同济大学
TONGJI UNIVERSITY

NaMI
网络与机器智能实验室

计算机网络

第三章 数据链路层

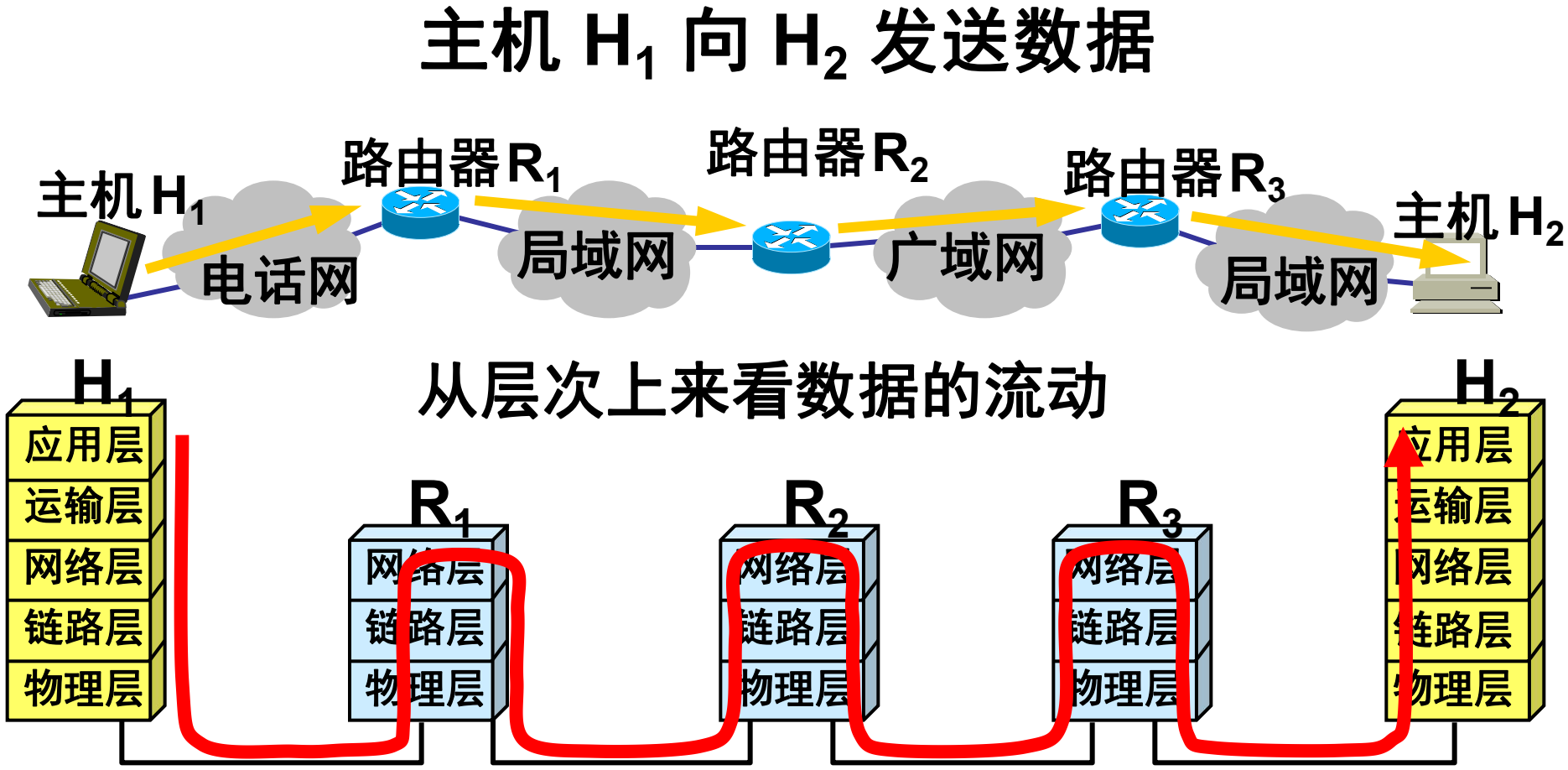
数据链路层简介

- 数据链路层于物理层之上。
- 物理层传输的是比特流，单位是比特。
- 数据链路层传输单位是帧，一帧一帧地处理。
- 其工作是从网络层获取IP包，进行封装成帧、并对帧进行一系列，如透明传输、差错控制等处理，从而传递给下一层（物理层）进行比特流传输。同时也对获得的物理层比特流进行处理，将数据递交给网络层。

数据链路层简介

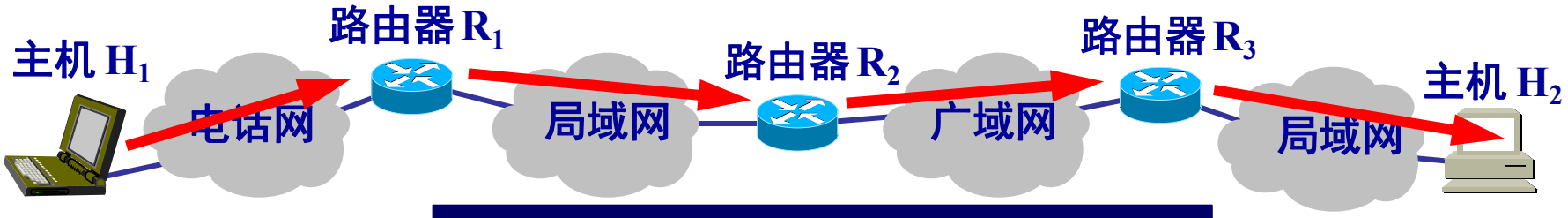
不同的网络类型，有不同的通信机制（即数据链路层协议），数据包在传输过程中通过不同类型的网络，就要使用该网络通信使用的协议，同时数据包也要重新封装成该网络的帧格式。

数据链路层的简单模型

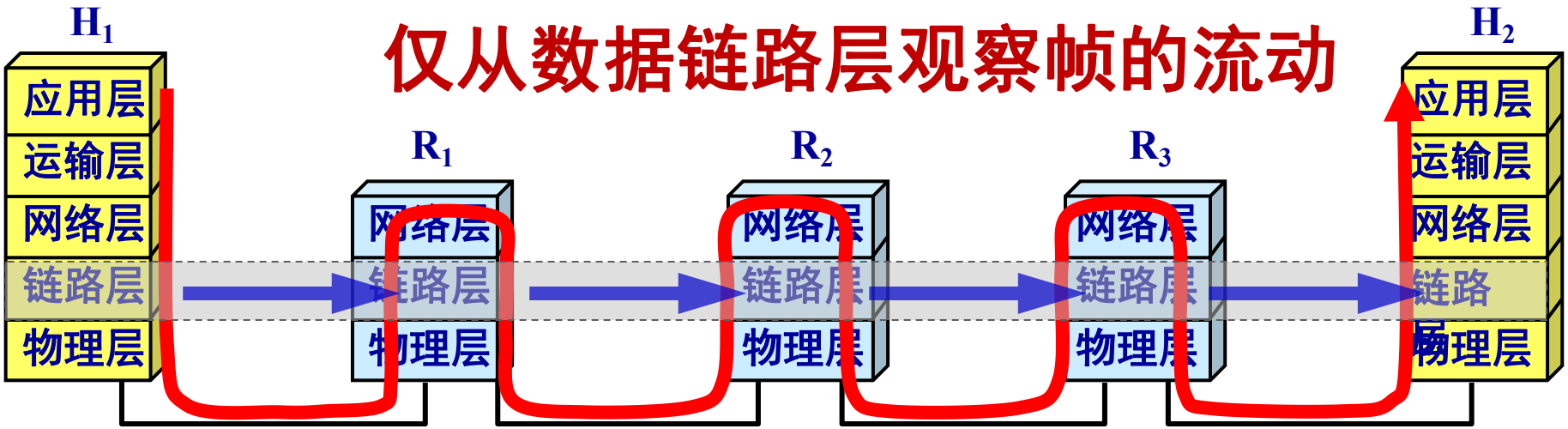

主机 H_1 向 H_2 发送数据
主机 H_1
路由器 R_1
路由器 R_2
路由器 R_3
主机 H_2
电话网
局域网
广域网
局域网
从层次上来看数据的流动
H_1
R_1
R_2
R_3
H_2
应用层
运输层
网络层
链路层
物理层

数据链路层的简单模型

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的



仅从数据链路层观察帧的流动

不同的链路层可能采用不同的数据链路层协议

只考虑数据在数据链路层的流动

链路和数据链路

- **链路(link)**是一条无源的点到点的物理线路段（有线或无线），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。
- 当需要在一条线路上传送数据，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(data link)**。
  - 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。

数据链路和帧

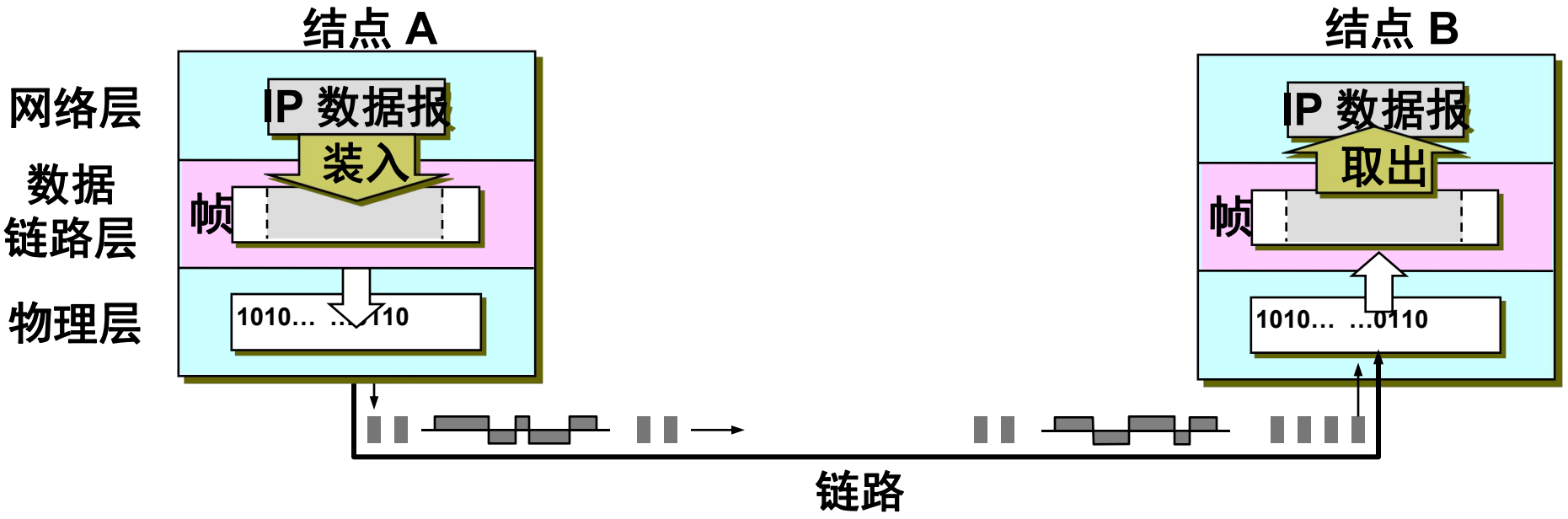
- 也有人采用另外的术语。这就是把链路分为物理链路和逻辑链路。
- 物理链路就是上面所说的链路。
- 逻辑链路就是上面的数据链路，是物理链路加上必要的通信协议。
- 早期的数据通信协议曾叫做通信规程(procedure)。因此在数据链路层，规程和协议是同义语。

数据链路层的信道

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 。

数据链路层传送的是帧

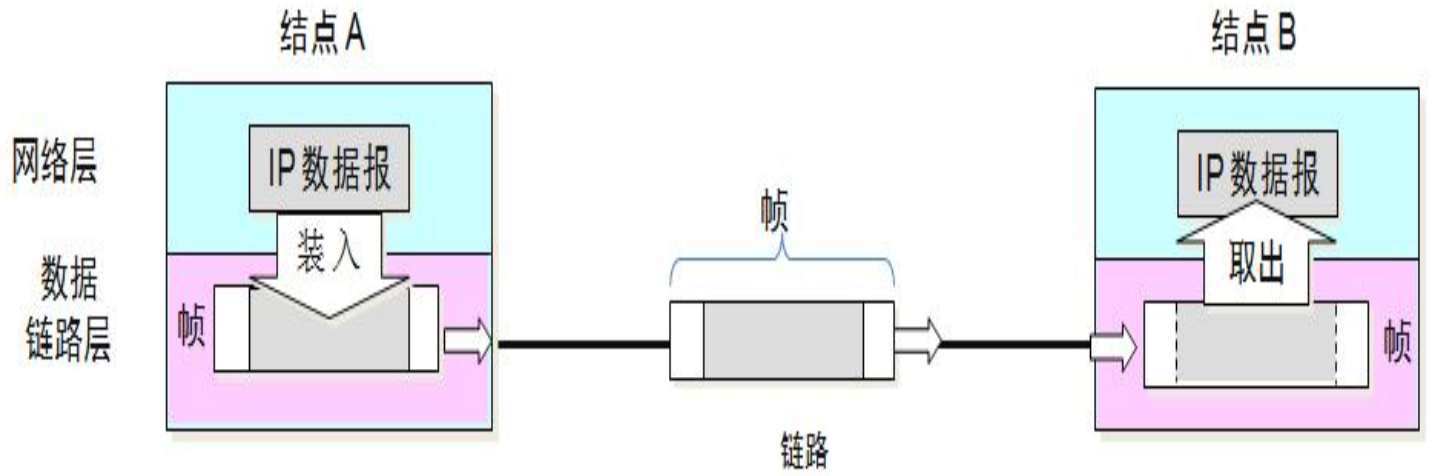


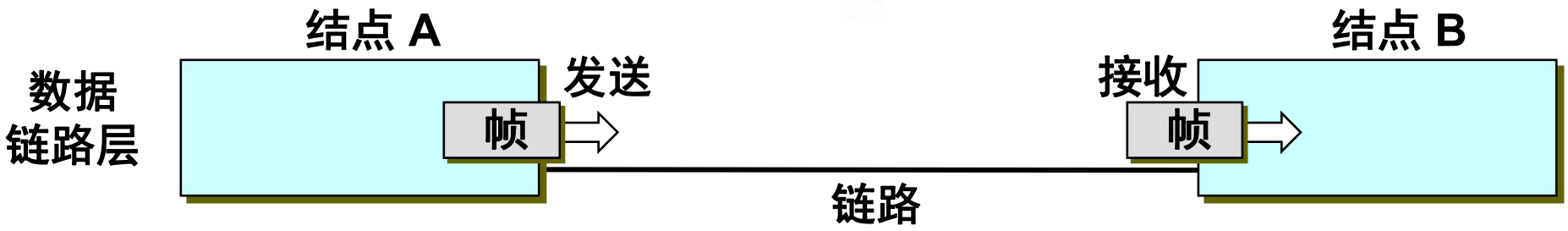
(a)

数据链路层把网络层交下来的数据封装成帧发送到链路上，以及把接收到的帧中的数据取出并上交给网络层。在因特网中，网络层协议数据单元就是IP数据报（或简称为数据报、分组或包）。数据链路层封装的帧，在物理层变成信号在链路上传输。

数据链路层和帧

本章探讨数据链路层，就不考虑物理层如何实现比特传输的细节，我们就可以简单的认为数据帧通过数据链路由节点A发送到节点B。





(b)

数据链路层三个基本问题

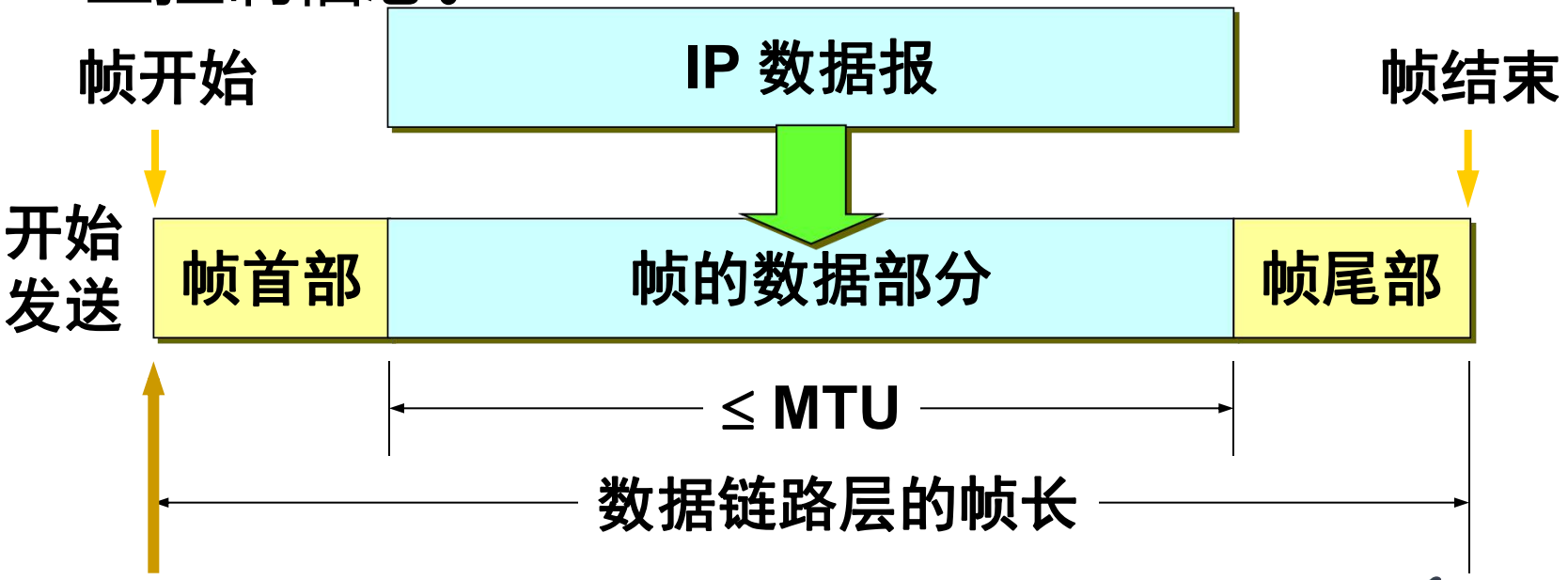
数据链路层协议有许多种，但有三个问题是共同的

(1) 封装成帧
(2) 透明传输
(3) 差错控制

数据链路层三个基本问题

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。还包括一些控制信息。



数据链路层三个基本问题

- 不同的数据链路层协议的帧的首部和尾部包含的信息有明确的规定，帧的首部和尾部有帧开始符和帧结束符，称为帧定界符。接收端收到物理层传过来的数字信号读取到帧开始字符一直到帧结束字符，就认为接收到了一个完整的帧。
- 在数据传输中出现差错时，帧定界符的作用更加明显。
- 每一种数据链路层协议都规定了所能够传送的帧的数据部分长度的上限--即最大传输单元MTU（Maximum Transfer Unit），以太网的MTU为1500个字节。

