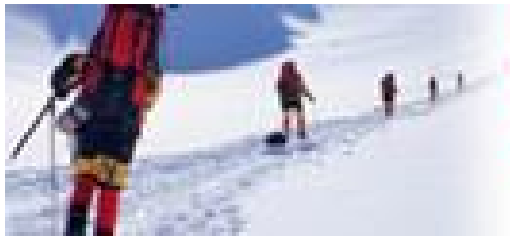


Data Communications and Networking Fourth Edition

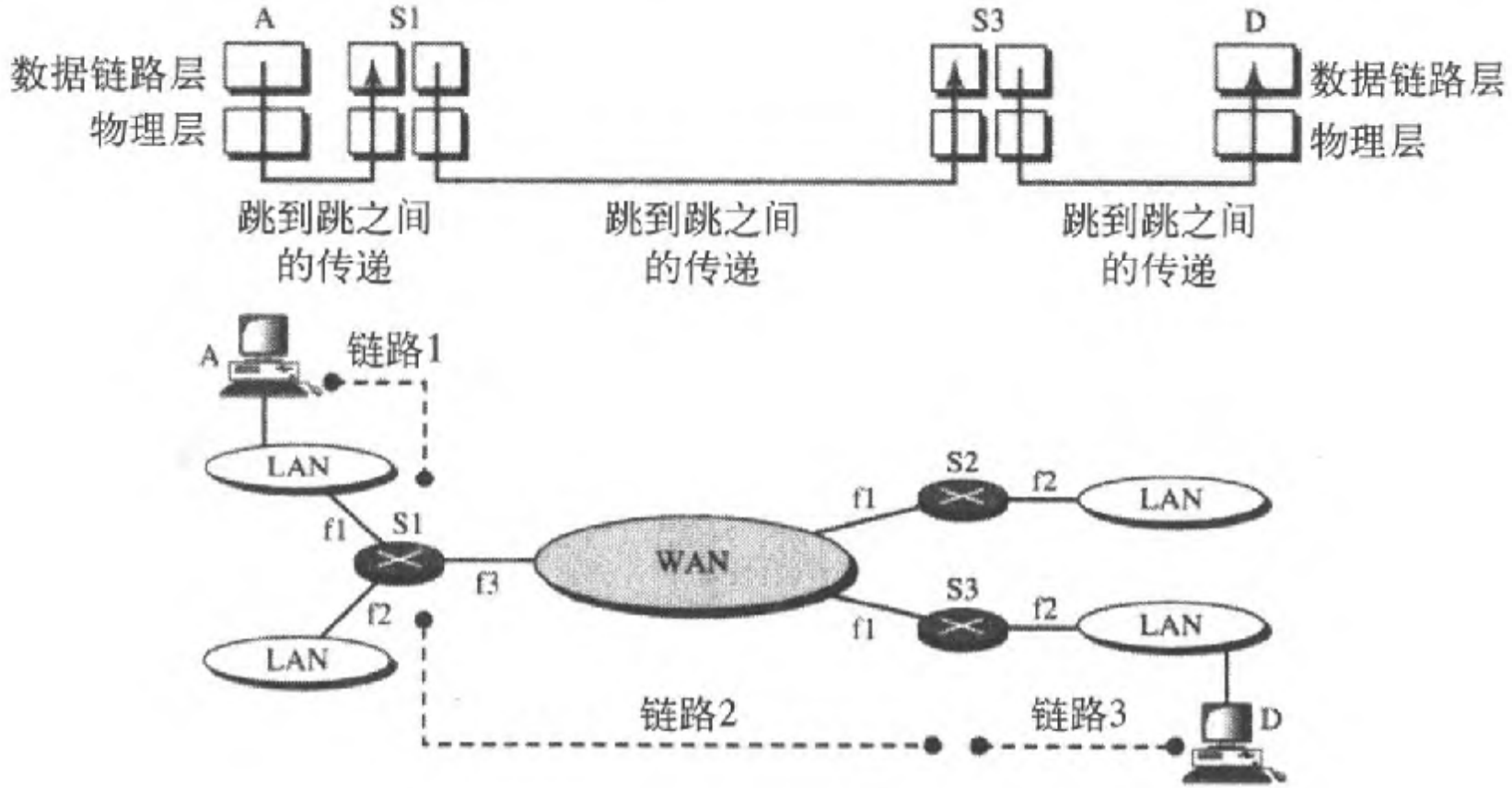
Forouzan

第20章

IP协议

20-1 网际互联

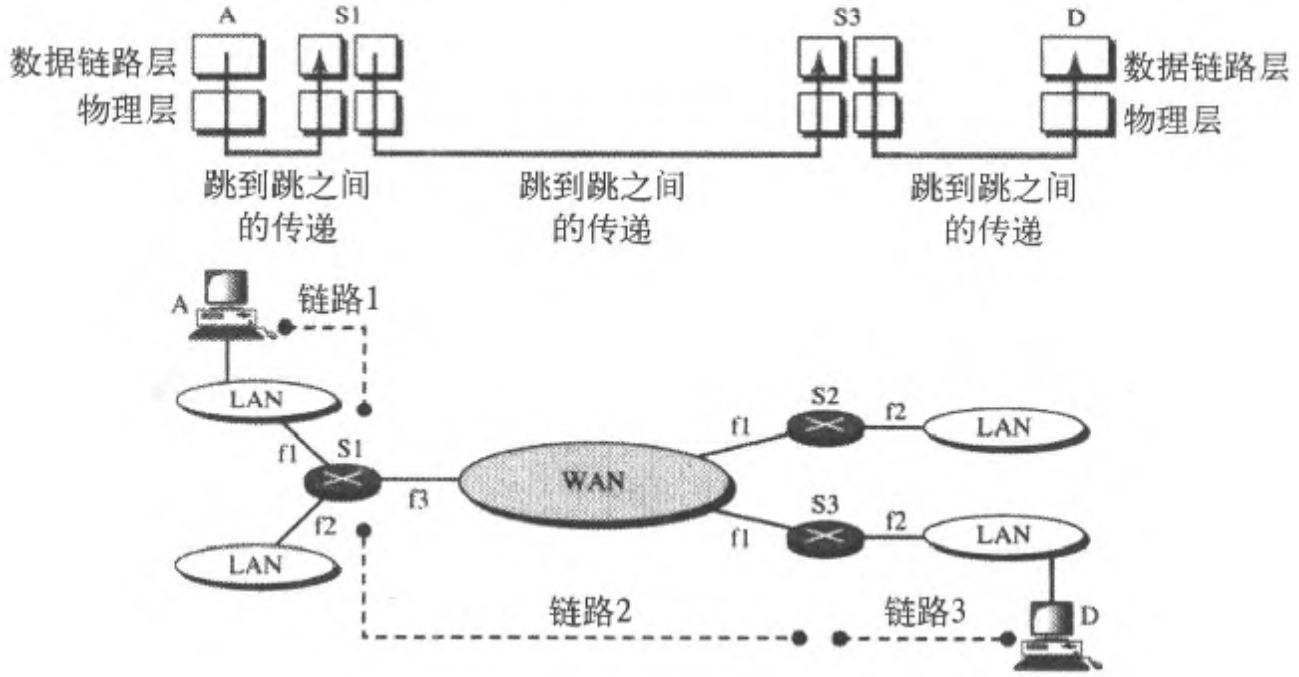
p网络的物理层和数据链路层在本地运行，这两层共同负责网络相邻节点间的数据传递



问题？

p 问题：A发送分组到D，到达路由器S1后，S1如何知道应该从f3接口将其发送出去?在数据链路层（或物理层）并没有规则来帮助S1做出正确的决策；

p 而且，帧也没有携带任何路由选择信息，帧中只包含主机A的MAC地址（源地址）和S1的MAC地址（目的地址）



网络层需求

p为了解决通过多条链路进行传递的问题，设计了网络层（或互联网络层）；网络层不仅负责主机间的传递，还负责通过路由器或（三层）交换机对分组进行路由选择

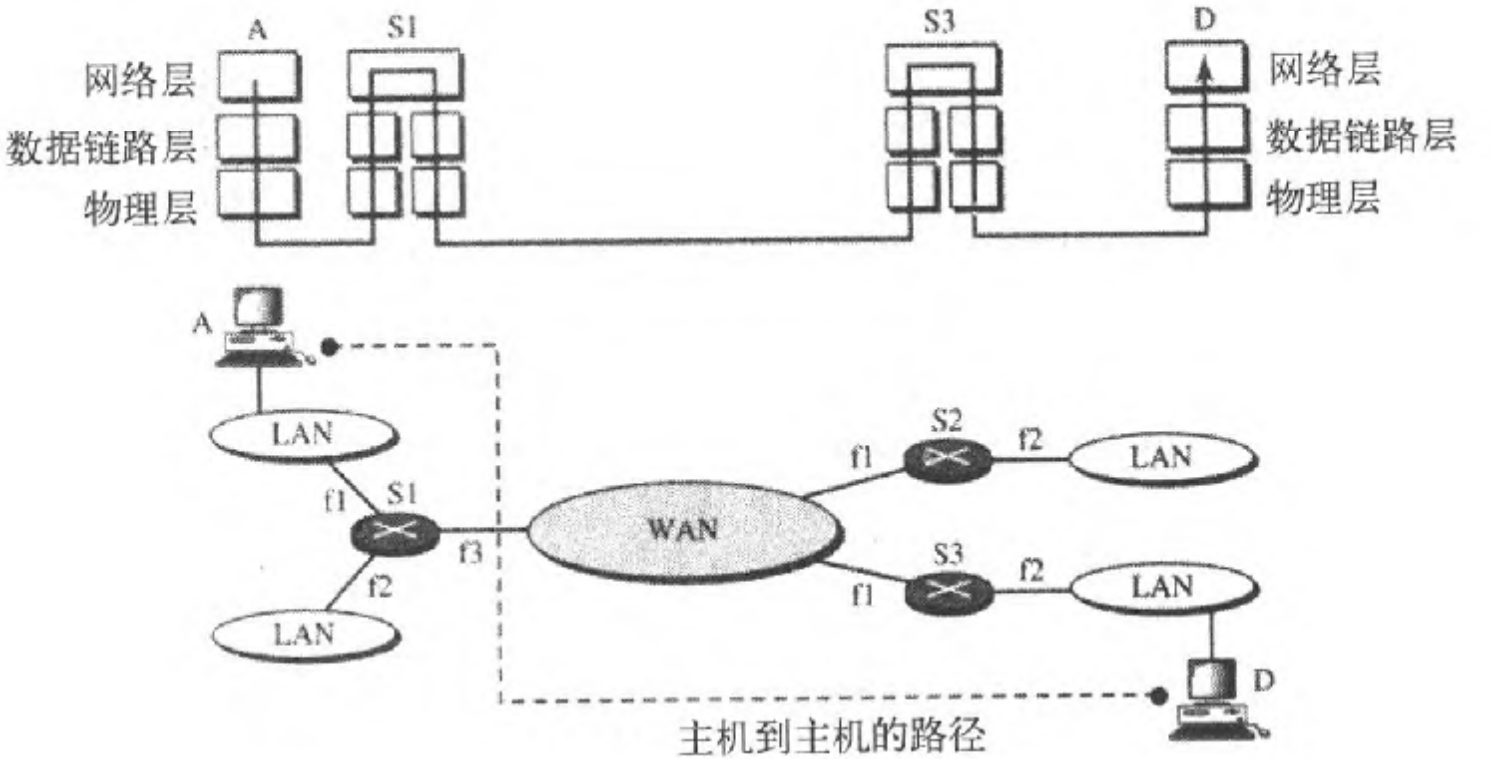
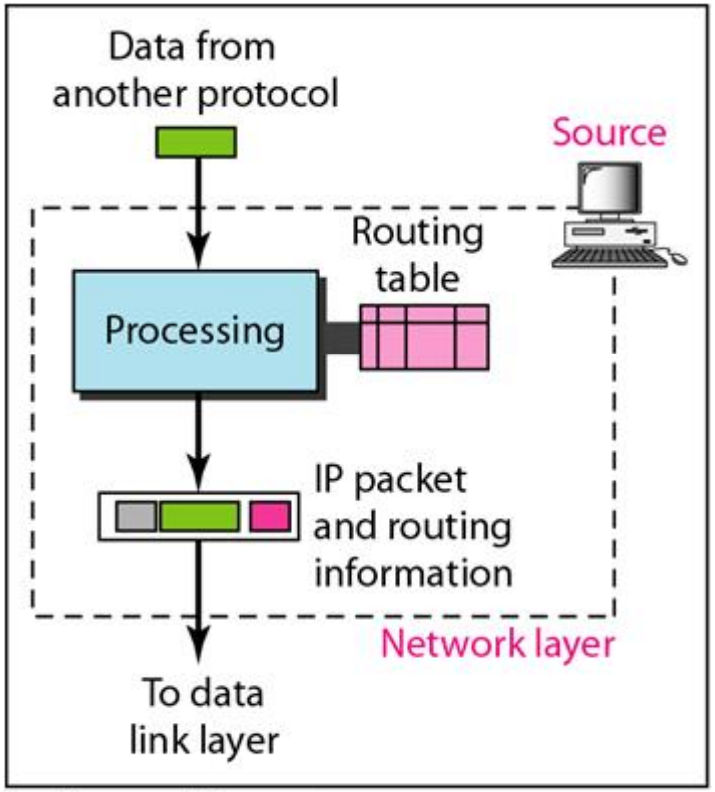


