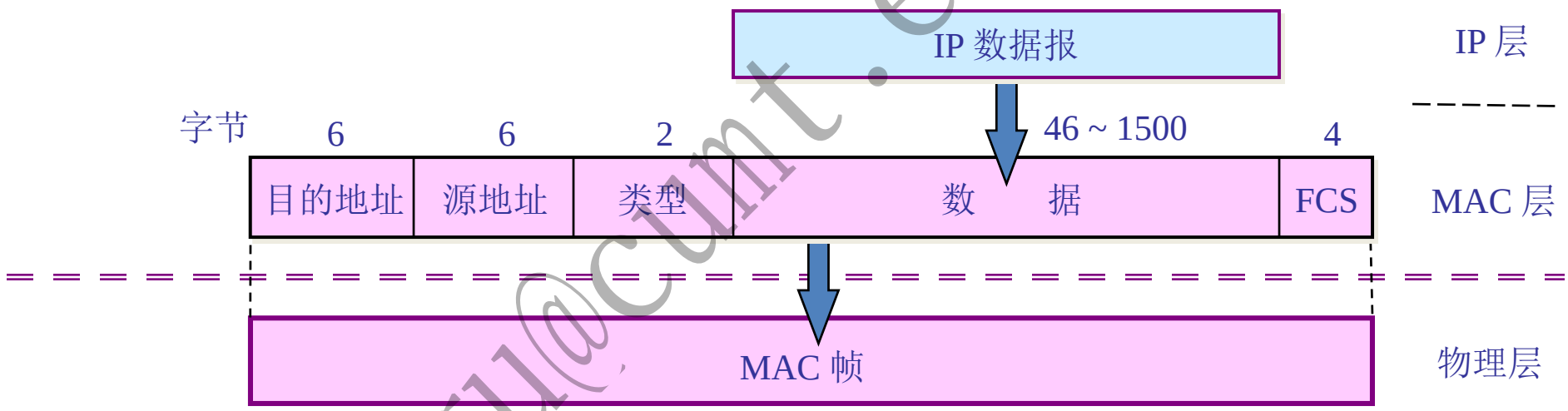
当传输媒体的误码率为 $1\times10^{-8}$ 时，
MAC 子层可使未检测到的差错小于 $1\times10^{-14}$。




FCS帧校验(Frame Check Sequence): 包括4字节循环冗余校检码(CRC)用于检查错误。

# 以太网 V2 的 MAC 帧格式

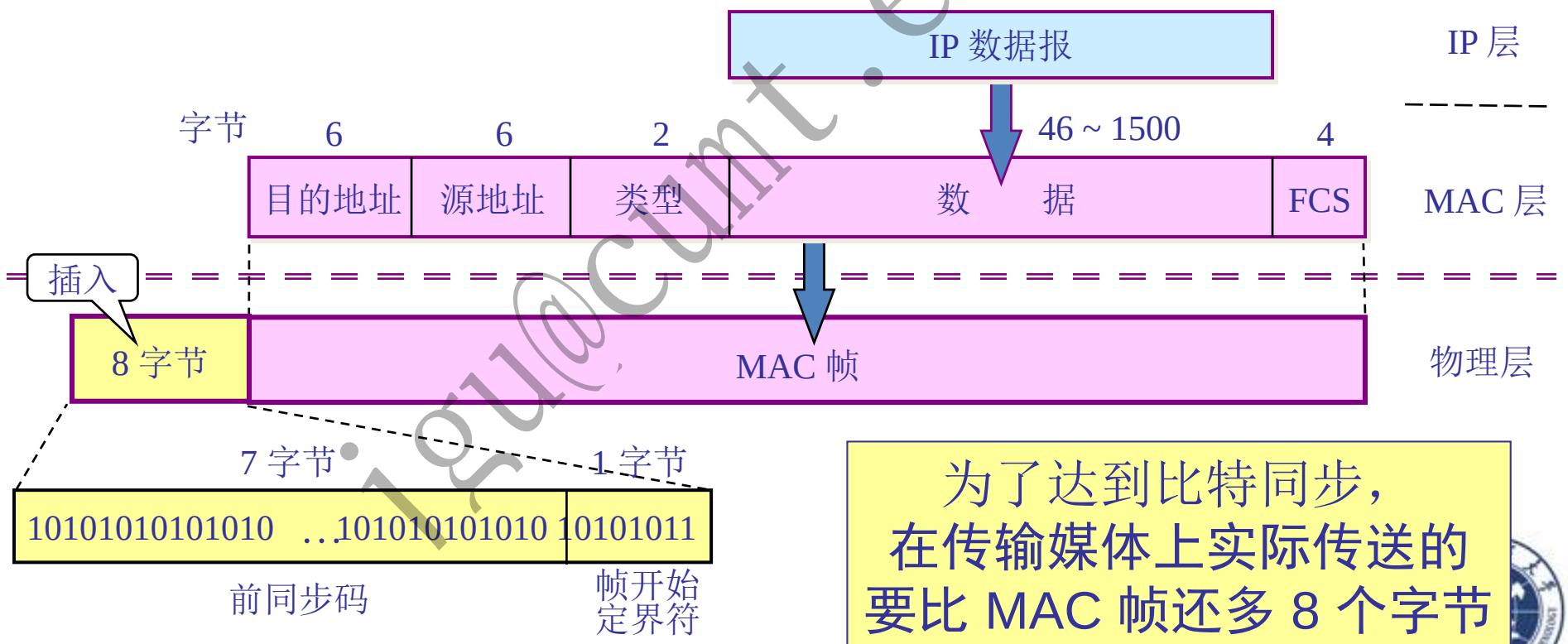
上层协议如何知道填充字段的长度？
上层使用IP协议时，其首部有一个“总长度”字段。
“总长度”+填充字段的长度应当等于MAC帧数据字段的长度。