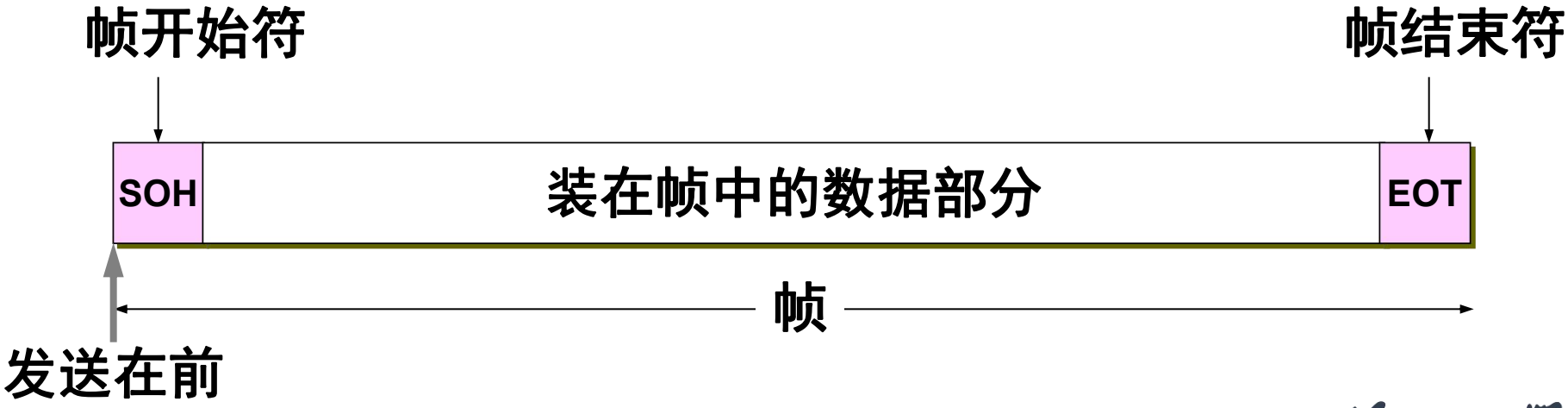
数据链路层具体工作过程

1. 封装成帧

用控制字符进行帧定界的方法举例

当数据由可打印的ASCII 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。另一个控制字符 EOT (End Of Transmission) 表示帧的结束。

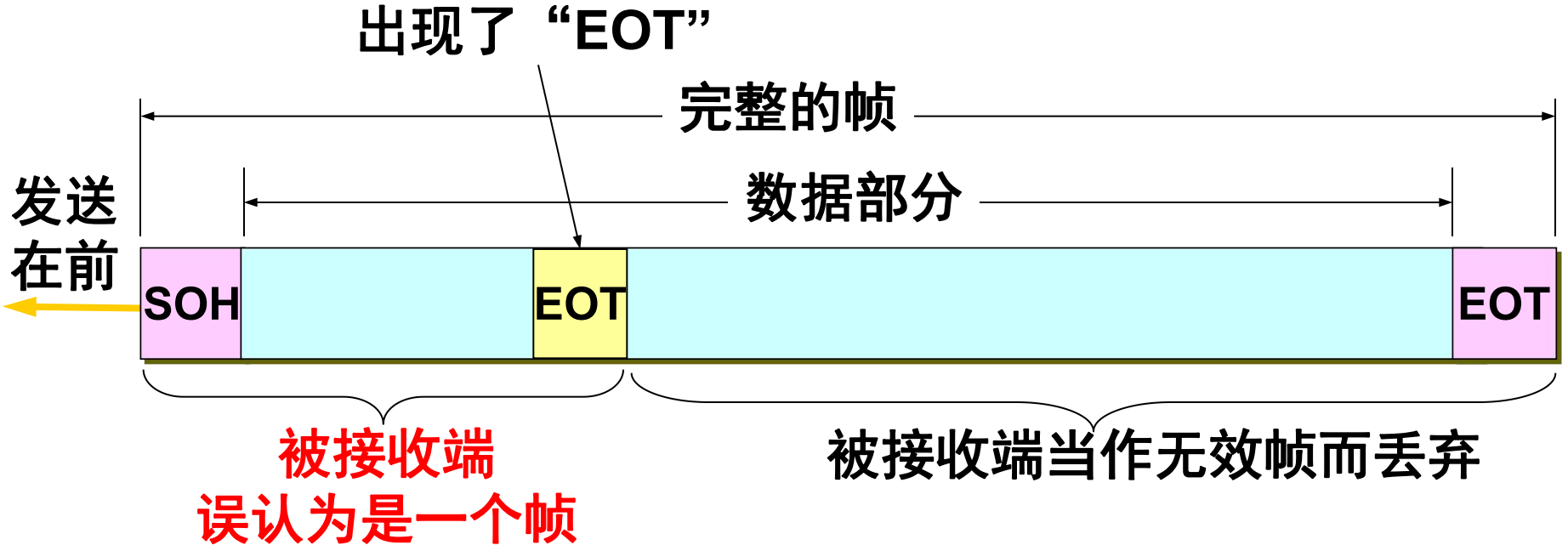


ASCII 字符代码表

高四位 / 低四位		ASCII非打印控制字符										ASCII 打印字符												
		0000					0001					0010		0011		0100		0101		0110		0111		
		0					1					2		3		4		5		6		7		
		+进制	字符	ctrl	代码	字符解释	+进制	字符	ctrl	代码	字符解释	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	◆	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	^Back space

数据链路层具体工作过程

2.透明传输
不管什么数据都能够通过数据链路层传输



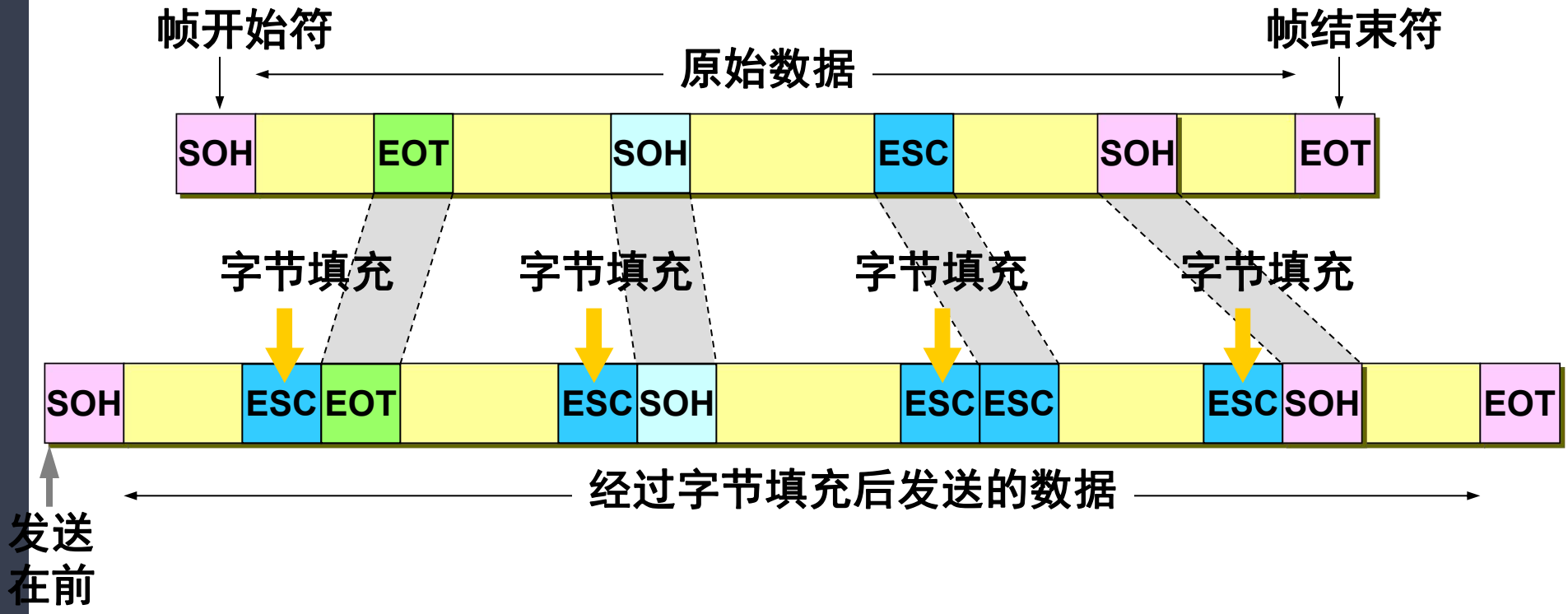
数据部分恰好出现与 EOT 一样的代码

数据链路层具体工作过程

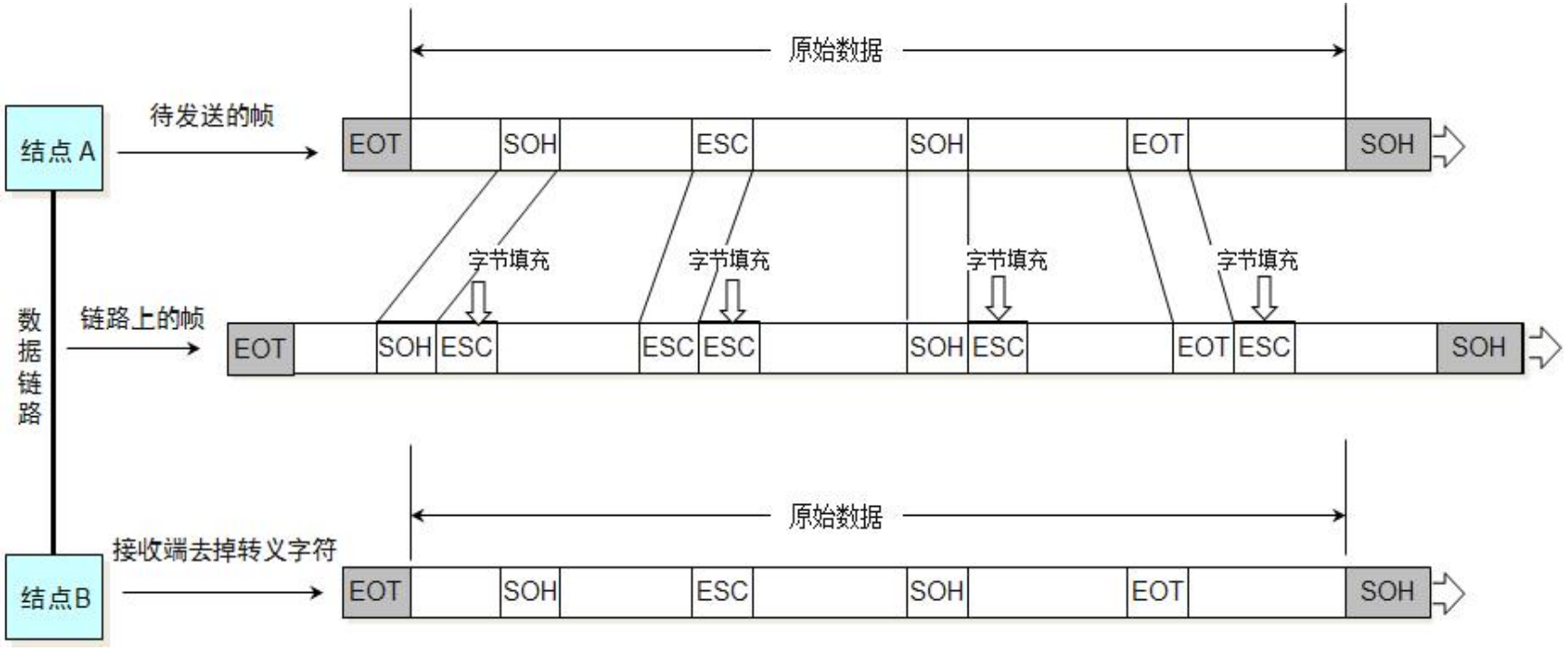
解决透明传输问题

- 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。

字节填充法解决透明传输问题



字节填充法解决透明传输问题



3.差错检测

- 现实的通信链路都不会是理想的。这就是说，比特在传输过程中可能会产生差错：1可能会变成0，而0也可能变成1，这就叫做比特差错。
- 在一段时间内，传输错误的比特占所传输比特总数的比率称为误码率 BER (Bit Error Rate)。
- 误码率与信噪比有很大的关系。
- 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。

差错控制

有奇偶检验码、循环冗余码、海明码等要求：加冗余码少，检错率高。

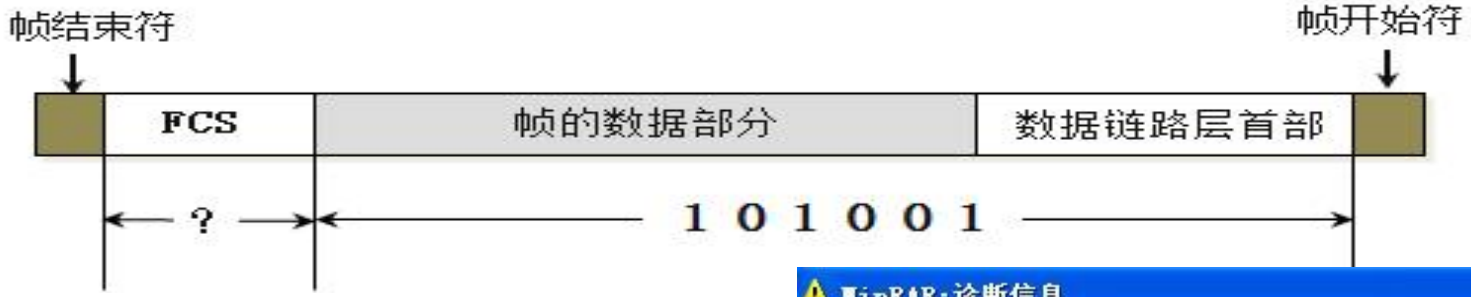
分类：

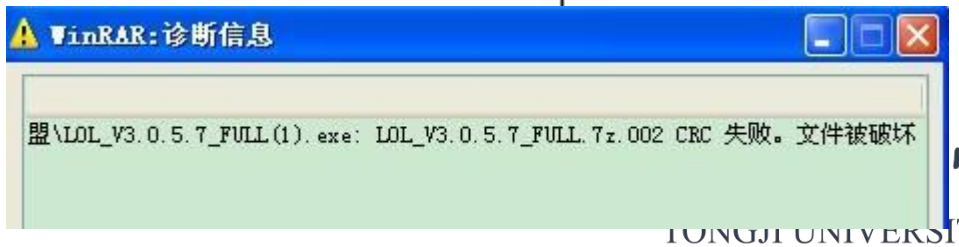
纠错型：对数据进行检验并纠正其中错误部分，用于实时控制系统。

检错型：对数据进行检验，发现错误要求对方进行重发，用于现在大部分网络系统中。

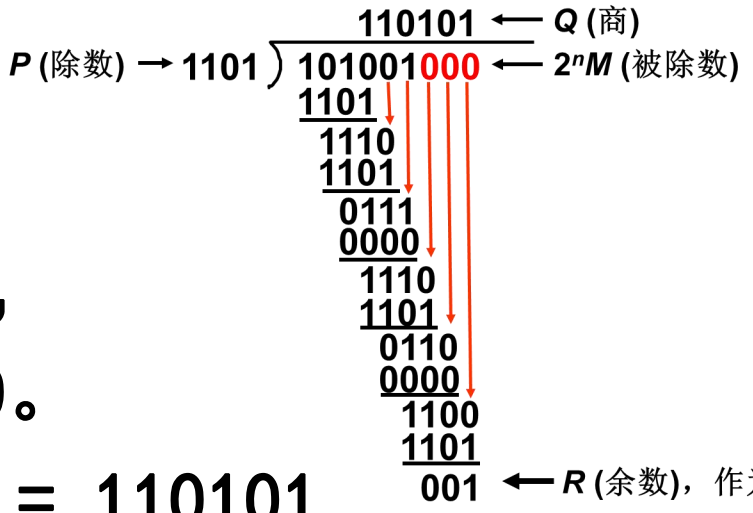
循环冗余检验

- 在数据链路层传送的帧中，广泛使用了循环冗余检验 CRC 的检错技术。
- n位CRC生成：
  - 用二进制的模 2 运算（不产生进位或借位）进行 2^n 乘 M 的运算，这相当于在 M 后面添加 n 个 0。
  - 得到的 $(k+n)$ bit 的数除以事先选定好的长度为 $(n+1)$ bit 的数 P，得出商是 Q 而余数是 R，余数 R 比除数 P 至少要少1 个比特。





冗余码的计算举例



现在 $k = 6$， $M = 101001$。

设 $n = 3$， 除数 $P = 1101$，

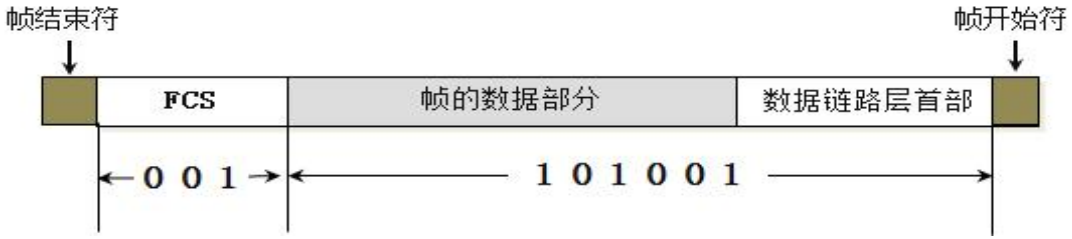
被除数是 $2^nM = 101001000$。

模 2 运算的结果是：商 $Q = 110101$，

余数 $R = 001$。

把余数 R 作为冗余码添加在数据 M 的后面发送出去。发送的数据是：$2^nM + R$

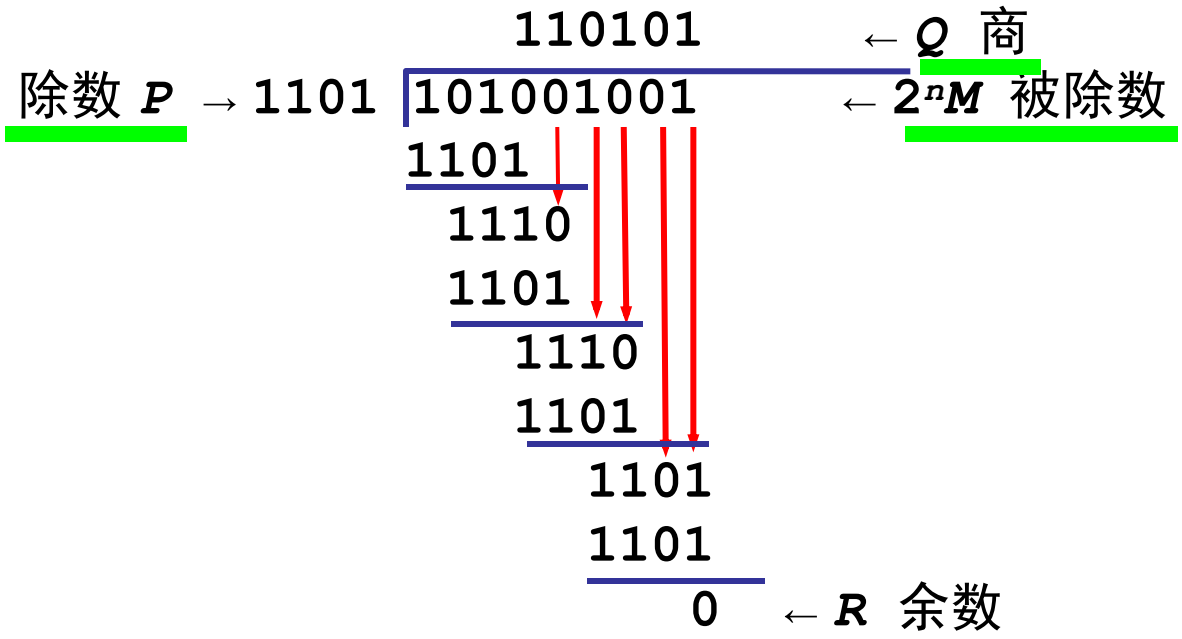
即：101001001，共（$k + n$）位。



检测出差错

- 只要得出的余数 R 不为 0，就表示检测到了差错。
- 但这种检测方法并不能确定究竟是哪一个或哪几个比特出现了差错。
- 一旦检测出差错，就丢弃这个出现差错的帧。
- 需要经过严格的挑选，并使用位数足够多的除数，那么出现检测不到的差错的概率就很小很小。

