



第二十章

IP协议

20-1 网际互联

关于网际互联和网络连接实现一个互联网或者因特网的技术。

本节主题:

网络层需求

作为数据报网络的因特网

作为无连接网络的因特网

Figure 20.1 两台主机之间的链路

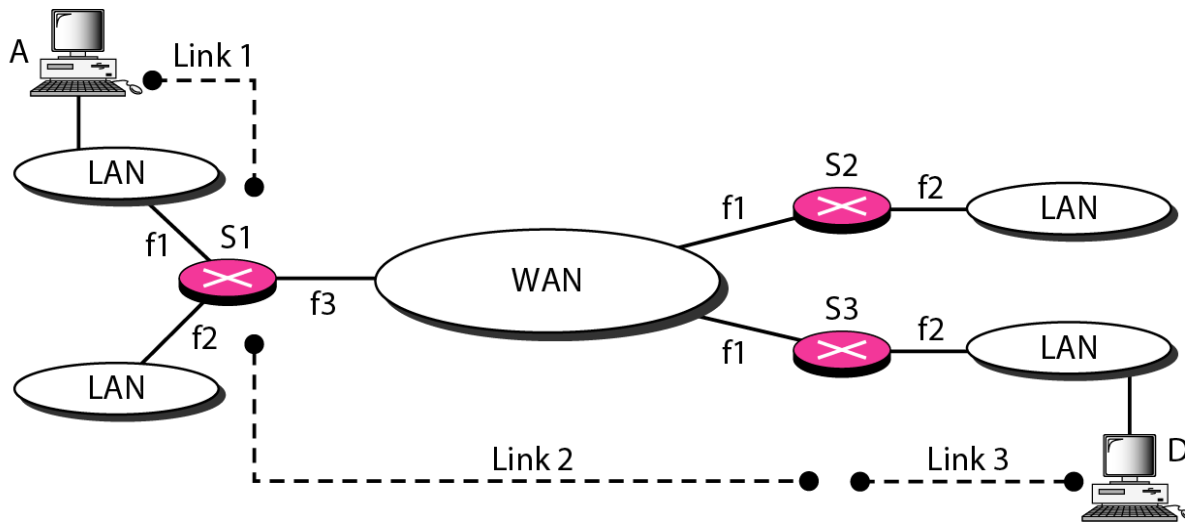


Figure 20.2 互联网中的网络层

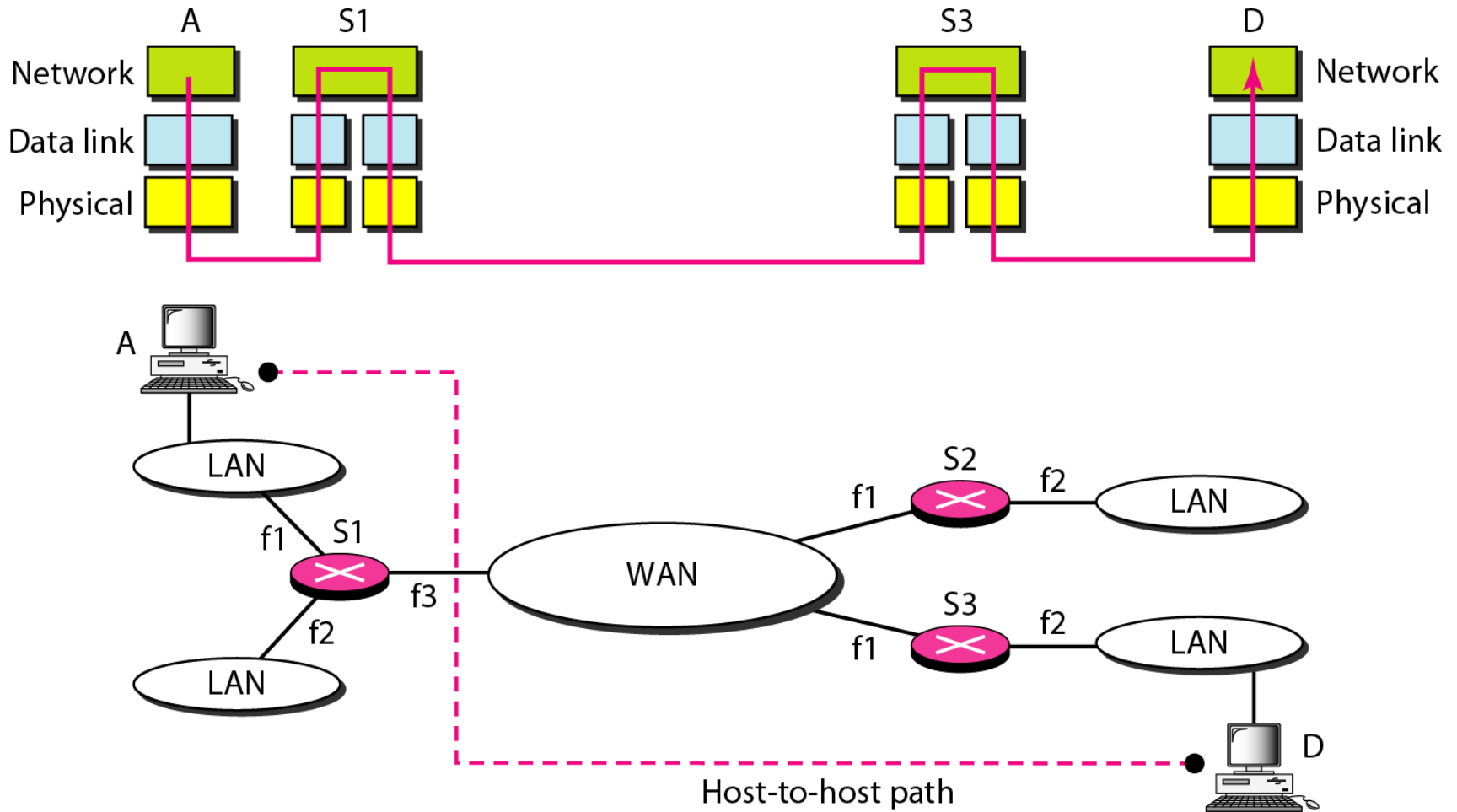
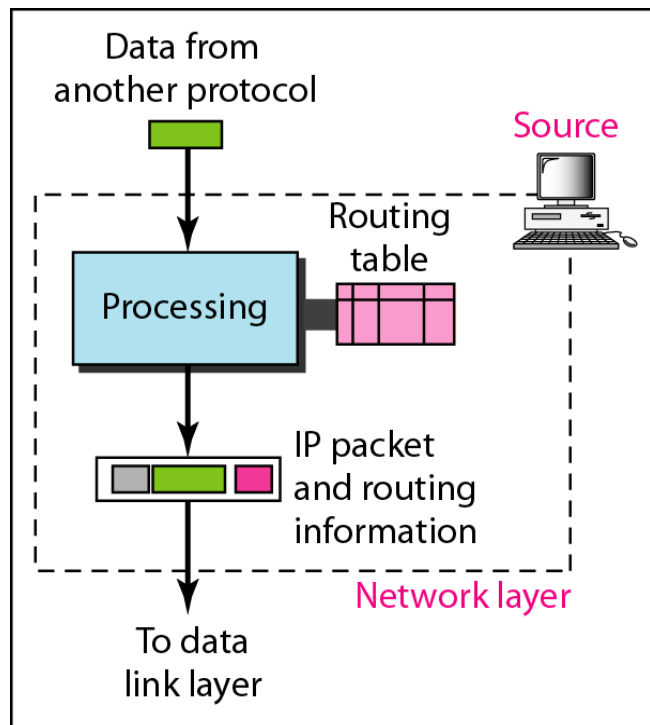
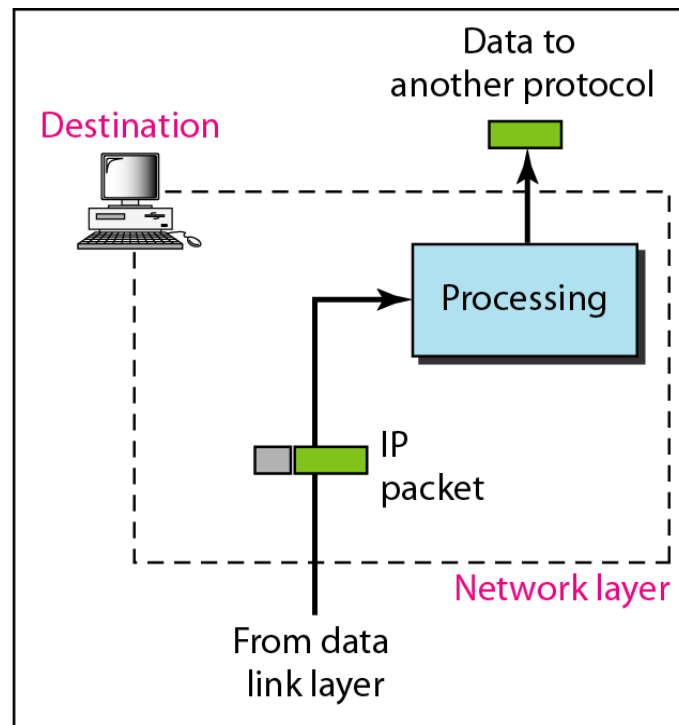


