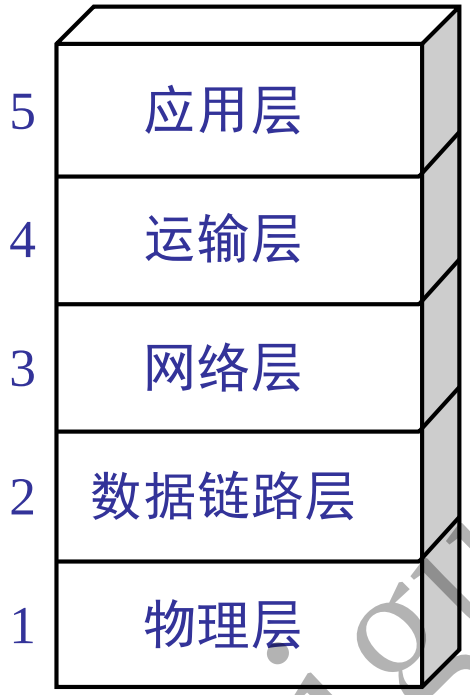
计算机网络

顾 军

计算机学院

jgu@cumt.edu.cn

专题4：数据包怎么在互联网中寻路和转发？

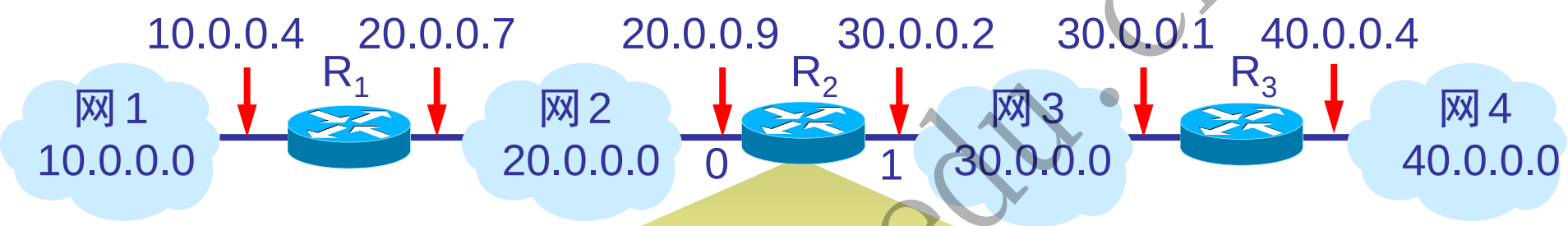


- 应用层(application layer)
- 运输层(transport layer)
- 网络层(network layer)
- 数据链路层(data link layer)
- 物理层(physical layer)

Q5：如何转发分类IP地址的分组？

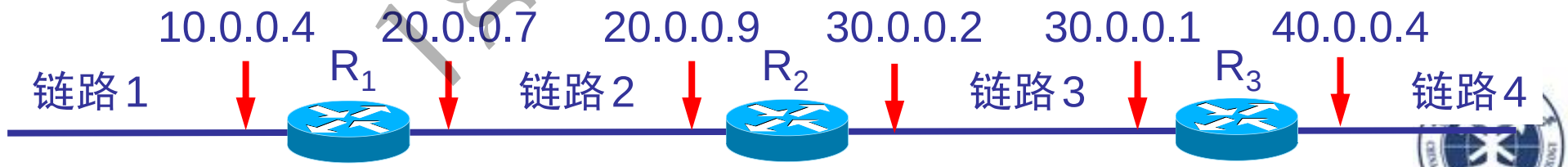
- 假设：有四个 A 类网络通过三个路由器连接在一起。每一个网络上都可能有成千上万个主机。
- 可以想像，若按目的主机号来制作路由表，每一个路由表就有 4 万个项目，即 4 万行（每一行对应于一台主机），则所得出的路由表就会过于庞大。
- 但若按主机所在的网络地址来制作路由表，那么每一个路由器中的路由表就只包含 4 个项目（每一行对应于一个网络），这样就可使路由表大大简化。

在路由表中，对每一条路由，最主要的是（目的网络地址，下一跳地址）



路由器 R_2 的路由表

目的主机所在的网络	下一跳地址
20.0.0.0	直接交付，接口 0
30.0.0.0	直接交付，接口 1
10.0.0.0	20.0.0.7
40.0.0.0	30.0.0.1



查找路由表

- 根据目的网络地址就能确定下一跳路由器，这样做的结果是：
  - IP 数据报最终一定可以找到目的主机所在目的网络上的路由器（可能要通过多次的间接交付）。
  - 只有到达最后一个路由器时，才试图向目的主机进行直接交付。
- 除路由表中的路由条目外，还有两类路由：
  - 特定主机路由
  - 默认路由

特定主机路由

- 虽然互联网所有的分组转发都是基于目的主机所在的网络，但在大多数情况下都允许有这样的特例，即为特定的目的主机指明一个路由。
- 采用特定主机路由可使网络管理人员能更方便地控制网络和测试网络，同时也可在需要考虑某种安全问题时采用这种特定主机路由。

默认路由(default route)

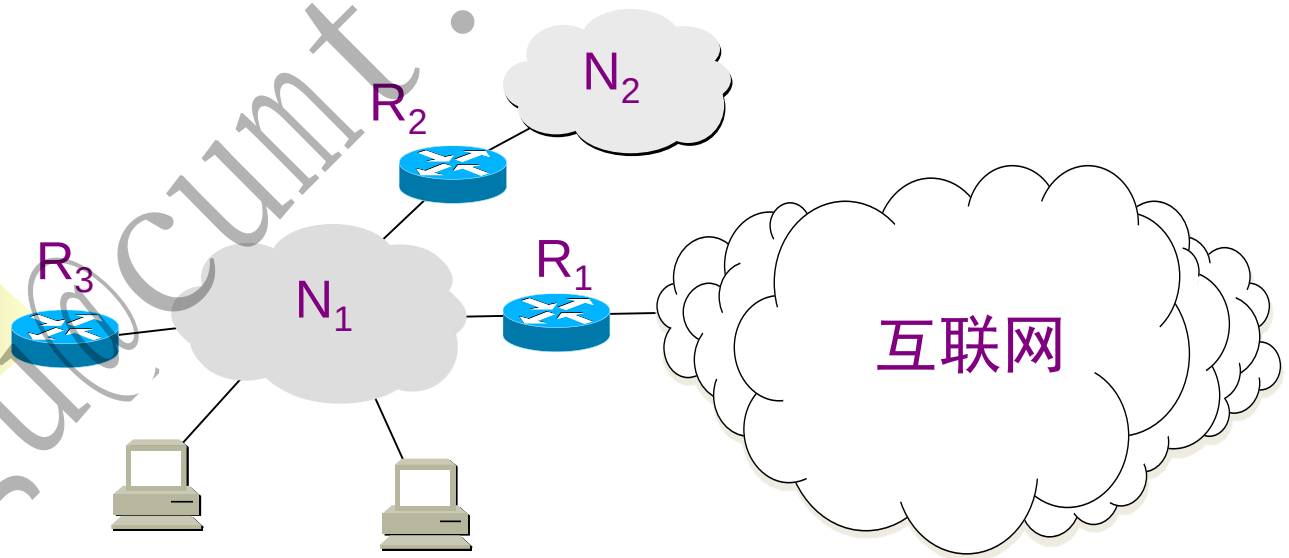
- 路由器还可采用默认路由以减少路由表所占用的空间和搜索路由表所用的时间。
- 这种转发方式在一个网络只有很少的对外连接时是很有用的。
- 默认路由在主机发送 IP 数据报时往往更能显示出它的好处。
- 如果一个主机连接在一个小网络上，而这个网络只用一个路由器和因特网连接，那么在这种情况下使用默认路由是非常合适的。

默认路由举例

只要目的网络不是 N_1 和 N_2，
就一律选择默认路由，
把数据报先间接交付路由器 R_1，
让 R_1 再转发给下一个路由器。

路由表

目的网络	下一跳
N_1	直接
N_2	R_2
默认	R_1



路由器 R_1 充当网络 N_1 的默认路由器

分类IP地址分组的路由转发算法

(1) 从数据报的首部提取目的主机的 IP 地址 D，得出目的网络地址为 N。

(2) 若网络 *N* 与此路由器直接相连，则把数据报直接交付目的主机 *D*；否则是间接交付，执行(3)。

(3) 若路由表中有目的地址为 *D* 的特定主机路由，则把数据报传送给路由表中所指明的下一跳路由器；否则，执行(4)。

(4) 若路由表中有到达网络 *N* 的路由，则把数据报传送给路由表指明的下一跳路由器；否则，执行(5)。

(5) 若路由表中有一个默认路由，则把数据报传送给路由表中所指明的默认路由器；否则，执行(6)。

(6) 报告转发分组出错。

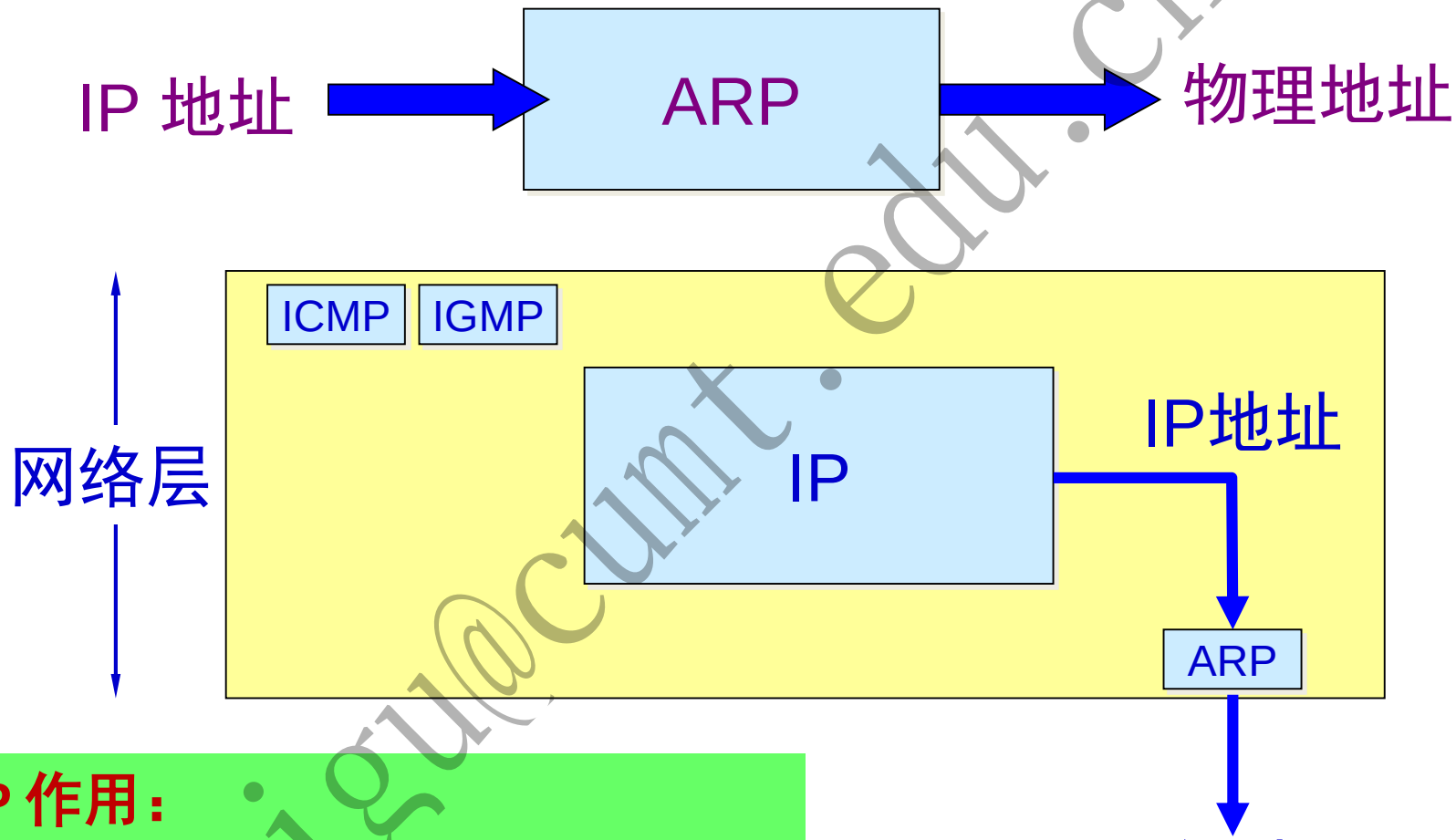
关于路由表

- 路由表没有给分组指明到某个网络的完整路径。
- 路由表指出，到某个网络应当先到某个路由器（即下一跳路由器）。
- 在到达下一跳路由器后，再继续查找其路由表，知道再下一步应当到哪一个路由器。
- 这样一步一步地查找下去，直到最后到达目的网络。