收到后的检验运算



商可舍弃，余数为0 收到正确，否则要求重发

CRC要求掌握

- 给予需传送数据M与P，列算式，在发送码后加正确的检验码
- 给予收到数据M与P，列算式，得出收到码是否正确

- 在数据链路层，发送端帧检验序列FCS的生成和接收端的 CRC 检验器都是用硬件完成的，处理迅速，因此并不会延误数据的传输。
- 通常使用**生成多项式**来表示循环冗余检验CRC过程
- CRC检验器能够自动丢弃检测到的出错帧。因此所谓的“丢弃出错帧”，对上层软件或用户来说都是感觉不到的。

应当注意

- 仅用循环冗余检验 CRC 差错检测技术只能做到无差错接受 (accept)。
- “无差错接受”是指：“凡是接受的帧（即不包括丢弃的帧），我们都能以非常接近于 1 的概率认为这些帧在传输过程中没有产生差错”。
- 也就是说：“凡是接收端数据链路层接受的帧都没有传输差错”（有差错的帧就丢弃而不接受）。
- 要做到“可靠传输”（即发送什么就收到什么）就必须再加上确认和重传机制。

应当注意

- 应当明确，“无比特差错”与“无传输差错”是不同的概念。
- 在数据链路层使用CRC检验，能够实现无比特差错的传输，但这还不是可靠传输。
- 本章介绍的数据链路层协议都不是可靠传输的协议。“无比特差错”可能会存在帧丢失、帧重复、帧无序等情况。

点对点协议 PPP

- 现在全世界使用得最多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因特网时，一般都是使用 PPP 协议。
- 高级数据链路控制HDLC协议也是点到点链路的一种协议 。

PPP 协议应满足的需求

- 简单 —— 这是首要的要求。
- 封装成帧 —— 必须规定特殊的字符作为帧定界符。
- 透明性 —— 必须保证数据传输的透明性。
- 多种网络层协议 —— 能够在同一条物理链路上同时支持多种网络层协议，如IPv4和IPv6网络层协议都可以封装到PPP帧中。
- 多种类型链路 —— 能够在多种类型的链路上运行。
- 差错检测 —— 能够对接收端收到的帧进行检测，并立即丢弃有差错的帧。

PPP 协议应满足的需求

- 检测连接状态 —— 能够及时自动检测出链路是否处于正常工作状态。
- 最大传送单元 —— 必须对每一种类型的点对点链路设置最大传送单元 MTU 的标准默认值，促进各种实现之间的互操作性。
- 网络层地址协商 —— 必须提供一种机制使通信的两个网络层实体能够通过协商知道或能够配置彼此的网络层地址。
- 数据压缩协商 —— 必须提供一种方法来协商使用数据压缩算法。

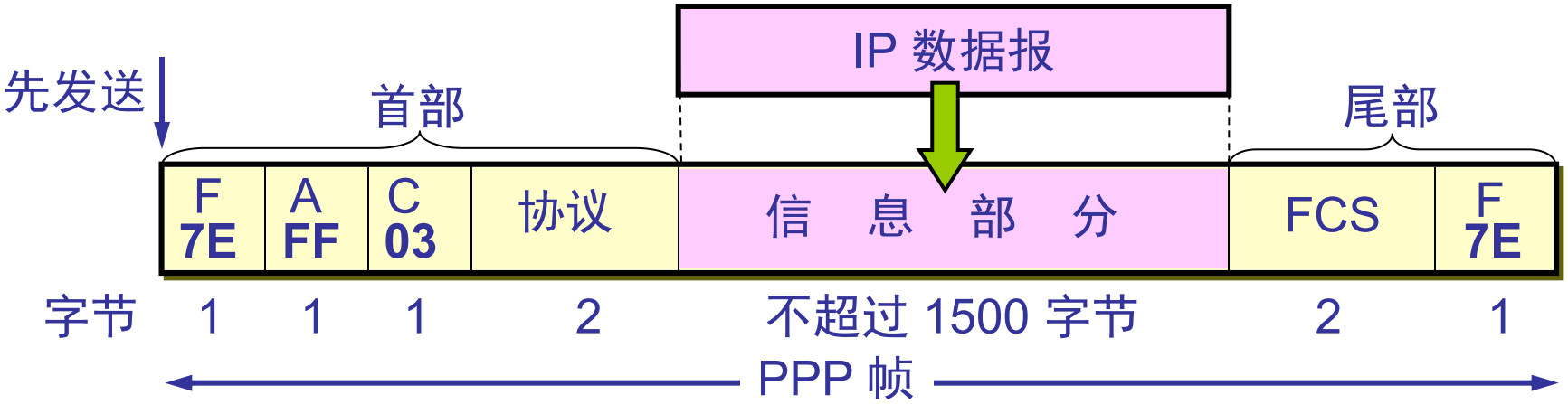
PPP 协议

- 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]。
- PPP协议有三个组成部分
  - 一个将 IP 数据报封装到串行链路的方法。
  - 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，通信的双方可协商一些选项
  - 网络控制协议 NCP (Network Control Protocol)：每一个协议支持不同的网络协议，如IPv6、appletalk等。

PPP 协议的帧格式

- PPP 帧的首部和尾部分别为 4 个字段和 2 个字段。
- 标志字段 F 为 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，可以看到没有源地址和目标地址。
- 控制字段 C 通常置为 0x03。
- 最初曾考虑以后对地址字段和控制字段的值进行其他定义，但至今也没给出
- 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。其值

- **若为 0x0021，则信息字段就是 IP 数据报。**
- **若为 0x8021，则信息字段是网络控制数据。**
- **若为 0xC021，则信息字段是 PPP 链路控制数据。**
- **若为 0xC023，则信息字段是鉴别数据。**

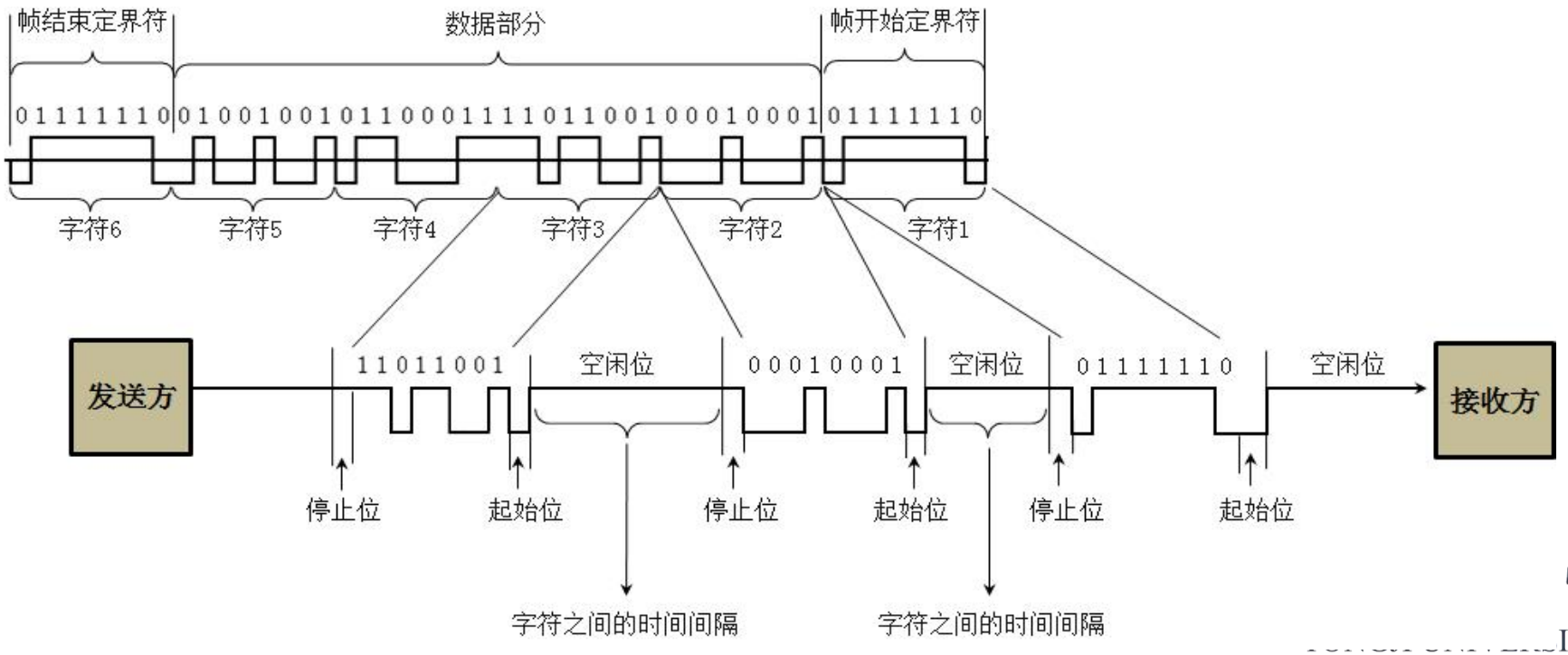
PPP 协议的帧格式

- 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

异步传输（Asynchronous Transmission）以字符为单位传输数据，发送端和接收端具有相互独立的时钟（频率相差不能太多），并且两者中任一方都不向对方提供时钟同步信号。

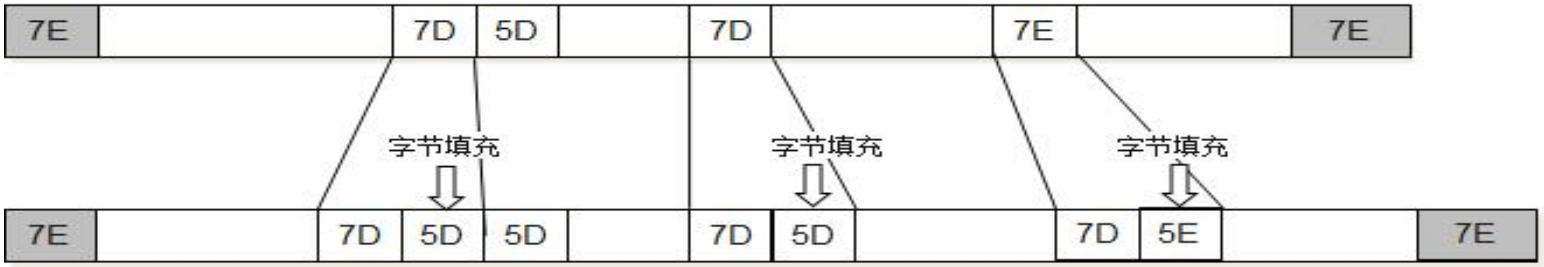
异步传输示意图



字符填充

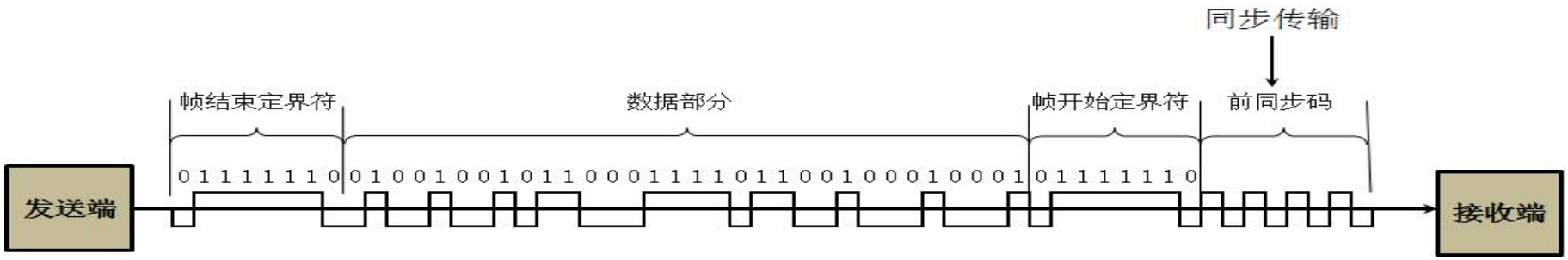
- 在异步传输的链路上，数据传输以字节为单位，PPP帧的转义符定义为0x7D，并使用字节填充。
- 将信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列(0x7D, 0x5E)。
- 若信息字段中出现一个 0x7D 的字节, 则将其转变成为 2 字节序列(0x7D, 0x5D)。

PPP帧字节填充



同步传输

同步传输（Synchronous Transmission）以数据帧为单位传输数据，可采用字符形式或位组合形式的帧同步信号，在短距离的高速传输中，该时钟信号可由专门的时钟线路传输，由发送端或接收端提供专用于同步的时钟信号。计算机网络采用同步传输方式时，常将时钟同步信号（前同步码）植入数据信号帧中，以实现接收端与发送端的时钟同步。



零比特填充

- PPP 协议用在光同步数字传输网SONET/SDH链路时，是使用同步传输（一连串的比特连续传送）。
- 大家把PPP协议帧界定符0x7E写成二进制01111110,也就是可以看到中间有连续的6个1,只要想办法在数据部分不要出现连续的6个1,就肯定不会出现这界定符。
- 这时 PPP 协议采用零比特填充方法来实现透明传输。即在发送端，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

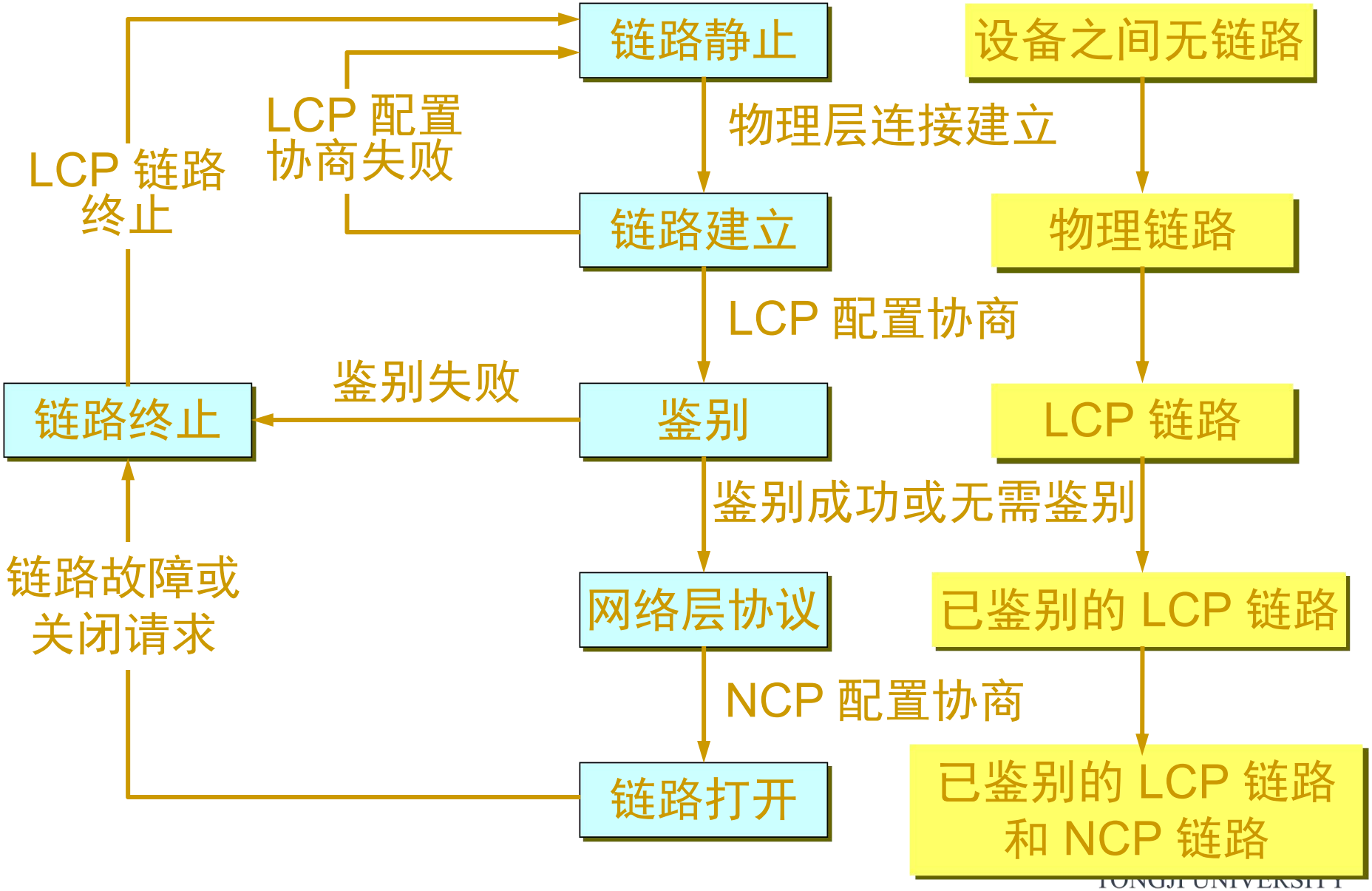
不使用序号和确认的可靠传输

- 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。
  - 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。
  - 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）。
- 这些分组及其响应选择一些 PPP 参数，和进行网络层配置，NCP 给新接入的 PC机分配一个临时的 IP 地址，使 PC 机成为因特网上的一个主机。
- 通信完毕时，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。
- 可见，PPP 协议已不是纯粹的数据链路层的协议，它还包含了物理层和网络层的内容。

PPP 协议的工作状态图



使用广播信道的数据链路层

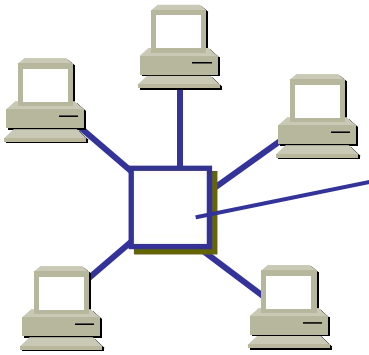
局域网的数据链路层

局域网最主要的特点是：网络为一个单位所拥有，且地理范围和站点数目均有限。

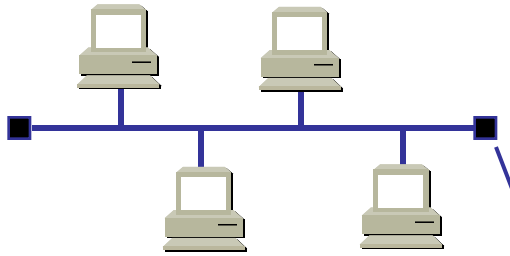
局域网具有如下的一些主要优点：

- 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、可用性和生存性。

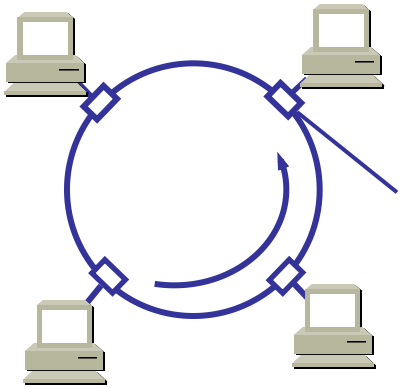
局域网的拓扑



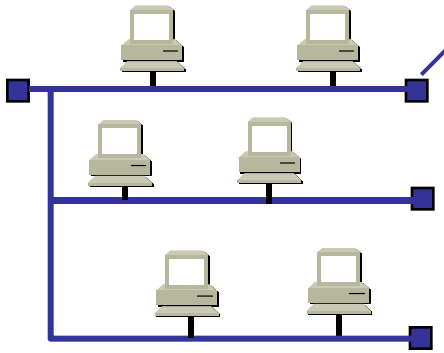
星形网



总线网



环形网



树形网

媒体共享技术

- 静态划分信道（信道固定分配给用户，代价高，不适合于局域网）
  - 频分复用
  - 时分复用
  - 波分复用
  - 码分复用
- 动态媒体接入控制（多点接入）
  - 随机接入（随机发送数据、存在碰撞或冲突）
  - 受控接入，如多点线路探询(polling)，或轮询。

