

第4章 数据链路层

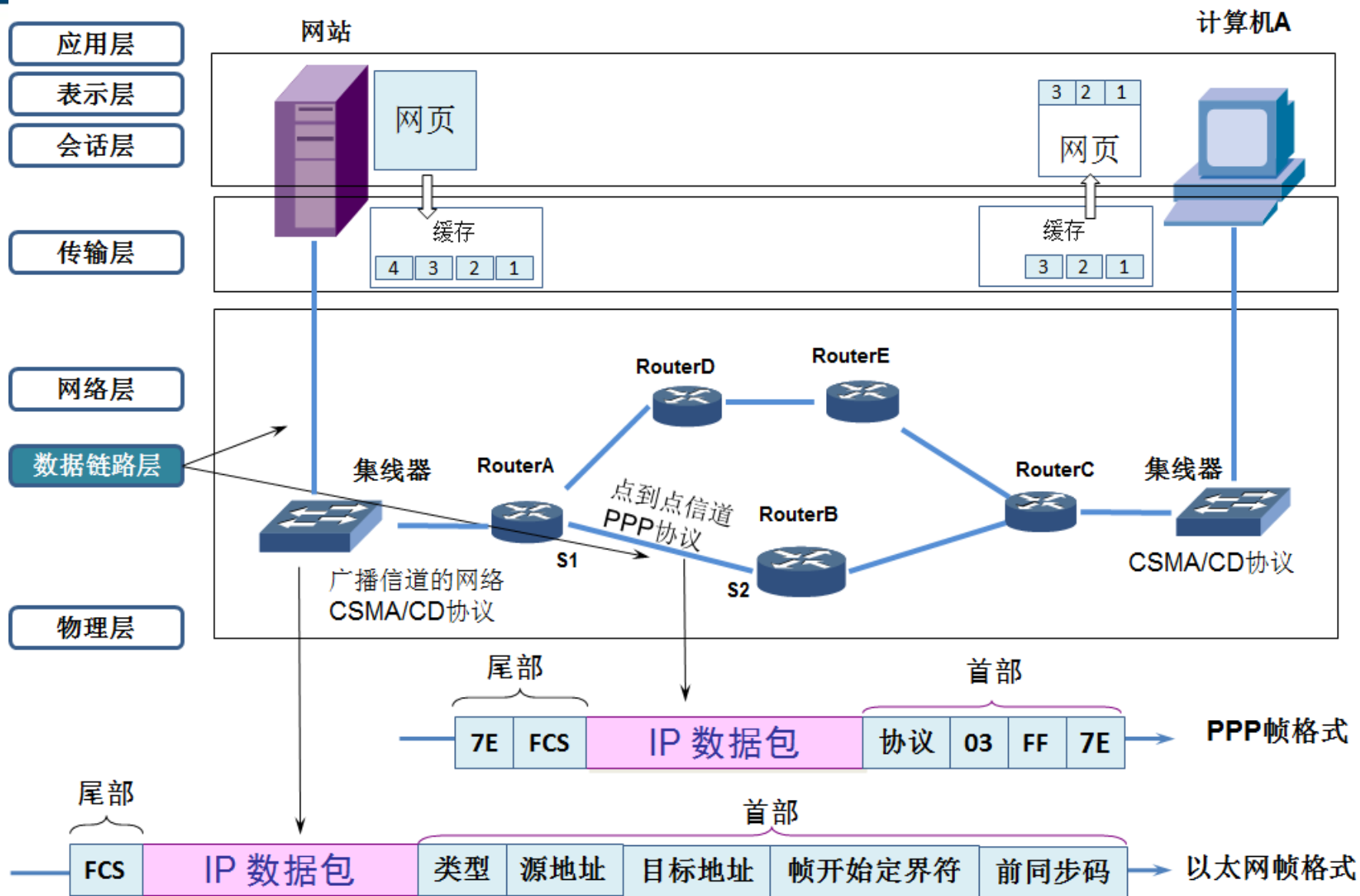


韩老师QQ 458717185
韩老师视频课程学习路线
www.91xueit.com
韩老师博客
<http://91xueit.blog.51cto.com>

讲师：韩立刚
河北师大软件学院讲师
微软最有价值专家（MVP）
微软企业护航专家（ESS）

2016年11月10日上午录制

本章图例



本章内容

■ 4.1数据链路层的三个基本问题

- 封装成帧、透明传输和差错检测

■ 4.2点到点信道的数据链路

■ 4.3广播信道的数据链路

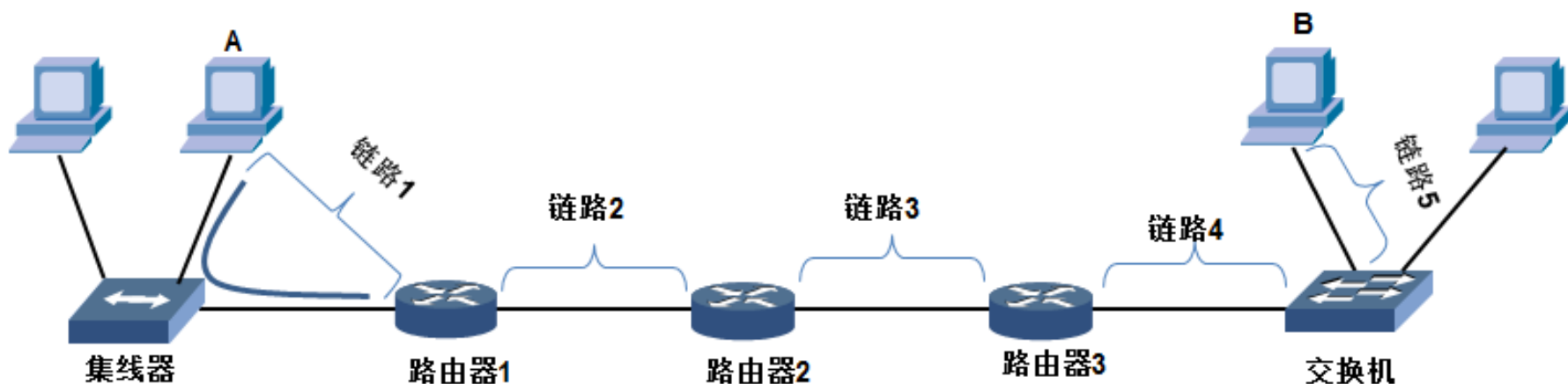
■ 4.4扩展以太网

■ 4.5高速以太网

4.1数据链路层的三个基本问题

■ 数据链路和帧

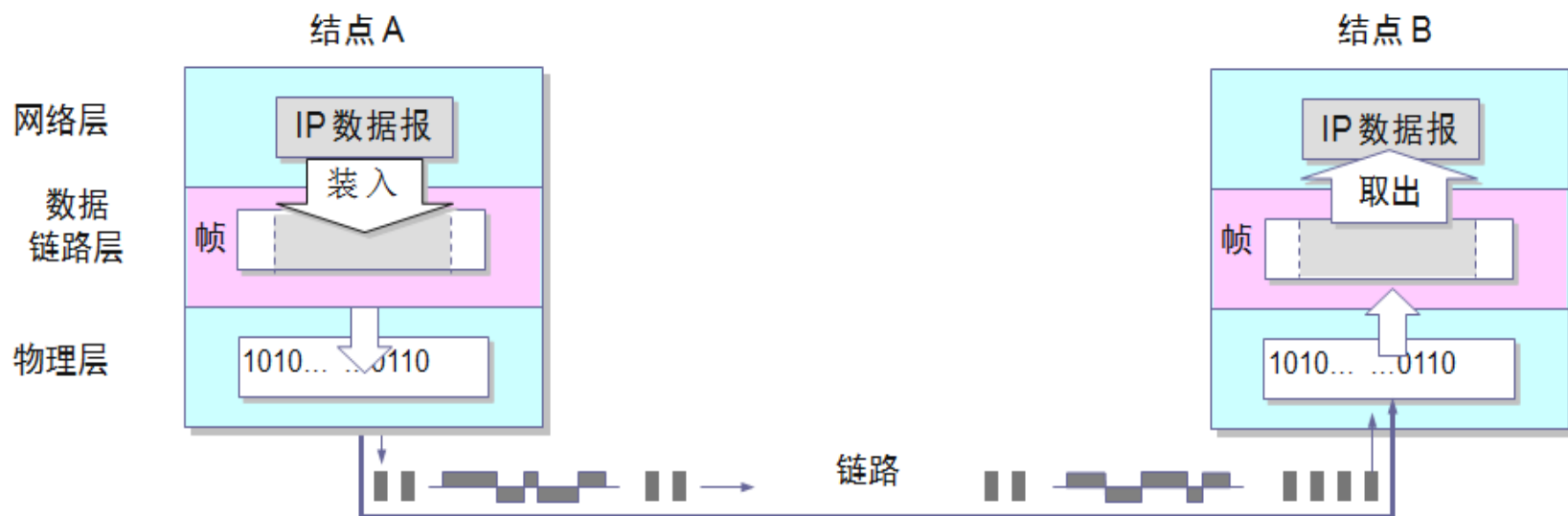
- 链路（Link）是指的从一个节点到相邻节点的一段物理线路（有线或无线），而中间没有任何其他的交换节点。
- 数据链路（Data Link）则是另一个概念，这是因为当需要在一条线路上传送数据时，除了必须有一条物理线路外，还必须有一些必要的通信协议来控制这些数据的传输。



4.1数据链路层的三个基本问题

■ 数据链路和帧

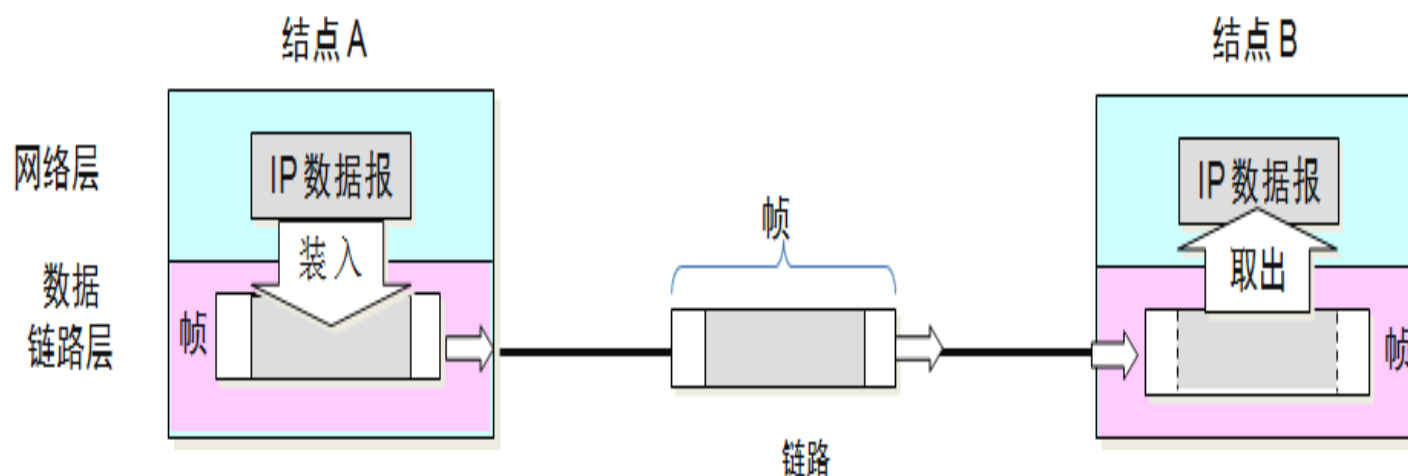
- 数据链路层把网络层交下来的数据封装成帧发送到链路上，以及把接收到的帧中的数据取出并上交给网络层。在因特网中，网络层协议数据单元就是IP数据报（或简称为数据报、分组或包）。数据链路层封装的帧，在物理层变成数字信号在链路上传输。



4.1数据链路层的三个基本问题

■ 数据链路和帧

- 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。



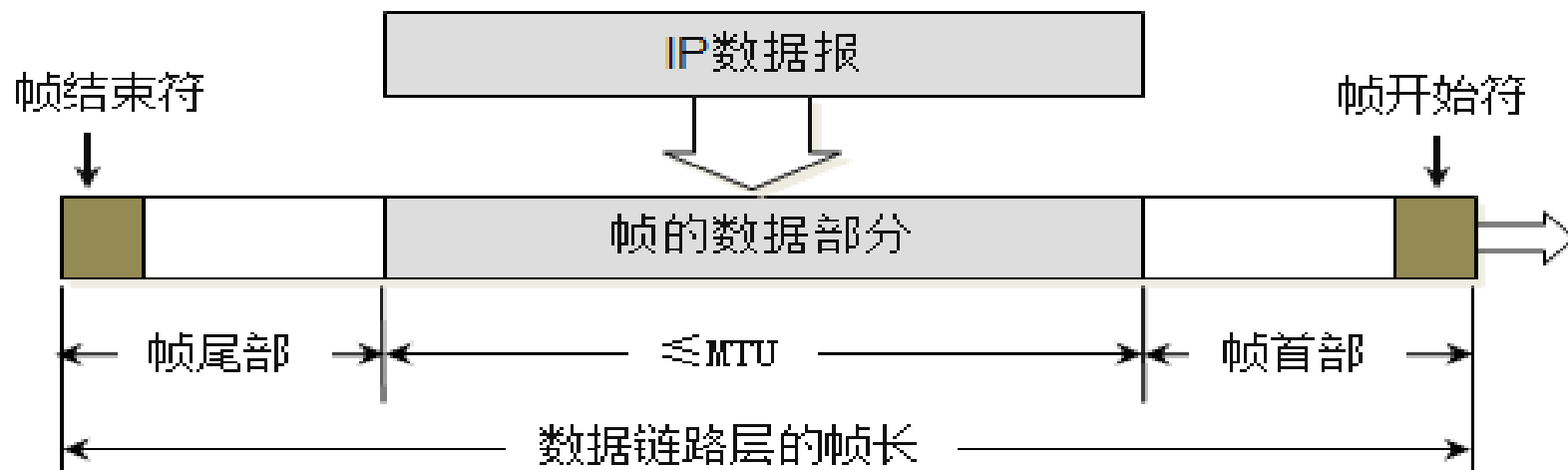
4.1数据链路层三个基本问题

■ 封装成帧

- 封装成帧，就是在将网络层的IP数据报的前后分别添加首部和尾部，这样就构成了一个帧。
- 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。

4.1数据链路层三个基本问题

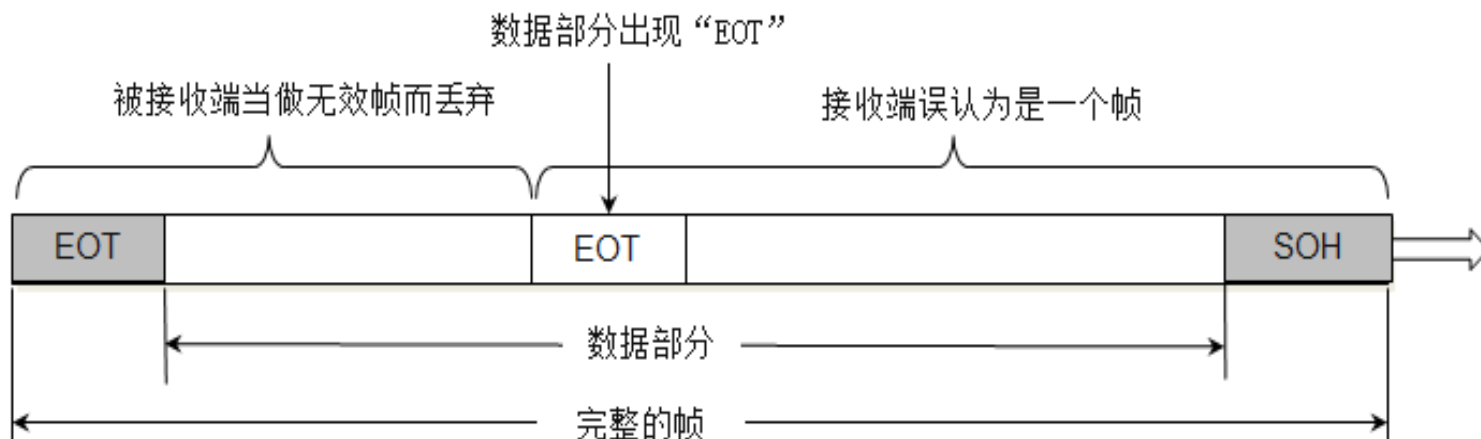
■ 封装成帧



4.1数据链路层三个基本问题

■ 透明传输

- 帧开始符和帧结束符最好是不会出现在帧的数据部分的字符，通常我们电脑键盘能够输入的字符是ASCII字符代码表中打印字符，在ASCII字符代码表中，还有非打印控制字符，在非打印字符中有两个字符专门用来做帧定界符，代码SOH（Start Of Header）作为帧开始定界符，对应的二进制编码为0000 0001，代码EOT（End Of Transmission）作为帧结束定界符。



ASCII字符码表

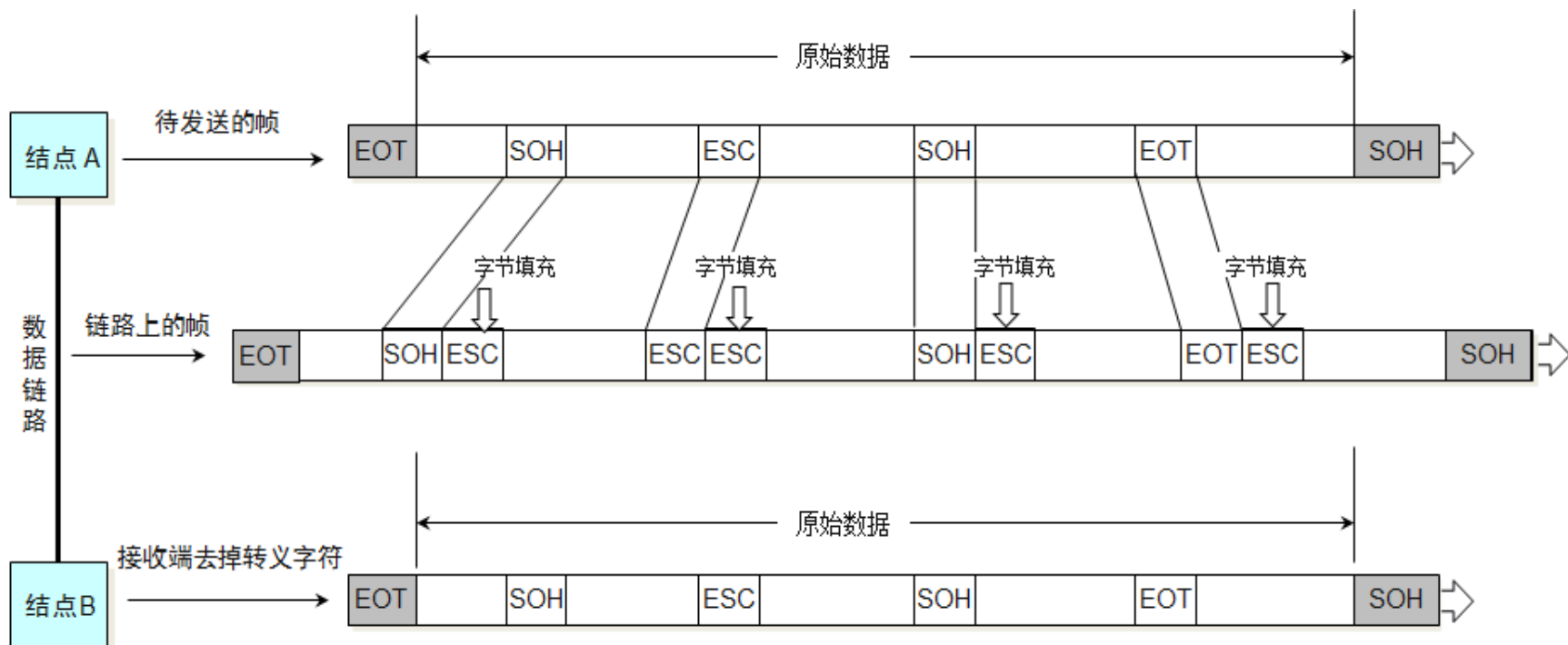
ASCII 字符代码表

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	坚直制表符	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		