图20.3 源端、路由器端和目的端的网络层

p在源端网络层负责将来自另一个协议（如一个传输层协议或一个路由协议）的输入数据生成一个分组，分组的头部包含源和目的逻辑地址以及其他信息；

p网络层负责检验路由表寻找路由选择信息（如分组出去的接口或下一节点的地址）；

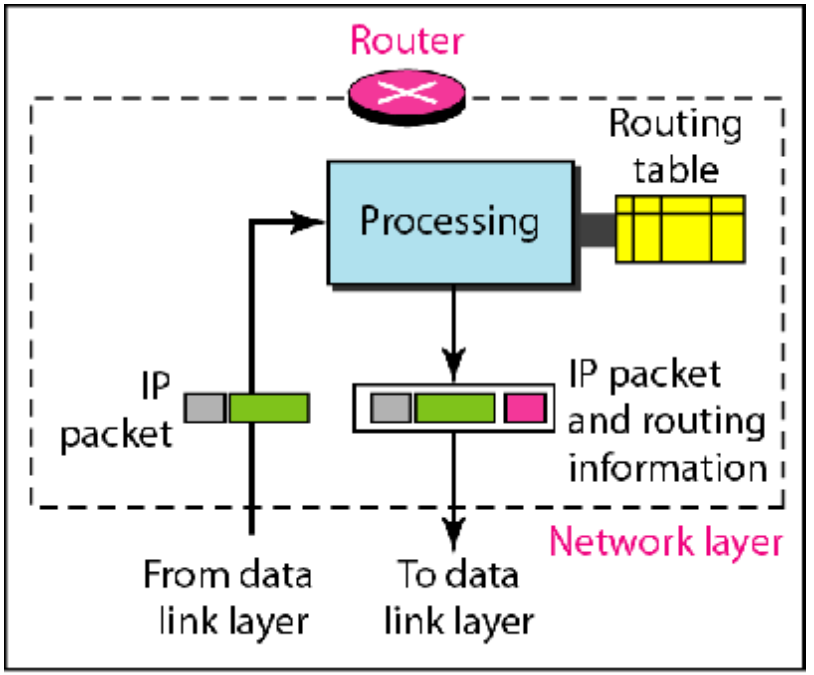
p如果分组太大，那么就得对其进行分段



a. Network layer at source

图20.3 源端、路由器端和目的端的网络层（cont.）

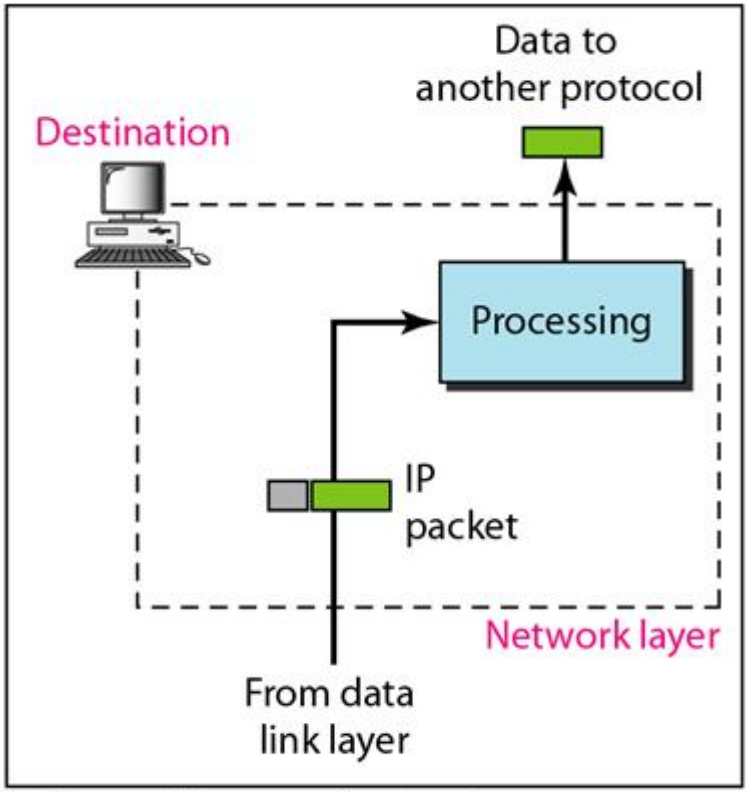
- 路由器中的网络层负责对分组进行路由选择；
- 当一个分组到达时，路由器就从它的路由选择表中为该分组找到一个必须将其发送出去的接口；
- 改变头部的某些内容（例如哪些内容？）后的分组按路由选择信息再传送给数据链路层。



c. Network layer at a router

图20.3 源端、路由器端和目的端的网络层（cont.2）

- 目的端的网络层负责地址验证；
- 它确保分组中的目的地址与主机地址是相同的；
- 如果分组是一个分段，网络层就等待所有的分段到达后再对其进行重组，然后再将重组后的分组交给传输层。



b. Network layer at destination

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。

面向连接的服务和无连接的服务

p分组传递可以用两种网络服务方式来实现；

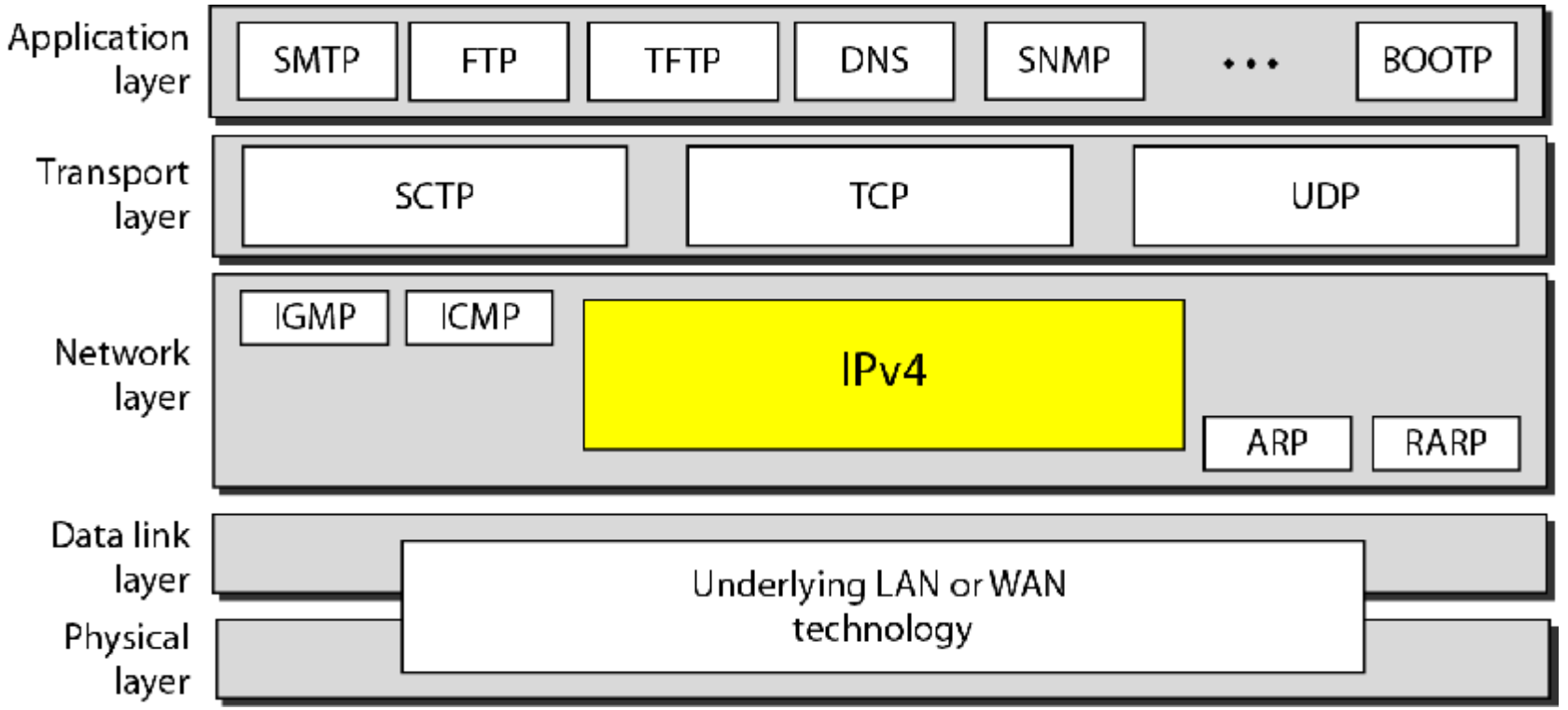
p面向连接的服务：源端在发送一个分组之前首先与目的端建立一个连接；在连接建立后，分组按顺序依次从相同的源端发送到相同的目的端；分组间存在一种关系，它们沿相同的路径按顺序发送；所有分组都传递完毕后，连接终止；一个连接建立之后，对那些具有相同源和目的地址的分组序列，只会进行一次路由策略；交换机不会为每个单独的分组重复计算路由，比如帧中继和ATM等虚电路分组交换方法中；

p无连接的服务：网络层协议独立地对待每个分组，每个分组与任何其他分组没有联系；一个报文中分组可能会也可能不会沿同样的路径到达其目的地；用于数据报分组交换方法中，比如因特网中