必须强调指出

- 当路由器收到待转发的数据报，不是将下一跳路由器的 IP 地址填入 IP 数据报，而是送交下层的网络接口软件。
- 不管网络层使用的是什么协议，在实际网络的链路上传送数据帧时，最终还是必须使用硬件地址。
- 网络接口软件使用 ARP 负责将下一跳路由器的 IP 地址转换成硬件地址，并将此硬件地址放在链路层的 MAC 帧的首部，然后根据这个硬件地址找到下一跳路由器。

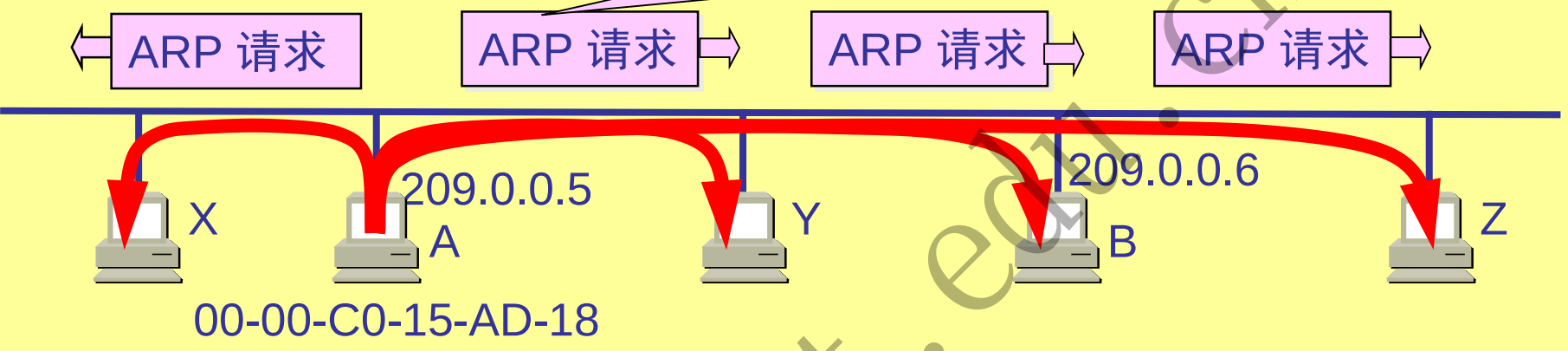
Q6：已知IP地址如何知道物理地址？



ARP 作用：
根据网络层使用的 IP 地址，解析出在数据链路层使用的硬件地址。

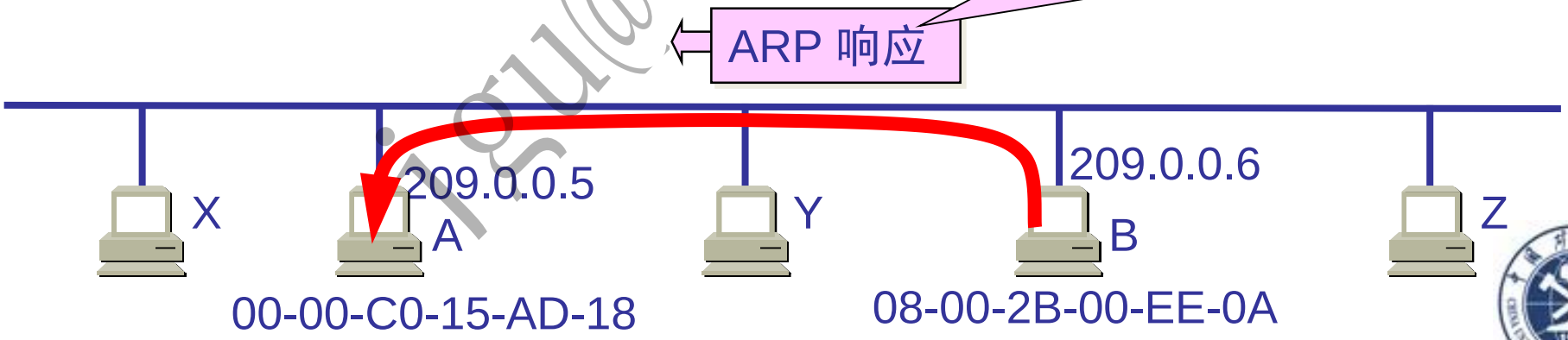
主机 A 广播发送
ARP 请求分组

我是 209.0.0.5，硬件地址是 00-00-C0-15-AD-18
我想知道主机 209.0.0.6 的硬件地址



主机 B 向 A 发送
ARP 响应分组

我是 209.0.0.6
硬件地址是 08-00-2B-00-EE-0A



ARP 协议分组

- ARP请求分组：包含发送方硬件地址 / 发送方 IP 地址 / 目标方硬件地址(未知时填 0) / 目标方 IP 地址。
  - 为了减少网络上的通信量，主机 A 在发送其 ARP 请求分组时，就将自己的 IP 地址到硬件地址的映射写入 ARP 请求分组。
- 本地广播 ARP 请求（路由器不转发ARP请求）。
- ARP 响应分组：包含发送方硬件地址 / 发送方 IP地址 / 目标方硬件地址 / 目标方 IP 地址。
  - 当主机 B 收到 A 的 ARP 请求分组时，就将主机 A 的这一地址映射写入主机 B 自己的 ARP 高速缓存中。这对主机 B 以后向 A 发送数据报时就更方便了。
- ARP 分组封装在物理网络的帧中直接传输。

ARP协议报文格式

硬件地址类型		协议地址类型
硬件地址长度	协议地址长度	操作
发送方硬件地址(第0~第3字节)		
发方硬件地址(第4~第5字节)		发方协议地址(第0~第1字节)
发方协议地址(第2~第3字节)		目的硬件地址(第0~第1字节)
目的硬件地址(第2~第5字节)		
目的协议地址(第0~第3字节)		

ARP报文使用链路层数据帧封装

ARP 高速缓存

- 每一个主机都设有一个 ARP 高速缓存 (ARP cache)，里面存放最近获得的所在局域网上的各主机和路由器的 IP 地址到硬件地址的映射表，以减少 ARP 广播的数量。

< IP address；MAC address；TTL >

TTL (Time To Live)：地址映射有效时间 。

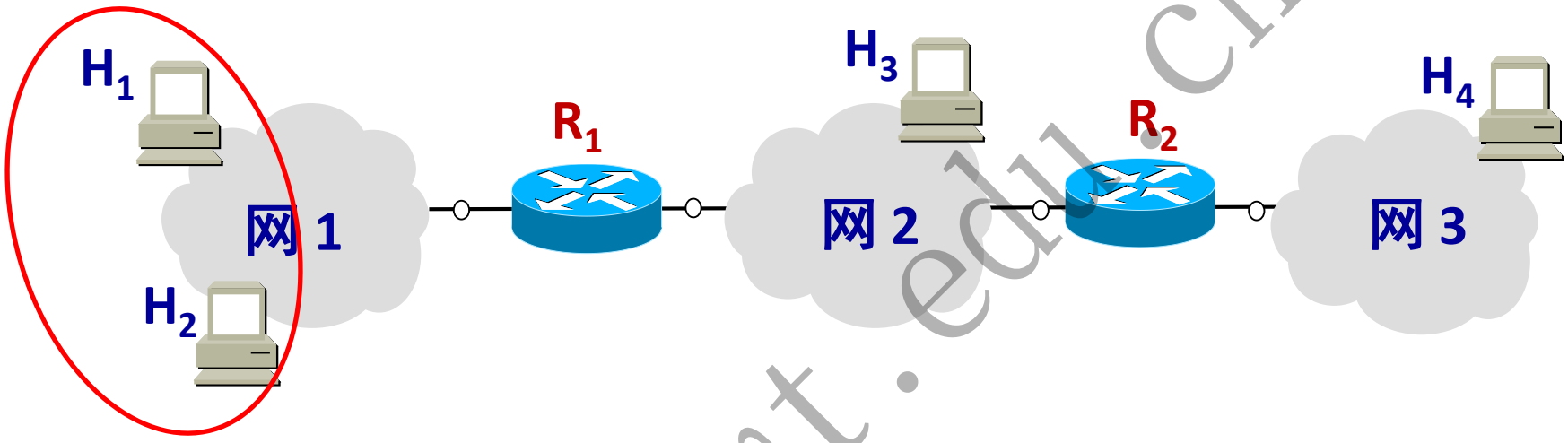
ARP 高速缓存的作用

- 存放最近获得的 IP 地址到 MAC 地址的绑定，以减少 ARP 广播的数量。
- 为了减少网络上的通信量，主机 A 在发送其 ARP 请求分组时，就将自己的 IP 地址到硬件地址的映射写入 ARP 请求分组。
- 当主机 B 收到 A 的 ARP 请求分组时，就将主机 A 的这一地址映射写入主机 B 自己的 ARP 高速缓存中。这对主机 B 以后向 A 发送数据报时就更方便了。

