



计算机与通信工程学院
School of Computer & Communication Engineering

第4章 数据链路层



4.1	数据链路层概述
4.2	数据链路层的三个基本问题
4.3	点对点信道的数据链路层协议
4.4	广播信道的数据链路层协议
4.5	扩展局域网
4.6	高速局域网
4.7	地址解析协议

计算机网络体系结构

OSI 的体系结构



(a)

(a) OSI 的七层协议

TCP/IP 的体系结构



(b)

(b) TCP/IP 的四层协议

五层协议的体系结构



(c)

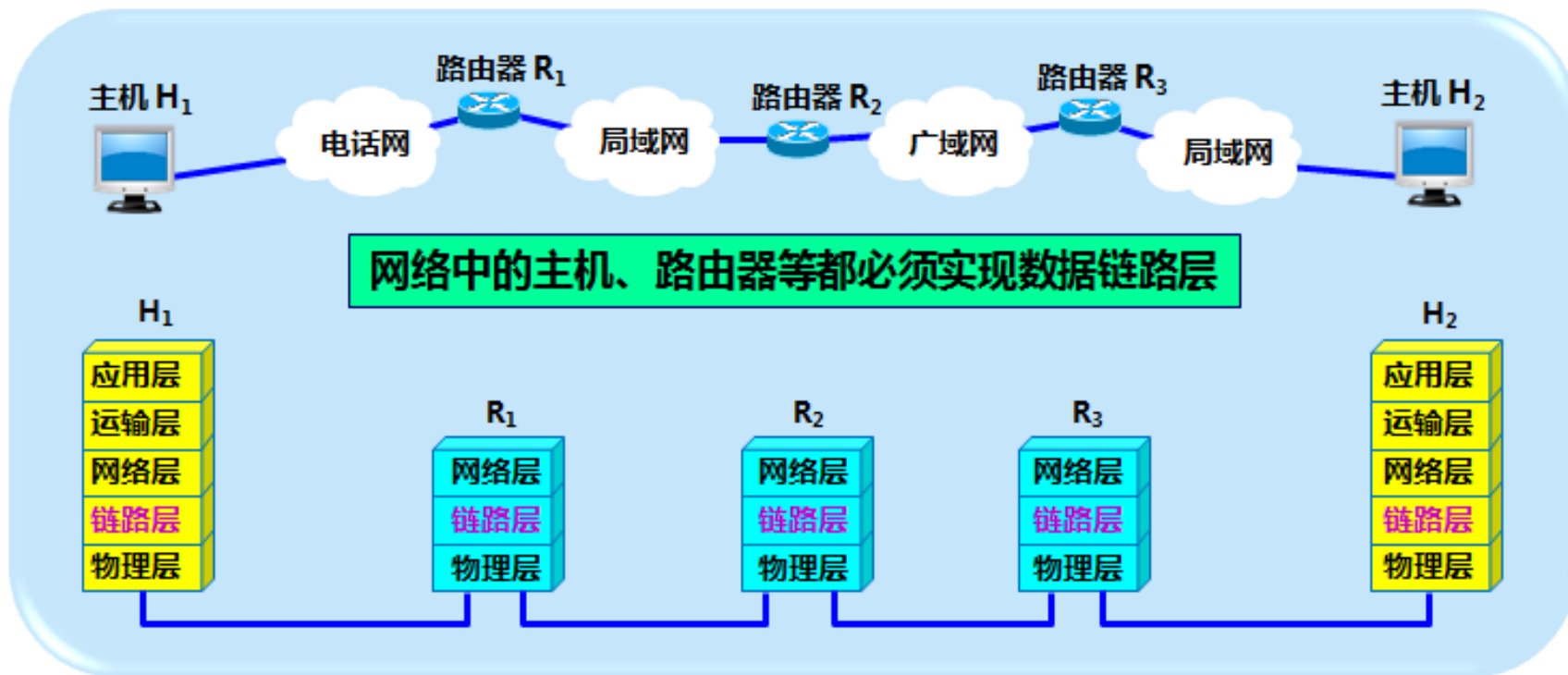
(c) 五层协议

为什么需要数据链路层？

- **网络节点**：主机、路由器等
- **物理层解决的问题**
解决了相邻节点透明传输比特的问题
- **物理层未解决的问题**
 - **传输错误问题**：发送端发送比特1，而接收端收到比特0，接收端无法知道收到的比特是否正确。
 - **谁接收的问题**：多个设备连接，谁能发送数据？数据发给谁？谁负责接收和处理？
 - **传输结束问题**：如何知道一组数据即将到来、何时结束？