因特网的网络层通信是无连接的。

20-2 IPv4

p网际协议第四版（Internet Protocol version 4，IPv4）是TCP/IP协议族（在网络层）使用的传输机制

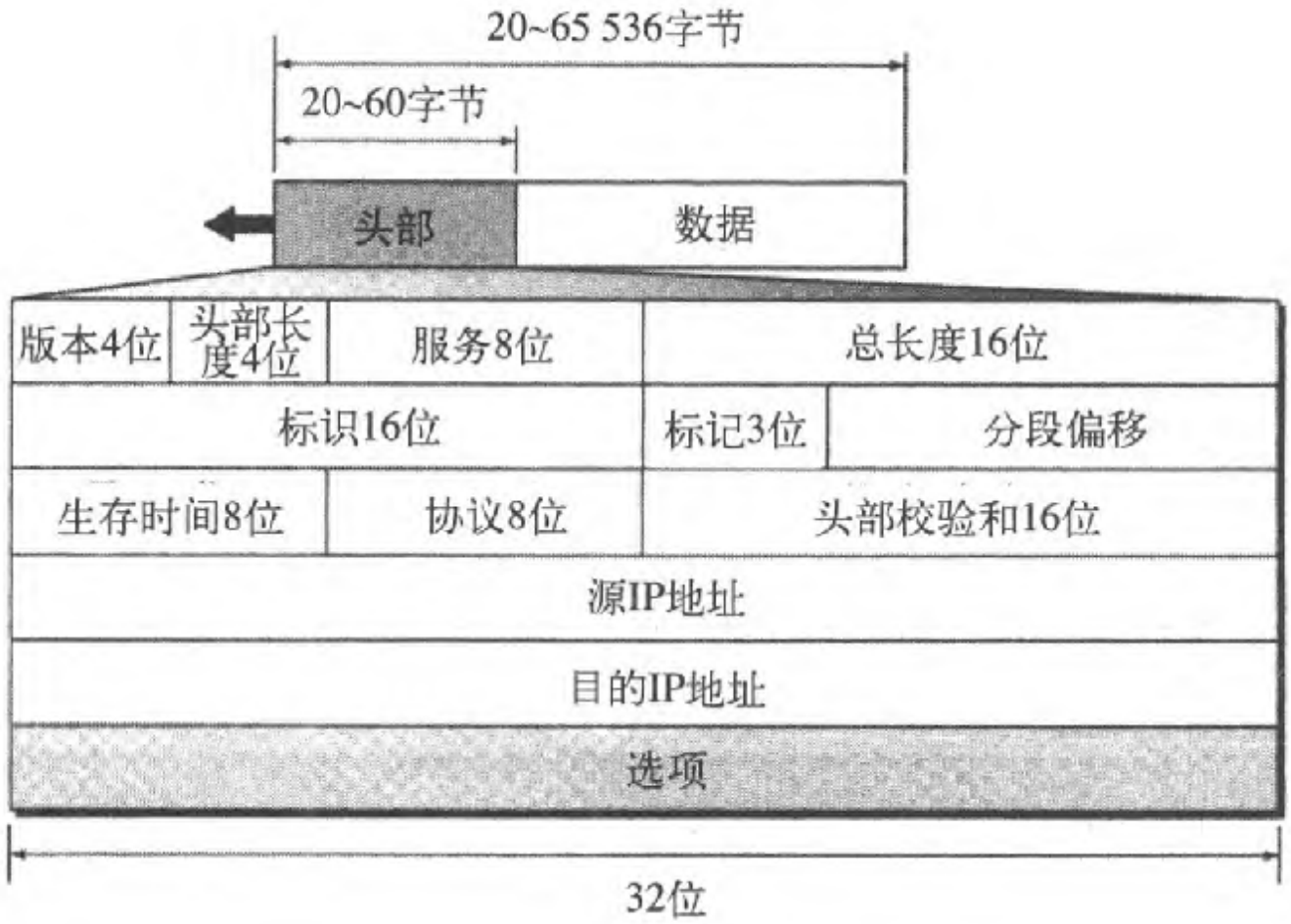


IPv4特点

p不可靠性（unreliable）：尽力传递（best-effort delivery），尽力传递指IPv4不提供差错控制和流量控制（除头部差错检测外）；不确保数据报能成功到达目的地（中间发生问题可能被丢弃）；

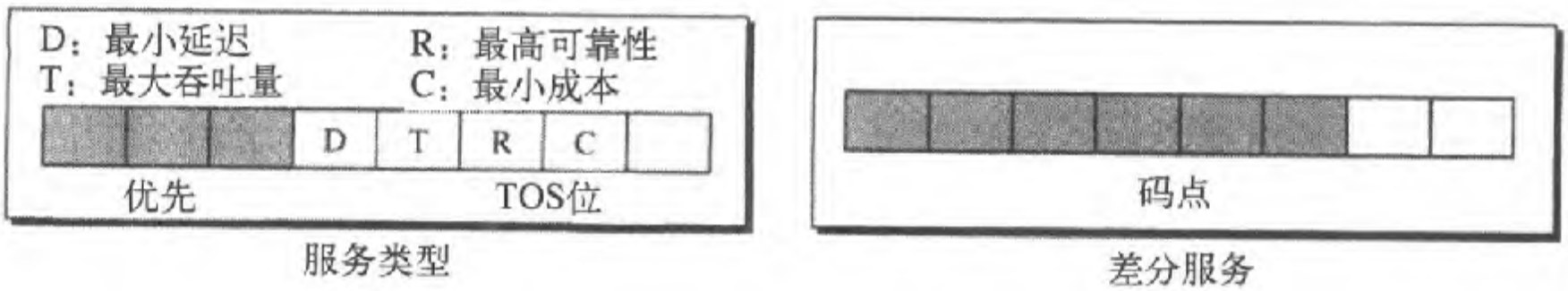
p无连接性（connectionless）：不维护任何关于后续数据报的状态信息；每一分组独立进行处理，而每一分组使用不同的路由传送到目的端；如果一个源端向同一目的端发送许多数据报，那么这些数据报有可能不按顺序到达（有一些数据报也可能遗失或损坏）。

图20.5 IPv4分组（称为数据报）的格式（希望能够记住！）



IPv4数据报

- IPv4数据报是一个可变长分组，由两部分组成：头部和数据部分；
- 头部长度可由20（固定部分）到60个字节组成（包含选项时），4字节对齐，包含与路由选择和传输有关的重要信息；
- 版本号：4位，固定是4（IPv6是6，但后面的格式不一样）；
- 头部长度：4位，以4字节为单位定义数据报头部的总长度；当没有选项时，头部长度是20个字节，而这个字段的值是5（5×4=20）；当选项字段为最大值时，这个字段的值是15（15×4=60）；
- 服务：8位，IETF已经改变了该字段的解释与名称，这个字段以前称为服务类型，现称为差分服务



服务类型

- 在这种解释中，前3位称为优先位，后面的4位称为服务类型（TOS），最后1位未使用；
- 优先（Precedence）：3位子字段，0-7，定义了在出现如拥塞等问题时，数据报的优先级；当路由器出现拥塞并必须丢弃一串数据报时，最低优先值的数据报将首先被丢弃；
- 在版本4中未使用优先子字段；
- TOS子字段：每一位都有其特殊的意义，只能有一位的值为1

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

表20.2 服务类型的默认值

Protocol	*TOS Bits*	*Description*
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

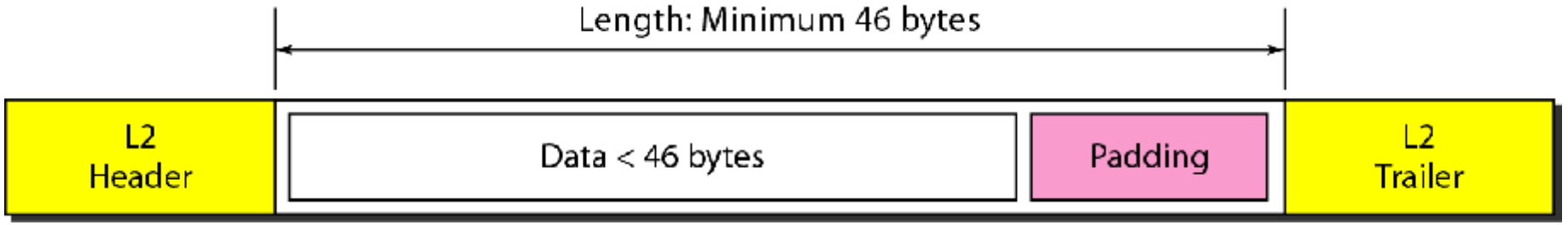
差分服务

- 在这种解释中，前6位组成码点（code point）子字段，而后2位不用；
- 码点子字段可有两种使用方法：
  - Ø当最右边3位都是0时，最左边3位与服务类型解释中的优先位相同；
  - Ø当最右边3位不全为0时，则6位由因特网或本地机构按表20.3（P386）赋予的优先级定义64种服务；第一类包含32种服务类型，由IETF分配，第二与第三类包含16种服务，分别由本地组织机构使用或是临时的用作实验目的；这些分配还未最终确定

IPv4数据报（cont.）

- 总长度：16位，定义了一个以字节计的IPv4数据报的总长度；某些网络不能将65535字节的数据报封装成它们的帧，要通过这些网络就必须将这些数据报进行分段；很多情况不需要该字段（接收到帧后除去头部和尾部即可），但有填充字节时用来确定实际的数据长度（见下页图）；
- 标识：用于分段中，与源IPv4地址唯一的定义一个数据报；
- 标记：用于分段中；
- 分段偏移：用于分段中；
- 生存时间TTL：用来控制一个数据报所通过路由器的最大跳数（路由器数），由源端设定初始值，每一个路由器将该值减1，减为0时丢弃数据报，可防止环路；

图20.7 一个小的数据报封装在以太网帧中



IPv4数据报（cont.2）

- 协议：8位，定义了使用此IPv4层服务的高层协议，高层协议如TCP、UDP、ICMP和IGMP等的数据能够封装到IPv4数据报中，这个字段指明IPv4数据报必须传递到的最终目的协议；
- 校验和：16位，头部校验和；
- 源地址：32位，源端的IPv4地址，传输期间保持不变；
- 目的地址：32位，目的端的IPv4地址，传输期间保持不变。

