第 5 章

网络层

网络层的功能是将分组从源主机一路送到目的主机。源主机和目的主机可能位于同一个局域网，更大的可能是源主机和目的主机位于完全不同的局域网中；源主机和目的主机可能距离很近，也可能相隔万水千山，不管是什么情形，网络层都能够将分组从源主机穿越中间网络，尽力推送到目的主机；这个过程称为**寻址**。这个中间网络可能很大也可能很小，里面是什么样子？它在寻址中起到什么作用呢？本章首先介绍的**通信网络及提供的服务**，就来探讨这个问题。

要找一个人，得知道人在哪，还得知道这个人的名字。为了找到目的主机，我们也得知道这台主机在哪里和它的标识。本章将介绍 IP 协议，IP 地址用于标识这台主机的位置，通过 IP 地址找到目的主机的过程称为 **IP 寻址**。找到目的主机的目的是要给它发数据，数据被封装成网络层的协议数据单元——**分组**，分组中除了数据之外，还包含了寻址过程（路由）所需要的信息。IP 协议正从 v4 版本向 v6 版本过渡，除了探讨 IPv4，我们也要探讨 IPv6。与 IP 相关的其他内容，比如无类域间路由（CIDR）、网络地址转换（NAT）、动态主机配置协议（DHCP）等都将是本章的学习内容。

IP 分组穿越的中间网络是怎么找到目的主机的呢？源主机和目的主机之间的中间网络，称为**通信子网络**[①]，通信子网络由中间设备和连接中间设备的链路所构成。中间设备是路由器，也可能是网关（路由器也是一种网关），而连接路由器的物理链路可以是铜缆、卫星链路，更多的情形是光纤。IP 分组穿越通信子网络，路由器功不可没，路由器的接力，将 IP 分组逐跳（hop，一跳指一台路由器）推向目的主机。路由器通过查找**路由表**（routing table）完成找路、寻址，而路由表通过运行**路由选择协议**完成建立、更新和维护。

从源主机到目的主机的 IP 分组的传递，所有经过的路由器都“尽力而为”地将分组推向目的主机，服务质量会因网络状况而发生变化，并不能保证满足所有的应用需求。本章将探讨为提供**服务质量**（QoS）保证而提出的模型和方法等。

① 本书的通信子网络，指的是核心骨干网络；与本书 5.2.3 节的子网不是一个概念，请读者注意结合上下文理解。为了区别，本书称通信子网络为通信网络，它的两种类型——数据报子网和虚电路子网，本书分别称为数据报网络和虚电路网络。

本章的主要内容围绕网络层的功能展开，如图 5-1 所示。

第5章　思维导图

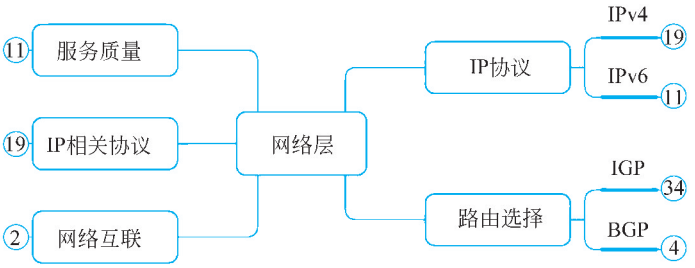


图 5-1　网络层的主要内容组织的上层思维导图

图 5-1 中的数字代表其下的主要知识点个数；读者可扫描二维码查看本章全部知识点的思维导图，并根据需要收起和展开。

5.1　通信网络及提供的服务

20 世纪 60 年代末 70 年代初，分组交换网络从理论落到了实地，不同的机构构建了不同的分组交换网络，将这些网络连接起来，实现更大范围的资源共享（resource sharing）是迫切的需求。但是，这些分组交换网络可能有很多不同，比如分组尺寸不同、错误处理机制不同、编址方式不同等，要把这些不同的分组交换网络连接起来，必须要消灭这些不同，或者说在某层面将其统一起来。在罗伯特·卡恩和温顿·瑟夫共同工作发明出 TCP/IP 早期版本的过程中，有 4 个基本原则起到了至关重要的作用。