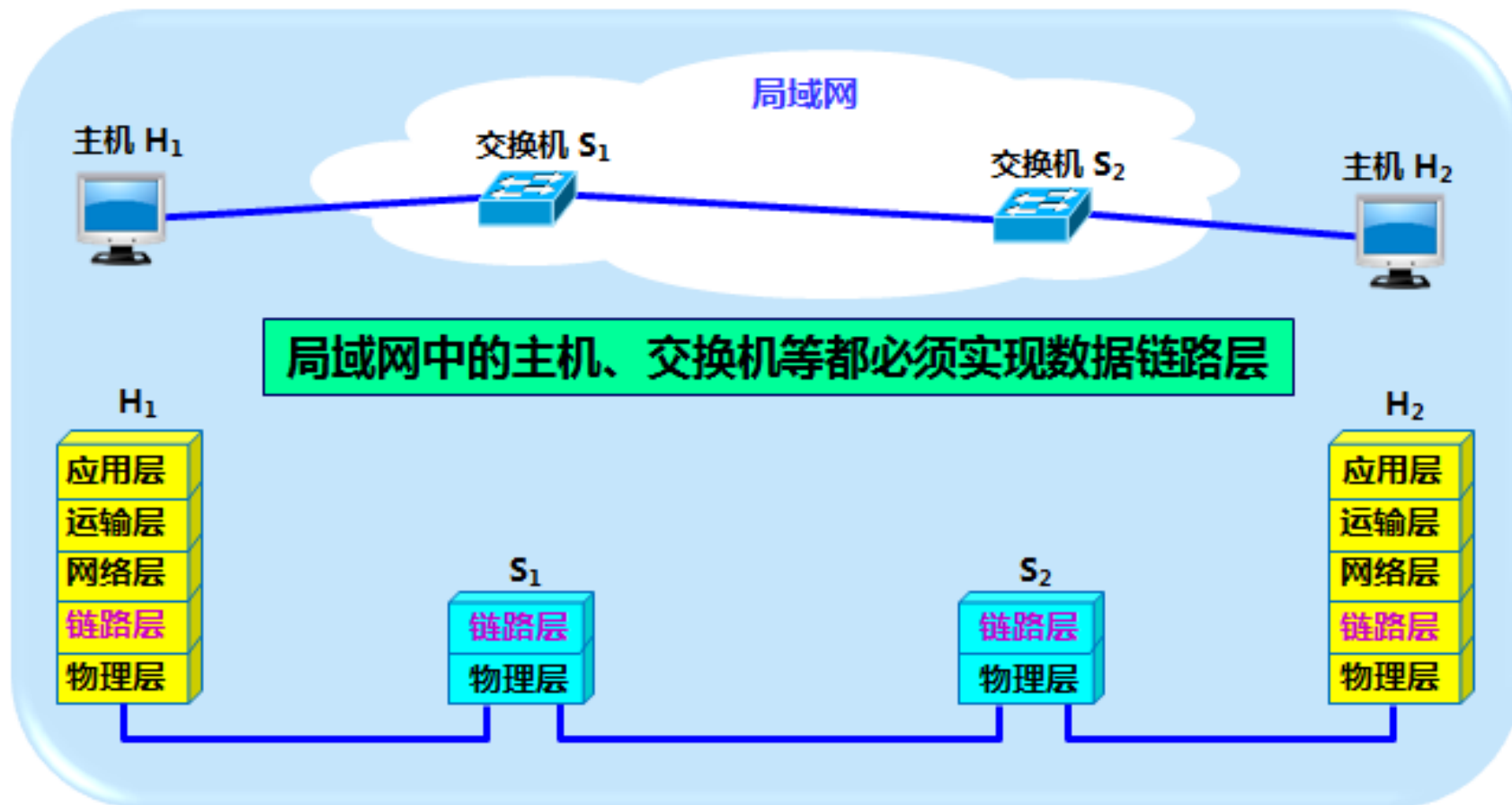
数据链路层要解决上述问题，从而向网络层提供透明、可靠的数据传输服务。

数据链路层的地位



