第三部分　IP和相关协议

本部分内容包括：

- IP协议家族
- IPv6

第9章　IP协议家族

作者：Mark A. Sportack

本章内容包括：

- TCP/IP模型
- 理解IP协议
- 理解传输控制协议(TCP)
- 理解用户数据报协议(UDP)

TCP/IP已成为描述基于IP通信的代名词。除了流行之外，很少人知道它实质是指整个的协议家族，每个协议都有自己的功能和限制。这一章讨论 IP协议家族内各种协议的结构、功能和使用。

9.1　TCP/IP模型

和其他网络协议一样， TCP/IP有自己的参考模型用于描述各层的功能。然而和绝大多数其他协议不一样， TCP/IP是在协议组件本身开发之后才有了 TCP/IP模型。因此， TCP/IP模型没有起到引导协议的开发作用， TCP/IP参考模型和OSI参考模型的比较见图9-1。

OSI参考模型层描述	OSI层号	TCP/IP层描述
应用层	7	进程/应用层
表示层	6	
会话层	5	主机到主机层
传输层	5	
网络层	3	网际层
数据链路层	2	网络访问层
物理层	1	

图9-1　TCP/IP模型与OSI模型之比较

从图9-1可以看出，TCP/IP参考模型实现了OSI模型中的所有相同功能。重要的不同之处是它们二者层的粒度不同。OSI模型对层的划分更精确，而TCP/IP模型使用更宽的层定义。

9.1.1 解剖TCP/IP模型

TCP/IP协议栈包括四个功能层：进程/应用层、主机到主机层、网际层及网络访问层。这四层大致相对于OSI参考模型中的七层。

1. 进程/应用层

应用层协议提供远程访问和资源共享。读者熟悉的应用包括 Telnet、FTP、SMTP、HTTP，很多其他应用程序驻留并运行在此层，并且依赖于底层的功能。相似的，需要在 IP网络上要求通信的任何应用(包括用户自己开发的和在商店买来的软件)也在模型的这一层中描述。

2. 主机到主机层

IP的主机到主机层大致对应于 OSI参考模型的会话层和传输层。这一层支持的功能包括：为了在网络中传输对应用数据进行分段，执行数学检查来保证所收数据的完整性，为多个应用同时传输数据多路复用数据流(传输和接收)。这意味着主机到主机层能识别特殊应用，对乱序收到的数据进行重新排序。

当前的主机到主机层包括两个协议实体：传输控制协议 (TCP)和用户数据报协议(UDP)。另一个协议正在定义中，这个协议针对于不断增长的面向事务的需要。这个协议称为事务 /事务控制协议(Transaction/Transmission Control Protocol，T/TCP)。

3. 网际层

IPv4的网际层由在两个主机之间通信所必须的协议和过程组成。这意味着数据报文必须是可路由的。网际层(IP)负责数据报文路由。

网际层也必须支持其他的路由管理功能，它必须提供第二层地址到第三层地址的解析及反向解析。这些功能由一对一针对于IP的协议提供，这在第5章中讨论过。

网际层必须支持路由和路由管理功能。这些功能由外部对等协议提供，称这些协议为路由协议。这些协议包括内部网关协议 (IGP)、外部网关协议(EGP)，它们标识为对等的这一点很重要，因为它们驻留在网络层中，但却不是 IP协议组件与生俱来的部分。实际上，许多路由协议能够在多路由协议地址结构中发现、计算路由。用于其他地址结构的路由协议例子包括IPX和AppleTalk。

4. 网络访问层

网络访问层包括用于物理连接、传输的所有功能。 OSI模型把这一层功能分为两层：物理层和数据链路层。由于在同名协议之后创建， TCP/IP参考模型把两层合在一起，是因为各种IP协议中止于网际层。IP假设所有底层功能由局域网或串口连接提供。

9.1.2 协议组件

虽然一般标识为“ TCP/IP ”，但实质上在IP协议组件内有好几个不同的协议。包括：

- IP —— 网际层协议。
- TCP —— 可靠的主机到主机层协议。
- UDP —— 尽力转发的主机到主机层协议。
- ICMP —— 在IP网络内为控制、测试、管理功能而设计的多层协议。

各种ICMP协议从主机到主机层延伸至进程/应用层，这些协议之间的关系如图9-2所示。

注意 驻留于进程/应用层中的应用(如Telnet、FTP和许多其他应用)必须认为是IP协议组件与生俱来的组成部分。然而，由于这些是应用而不是协议，因此它们不在本章中讨论。