（1）**独立自治**：每个网络都是独立自治的，且各有其主，并不因为需要连接而改变自身。

（2）**尽力传输**：基于尽力而为的传输，分组有可能丢失，如果一个分组没有送达，源主机可以重发。

（3）**黑盒连接**：使用黑盒连接不同的分组交换网络，黑盒不记录任何过往分组的信息，也不做任何复杂的错误处理和恢复的事情，尽量保持简单。这些黑盒就是后来的网关和路由器。

（4）**无中控全分布**：在操作层面，不需要全局中央控制，整个连接起来的大网络是全分布的。

网络互联还必须解决以下关键问题。

（1）**重传算法**：丢失的分组可以成功重传，阻止其永久丢失。

（2）“**主机到主机**”管道：多个分组可以独立寻找从源主机到目的主机的路径，只要中间网络允许。

（3）**网关功能**：可以恰当地转发分组，包括翻译 IP 分组头部、处理接口、必要时分割分组为更小的分片等。

（4）**终端处理**：需要终端参与较复杂的处理工作，比如处理错误、计算校验和，如果有分片，需要重组。

（5）**流量控制**：采用主机到主机的流量控制技术，这是比较困难的全局任务。

（6）**全局寻址**：互联后的网络能够全局寻址，每台主机或设备都有一个能够标识它自己的地址。

(7) **接口**：各种不同操作系统的接口。

(8) **其他因素**：比如实现效率、互联性能等，但一开始，这些并不是首先要考虑的。

接下来的几年间，卡恩和瑟夫发明了 TCP/IP 协议栈，这个协议栈在各种操作系统中得以实现，奠定了现代互联网[①]的坚实基础，网络的商用化让网络互联到全球的每个角落。

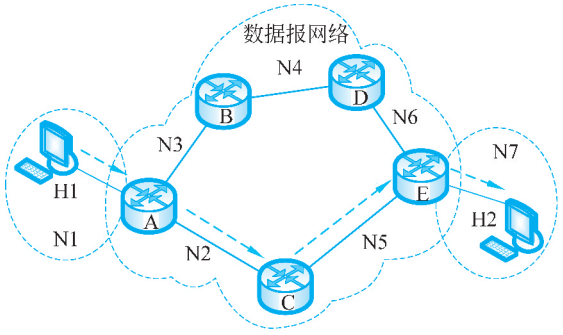
源主机和目的主机之间相隔了一个可大可小的中间网络：通信网络。有两种类型的通信网络——数据报网络和虚电路网络，分别为上层提供了两种服务：无连接的服务和面向连接的服务。

早期的 X.25、帧中继网络是面向连接的，而现在的主流 IP 网络却是无连接的。

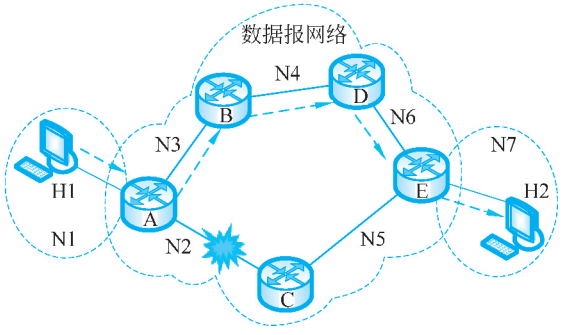
5.1.1 无连接的服务

提供无连接的服务的是数据报网络。如图 5-2(a)是一个数据报网络，它可以是一个互联网服务提供商(ISP)的网络，也可能是若干 ISP 的网络连接起来的大网络。

图 5-2(a)中，当处于网络 N1 中的主机 H1 向处于网络 N7 中的主机 H2 发送数据消息时，它首先将消息按照规范，封装成一个一个的分组，这些分组头部写着目的主机 H2 的地址，它们首先来到路由器 A，因为 A 是源主机 H1 的**默认网关**，是通往外部(相对于内部网络 N1 而言)网络的跳板或全权代理，是整个路径的**第一跳**。