Figure 20.3 源端、路由器端和目的端的网络层



a. Network layer at source

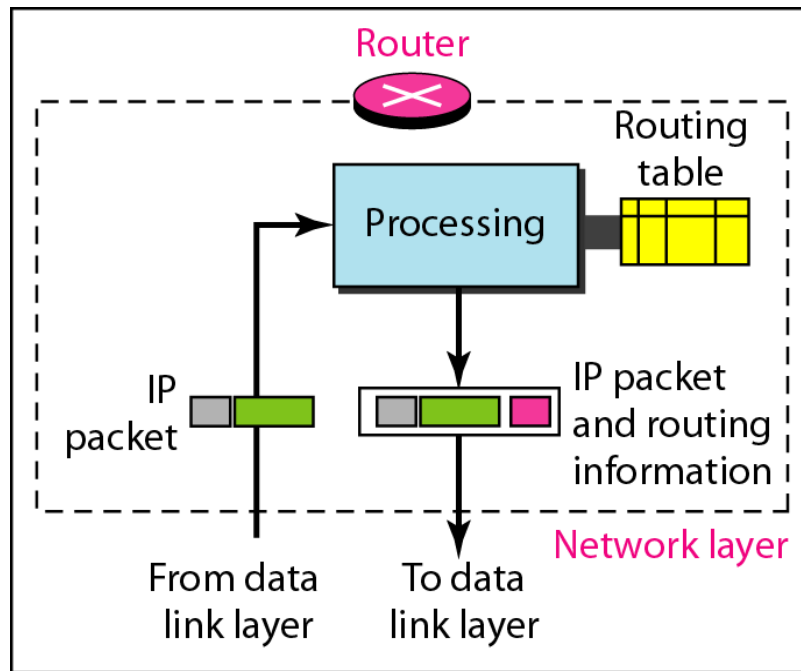
检验路由表寻找路由选择信息；分段



b. Network layer at destination

地址验证；重组

Figure 20.3 源端、路由器端和目的端的网络层



c. Network layer at a router

进行路由选择

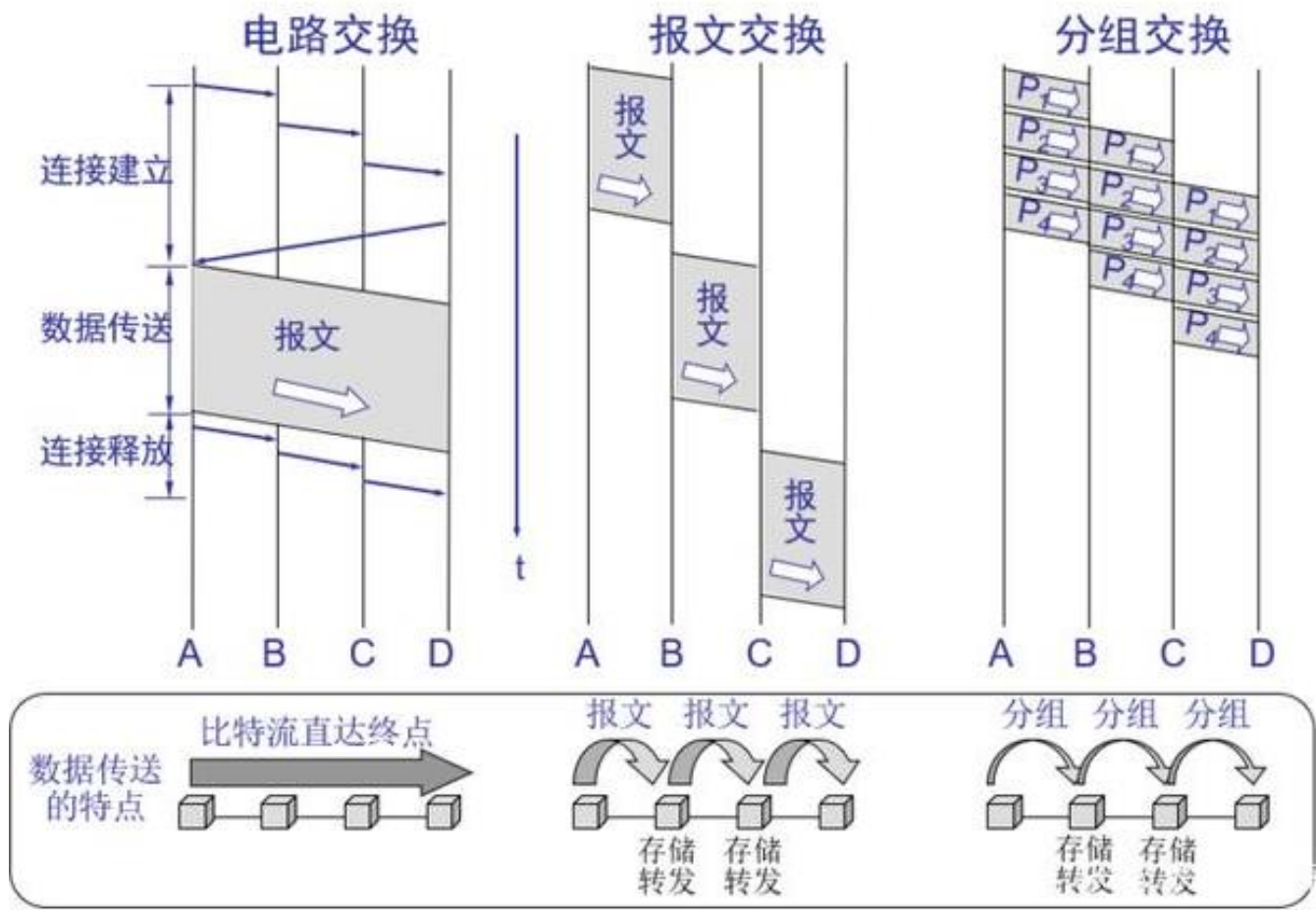


注意

- 因特网中的网络层交换是利用数据报分组交换的方法实现的。
- 分组传递可利用面向连接的服务来实现，也可以利用无连接的服务来实现。
- 因特网的网络层通信是无连接的。

注：无连接是指交换机或路由器不保存有关连接状态的信息，不需要建立连接，也不需要拆除连接。

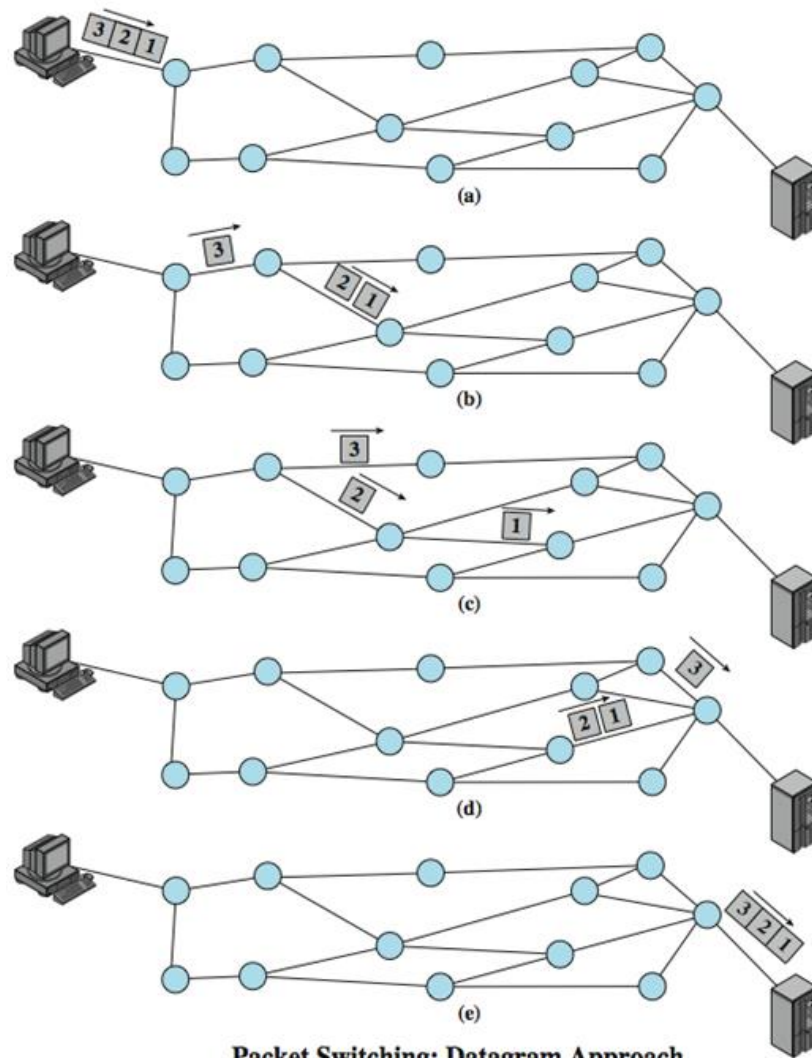
三种交换模式



三种交换模式比较

- ◆ 电路交换：两个结点间必须建立一条专用的物理通信路径，分为建立连接、数据传输、连接释放三个阶段。在数据传输过程中，用户始终占用端到端的固定传输带宽。
 - 优点：时延小，有序传输，无冲突，适用范围广，实时性强，控制简单。
 - 缺点：连接时间长，线路独占，灵活性差，难以规格化。
- ◆ 报文交换：单位是报文，携带有目的地址、源地址等信息。
 - 优点：无需建立连接，动态分配线路，提高线路可靠性，提高线路利用率，提供多目标服务。
 - 缺点：需要存储转发，转发时延大；报文没有大小限制，要求结点有较大的缓存空间，主要用于早期的电报通信网。
- ◆ 分组交换：采用存储转发方式，把大数据划分为合理的小数据块，再加上源地址、目的地址、编号信息等控制信息构成分组。
 - 优点：无建立连接延时，线路利用率高，简化存储管理，加速传输，减少出错率和重发数据量。
 - 缺点：存在传输延时，需要传输额外的信息量，可能出现失序、丢失或重复分组。

数据报分组交换



Packet Switching: Datagram Approach

20-2 IPv4

网际协议第四版 (**IPv4**) 是TCP/IP协议使用的传输机制。
IPv4是一种不可靠的无连接的数据报协议，它尽力传递
(**Best-effort delivery**) 数据报，但是不提供差错控制或
流量控制（除首部的检验和之外），因此不保证可靠性。

本节主题:

数据报
分段
校验和
选项

Figure 20.4 IPv4在TCP/IP协议族中的位置

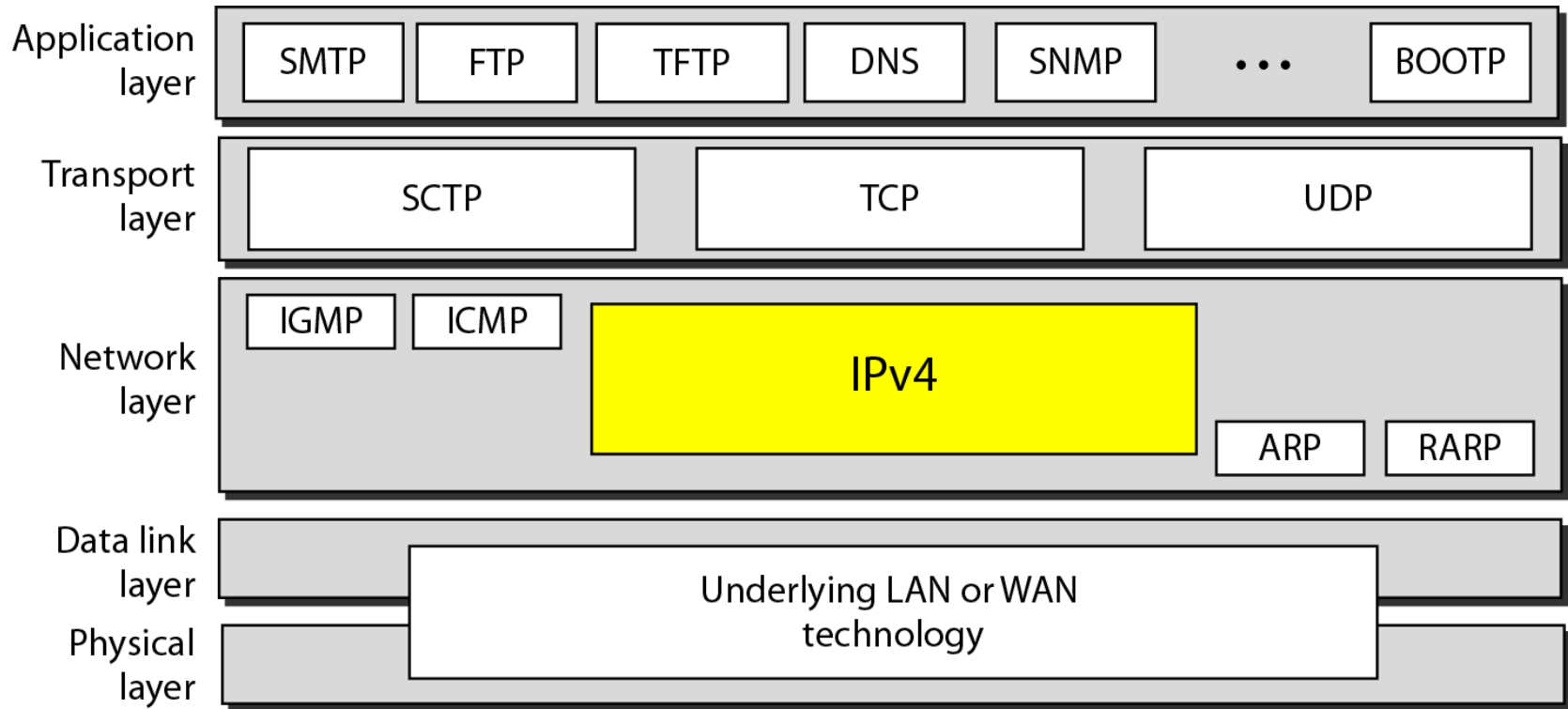
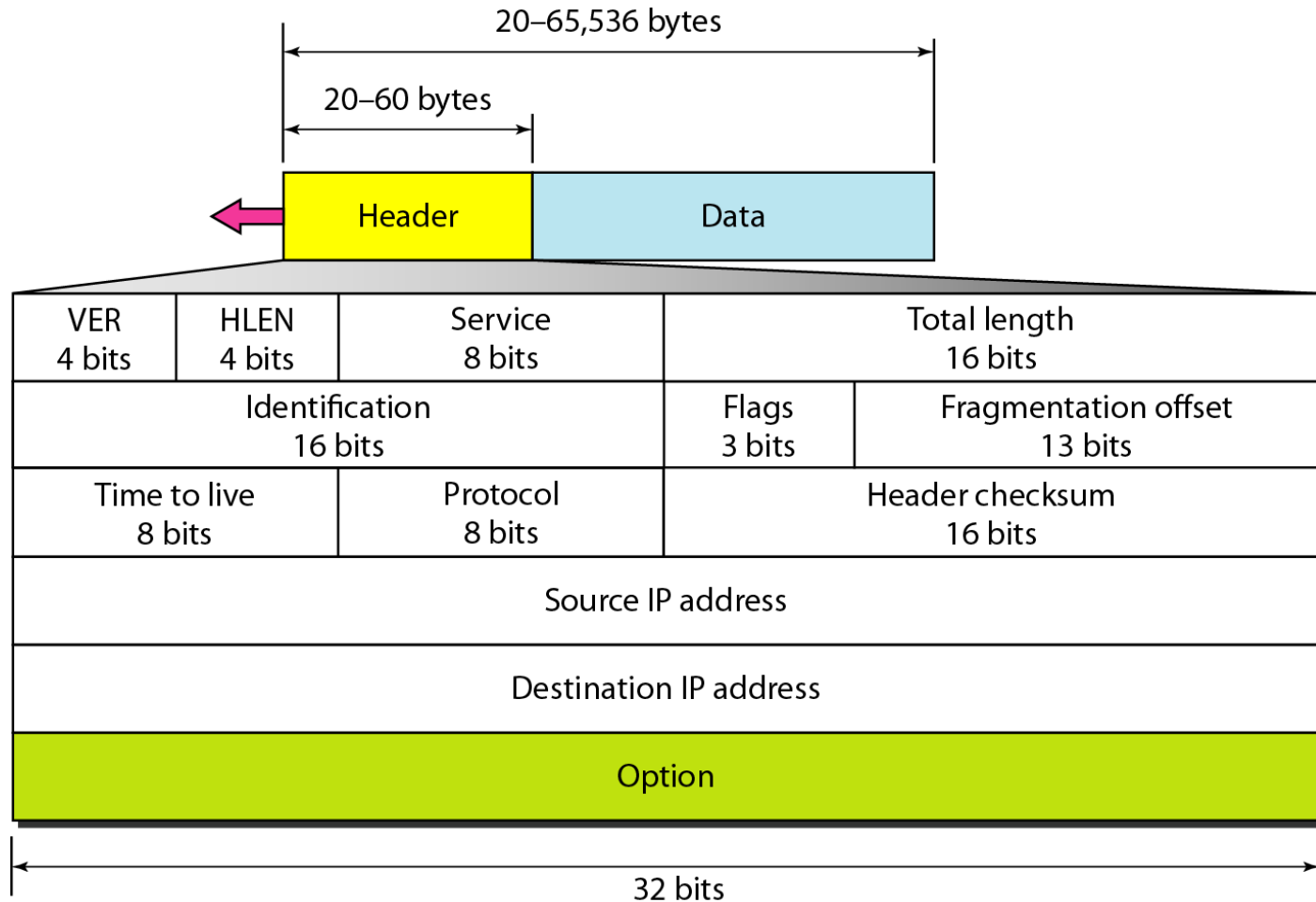