ARP 协议的使用

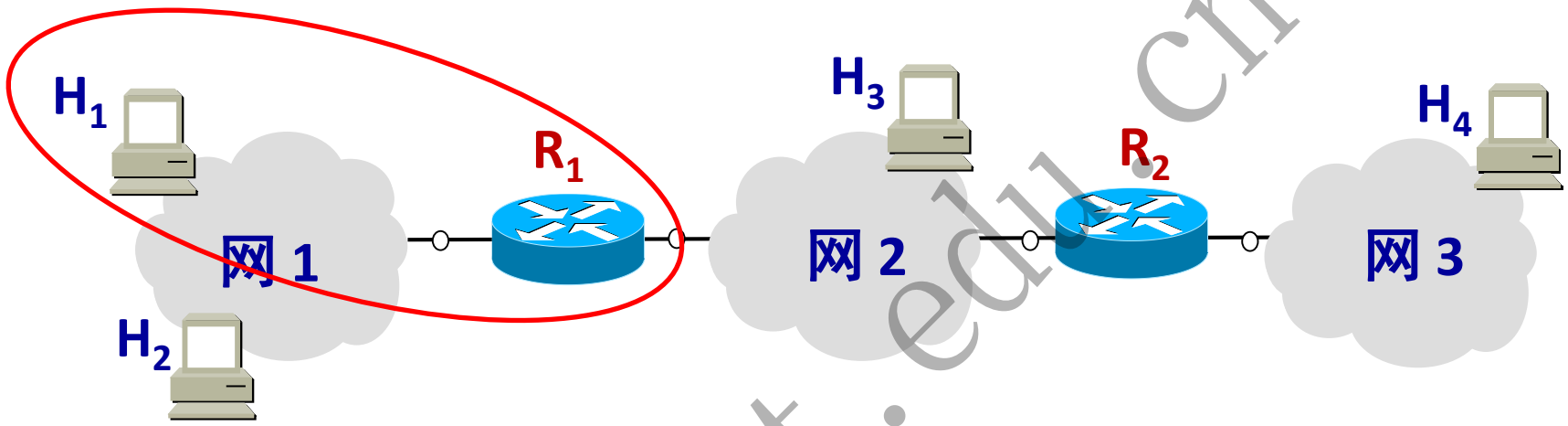
- 当主机 A 欲向本局域网上的某个主机 B 发送 IP 数据报时，就先在其 ARP 高速缓存中查看有无主机 B 的 IP 地址。
  - 如有，就可查出其对应的硬件地址，再将此硬件地址写入 MAC 帧，然后通过局域网将该 MAC 帧发往此硬件地址。
  - 如没有， ARP 进程在本局域网上广播发送一个 ARP 请求分组。收到 ARP 响应分组后，将得到的 IP 地址到硬件地址的映射写入 ARP 高速缓存。

Q7：什么情况下使用ARP协议？



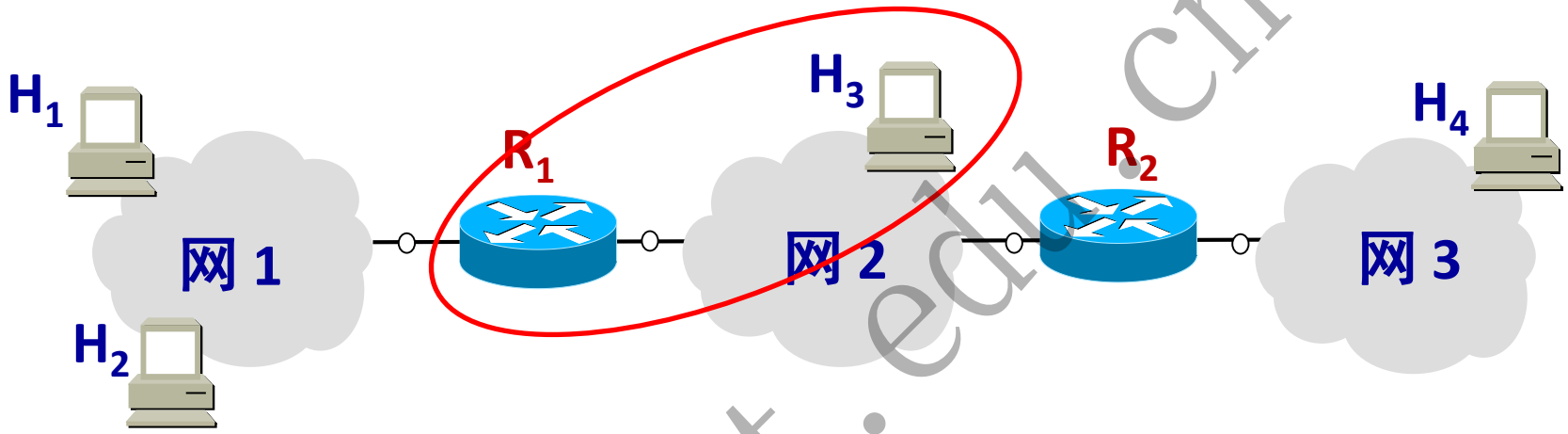
- 典型情况一：
  - 发送方是主机，要把IP数据报发送到本网络上的另一个主机。这时用 ARP 找到目的主机的硬件地址。

使用 ARP 的四种典型情况



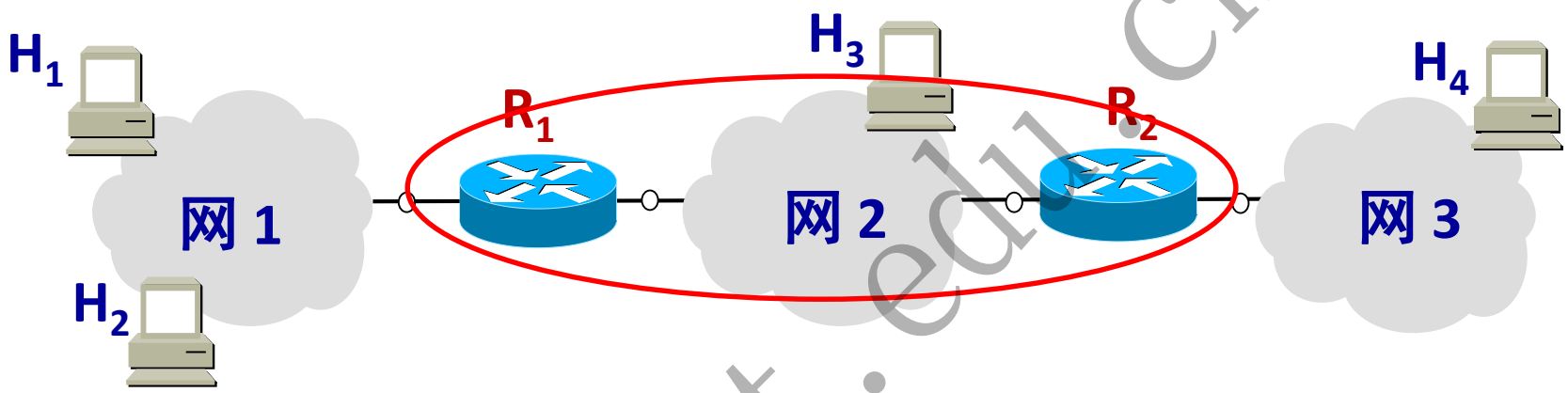
- 典型情况二：
  - 发送方是主机，要把 IP 数据报发送到另一个网络上的一个主机。这时用 ARP 找到本网络上的一个路由器接口（默认网关）的硬件地址。剩下的工作由这个路由器来完成。

使用 ARP 的四种典型情况



- 典型情况三：
  - 发送方是路由器，要把 IP 数据报转发到本网络上的一个主机。这时用 ARP 找到目的主机的硬件地址。

使用 ARP 的四种典型情况



- 典型情况四：
  - 发送方是路由器，要把 IP 数据报转发到另一个网络上的一个主机。这时用 ARP 找到本网络上另一个路由器的硬件地址。剩下的工作由这个路由器来完成。

ARP解析过程是自动执行的

- IP地址到硬件地址的解析由网络接口软件自动进行，主机的用户对这种地址解析过程是看不见的，从而给广大计算机用户带来很大的方便。
  - 只要主机或路由器要和本网络上的另一个已知 IP 地址的主机或路由器进行通信，ARP 协议会自动地将该 IP 地址解析为链路层所需的硬件地址。——直接交付
  - 如果所要找的主机和源主机不在同一个局域网上，那么就要通过 ARP 找到一个位于本局域网上的某个路由器的硬件地址，然后把分组发送给这个路由器，由这个路由器把分组转发给下一个网络，剩下工作由下一个网络来做。——间接交付

ARP命令

显示和修改地址解析协议(ARP)使用的“IP 到物理”地址转换表。

```
ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]
```

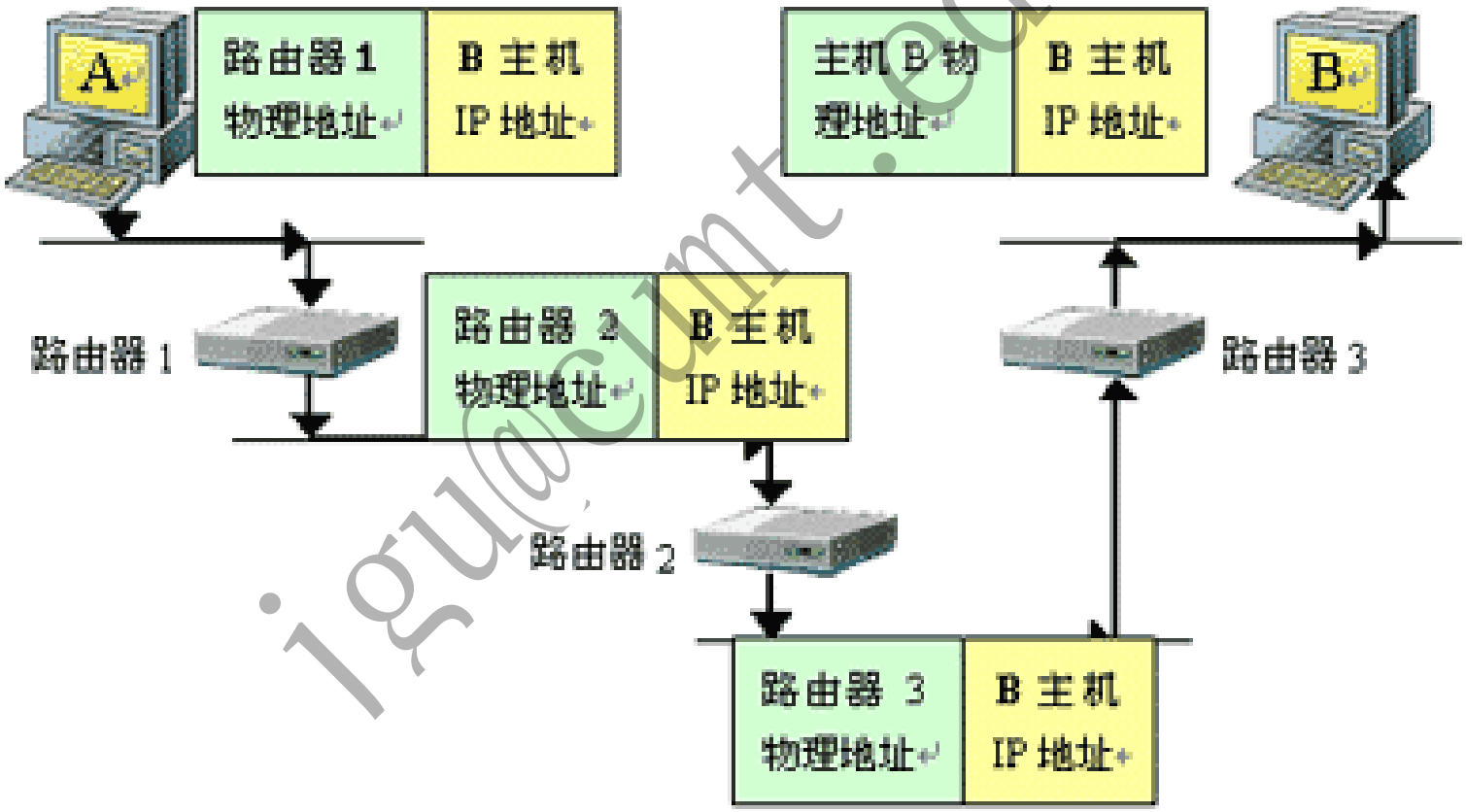
-a	通过询问当前协议数据，显示当前 ARP 项。如果指定 inet_addr，则只显示指定计算机的 IP 地址和物理地址。如果不止一个网络接口使用 ARP，则显示每个 ARP 表的项。
-g	与 -a 相同。
-v	在详细模式下显示当前 ARP 项。所有无效项和环回接口上的项都将显示。
inet_addr	指定 Internet 地址。
-N if_addr	显示 if_addr 指定的网络接口的 ARP 项。
-d	删除 inet_addr 指定的主机。inet_addr 可以是通配符 *，以删除所有主机。
-s	添加主机并且将 Internet 地址 inet_addr 与物理地址 eth_addr 相关联。物理地址是用连字符分隔的 6 个十六进制字节。该项是永久的。
eth_addr	指定物理地址。
if_addr	如果存在，此项指定地址转换表应修改的接口的 Internet 地址。如果不存在，则使用第一个适用的接口。