(a) 网络正常时 H1→H2的穿越路径



(b) A 和 C 之间的网络故障时

图 5-2 数据报网络工作情形

① 现代互联网，有别于早期的 ARPANET，虽然 ARPANET 也是分组交换网，但那时还没有采用 TCP/IP。这里的现代互联网，指的是采用了 TCP/IP 协议栈的覆盖全球的大型计算机网络。如无特别说明，本书中的互联网，指的都是现代互联网。

A 收到分组，首先解封装，提取出目的地址，即 H2 的地址，获取其所在的网络 N7，查找自己的路由表（表 5-1），找到去往 N7 对应的下一跳是 C；A 将分组重新封装好，向 C 转发；路由器 C 收到分组后，所做的工作与 A 一样，解封装，提取目的地址 H2，获取目的网络 N7，查找自己的路由表（表 5-1），找到去往 N7 对应的下一跳是 E；A 将分组重新封装好，向 E 转发；路由器 E 收到分组后，所作的工作与 A 一样，解封装，提取目的地址 H2，获取目的网络 N7，查找自己的路由表（表 5-1），发现 N7 是自己的直连网络，将分组重新封装，从 N7 所在的接口将分组转出，分组在 N7 网络（子网）内通过物理地址定位（即 MAC 寻址）到目的主机 H2。

表 5-1　路由器 A、C、E 各自的路由表

A 的路由表项		C 的路由表项		E 的路由表项	
目的网络	下一跳	目的网络	下一跳	目的网络	下一跳
N1	直连	N1	A	N1	C
N2	直连	N2	直连	N2	C
N3	直连	N3	A	N3	D
N4	B	N4	A	N4	D
N5	C	N5	直连	N5	直连
N6	B	N6	E	N6	直连
N7	C	N7	E	N7	直连

假如主机 H1 发出的分组有 100 个，当前 50 个分组通过路径 H1→A→C→E→H2 到达目的主机后，突然一个响雷打坏了 A 和 C 间的连接，或者 AC 间的线路被野蛮挖断了，路由器通过路由协议感受到这个灾难带来的变化：A 和 C 之间的路径不可用了，并反映到路由表中。

表 5-2 是灾难发生前后，路由器 A 的路由表发生的变化，灾难前，到目的网络 N5 和 N7 的下一跳都是 C；灾难发生之后，经过 C 的路不通了，所以，绕开 A→C 段，从 A→B 段走。灾难后，A 的路由表中显示到网络 N5 和 N7，下一跳已经改为 B 了。其余感知到灾难的路由器都更新了自己的路由表，这些路由器合力的结果是：后续的分组传输路径已经变为 H1→A→B→D→E→H2 了，如图 5-2(b)所示。

表 5-2　路由器 A 的路由表在灾难前后发生的变化

A 的路由表项（灾难前）		A 的路由表项（灾难后）	
目的网络	下一跳	目的网络	下一跳
N4	B	N4	B
N5	C	N5	B
N6	B	N6	B
N7	C	N7	B

数据报网络提供的数据分组传输服务，无须在传输前搭起通路，每个分组独自找路，这个特性让网络不怕灾难或故障，分组可以灵活地绕开故障点；整个从源主机到目的主机的寻址过程依靠路由器之间的接力，一跳一跳地靠近目的网络，每跳的转发决