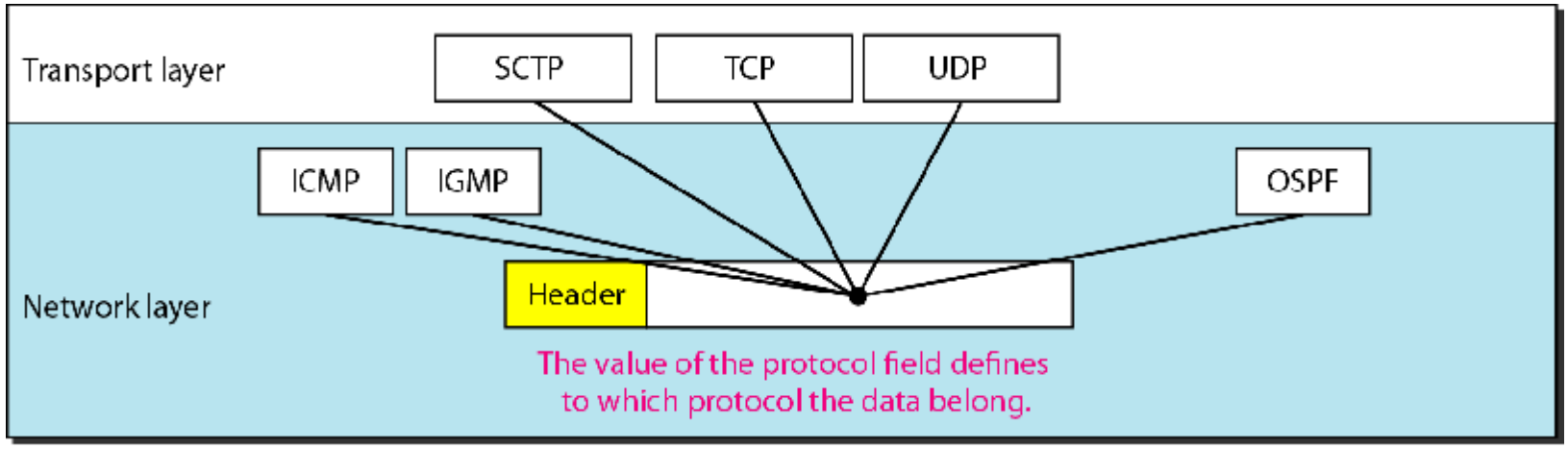


表20.4 协议值

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

例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，是一个无效长度（2×4=8），因为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。

例20.2

在一个IPv4分组中，头部长度字段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：

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

例20.3

在一个IPv4分组中，头部长度字段的值是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：

头部长度字段值是5，就是说头部的总字节数为5×4=20字节（无选项），总长度是40字节，也就是说，这个分组携带40-20=20个字节的数据。

例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，它经历了多少跳？数据是属于上层的哪一个协议？

解：

为了求生存时间字段，我们跳过了8个字节（16个十六进制数字），生存时间字段是第9个字节，它的值是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。

分段

- 一个数据报可以通过几个不同的网络进行传输；
- 每个路由器将它所接收的帧拆封成IPv4数据报，对它进行处理，然后再将它封装成另一个帧；
- 接收到的帧的格式和长度取决于此帧刚刚经过的物理网络所使用的协议，发出去的帧的格式和长度则取决于此帧将要经过的物理网络所使用的协议；
- 例如，如果一个路由器将一个局域网连接到一个广域网，那么它接收到的帧是一个局域网格式的帧，发出去的帧是广域网的格式。

图20.9 最大传输单元（MTU）

- 每一个数据链路层协议都有其自己的帧格式，这格式中定义的一个字段是数据字段的最大长度；
- 换言之，当数据报封装成帧时，该数据报的总长度必须小于这个最大数据长度，这是由网络所使用的硬件和软件给出的限制所定义的；
- MTU的值取决于物理网络协议

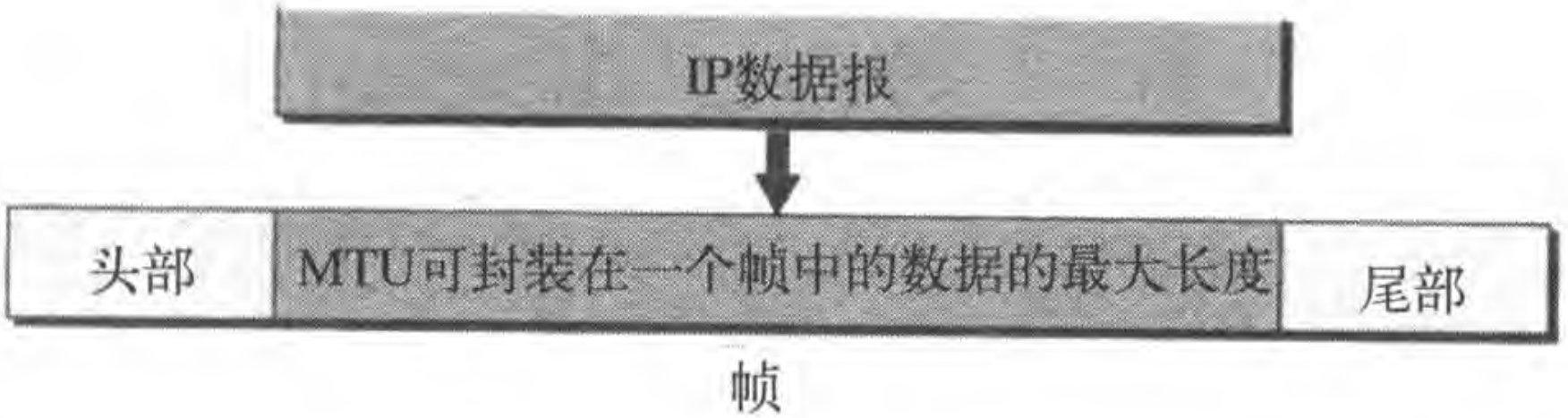


表20.5 某些网络的MTU值

Protocol	*MTU*
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

分段（cont.）

- 为了使IPv4与物理网络无关，协议设计者决定使IPv4数据报的最大长度等于最大传输单元（目前值为65535字节）；
- 如果MTU等于65535，可使传输更加有效；然而，对其他一些物理网络，就要将数据报进行分割，使其能够通过这些网络，这个过程称为分段（fragmentation）；
- 源端通常不对IPv4分组进行分段，传输层将数据分割成能适合在数据链路层所用IPv4的长度；
- 当对一个数据报进行分段时，每个分段都有其自己的头部，其中大部分的字段是重复的，但有些发生了变化；
- 如果一个已分段的数据报遇到一个更小MTU的网络，那么已分段的数据报还可再进行分段（可能经过多次分段）；
- 在IPv4中，数据报可能被主机或其路径中的任何路由器进行分段，然而，数据报的重组只能在目的主机上进行

与分段相关的字段

- 对一个数据报进行分段的主机或路由器必须改变三个字段的值：标志、分段偏移和总长度；
- 与分段和重组相关的字段是：标识、标记和分段偏移；
- 标识：16位，标识一个从源主机发出的数据报；当数据报离开源主机时，这个标识与源IPv4地址唯一地定义这个数据报；
- 标记：3位，第一位保留，第二位称为“不分段位”，如果值是1则机器不能将该数据报进行分段（若必须分段，则丢弃报文并发送差错报文）；第三位称为“多分段位”，值为1表示在该分段后还有更多分段，值为0表示它是最后一个或唯一的分段；
- 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

D: Do not fragment
M: More fragments

图20.11 分段示例

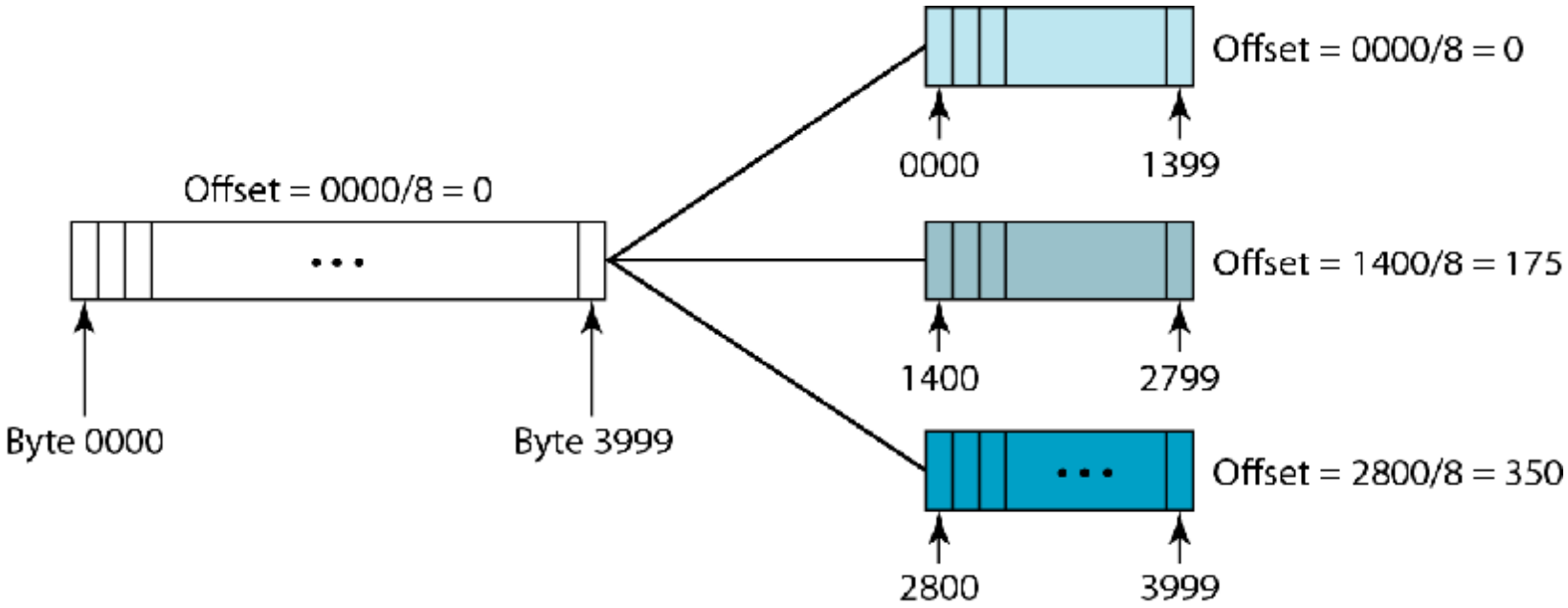
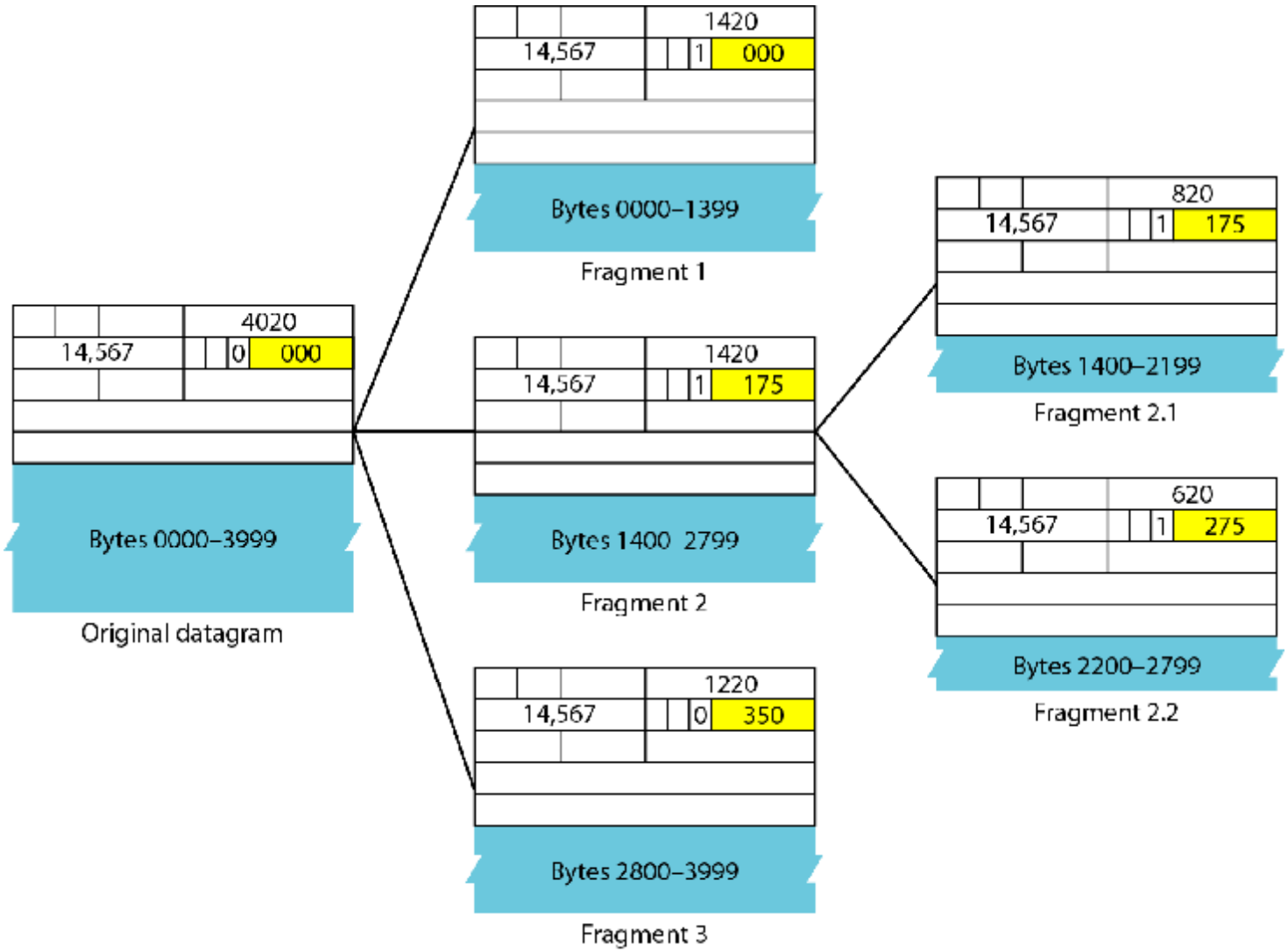