当数据字段的长度小于 46 字节时，
应在数据字段的后面加入整数字节的填充字段，
以保证以太网的 MAC 帧长不小于 64 字节。

# 以太网 V2 的 MAC 帧格式

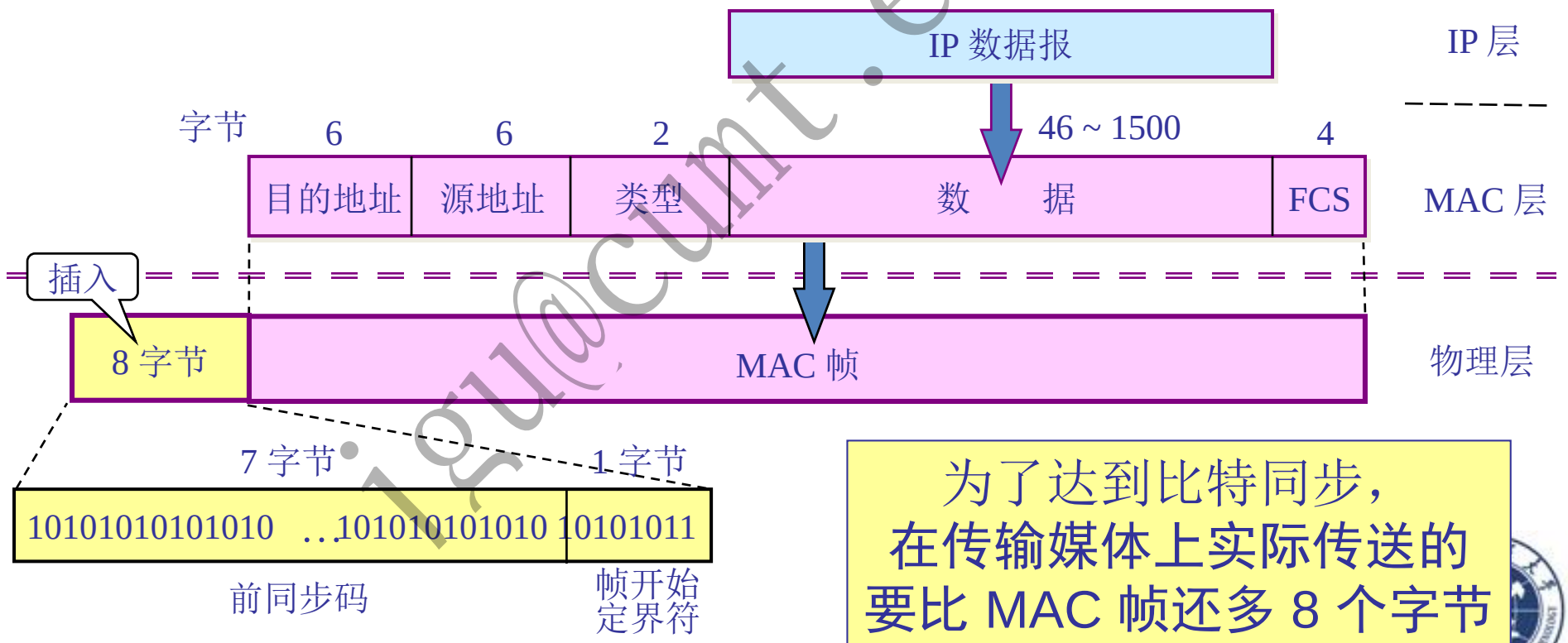
在帧的前面插入（硬件生成）的 8 字节中的第一个字段共 7 个字节，是前同步码(1和0交替码)，使接收端能够迅速调整其时钟频率，使它和发送端的时钟同步，实现 MAC 帧的比特同步。