策都离不开路由表，而路由表的建立、更新和维护是靠路由选择协议进行的，5.5 节将学习路由选择协议的知识。

从上面的数据报网络提供无连接的数据分组传输服务例子中，我们可以总结出无连接的服务的主要技术特点如下。

（1）**无连接**：数据分组传输无须建立连接，不管是虚连接还是物理连接。

（2）**独立寻径**：每个分组都携带目的地址，可以独立寻找达到目的网络的路径。

（3）**乱序到达**：先发的分组可能因为重传或走了一条慢的路径而后到，后发的分组也可能因为走了捷径而先到。

（4）**重组**：目的主机需要对收到的分组进行重新排序。

（5）**无状态**：路由器的工作方式简单，无须保留过往分组的状态，只是尽力而为地转发。

（6）**抗毁性**：在一个四通八达的数据报网络中，单点故障，甚至多点故障，都不会让网络瘫痪，分组可以在尚通达的网络中自由寻路。

（7）**灵活分片**：路由器可能承担分片的任务，当它收到一个超过转出网络承载能力的大尺寸分组时；当然，当条件不允许时，路由器可以拒绝分片。

（8）**服务质量难以保证**：在数据报网络中进行拥塞控制或服务质量保证，非常困难，因为这些自由的分组在网络中的路径并不确定，无法预知。

可见，无连接的数据分组传输有优点，也有缺点，但 IP 分组传输作为一个最成功的例子，“战胜”了曾经风光无限的公共交换电话网络（提供面向连接的语音通信），彰显了无连接的服务的魅力。

5.1.2 面向连接的服务

顾名思义，面向连接的服务，是在分组传输之前，搭建一条连接，从源主机到目的主机的分组只需要沿着这条连接就可以穿越整个虚电路网络，到达目的主机。

图 5-3 中，源主机和目的主机之间是一个虚电路网络，提供面向连接的分组传输服务。但源主机 H1 要向主机 H2 传输数据分组时，H1 和 H2 之间首先协商出一条虚连接（虚电路），如果路由器参加协商并可以为双方提供通信服务而成为虚连接中的一段，则路由器在自己的表中做记录。

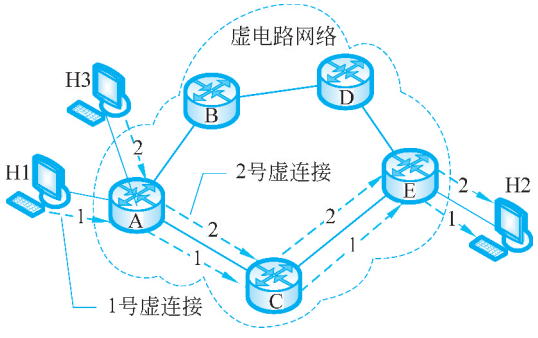


图 5-3 虚电路网络工作情形

经过协商之后，H1 和 H2 之间搭建起了 1 号虚连接，沿途的路由器记录下了虚连接的信息，如表 5-3 所示。表中的“入境”和“出境”表示站在本路由器的角度，分别对

应分组从哪里进来和出去，“入连接号”和“出连接号”表示协商的连接号，一台路由器的入连接号和出连接号可能不相同。举个例子，如果路由器 A 上挂有另外一台主机 H3，也想向 H2 发送分组，协商建立的 2 号虚连接如图 5-3 所示。

表 5-3　路由器内部的虚连接表

路由器 A 的虚连接表				路由器 C 的虚连接表				路由器 E 的虚连接表			
入境	入连接号	出境	出连接号	入境	入连接号	出境	出连接号	入境	入连接号	出境	出连接号
H1	1	C	1	A	1	E	1	C	1	H2	1
H3	1	C	2	A	2	E	2	C	2	H2	2

图 5-3 中，1 号虚连接和 2 号虚连接所走的线路在虚电路网络中完全重合，区分它们的方法就是**连接号**；连接号具有本地属性，也就是在本地管理和生效。如果 H3-H2 的虚连接是 H3 的第一条虚连接，路由器 A 赋给该连接的入境为 H3 且入连接号为 1；A 发现这条连接出境应该是 C，但出连接号“1”已经被占用，所以，H3-H2 连接的出连接号被赋值“2”。当 2 号虚连接搭建成功开始传输分组时，H3 发出的分组中携带了连接号“1”，当分组到达路由器 A 时，A 打开分组，并查找它的虚连接表，来自 H3 且连接号为“1”的入境分组，应该从 C 转出，且出境分组的连接号被改写为“2”。所以，有的教材把这个过程称为**标签交换**(label switching)，这里的标签指的就是连接号。