Offset = 0000/8 = 0
Byte 0000
Byte 3999
Offset = 0000/8 = 0
0000
1399
Offset = 1400/8 = 175
1400
2799
Offset = 2800/8 = 350
2800
3999

图20.12 分段的细节示例

- 所有分段的标识字段的值都相同；
- 除最后一个分段外，所有分段的标记中的值都是设置为多个分段；
- 分段偏移值永远是相对于原始的数据报；
- 分段增加传输数据



重组策略

- 显然，即使每一个分段走不同的路径，并在到达时失序，最终目的主机也能用收到的这些分段（假定没有丢失）重组成原始数据报；
- 所使用的策略如下：
  - Ø1.第一个分段的偏移字段的值为0；
  - Ø2. 将第一个分段长度除以8，其结果就是第二个分段的偏移值；
  - Ø3. 将第一个和第二个分段的总长度除以8，其结果为第三个分段的偏移值；
  - Ø4. 继续以上过程，最后一个分段的多个分段位的值为0。

例20.5

到达的一个分组的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。

例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）。

例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。

例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？我们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，我们将偏移值乘8，这就是说第一个字节的编号是800。但我们不能确定最后字节的编号，除非知道数据的长度。

例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：

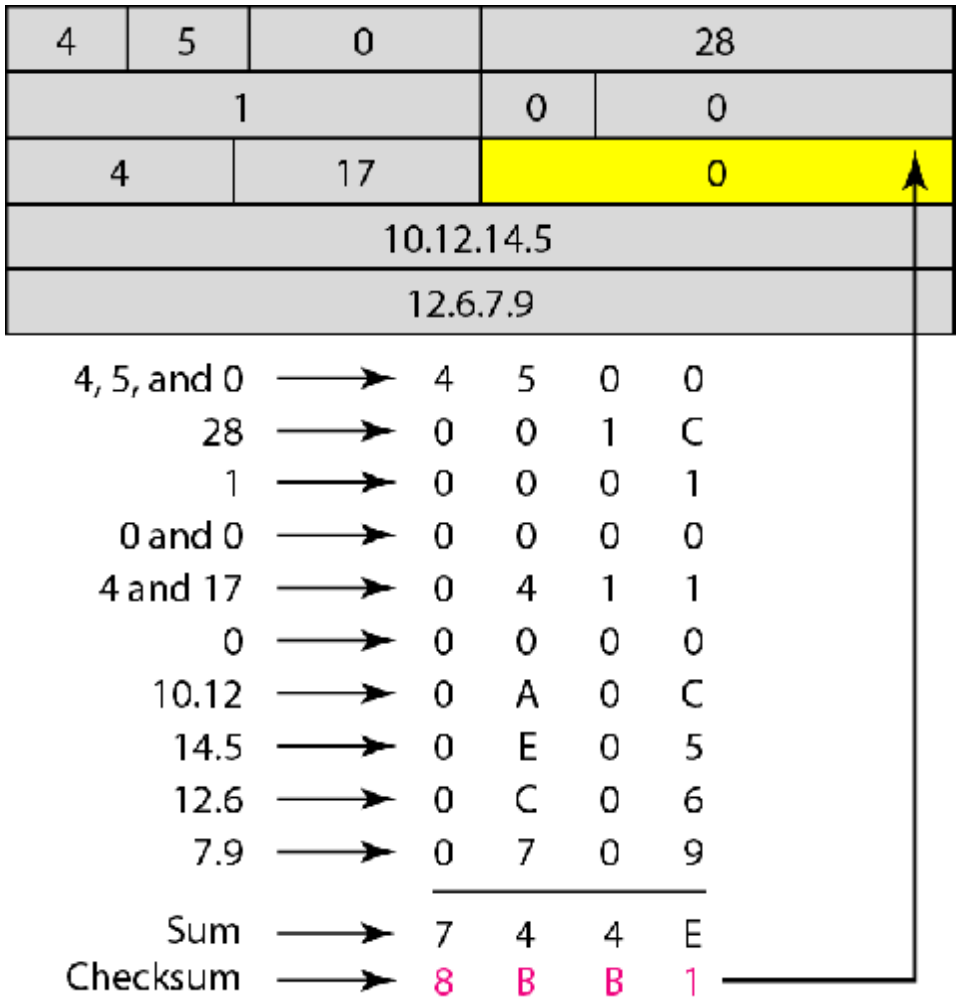
第一个字节的编号是100×8=800。总长度是100字节，头部长度是20个字节（5×4），这就是说这个数据报有80个字节；如果第一个字节的编号是800，则最后字节的编号是879。

校验和字段（16位）

- 计算方法：首先将校验和字段置为0，然后将整个头部划分为16位的部分，并将各部分相加，将计算结果（和）取反码，插入到校验和字段中（接收方如何校验？）；
- IPv4分组中的校验和只对头部进行，而不在数据部分进行，这有两点很充足的理由；
- 首先，所有将数据封装在IPv4数据报中的高层协议中都有覆盖整个分组的校验和，因此，IPv4数据报的校验和就不必校验所封装的数据部分；
- 其次，每经过一个路由器，IPv4数据报的头部就要改变一次（？），但数据部分不改变，因此，校验和只对发生变化的部分进行校验，如果包含数据部分，会降低处理效率。