Figure 20.5 IPv4 数据报格式





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

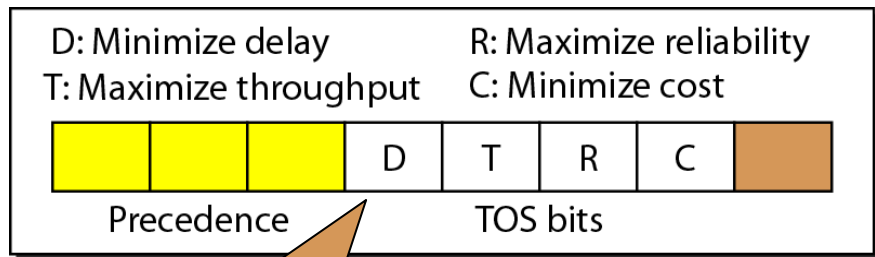


首部长度——占 4 bit，可表示的最大数值是 15 个单位（一个单位为 4 字节），因此 IP 的首部长度的最大值是 60 字节。



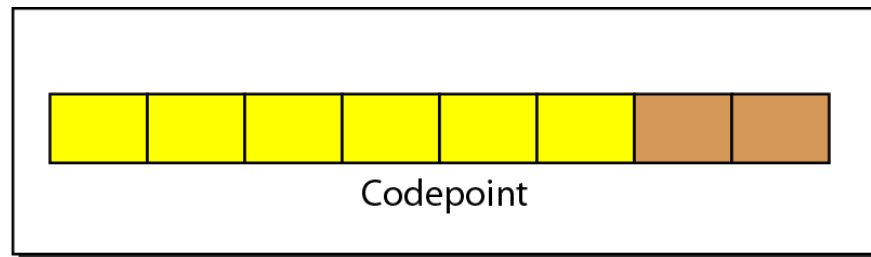
服务类型——占 8 bit，用来获得更好的服务。这个字段以前一直没有使用。

Figure 20.6 服务类型或差分服务



Service type

TOS(Type of Service)



Differentiated services

差分服务

在版本4中未使用优先子字段。

<i>TOS Bits</i>	<i>Description</i>
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Table 20.2 服务类型的默认值

<i>Protocol</i>	<i>TOS Bits</i>	<i>Description</i>
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

差分服务

- 前6位为码点子字段，后2位不用。
- 当码点子字段最右边3位都是0时，最左边3位与服务类型中的优先级相同。
- 当码点子字段最右边3位不全是0时，则6位的含义如下图。
 - 0,2,4,, 62, 由IETF分配
 - 3,7,11,15,, 63, 本地组织结构时候用
 - 1,5,9,, 61, 临时的用作实验目的

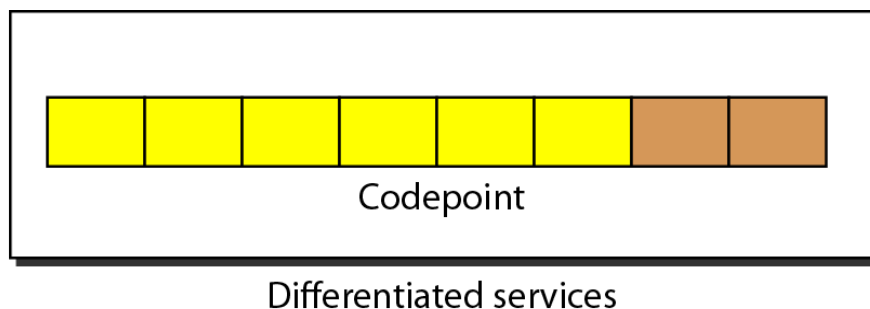


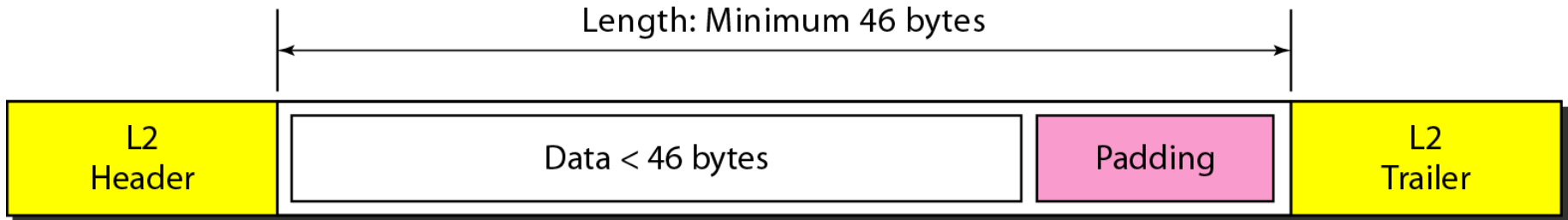
Table 20.3 *Values for codepoints*

<i>Category</i>	<i>Codepoint</i>	<i>Assigning Authority</i>
1	XXXXXO	Internet
2	XXXXXI	Local
3	XXXXOI	Temporary or experimental



总长度——占 16 bit，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。总长度必须不超过最大传送单元 MTU。

Figure 20.7 一个小的数据报封装在以太网帧中



以太网限制帧中数据报的长度在46到1500字节之间，当数据报长度小于46字节时，必须进行填充。



标识 (identification) 占 16 bit, 它是一个计数器, 用来产生数据报的标识。



标志 (flag) 占 3 bit, 只有后两位有意义。

- **More Fragment:** MF=1 (还有分片), MF=0 (最后一片)
- **Don't Fragment:** DF=1 (不能分片), DF=0 (允许分片)



片偏移占13 bit，用于指出较长的分组在分片后某片在原分组中的相对位置，以 8 个字节为偏移单位。



生存时间（8 bit）记为 TTL（Time To Live），表示数据报在网络中可通过的路由器数的最大值。



协议（8 bit）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那个处理过程。

Figure 20.8 协议字段和封装的数据

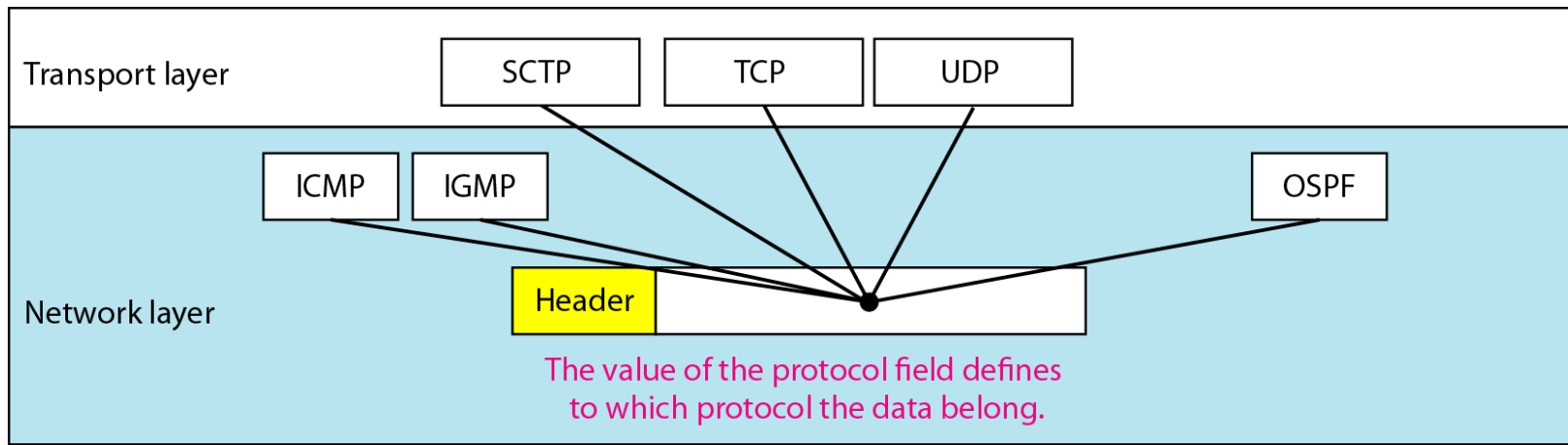


Table 20.4 协议值

<i>Value</i>	<i>Protocol</i>
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF



首部检验和(16 bit)字段只检验数据报的首部，不包括数据部分。这里不采用 CRC 检验码而采用简单的计算方法。

发送端

接收端

数据报首部



算术运算求和

16 bit

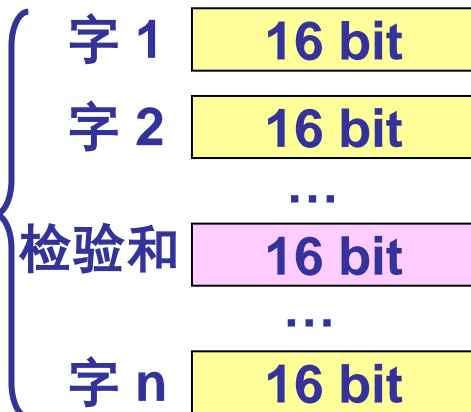
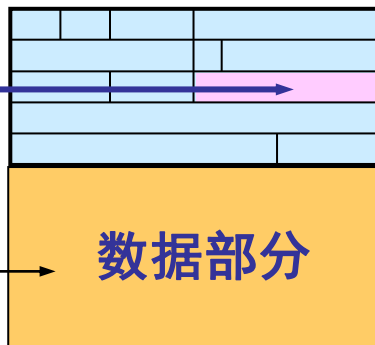
取反码

检验和

16 bit

数据部分
不参与检验和的计算

IP 数据报



算术运算求和

16 bit

取反码

结果

16 bit

若结果为 0, 则保留;
否则, 丢弃该数据报

Figure 20.13 IPv4中校验和的计算

4	5	0	28	
1			0	0
4	17	0		
10.12.14.5				
12.6.7.9				

4, 5, and 0

28

1

0 and 0

4 and 17

0

10.12

14.5

12.6

7.9

Sum

Checksum

4	5	0	0
0	0	1	C
0	0	0	1
0	0	0	0
0	4	1	1
0	0	0	0
0	A	0	C
0	E	0	5
0	C	0	6
0	7	0	9
<hr/>			
7	4	4	E
8	B	B	1



源地址和目的地址都各占 4 字节。

Example 20.1

一个到达的 IPv4 分组的前8位如下：

01000010

接收方是否应丢弃该分组？为什么？

解：

这个分组有错误，其中最左的4位是版本，它是正确的。下一个4位是0010，则表明它是一个无效长度 ($2 \times 4 = 8$)。而头部的最小字节数是20。因此，这个分组在传输过程中被损坏了。

Example 20.2

在一个 IPv4 分组中，头部长度的值用二进制表示为1000，试问这个分组携带的选项是几个字节？

解：

头部的长度值是 8，说明头部的总字节数是 $8 \times 4 = 32$ 字节。前面的 20 个字节是基本头部，后面的 12 个字节是选项。

Example 20.3

在一个 IPv4 分组中，头部长度的值是5，而总长度字段的值是 0x0028，试问这个分组携带的数据是多少字节？

解：

头部长度的值是 5，就是说头部的总字节数为 $5 \times 4 = 20$ 字节 (无选项)。总长度是40字节，即这个分组携带 $(40 - 20) = 20$ 个字节的数据。

Example 20.4

一个IPv4 分组已到达，最前面几个十六进制数字如下

0x45000028000100000102...