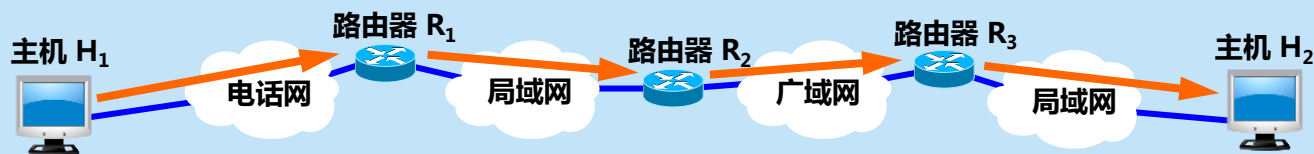
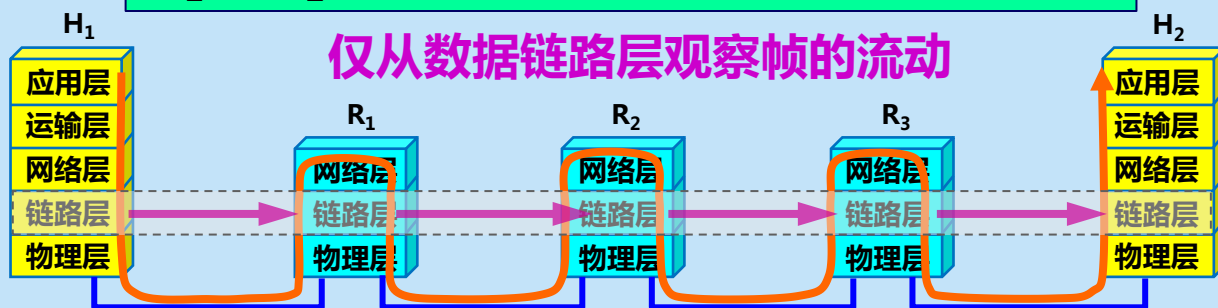
数据链路层是实现设备之间通信的非常重要的一层