为了达到比特同步，
在传输媒体上实际传送的
要比 MAC 帧还多 8 个字节

# 以太网 V2 的 MAC 帧格式

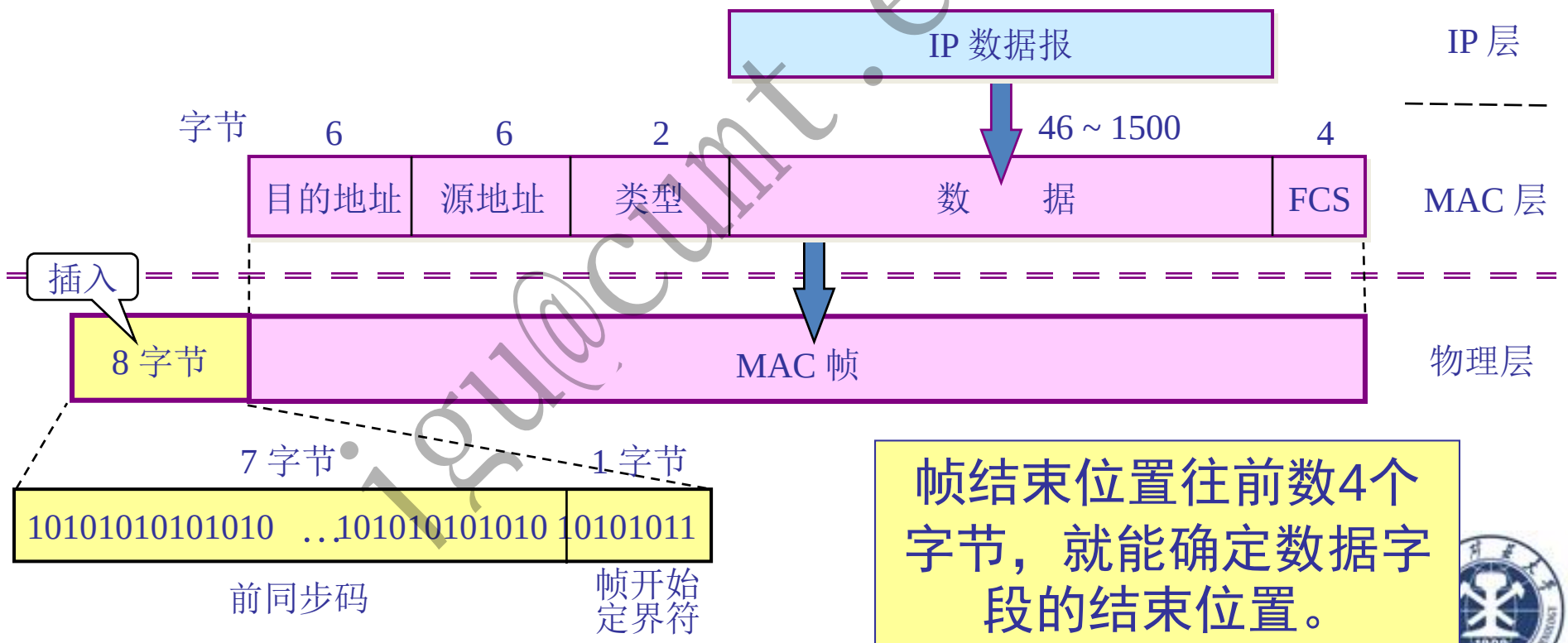
第二个字段(1个字节)是帧开始定界符，定义为10101011，表示后面的信息就是MAC 帧。
最后两个连续的1就是告诉接收端适配器注意接收。
FCS的检验范围不包括前同步码和帧开始定界符。