示例:

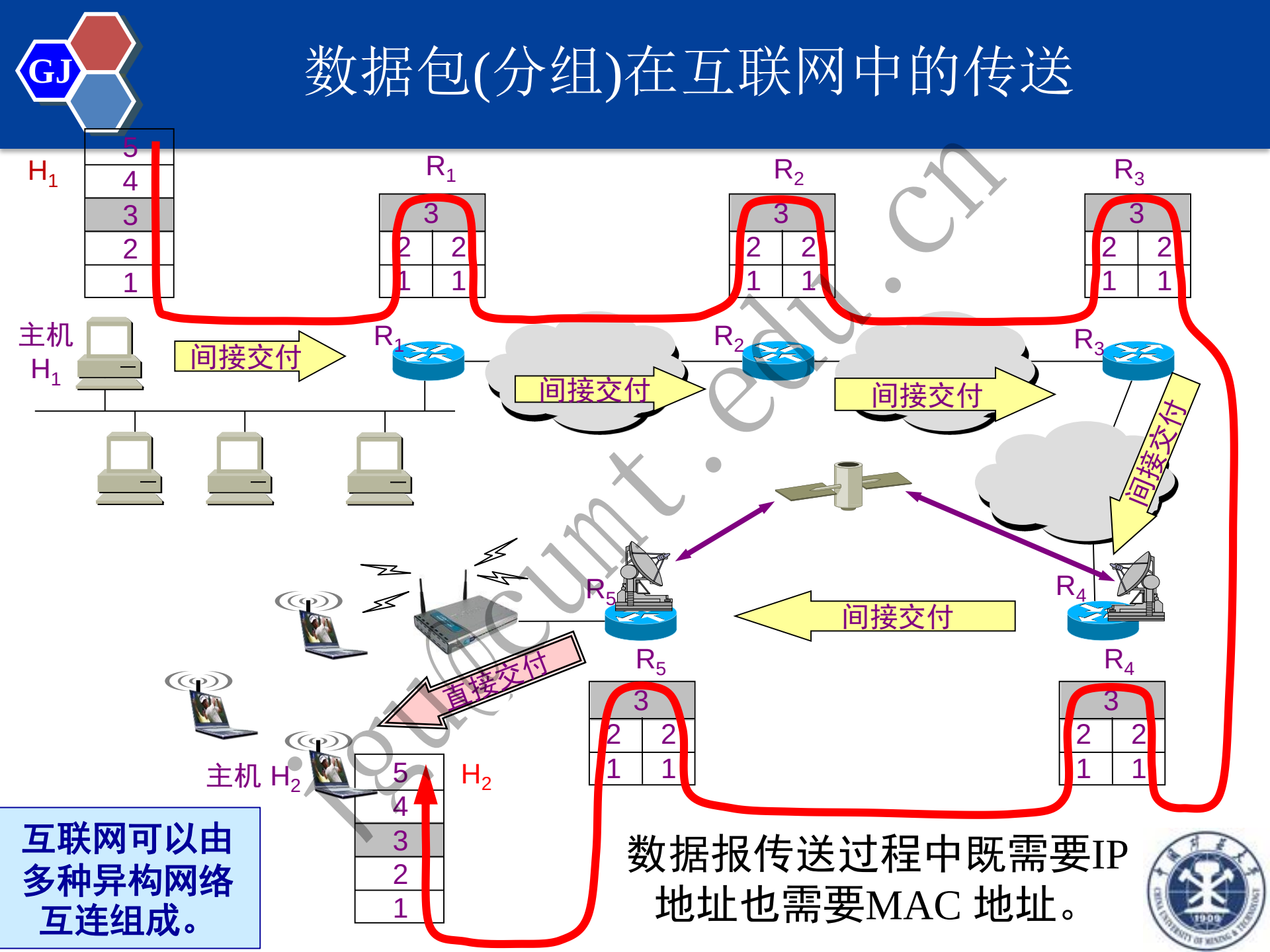
```
  > arp -s 157.55.85.212   00-aa-00-62-c6-09.... 添加静态项。
  > arp -a                                .... 显示 ARP 表。
```

Q8：IP地址与物理地址的动态映射？

IP数据包转发过程中，IP报文首部中的源和目的IP地址保持不变，但每条链路上数据帧的源和目的MAC地址是不同的。



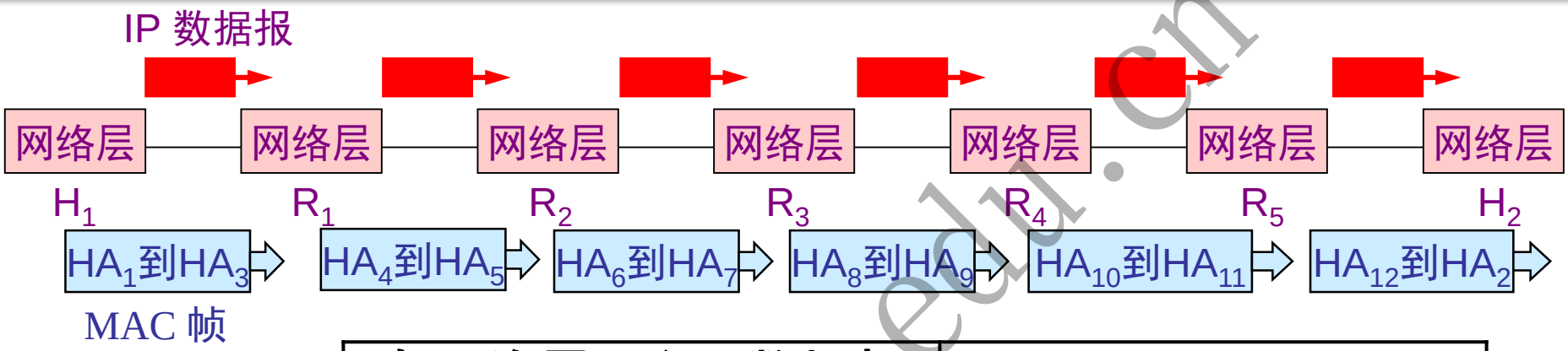
数据包(分组)在互联网中的传送



互联网可以由多种异构网络互连组成。

数据报传送过程中既需要IP地址也需要MAC 地址。

主机 H_1 与 H_2 通信中使用的 IP地址 与 硬件地址HA

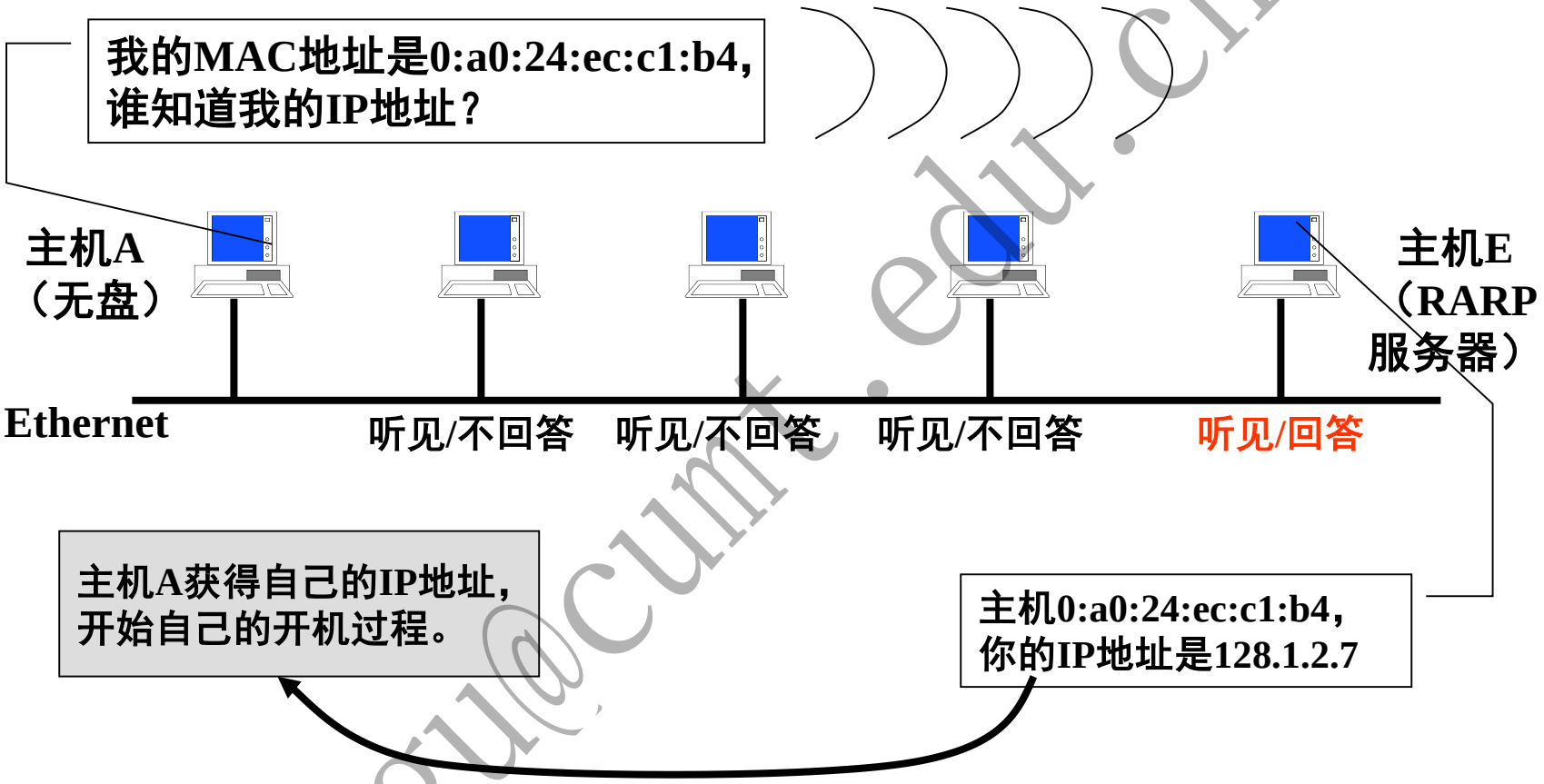


	在网络层写入IP数据报首部的地址		在数据链路层写入MAC帧首部的地址	
	源地址	目的地址	源地址	目的地址
从H_1到R_1	IP_1	IP_2	HA_1	HA_3
从R_1到R_2	IP_1	IP_2	HA_4	HA_5
从R_2到R_3	IP_1	IP_2	HA_6	HA_7
从R_3到R_4	IP_1	IP_2	HA_8	HA_9
从R_4到R_5	IP_1	IP_2	HA_{10}	HA_{11}
从R_6到H_2	IP_1	IP_2	HA_{12}	HA_2