数据链路层的地位



数据链路层是实现设备之间通信的非常重要的一层

数据链路层的作用

主机 H_1 向 H_2 发送数据 H_1 到 H_2 所经过的网络可以是多种不同类型的

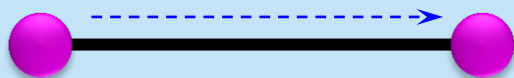
仅从数据链路层观察帧的流动

数据链路层的地位

注意：不同的链路层可能采用不同的数据链路层协议

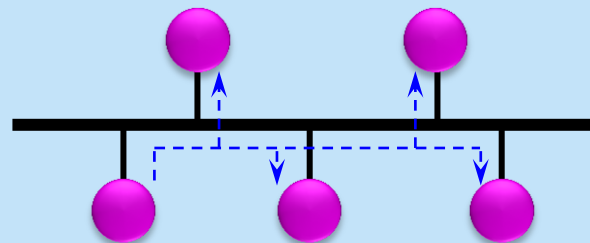
数据链路层实现是相邻节点之间的通信，为网络层实现的主机之间的通信提供了支持。

数据链路层信道类型



(a) 点对点信道

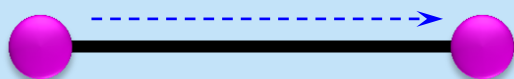
- 使用**一对一**的**点对点**通信方式。



(b) 广播信道

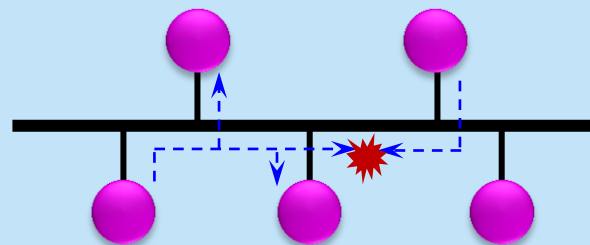
- 使用**一对多**的**广播**通信方式。

数据链路层信道类型



(a) 点对点信道

- 使用**一对一**的**点对点**通信方式。

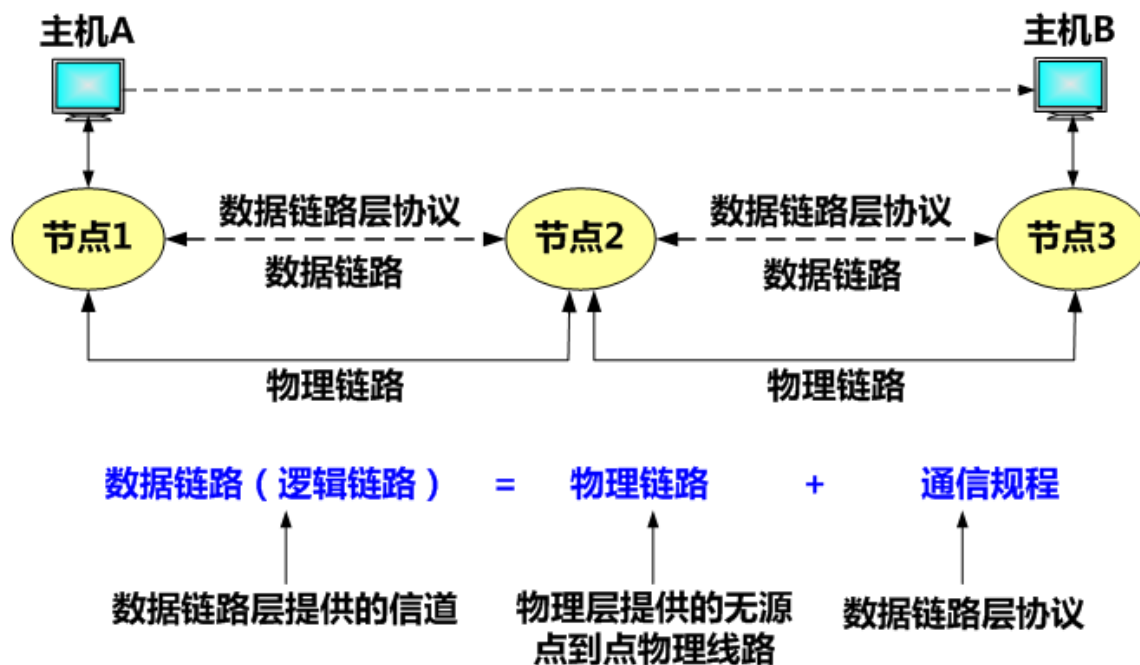


(b) 广播信道

- 使用**一对多**的**广播**通信方式。