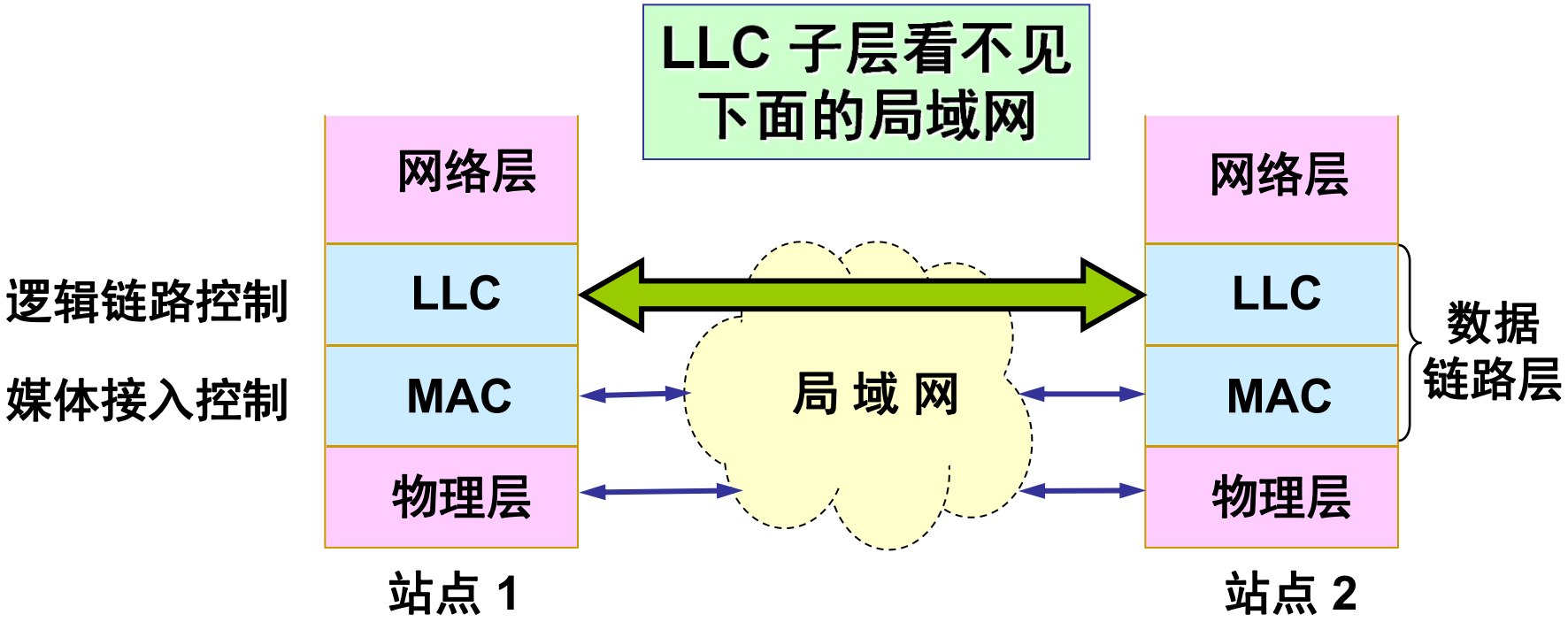
以太网的两个标准

- DIX Ethernet V2 是世界上第一个局域网产品（以太网）的规约，数据率10Mb/s。
- IEEE 的 802.3 标准,数据率10Mb/s 。
- DIX Ethernet V2 标准与 IEEE 的 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
- 以太网属于随机接入共享信道方式

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，IEEE802 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



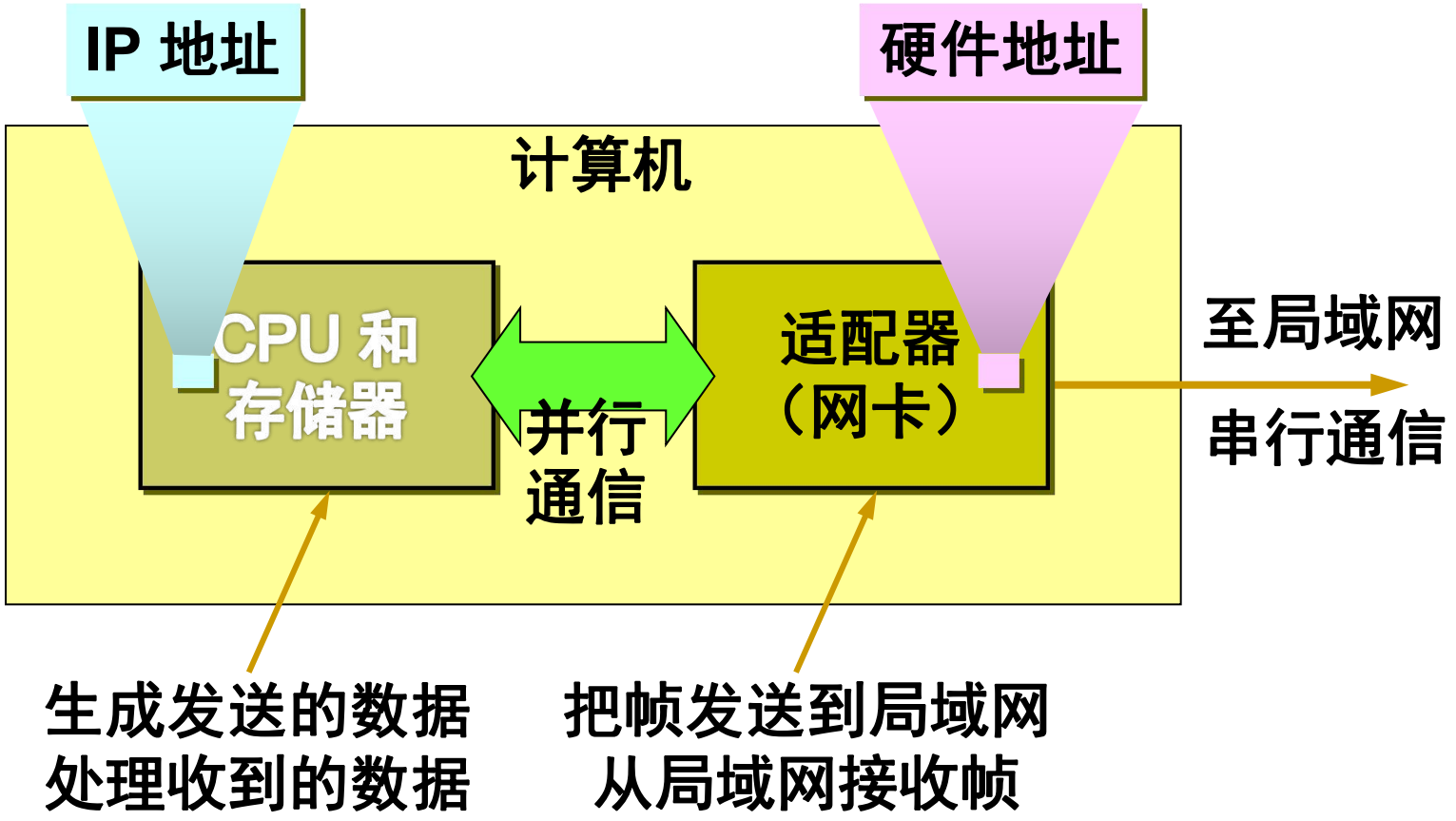
LLC 子层趋向淘汰

- 由于 TCP/IP 体系经常使用的局域网是 DIX Ethernet V2 而不是 802.3 标准中的局域网，因此现在 802 委员会制定的逻辑链路控制子层 LLC（即 802.2 标准）的作用已经不大了。
- 很多厂商生产的适配器上就仅装有MAC 协议而没有 LLC 协议。

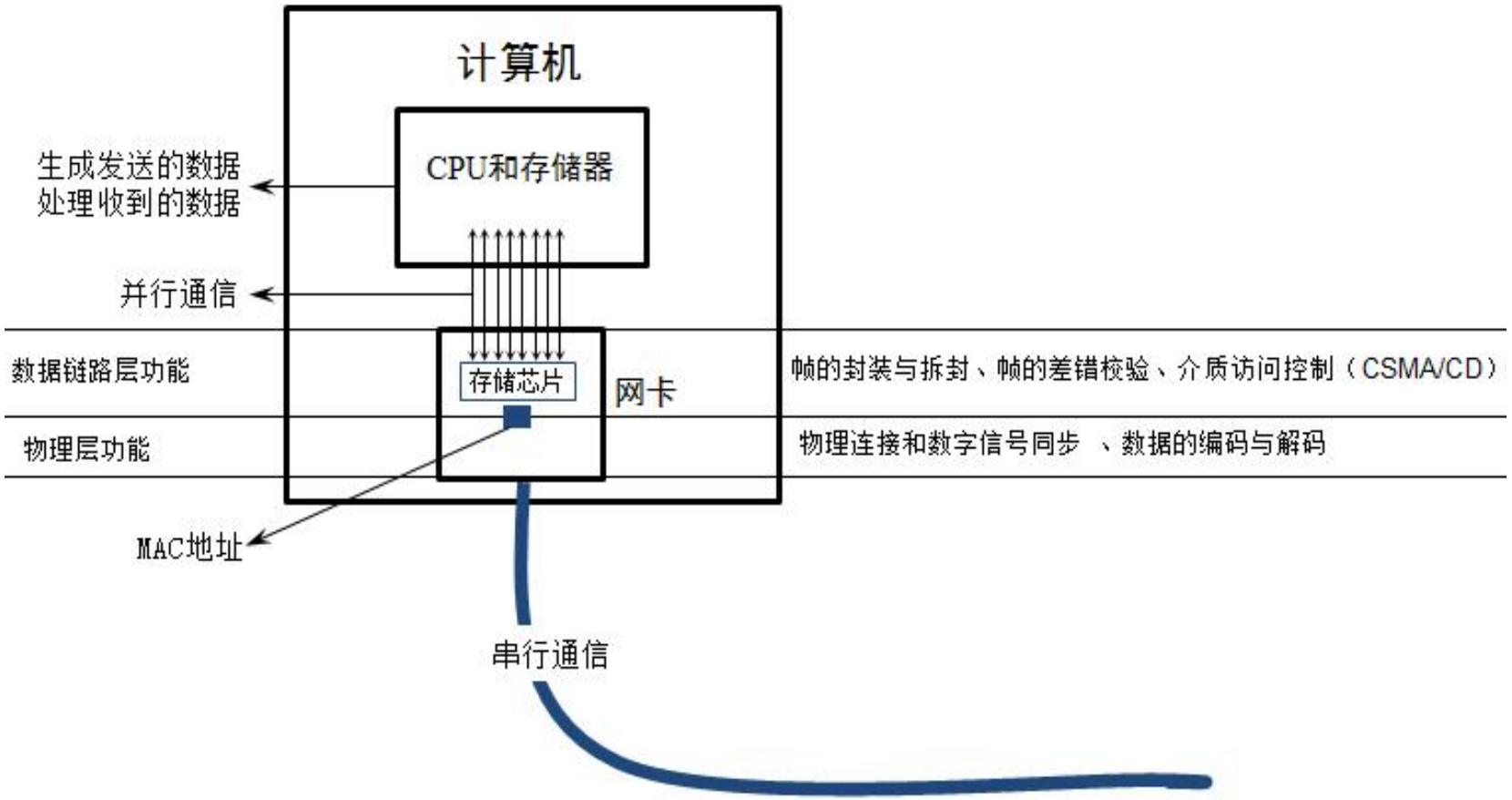
网卡（适配器）的作用

- 网络接口板又称为通信适配器(adapter)或网络接口卡 NIC (Network Interface Card)，或“网卡”。
- 适配器的重要功能：
  - 进行串行/并行转换。
  - 对数据进行缓存。
  - 在计算机的操作系统安装设备驱动程序。
  - 实现以太网协议。

计算机通过网卡和局域网进行通信

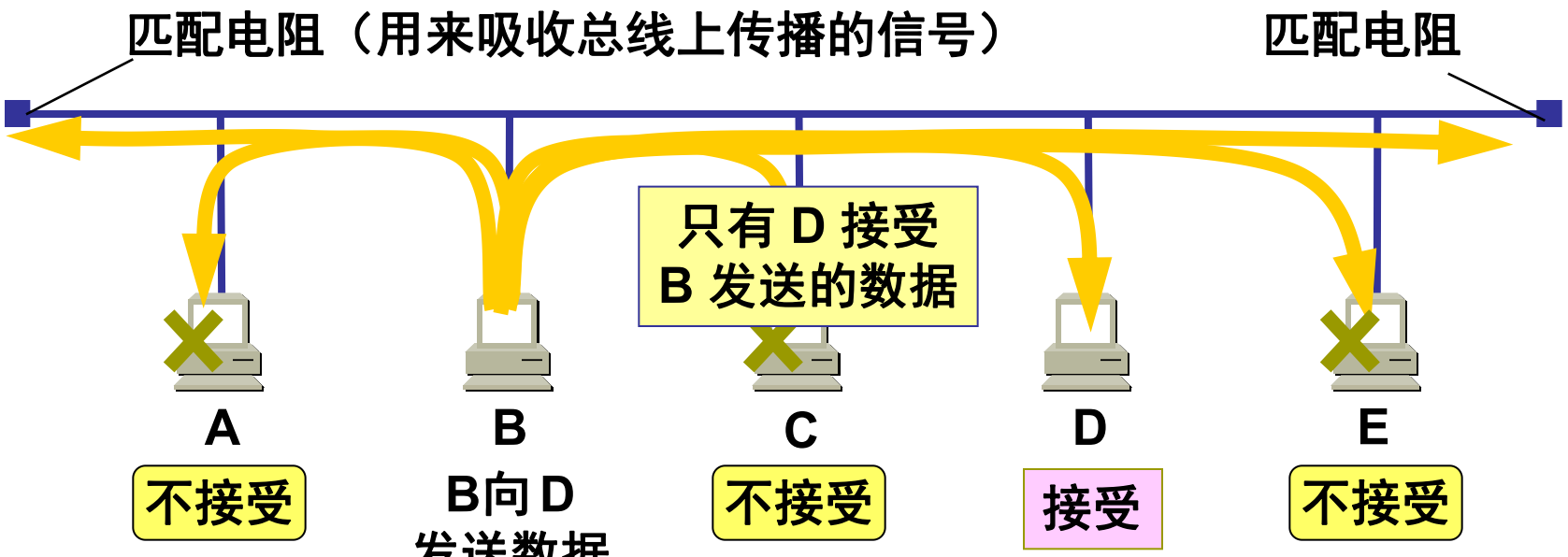


网卡的作用


计算机
CPU和存储器
生成发送的数据
处理收到的数据
并行通信
数据链路层功能
存储芯片
网卡
帧的封装与拆封、帧的差错校验、介质访问控制（CSMA/CD）
物理层功能
物理连接和数字信号同步 、数据的编码与解码
MAC地址
串行通信

CSMA/CD 协议

- 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。

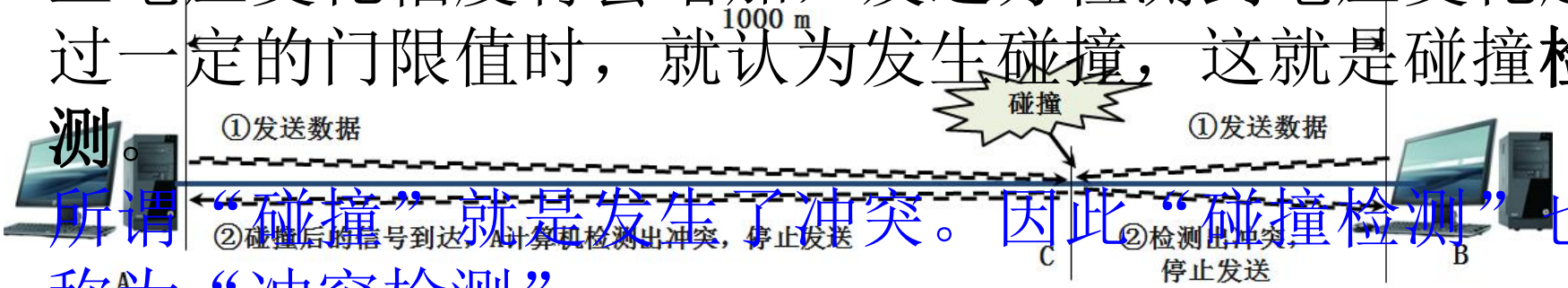
CSMA/CD 协议

载波监听多点接入/碰撞检测（CSMA/CD）

- CSMA/CD 表示 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。
- 总线上并没有什么“载波”。因此，“载波监听”就是用电子技术检测总线上有没有其他计算机发送的数据信号。

CSMA/CD协议

- “碰撞检测”也就是“边发送边监听”，即适配器边发送数据边检测信道上的信号电压的变化情况。
- 比如，A计算机发送的信号和B计算机发送的信号在链路C处发生碰撞，碰撞后的信号相互叠加，在总线上电压变化幅度将会增加，发送方检测到电压变化超过一定的门限值时，就认为发生碰撞，这就是碰撞**检测**。
- 所谓“碰撞”就是发生了冲突。因此“碰撞检测”也称为“冲突检测”。



CSMA/CD协议

检测到碰撞后

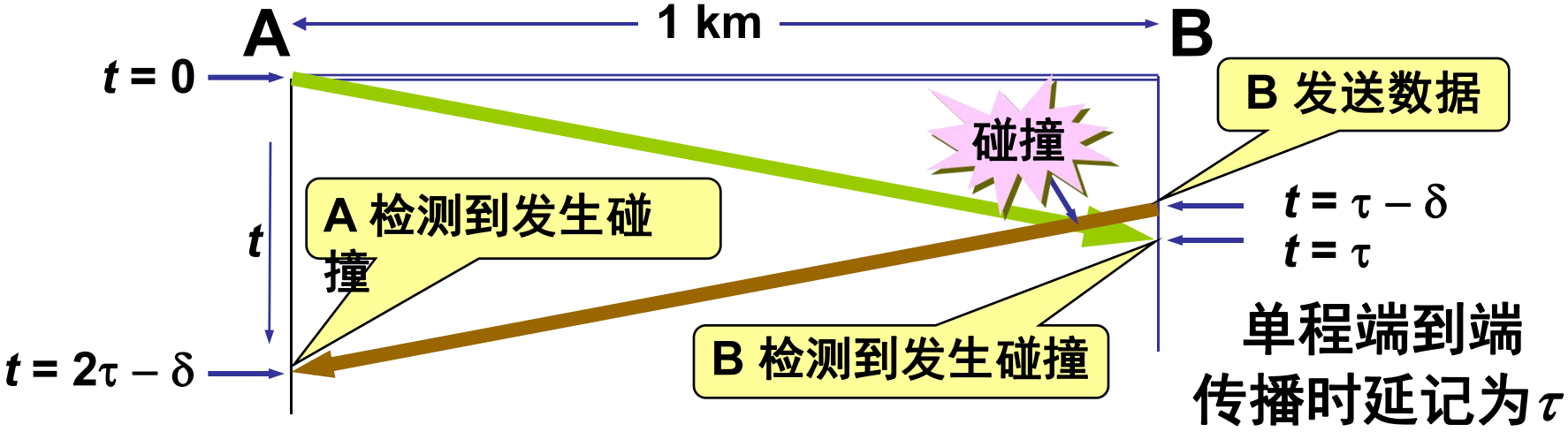
- 在发生碰撞时，总线上传输的信号产生了严重的失真，无法从中恢复出有用的信息来。
- 每一个正在发送数据的站，一旦发现总线上出现了碰撞，就要立即停止发送，免得继续浪费网络资源，然后等待一段随机时间后再次发送。