4.1数据链路层三个基本问题

■ 透明传输

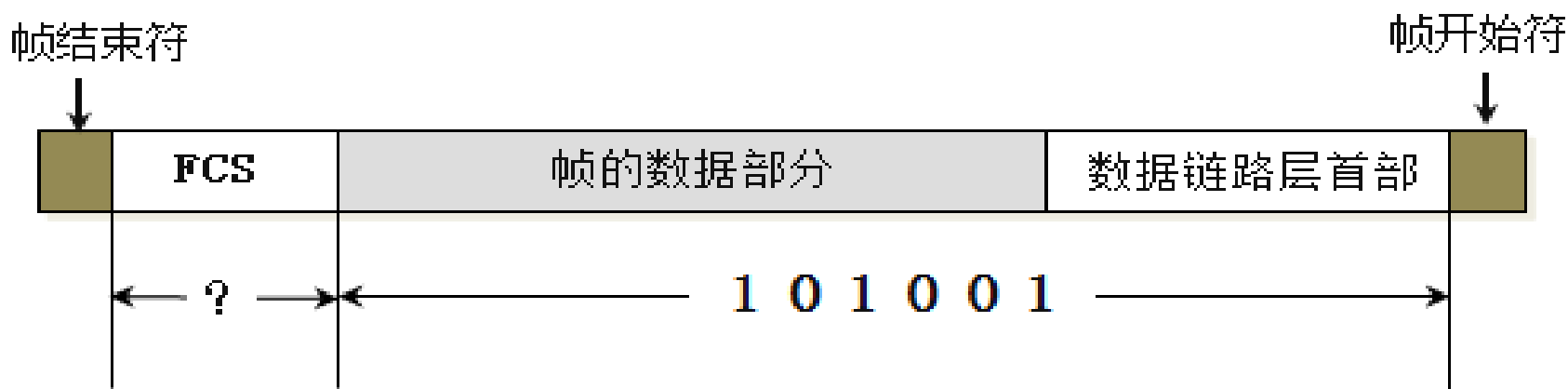
- 当数据部分是非ASCII字符代码表的文本文件时（比如二进制代码的计算机程序或图像等），情况就不同了。如果数据中的某一段二进制代码正好和SOH或EOT帧定界符编码一样，接收端就会误认为这就是帧的边界。



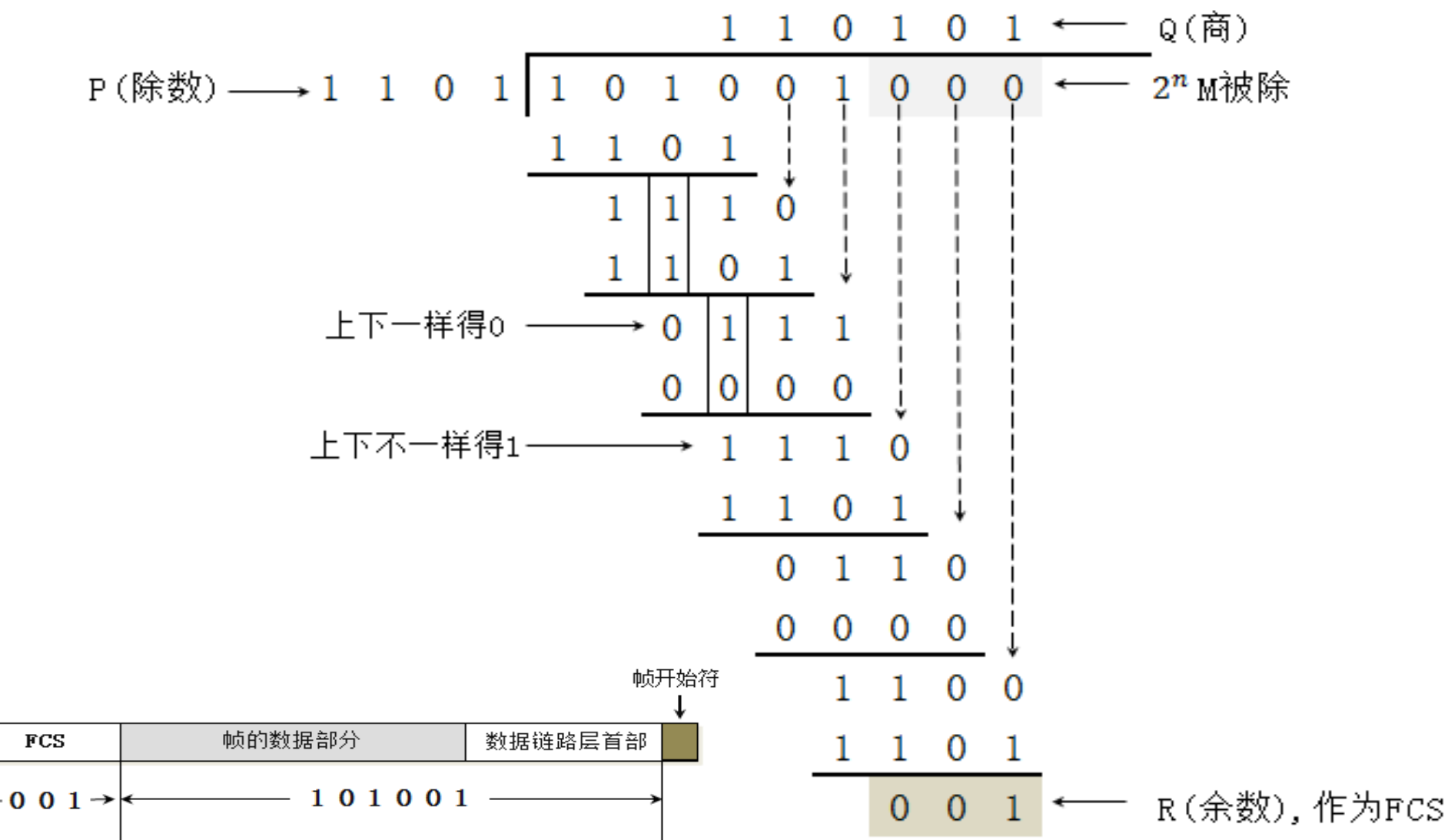
4.1数据链路层三个基本问题

■ 差错检验

- 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。
- 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。目前在数据链路层广泛使用了循环冗余检验CRC(Cyclic Redundancy Check) 的差错检验技术。

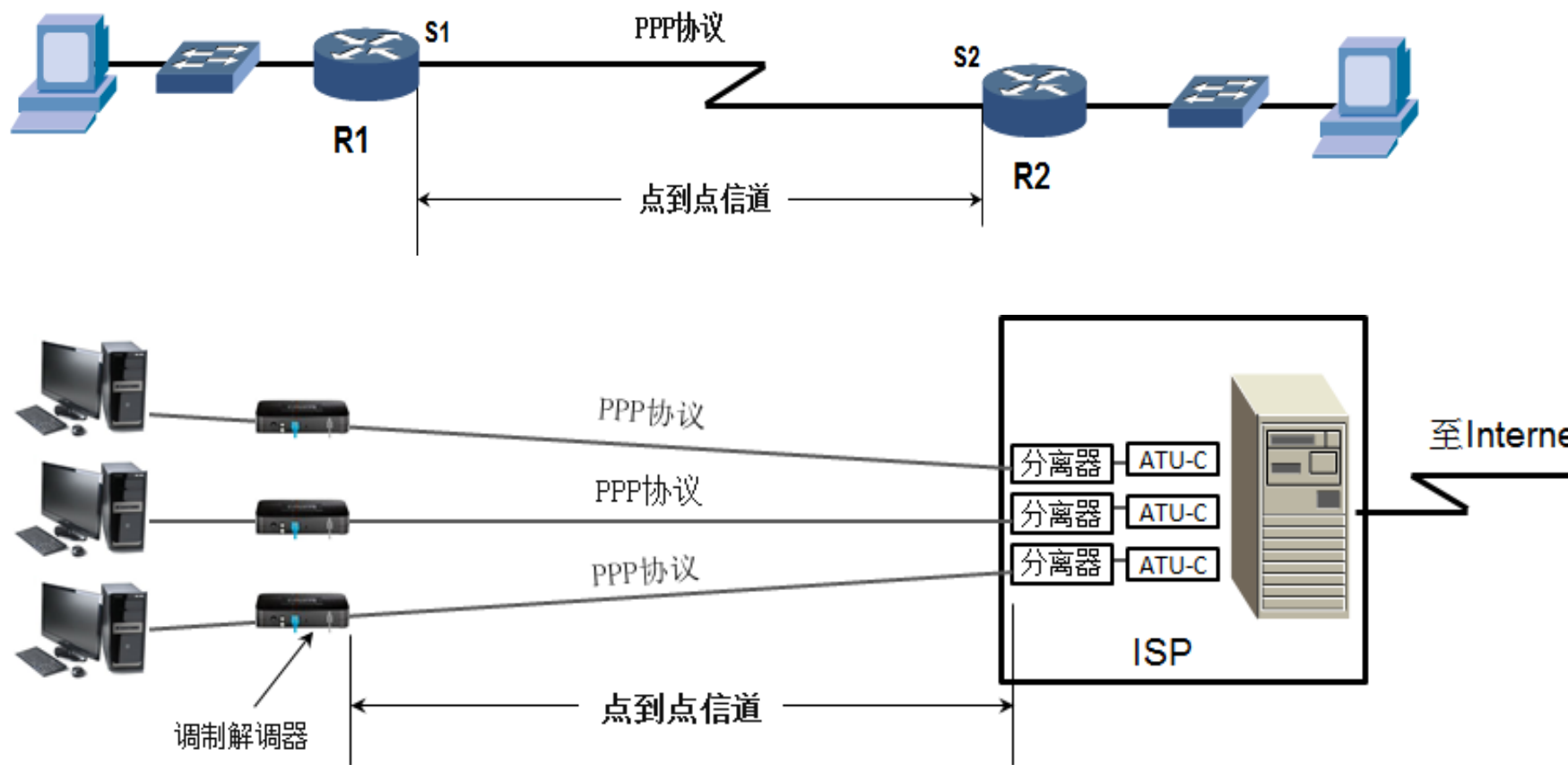


CRC运算示例



4.2点到点信道的数据链路

- 点到点信道是指的一条链路上就一个发送端和接收端的信道，通常用在广域网链路。

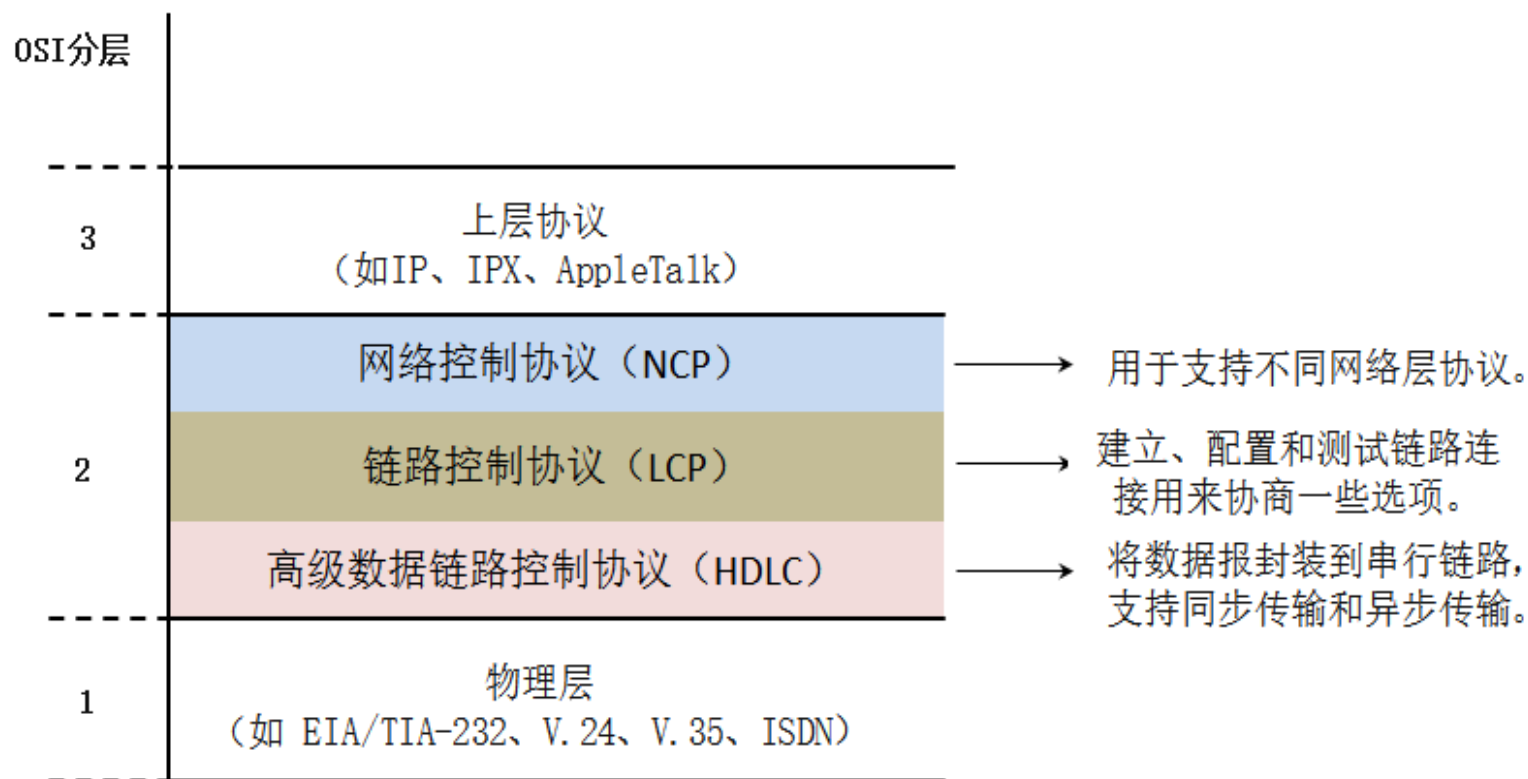


4.2.1 PPP协议的特点

- (1) 简单 不提供可靠传输
- (2) 封装成帧 首部和尾部 帧开始符 帧结束符
- (3) 透明传输 加转义字符 收到后去掉转移字符
- (4) 差错检测 CRC计算FCS
- (5) 支持多种网络层协议 IPv4和IPv6网络层协议都可以封装到PPP帧中
- (6) 多种类型链路 光纤 铜线 同步传输 异步传输 串行、并行链路均可
- (7) 检测连接状态 检测连接状态
- (8) 最大传送单元 最大传输单元 1500字节
- (9) 网络层地址协商 能够为拨号的一段分配IP地址，子网掩码 网关和DNS
- (10) 数据压缩协商