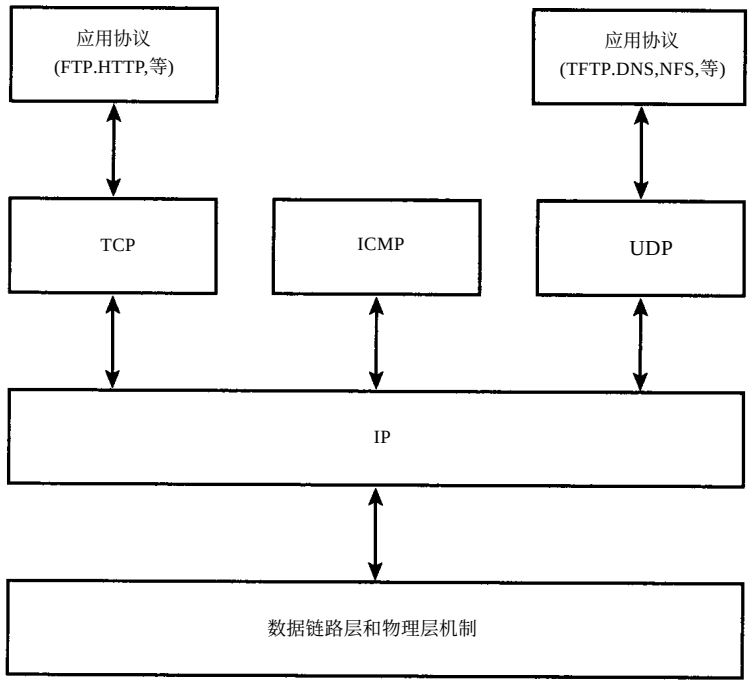

图9-2 TCP/IP实际上是一个相关协议组，而不仅仅是一个协议

9.2 理解网际协议(IP)

IP协议已经成为世界上最重要的网际协议。因为 IP协议的开放性，所以其他的如 OSI、AppleTalk，甚至IPX最终会被IP淘汰。IP的功能由IP头结构中的数据定义。IP头结构及其功能，最被由一系列RFC文档和IETF创建时公开发表的一系列文档定义。 1981年9月出版的RFC791，是今天IP版本的基础文档。

IP一直在演进，这要归功于IETF的不懈努力。许多新的特性和功能在后续RFC文档中得到扩充，然而所有这些都建造在RFC791基础之上。从结构角度讲，现在的IP版本是4。新的版本6几近完成。但只有IPv4是当前的标准且被广泛接受。若想获得IPv6的知识，可参考第10章。

9.2.1 IPv4结构

图9-3示出了IP头结构，以及其中各域的大小。 IP头有以下各域：

- 版本 —— IP头中前四位标识了IP的操作版本，比如版本4或版本6。
- Internet头长度 —— 头中下面4位包括头长度，以32位为单位表示。
- 服务类型 —— 下面的一个字节包括一系列标志，这些标志能保证优先级 (相对于其他IP报