Q9：只有硬件地址如何知道IP地址？

- 逆地址解析协议 RARP 使只知道自己硬件地址的主机能够知道其 IP 地址。
- 这种主机往往是无盘工作站。 因此 RARP 协议目前已很少使用。

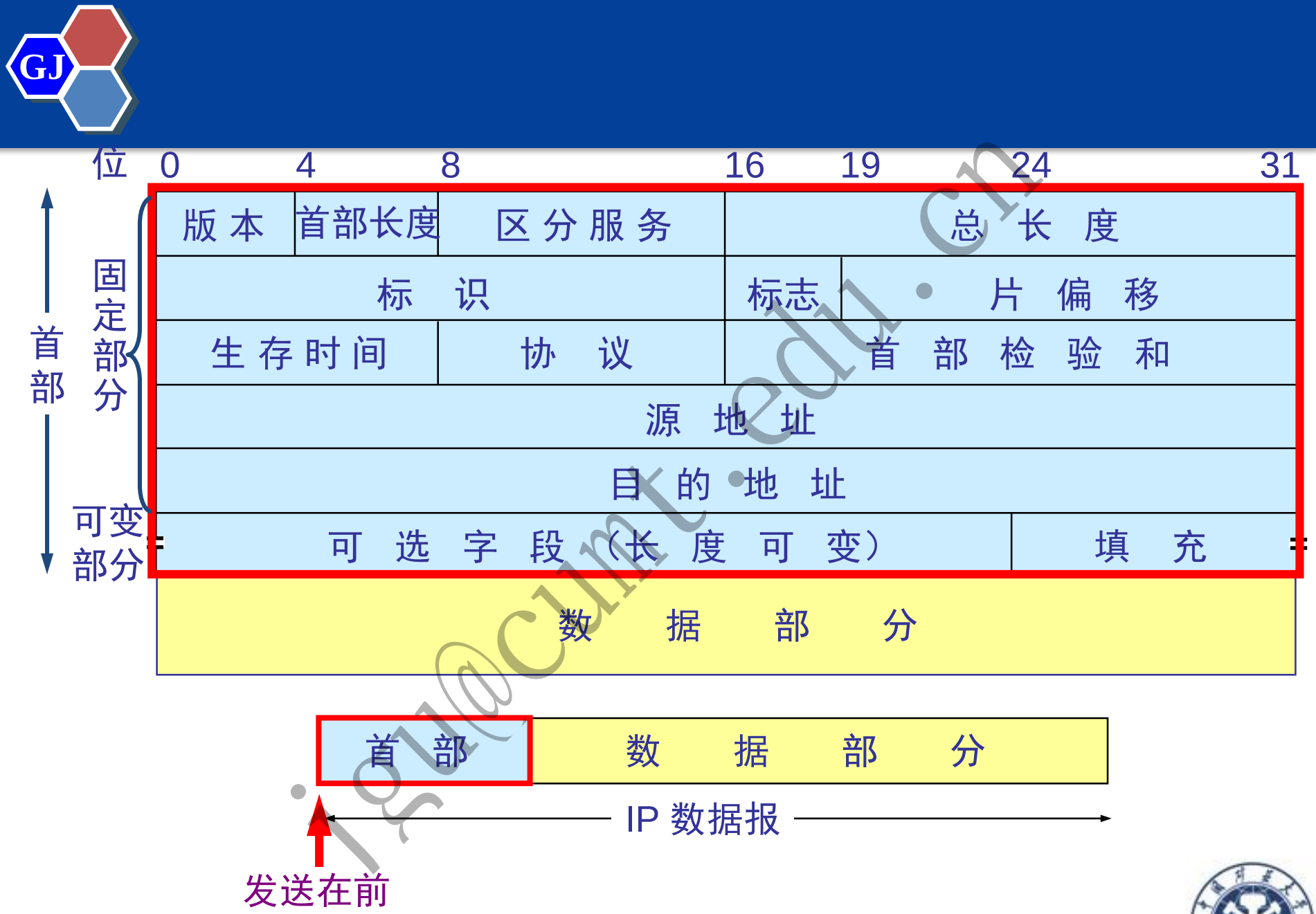
RARP地址的工作过程



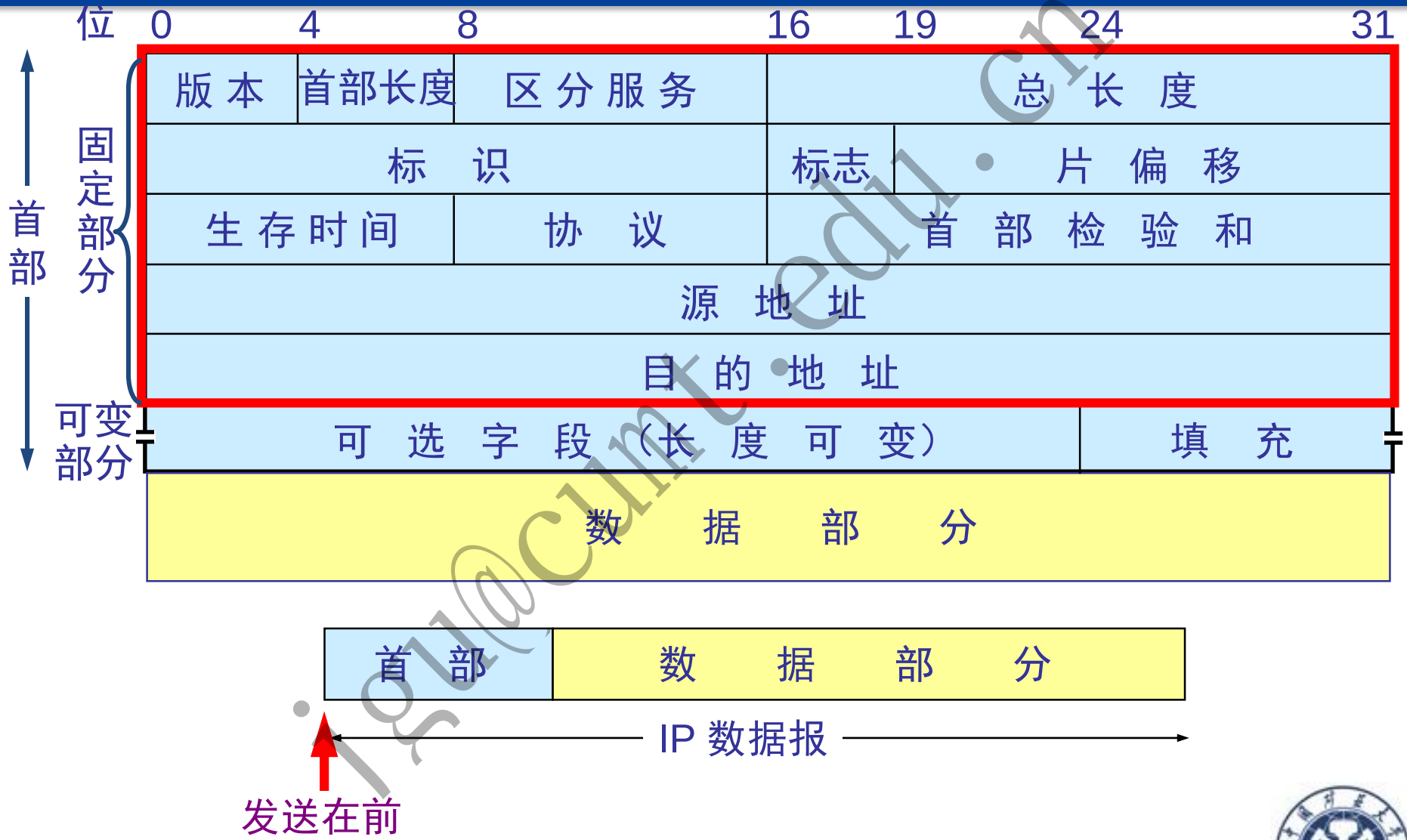
RARP和ARP的报文格式相同，都使用链路层数据帧封装

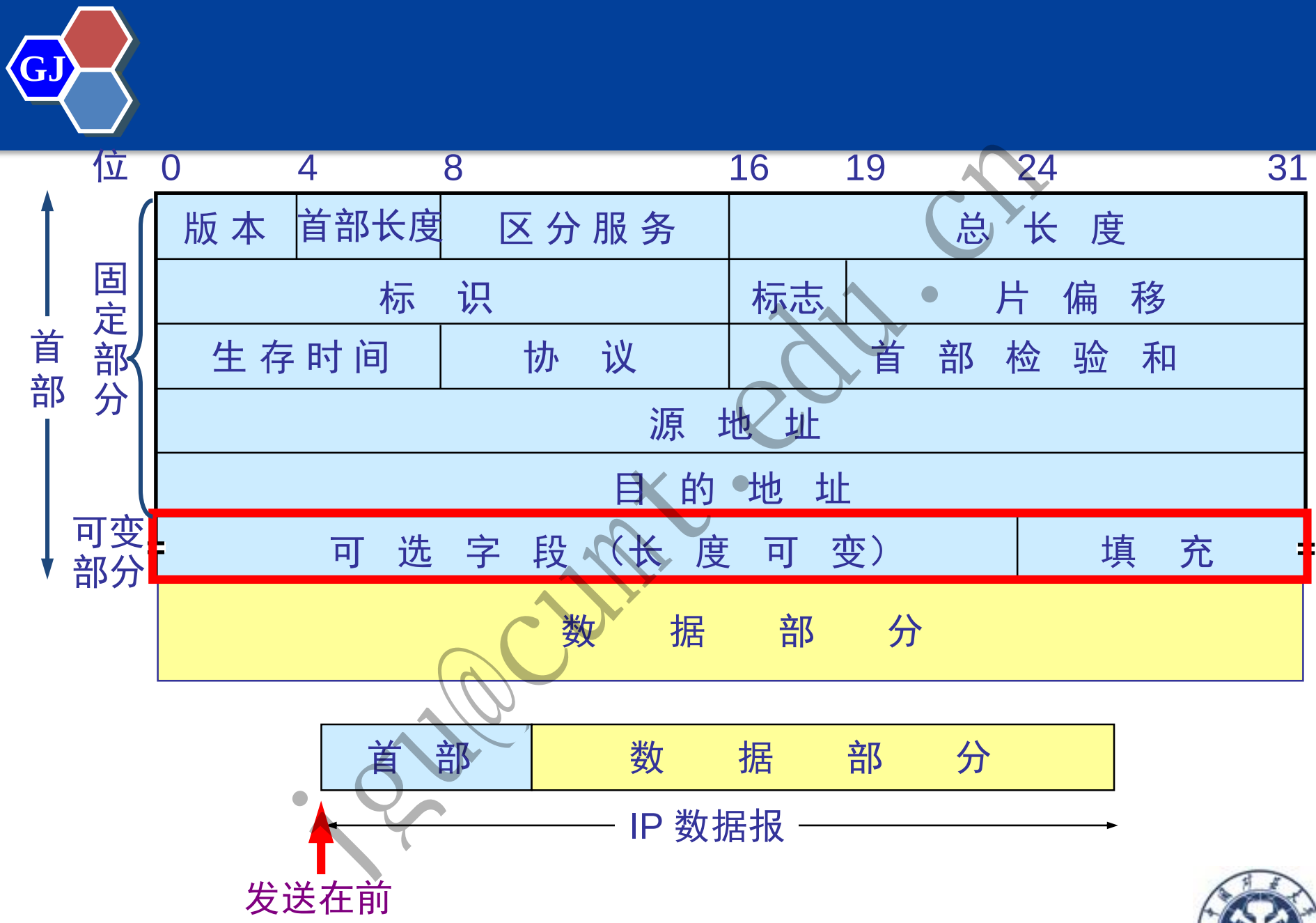
Q10：IP 数据报长啥样？

- 一个 IP 数据报（分组）由首部和数据两部分组成。
- 首部的前一部分是固定长度，共 20 字节，是所有 IP 数据报必须具有的。
- 在首部的固定部分的后面是一些可选字段，其长度是可变的。