4.2.2 PPP协议的组成

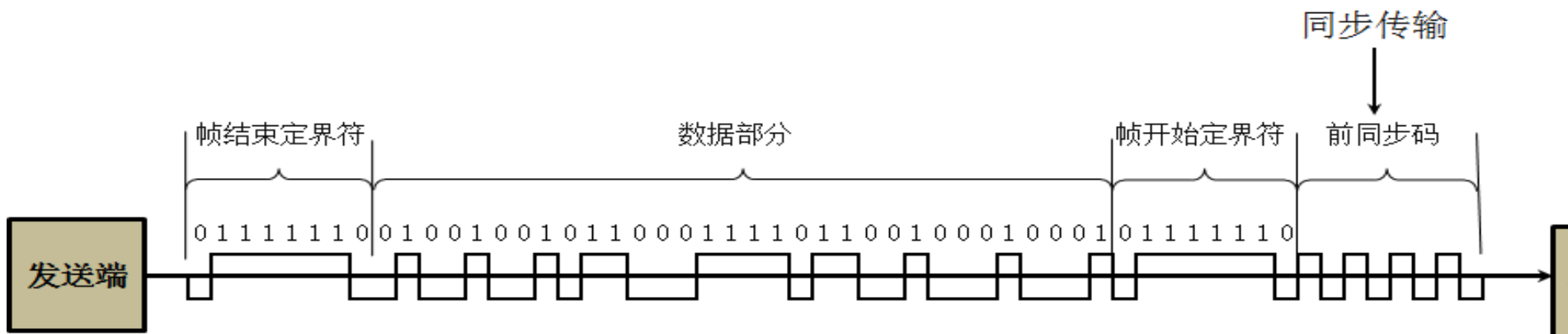
- PPP协议有三个组成部分：



4.2.3 同步传输和异步传输

■ 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。

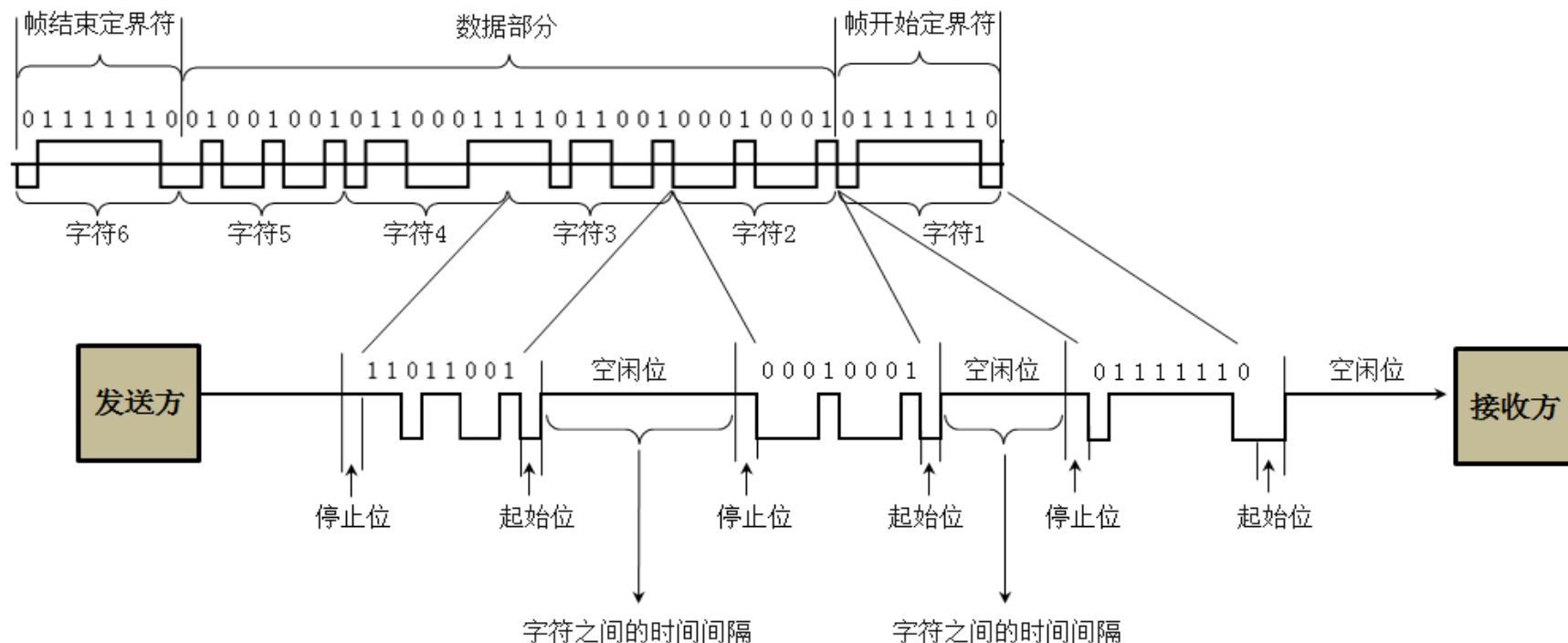


4.2.3 同步传输和异步传输

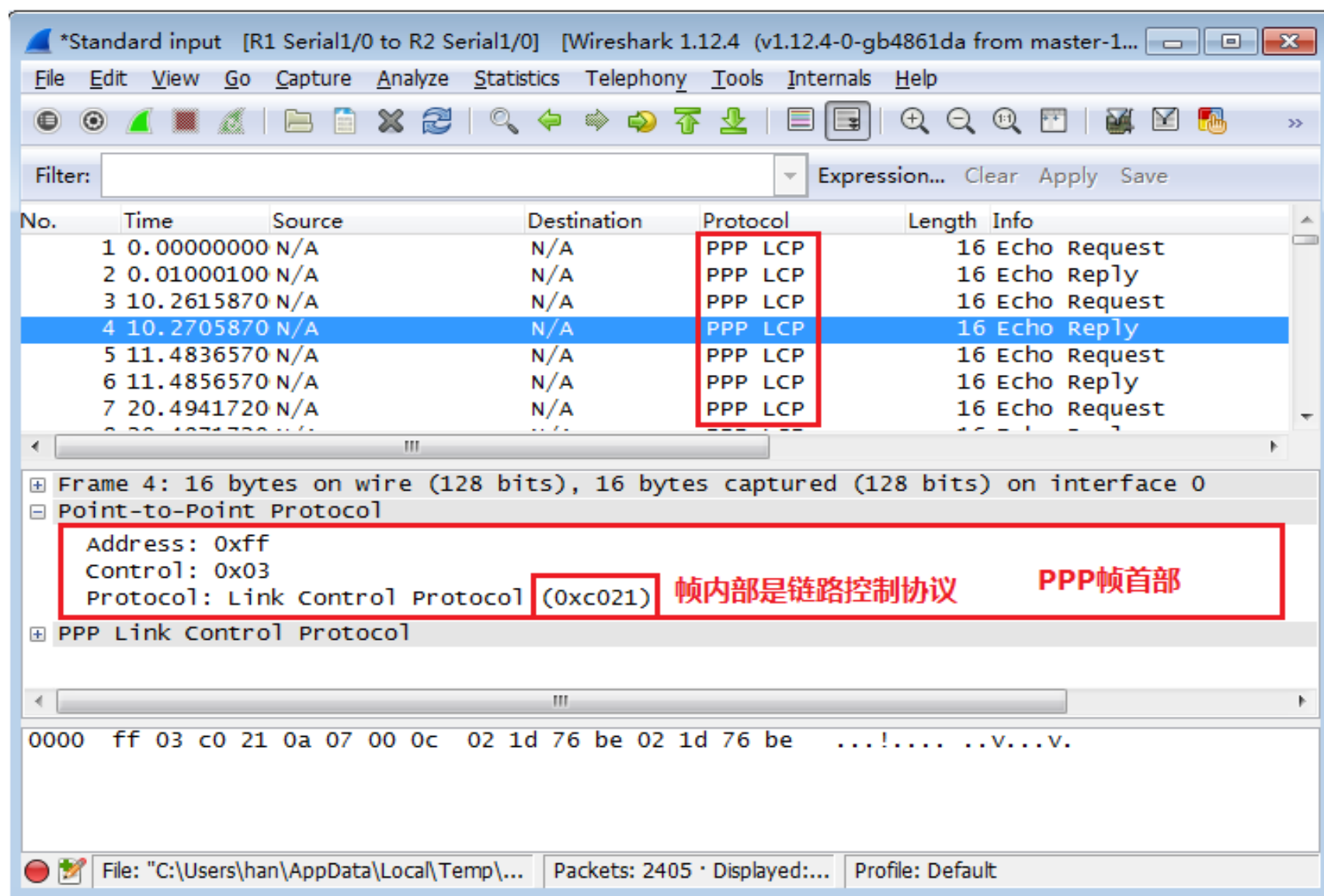
■ 异步传输

- 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），并且两者中任何一方都不向对方提供时钟同步信号。

异步传输示意图

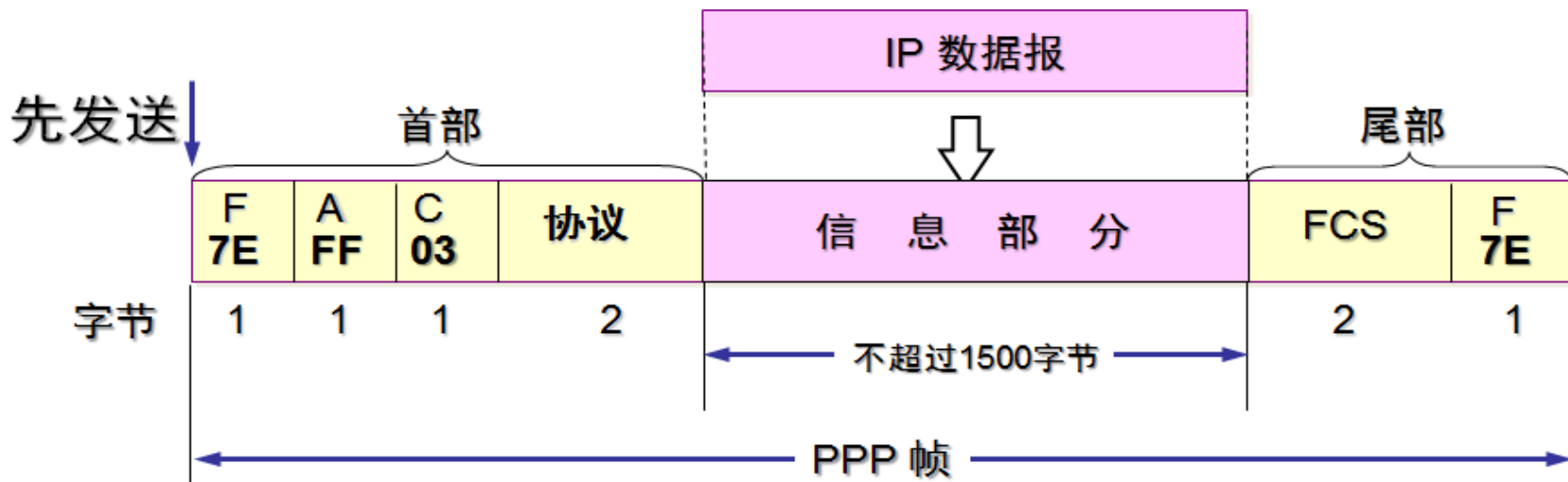


4.2.4抓包查看PPP的帧首部



4.2.5 PPP协议帧格式

- Address字段的值为0xff，0x表示后面的ff为十六进制数，写成二进制为1111 1111，占一个字节的长度。点到点信道PPP帧中的地址字段形同虚设，可以看到没有源地址和目标地址。
- Control字段的值为0x03，写成二进制为0000 0011，占一个字节长度。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。

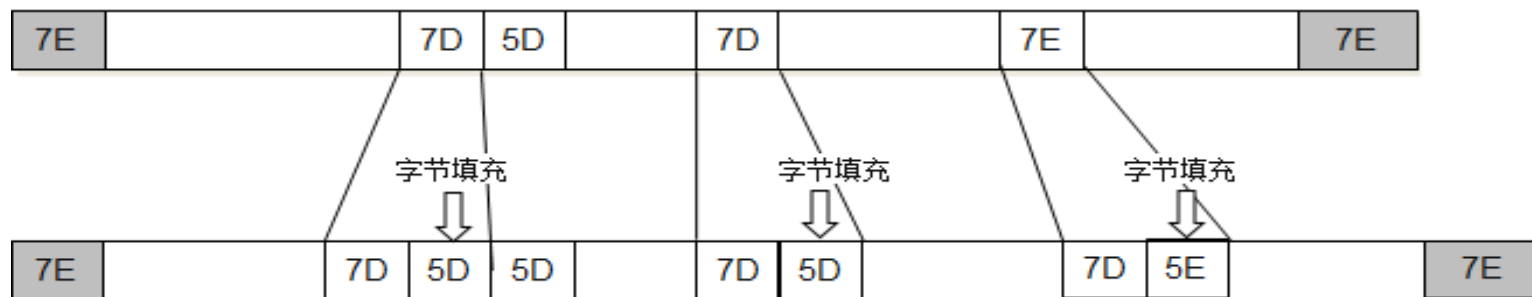


4.2.6 PPP帧填充方式

■ 异步传输使用字节填充

- 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。
- 把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（即出现了和转义字符一样的比特组合），则把0x7D转变成为2字节序列（0x7D，0x5D）。

PPP帧字节填充



4.2.6 PPP帧填充方式

■ 同步传输使用零比特填充

- 在同步传输的链路上，数据传输以帧为单位，PPP协议采用零比特填充方法来实现透明传输。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,这样就可以避免出现帧界定符。一旦在数据部分出现“零比特填充”

PPP帧零比特填充

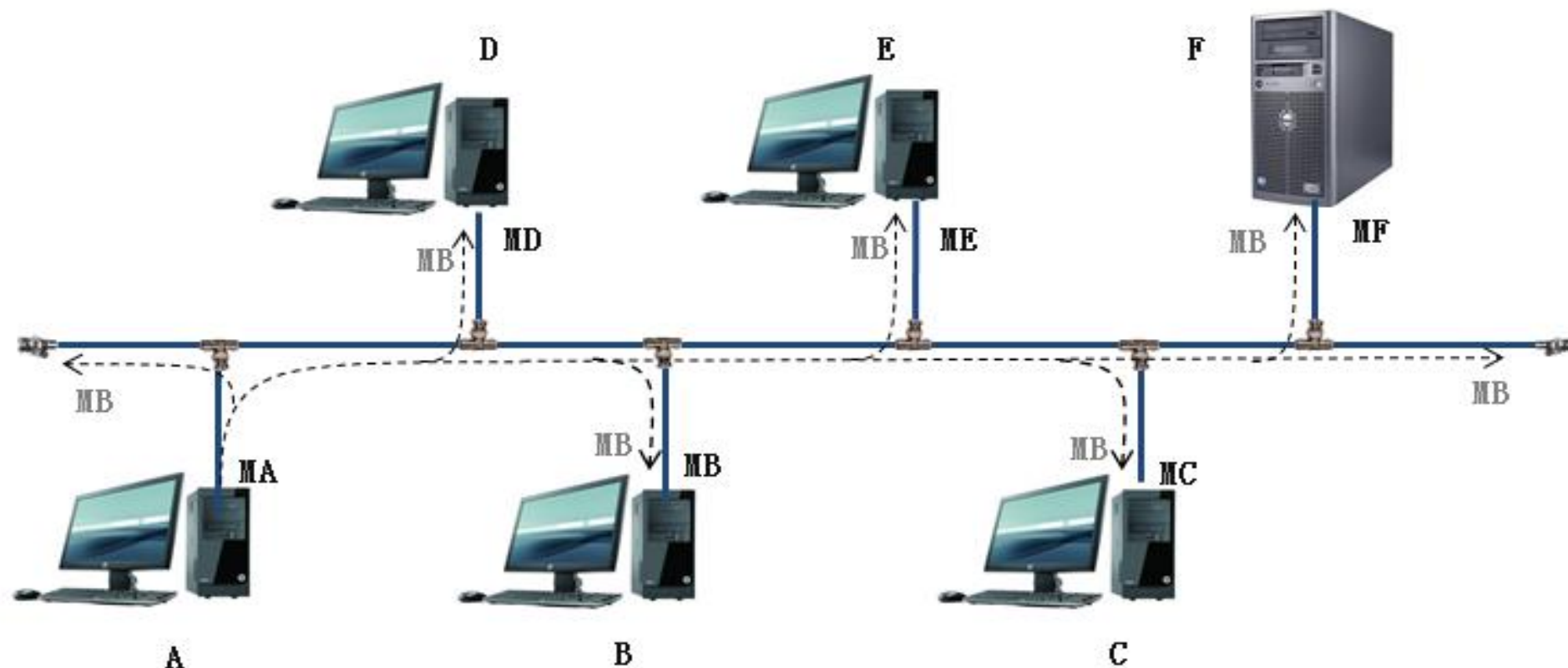


4.3广播信道的数据链路

- 4.3.1 广播信道的局域网
- 4.3.2 以太网标准
- 4.3.3 CSMA/CD协议
- 4.3.4 以太网最短帧
- 4.3.5 冲突解决方法--退避算法
- 4.3.6 以太网帧格式
- 4.3.7 以太网信道利用率
- 4.3.8 网卡的作用
- 4.3.9 MAC地址
- 4.3.10 实战：查看和更改MAC地址

4.3.1 广播信道的局域网

广播信道局域网—总线型

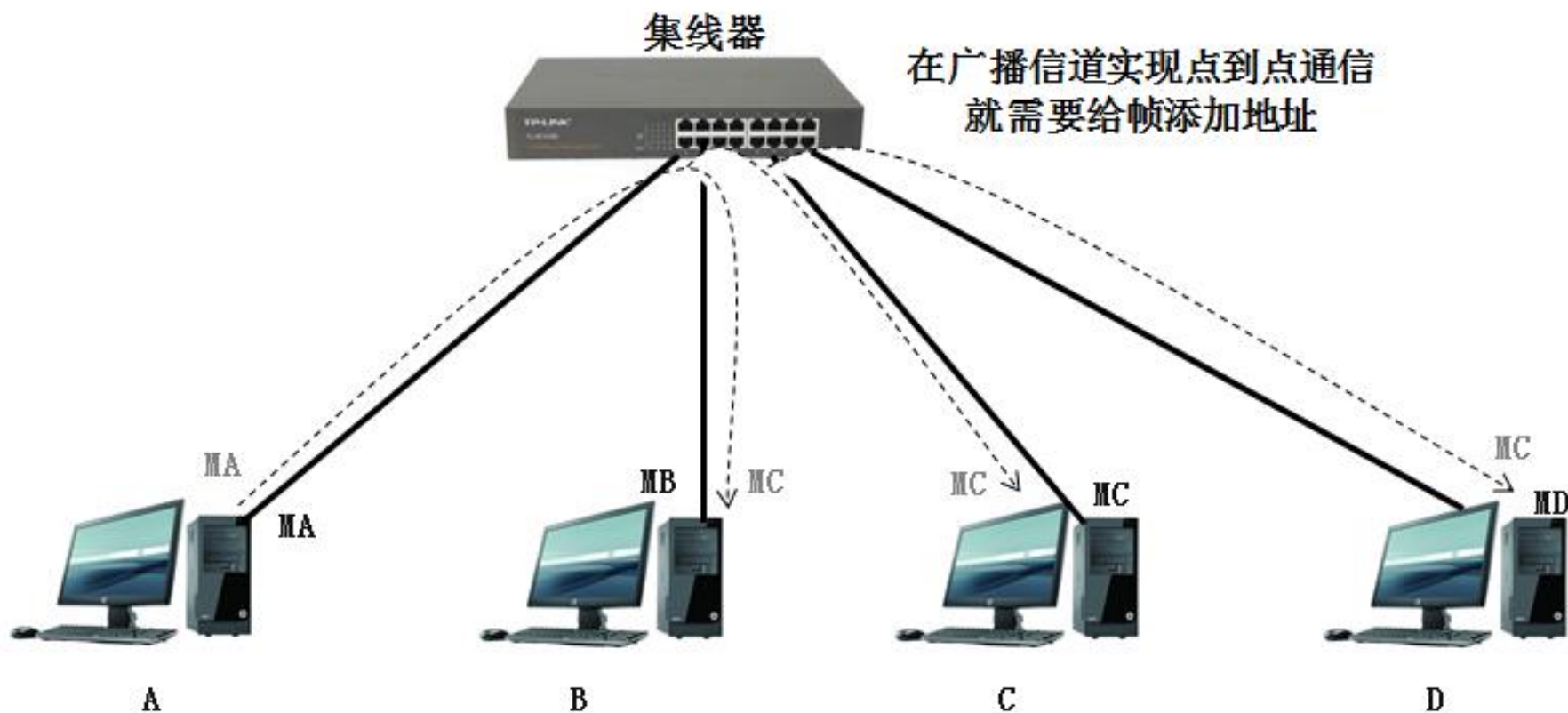


在广播信道实现点到点通信需要给帧添加地址，并且要冲突检测

4.3.1 广播信道的局域网

- 使用集线器组建的局域网也是广播信道，是总线型拓扑。

广播信道局域网—星型



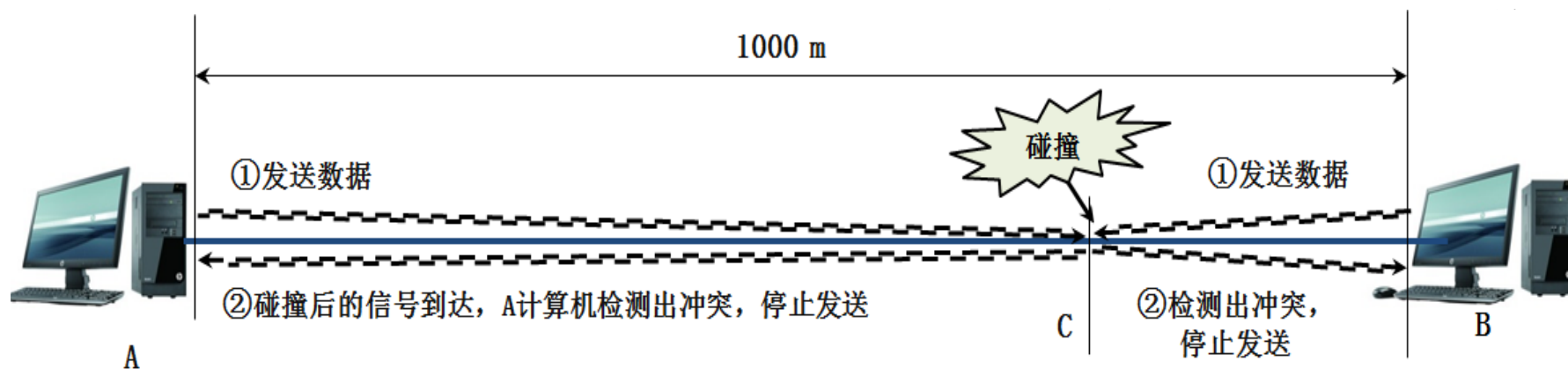
4.3.2 以太网标准

- 以太网（Ethernet）是一种计算机局域网组网技术。IEEE制定的IEEE 802.3标准给出了以太网的技术标准，即以太网的介质访问控制协议（CSMA/CD）及物理层技术规范（包括物理层的连线、电信号和介质访问层协议的内容）。
- 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↵	早期电缆，已经废弃↵
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↵