从上面的虚电路网络提供面向连接的数据分组传输服务例子中，我们可以总结出面向连接的服务的主要技术特点如下。

(1) **按路径传输**：数据分组传输之前，需要建立连接；所以，每个分组不需要独立寻路，分组一个接一个沿着虚连接从源主机到达目的主机。

(2) **不需要地址寻径**：每个分组携带连接号，而不需要携带目的主机的地址。

(3) **按序到达**：先发的先到，后发的后到。

(4) **连接损坏**：分组传输中的连接如果发生故障，连接中断，则需要重新搭建虚连接。

(5) **有状态**：路由器工作较为复杂，要保存和维护所有虚连接的状态信息。

(6) **服务质量保证**：可以容易地进行拥塞控制和服务质量保证。

面向连接的服务在公共交换电话网络中广为使用，也在早期的广域网技术 X.25、帧中继中使用。面向连接的服务并没有在现代互联网中大行其道，但人们从没有放弃在适当时再想起它，比如，20 世纪末 21 世纪初出现的多协议标签交换(MPLS，5.6.3 节)，旨在进行高速交换，且有一定的服务质量保证。

5.2　IPv4 协议

互联网协议(IP)统一了数据传输的分组格式和地址标识，为分组的路由提供了必需的信息。到目前为止，IP 协议有两个在用的版本—— IPv4 和 IPv6，本节将围绕 IPv4 分组和 IPv4 地址展开；5.4 节将围绕 IPv6 分组和 IPv6 地址展开。当我们提到 IP 时，如果没有加版本信息，通常指的是 IPv4，比如 IP 分组指的是 IPv4 分组，IP 地址

指的是 IPv4 地址。

5.2.1 IPv4 分组

从发送方的网络层来看，网络层实体从传输层（上一层）接收数据段，加上分组头部，形成分组后，再传给数据链路层（下一层）形成数据帧，这是封装中的一个环节。传输层的数据段、网络层的分组和数据链路层的数据帧之间的关系如图 5-4 所示。

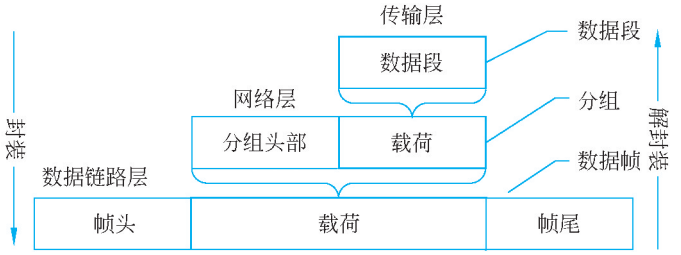


图 5-4 数据段、分组与数据帧之间的关系

接收方的网络层，从数据链路层接收到分组，提取出其中的载荷——数据段，送给传输层，这是解封装中的一个环节。载荷指的是协议数据单元中除了头部、尾部等的开销之外的数据，有时也称为净数据，比如分组中的载荷是传输层的数据段，而数据帧中的载荷就是网络层的分组。

1. 分组头部各字段含义

IP 分组包括**头部**和**载荷**两大部分，头部至少包括 14 个字段，如果有选项，头部包括 15 个字段，如图 5-5 所示，图中的一行占 32 位，合 4 字节。

下面逐一介绍各字段的名称和含义。

版本：第①个字段，长 4 位，表示 IP 的版本。目前只有两个值被广泛使用，即 0100 和 0110，分别代表当前的分组是 IPv4 分组或 IPv6 分组。

头部长度：第②个字段，长 4 位，表示 IP 分组头部的长度，其值可以从 0101 到 1111 变化（对应的十进制数为 5～15），值得注意的是，头部长度的单位是 32 位，即 4 字节，所以，分组头部的长度可在 20～60 字节变化。当头部不包括选项时，头部长度取值 0101，对应 5×4 字节＝20 字节；如果头部中有选项，必须保证选项长度是 4 字节的整数倍，最长选项长 40 字节，因为头部长度最大取值 1111，对应 15×4 字节＝60 字节，扣除固定头部 20 字节就是最长选项的长度。