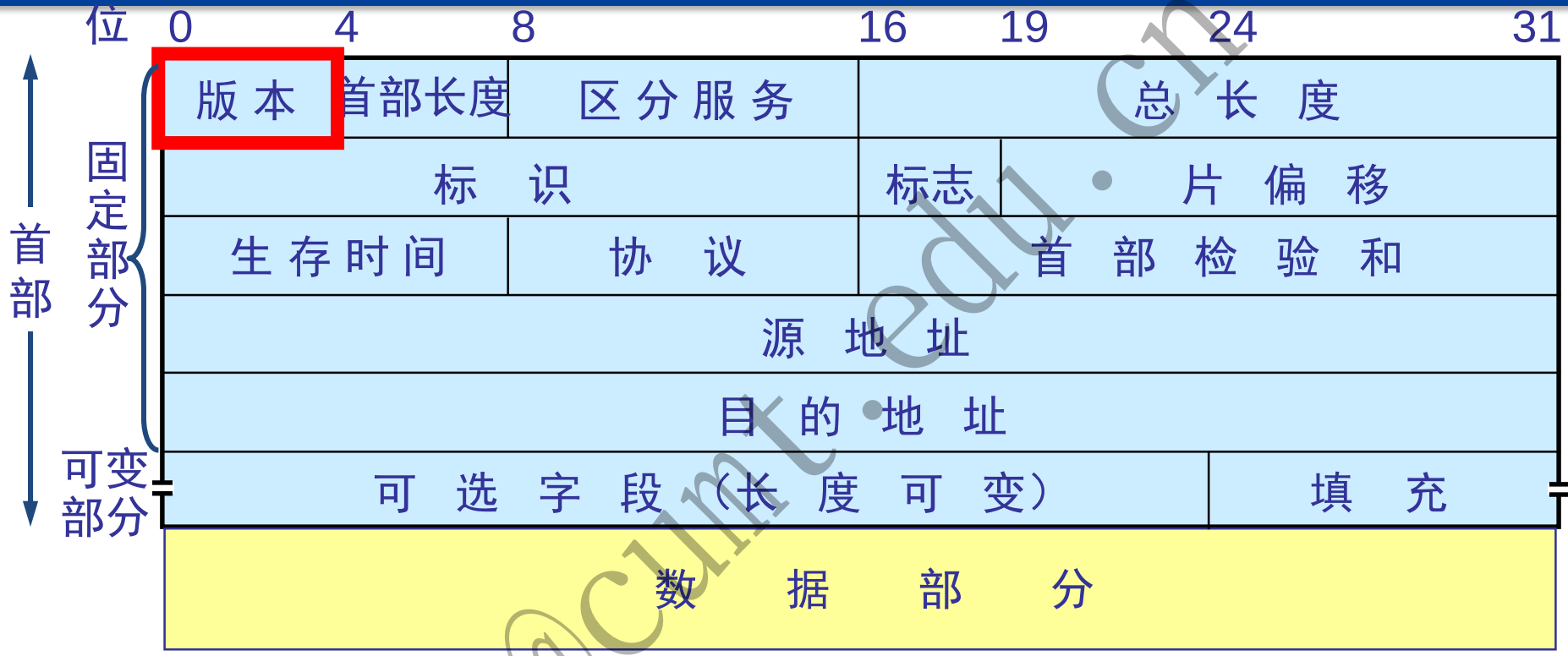

位
0
4
8
16
19
24
31
首部
固定部分
可变部分
版本
首部长度
区分服务
总长度
标识
标志
片偏移
生存时间
协议
首部检验和
源地址
目的地址
可选字段（长度可变）
填充
数据部分
首部
数据部分
IP 数据报
发送在前

首部的前一部分是固定长度，共 20 字节，是所有 IP 数据报必须具有的。

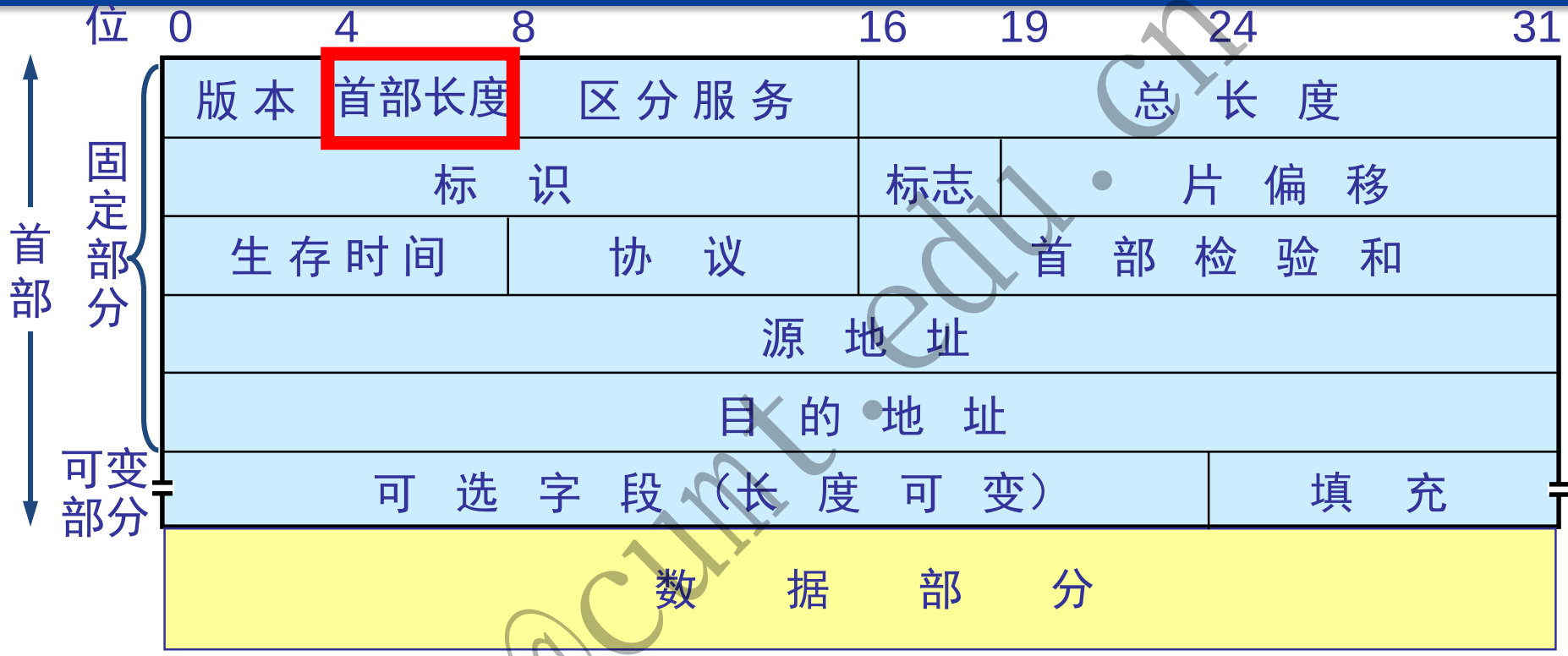



位
0
4
8
16
19
24
31
首部
固定部分
版本
首部长度
区分服务
总长度
标识
标志
片偏移
生存时间
协议
首部检验和
源地址
目的地址
可变部分
可选字段（长度可变）
填充
数据部分
首部
数据部分
IP 数据报
发送在前

IP 数据报首部的固定部分



版本——占 4 位，指 IP 协议的版本
目前的 IP 协议版本号为 4 (即 IPv4)



首部长度——占 4 位，可表示的最大数值是 15 个单位(一个单位为 4 字节)，首部最大长度是 60 字节。

数据部分在4字节的整数倍时开始，可以方便IP协议的实现。

区分服务——占 8 位，用来获得更好的服务

在旧标准中叫做服务类型，但实际上一直未被使用过。

1998 年这个字段改名为区分服务。

只有在使用区分服务（DiffServ）时，这个字段才起作用。

在一般的情况下都不使用这个字段

总长度——占 16 位，指首部和数据之和的长度，单位为字节，因此数据报的最大长度为 65535 字节。实际上这样长的数据报极少遇到。

总长度（首部加上数据部分）一定不能超过下面的数据链路层所规定的最大传送单元 MTU。

以太网规定其MTU值是1500字节， 如果超过MTU值，就必须进行分片处理。

- ✓ 如果IP数据报的数据部分长度尽可能长，那么首部长度占数据报总长度的比例就会减小，从而提高传输效率。
- ✓ 但是，数据报短些也有好处。每一个IP数据报越短，路由转发的速度就越快。

- IP协议规定，在互联网中所有主机和路由器，必须能够接受长度不超过576字节的数据报。
- 这是假定上层交下来的数据长度有512字节（合理长度），加上最长的IP首部60字节，再加上4字节的富余量。

- 当主机需要发送长度超过576字节的数据报时，应当先了解目的主机是否能够接受所要发送的数据报长度。否则，就要分片。

标识(identification) 占 16 位，
IP软件在存储器中维持一个计数器，每产生一个数据报，标识就加1。
但整个“标识”并不是序号，IP不存在按序接收问题