4.3.3 CSMA/CD协议

- 总线型网络使用CSMA/CD协议进行通信，即带冲突检测的载波侦听多点接入技术。
- 即便检测出总线上没有信号，开始发送数据后也有可能和迎面而来的信号在链路上发生碰撞。
- 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，就认为发生冲突，这就是**冲突检测**。



4.3.4 以太网最短帧

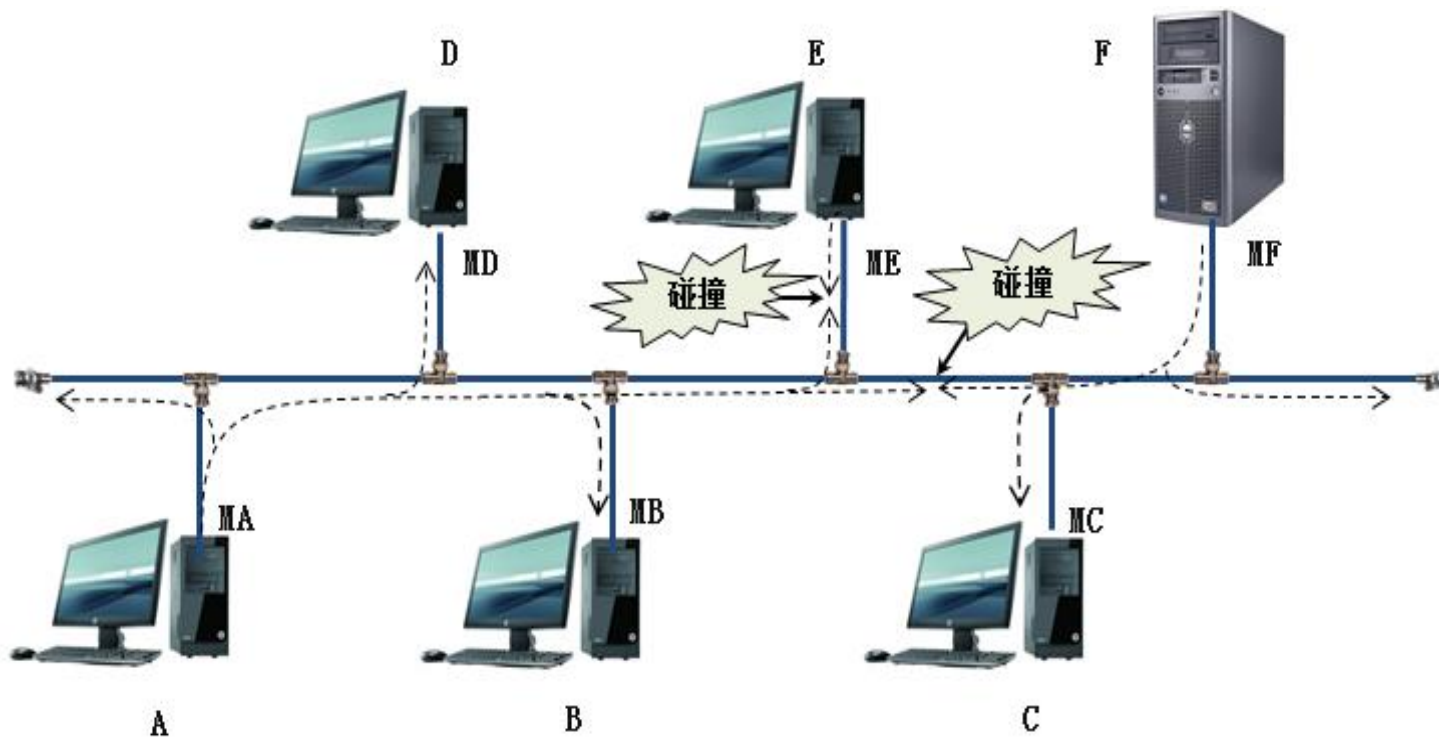
- 以太网设计最大端到端长度为5km（实际上的以太网覆盖范围远远没有这么大），单程传播时延为大约为25.6μs,往返传播时延为51.2μs，10M标准以太网最小帧为：

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

- 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。

4.3.5 冲突解决方法--退避算法

- 总线型网络中的计算机数量越多，在链路上发送数据产生冲突机会就多。



4.3.5 冲突解决方法--退避算法

- 计算机要想知道发送的帧在链路上是否发生碰撞必须等待 2τ ， 2τ 称为争用期。
- 以太网使用截断二进制指数退避（truncated binary exponential backoff）算法来解决碰撞问题。
 - 1) 确定基本退避时间，它就是争用期 2τ 。以太网把争用期定为 $51.2\mu\text{s}$ 。对于 10Mb/s 以太网，在争用期内可发送 512bit ，即 64 字节。也可以说争用期是 512 比特时间。 1 比特时间就是发送 1 比特所需的时间。所以这种时间单位与数据率密切相关。
 - 2) 从离散的整数集合 $[0, 1, \dots, (2^k-1)]$ 中随机取出一个数，记为 r 。重传应推后的时间就是 r 倍的争用期。上面的参数 k 按下面的公式计算：

$$k = \text{Min}[\text{重传次数}, 10]$$

可见当重传次数不超过 10 时，参数 k 等于重传次数；但当重传次数超过 10 时， k 就不再增大而一直等于 10 。

- 3) 当重传达 16 次仍不能成功时（这表明同时打算发送数据的站太多，以致连续发生冲突），则丢弃该帧，并向高层报告。

4.3.6 以太网帧格式

- 常用的以太网MAC帧格式有两种标准，一种是EthernetV2标准（即以太网V2标准），另一种是IEEE的802.3标准。使用得最多的是以太网V2的MAC帧格式。

The screenshot shows the Wireshark interface with a packet list and packet details pane. The packet list shows a sequence of network events, including a loop, ARP requests, and several ICMP Echo (ping) requests and replies. The packet details pane is expanded for the selected packet (Frame 4), showing the Ethernet II header, Internet Protocol Version 4 header, and Internet Control Message Protocol header. The Ethernet II header details are highlighted with arrows pointing to the target MAC address, source MAC address, and protocol type.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	cc:01:27:44:00:00	cc:01:27:44:00:00	LOOP	60	Reply
2	2.42713900	Vmware_c7:13:1f	Broadcast	ARP	42	who has 192.168.10.10? Tell 192.168.10.
3	2.43113900	cc:01:27:44:00:00	Vmware_c7:13:1f	ARP	60	192.168.10.10 is at cc:01:27:44:00:00
4	2.43113900	192.168.10.20	192.168.10.10	ICMP	74	Echo (ping) request id=0x0200, seq=1536
5	2.44113900	192.168.10.10	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0200, seq=1536
6	3.42719600	192.168.10.20	192.168.10.10	ICMP	74	Echo (ping) request id=0x0200, seq=1792
7	3.43519600	192.168.10.10	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0200, seq=1792
8	4.42725300	192.168.10.20	192.168.10.10	ICMP	74	Echo (ping) request id=0x0200, seq=2048
9	4.43725400	192.168.10.10	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0200, seq=2048

Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

Ethernet II, Src: Vmware_c7:13:1f (00:0c:29:c7:13:1f), Dst: cc:01:27:44:00:00 (cc:01:27:44:00:00)

- Destination: cc:01:27:44:00:00 (cc:01:27:44:00:00) ← 目标MAC地址
- Source: Vmware_c7:13:1f (00:0c:29:c7:13:1f) ← 源MAC地址
- Type: IP (0x0800) ← 协议类型

Internet Protocol Version 4, Src: 192.168.10.20 (192.168.10.20), Dst: 192.168.10.10 (192.168.10.10)

Internet Control Message Protocol

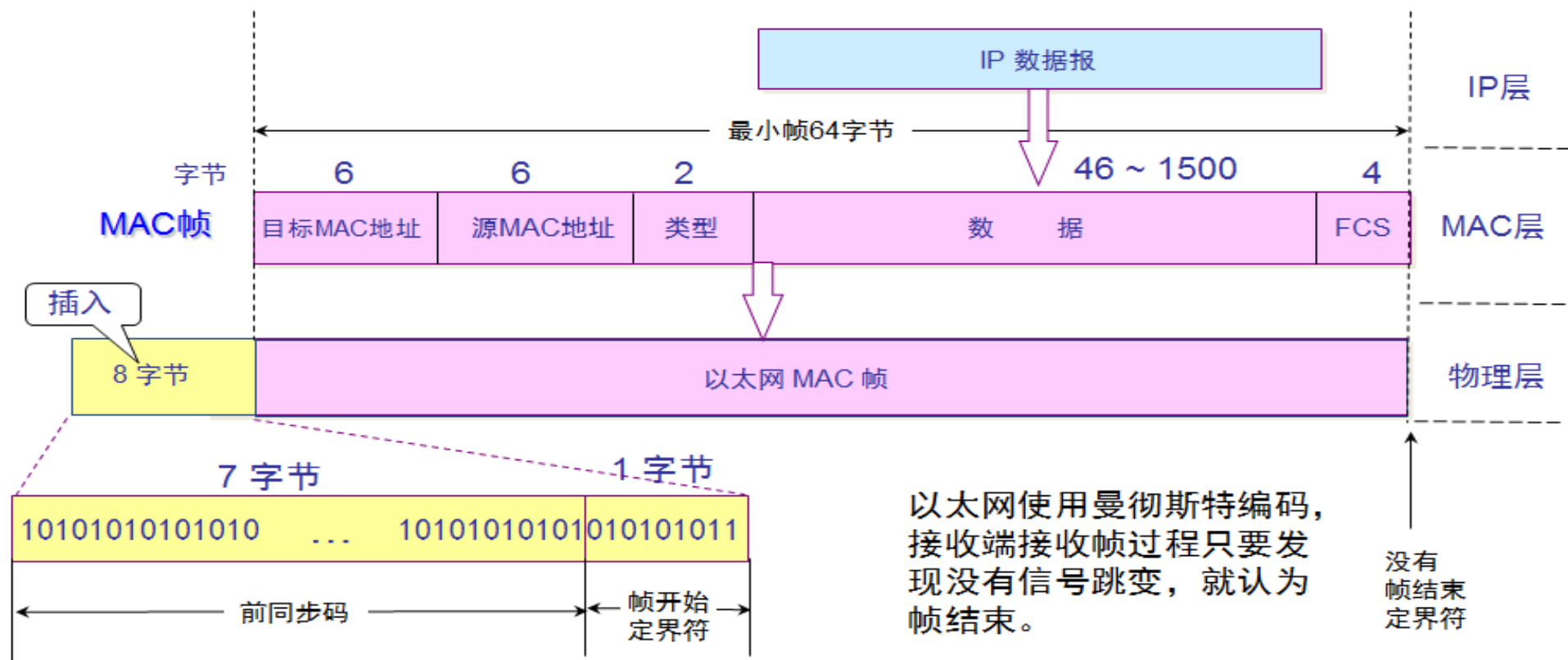
0000 cc 01 27 44 00 00 00 0c 29 c7 13 1f 08 00 45 00 .. 'D....).....E.
0010 00 3c 00 82 00 00 80 01 a4 d0 c0 a8 0a 14 c0 a8 ..<.....
0020 0a 0a 08 00 45 5c 02 00 06 00 61 62 63 64 65 66E\.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

Ethernet (eth), 14 bytes Packets: 12 · Displayed: 12 (100.0%) · Dropped: 0 (0.0%) Profile: Default

4.3.6 以太网帧格式

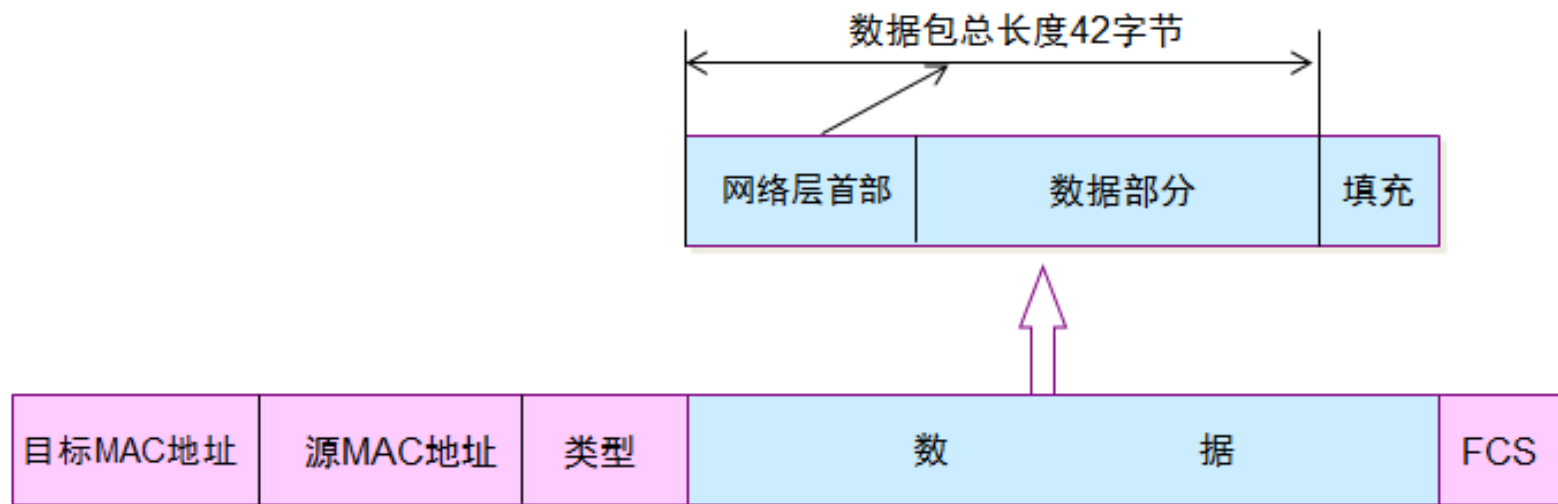
- Ethernet II的帧比较简单，由五个字段组成。

以太网帧格式



4.3.6 以太网帧格式

- 当数据字段的长度小于46字节时，数据链路层就会在数据字段的后面加入一个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，接收端还必须能够将添加的字节去掉。



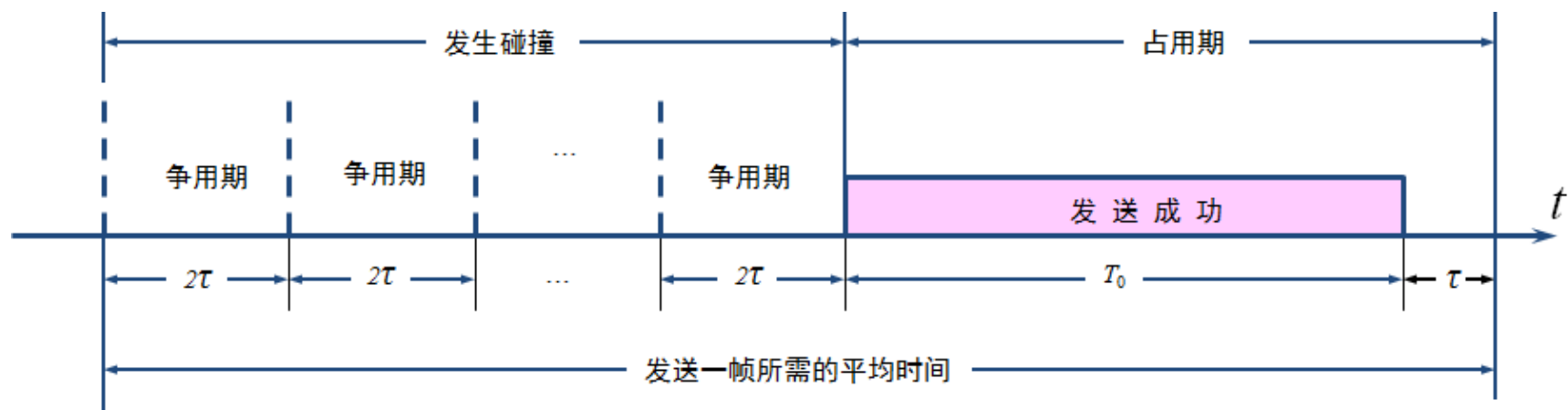
4.3.6 以太网帧格式

- IEEE802.3标准规定凡出现下列情况之一的即为无效的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。
 - 收到的帧的MAC客户数据字段的长度不在46-1500字节之间。考虑到MAC帧首部和尾部的长度共有18字节，可以得出有效的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。

4.3.7 以太网信道利用率

- 利用率是指的发送数据的时间占整个时间的比例。

如图所示，平均发送一帧所需要的时间，经历了 n 倍争用期 2τ ， T_0 为发送该帧所需时间， τ 为该帧传播时延。



有冲突时信道利用率为:

$$S = \frac{T_0}{n2\tau + T_0 + \tau}$$

4.3.7 以太网信道利用率

- 从公式可以看出，要想提高信道利用率最好是n为0，这就意味着以太网上的各个计算机发送数据不会产生碰撞（这显然已经不是CSMA/CD，而需要一种特殊的调度方法），并且能够非常有效的利用网络的传输资源，即总线一旦空闲就有一个站立即发送数据。这种情况算出来的信道利用率是极限信道利用率。

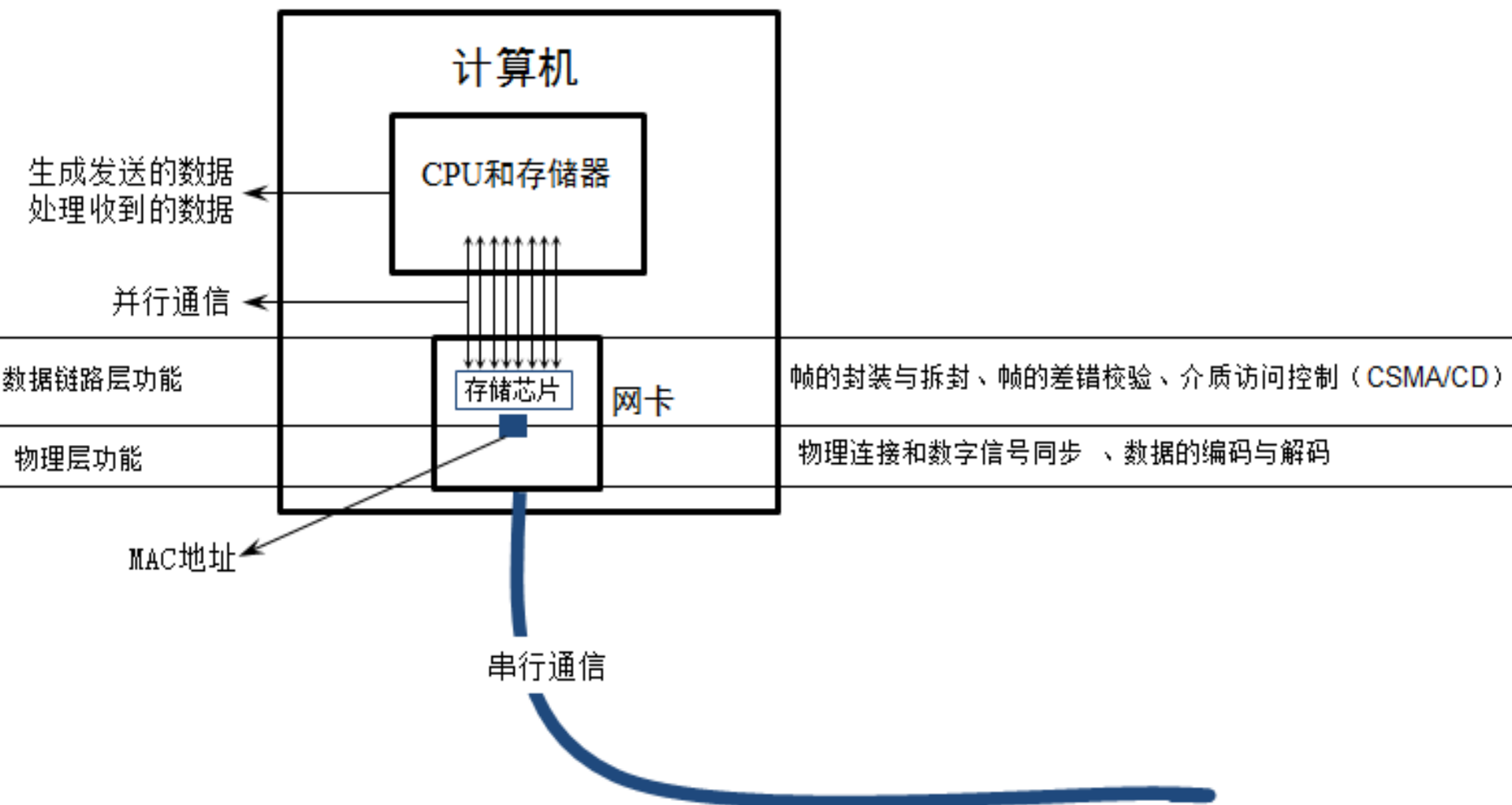
$$S_{\max} = \frac{T_0}{T_0 + \tau} = \frac{1}{1 + \frac{\tau}{T_0}}$$