CSMA/CD协议

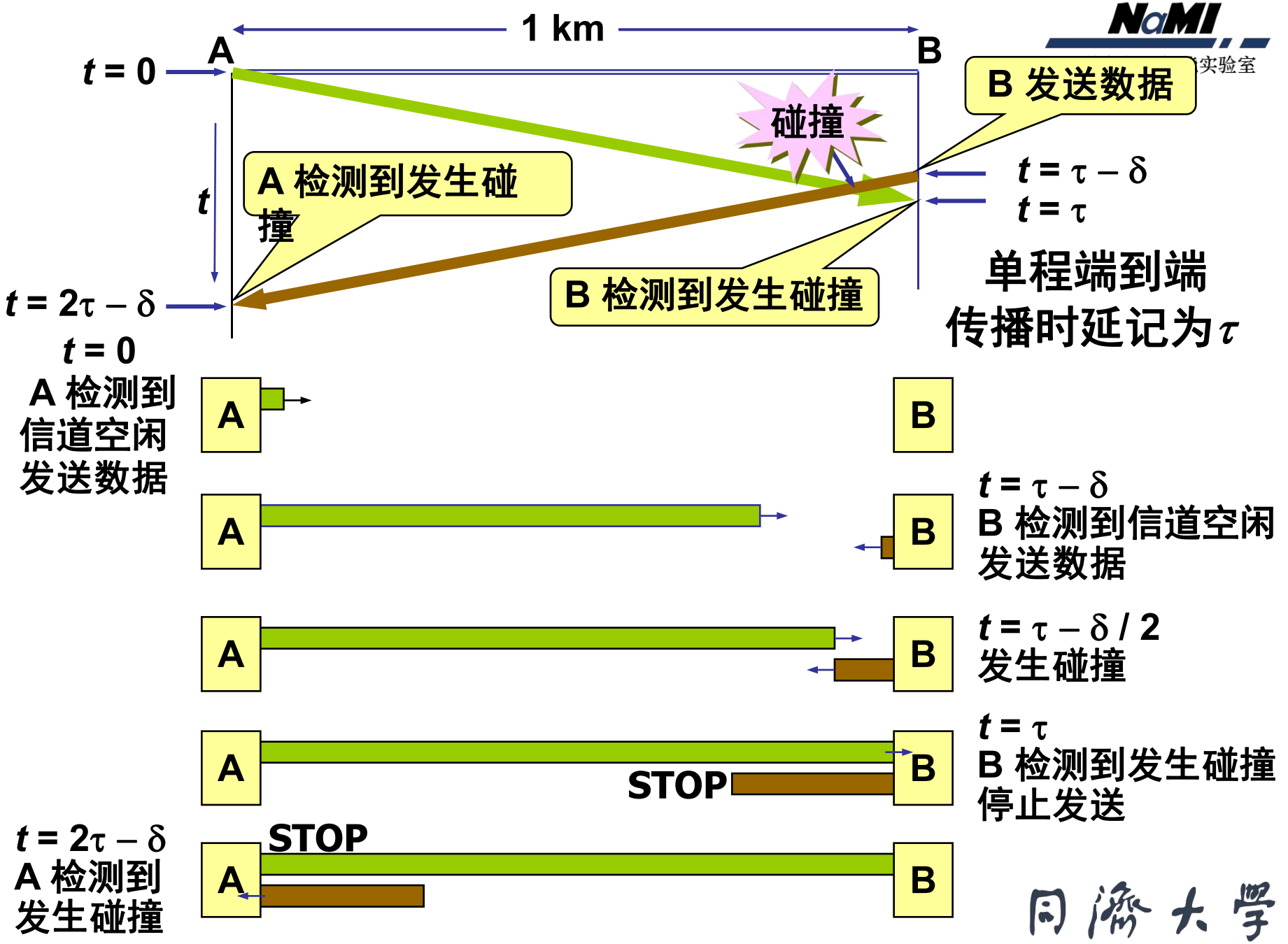
电磁波在总线上的有限传播速率的影响

- 当某个站监听到总线是空闲时，也可能总线并非真正是空闲的。
- A 向 B 发出的信息，要经过一定的时间后才能传送到 B。
- B 若在 A 发送的信息到达 B 之前发送自己的帧(因为这时 B 的载波监听检测不到 A 所发送的信息)，则必然要在某个时间和 A 发送的帧发生碰撞。
- 碰撞的结果是两个帧都变得无用。
- 所以需要在发送期间进行碰撞检测，以检测冲突。

传播时延对载波监听的影响



A最长需要单程传播时延的 2 倍的时间，才能检测到与 B 的发送产生了冲突


1 km
A
B
t = 0
t
碰撞
B 发送数据
A 检测到发生碰撞
t = τ – δ
t = τ
t = 2τ – δ
B 检测到发生碰撞
单程端到端传播时延记为τ
t = 0
A 检测到信道空闲发送数据
A
B
t = τ – δ
B 检测到信道空闲发送数据
t = τ – δ / 2
发生碰撞
STOP
t = τ
B 检测到发生碰撞停止发送
t = 2τ – δ
A 检测到发生碰撞
STOP

重要特性

- 使用 CSMA/CD 协议的以太网不能进行全双工通信而只能进行双向交替通信（半双工通信）。
- 每个站在发送数据之后的一小段时间内，存在着遭遇碰撞的可能性。称为：发送的不确定性。
- 这种发送的不确定性使整个以太网的平均通信量远小于以太网的最高数据率。

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 （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。

截断二进制指数退避算法

- 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。
  - 确定基本退避时间，一般是取为争用期 2τ。
  - 定义重传次数 k，$k \le 10$，即

    $$k = \text{Min}[\text{重传次数}, 10]$$

  - 从整数集合$[0,1,\ldots,(2^k-1)]$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。
  - 当重传达 16 次仍不能成功时即丢弃该帧，并向高层报告。

以太网最短帧

- 以太网设计最大端到端长度为5km（实际上的以太网覆盖范围远远没有这么大），单程传播时延为大约为25.6μs,往返传播时延为51.2μs，10M标准以太网最小帧为：

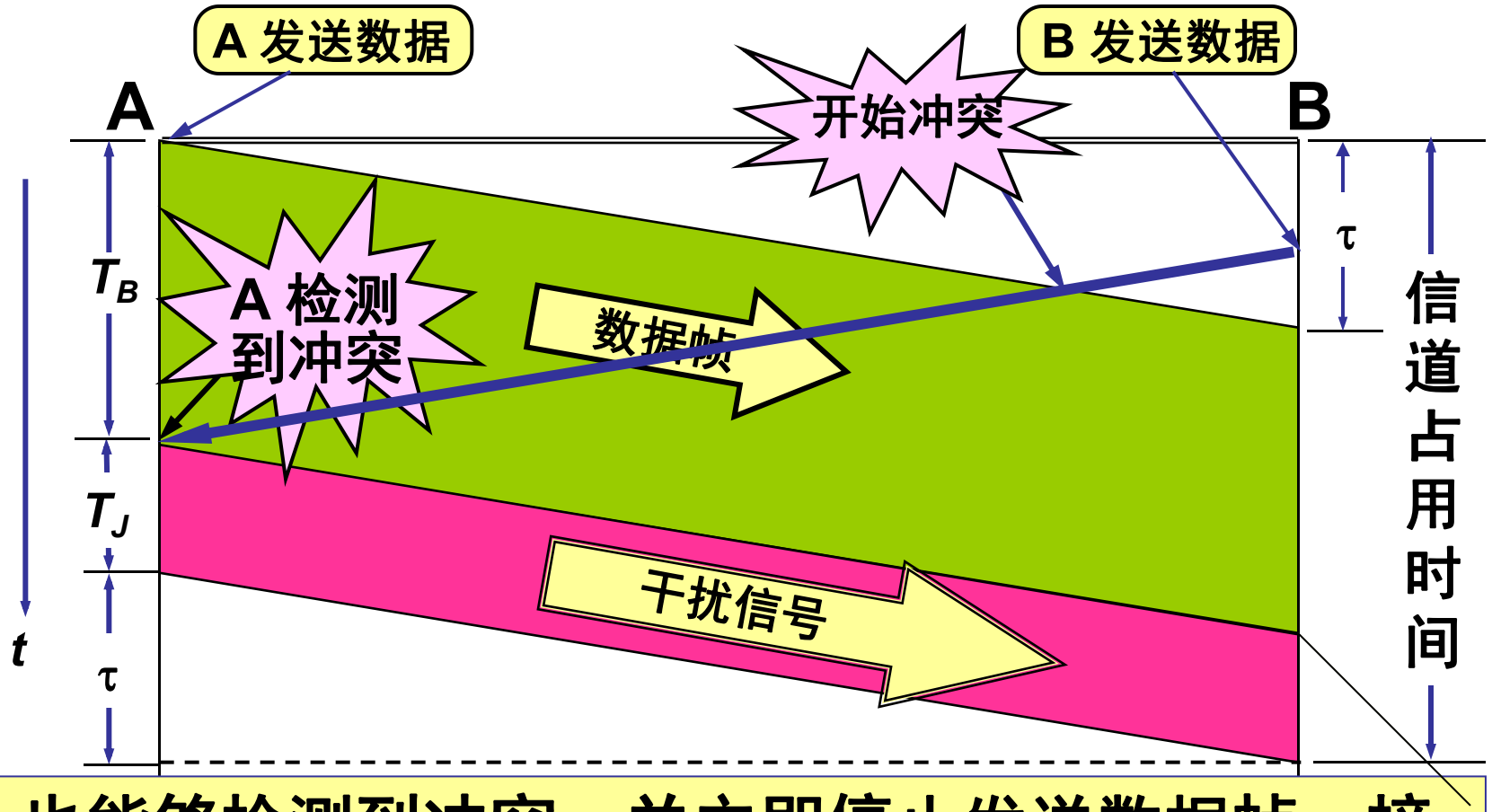
$$10Mb/s \times 51.2\mu s = 10^7 b/s \times 51.2 \times 10^{-6} s = 512b$$

- 512比特也就是64字节，这就意味着以太网发送数据帧如果前64字节没有检测出冲突，后面发送的数据就一定不会发生冲突。换句话说，如果发生碰撞，就一定在发送前64字节之内。
- 由于一旦检测出冲突就立即终止发送，这时发送的数据一定小于64字节，因此凡是长度小于64字节的帧都是由于冲突而异常终止的无效帧，只要收到了这种无效帧，就应当立即将其终止。

强化碰撞

- 当发送数据的站一旦发现发生了碰撞时：
  - 立即停止发送数据；
  - 再继续发送若干比特（一般为32比特或48比特）的人为干扰信号(jamming signal)，以便让所有用户都知道现在已经发生了碰撞。

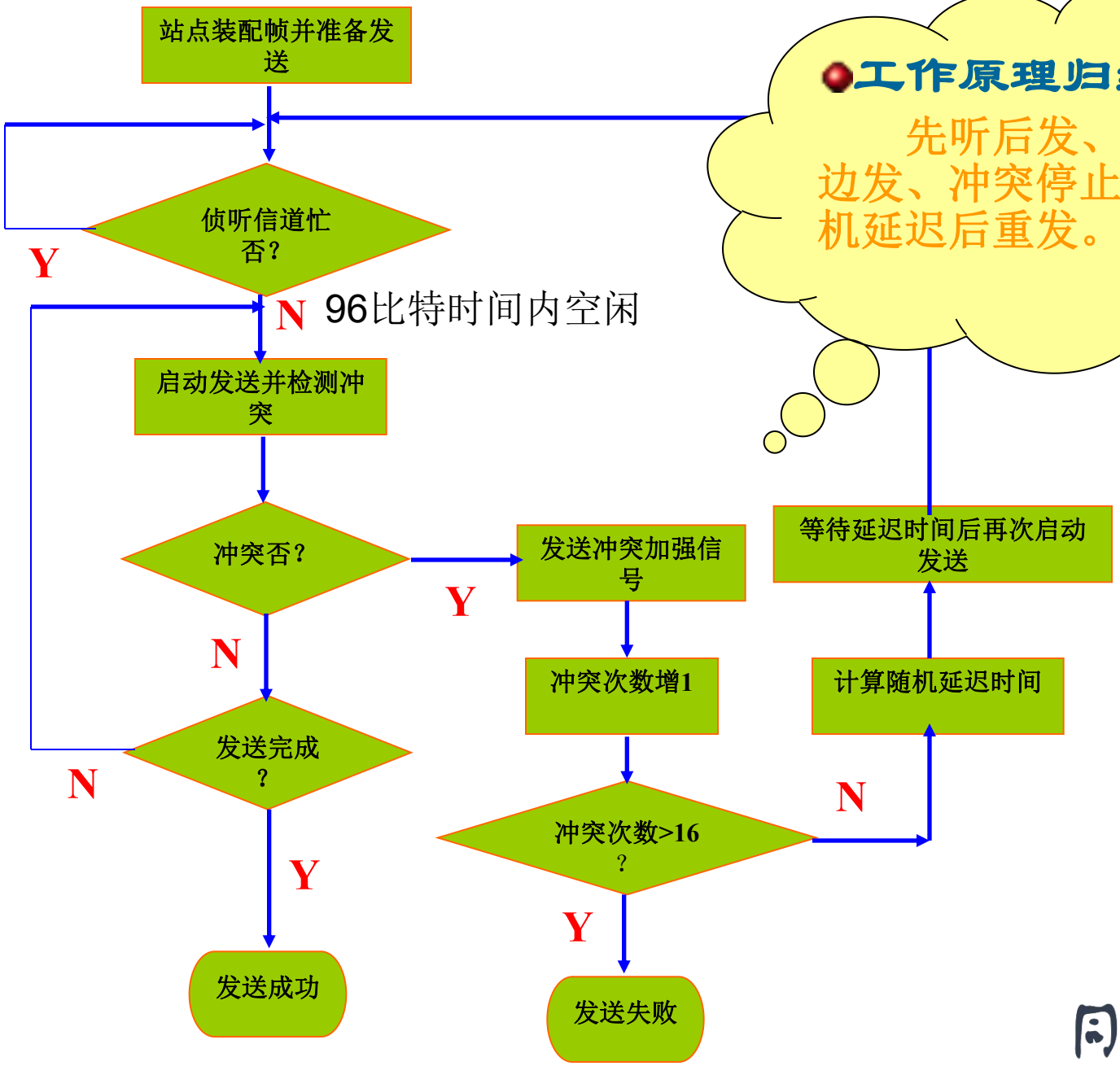
人为干扰信号



B 也能够检测到冲突，并立即停止发送数据帧，接着就发送干扰信号。这里为了简单起见，只画出 A 发送干扰信号的情况。

帧间最小间隔

- 帧间最小间隔为 9.6 μs，相当于 96 bit 的发送时间。
- 一个站在检测到总线开始空闲后，还要等待 9.6 μs 才能再次发送数据。
- 这样做是为了使刚刚收到数据帧的站的接收缓存来得及清理，做好接收下一帧的准备。



使用广播信道的以太网

使用集线器的星形拓扑

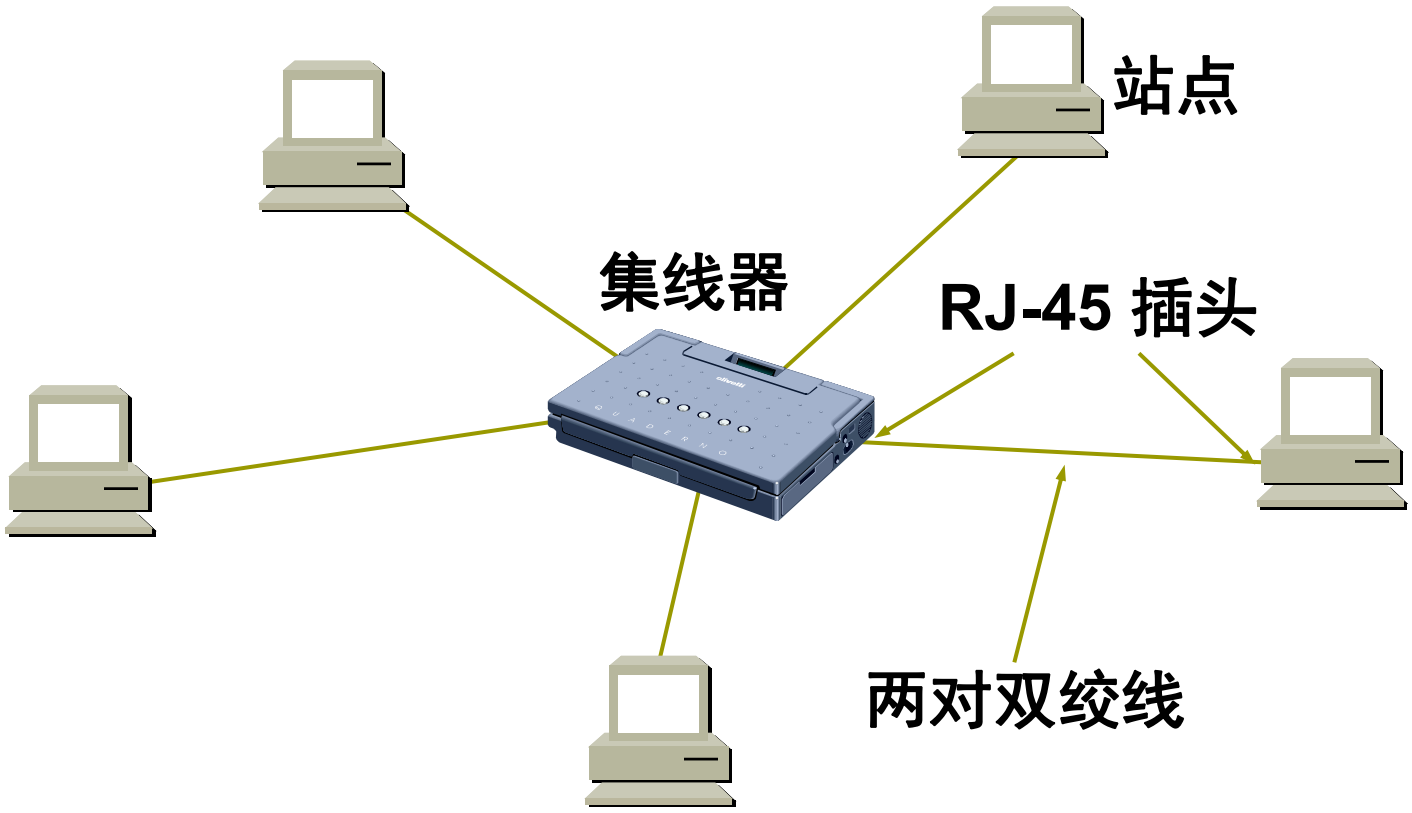
- 传统以太网最初是使用粗同轴电缆，后来演进到使用比较便宜的细同轴电缆，最后发展为使用更便宜和更灵活的双绞线。
- 这种以太网采用星形拓扑，在星形的中心则增加了一种可靠性非常高的设备，叫做集线器(hub)