文的绝对优先级)、延时、吞吐量以及报文数的可靠性参数。优先级标志3位长，而延时，吞吐量和可靠性标志每个1位长。剩下的两位保留为将来之用。

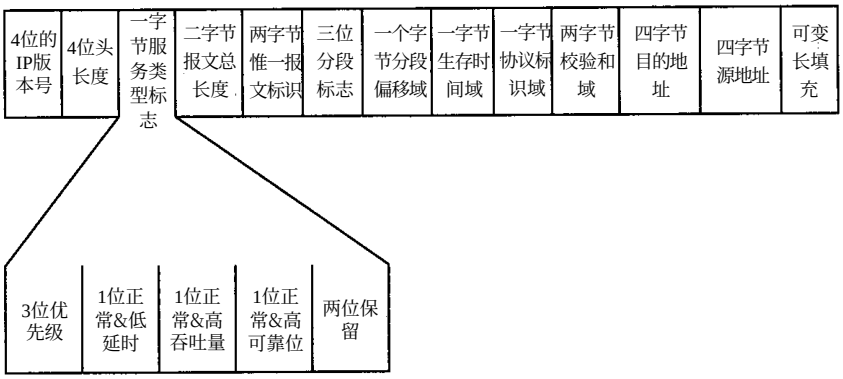

图9-3 IP头结构显示了许多IP支持的功能域

- 总长度(Total Length) —— 报文总长度，16位域，长度以字节为单位。有效值范围最大至65 535个字节。
- 标识(Identifier) —— 每个IP报文被赋予一个惟一的16位标识，用于标识数据报的分段。
- 分段标志(Fragmentation Flag) —— 下一个域包括3个1位标志，标识报文是否允许被分段和是否使用了这些域。第一位保留并总设为 0，第二位标识报文能否被分段。如果这位等于0，说明内容可以被分段。如果等于 1，它就不能被分段。第三位只有在第二位为 0 时才有意义。如果这一位等于0(数据可分成多个报文)，这一位标识此报文是否是这一系列分段的最后一个，或者接收应用程序是否还希望有更多的段。 0指示报文是最后一个。
- 分段偏移(Fragment Offest) —— 8位的域指出分段报文相对于整个报文开始处的偏移。这个值以64位为单位递增。
- 生存时间(TTL) —— IP报文不允许在广域网中永久漫游。它必须限制在一定的 TTL内。8位的TTL在经过每一跳时加1。在到达它的最大限制之后，报文就被认为是不可转发的。之后产生一个ICMP报文并发回源机器，不可转发的报文被丢弃。
- 协议 —— 8位域指示IP头之后的协议，如VINES、TCP、UDP等。
- 校验和(checksum) —— 校验和是16位的错误检测域。目的机、网络中的每个网关要重新计算报文头的校验和，就如同源机器所做的一样。如果数据没有被改动过，两个计算结果应该是一样的。这个域也通知目的主机所接收的数据的量。
- 源IP地址 —— 源计算机的IP地址。
- 目的IP地址 —— 目的计算机的IP地址。
- 填充 —— 为了保证IP头长度是32位的整数倍，要填充额外的0。

这些头域说明IPv4的网际层是无连接的：网络中的转发设备可以自由决定通过网络的报文的理想转发路径。它也不提供任何上层协议如 TCP所提供的应答、流控、序化功能。 IP也不能用于引导IP报文中的数据到正确的目的应用程序。这些功能留给上层协议，如 TCP和UDP。

9.2.2 IP做什么

IP报文头中含有使一些重要网络功能成为可能的所有必要信息，包括：

- 寻址和路由
- 分段和重组
- 传输过程中数据损坏检测和更正

1. 寻址和路由

IP最明显的一个功能是能使报文送到特定目的地。连接源和目的地网络中的路由器和交换机使用目的IP地址确定经过网络的最优路径。

相似的，IP报文也包括源机器地址。源地址的出现是因为目的机可能会和源机通信。

2. 分段和重组

有时应用数据的一段不能完全包括在一个 IP报文中；它们必须分段成两个或更多的报文。当分段发生时，IP必须能重组报文(不管有多少个报文要到达其目的地)。

重要的一点是源和目的机必须理解，遵守完全相同的分段数据过程。否则，重组为了报文转发而分成多个段的过程将是不可能的。数据恢复到源机器上的相同格式时，传输数据就被成功重组了。IP头中的分段标志标识分段的数据片。

注意 重组分段的数据和乱序帧经重排序到达的数据是非常不同的。重新排序是TCP的功能。

3. 损坏报文补偿

IP的最后一个主要功能是检测和补偿在传输过程中遭到破坏或丢失的报文。有许多方式可以让一个报文被破坏：无线电频率干扰 (RFI)和电磁干扰(EMI)是两种比较显然的干扰因素。报文以与源机创建时不同的位模式到达目的机时，就认为报文被破坏了。

有许多原因可造成报文丢失。网络拥塞会导致报文 TTL超时，检测到报文TTL超时的路由器会简单地把报文丢弃。另一种情况是，报文遭到 EMI或RFI干扰，可能使头信息变得没有意义。在这种情况下，报文也将被丢弃。

当报文不可能转发或不可用时，路由器必须通知源机。 IP头中包含源机器的 IP地址使通知源机器成为可能。虽然 IP不包括重传机制，但通知源主机可能会导致重传，因此通知源主机起着重要作用。

4. IP结论

尽管有这些功能，但必须承认 IP仅是一个网际协议。为了使其发挥作用，必须和传输协议(OSI中第四层)及数据链路层协议(OSI参考模型中第二层)一起工作。虽然本书不讨论数据链路层结构，但本章的剩余部分会讨论两个依赖于 IP互连的传输协议。它们是TCP和UDP。