在丢弃这分组之前，它还能跳多少跳？数据是属于上层的哪一个协议？

解：

生存时间字段是第9个字节，跳过了前8个字节（16个十六进制数字），得到值是01，这就是说分组仅能跳一次。协议字段是下一个字节02，也就是说上层协议是IGMP。

通过抓包查看IPv4首部

The image shows a Wireshark packet capture window titled "Realtek PCIe GBE Family Controller - Wireshark". The packet list on the left shows several packets, with packet 82 selected. The packet details pane on the right shows the structure of packet 82, which is an IPv4 packet containing a TCP segment. The packet bytes pane at the bottom shows the raw data of the packet.

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
70	6.604263	168.168.214.20	113.96.233.139	TLSv1.2	Application Data
71	6.657756	113.96.233.139	168.168.214.20	TCP	https > 57659 [ACK] Seq=2446 Ack=3978 win=23936 Len=0
72	6.661806	113.96.233.139	168.168.214.20	TCP	https > 62245 [ACK] Seq=199 Ack=1553 win=1089 Len=0
73	6.709764	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
74	6.709766	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
75	6.709767	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
76	6.709925	168.168.214.20	113.96.233.139	TCP	57659 > https [ACK] Seq=3978 Ack=6646 win=65800 Len=0
77	6.709957	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
78	6.710756	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
79	6.710757	113.96.233.139	168.168.214.20	TLSv1.2	Application Data
80	6.710894	168.168.214.20	113.96.233.139	TCP	57659 > https [ACK] Seq=3978 Ack=10831 win=65800 Len=0
81	6.799151	68:8f:84:04:a7:bd	Spanning-tree-(for-STP	MST	Root = 32768/0/68:8f:84:04:a7:bd Cost = 0 Port = 0x8003
82	6.999540	fe80::21ec:72e5:26c:ff02::c		SSDP	M-SEARCH * HTTP/1.1

Frame 80: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)

Ethernet II, Src: 18:60:24:ae:44:3e (18:60:24:ae:44:3e), Dst: 78:2c:29:f5:d0:5b (78:2c:29:f5:d0:5b)

Internet Protocol, Src: 168.168.214.20 (168.168.214.20), Dst: 113.96.233.139 (113.96.233.139)

- Version: 4
- Header length: 20 bytes
- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
- Total length: 40
- Identification: 0x470e (18190)
- Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 64
- Protocol: TCP (6)
- Header checksum: 0x0000 [incorrect, should be 0x1a19]
- Source: 168.168.214.20 (168.168.214.20)
- Destination: 113.96.233.139 (113.96.233.139)

Transmission Control Protocol, Src Port: 57659 (57659), Dst Port: https (443), Seq: 3978, Ack: 10831, Len: 0

0000 78 2c 29 f5 d0 5b 18 60 24 ae 44 3e 08 00 45 00 x,)...[.~\$.D>..E.
0010 00 28 47 0e 40 00 40 06 00 00 a8 a8 d6 14 71 60 .(G.@.@.q
0020 e9 8b e1 3b 01 bb a6 37 0b 67 dc b3 44 5c 50 107.g..D\P.
0030 40 42 d9 c3 00 00 @B....

Frame (frame), 54 bytes Packets: 150 Displayed: 150 Marked: 0 Dropped: 0 Profile: Default

Figure 20.9 最大传输单元 (MTU)

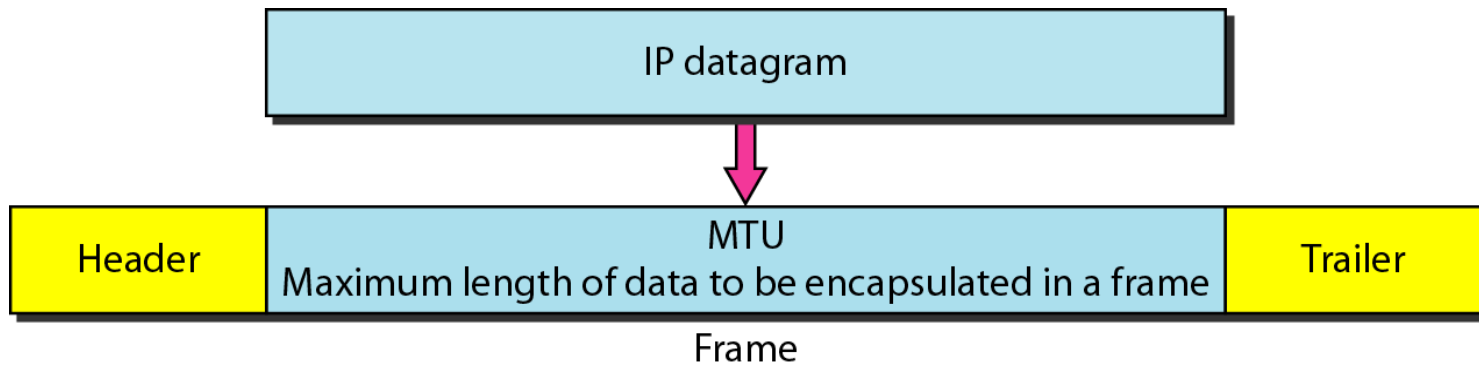
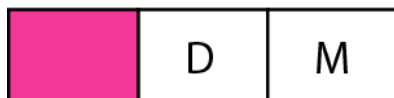


Table 20.5 某些网络的 MTU值

<i>Protocol</i>	<i>MTU</i>
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

与分段相关的字段



D: Do not fragment
M: More fragments

- **D:** **DF=1** (不能分片)
DF=0 (允许分片)
- **M:** **MF=1** (还有分片)
MF=0 (最后一片)

Figure 20.10 标记字段

分段偏移：13位，表示这个分段在整个数据报中的相对位置，以8字节为度量单位。

Figure 20.11 分段示例

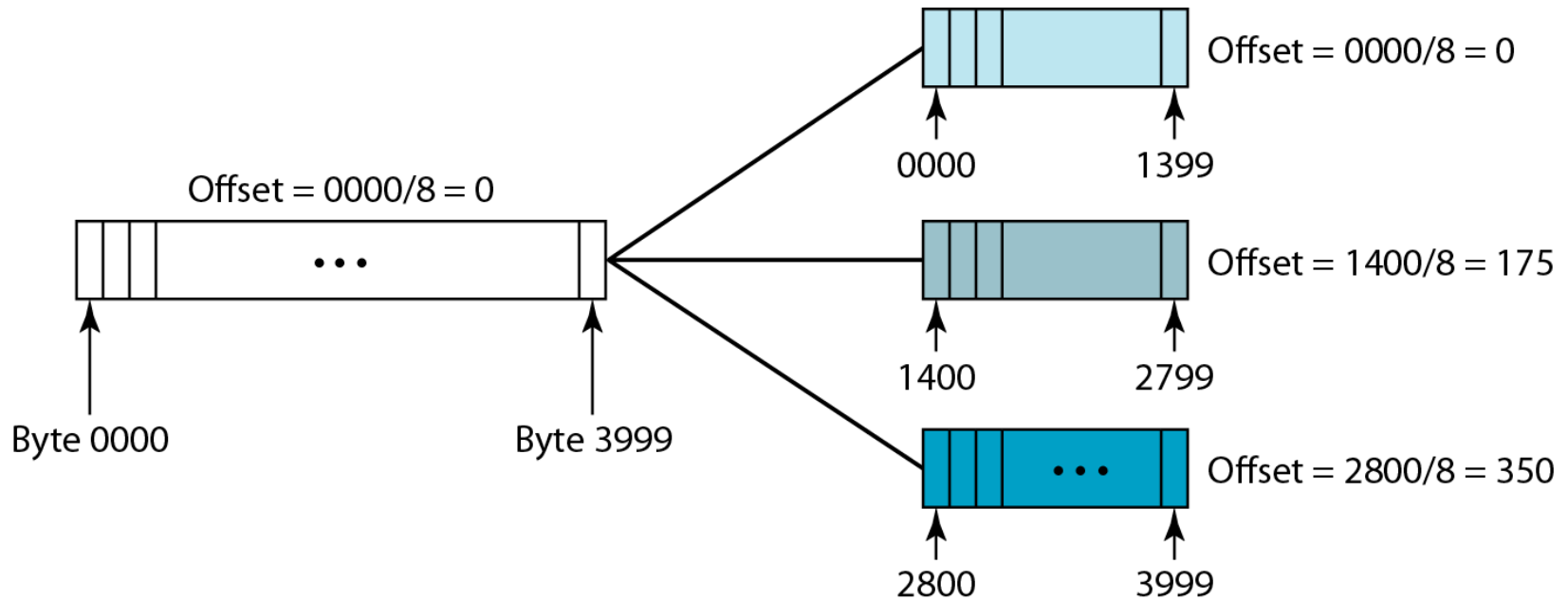
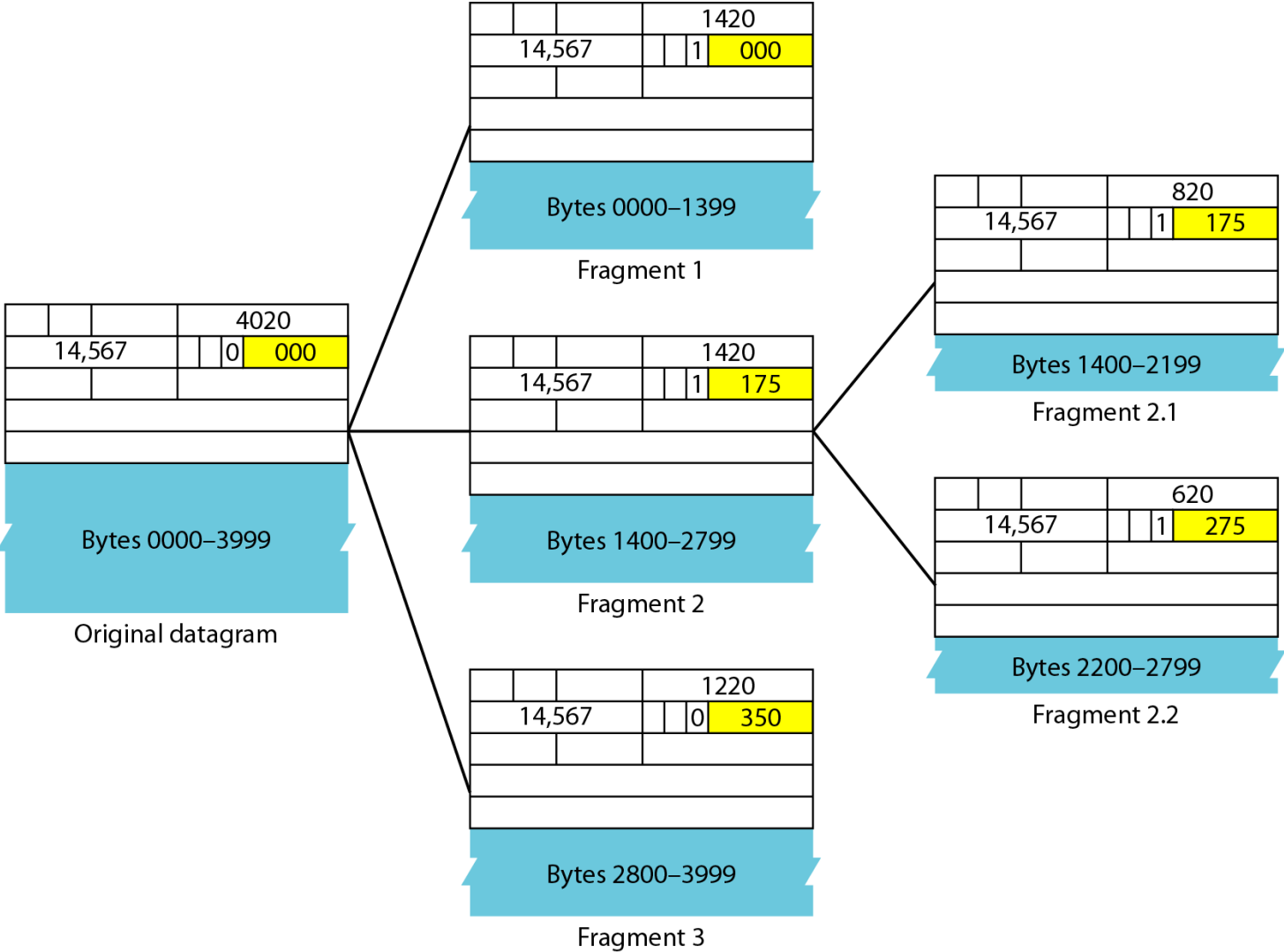