- 要想提高极限信道利用率就要降低公式中

$$\frac{\tau}{T_0}$$

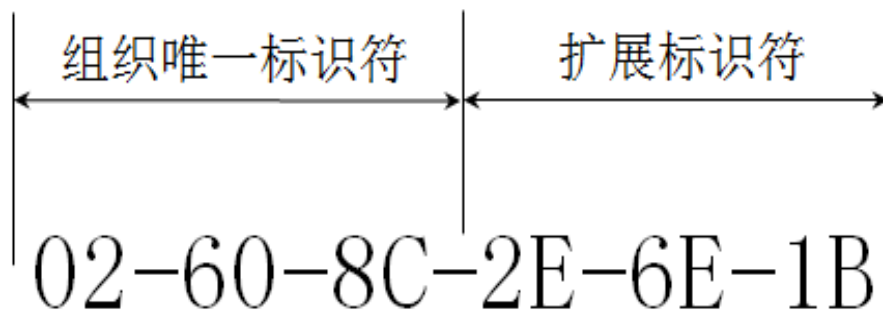
- τ 值和以太网连线的长度有关，这就意味着 τ 值要小，以太网网线的长度就不能太长。带宽一定的情况下 T_0 和帧的长度有关，这就意味着，以太网的帧不能太短。

4.3.8 网卡的作用



4.3.9 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。



4.3.9 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。

4.3.10 实战：查看和更改MAC地址

```
C:\Windows\system32\cmd.exe

C:\Users\han>ipconfig /all

Windows IP 配置

   主机名 . . . . . : xueitPC
   主 DNS 后缀 . . . . . :
   节点类型 . . . . . : 混合
   IP 路由已启用 . . . . . : 否
   WINS 代理已启用 . . . . . : 否


以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . :
   描述. . . . . : Realtek PCIe GBE Family Controller
   物理地址. . . . . : C8-60-00-2E-6E-1B  MAC地址
   DHCP 已启用 . . . . . : 是
   自动配置已启用. . . . . : 是
   本地链接 IPv6 地址. . . . . : fe80::f5fb:9013:6ee2:bbd5%20(首选)
   IPv4 地址 . . . . . : 192.168.0.102(首选)
   子网掩码 . . . . . : 255.255.255.0
   获得租约的时间 . . . . . : 2016年5月26日 22:56:40
   租约过期的时间 . . . . . : 2016年5月27日 22:56:39
   默认网关. . . . . : 192.168.0.1
   DHCP 服务器 . . . . . : 192.168.0.1
   DNS 服务器 . . . . . : 10.7.1.6
                           114.114.114.114
```

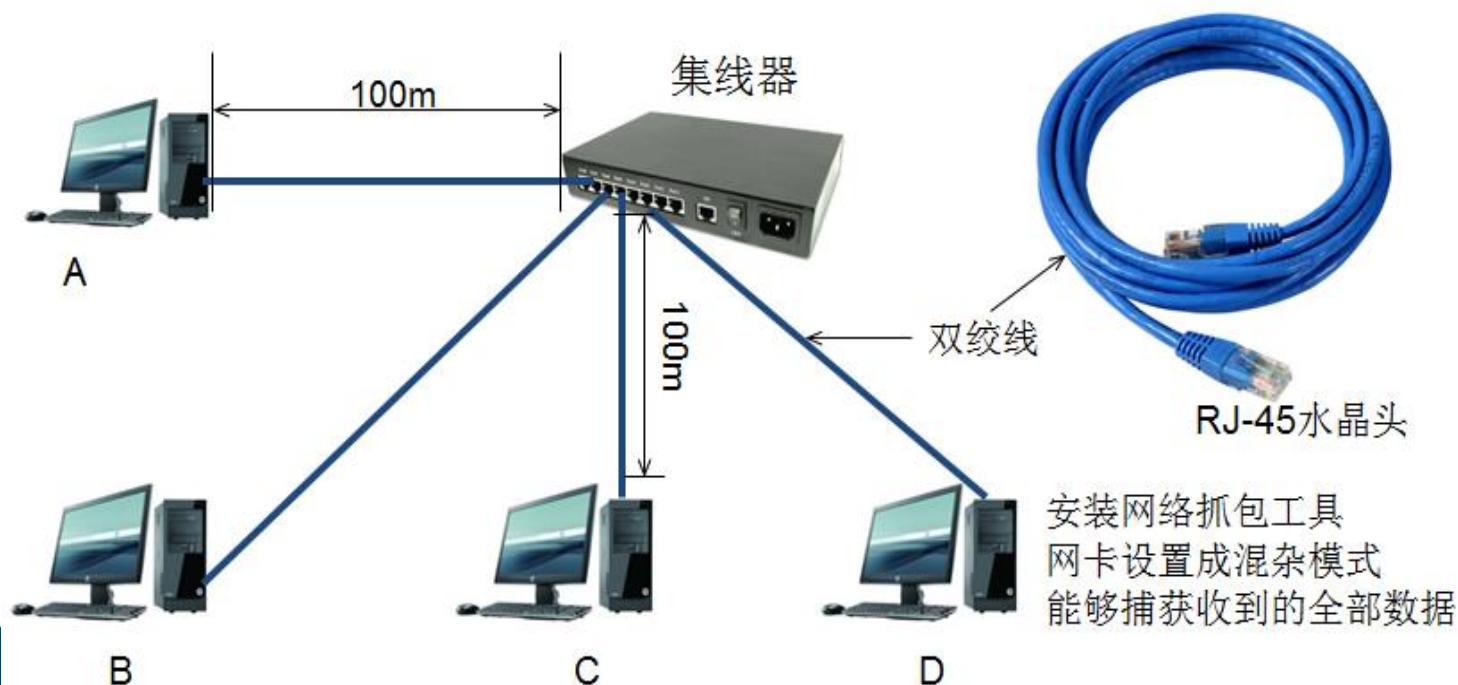


4.4扩展以太网

- 4.4.1集线器
- 4.4.2计算机数量和距离上扩展
- 4.4.3使用网桥优化以太网
- 4.4.4网桥自动构建MAC地址表
- 4.4.5多接口网桥--交换机
- 4.4.6实战：查看交换机MAC地址表
- 4.4.7实战：验证交换机端口安全
- 4.4.8实战：验证集线器不安全
- 4.4.8生成树协议

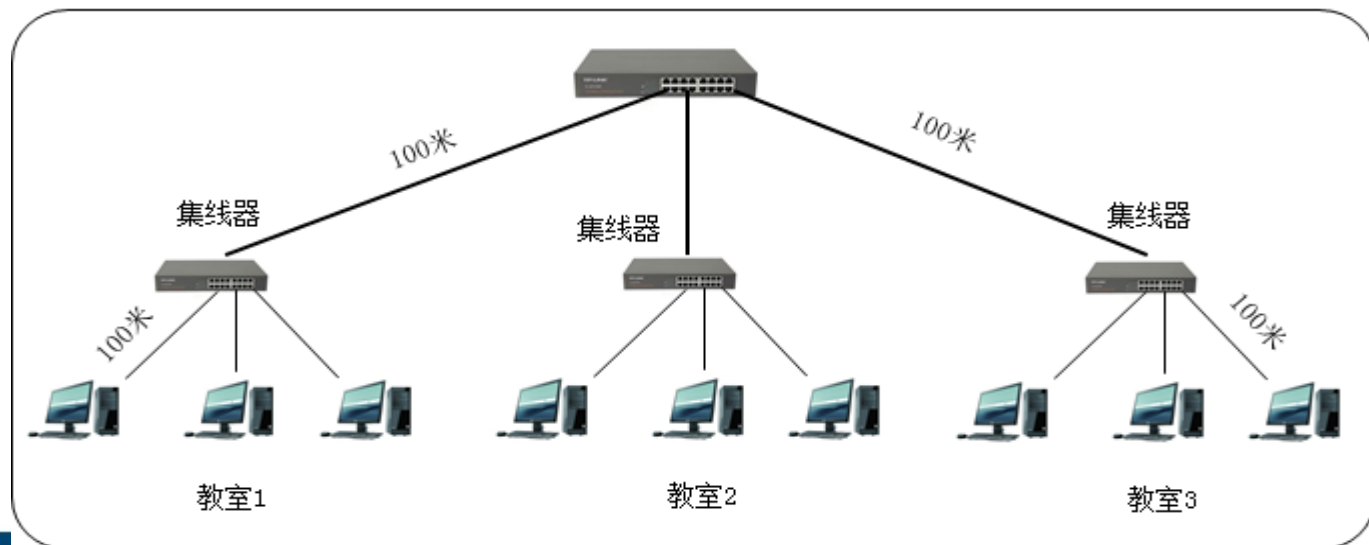
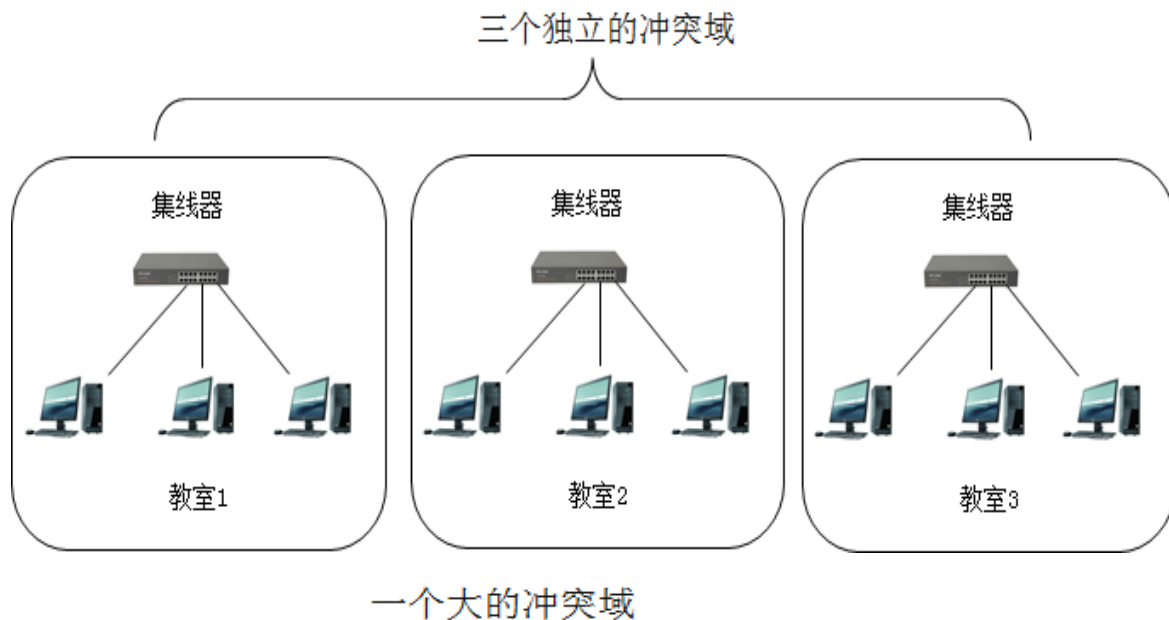
4.4.1集线器

- 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。
- 1990年IEEE制定出星形以太网10BASE-T的标准802.3i。“10”代表10Mb/s的数据率，BASE表示连接线上的信号是基带信号，T代表双绞线。
- 10BASE-T以太网的通信距离稍短，每个站到集线器的距离不超过100m。
- 集线器和网线一样工作在物理层。



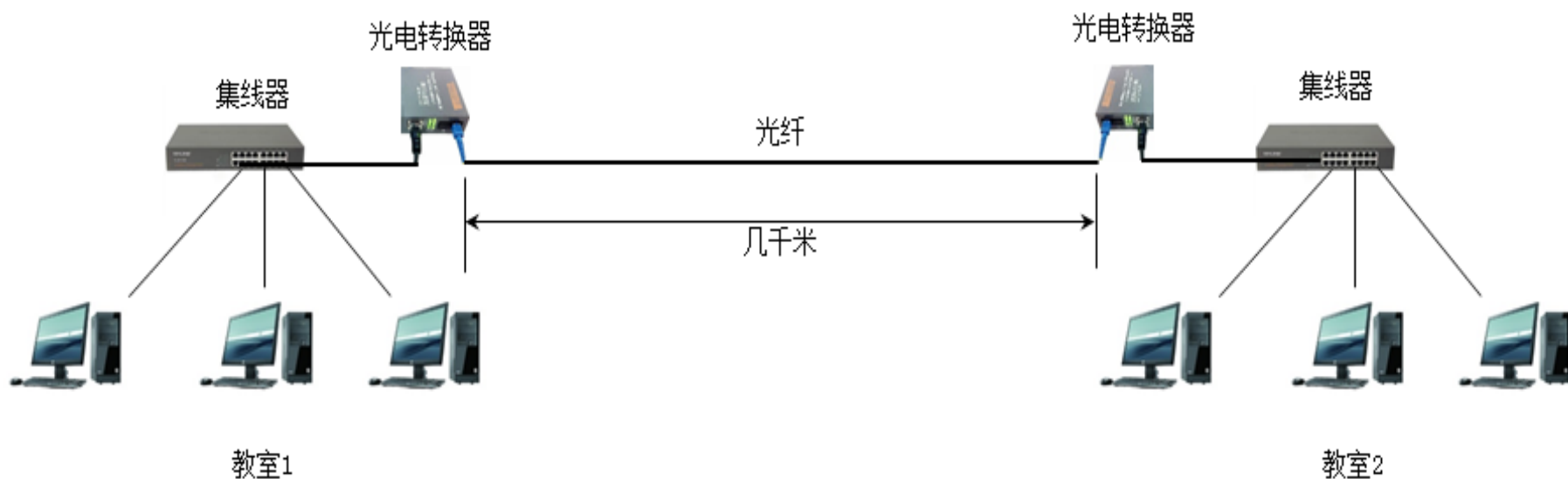
4.4.2 计算机数量和距离上扩展

- 独立的冲突域
- 可以将多个集线器连接在一起形成一个更大的以太网，这不仅可以扩以太网中计算机的数量，还可以扩展以太网的覆盖范围。使用主干集线器连接教室中集线器，形成一个大的以太网，计算机之间的最大距离可以达到400米。



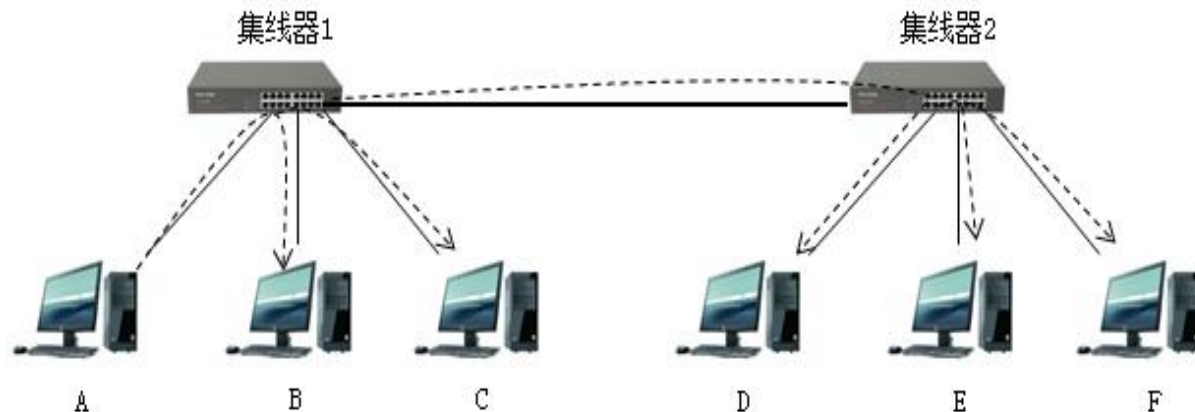
4.4.2 计算机数量和距离上扩展

- 要是两个集线器的距离超过100米，还可以光纤从将两个集线器连接起来，集线器之间通过光纤连接，可以将相距几千米的集线器连接起来，需要通过光电转换器，实现光信号和电信号的相互转换。



4.4.3使用网桥优化以太网

■ 大的冲突域



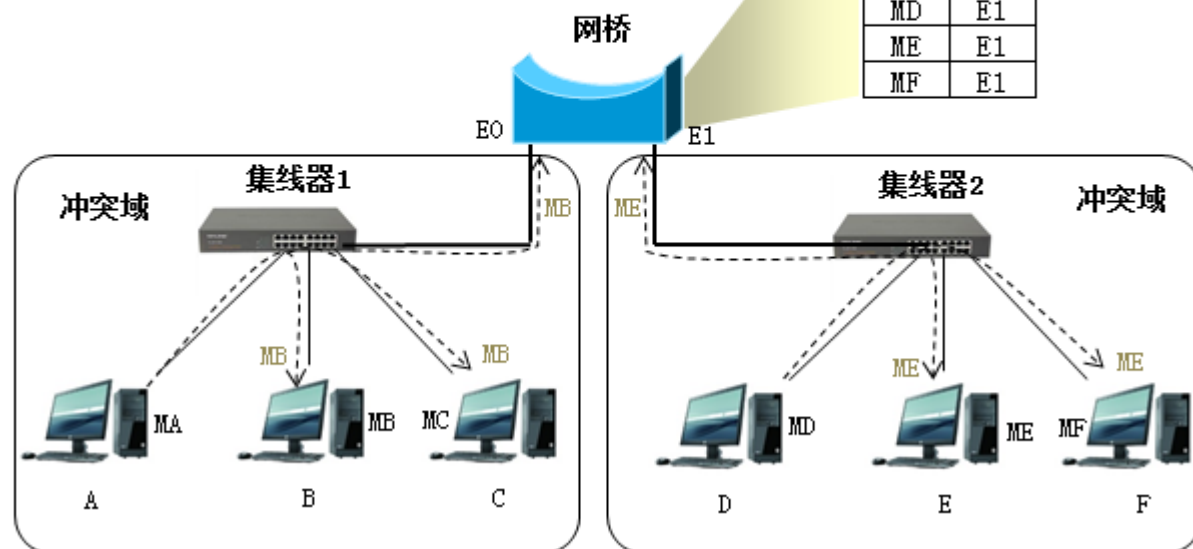
■ 网桥基于MAC地址转发

1. 网桥基于MAC地址转发帧，工作在数据链路层。
2. 一个接口一个冲突域。冲突域数量增加，冲突减少。
3. 实现帧的存储转发，增加了时延。
4. E1接口和E2接口可以是不同的带宽。