服务类型：第③个字段[①]，长 8 位，表示 IP 分组的优先级别或重要程度，为区分服务提供被怎样区别处理的依据。

其中的最左边 6 位定义了不同的分组**优先级**（定义了 0～7 共 8 个优先级别）和**丢弃级别**，当路由器处理分组时，可以根据这个字段做出不一样的处理策略，以提供不同的服务级别。比如，网络控制（network control）的分组可以被赋予高优先级别（0b111＝0d7），一般的分组都是普通级别[BE（best effort）级，0b000]，当某台路由器资源紧张

① 在最初的 IPv4 标准（RFC 791）中，该字段称为服务类型（Type of Service，ToS），用以指示所需要的服务质量（QoS）参数，其中最左边 3 位指明了不同的优先级（precedence），接下来的三位分别是时延、吞吐量、可靠性的标记位，最后 2 位未使用；1998 年，该字段被称为区分服务（RFC 2474），被重新定义。

(拥塞)而不得不丢包时,首先丢弃的是普通分组,而优先转发处理网络控制分组;该字段的最右边 2 位当时未定义,即未被使用(Currently Unused,CU)。更多区分服务的细节可参考 5.6.2 节。

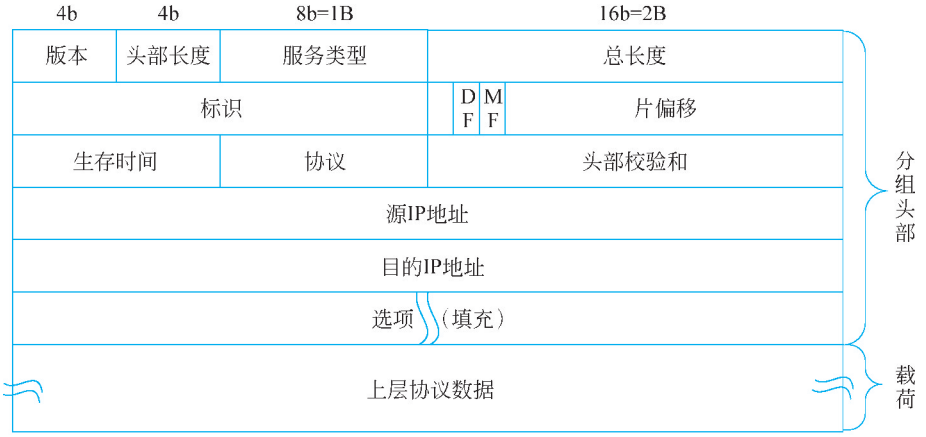


图 5-5　IPv4 分组的格式

总长度:第④个字段,长 16 位,表示整个 IP 分组(头部和载荷)的总长度,单位是字节,分组长度的上限是 $2^{16}-1=65\ 535$ 字节。一方面,随着技术的进步,传输速度越来越快,我们希望打破这个上限,更长的分组可以获得适配的高效率;另一方面,分组的大小还受制于下层网络的承载能力。承载能力可用最大传输单元(Maximum Transfer Unit,MTU)量度;最常见的以太网 MTU 为 1500 字节。所以,我们经常见到分组长度是 1500 字节或以下,当分组长度超过网络的 MTU 时,分组被分片,每个分片的长度小于等于对应的 MTU。

所有的互联网主机必须可以接收总长度为 576 字节的分组[①],不管它是完整分组还是分片。

标识:第⑤个字段,长 16 位,用来标识分组的序号。由发送方维护,每发出一个数据分组,标识增加 1。该字段和第⑦、⑧、⑨共 4 个字段一起,为分片提供信息。本节稍后将专门讨论分片及分片表示。