- 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。

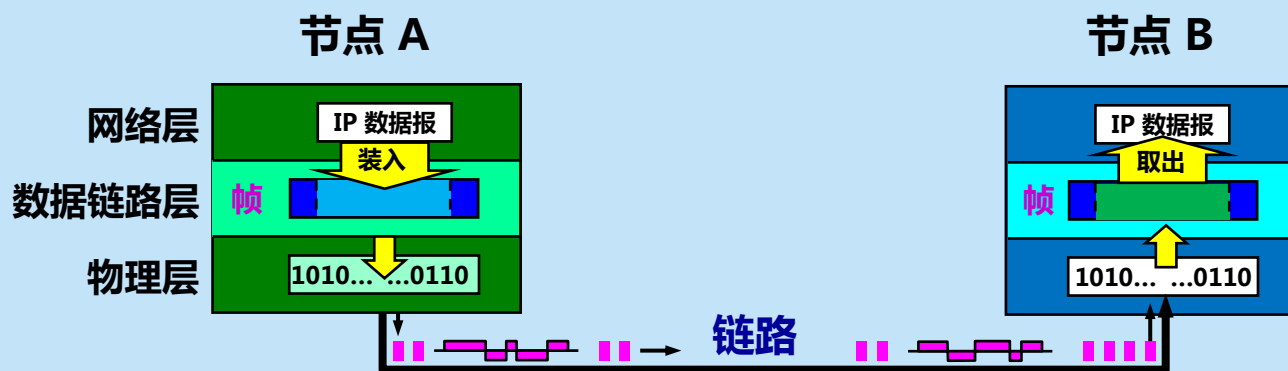
数据链路层的基本术语

- **物理链路**：简称**链路**，是相邻两个节点（中间没有任何其他的交换节点）之间的一条无源的点到点的物理线路。**一条链路只是一条通路的一个组成部分。**
- **数据链路**：链路加上实现通信协议的硬件和软件就构成了数据链路。
 - **逻辑链路**
 - **典型实现——适配器（即网卡）**：一般包括了**数据链路层**和**物理层**这两层的功能。



数据链路层的基本术语

- **帧**：数据链路层的协议数据单元。

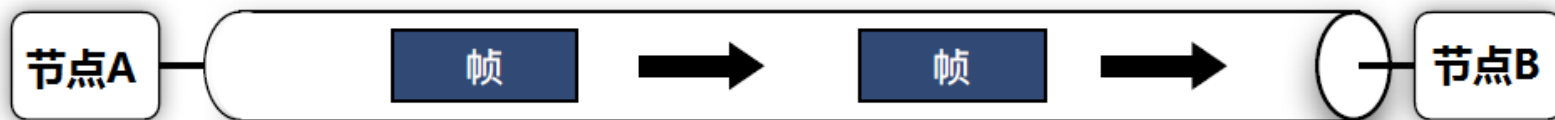
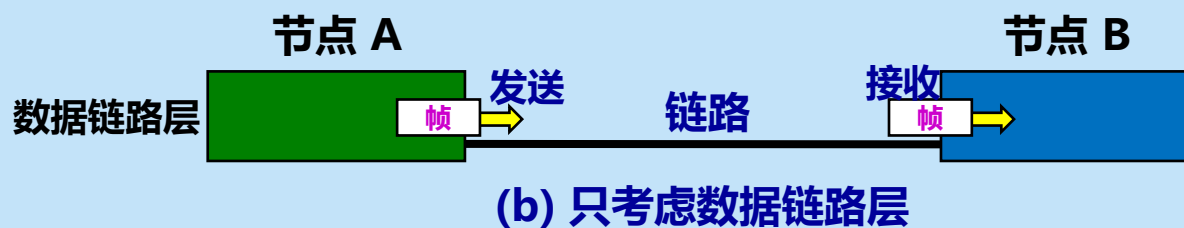


(a) 三层的简化模型

- A的数据链路层把网络层交下来的IP数据报**添加首部和尾部封装成帧**。
- A把封装好的帧**发送给B的数据链路层**。
- B的数据链路层收到的帧无差错，从帧中**取出IP数据报交给网络层**；否则丢弃这个帧。