以太网标准

在IEEE 802.3标准中，为不同的传输介质制定了不同的物理层标准，在这些标准中前面的数字表示传输速度，单位是“Mbps”，最后的一个数字表示单段网线长度（基准单位是100m），Base表示“基带”的意思。

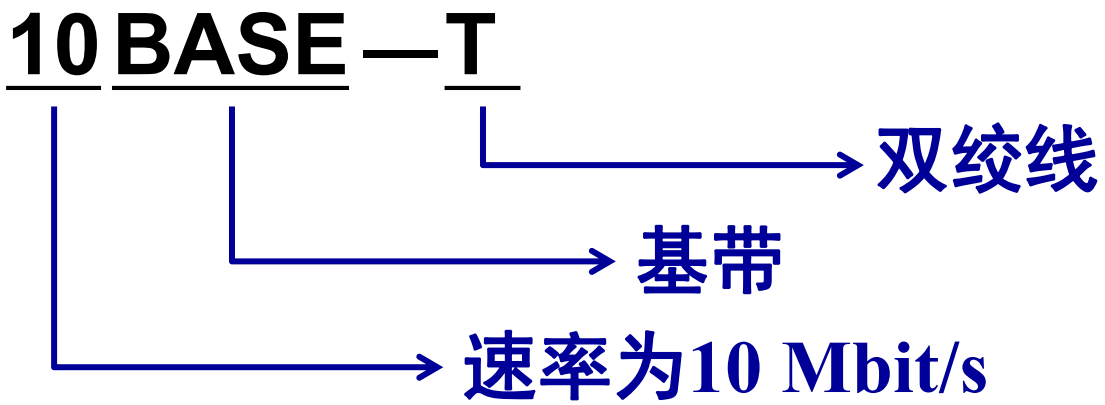
名称	传输介质	网段最大长度	特点
10Base-5	粗同轴电缆	500 米	干线电缆，不经济
10Base-2	细同轴电缆	185 米	不需要集线器
10Base-T	非屏蔽双绞线	100 米	最便宜的系统
10Base-F	光纤	2000 米	适合于楼间使用

使用集线器的双绞线以太网



星形网 10BASE-T

- 1990 年，IEEE 制定出星形以太网 10BASE-T 的标准 802.3i。



星形网 10BASE-T

- 使用无屏蔽双绞线，采用星形拓扑。
- 每个站需要用两对双绞线，分别用于发送和接收。
- 双绞线的两端使用 RJ-45 插头。
- 集线器使用了大规模集成电路芯片，因此集线器的可靠性提高。
- 10BASE-T 的通信距离稍短，每个站到集线器的距离不超过 100 m。

星形网 10BASE-T

- 这种 10 Mbit/s 速率的无屏蔽双绞线星形网的出现，既降低了成本，又提高了可靠性。 具有很高的性价比。
- 10BASE-T 双绞线以太网的出现，是局域网发展史上的一个非常重要的里程碑，它为以太网在局域网中的统治地位奠定了牢固的基础。
- 从此以太网的拓扑就从总线形变为更加方便的星形网络，而以太网也就在局域网中占据了统治地位。

以太网的 MAC 层

重点介绍：

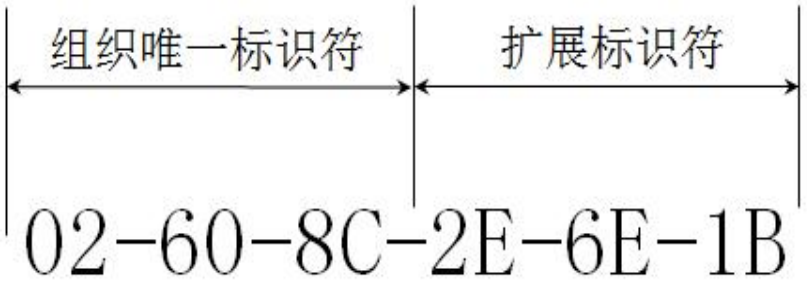
- 1. MAC 层的硬件地址
- 2. MAC 帧的格式

MAC地址

- 在广播信道实现点到点通信，这就需要网络中的每个网卡有一个地址。这个地址称为物理地址或MAC地址（因为这种地址用在MAC帧中）。IEEE802标准为局域网规定了一种48位的全球地址。
- 这种6字节的MAC地址已被固化在网卡的ROM中。因此，MAC地址也叫作硬件地址（hardware address）或物理地址。当这块网卡插入（或嵌入）到某台计算机后，网卡上的MAC地址就成为这台计算机的MAC地址了。

48 位的 MAC 地址

- IEEE 的注册管理机构 RA 负责向厂家分配地址字段的前三个字节(即高位 24 位)。
- 地址字段中的后三个字节(即低位 24 位)由厂家自行指派，称为扩展标识符，必须保证生产出的适配器没有重复地址。
- 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（Extended Unique Identifier）。
- “MAC地址”实际上就是适配器地址或适配器标识符EUI-48。



MAC地址

- 网卡有过滤功能，适配器从网络上每收到一个MAC帧就先用硬件检查MAC帧中的目的地址。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。这样做就不浪费主机的处理机和内存资源。这里“发往本站的帧”包括以下三种帧：

  （1）单播（unicast）帧（一对一），即收到的帧的MAC地址与本站的硬件地址相同。

  （2）广播（broadcast）帧（一对全体），即发送给本局域网上所有站点的帧（全1地址）。

  （3）多播（multicast）帧（一对多），即发送给本局域网上一部分站点的帧。

MAC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^{23} 个单个站地址和 2^{23} 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。

MAC帧的格式

常用的以太网MAC帧格式有两种标准：

DIX Ethernet V2 标准

IEEE 的 802.3 标准

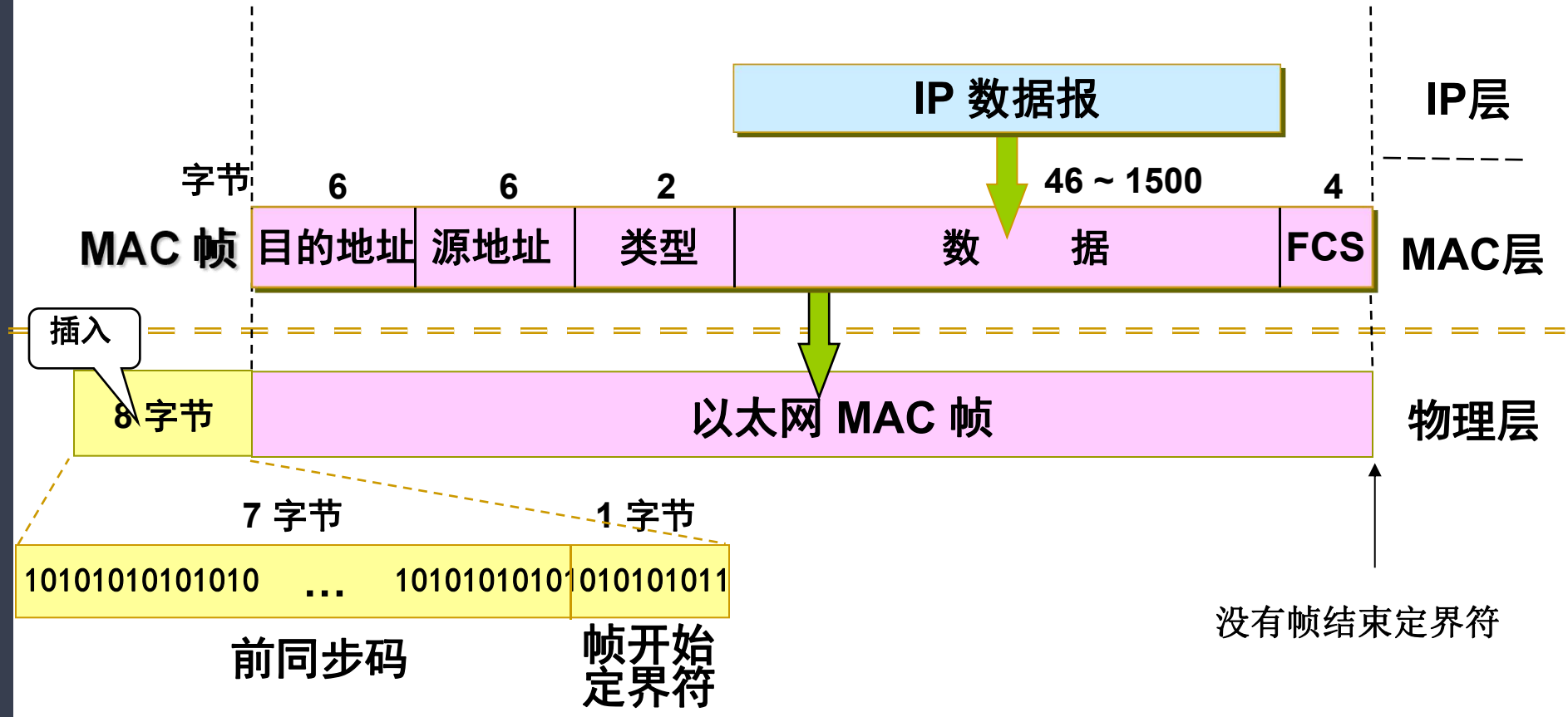
最常用的 MAC 帧是以太网 V2 的格式

查看物理地址方法:

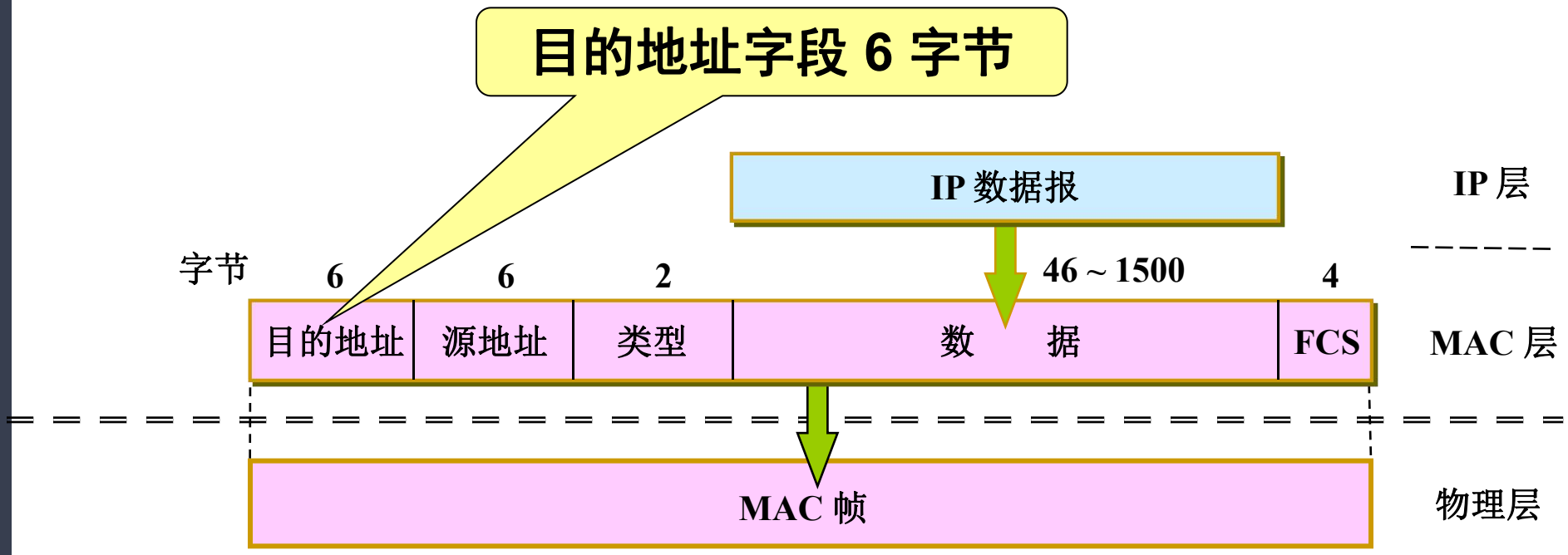
DOS 状态下输入 IPCONFIG/ALL

可用于网络安全性设置

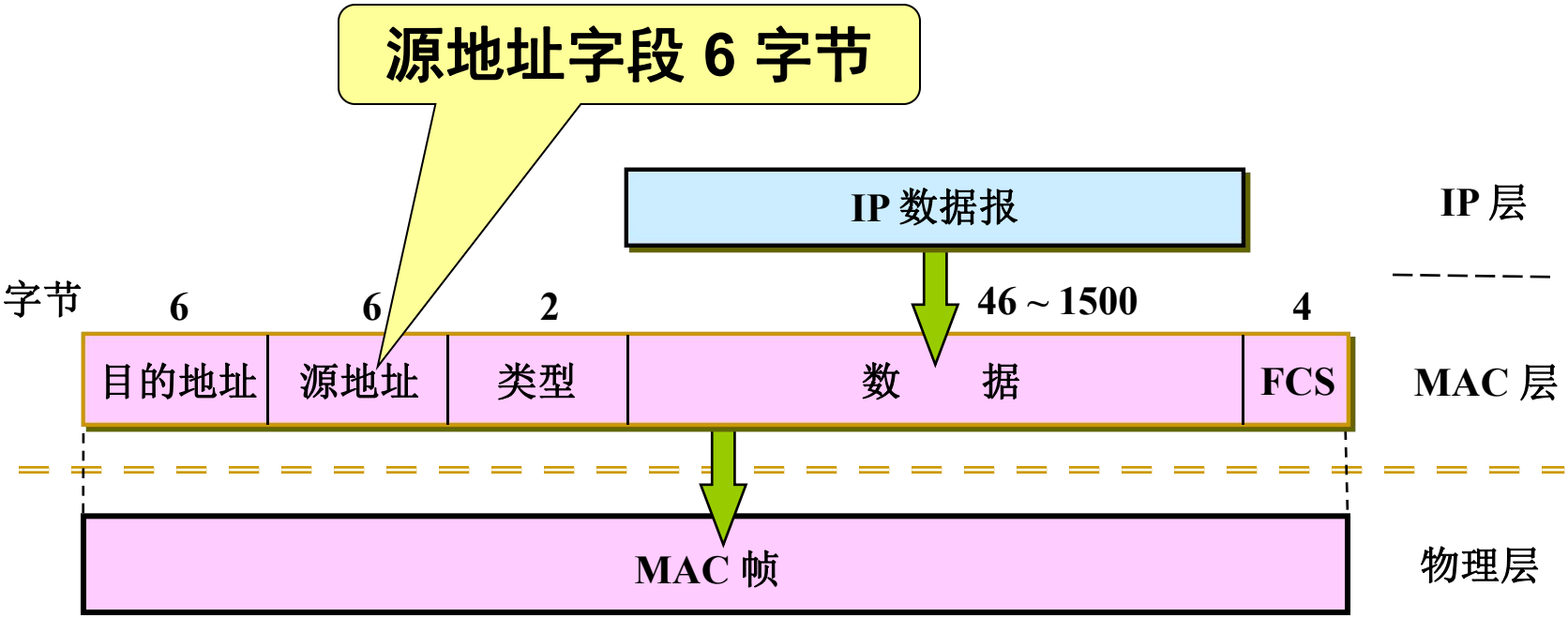
以太网V2的MAC帧的格式



MAC 帧目的地址

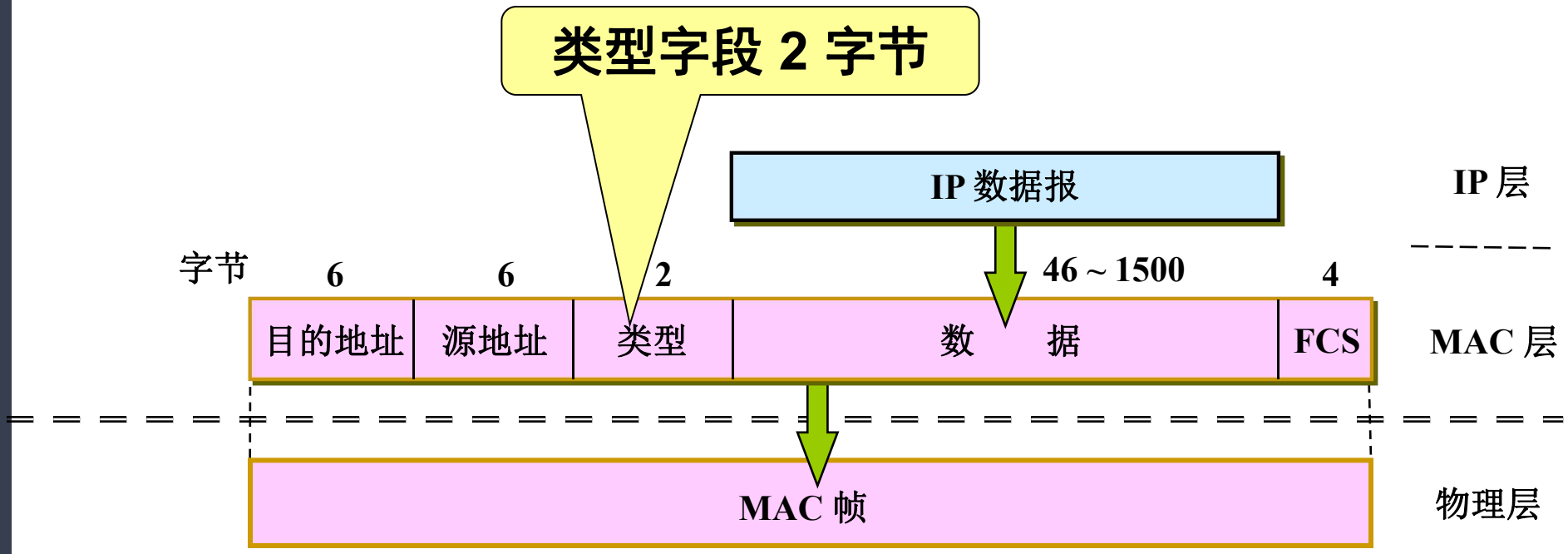

目的地址字段 6 字节
IP 数据报
IP 层
字节
6
6
2
46 ~ 1500
4
目的地址
源地址
类型
数
据
FCS
MAC 层
MAC 帧
物理层

MAC 帧源地址


源地址字段 6 字节
IP 数据报
IP 层
字节
6
6
2
46 ~ 1500
4
目的地址
源地址
类型
数
据
FCS
MAC 层
MAC 帧
物理层

MAC 帧协议类型格式

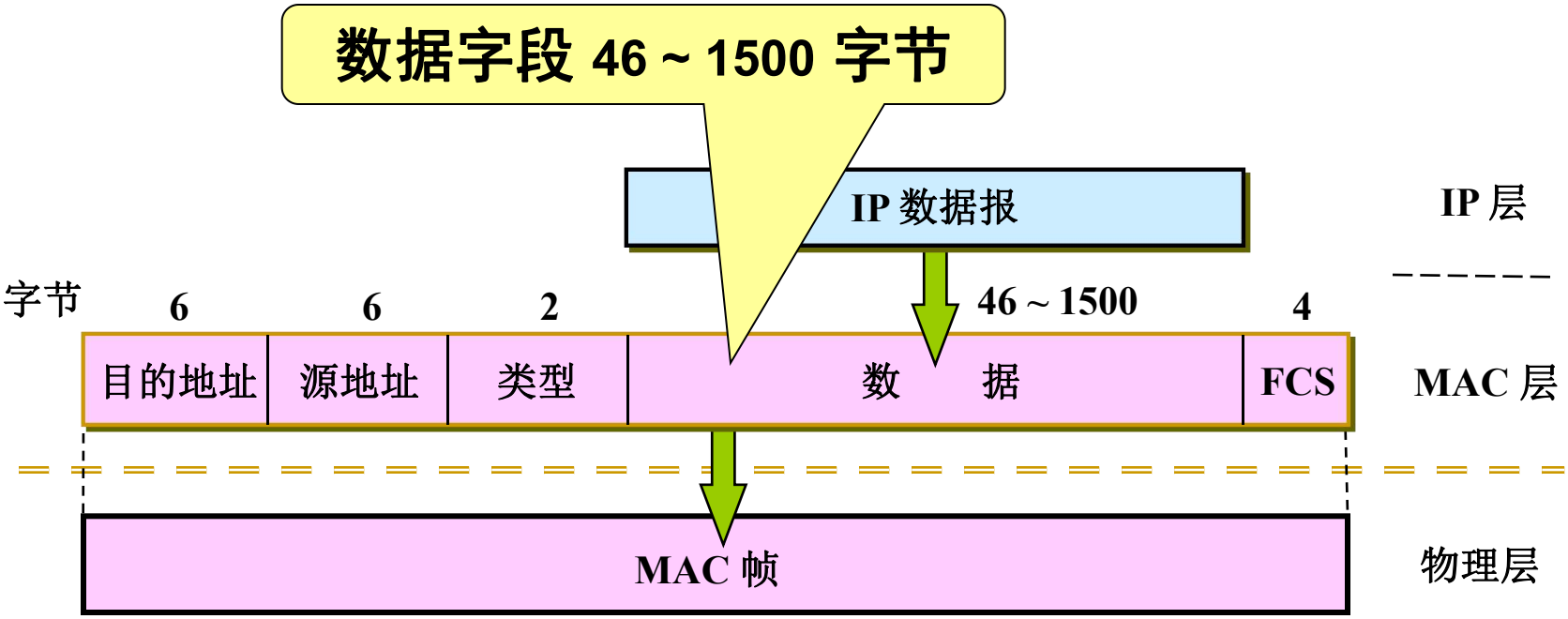
类型字段用来标志上一层使用的是什么协议，以便把收到的 MAC 帧的数据上交给上一层的这个协议。