Figure 20.12 分段的细节示例





Example 20.5

到达到一个分组的M位的值是0，试问这是第一个分段还是中间的分段，还是最后的分段？我们是否知道这个分组已被分段？

解：

如果M位是0，这就是说不存在更多的分段，该分段是最后的一个分段。但是我们不能说原来的分组是否已经被分段。没有分段的分组被认为是最后一个分段。



Example 20.6

到达的一个分组的M位的值是1，试问这是第一个分段还是中间的分段，还是最后的分段？我们是否知道这个分组已被分段？

解：

如果M位是1，这就是说至少还有一个分段，这个分段是第一个或中间的分段，而不是最后的分段。但我们不知道它是第一个分段还是中间分段，还需要有更多的信息（分段偏移值）。



Example 20.7

到达的一个分组的M位的值是1，而偏移值是0，试问这是第一个分段还是最后的分段，或是最后的分段？

解：

因为M位是1，它或是第一个分段或是中间分段。由于偏移值为0，因此它是第一个分段。



Example 20.8

到达的一个分组的偏移值是100，试问第一个字节的编号是什么？能知道最后一个分段的编号吗？

解：

为了求第一个字节的编号，将偏移值乘8，即第一个字节的编号是800。但不能确定最后字节的编号，除非知道数据的长度。



Example 20.9

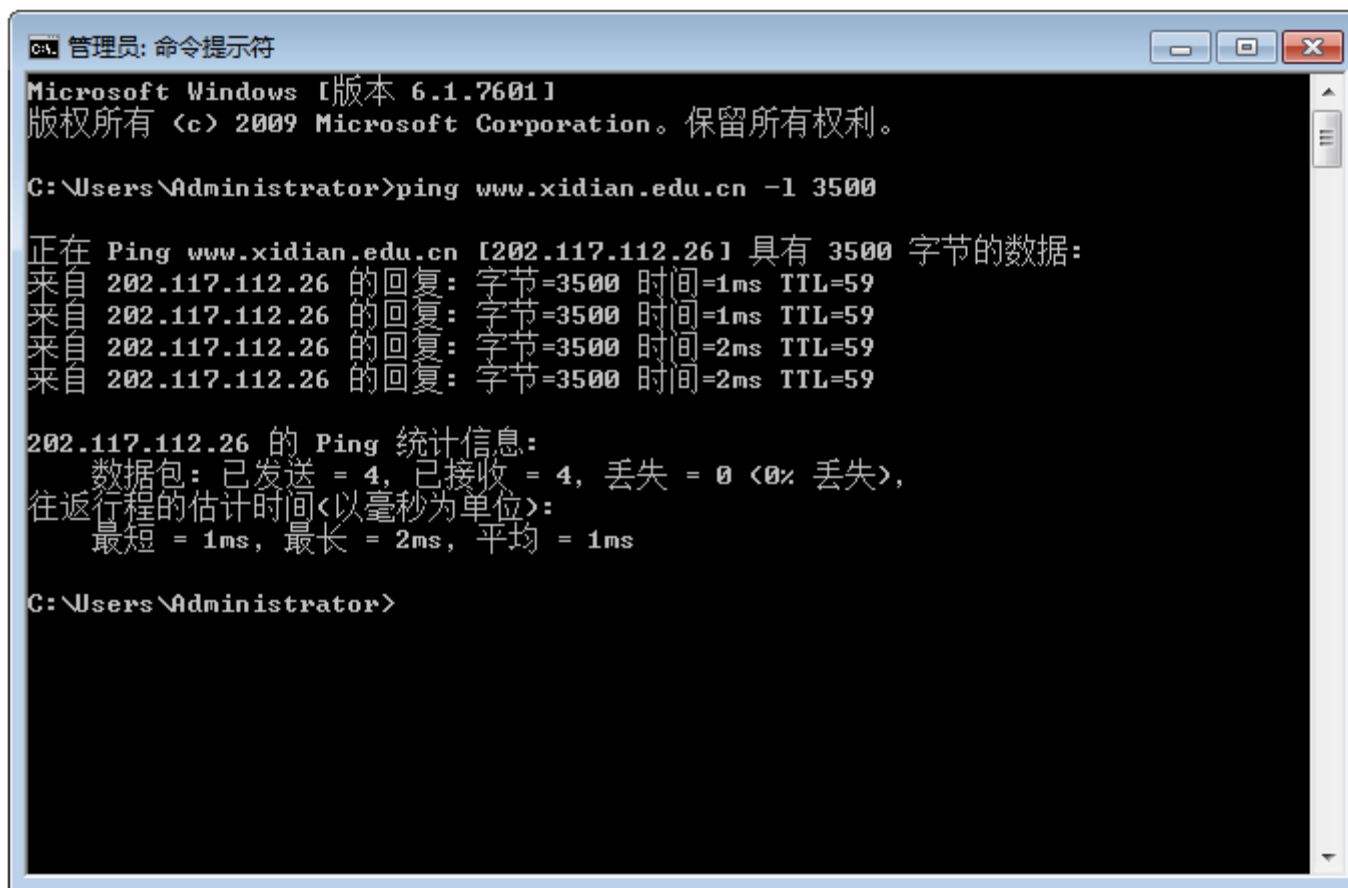
到达的一个分组的偏移值是100，而HLEN字段值为5，总长度字段的值是100。试问第一个字节和最后字节的编号是多少？

解：

第一个字节的编号是 $100 \times 8 = 800$ ，总长度是100字节，头部长度是20个字段(5×4)，这就是说这个数据报有80个字节。如果第一个字节的编号是800，则最后字节的编号是879。

捕获并观察数据报分片

指定数据报大小为3500字节，以太网MTU为1500字节，因此该数据报被分为3片。



```
管理员: 命令提示符
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping www.xidian.edu.cn -l 3500

正在 Ping www.xidian.edu.cn [202.117.112.26] 具有 3500 字节的数据:
来自 202.117.112.26 的回复: 字节=3500 时间=1ms TTL=59
来自 202.117.112.26 的回复: 字节=3500 时间=1ms TTL=59
来自 202.117.112.26 的回复: 字节=3500 时间=2ms TTL=59
来自 202.117.112.26 的回复: 字节=3500 时间=2ms TTL=59

202.117.112.26 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 2ms, 平均 = 1ms

C:\Users\Administrator>
```

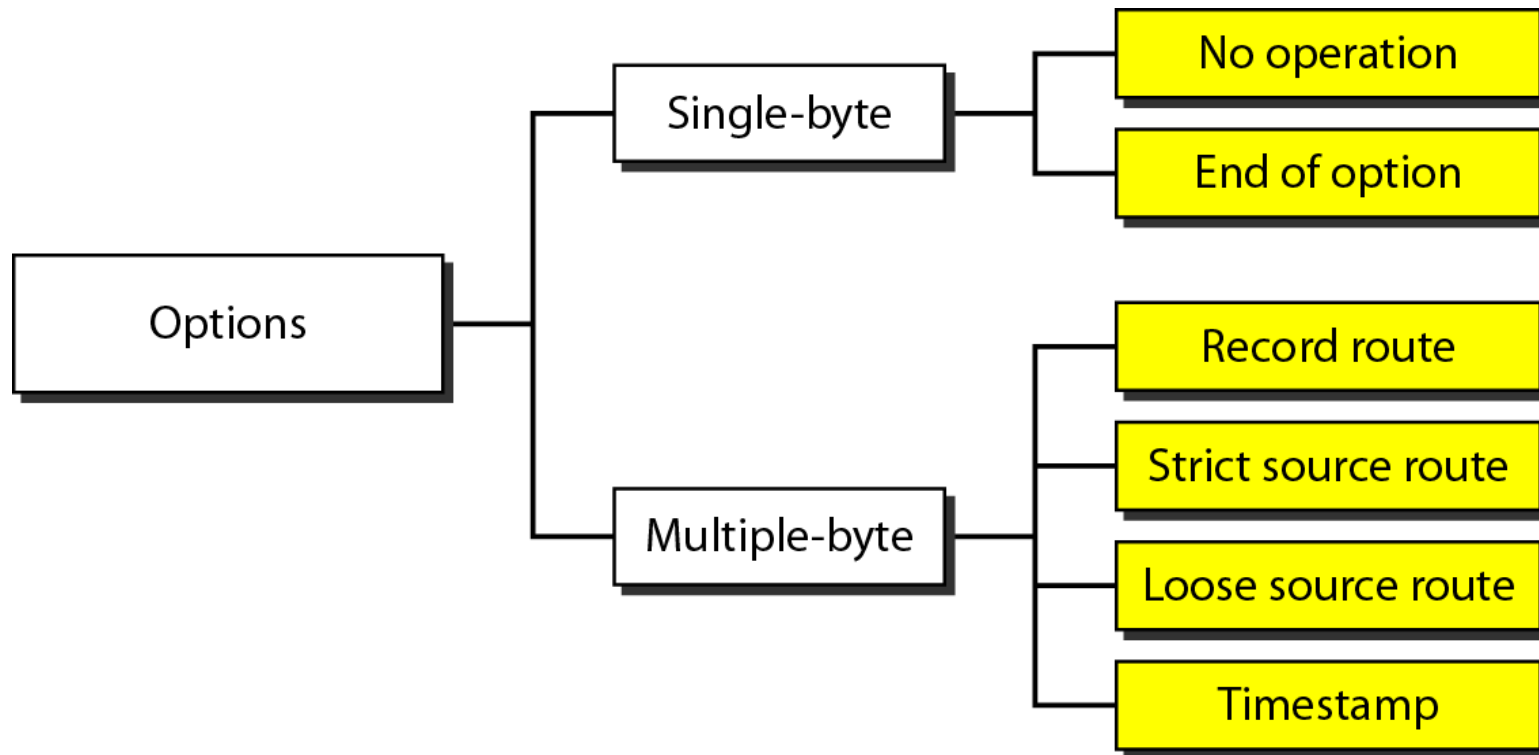

捕获并观察数据报分片

The image shows a Wireshark capture window titled "Realtek PCIe GBE Family Controller - Wireshark". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Help), a toolbar, and a filter bar. The main display area shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Info. The selected packet is #3, which is a fragmented IP protocol packet (proto=ICMP 0x01, off=1480, ID=4b97). The packet details pane shows the following information:

- Frame 3: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
- Ethernet II, Src: 18:60:24:ae:44:3e (18:60:24:ae:44:3e), Dst: 78:2c:29:f5:d0:5b (78:2c:29:f5:d0:5b)
- Internet Protocol, Src: 168.168.214.20 (168.168.214.20), Dst: 202.117.112.26 (202.117.112.26)
- Version: 4
- Header length: 20 bytes
- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
- Total Length: 1500
- Identification: 0x4b97 (19351)
- Flags: 0x01 (More Fragments)
- 0... = Reserved bit: Not set
- .0... = Don't fragment: Not set
- ..1. = More fragments: Set
- Fragment offset: 1480
- Time to live: 64
- Protocol: ICMP (1)

The packet bytes pane shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates "Frame (frame), 1514 bytes", "Packets: 119 Displayed: 119 Marked: 0 Dropped: 0", and "Profile: Default".