9.3 理解传输控制协议(TCP)

TCP是传输层协议(OSI参考模型中第四层)，它使用IP，提供可靠的应用数据传输。 TCP在两个或多个主机之间建立面向链接的通信。 TCP支持多数据流操作，提供流控和错误控制，甚至完成对乱序到达报文的重新排序。传输控制协议也是通过一系列公开出版的 IETF文档进行开发的。这个反复的开发过程在 1981年9月出版RFC793时到达顶点，同 RFC791 IP一起，RFC793 TCP在过去的18年中不断得到扩充，但是这项工作一直没有全部完成。因此 RFC的内容只保留TCP的核心内容。

9.3.1 TCP头结构

和IP一样，TCP的功能受限于其头中携带的信息。因此理解 TCP的机制和功能需要了解

TCP头中的内容。图9-4显示了TCP头结构，和其中各域的大小。

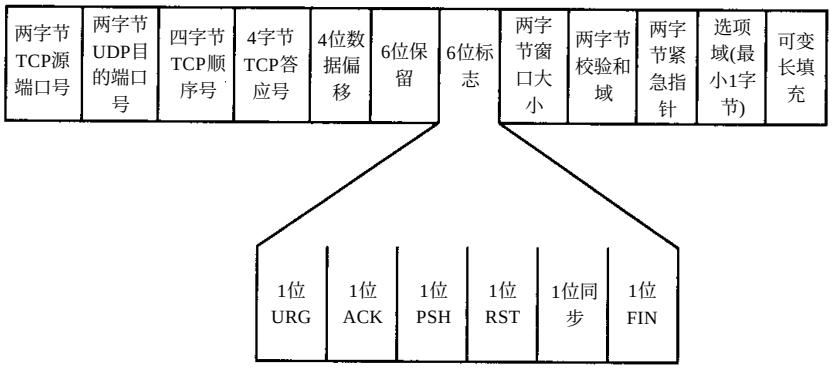

图9-4 TCP头结构显示了TCP能正常工作所依赖的几个域

TCP协议头最少20个字节，包括以下各域：

- TCP源端口 —— 16位的源端口域包含初始化通信的端口号。源端口和源 IP地址的作用是标识报文的返回地址。
- TCP目的端口 —— 16位的目的端口域定义传输的目的。这个端口指明报文接收计算机上的应用程序地址接口。
- TCP序列号 —— 32位的序列号由接收端计算机使用，重组分段的报文成最初形式。在动态路由网络中，一些报文很可能使用不同的路由，因此，报文会乱序到达。这个序列号域可以补偿传输中的不一致。
- TCP应答号 —— TCP使用32位的应答(ACK)域标识下一个希望收到的报文的第一个字节。对一些没发生的事情作应答有点不直观，但收到 ACK报文的源计算机会知道特定的段已经被收到。标识每个 ACK的号是应答报文的序列号。这个域只在 ACK标志被设置时才有效。
- 数据偏移 —— 这个4位域包括TCP头大小，以32位数据结构或称为“字”为单位。
- 保留 —— 6位恒置0的域。为将来定义新的用途保留。
- 标志 —— 6位标志域，每1位标志可以打开一个控制功能，这六个标志是：紧急标志、有意义的应答标志、推、重置连接标志、同步序列号标志、完成发送数据标志。这些标志，以出现的先后顺序排列为 URG、ACK、PSH、RST、SYN和FIN。考虑到前面对它们功能进行的描述，这些标志的意义容易理解。
- 窗口大小 —— 目的机使用16位的域告诉源主机，它想收到的每个 TCP数据段大小。
- 校验和 —— TCP头也包括16位的错误检查域—“校验和”域。源主机基于数据内容计算一个数值。目的主机要进行相同的计算。如果收到的内容没有被改变过，两个计算结果应该完全一样，从而证明了数据的有效性。
- 紧急 —— 紧急指针域是一个可选的16位指针，指向段内的最后一个字节位置，这个域只在URG标志设置了时才有效。如果 URG标志没被设置，紧急域作为填充。在源与目的主机之间网络中的设备要加快处理标识为紧急的数据段。
- 选项 —— 至少一字节的可变长域标识哪个选项 (如果是有的话)有效。如果没有选项，这一字节的域等于0，说明选项域的结束。这个字节等于 1表示无需再有操作。值2表示下