数据链路层的基本术语

- **帧**：数据链路层的协议数据单元。



- 数据链路层不必考虑物理层如何实现**比特传输的细节**。
- 甚至还可以简单地设想好像是沿着两个数据链路层之间的水平方向把**帧直接发送到对方**。

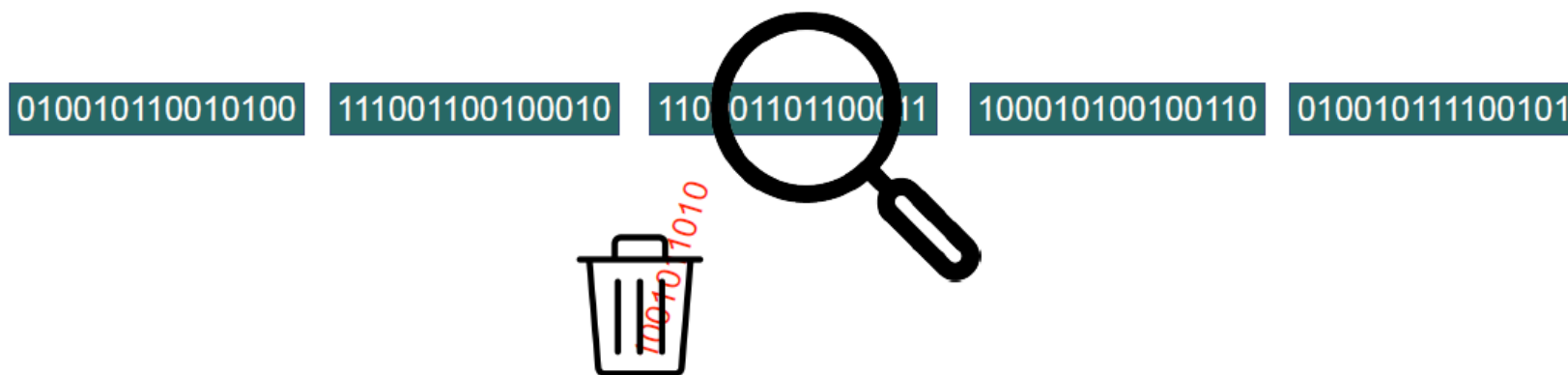
数据链路层的主要功能

- **数据链路的建立、拆除和管理**。提供识别和寻址一个特别的发送端或接收端的手段，该发送端或接收端可能是多点连接设备中的一个，即在两个实体之间如何建立数据链路、如何拆除数据链路和如何对数据链路进行管理。
- **封装成帧**。把网络层的数据分割成数据块并组织成帧，且帧的起始和终止都要有明确的标志。
- **透明传输**。不管链路上传输的是何种形式的比特组合，都能够按照原样无差错地通过数据链路层。
- **差错控制**。提供检错和纠错机制。
- **流量控制**。控制相邻两个节点之间数据链路上的流量。

数据链路和帧

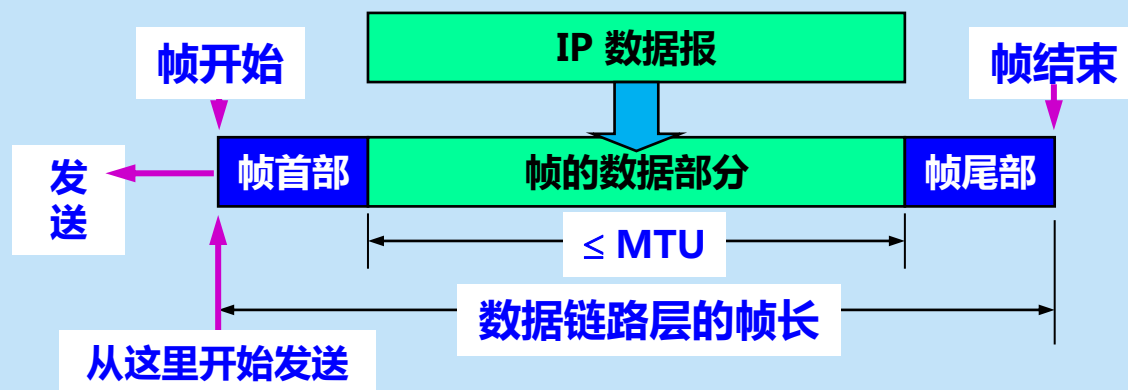
- 数据链路层的协议有很多种，但不论哪种协议，都必须解决以下三个基本问题：

- 封装成帧
- 透明传输
- 差错检测



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。
- 首部和尾部的一个重要作用就是进行**帧定界**（即确定帧的界限）。