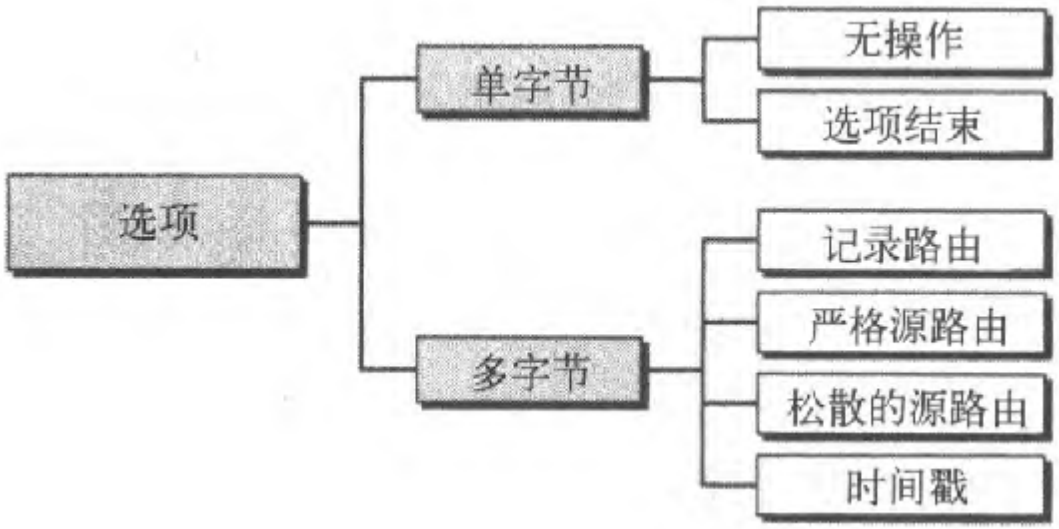
例20.10

图20.13表示了一个对没有选项的IPv4首部计算校验和的过程。首部分成多个16位的部分，将所有这些部分相加，再将得到的和取反码，将其结果插入到校验和字段中。



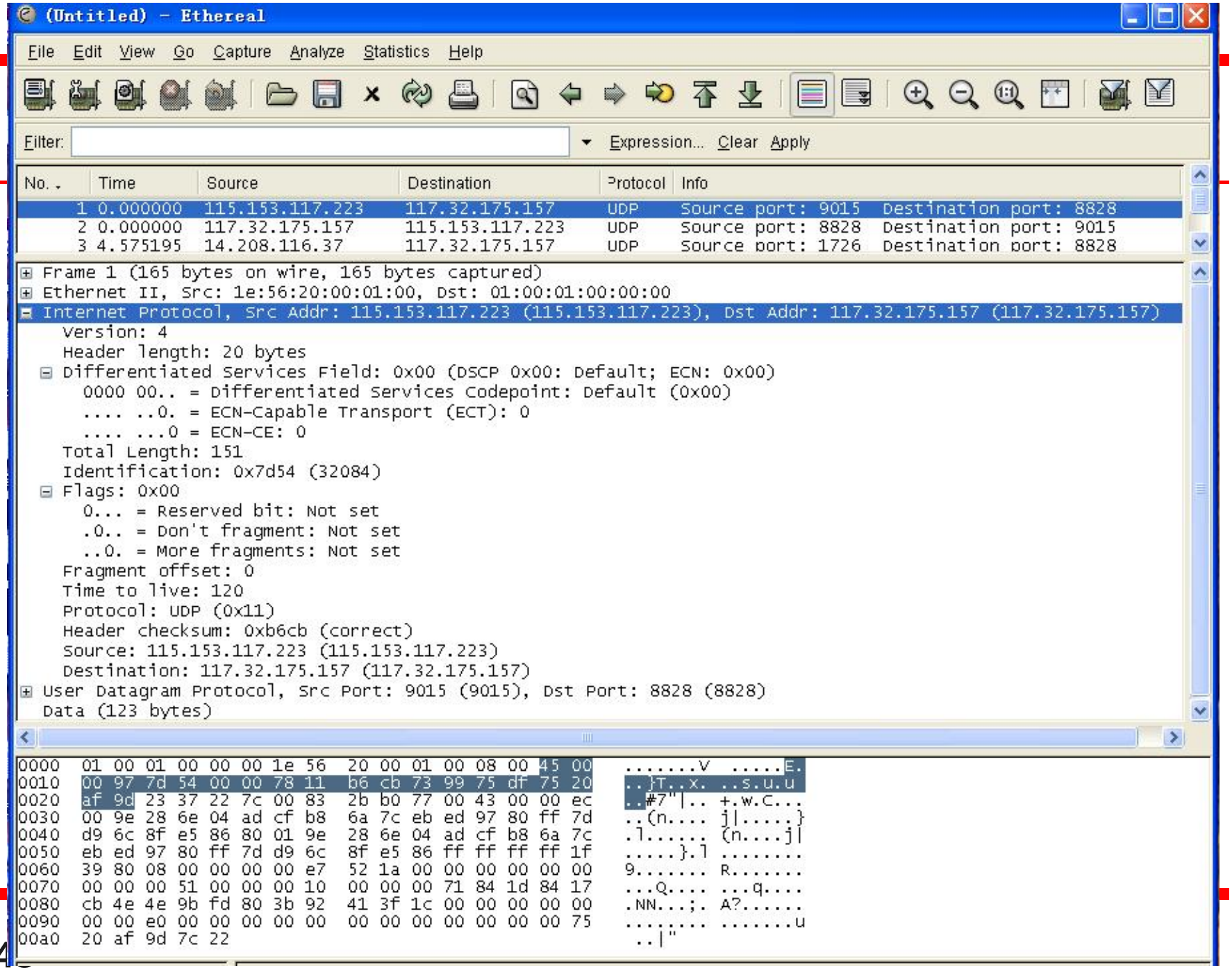
选项字段

- IPv4数据报的头部由两部分组成：固定部分（20个字节）与可变部分（由若干选项组成，最长可达40个字节）；
- 选项并不是必需的，它们用于网络测试和调试；
- 虽然选项并非IPv4数据报必需的部分，但选项处理却是IPv4软件的必需部分



IPv4中主要选项概述

- 无操作选项：1字节选项，用做选项之间的填充符；
- 选项结束选项：1字节，用于选项字段结束时的填充，只能用做最后一个选项；
- 记录路由选项：用来记录处理数据报的因特网路由器，用于调试和管理；
- 严格源路由选项：用来预先确定数据报在因特网中传送时的路由；
- 松散的源路由选项：与严格源路由选项相似，但更加宽松，必须访问表中的路由器，但数据报还可以访问其他的路由器；
- 时间戳选项：用来记录路由器处理数据报的时间


(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 115.153.117.223 117.32.175.157 UDP Source port: 9015 Destination port: 8828
2 0.000000 117.32.175.157 115.153.117.223 UDP Source port: 8828 Destination port: 9015
3 4.575195 14.208.116.37 117.32.175.157 UDP Source port: 1726 Destination port: 8828
Frame 1 (165 bytes on wire, 165 bytes captured)
Ethernet II, Src: 1e:56:20:00:01:00, Dst: 01:00:01:00:00:00
Internet Protocol, Src Addr: 115.153.117.223 (115.153.117.223), Dst Addr: 117.32.175.157 (117.32.175.157)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 151
Identification: 0x7d54 (32084)
Flags: 0x00
0... = Reserved bit: Not set
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 120
Protocol: UDP (0x11)
Header checksum: 0xb6cb (correct)
Source: 115.153.117.223 (115.153.117.223)
Destination: 117.32.175.157 (117.32.175.157)
User Datagram Protocol, Src Port: 9015 (9015), Dst Port: 8828 (8828)
Data (123 bytes)
0000 01 00 01 00 00 00 1e 56 20 00 01 00 08 00 45 00 .....V ....E.
0010 00 97 7d 54 00 00 78 11 b6 cb 73 99 75 df 75 20 ..}T..x. ..s.u.u
0020 af 9d 23 37 22 7c 00 83 2b b0 77 00 43 00 00 ec ..#7"|.. +.w.C...
0030 00 9e 28 6e 04 ad cf b8 6a 7c eb ed 97 80 ff 7d ..(n.... j|.....}
0040 d9 6c 8f e5 86 80 01 9e 28 6e 04 ad cf b8 6a 7c .l...... (n....j|
0050 eb ed 97 80 ff 7d d9 6c 8f e5 86 ff ff ff ff 1f .....}.l ....
0060 39 80 08 00 00 00 00 e7 52 1a 00 00 00 00 00 00 9....... R.......
0070 00 00 00 51 00 00 00 10 00 00 00 71 84 1d 84 17 ...Q.... ...q....
0080 cb 4e 4e 9b fd 80 3b 92 41 3f 1c 00 00 00 00 00 .NN...;. A?......
0090 00 00 e0 00 00 00 00 00 00 00 00 00 00 00 00 75 ....u
00a0 20 af 9d 7c 22 ..|"

20-3 IPv6

- IPv4的一些缺点使它对飞速发展的因特网有些不适应：
  - 虽然有子网化、无类寻址和NAT等，但地址耗尽是一个长期的问题；
  - 因特网必须能适应实时音频和视频传输，这要求最小延迟的策略和预留资源，而IPv4并没有提供；
  - 对于某些应用，因特网必须能够对数据进行加密和鉴别，IPv4不提供数据的加密和鉴别
- 为了克服这些缺点，IPv6（也称为下一代网际协议，IPng）被提出，并已成为标准；
- IPv6网际协议修改了很多，以适应因特网不可预见的增长；
- IP地址格式和长度以及分组的格式都改变了，相关的一些协议，如ICMP也修改了；网络层的其他一些协议，如ARP、RARP和IGMP则被取消或包含在ICMPv6协议之中；路由选择协议，如RIP和OSPF也进行了少量的修改以适应变化；
- 通信专家预计IPv6及其相关协议将（很快？）取代当前的IP版本。

IPv6优点

- 更大的地址空间：比IPv4增加了很多（2^{96}倍）；
- 更好的头部格式：新的头部格式，选项与基本头部分开，若需要可将选项插入到基本头部与上层数据之间，从而简化和加速了路由选择过程，因为大多数的选项不需要由路由器检查；
- 新的选项：IPv6有一些新的选项来实现附加的功能；
- 允许扩充：满足新的技术或应用的需要；
- 支持资源分配：服务类型字段被取消，增加了流标号机制使得源端可以请求对分组进行特殊的处理，可用来支持像实时音频和视频的通信量；
- 支持更多的安全性：IPv6中的加密和鉴别选项提供了分组的保密性和完整性。

图20.15 IPv6 数据报头部和有效载荷

- 每个分组由基本头部和有效载荷组成；
- 有效载荷由两部分组成：可选的扩展头部和来自上层的数据；
- 基本头部占用40字节，而扩展头部和来自上层的数据可以包含多达65535字节的信息

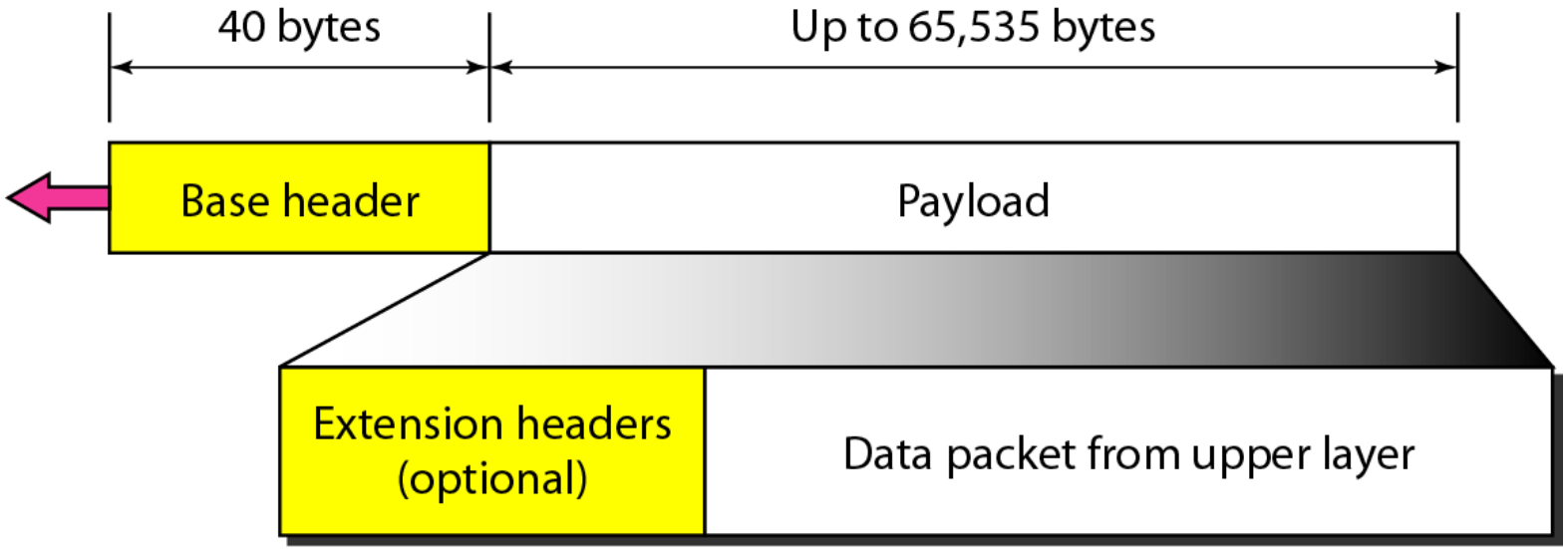
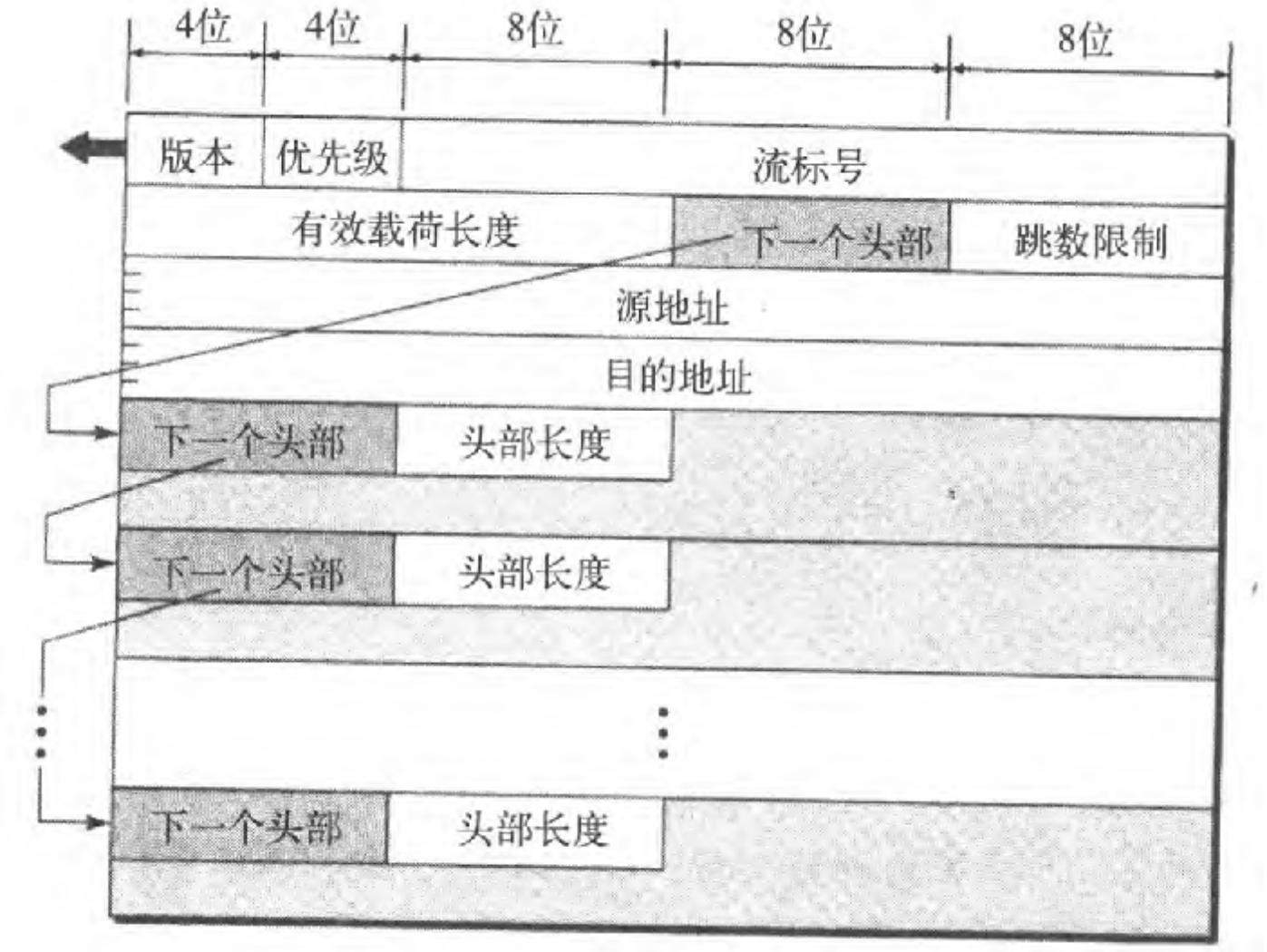