标记位:⑥、⑦和⑧这 3 个字段合起来称为标记,共长 3 位。

⑥标记位未使用,须为 0。

⑦标记位称为**不要分片**(Don't Fragment,DF),取值 0 表明允许分片,取值 1 表明不允许分片。

⑧标记位称为**更多分片**(More Fragment,MF),取值 0 表明这是最后一个分片,取值 1 表明这不是最后一个分片,即后面还有更多的分片。⑦和⑧两个字段表示分片标记位,稍后探讨其使用场景和方法。

片偏移:第⑨个字段,长 13 位,单位是 8 字节。其值指明了当前分片在原分组中的起始位置,最小值 0 指明当前分片是第一个分片,即起始分片;最大值为 8191($2^{13}-1$),

① 源自 IPv4 标准(RFC 791),除非发送方确认接收方可以接收超过 576 字节的分组,发送方发送的分组总长度才会超过 576 字节。之所以选择 576 字节,是因为上层的数据段长度通常为 512 字节或以下,加上分组最长头部 60 字节,和 4 字节的余量,正好 576 字节。

说明一个分组最多分为 8192 个分片。

生存时间：第⑩个字段，常称为 TTL(Time To Live)，长 8 位，取值范围是 0～255，单位是 s。该字段的值规定了当前分组在互联网中存在的时间，每经过一台路由器，此值都被改写，减去在此路由器处理的时间(如果处理时间小于 1s，按照 1s 计)；当该值被改写为 0 时，即使还没有找到目的网络，当前分组也会被销毁。这个字段的设计，防止了一个分组在网络中无限循环而成为幽灵。

随着路由器处理分组的速度变得越来越快，远远小于 1s，每台路由器会把转出分组的 TTL-1，所以，这个字段的单位逐渐从 s 演变成了**跳数**(hops)，即路由器的台数，表明分组最多可以在网络中经过的路由器台数。一个分组的 TTL 值，每过一台路由器就减 1，绝大多数分组的 TTL 在未减到 0 时就到达了目的主机；对于 TTL=0，但还未到达目的网络的分组，可能遭遇了环路由等异常情况，路由器将其销毁，并使用 ICMP 消息(参考 5.3.2 节)通知源发生的事情，避免分组无休止地在网络中转悠。一般来说，Windows 操作系统产生的分组，其 TTL 初值为 **128**，而 Linux 操作系统产生的分组，其 TTL 初值为 **64**，这两个值足够保证分组到达它的目的，或者迷路后耗尽 TTL 而被销毁。

协议：第⑪个字段，长 8 位，表示网络层之上的传输层使用的协议。该字段的值代表了当前分组中搭载的数据类型，即使用的上层协议数据单元，指示接收方应该采用什么上层协议处理分组中的载荷。比如，如果此字段取值 6，表示当前分组中的载荷搭载了 TCP 数据段，接收方应该使用 TCP 处理分组中的 TCP 数据段。表 5-4 是协议字段的著名取值及含义。

表 5-4 协议字段的著名取值及含义

协议字段值	上层协议	备　注
1	ICMP	互联网控制消息协议，用以报告 IP 分组传输的错误和测试网络。参考 5.3.2 节
2	IGMP	互联网组管理协议，用于组播成员管理。参考 5.5.5 节
6	TCP	传输控制协议，用以提供可靠的数据段传输。参考 6.3 节
9	IGP	内部网关协议，用以支持私人定制的内部网关协议。参考 5.5.1 节
17	UDP	用户数据报协议，用以提供简洁高效的数据段传输。参考 6.2 节
41	IPv6	应用于 IPv4 向 IPv6 的过渡技术，表示载荷搭载的是 IPv6 分组。参考 5.4 节
89	OSPF	开放最短路径优先协议，最流行的内部网关协议。参考 5.5 节