Figure 20.14 IPv4中选项的种类



20-3 IPv6

TCP/IP协议族中的网络层协议现在是IPv4。虽然IPv4设计得很好，但自从20世纪70年代IPv4问世以来，数据通信已经有了很大的发展。IPv4有一些缺点，这使得它对飞速发展的因特网有些不适应。

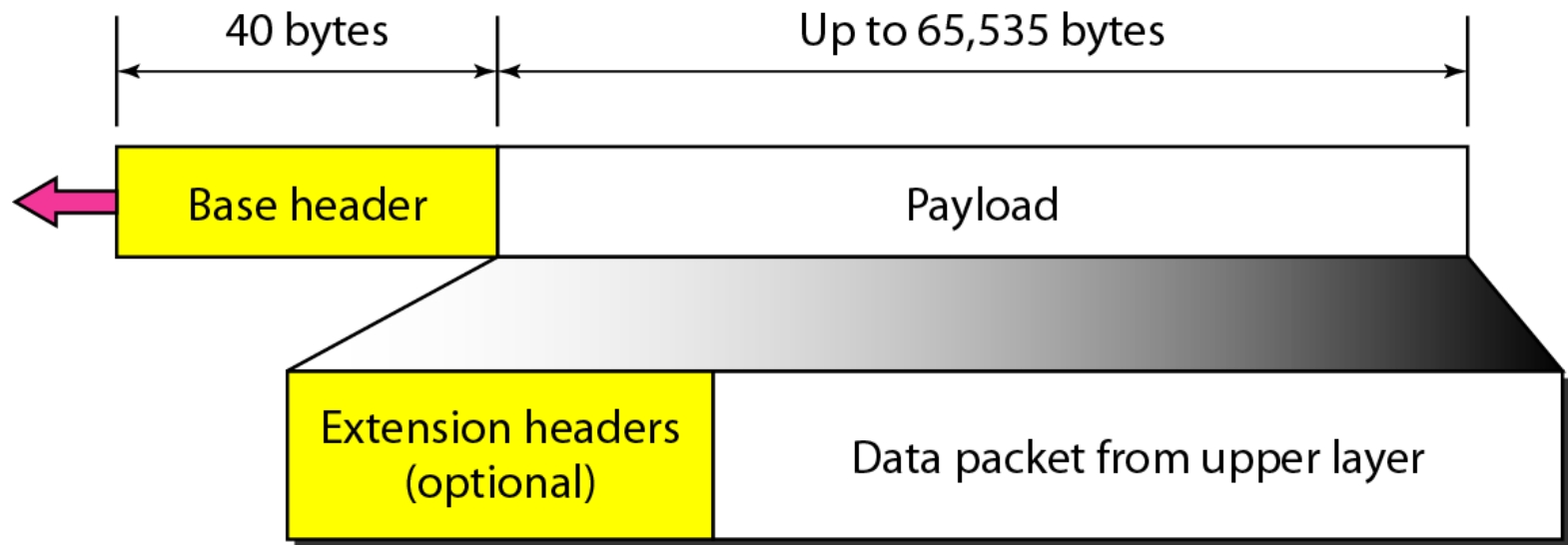
本节主题:

优点
分组格式
扩展头部

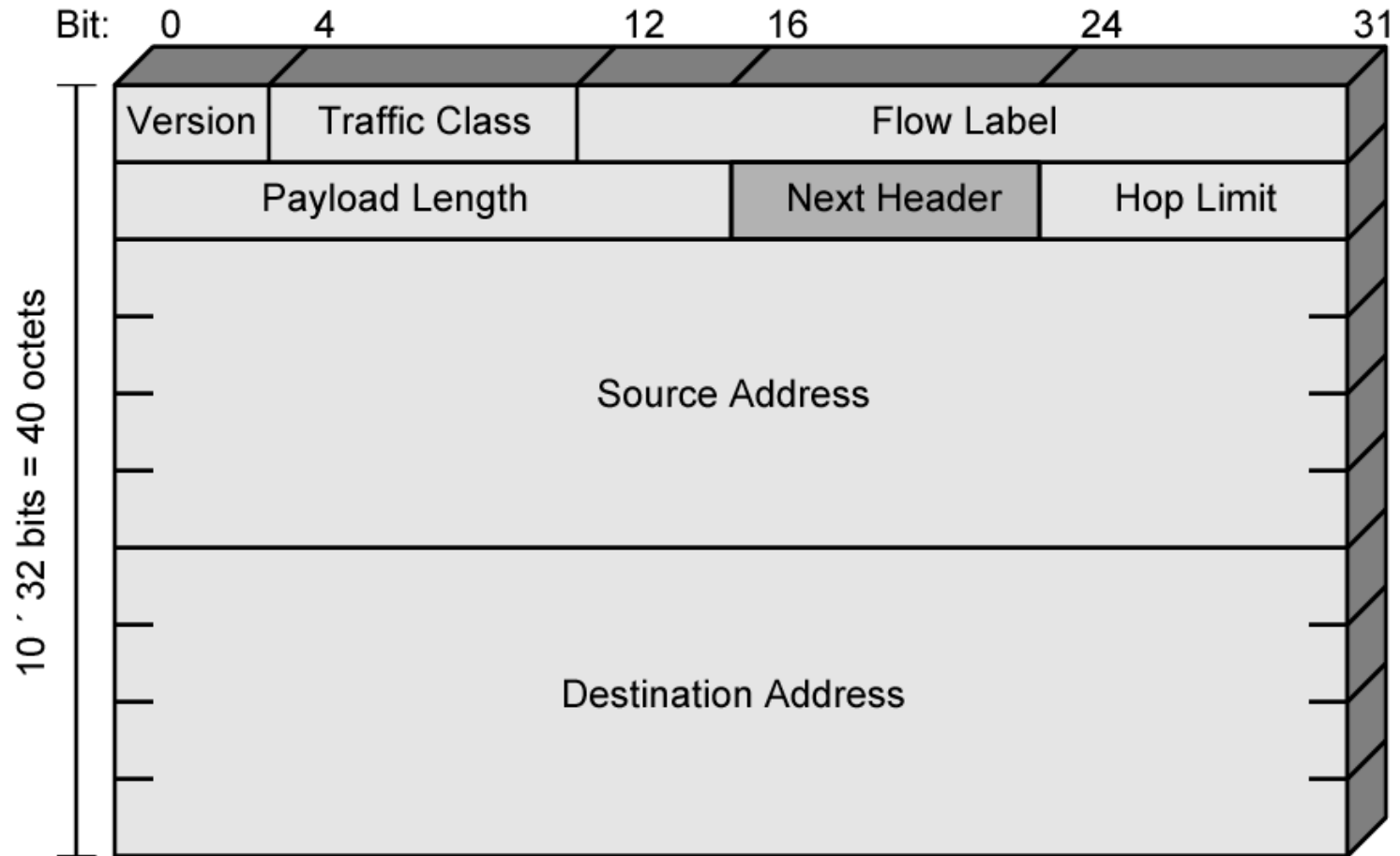
IPv6的优点

- 更大的地址空间。IPv6 将地址从 IPv4 的 32 位 增大到了 128 位。
- 扩展的地址层次结构。
- 灵活的首部格式。
- 改进的选项。
- 允许协议继续扩充。
- 支持即插即用（即自动配置）
- 支持资源的预分配。
- 支持更多的安全性。
- IPv6首部改为8字节对齐。

Figure 20.15 IPv6 数据报头部和有效载荷



IPv6 Base Header



字段说明

- 版本(version)—— 4 bit。它指明了协议的版本，对 IPv6 该字段总是 6。
- 通信量类—— 8 bit，用于区分具有不同交付要求的不同有效载荷，它取代了IPv4中的服务类型字段。
- 流标号(flow label)—— 20 bit。“流”是互联网络上从特定源点到特定终点的一系列数据报，“流”所经过的路径上的路由器都保证指明的服务质量。所有属于同一个流的数据报都具有同样的流标号。
- 有效载荷长度(payload length)—— 16 bit。它指明 IPv6 数据报除基本首部以外的字节数（所有扩展首部都算在有效载荷之内），其最大值是 65535。
- 下一个首部(next header)—— 8 bit。它相当于 IPv4 的协议字段或可选字段。
- 跳数限制(hop limit)—— 8 bit。源站在数据报发出时即设定跳数限制。路由器在转发数据报时将跳数限制字段中的值减1。当跳数限制的值为零时，就要将此数据报丢弃。

优先级

- 定义从相同源端发出的每一个分组相对于其他分组的优先级。
- IPv6通信量被分为2类：
 - 可进行拥塞控制的通信量(congestion-controlled traffic)
 - 不可进行拥塞控制的通信量(noncongestion-controlled traffic)

Table 20.7 可进行拥塞控制的通信量的优先级 **Table 20.8** 不可进行拥塞控制的通信量的优先级

<i>Priority</i>	<i>Meaning</i>
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

<i>Priority</i>	<i>Meaning</i>
8	Data with greatest redundancy
...	...
15	Data with least redundancy

流标号

- 支持流标号处理的路由器中包含有一个流标号表；
- 可用来加速路由器对分组的处理；
- 可用来支持实时音频和视频的传输；
- 流标号使用的三个原则：
 - 流标号由源主机指定给分组，是1到 $2^{24} - 1$ 之间的随机数。
 - 如果主机不支持流标号，则置为0
 - 所有属于同一个流的分组必须具有相同的源地址、目的地址、优先级和选项。