图20.16 IPv6数据报的格式



IPv6头部字段

- 版本：4位，值为6；
- 优先级：4位，定义了当发生通信量拥塞时分组的优先级（后面讨论）；
- 流标号：3字节，24位，用来对特殊的数据流提供专门处理（后面讨论）；
- 有效载荷长度：2字节，定义了不包括基本头部的IP数据报剩余部分的总长度；
- 下一个头部：8位，定义了数据报中跟随在基本头部之后的头部；下一个头部或者是IP所使用的可选的扩展头部，或者是上层协议的（如UDP或TCP）头部，每一个扩展头部也包含这个字段，表20.6 给出了下一个头部的值（见下页）；
- 跳数限制：8位，与IPv4中的TTL字段作用一样；
- 源地址：16字节，用来识别数据报的原始源端；
- 目的地址：16字节，通常用来识别数据报的最终目的地，然而若使用了源路由选择，该字段就包含了下一个路由器的地址。

表20.6 IPv6下一个头部的代码

Code	*Next Header*
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

优先级

- 优先级字段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；
- 如果由于拥塞原因两个连续的数据报中必须丢弃一个，则具有较低的分组优先级的数据报将被丢弃；
- IPv6将通信量划分为两大类：可进行拥塞控制的和不可进行拥塞控制的；
- 如果源端在出现拥塞时能够适应这种情况使其通信量下降，则此通信量称为可进行拥塞控制的通信量；例如，使用滑动窗口协议的TCP协议能够很容易地对通信量进行响应；
- 在可进行拥塞控制的通信量中，分组可以延迟到达，或丢失，或不按序接收；
- 可进行拥塞控制的数据被指定为从0（低）到7（高）的优先级。

表20.7 可进行拥塞控制的通信量的优先级

Priority	*Meaning*
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

0-7优先级描述

- 未指明的通信量：不需要定义优先级时，就分配优先级0；
- 后台数据：优先级1，定义的数据通常是在后台传递，如新闻的传递；
- 不关注的数据通信量：优先级2，用户并不等待（关注）要接收的数据，如电子邮件；
- 关注的批量数据通信量：优先级4，当传送批量数据而用户正在等待（关注）接收这个数据（可能会有些延迟时），如FTP和HTTP；
- 交互式通信量：优先级6，需要与用户交互的协议，如TELNET；
- 控制通信量：优先级7，路由选择协议（如OSPF和RIP）以及管理协议（如SNMP）都使用这个优先级。

不可进行拥塞控制的通信量

- 指期望最小延迟的通信量类型，不希望丢弃分组，在大多数情况下也不可能重传；换言之，源端不能使自己适应拥塞，如实时音频和视频；
- 优先级号：8到15；
- 虽然还没有特殊的标准指定给这类数据，但优先级的指定是基于当丢弃一些分组时接收到的数据的质量会有多大的影响来确定的；
- 包含较小的冗余度的数据（如低保真音频和和视频）给予较高优先级（15），包含较大冗余度的数据（如高保真音频或视频）可以给予较低的优先级（8）。

流标号

- 从特定源端向特定目的端发送的分组序列，如果需要路由器的特殊处理，则称为分组流；
- 源地址与流标号的值组合唯一地定义了一个分组流；
- 对路由器来说，一个流是共享某些特性的分组序列，例如经过相同路径，使用相同的资源，具有相同安全性等；
- 支持流标号处理的路由器有一个流标号表，这个表为每一个活动的流标号设置一个项目，每一个项目定义相应的流标号所需的服务；
- 当路由器收到一个分组时，它就从其流标号表中找出在分组中定义的流标号值所对应的项目；
- 但注意：流标号本身并不给流标号表项目提供信息，信息是由其他方法提供的，如逐跳选项或其他协议；

流标号（cont.）

- 在最简单形式中，流标号可用来加速路由器对分组的处理；当路由器收到一个分组时，它不用查找路由表并用路由选择算法确定下一跳的地址，而是可以很容易地在流标号表中找到下一跳的地址；
- 在更复杂的形式中，流标号可用来支持实时音频和视频的传输；特别是数字形式的实时音频或视频，需要高带宽、大缓存、长处理时间等资源，进程可以事先对这些资源进行预留，以保证实时数据不会因资源不够而被延迟；使用实时数据和预留这些资源需要IPv6外的一些其他协议，如实时协议（RTP）和资源预留协议（RSVP）

流标号（cont.2）

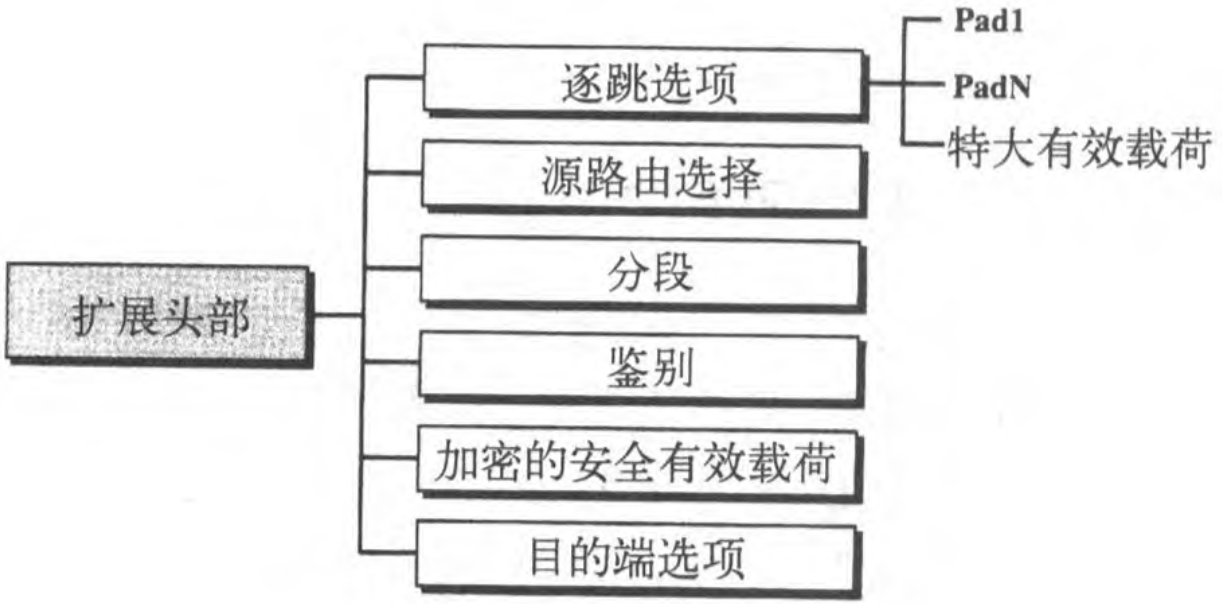
- 为了有效地使用流标号，已定义了三个原则；
  - Ø1.流标号由源主机指定给分组，这个标号是在1到2^{24}-1之间的随机数；当已存在的流仍处于活跃状态时，源端一定不能给新的流重复使用一个用过的流标号；
  - Ø2. 如果主机不支持流标号，它就将这个字段置为0 ；如果路由器不支持流标号，它就简单地忽略它；
  - Ø3. 所有属于同一个流的分组必须具有相同的源地址、相同的目的地址、相同的优先级和相同的选项。

表20.9 IPv4和IPv6分组头部的比较

比较
1. IPv6取消了头部长度字段，因为在此版本中头部长度是固定的
2. IPv6取消了服务类型字段，优先级和流标号字段合在一起取代服务类型字段的功能
3. IPv6取消了总长度字段，替代的是有效载荷长度字段
4. 在IPv6中的基本头部中取消了标识、标记和偏移字段，这些都包含在分段扩展头部中
5. 在IPv6中，将TTL字段称为跳数限制字段
6. 协议字段被替换为下一个头部字段
7. 头部校验和取消了，因为校验和由上层协议提供，因此这一层不需要了
8. IPv4的选项字段在IPv6中被实现为扩展头部

图20.17 扩展头部的类型

- 基本头部的长度是固定的40字节，但是要使IP数据报有更多的功能，在基本头部的后面还可增加多达6个扩展头部；
- 这些头部中的许多都是IPv4的选项；
- 共有6种不同类型的头部



IPv6扩展头部类型介绍

- 逐跳选项：当源端需要将信息传递给数据报经过的所有路由器时，需要使用逐跳选项；定义了三种选项：Pad1、PadN和特大有效载荷（jumbo payload），前两者用来对齐，特大有效载荷选项用来定义比65535字节更大的有效载荷（高达40亿字节）；
- 源路由选择：将IPv4中的两种源路由结合；
- 分段：概念与IPv4一样，但分段发生的地方不同；IPv4中可能发生在源端或路由器，IPv6中只有原始的源端才能进行分段；源端必须使用路径MUT发现技术找出在其路径上的任何网络所支持的最小MTU，然后源端利用得到的这个知识进行分段；
- 鉴别：证实报文发送者并保证数据的完整性；
- 加密的安全有效载荷：提供保密性并能防止窃听；
- 目的端选项：用于当源端需要将信息仅传递给目的端，中间的各路由器不允许读取这些信息。