MAC地址表

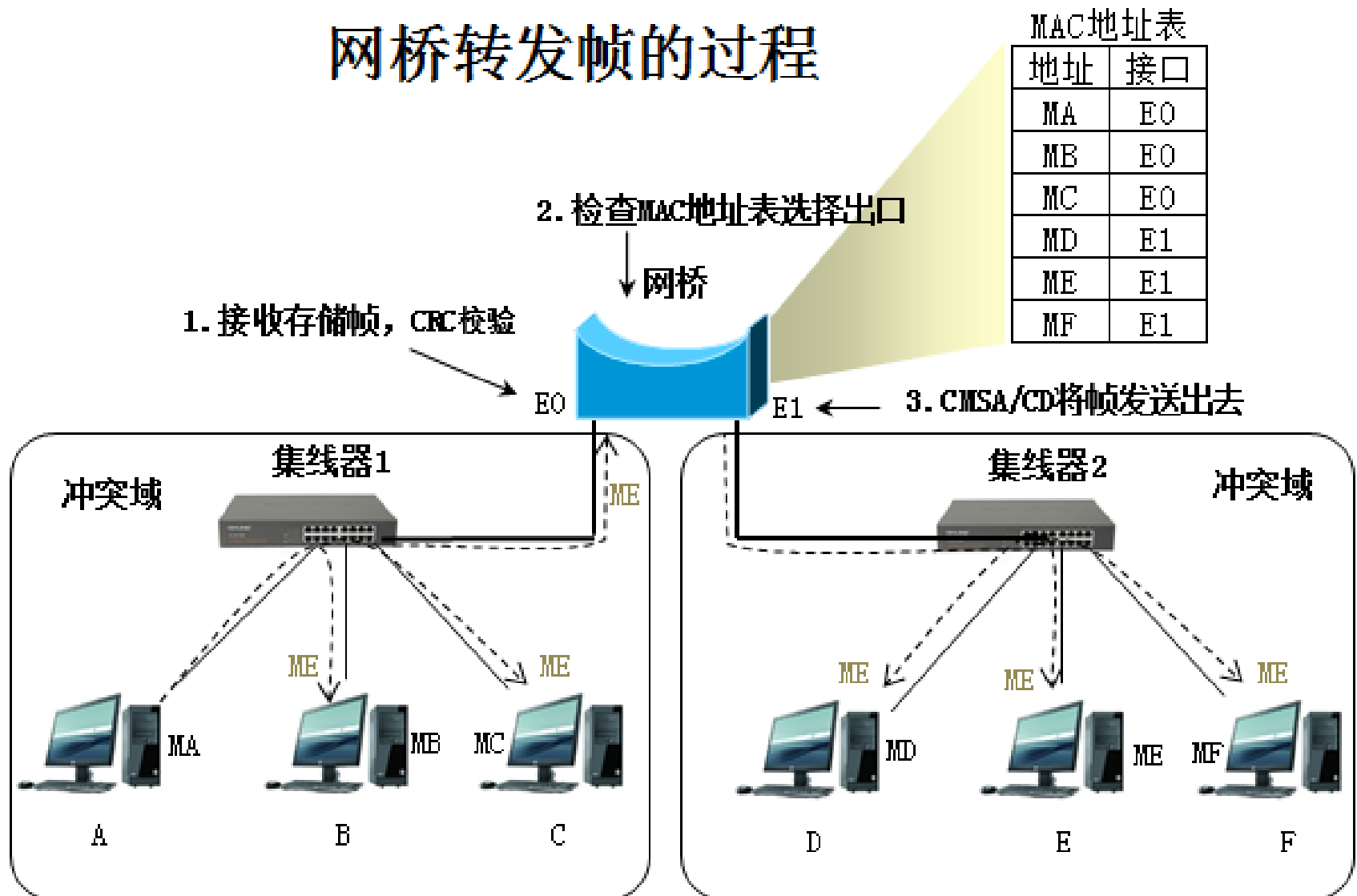
地址	接口
MA	E0
MB	E0
MC	E0
MD	E1
ME	E1
MF	E1

■ 隔绝冲突



4.4.3使用网桥优化以太网

网桥转发帧的过程

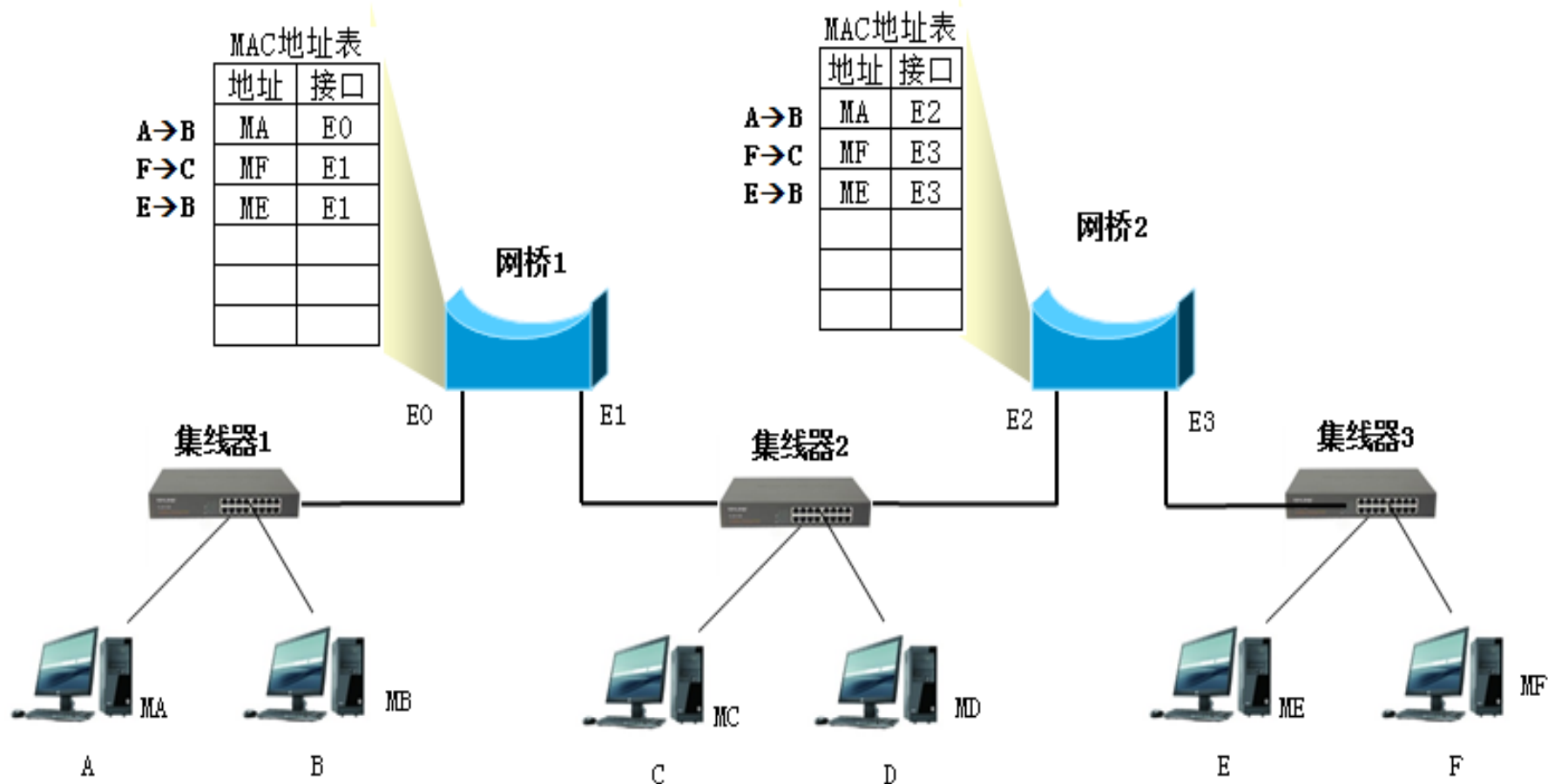


4.4.4网桥自动构建MAC地址表

- 使用网桥优化以太网，对于网络中的计算机是没有感觉的，也就是以太网中的计算机是不知道网络中有网桥存在，也不需要网络管理员配置网桥的MAC地址表，因此我们称网桥是**透明桥接**。
- 网桥接入以太网时，MAC地址表示空的，网桥会在计算机通信过程中自动构建MAC地址表，这称为“自学习”。
- (1) 自学习
 - 网桥的接口收到一个帧，就要检查MAC地址表中与收到的帧源MAC地址有无匹配的项目，如果没有，就在MAC地址表中添加该接口和该帧的源MAC地址对应关系以及进入接口的时间，如果有，则把原有的项目进行更新。
- (2) 转发帧
 - 网桥接口收到一个帧，就检查MAC地址表中有没有该帧目标MAC地址对应端口，如果有，就会将该帧转发到对应的端口，如果没有，则将该帧转发到全部端口（接收端口除外）。

4.4.4网桥自动构建MAC地址表

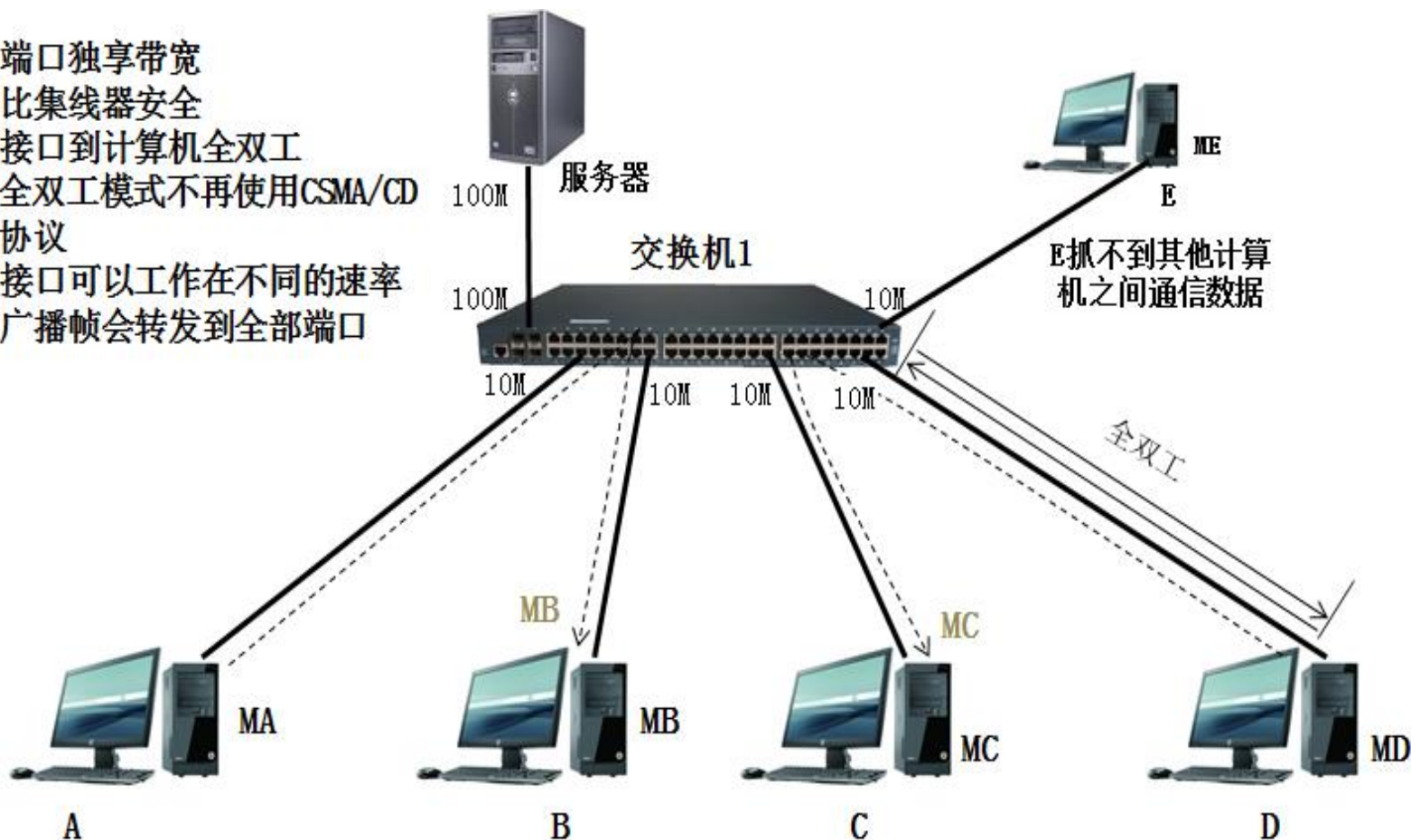
网桥MAC地址表构建过程



4.4.5多接口网桥--交换机

- 随着技术的发展，网桥接口增多，网桥的接口就直接连接计算机了，网桥就发展成现在的交换机。

1. 端口独享带宽
2. 比集线器安全
3. 接口到计算机全双工
4. 全双工模式不再使用CSMA/CD协议
5. 接口可以工作在不同的速率
6. 广播帧会转发到全部端口

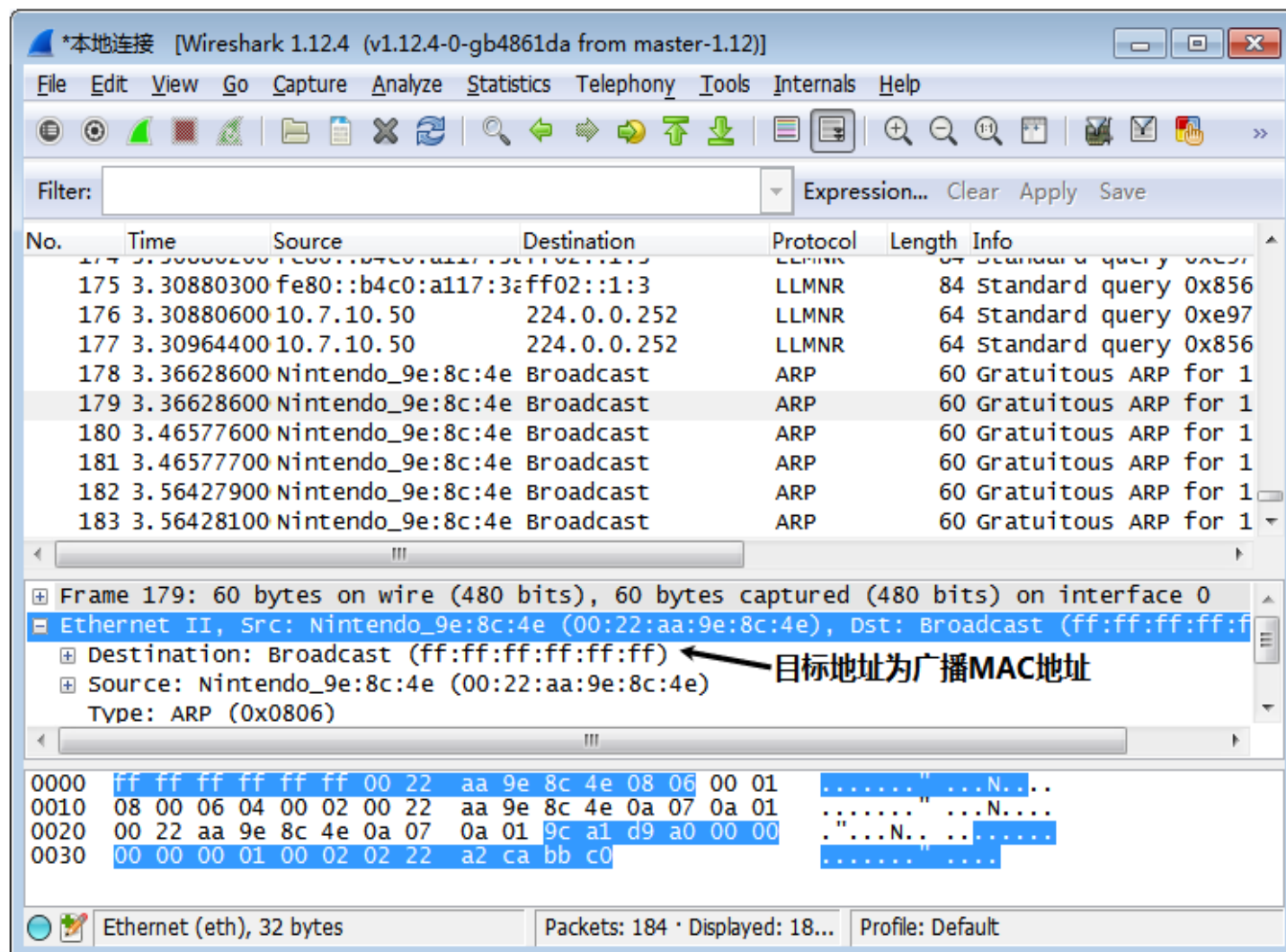


4.4.5多接口网桥--交换机

- 使用交换机组网与集线器组网相比有以下特点：
- 独享带宽
 - 交换机的每个端口独享带宽，10M交换机，则每个端口带宽是10M，24口10M交换机，交换机的总体交换能力是240M，这和集线器不同。
- 安全
 - 使用交换机组建的网络比集线器安全，比如计算机A给计算机B发送的帧，以及计算机D给计算机C发送的帧，交换机根据MAC地址表只转发到目标端口，E计算机根本收不到其他计算机的通信的数字信号，即便安装了抓包工具也没用。
- 全双工通信
 - 交换机接口和计算机直接相连，计算机和交换机之间的链路可以使用全双工通信。
- 全双工不再使用CSMA/CD协议
 - 交换机接口和计算机直接相连接，使用全双工通信数据链路层就不需要使用CSMA/CD协议，但我们还是称交换机组建的网络是以太网，是因为帧格式和以太网一样。
- 接口可以工作在不同的速率