Figure 20.16 IPv6数据报的格式

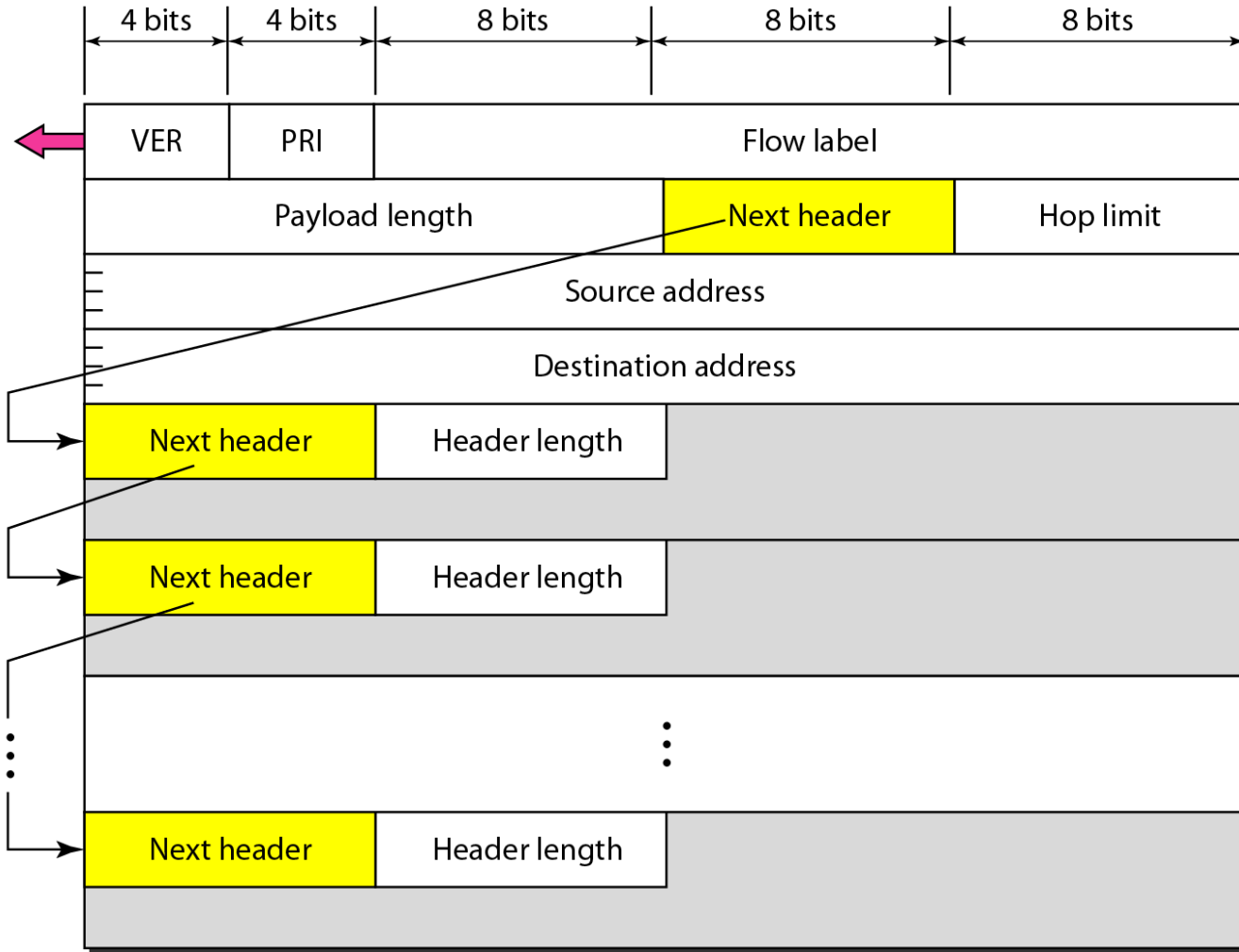


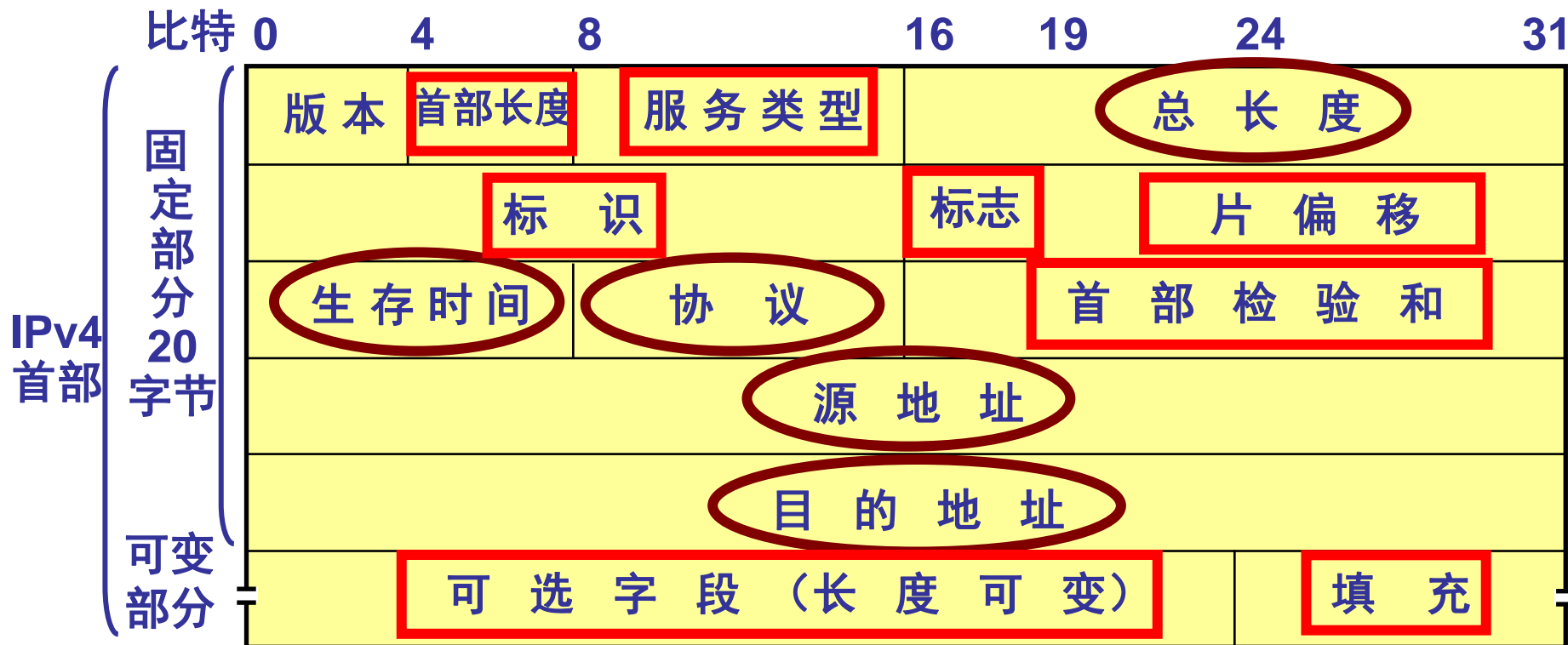
Table 20.6 IPv6的下一个首部的代码

<i>Code</i>	<i>Next Header</i>
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

IPv6 数据报首部与 IPv4 数据报首部的对比

有变化

取消



上面是 IPv4 数据报的首部

Table 20.9 IPv4和IPv6分组头部的比较

<i>Comparison</i>
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

通过抓包查看IPv6分组的首部

Realtek PCIe GBE Family Controller - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
70	6.604265	168.168.214.20	113.96.233.139	TLSv1.2	Application Data
71	6.657756	113.96.233.139	168.168.214.20	TCP	https > 57659 [ACK] Seq=2446 Ack=3978 win=23936 Len=0
72	6.661806	113.96.233.139	168.168.214.20	TCP	https > 62245 [ACK] Seq=199 Ack=1553 win=1089 Len=0
73	6.709764	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
74	6.709766	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
75	6.709767	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
76	6.709925	168.168.214.20	113.96.233.139	TCP	57659 > https [ACK] Seq=3978 Ack=6646 win=65800 Len=0
77	6.709957	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
78	6.710756	113.96.233.139	168.168.214.20	TCP	[TCP segment of a reassembled PDU]
79	6.710757	113.96.233.139	168.168.214.20	TLSv1.2	Application Data
80	6.710894	168.168.214.20	113.96.233.139	TCP	57659 > https [ACK] Seq=3978 Ack=10831 win=65800 Len=0
81	6.799151	68:8f:84:04:a7:bd	Spanning-tree-(for- STP	MST.	Root = 32768/0/68:8f:84:04:a7:bd Cost = 0 Port = 0x8003
82	6.999540	fe80::21ec:72e5:26cc:ff02::c		SSDP	M-SEARCH * HTTP/1.1

Frame 82: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits)

Ethernet II, Src: 18:60:24:96:fe:83 (18:60:24:96:fe:83), Dst: IPv6mcast_00:00:00:0c (33:33:00:00:00:0c)

Internet Protocol Version 6, Src: fe80::21ec:72e5:26cc:a174 (fe80::21ec:72e5:26cc:a174), Dst: ff02::c (ff02::c)

- 0110 = Version: 6
- 0000 0000 = Traffic class: 0x00000000
- 0000 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
- Payload length: 154
- Next header: UDP (0x11)
- Hop limit: 1
- Source: fe80::21ec:72e5:26cc:a174 (fe80::21ec:72e5:26cc:a174)
- Destination: ff02::c (ff02::c)

User Datagram Protocol, Src Port: 62481 (62481), Dst Port: ssdp (1900)

Hypertext Transfer Protocol

```
0000 33 33 00 00 00 0c 18 60 24 96 fe 83 86 dd 60 00 33.....`$......`
0010 00 00 00 9a 11 01 fe 80 00 00 00 00 00 00 21 ec .....!
0020 72 e5 26 cc a1 74 ff 02 00 00 00 00 00 00 00 00 r.&.t...
0030 00 00 00 00 00 0c f4 11 07 6c 00 9a 46 3b 4d 2d .....!.F;M-
0040 53 45 41 52 43 48 20 2a 20 48 54 54 50 2f 31 2e SEARCH * HTTP/1.
0050 31 0d 0a 48 6f 73 74 3a 5b 46 46 30 32 3a 3a 43 1..Host: [FF02::C
0060 5d 3a 31 39 30 30 0d 0a 53 54 3a 75 72 6e 3a 4d ]:1900.. ST:urn:M
0070 69 63 72 6f 73 6f 66 7a 20 57 69 6e 64 6f 77 73 icrosoft windows
```

File: "C:\Users\ADMINI~1\AppData\Loc... Packets: 150 Displayed: 150 Marked: 0 Dropped: 0 Profile: Default

Figure 20.17 扩展首部的类型

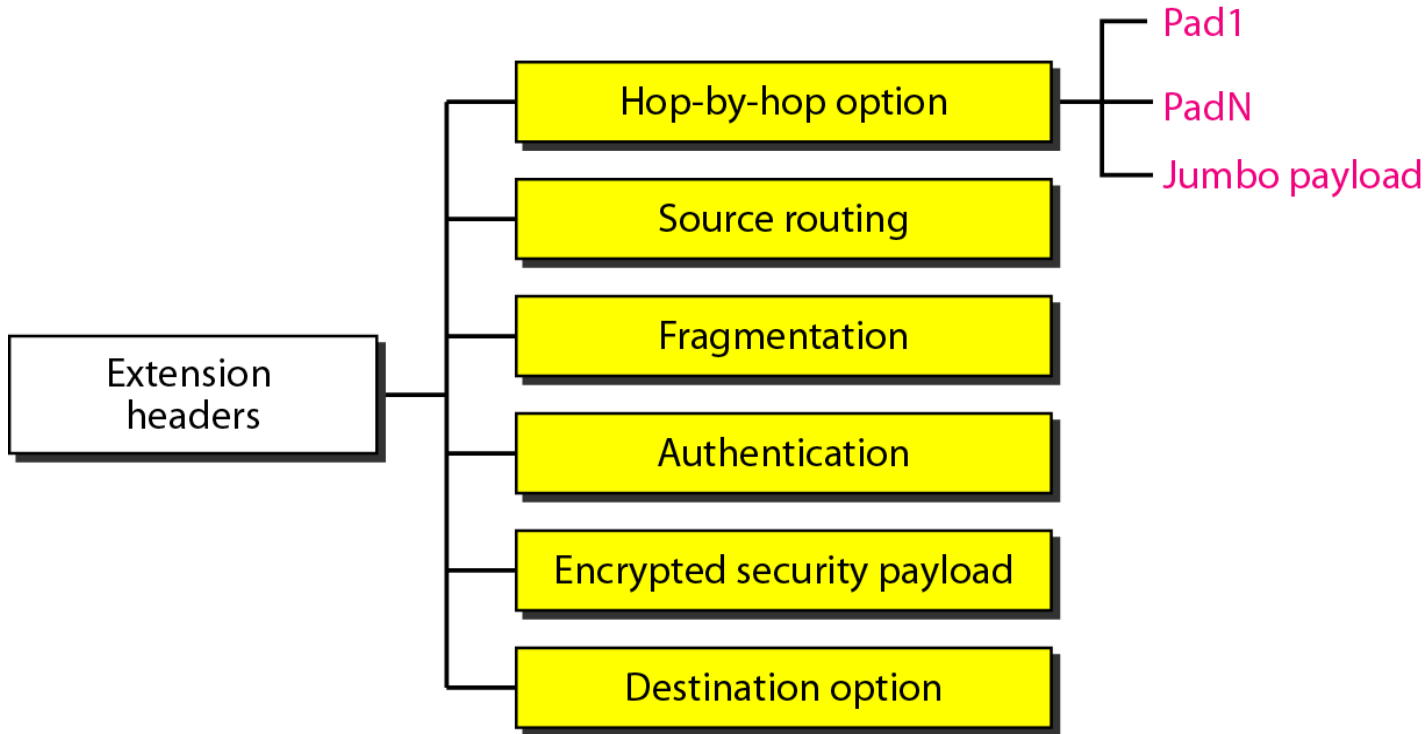


Table 20.10 IPv4选项和IPv6扩展头部的比较

<i>Comparison</i>
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

20-4 IPv4 到 IPv6的过渡

因特网上的系统非常多，所以从 IPv4 过渡到 IPv6不能突然发生。要使每一个在因特网中的系统从IPv4 过渡到 IPv6，需要花费相当长的时间。这种过渡必须是平滑的，以防止 IPv4 和 IPv6 系统间出现任何问题。

本节主题:

双协议栈
隧道技术
头部转换

Figure 20.18 三种过渡策略

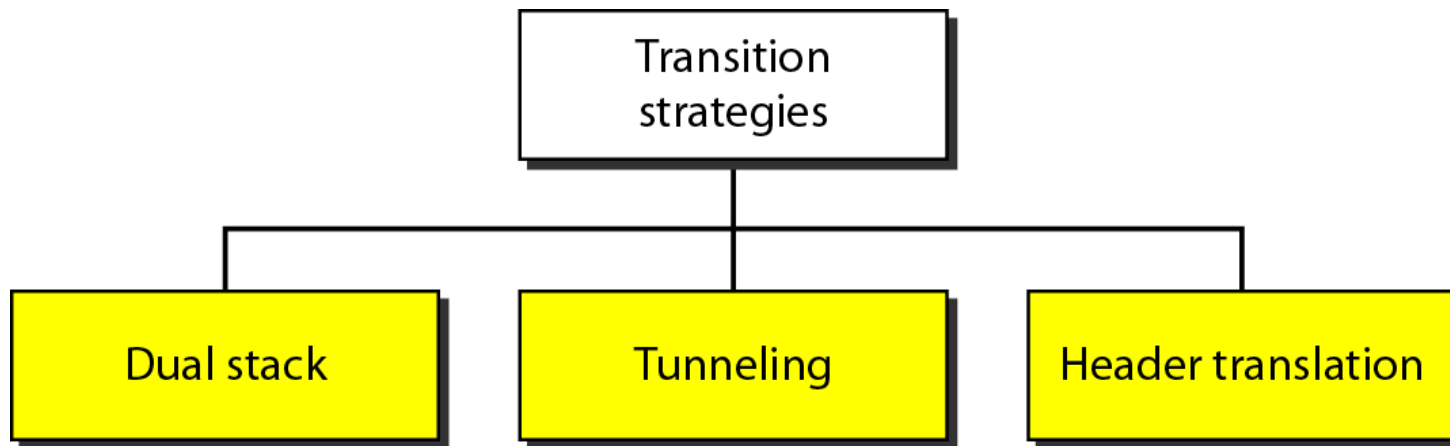
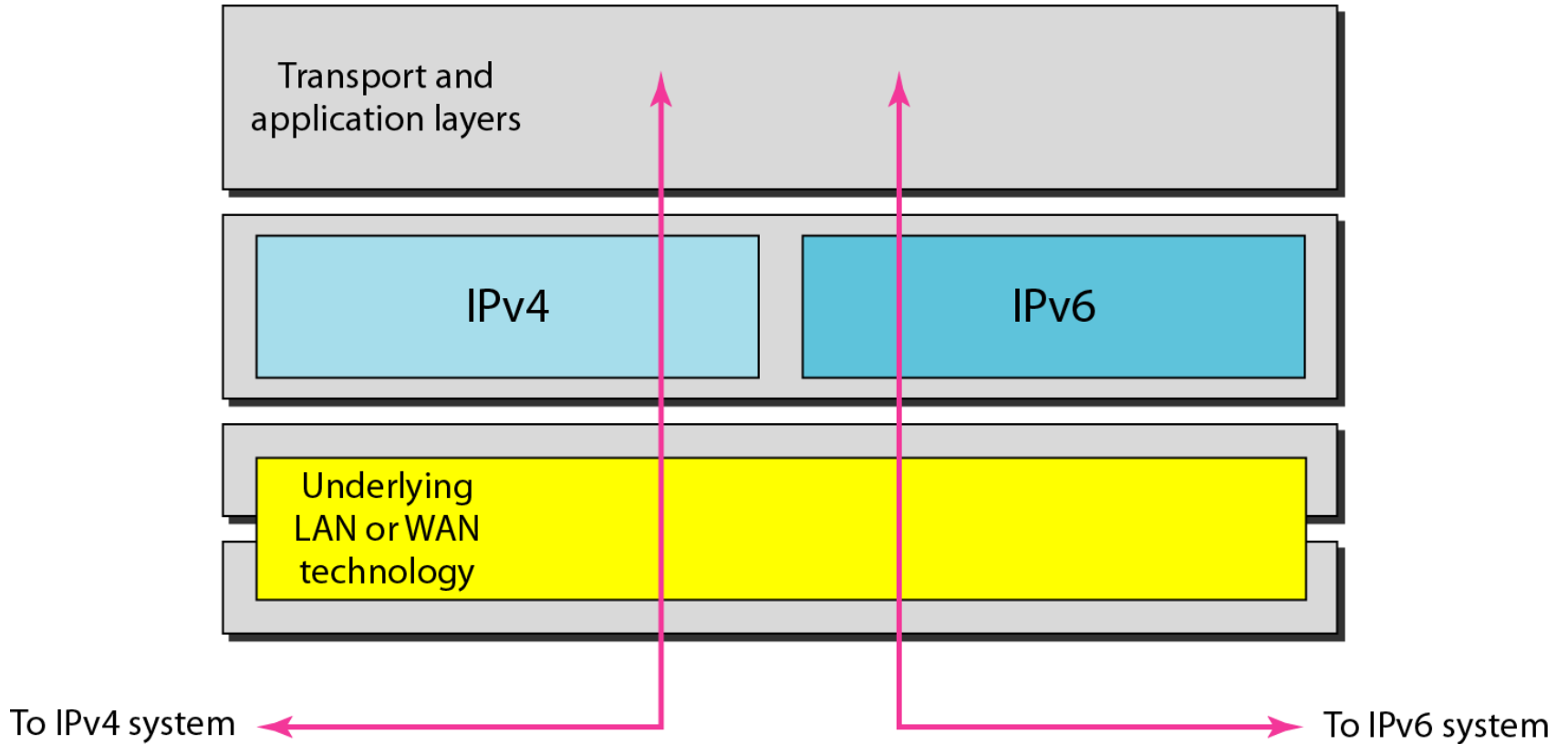


Figure 20.19 双协议栈



其中要使用**DNS**来确定使用哪个协议栈

双协议栈

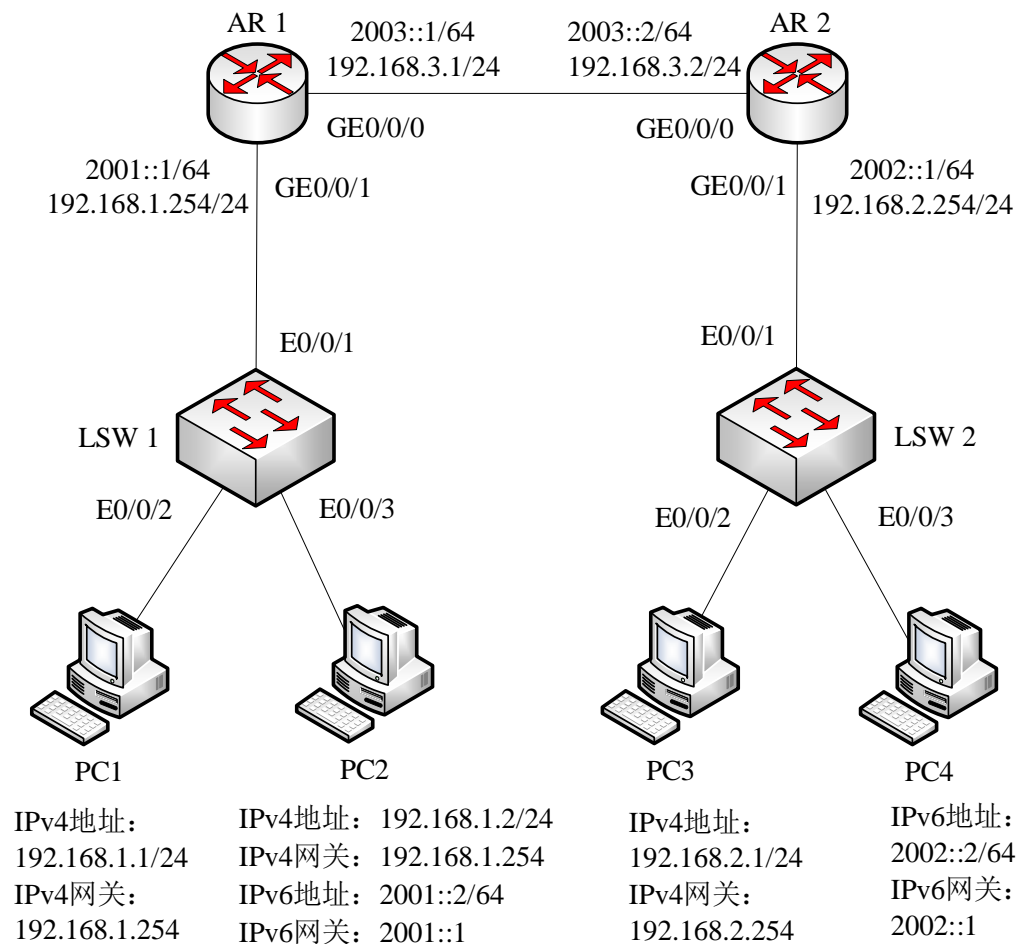


Figure 20.20 隧道策略

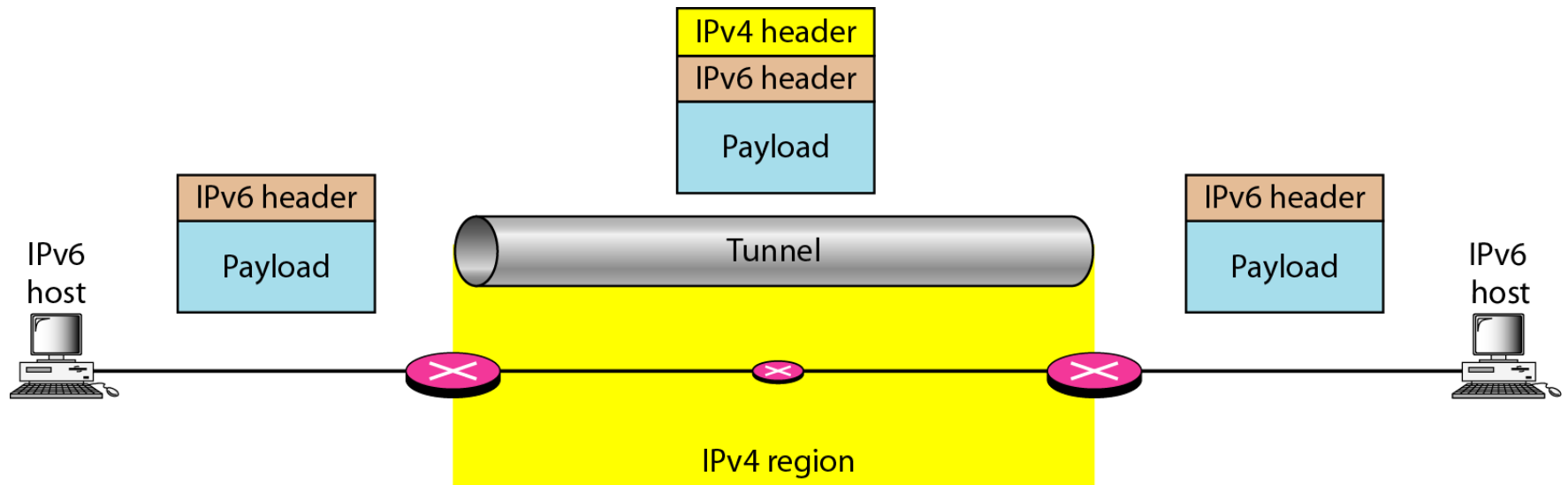


Figure 20.21 头部转换策略

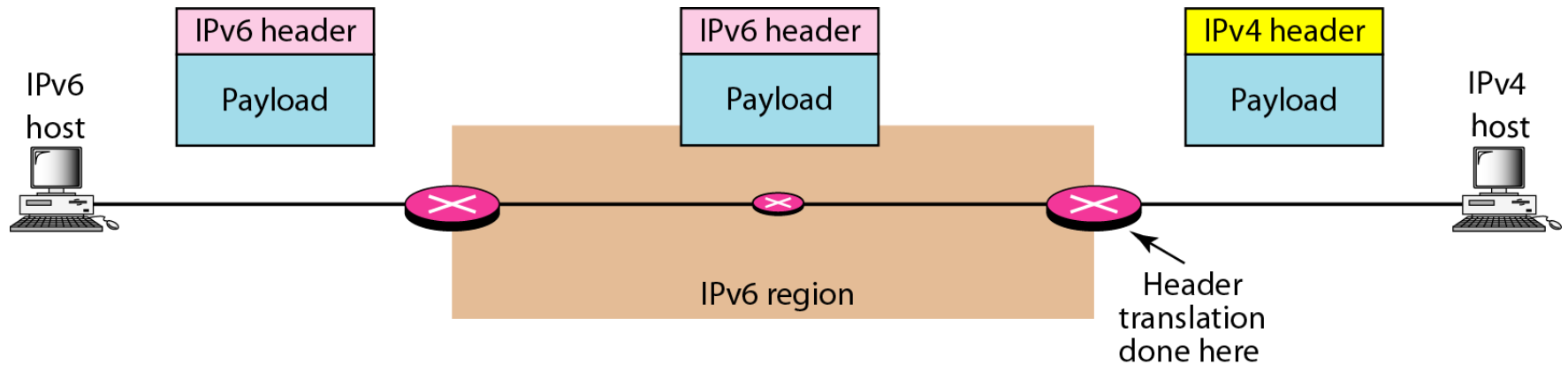


Table 20.11 首部转换

<i>Header Translation Procedure</i>
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

作业

- P401页
- 19, 23

23题的首部信息改为:

0x45 00 00 54 00 03 00 00 20 06 58 41 7C
4E 03 02 B4 0E 0F 02