表20.10 IPv4选项和IPv6选项的比较

比较
1. 在IPv4中的无操作和选项结束选项被替换成IPv6 中的Pad1和PadN选项
2. 在IPv6中，没有记录路由选项，因为它未被使用
3. 在IPv6中，时间戳选项没有实现，因为它未被使用
4. 在IPv6中，源路由器选项称为源路由扩展头部
5. 在IPv4中的基本头部的分段字段已经移到IPv6 中的分段扩展头部
6. 在IPv6中的鉴别扩展头部是新的
7. 在IPv6中，加密的安全有效载荷扩展头部是新的

20-4 IPv4到IPv6的过渡

- 因为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；
- IETF设计了三种策略：双栈技术、隧道技术和头部转换

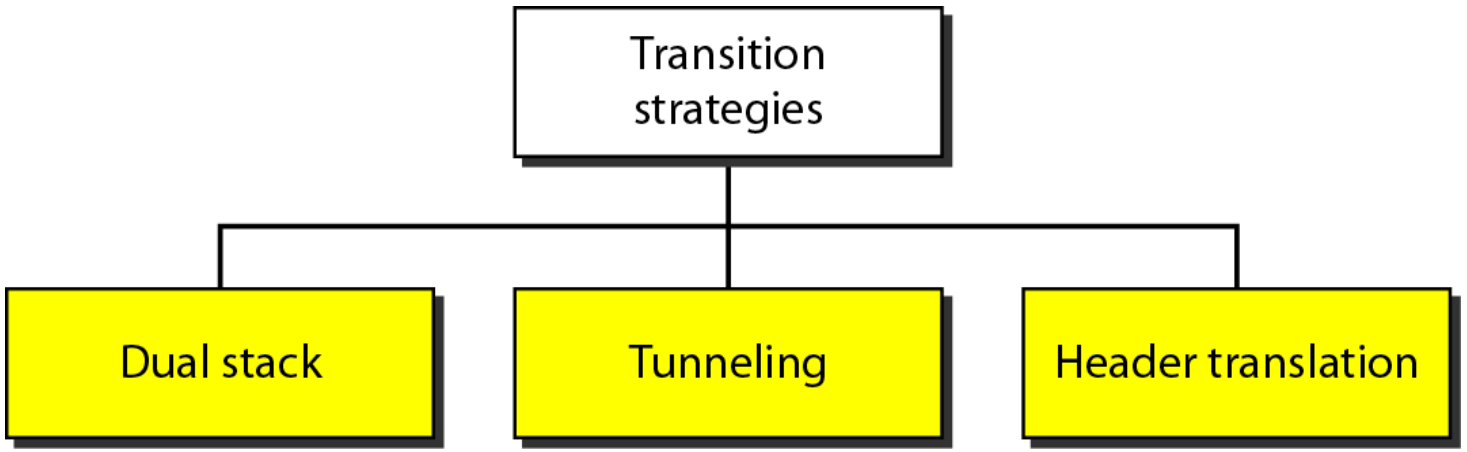


图20.19 双协议栈

- 一个站应同时运行IPv4和IPv6，直到整个因特网使用IPv6；
- 为了确定使用哪个版本，主机要向DNS进行查询；如果DNS返回一个IPv4地址，那么源主机就发送一个IPv4分组，如果返回一个IPv6地址，就发送一个IPv6分组。

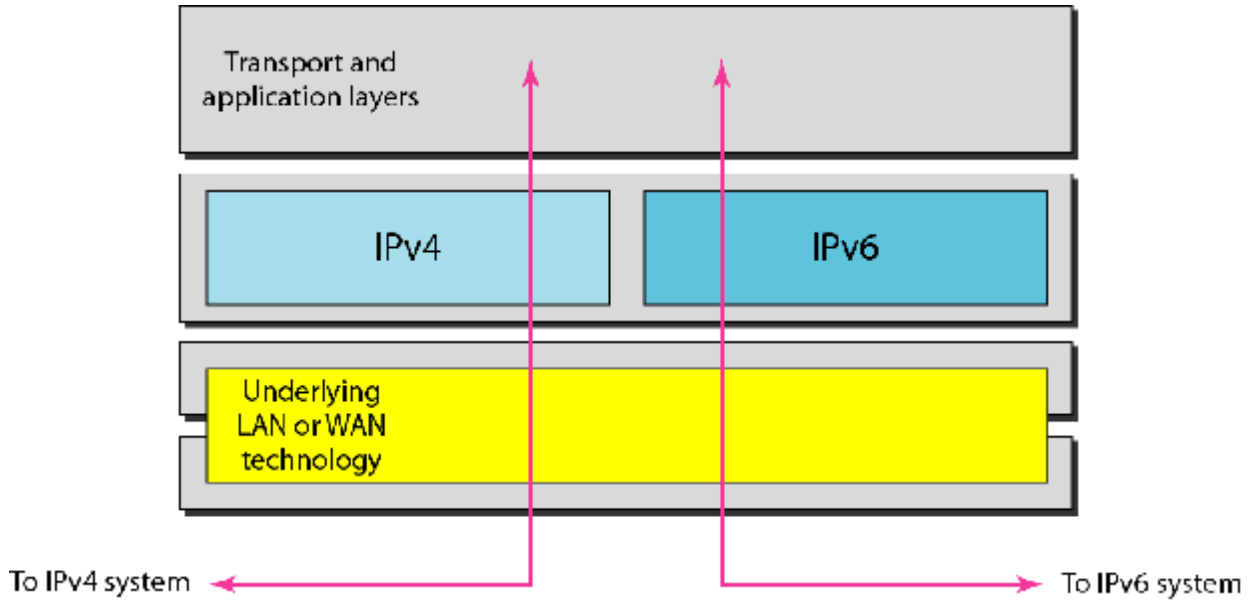


图20.20 隧道策略

p 当两台使用IPv6的计算机要进行相互通信，但其分组要通过使用IPv4的区域时，就要使用隧道策略（反之一样）；

p 要经过IPv4区域，分组必须具有IPv4地址，因此，进入区域时IPv6分组要封装成IPv4分组，而当分组离开区域时再去掉这个封装，这就好像IPv6分组进入隧道一端，而在另一端流出来；

p 为了说明利用IPv4分组携带IPv6分组，其协议的值设置成41。

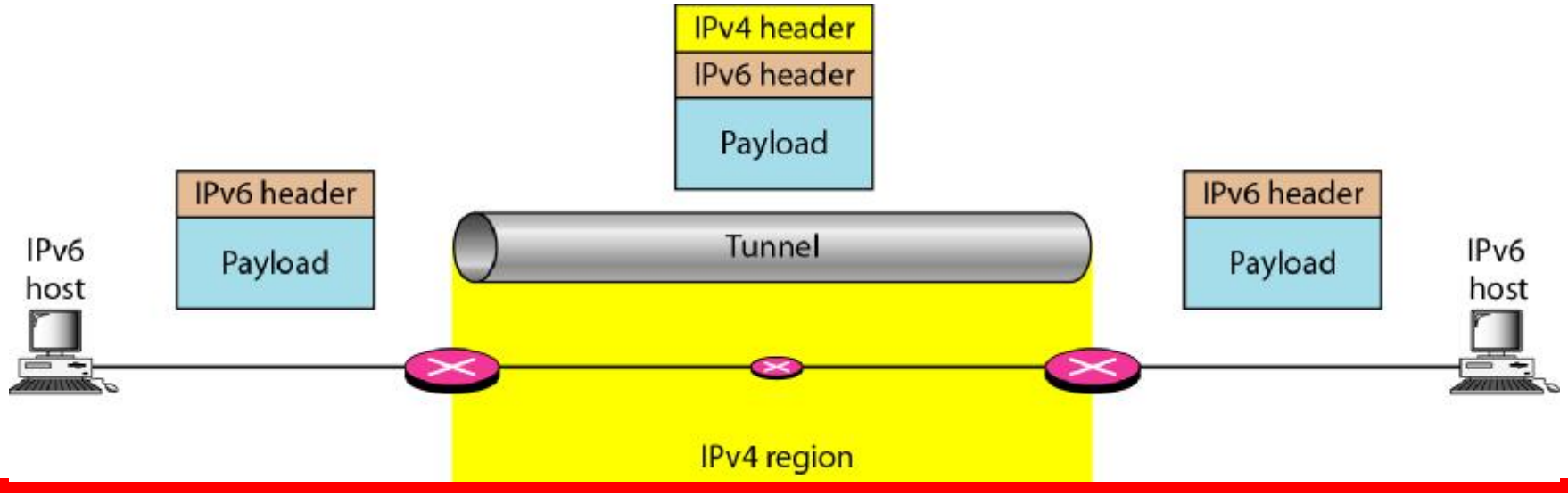


图20.21 头部转换策略

p当因特网中绝大部分已经过渡到IPv6，但一些系统仍然使用IPv4时，需要使用头部转换；

p发送方想使用IPv6，但接收方不能识别IPv6，此时隧道技术无法工作，因为分组必须是IPv4的格式才能被接收方识别；

p在此情况下，头部格式必须通过头部转换彻底改变，IPv6的头部就转换成IPv4的头部（P399图20.21上写错了）。

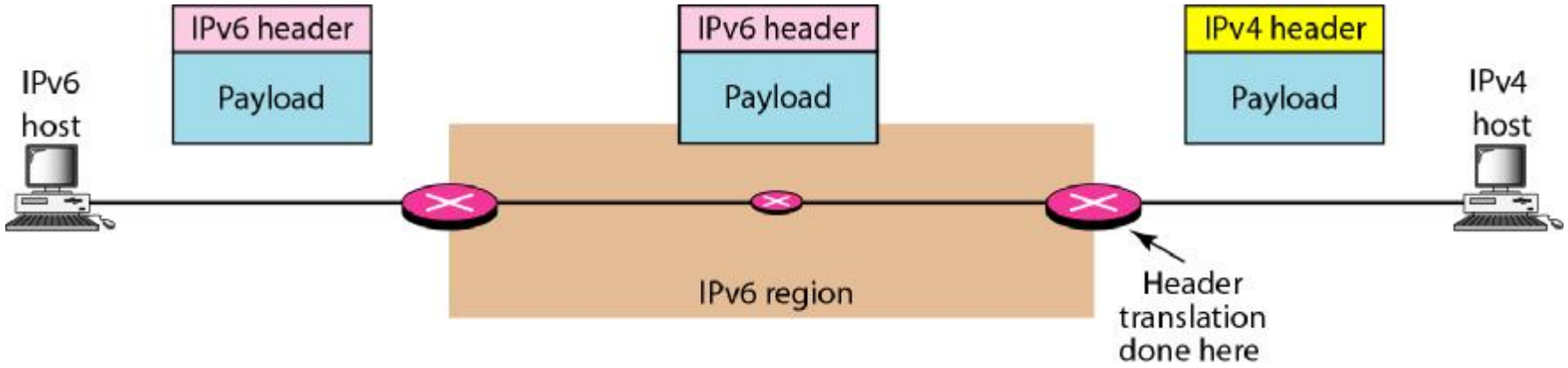


表20.11 IPv6到IPv4的头部转换

头部转换过程
1. 用提取最右边32位的方法，将IPv6被映射的地址改变为IPv4地址
2. 将IPv6优先级字段的值丢弃
3. 将IPv4的服务类型置为0
4. 计算IPv4的校验和，并插入到相应的字段中
5. 忽略IPv6的流标号
6. 兼容的扩展头部要转换成选项，插入到IPv4的头部中
7. 计算出IPv4头部的长度，将其插入到相应的字段中
8. 计算出IPv4分组的总长度，将其插入到相应的字段中