为了达到比特同步，
在传输媒体上实际传送的
要比 MAC 帧还多 8 个字节

# 以太网 V2 的 MAC 帧格式

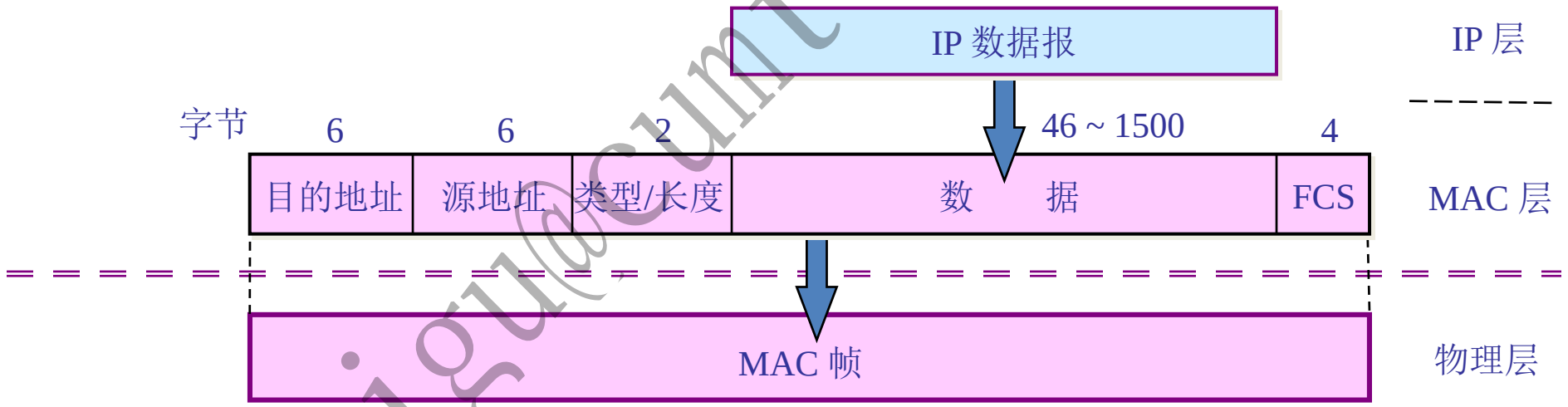
V2的MAC帧格式中没有一个帧长度（或数据长度）字段。
曼彻斯特编码的每一个码元的正中间一定有一个电压的转换，当一个帧发送完毕后，就不再发送其它码元了，发送方NIC接口上的电压也就不再发生变化了，接收方就找到了帧结束位置。




帧结束位置往前数4个字节，就能确定数据字段的结束位置。

# IEEE 802.3 MAC 帧格式

- (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”。
- (2) 当“长度/类型”字段值小于 0x0600 时，数据字段必须装入上面的逻辑链路控制 LLC 子层的 LLC 帧。

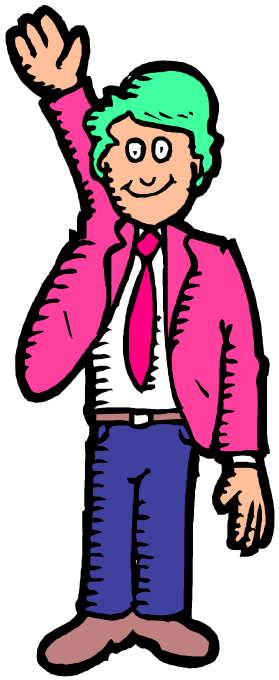



现在市场上流行的都是以太网V2 的 MAC 帧，但大家也常常把它称为 IEEE 802.3 标准的 MAC 帧。

# 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 46 ~ 1500 字节之间。
- 有效的 MAC 帧长度为 64 ~ 1518 字节之间。

**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。**



# THANK YOU!