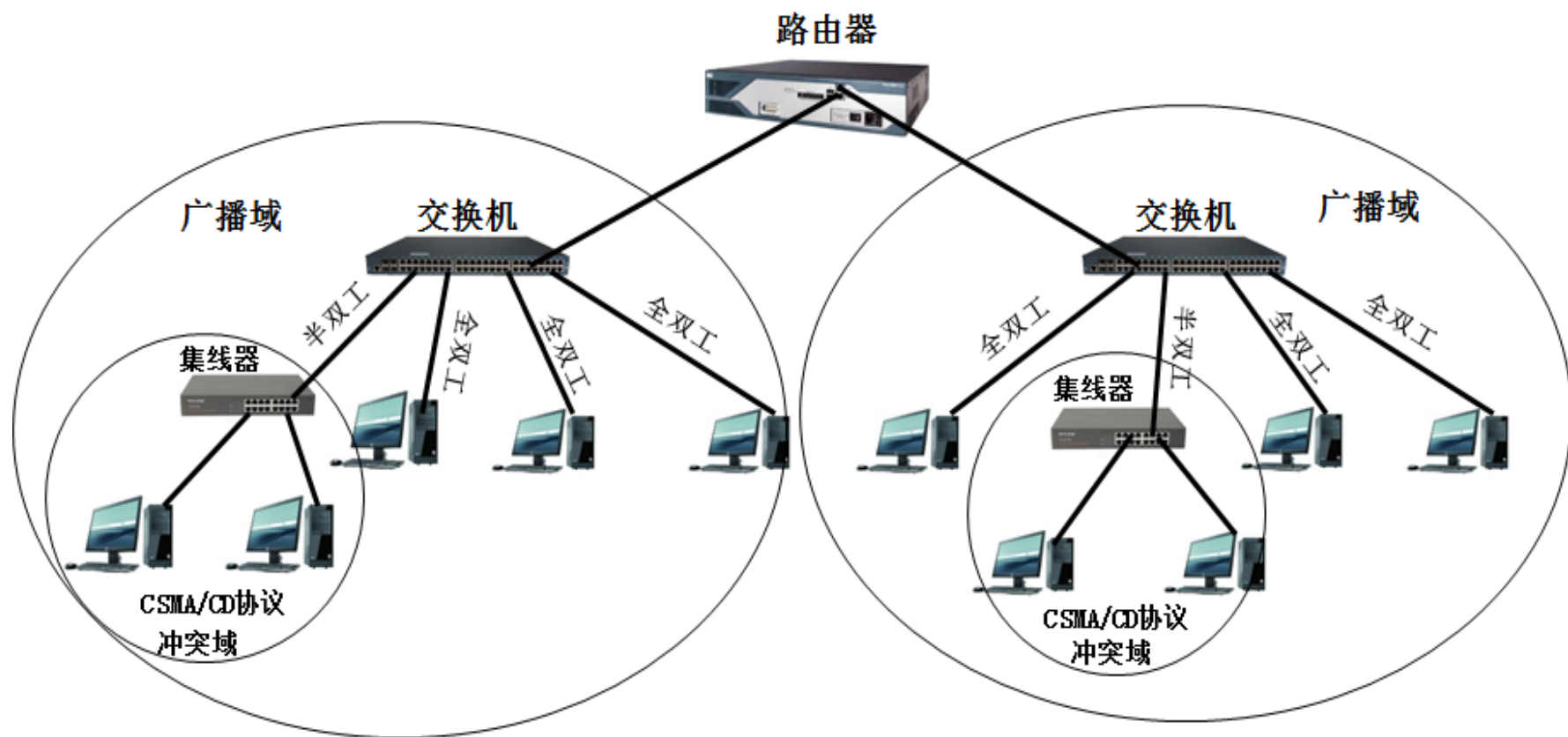
4.4.5多接口网桥--交换机

■ 转发广播帧到所有端口



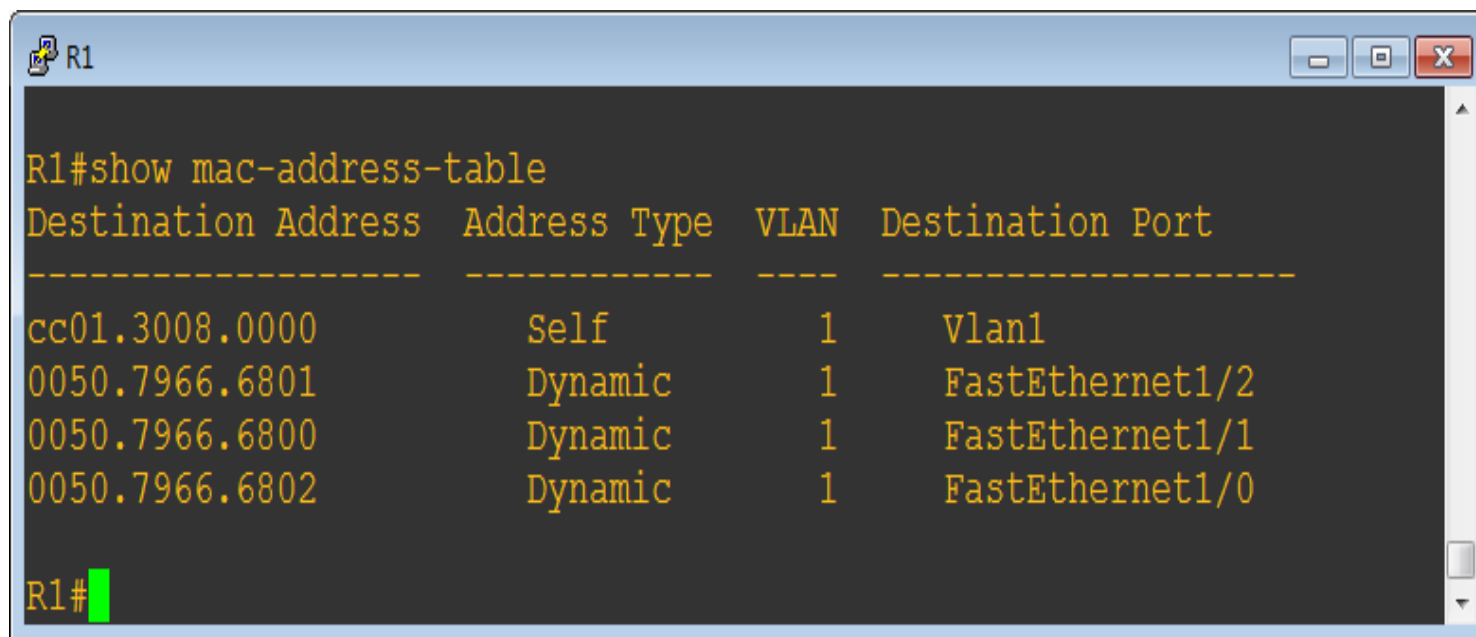
4.4.5多接口网桥--交换机

- 集线器是冲突域
- 交换机是广播域
- 路由器隔绝广播



4.4.6实战：查看交换机MAC地址表

- 交换机能够学习根据帧的源MAC地址构造MAC地址表



The screenshot shows a terminal window titled 'R1' with a dark background and yellow text. The command 'R1#show mac-address-table' has been entered, and the output displays the MAC address table. The table has four columns: Destination Address, Address Type, VLAN, and Destination Port. The output shows four entries: a self-address (cc01.3008.0000) on Vlan1, and three dynamic addresses (0050.7966.6801, 0050.7966.6800, 0050.7966.6802) on FastEthernet1/2, FastEthernet1/1, and FastEthernet1/0 respectively. The prompt 'R1#' is visible at the bottom left.

```
R1#show mac-address-table
Destination Address  Address Type  VLAN  Destination Port
-----
cc01.3008.0000      Self          1      Vlan1
0050.7966.6801      Dynamic       1      FastEthernet1/2
0050.7966.6800      Dynamic       1      FastEthernet1/1
0050.7966.6802      Dynamic       1      FastEthernet1/0
R1#
```


4.4.7实战：验证交换机端口安全

Packet capture

Please select a port:

R1 port FastEthernet1/2 (Ethernet encapsulation: DLT_EN10MB)

OK Cancel

Start capture

Delete

Standard input [R1 FastEthernet1/2 to PC3 Ethernet0] [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
2	1.98701900	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
3	4.04903800	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
4	6.01505100	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
5	8.02006300	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
6	10.06507800	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
7	12.03308900	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
8	14.00310300	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
9	14.29810400	Private_66:68:00	Broadcast	ARP	64	who has 192.168.1.10? Tell 192.168.1.20
10	16.07311800	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
11	18.03013200	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
12	20.01114600	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
13	22.02015900	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
14	24.01617100	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
15	26.03918500	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
16	28.03219900	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
17	28.77220300	Private_66:68:00	Broadcast	ARP	64	who has 192.168.1.30? Tell 192.168.1.20
18	28.77420300	Private_66:68:01	Private_66:68:00	ARP	64	192.168.1.30 is at 00:50:79:66:68:01 [ETH
19	28.77520300	192.168.1.20	192.168.1.30	ICMP	98	Echo (ping) request id=0xc331, seq=1/256
20	28.77520300	192.168.1.30	192.168.1.20	ICMP	98	Echo (ping) reply id=0xc331, seq=1/256
21	29.78121000	192.168.1.20	192.168.1.30	ICMP	98	Echo (ping) request id=0xc431, seq=2/512
22	29.78121000	192.168.1.30	192.168.1.20	ICMP	98	Echo (ping) reply id=0xc431, seq=2/512
23	30.03821100	cc:01:17:5c:f1:02	Spanning-tree-(for STP	60	Conf.	Root = 32768/0/cc:01:30:08:00:00
24	30.79921700	192.168.1.20	192.168.1.30	ICMP	98	Echo (ping) request id=0xc531, seq=3/768
25	30.79921700	192.168.1.30	192.168.1.20	ICMP	98	Echo (ping) reply id=0xc531, seq=3/768
26	31.81022400	192.168.1.20	192.168.1.30	ICMP	98	Echo (ping) request id=0xc631, seq=4/102

没有捕获 PC2 和 PC1 通信的帧

捕获 PC2 和 PC3 通信的帧

广播帧

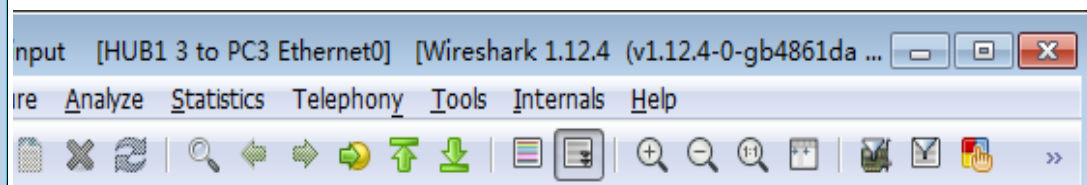
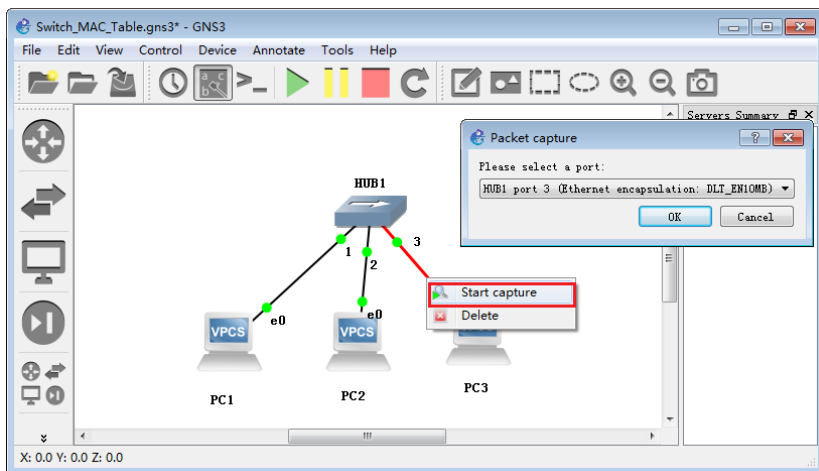
广播帧

源IP地址

目标IP地址

4.4.8实战：验证集线器不安全

- 能捕获集线器网络中的全部数据包



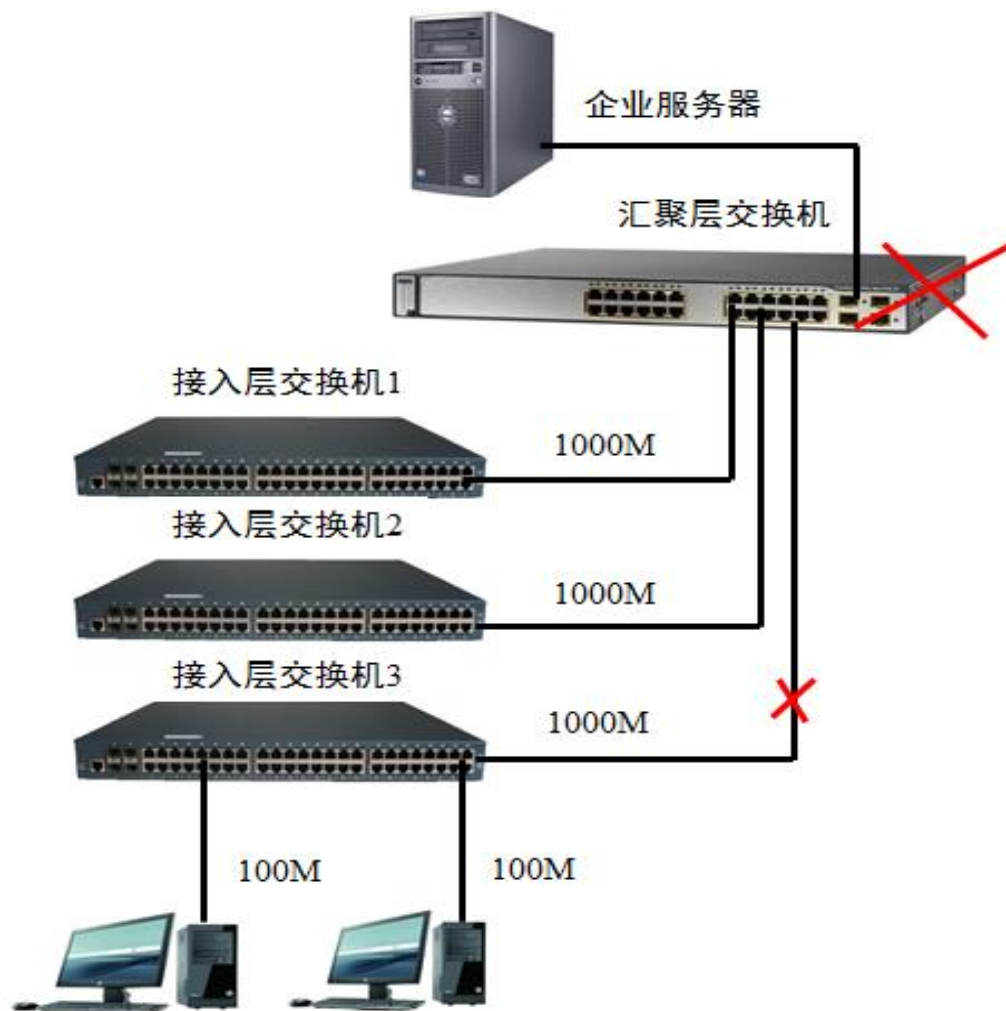
Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	Private_66:68:00	Broadcast	ARP	64	who has 192.168.1.10?
2	0.00100000	Private_66:68:02	Private_66:68:00	ARP	64	192.168.1.10 is at 00:50
3	0.00200000	192.168.1.20	192.168.1.10	ICMP	98	Echo (ping) request id=
4	0.00200000	192.168.1.10	192.168.1.20	ICMP	98	Echo (ping) reply id=
5	1.00305800	192.168.1.20	192.168.1.10	ICMP	98	Echo (ping) request id=
6	1.00305800	192.168.1.10	192.168.1.20	ICMP	98	Echo (ping) reply id=
7	2.00411500	192.168.1.20	192.168.1.10	ICMP	98	Echo (ping) request id=
8	2.00411500	192.168.1.10	192.168.1.20	ICMP	98	Echo (ping) reply id=
9	3.00517200	192.168.1.20	192.168.1.10	ICMP	98	Echo (ping) request id=
10	3.00517200	192.168.1.10	192.168.1.20	ICMP	98	Echo (ping) reply id=
11	4.00622900	192.168.1.20	192.168.1.10	ICMP	98	Echo (ping) request id=
12	4.00622900	192.168.1.10	192.168.1.20	ICMP	98	Echo (ping) reply id=

Standard input: <live capture in progress>... Packets: 12 · Displayed: 12 ... Profile: Default