MAC 帧数据字段

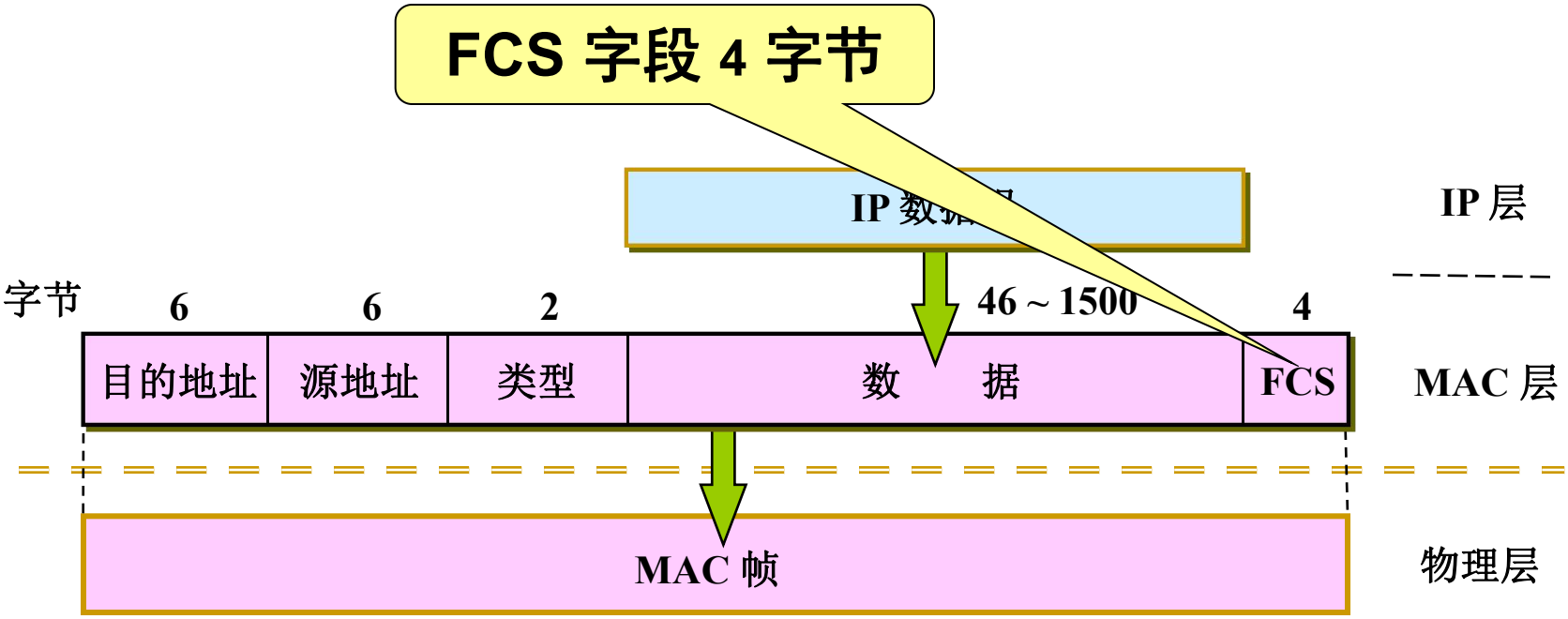
数据字段的正式名称是 MAC 客户数据字段

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



MAC 帧的FCS字段

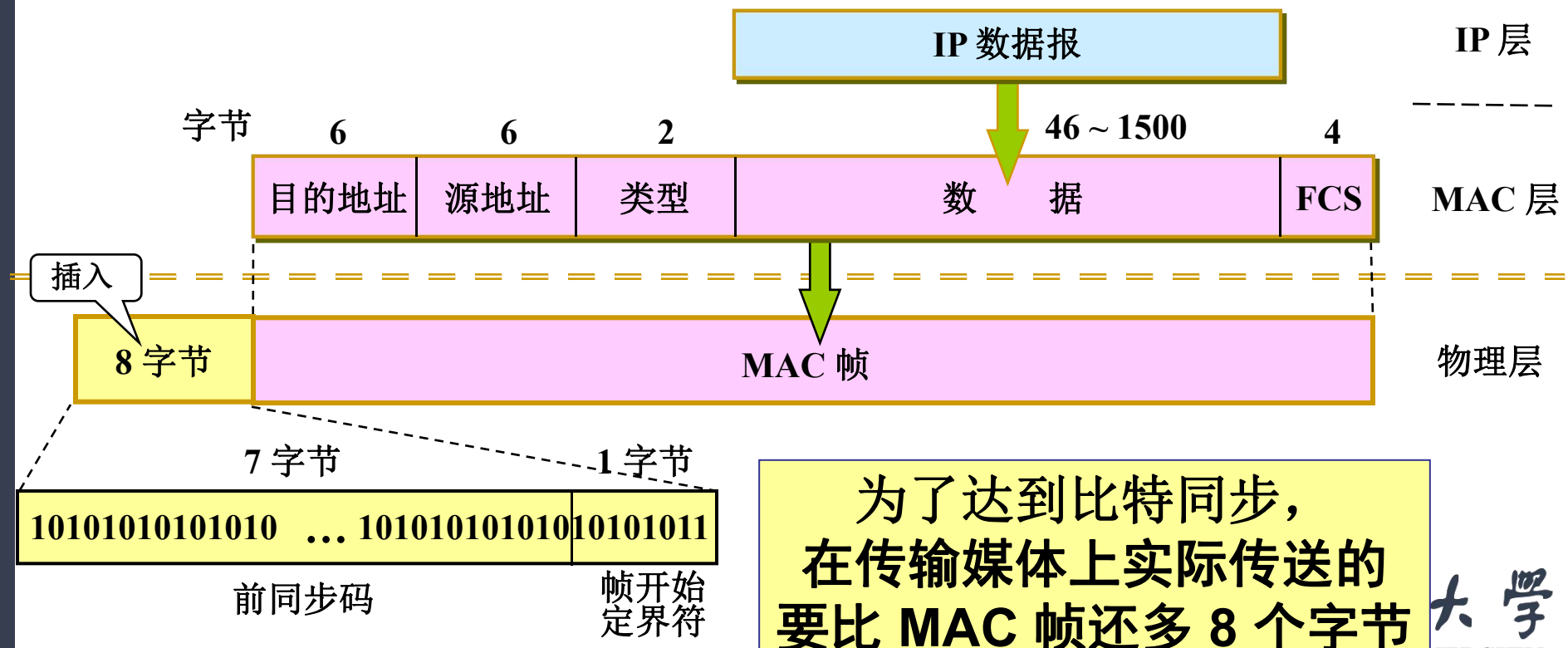
当传输媒体的误码率为 1×10^{-8} 时，
MAC 子层可使未检测到的差错小于 1×10^{-14}。



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 帧完整格式

在帧的前面插入的 8 字节中的第一个字段共 7 个字节，是前同步码，用来迅速实现 MAC 帧的比特同步。
第二个字段是帧开始定界符，表示后面的信息就是MAC 帧。



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

无效的 MAC 帧

- 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。

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。

扩展以太网

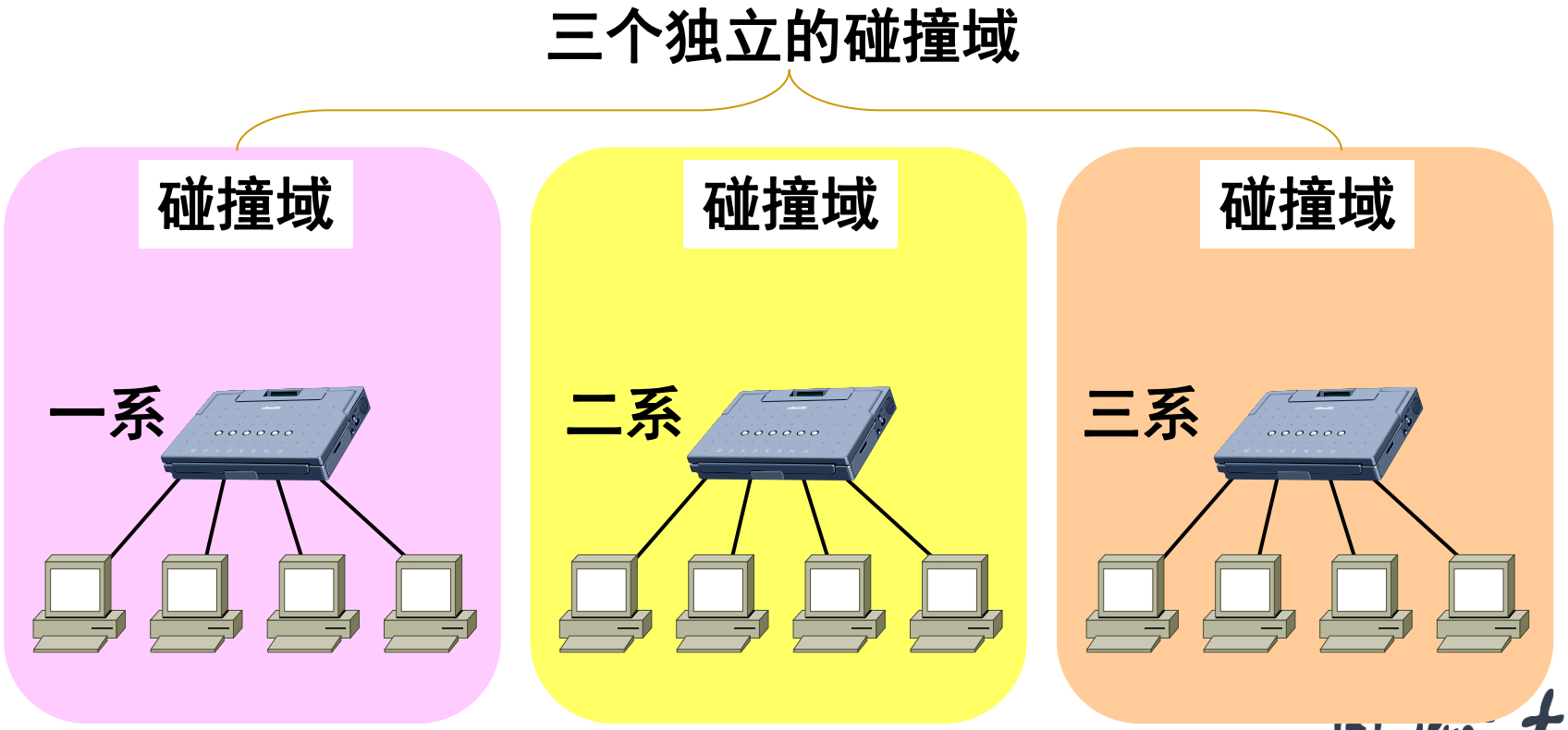
- 在物理层扩展以太网（集线器）
- 在数据链路层扩展以太网（网桥）

集线器的一些特点

- 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。
- 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。
- 集线器很像一个多接口的转发器，工作在物理层。

多个集线器可连成更大局域网

- 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。

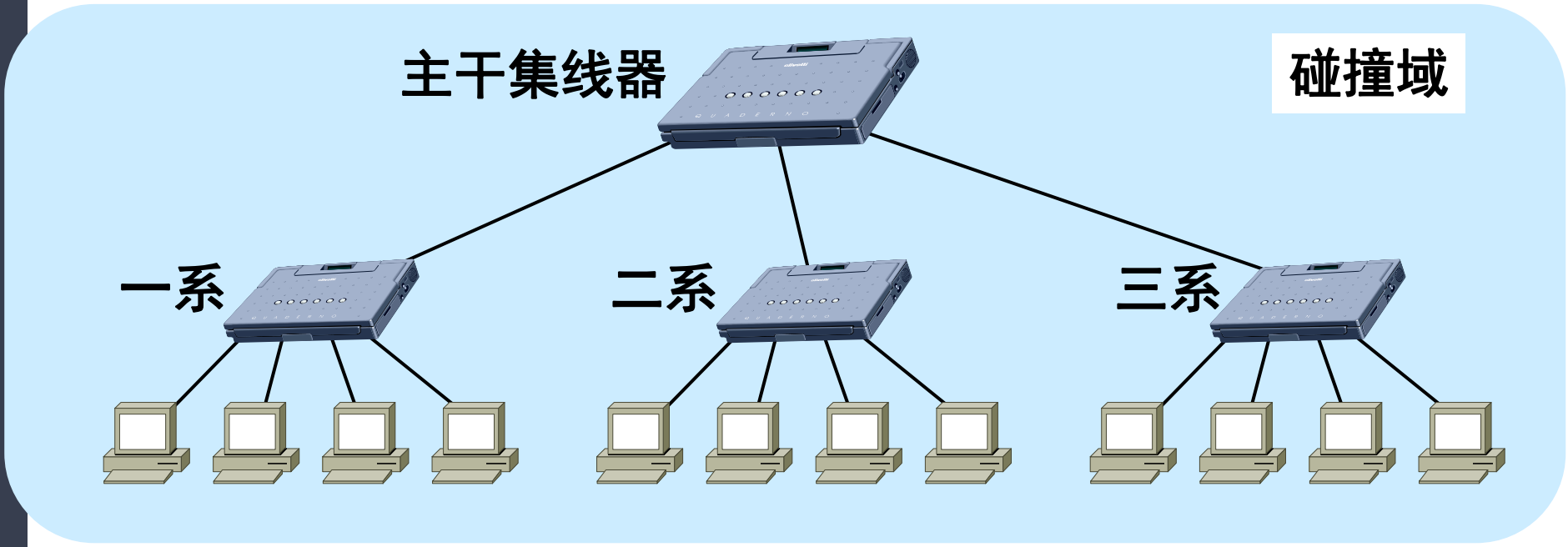


- **某大学有三个系，各自有一个局域网**

多个集线器可连成更大局域网

用集线器组成更大的局域网都在一个碰撞域中

一个更大的碰撞域



计算机之间的最大距离可以达到400米。

用集线器扩展局域网

- 优点
  - 使原来属于不同碰撞域的局域网上的计算机能够进行跨碰撞域的通信。
  - 扩大了局域网覆盖的地理范围。
- 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。