标志(flag) 占 3 位，目前只有前两位有意义。

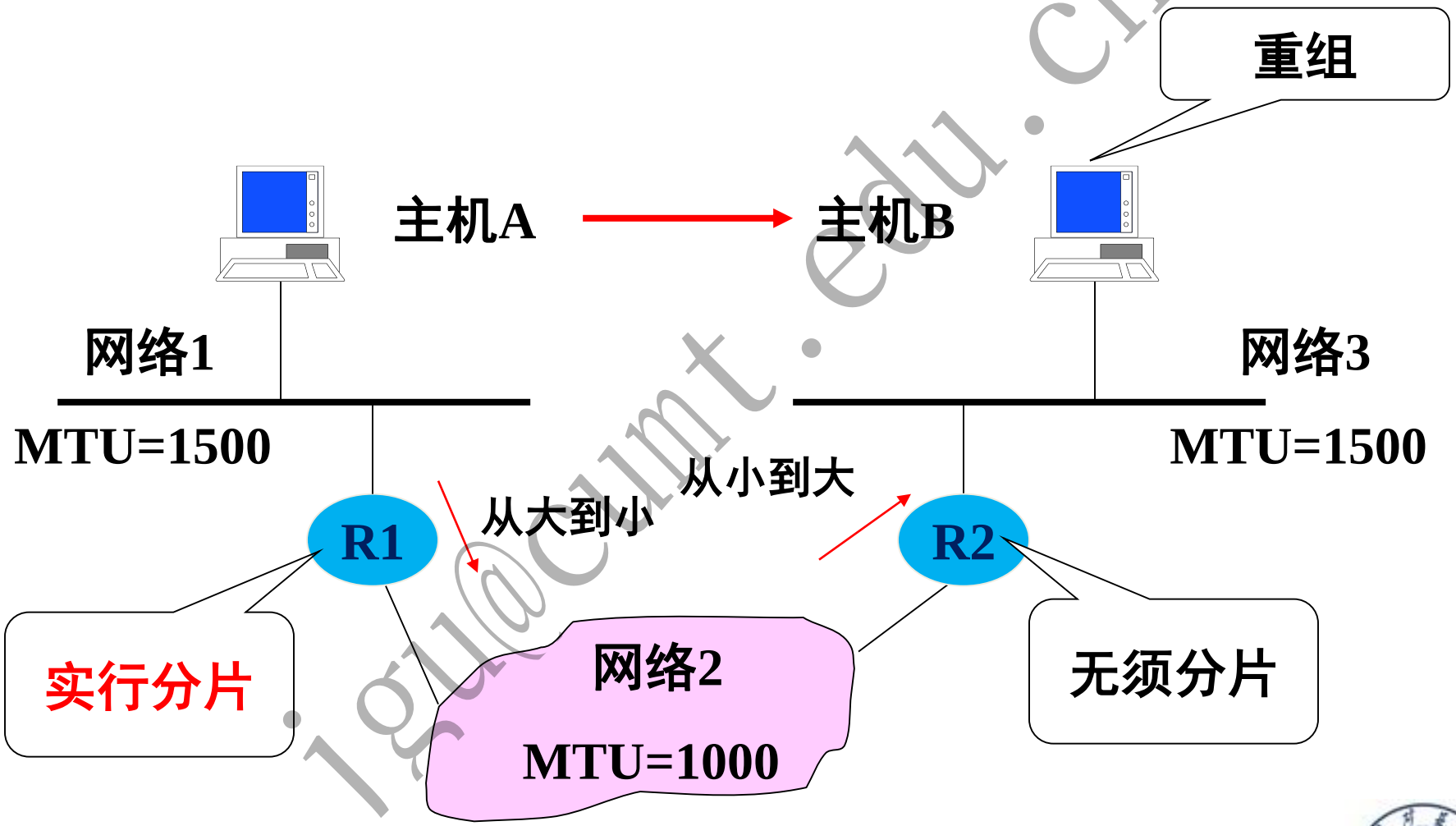
标志字段的最低位是 **MF** (More Fragment)。

MF = 1 表示后面“还有分片”。MF = 0 表示最后一个分片。

标志字段中间的一位是 **DF** (Don't Fragment) 。

只有当 DF = 0 时才允许分片。

IP 数据报的分片与重组

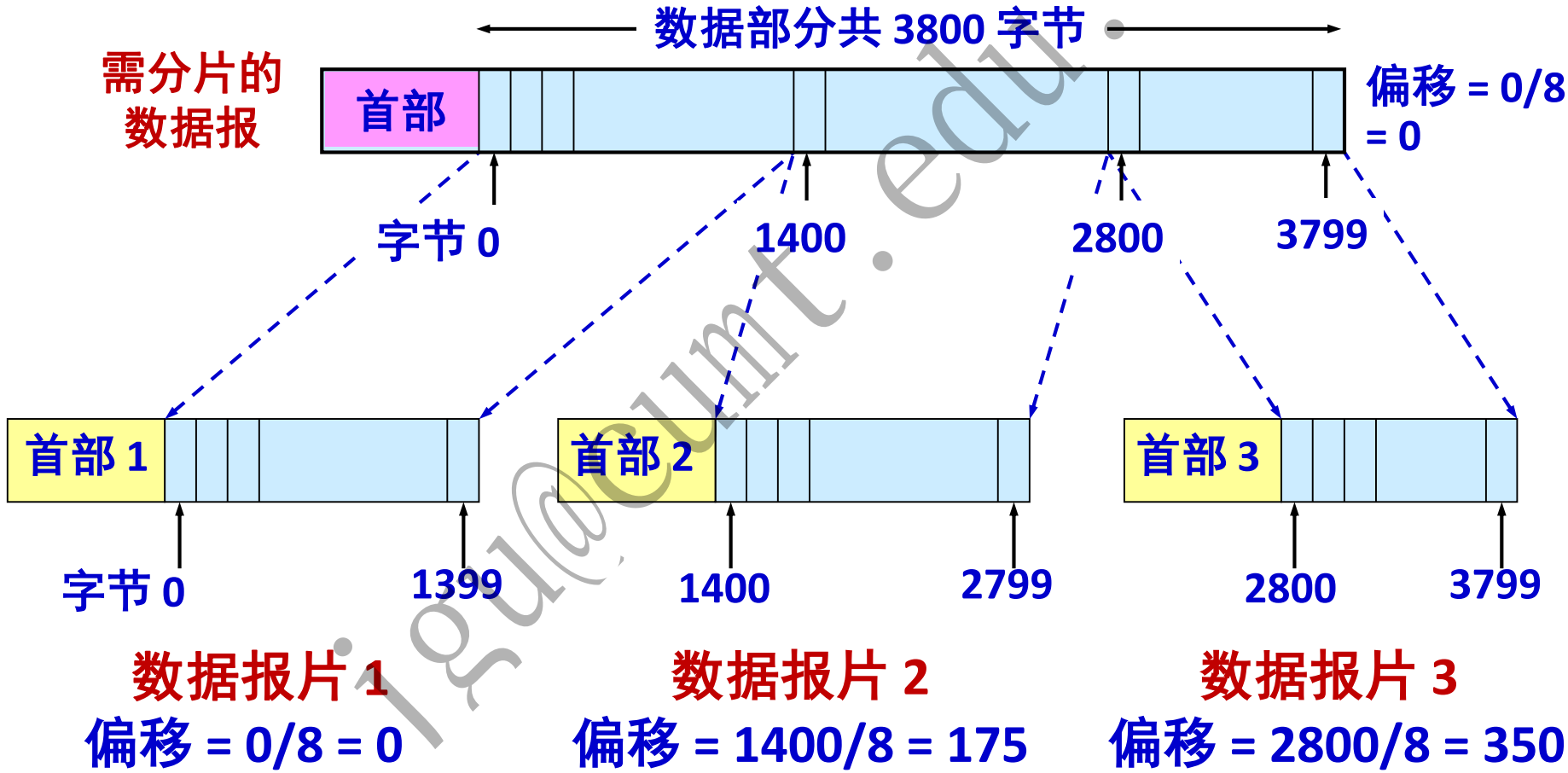


片偏移(13 位)指出：较长的分组在分片后某片在原分组中的相对位置。

片偏移以 8 个字节为偏移单位。

IP 数据报分片示例

总长度3820字节(20+3800)，分片为长度不超过1420字节(1400+20)。



IP 数据报分片示例

IP 数据报首部中与分片有关的字段中的数值

	总长度	标识	MF	DF	片偏移
原始数据报	3820	12345	0	0	0
数据报片1	1420	12345	1	0	0
数据报片2	1420	12345	1	0	175
数据报片3	1020	12345	0	0	350

原始数据报首部被复制为各数据报片的首部，但必须修改有关字段的值。

除了最后一个分片，之前的分片必须是8的整数倍

IP 数据报分片示例

IP 数据报首部中与分片有关的字段中的数值

	总长度	标识	MF	DF	片偏移
原始数据报	3820	12345	0	0	0
数据报片1	1420	12345	1	0	0
数据报片2	1420	12345	1	0	175
数据报片3	1020	12345	0	0	350

再分片：800、600

数据报片2-1	820	12345	1	0	175
数据报片2-2	620	12345	1	0	275

175 + 800/8 = 275

生存时间(8 位)记为 TTL (Time To Live)

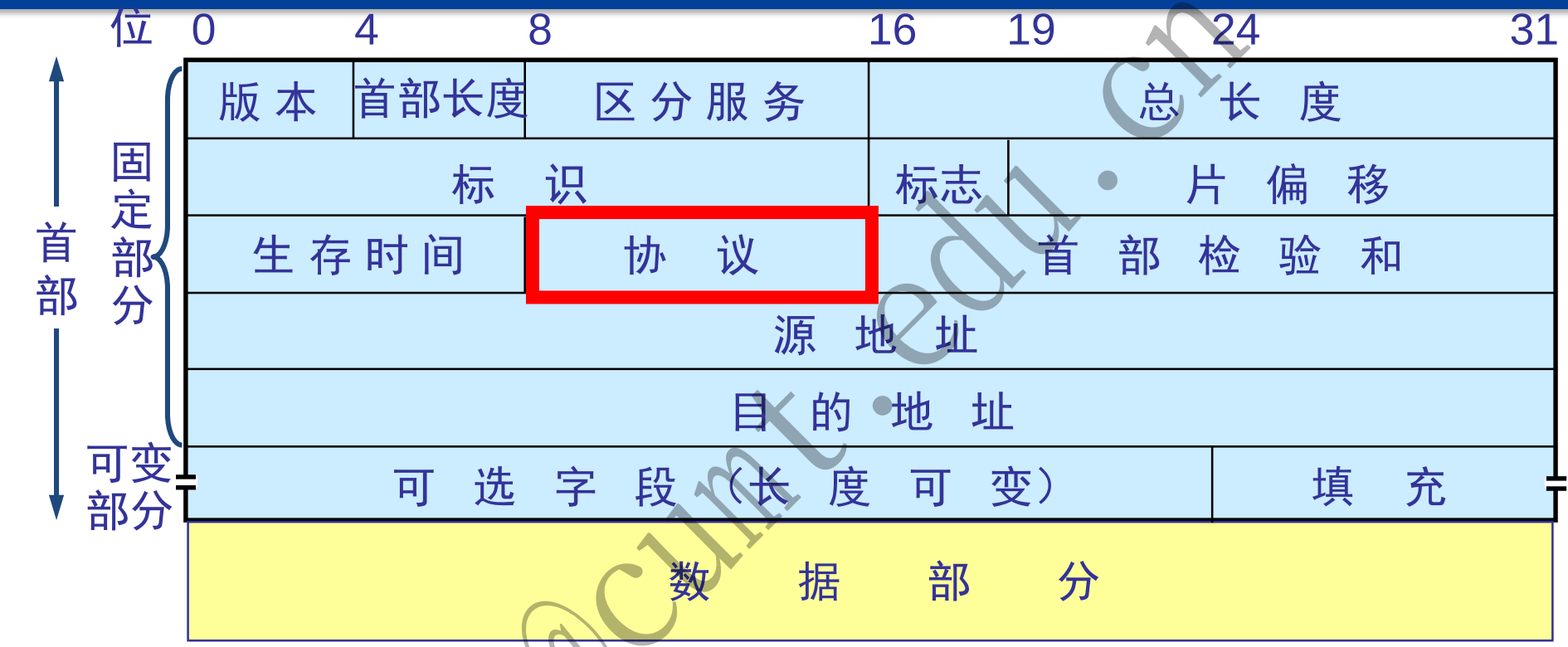
由发出数据报的源点设置，目的是防止无法交付的数据报无限制地在互联网中兜圈子，因而白白浪费网络资源。