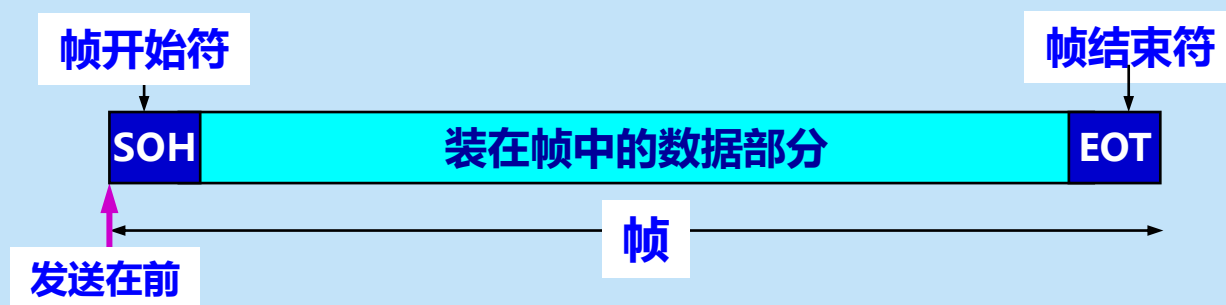
用帧首部和帧尾部封装成帧

最大传送单元MTU (Maximum Transfer Unit) : 规定了所能传送的帧的数据部分长度上限。

封装成帧

用控制字符作为帧定界符

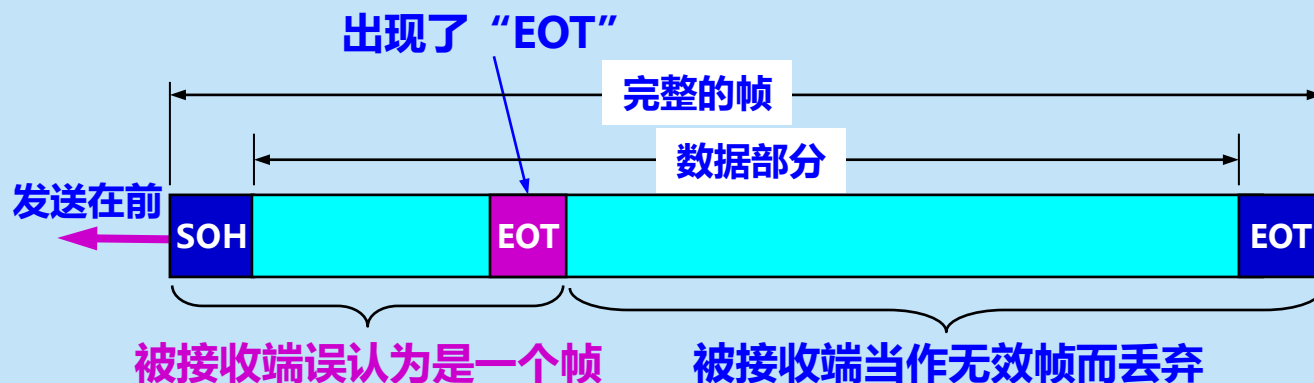
- 当数据是由可打印的 ASCII 码组成的文本文件时，帧定界可以使用特殊的**帧定界符**。
- 控制字符 SOH (Start Of Header) 放在一帧的最前面，表示帧的首部开始。
- 控制字符 EOT (End Of Transmission) 表示帧的结束。