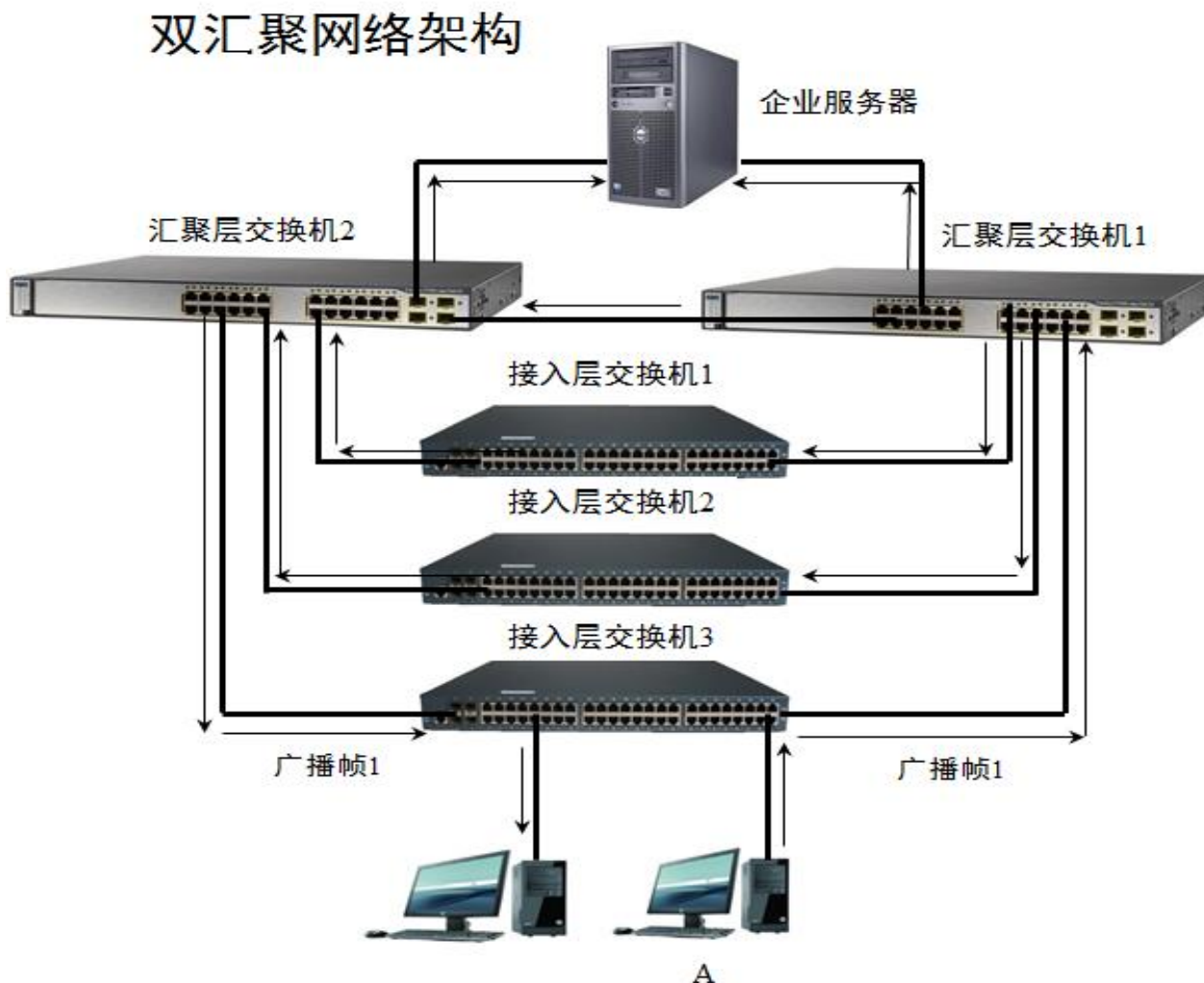
4.4.9生成树协议

■ 存在单点故障



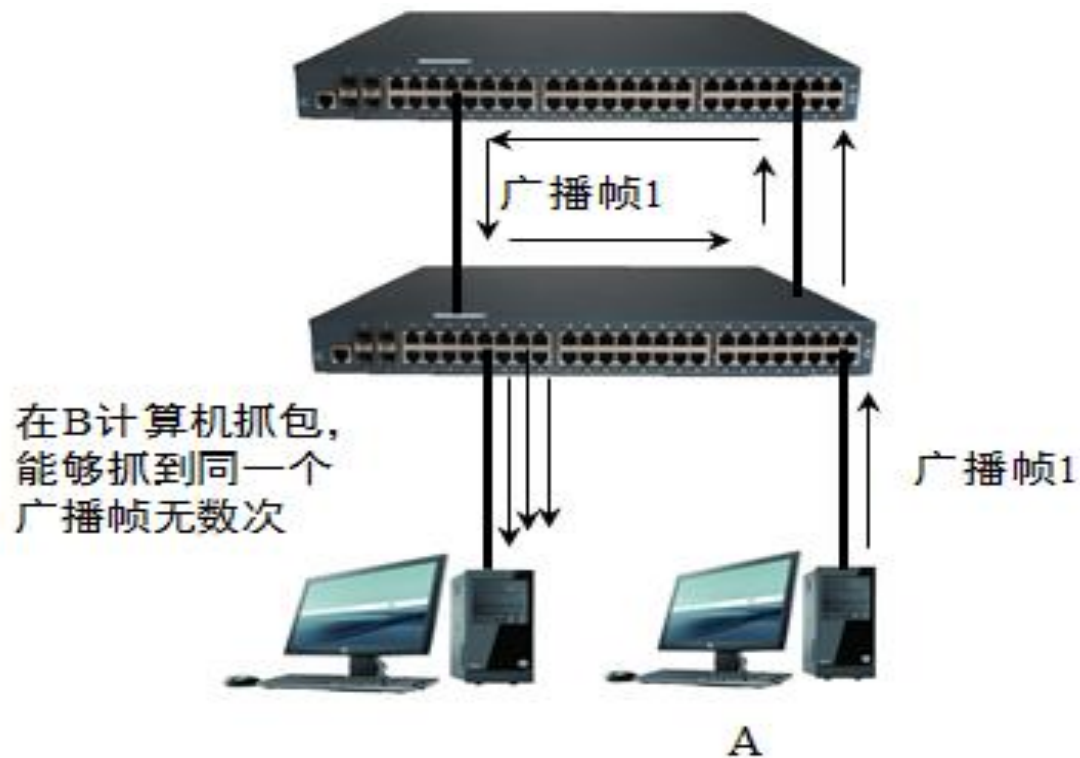
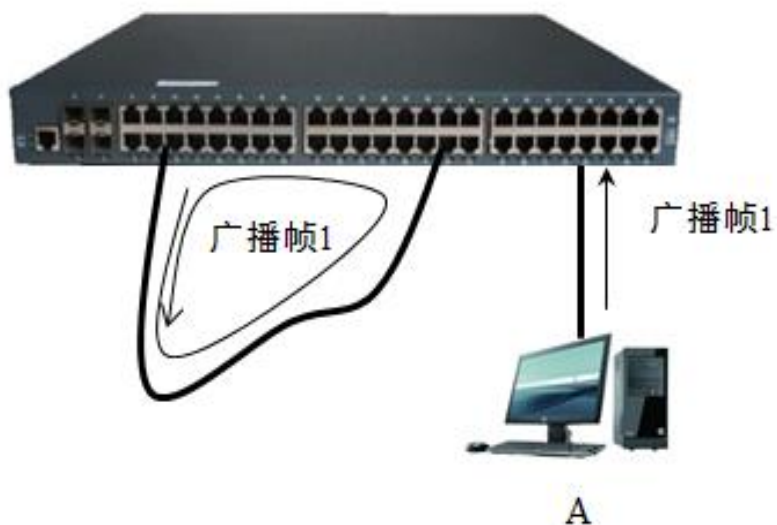
4.4.9生成树协议

- 双汇聚层
- 有环路
- 形成广播风暴



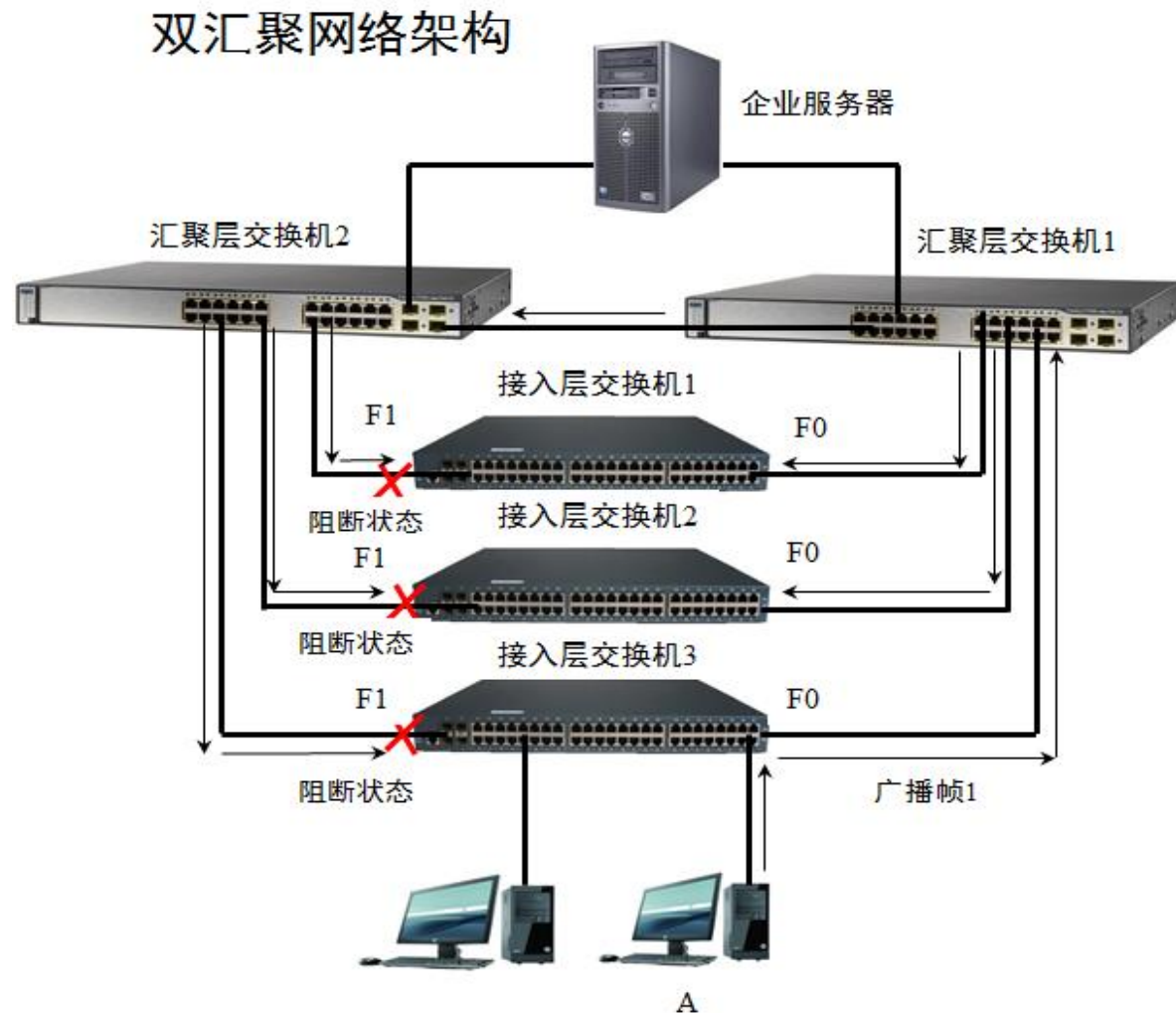
4.4.9生成树协议

- 只要有环路就能形成广播风暴



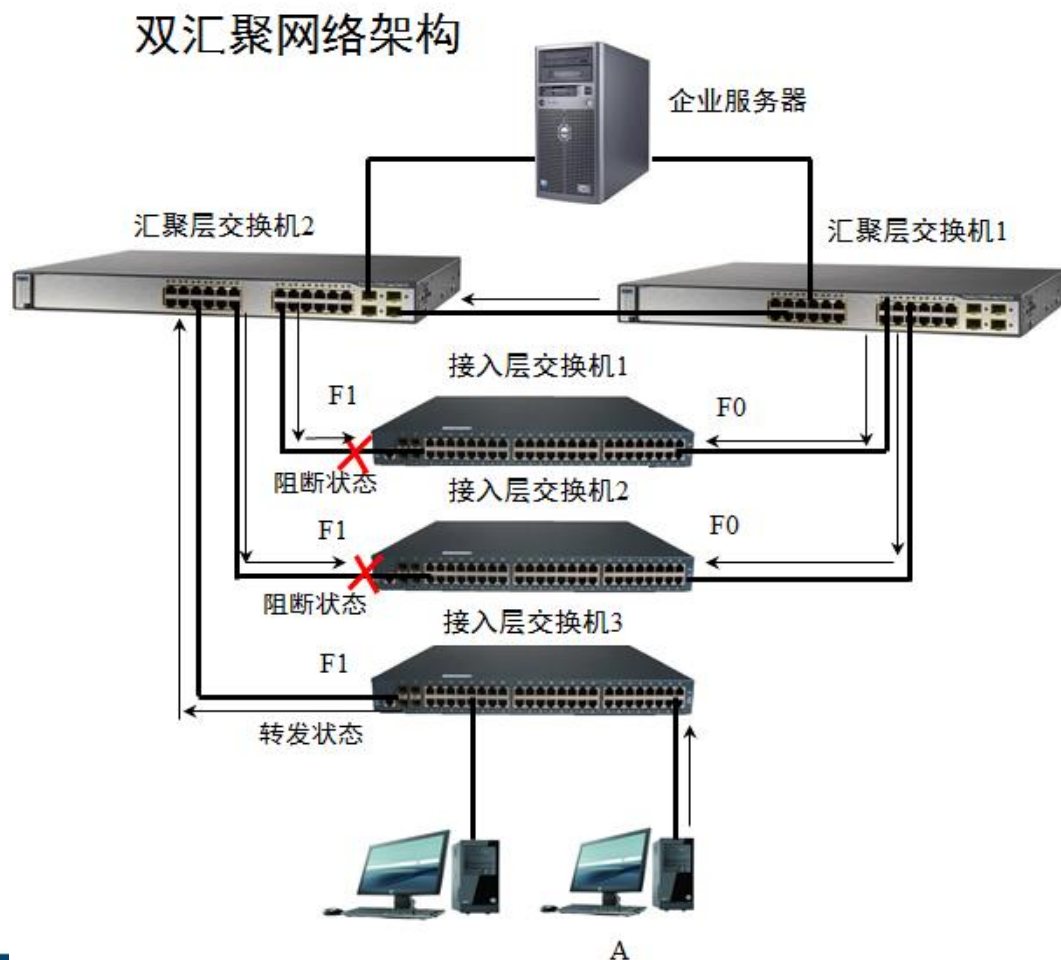
4.4.9生成树协议

■ 生成树协议阻断环路



4.4.9生成树协议

- 链路有变化会重新运行生成树算法
- 将原来的阻断端口更为转发端口



4.5高速以太网

■ 4.5.1 100M以太网

- 100BASE-T是在双绞线上传送100Mb/s基带信号的星型拓扑的以太网，仍使用IEEE802.3的CSMA/CD协议，它又称为快速以太网（FastEthernet）。
- 使用交换机组建的100BASE-T以太网，可在全双工方式下工作而无冲突发生。因此，CSMA/CD协议对全双工方式工作的快速以太网是不起作用的。因为其帧格式和以太网一样，所以依然称交换机组件的网络为以太网。
- 以太网的最短帧和带宽和链路长度有关，100M以太网比10M以太网速率提高10倍，要想和10M以太网兼容，就要确保最短帧也是64字节，那就将电缆最大长度由1000m降到100m，因此以太网的争用期依然是5.12 μ s，最短帧依然是64字节。

4.5.1 100M以太网

- 快速以太网100M带宽，有以下标准：

名称↵	传输介质↵	网段最大长度↵	特点↵
100BASE-TX↵	铜缆↵	100 米↵	两对 UTP5 类线或屏蔽双绞线↵
100BASE-T4↵	铜缆↵	100 米↵	4 对 UTP3 类线或 5 类线↵
100BASE-FX↵	光纤↵	2000 米↵	两根光纤，发送和接收各用一根，全双工，长距离↵

4.5.2吉比特以太网

- 吉比特以太网的标准IEEE802.3 z有以下几个特点：
 - 允许在1Gb/s下全双工和半双工两种方式工作。
 - 使用IEEE802.3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。

4.5.2吉比特以太网

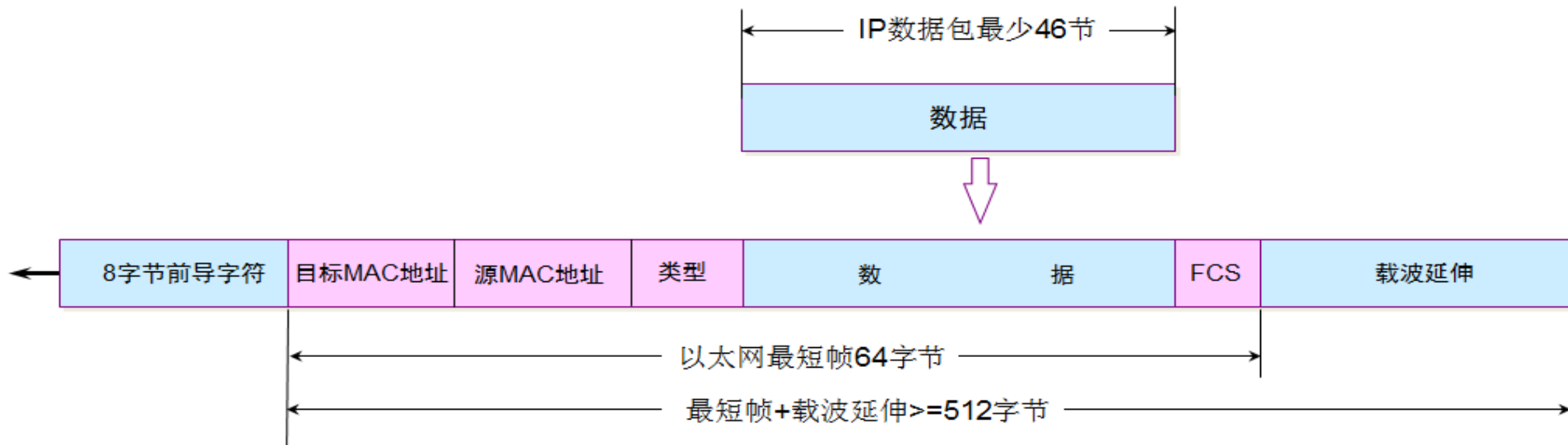
- 吉比特以太网1000M带宽，有以下标准：

名称↵	传输介质↵	网段最大长度↵	特点↵
1000BASE-SX↵	光缆↵	550 米↵	多模光纤（10 和 62.5μm）↵
1000BASE-LX↵	光缆↵	5000 米↵	单模光纤（10μm）多模光纤（50μm 和 62.5μm）↵
1000BASE-CX↵	铜线↵	25 米↵	使用两对屏蔽双绞线电缆 STP↵
1000BASE-T↵	铜线↵	100 米↵	使用 4 对 UTP 5 类线 ↵

4.5.2吉比特以太网

- 吉比特以太网工作在半双工时，就必须进行碰撞检测，数据速率提高了，要想和10M以太网兼容，就要确保最短帧也是64字节，这只能减少最大电缆长度，以太网最大电缆长度就要缩短到10m，短到几乎没有什么实用价值。吉比特以太网为了增加最大传输距离，将最短帧增加到4096比特。
- 当数据帧长度小于512字节（即4096比特）时，在FCS域后面添加“载波延伸”域。主机发送完短数据帧之后，继续发送载波延伸信号，冲突信号传回来时，发送端就能感知到了。

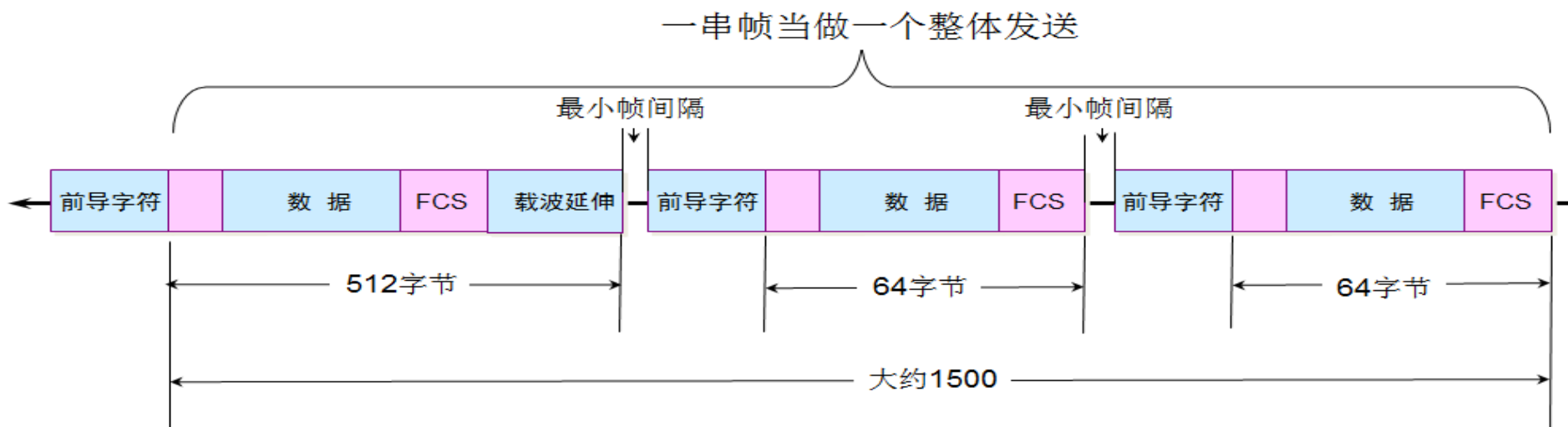
1000M以太网载波延伸示意图



4.5.2吉比特以太网

- 如果发送的数据帧都是64字节的短报文，那么链路的利用率就很低，因为“载波延伸”域将占用大量的带宽。
- 千兆以太网标准中，引入了“分组突发”（packet bursting）机制来改善这个问题。这就是当很多短帧要发送时，第一个短帧采用上面所说的载波延伸的方法进行填充，随后的一些短帧则可以一个接一个发送，它们之间只需要留有必要的帧间最小间隔即可。

分组突发示意图



4.5.3 10吉比特以太网

- 10GE的帧格式与10Mb/s, 100Mb/s和1Gb/S以太网的帧格式完全相同。10GE还保留了802.3标准规定的以太网最小和最大帧长。
- 由于数据率很高, 10GE不再使用铜线而只使用光纤作为传输媒体。它使用长距离(40km)的光收发器与单模光纤接口, 以便能够工作在广域网和城域网的范围。
- 10GE只工作在全双工模式, 因此不存在争用问题, 也不使用CSMA/CD协议。这就使得10GE的传输距离不再受碰撞检测的限制而大大提高了。

4.5.3 10吉比特以太网

■ 10GE的物理层有以下标准：

名称	传输介质	网段最大长度	特点
10GBASE-SR	光缆	300m	多模光纤（0.85μm）
10GBASE-LR	光缆	10km	单模光纤（1.3μm）
10GBASE-ER	光缆	40km	单模光纤（1.5μm）
10GBASE-CX4	铜线	15m	使用 4 对双芯同轴电缆
10GBASE-T	铜线	100m	使用 4 对 6A 类 UTP 双绞线