头部校验和：第⑫个字段，长 16 位，用以检查 IP 分组头部有无发生传输错误。头部校验和采用分组头部的数据计算**互联网校验和**(具体计算参考 3.3.1 节)，发送方计算，接收方检查，确保分组头部数据没有错误，因为头部携带了诸如地址之类的重要信息。分组经过的每台路由器，都会修改头部中的 TTL 等信息，则需要重新计算一次头部校验和，这就产生了不可忽视的计算开销，因此 IPv6 分组已经取消了该字段。

源 IP 地址：第⑬个字段，长 32 位，即 4 字节，用以标识当前分组的发出接口，即从哪儿来。

目的 IP 地址：第⑭个字段，长 32 位，即 4 字节，用以标识当前分组的目的接口，

即到哪儿去。32 位的 IP 地址将在 5.2.2 节介绍。

选项：第⑮个字段，其长度必须是 4 字节的整数倍，如果不是，需要用 0 **填充**；选项的长度上限是 40 字节。选项可用于完成一些特殊用途，比如安全、时间戳、源路由等。

2. 分组中的选项

IPv4 分组头部包括 14 个字段共 20 字节的固定部分，选项为没有表示在固定部分的其他内容提供了一个空间。当技术进步或者有额外的需求时，可以使用选项，而不必重新设计分组。是否使用选项由发送方决定，但是要求其他互联网节点能够解析这些选项。

在 RFC 791 中，对选项做了详细的规定，有 2 个 1 字节的选项：**无操作**(no-operation)和**选项列表结束**(end of option list)。

8 位无操作选项用于作指示，值为 00000001，用于在选项之间作填充，以保证选项是在 4 字节的边界；8 位选项列表结束选项表示为 00000000，用作所有列表之后的填充，只能出现一次。除了这两个 1 字节选项之外，其余的选项都是多字节选项，常用“类型-长度-值(TLV)”的形式表达，主要的选项包括：

(1) **安全**：用以说明当前分组的保密级别，可以根据此值绕开一些路由器。在全分布、独立寻径的网络中，安全选项的初衷几乎无法实现。

(2) **松散源路由**：源主机在分组中填写的这个选项，是从源主机到目的主机之间经过的路由器列表，分组需要按照这个列表顺序到达列表中的路由器，但是允许中途经过其他路由器。好像我们出去自驾旅游，开启 GPS 导航，设定好中途经过哪些景点，跟着导航走就可以了。

(3) **严格源路由**：源主机规定了到达目的主机的严格路径，即要求经过的路由器列表，且不允许经过其他路由器。(2)和(3)选项可以通过源路由的设定，有意识地避开某些路由器，或者在路由表被破坏的紧急情况下，有一条有效的路径传输分组。

(4) **时间戳**：记录每台路由器处理分组的全球通用时间(UT)，从午夜开始计，单位为 ms。可用于追踪路由器行为，为管理提供依据。

(5) **流标识**：在不支持流(stream)概念的网络中传输时，提供一种对流的支持方法。我们很快会发现，这个几乎失传的选项，却出现在了 IPv6 分组头部。

这些选项是为了一些在分组固定头部未能展现的功能和目的而设计的，但这些设计几乎不被关心和使用，无疑当初的设计是失败的，因此在 IPv6 分组中，不再存在选项。

3. 分片及分片表示

源主机和目的主机之间可能穿越不同 ISP 构建的异构网络，这些网络链路的承载能力并不相同，用最大传输单元表示。**最大传输单元**(MTU)指的是数据帧中载荷的最大长度。数据帧的载荷是分组，所以，MTU 指的也是分组的总长度的最大值，如图 5-6 所示。

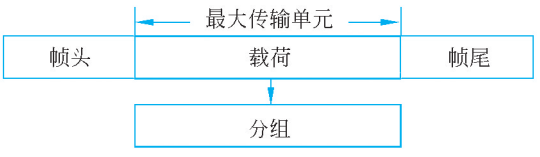


图 5-6　最大传输单元

读者已经在第 4 章知悉以太网数据帧的最大长度为 1518 字节，去掉帧头、帧尾，剩下的载荷(净数据)最长为 1500 字节，这就是以太网的 MTU。常见网络的 MTU 如