用控制字符进行帧定界的方法举例

透明传输

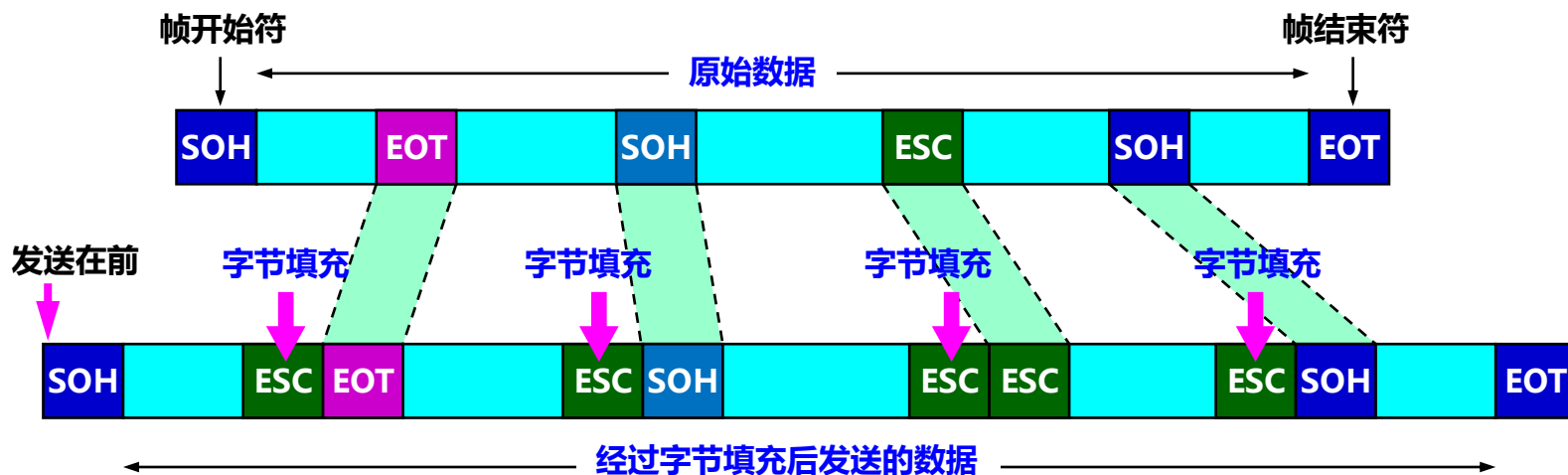
- **透明**是指某一个实际存在的事物看起来却好像不存在一样。
- **在数据链路层透明传送数据**，表示无论发送什么样的比特组合的数据，这些数据都能够按照原样没有差错地通过这个数据链路层。
- 如果数据中的某个字节的二进制代码恰好和 SOH 或 EOT 一样，数据链路层就会**错误地“找到帧的边界”**。



数据部分恰好出现与 EOT 一样的代码

透明传输

- 解决方法：**字节填充**(byte stuffing)或**字符填充**(character stuffing)。
- 发送端的数据链路层在数据中出现控制字符“SOH”或“EOT”的前面插入一个转义字符“ESC” (其十六进制编码是1B)。
- 接收端的数据链路层在将数据送往网络层之前删除插入的转义字符。如果转义字符也出现在数据当中，那么应在转义字符前面插入一个转义字符 ESC。当接收端收到连续的两个转义字符时，就删除其中前面的一个。



差错检测

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。

发送方

0	0	0	1	1	0	1	1
---	---	---	---	---	---	---	---



接收方

0	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---

一位比特错

发送方

0	0	0	1	0	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---



接收方

0	0	0	0	1	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---

多位比特错

- 通常用**误比特率** (BER, Bit Error Rate)或**误码率**来衡量通信链路的质量。
在一段时间内传输错误的比特占所传输比特总数的比率
- 降低误比特率的措施
 - 误比特率与信噪比有很大的关系。
 - 采用各种差错检测、差错纠正等差错控制措施。
 - 数据链路层传送的帧，广泛使用**循环冗余检验 (CRC)** 的检错技术。