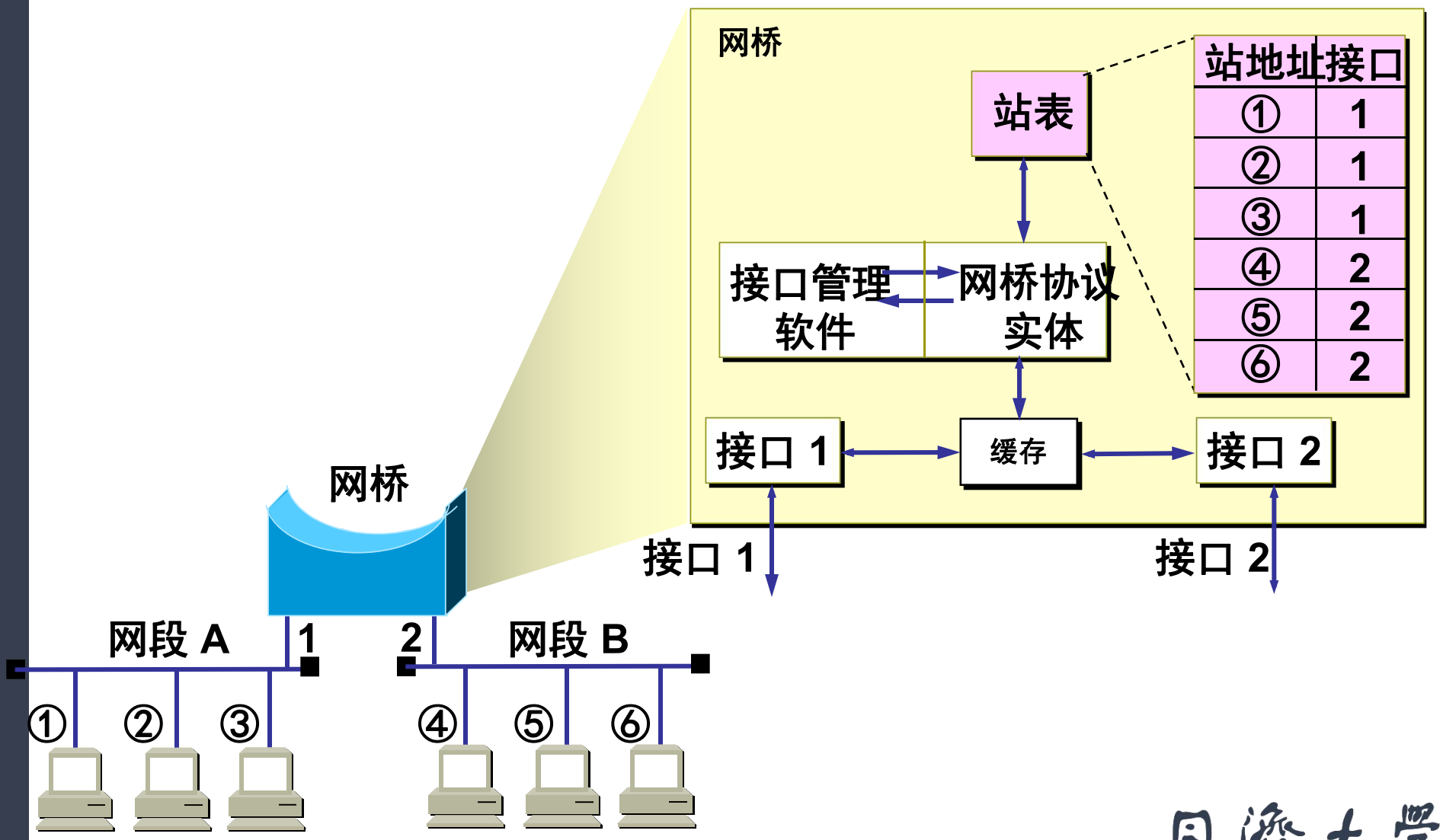
在数据链路层扩展局域网

- 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，现在使用以太网交换机。

- **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。**

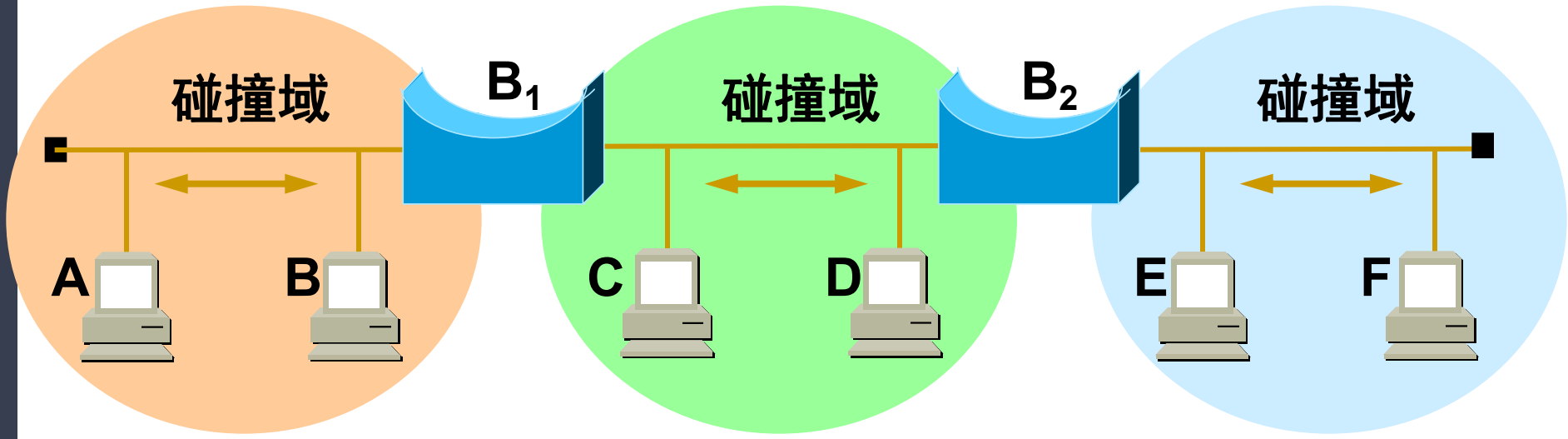
网桥的内部结构


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

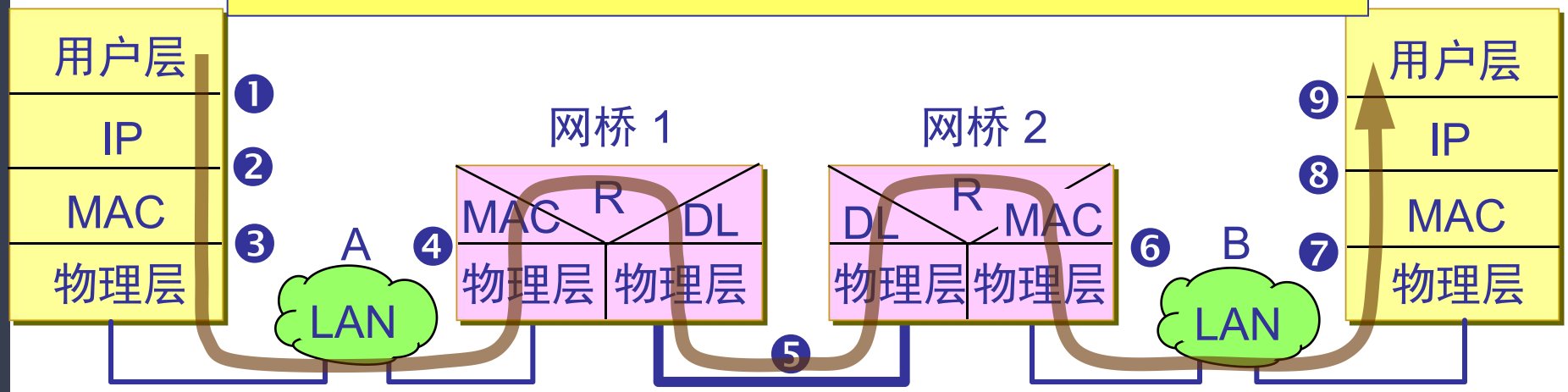
使用网桥的优点

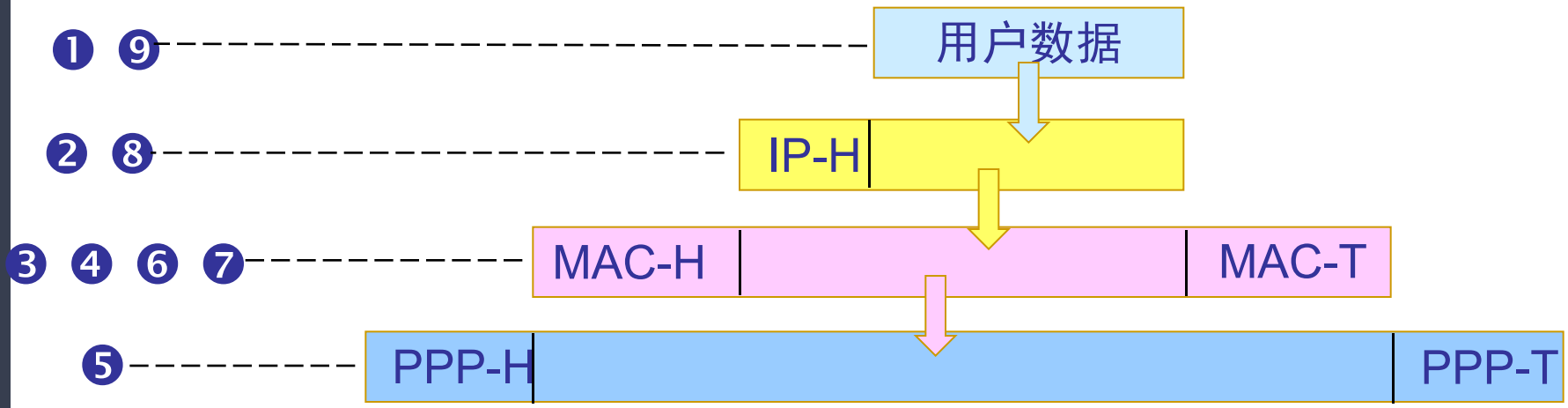
- 过滤通信量。
- 扩大了物理范围。
- 提高了可靠性。
- 可互连不同物理层、不同 MAC 子层和不同速率（如10 Mb/s 和 100 Mb/s 以太网）的局域网。

网桥使各网段成为隔离开的碰撞域



两个网桥之间还可使用一段点到点链路





网桥不改变它转发的帧的源地址

多接口网桥——以太网交换机

- 1990年问世的交换式集线器(switching hub)，可明显地提高局域网的性能。
- 交换式集线器常称为以太网交换机(switch)或第二层交换机（表明此交换机工作在数据链路层）。
- 以太网交换机通常都有十几个接口。因此，以太网交换机实质上就是一个多接口的网桥，可见交换机工作在数据链路层。

以太网交换机的特点

- 以太网交换机实质上就是一个多接口的网桥。
  - 通常都有十几个或更多的接口
- 每个接口都直接与一个单台主机或另一个以太网交换机相连，并且一般工作在全双工方式。
- 以太网交换机具有并行性。
  - 能同时连通多对接口，使多对主机能同时通信。
- 相互通信的主机都是独占传输媒体，无碰撞地传输数据。

以太网交换机的特点

- 以太网交换机的接口有存储器，能在输出端口繁忙时把到来的帧进行缓存。
- 以太网交换机是一种即插即用设备，其内部的帧交换表（又称为地址表）是通过自学习算法自动地逐渐建立起来的。
- 以太网交换机使用了专用的交换结构芯片，用硬件转发，其转发速率要比使用软件转发的网桥快很多。

独占传输媒体的带宽

- 对于普通 10 Mb/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 Mb/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 Mb/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 对接口的交换机的总容量为 $N \times 10$ Mb/s。这正是交换机的最大优点。

以太网交换机的交换方式

- 存储转发方式
  - 把整个数据帧先缓存后再进行处理。
- 直通 (cut-through) 方式
  - 接收数据帧的同时就立即按数据帧的目的 MAC 地址决定该帧的转发接口，因而提高了帧的转发速度。
  - 缺点是它不检查差错就直接将帧转发出去，因此有可能也将一些无效帧转发给其他的站。

在某些情况下，仍需要采用基于软件的存储转发方式进行交换，例如，当需要进行线路速率匹配、协议转换或差错检测时。

以太网交换机的自学习功能

- 以太网交换机运行自学习算法自动维护交换表。
- 开始时，以太网交换机里面的交换表是空的。

交换表一开始是空的

自学习算法

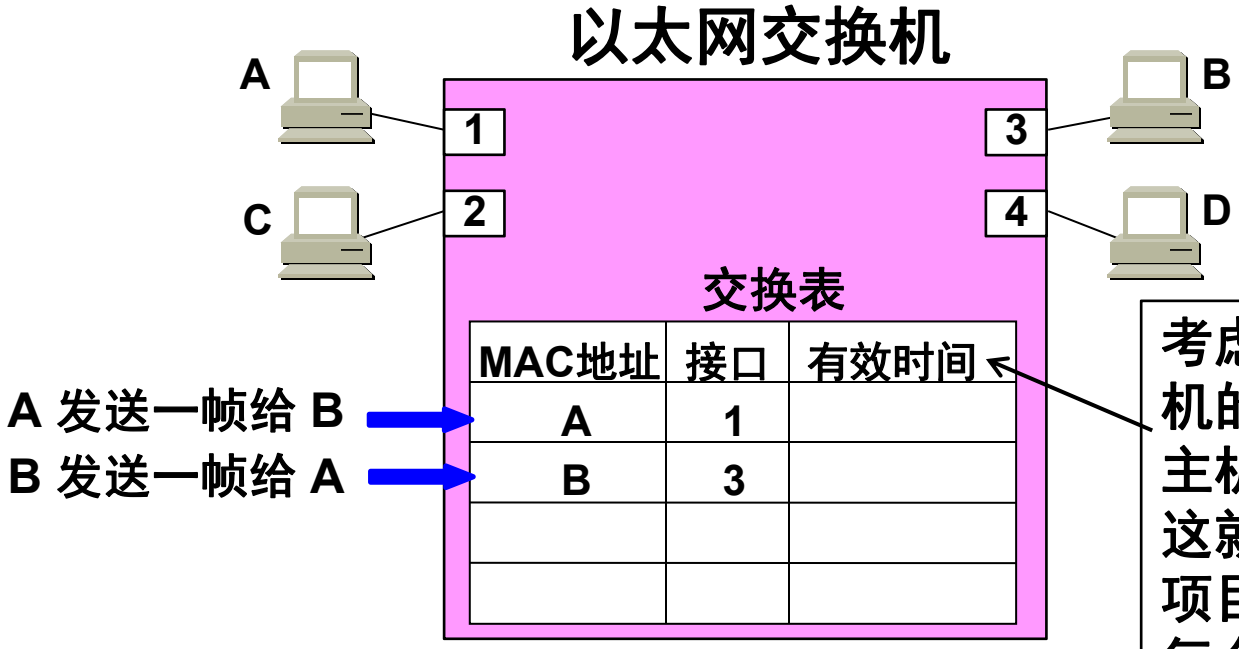
按照以下自学习算法处理收到的帧和建立交换表:

- A 先向 B 发送一帧，从接口 1 进入到交换机。
- 交换机收到帧后，先查找交换表，没有查到应从哪个接口转发这个帧。
- 交换机把这个帧的源地址 A 和接口 1 写入交换表中，并向除接口1以外的所有的接口广播这个帧。
- C 和 D 将丢弃这个帧，因为目的地址不对。只 B 才收下这个目的地址正确的帧。这也称为过滤。
- 从新写入交换表的项目 (A, 1) 可以看出，以后不管从哪一个接口收到帧，只要其目的地址是A，就应当把收到的帧从接口1转发出去。

自学习算法

- B 通过接口 3 向 A 发送一帧。
- 交换机查找交换表，发现交换表中的 MAC 地址有 A。表明要发送给A的帧（即目的地址为 A 的帧）应从接口1转发。于是就把这个帧传送到接口 1 转发给 A。显然，现在已经没有必要再广播收到的帧。
- 交换表这时新增加的项目 (B, 3)，表明今后如有发送给 B 的帧，就应当从接口 3 转发出去。
- 经过一段时间后，只要主机 C 和 D 也向其他主机发送帧，以太网交换机中的交换表就会把转发到 C 或 D 应当经过的接口号（2 或 4）写入到交换表中。

自学习算法



交换了两帧后的交换表

考虑到可能有时要在交换机的接口更换主机，或者主机要更换其网络适配器，这就需要更改交换表中的项目。为此，在交换表中每个项目都设有一定的有效时间。过期的项目就自动被删除。

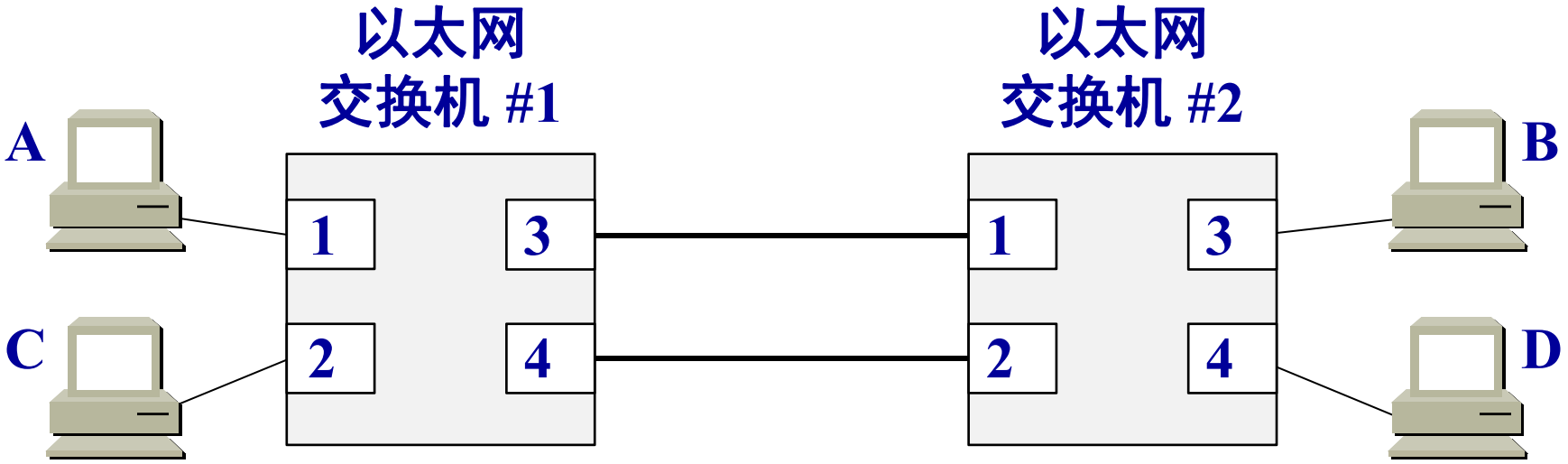
以太网交换机的这种自学习方法使得以太网交换机能够即插即用，不必人工进行配置，因此非常方便。

交换机自学习和转发帧的步骤归纳

- 交换机收到一帧后先进行自学习。查找交换表中与收到帧的源地址有无相匹配的项目。
  - 如没有，就在交换表中增加一个项目（源地址、进入的接口和有效时间）。
  - 如有，则把原有的项目进行更新（进入的接口或有效时间）。
- 转发帧。查找交换表中与收到帧的目的地址有无相匹配的项目。
  - 如没有，则向所有其他接口（进入的接口除外）转发。
  - 如有，则按交换表中给出的接口进行转发。
  - 若交换表中给出的接口就是该帧进入交换机的接口，则应丢弃这个帧（因为这时不需要经过交换机进行转发）。

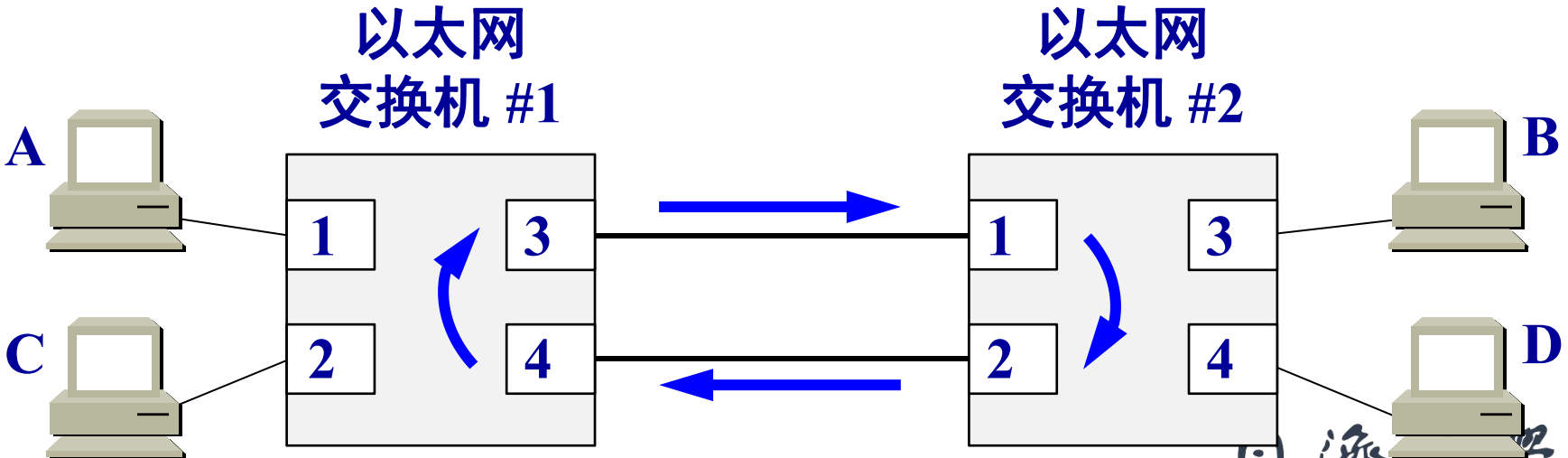
交换机使用了生成树协议

- 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。

本章重点

- 数据链路层的三个基本问题
- 数据链路层信道类型，分别采用什么协议，各个不同协议各自的特点
- CSMA/CD协议
- MAC帧格式
- 网桥、交换机

同济大学
TONGJI UNIVERSITY

NaMI
网络与机器智能实验室

Thanks for your attention !