TTL:Time To Live

最初的设计是以秒为单位，随着路由器处理数据报所需时间不断缩小，一般都远远小于1秒，所以功能改为“跳数限制”，数据报在网络中可通过的路由器数的最大值。

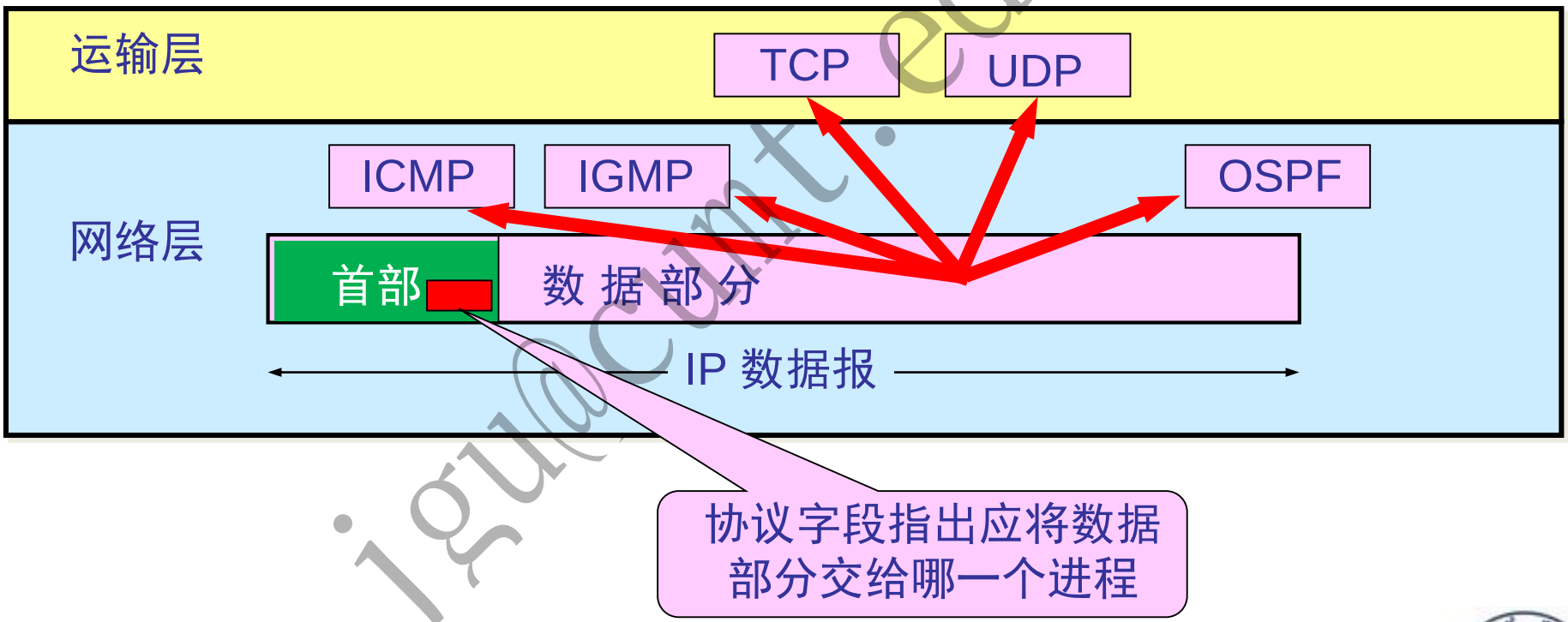
TTL最大为255

若TTL值减小到0，就丢弃这个数据报，不再转发



协议(8 位)字段指出此数据报携带的数据使用何种协议以便目的主机的 IP 层将数据部分上交给哪个处理过程

IP 协议支持多种协议，
IP 数据报可以封装多种协议 PDU。



Protocol	Keyword	Protocol
0		Reserved
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
3	GGP	Gateway-to-Gateway
4	IP	IP in IP (encasulation)
5	ST	Stream
6	TCP	Transmission Control
8	EGP	Exterior Gateway Protocol
9	IGP	any private interior gateway
17	UDP	User Datagram
29	ISO-TP4	ISO Transport Protocol Class 4
41	SIP	Simple Internet Protocol
55-60		Unassigned
80	ISO-IP	ISO Internet Protocol
92	MTP	Multicast Transport Protocol
101-254		Unassigned
255		Reserved

首部检验和(16 位)字段只检验数据报的首部

不检验数据部分。

这里不采用 CRC 检验码而采用简单的计算方法。

IP/ICMP/IGMP/TCP/UDP等协议的校验和算法都是相同的

- **在发送数据时，计算IP数据包校验和的步骤：**

（1）把IP数据包的校验和字段置为0；

（2）把首部看成以16位为单位的数字组成，依次进行二进制反码求和；

（3）把得到的结果存入校验和字段中。

- **在接收数据时，计算数据包的校验和步骤：**

（1）把首部看成以16位为单位的数字组成，依次进行二进制反码求和，包括校验和字段；

（2）检查计算出的校验和的结果是否等于零（反码应为16个0）；

（3）如果等于零，说明被整除，校验是和正确。否则，校验和就是错误的，协议栈要抛弃这个数据包。

发送端

由低向高位逐列计算

0+0=0，0+1=1

1+1=0 并向下一列进位

最高位的进位加到结果中

接收端

数据报首部

字 1 16 位

字 2 16 位

…

检验和 置为全 0

…

字 n 16 位

反码算术运算求和 16 位

取反码

检验和 16 位

IP 数据报

数据部分

数据部分

不参与检验和的计算

字 1 16 位

字 2 16 位

…

检验和 16 位

…

字 n 16 位

反码算术运算求和 16 位

取反码

结果 16 位

若结果为 0, 则保留；否则，丢弃该数据报

IP首部检验和的计算

（1）把IP数据包的校验和字段置为0；

（2）把首部看成以16位为单位的数字组成，依次进行二进制求和（注意：求和时应将最高位的进位保存，所以加法应采用32位加法）；

（3）将上述加法过程中产生的进位（最高位的进位）加到低16位（采用32位加法时，即为将高16位与低16位相加，之后还要把该次加法最高位产生的进位加到低16位）

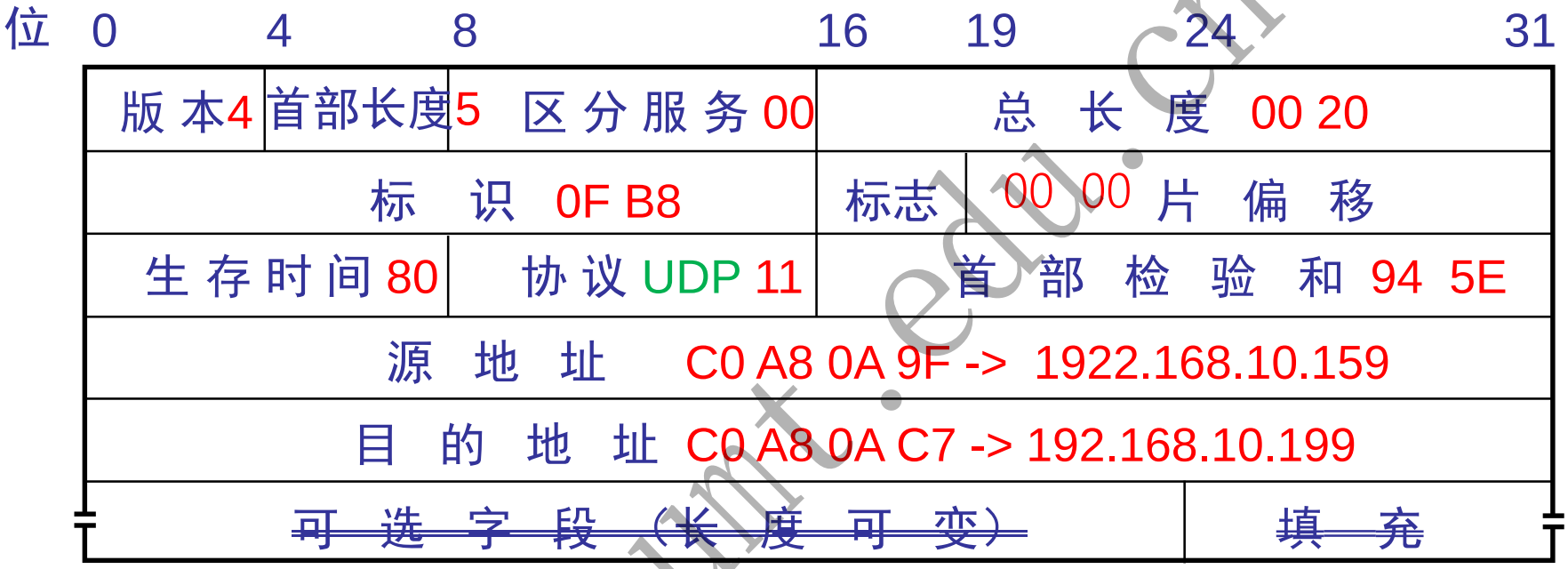
（4）将上述的和取反，即得到校验和。

发送端IP首部检验和计算（十六进制表示）

位 0	4	8	16	19	24	31
版 本4	首部长度5	区 分 服 务 00	总 长 度 00 20			
标 识 0F B8			标志	00 00 片 偏 移		
生 存 时 间 80		协 议 UDP 11	首 部 检 验 和 00 00			
源 地 址 C0 A8 0A 9F -> 1922.168.10.159						
目 的 地 址 C0 A8 0A C7 -> 192.168.10.199						
~~可 选 字 段（长 度 可 变）~~					~~填 充~~	

- 0x4500 + 0x0020 + 0x0FB8 + 0x0000 + 0x8011 + 0x0000 + 0xC0A8 + 0x0A9F + 0xC0A8 + 0x0AC7
- = 0x26B9F
- 0x0002 + 0x6B9F = 0x6BA1
- 将 0x6BA1 取反得 0x945E，即为填入的首部校验和

接收端IP首部检验和计算（十六进制表示）



- 0x4500 + 0x0020 + 0x0FB8 + 0x0000 + 0x8011 + 0x945E + 0xC0A8 + 0x0A9F + 0xC0A8 + 0x0AC7
- = 0x2FFFD
- 0x0002 + 0xFFFD = 0xFFFF
- 将 0xFFFF 取反得 0x0000，即为0，则保留该IP数据报

- 源地址和目的地址都各占 4 字节

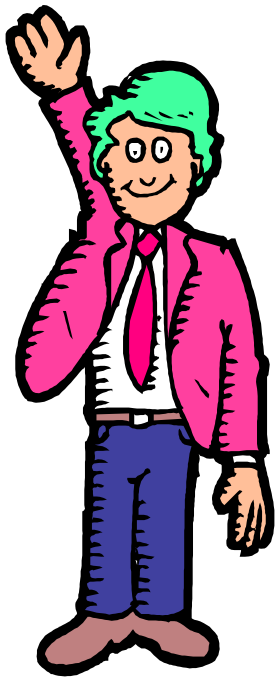
IP 数据报首部的可变部分

- IP 首部的可变部分就是一个选项字段，用来支持排错、测量以及安全等措施，内容很丰富。
- 选项字段的长度可变，从 1 个字节到 40 个字节不等，取决于所选择的项目。
- 增加首部的可变部分是为了增加 IP 数据报的功能，但这同时也使得 IP 数据报的首部长度成为可变的。这就增加了每一个路由器处理数据报的开销。

目前定义了5种选项

选 项	描 述
安全性（security）	指明数据报的机密程度（IPSec）
严格的源路由选择（strict source routing）	指定一个IP地址列表，不能和指定的路径有任何背离
松散的源路由选择（loose source routing）	指定一个IP地址列表，指定的路径可以发生变化
记录路由（record route）	路由器执行数据报路径的跟踪任务，存储动态增长的路由列表
时间戳（time stamp）	每个路由器都附上它的地址和时间标记

实际上这些选项很少被使用。



THANK YOU!