四个字节包括源机器的最大段长度(Maximum Segment Size，MSS)。MSS是数据域中可包含的最大数据量，源和目的机器要对此达成一致。

- 数据——技术上讲，并不是TCP头的一部分，认识到应用数据段在紧急指针和/或选项域之后，但在填充域之前是很重要的。域的大小是最大的MSS，MSS可以在源和目的机器之间协商。数据段可能比MSS小，但却不能比MSS大。
- 填充——和它名字所暗示的不同，填充在数据通信中总是为数学目的而存在。其目的是确保空间的可预测性；定时和规范大小。这个域中加入额外的零以保证TCP头是32位的整数倍。

9.3.2 TCP做什么

TCP在通信会话中提供几个重要作用。可以认为它是多个应用和网络之间的连络。其功能包括：

- 多路复用多种应用数据。
- 测试所接收数据的完整性。
- 顺序化乱序接收的数据。
- 对成功收到数据作出应答。
- 速率——适应的流控(通过TCP窗口大小)。
- 定时功能。
- 重传在传输过程中损坏或丢失的数据。

1. 多路复用数据流

TCP是用户应用与许多网络通信协议之间的接口。因为，实际上没有人听说过TCP只被限制于一个应用，所以TCP必须能同时接收多个应用数据，TCP把它们打包到数据段中，之后传给IP。相似的，TCP必须能同时接收多个应用的数据。

TCP必须跟踪记录到达的报文要转发到的应用程序。这可以通过端口来实现，因此，源机和目的机对通用的应用端口集达成一致是非常有用的。不幸的是有如此众多运行在IP之上的应用，实际上对这些应用对应的端口不可能达成某种形式的一致性。因此IANA，也就是现在的ICANN正加速对至少一部分可用端口号的规范工作。

许多应用很常见，所以它们被认为是众所周知的，这可以简化ICANN的任务。这样一来，ICANN可以给这些应用分配端口号，任何用户可以期望能处理IP报文的主机认识它们，众所周知的端口例子包括：

- 端口80(超文本传输协议，http)。
- 端口119(网络新闻传输协议，nntp)。
- 端口69(纯文本文件传输协议，tftp)。

因为有1024个众所周知的端口(从0到1023)，因此不可能把它们都列出来。TCP和UDP的周知端口的完整列表可参考RFC1700。

如果读者注意，会发现端口域包含一个16位的二进制数。因此，有65 535个可能的端口号。而0~1023是周知口(众所周知端口)，比1023大的端口号通常被称为高端口号。对高端口号ICANN不加以管理。因此，不应排除非众所周知的应用为了通信而使用IP。它们可以选择任何一个高端口号用于通信。

TCP段中既有源应用端口号又有目的端口号。另一个经常使用的术语是套接字，虽然 TCP头中没有套接字域。套接字由驻留在主机上的特定应用端口号和机器 IP地址联合构成。因此，套接字描述了惟一的主机和应用。“ :”号把两个号分开。比如，套接字 10.1.1.19:666标识了主机10.1.1.19上的应用，其端口号为666(DOOM端口号)。

2. 测试数据的完整性

封装在TCP段中的数据经过TCP执行的数学计算，并把结果放在 TCP头的校验和域中。一旦数据到达目的地，对接收数据执行相同的数学计算，产生的结果应该和 TCP头中存储的结果相同。如果二者相同，有理由相信数据没被改变过。否则，就要给源主机发一请求，要求其重发一份数据拷贝。

3. 重新排序

到达目的机的报文段经常是乱序的。其中有许多原因，比如，在一个利用率非常高的网络中，路由协议很可能对报文选择通过网络的不同路径。这会导致数据段乱序到达。另一种情况是，报文在传输过程中可能丢失或损坏。因此，接收应用程序所需的数据序列会被丢弃。目的机器的TCP协议会缓冲接收到的数据段，直到能把它们正确地重新排序。

通过查看TCP头中的序列号域可以完成这个任务，重新排序就是基于这个域对接收数据段的数学排序。

4. 流控

TCP会话中的源和目的机器称为对等实体。每一对等实体有对流向其物理缓冲中数据流的控制能力。流量控制使用的是 TCP窗口大小。源和目的机的窗口大小通过 TCP头进行通信。任何一台主机将被所收数据淹没时，会减小发送机的速率。这可以通过通知其新的窗口大小即可，如果机器的缓冲完全被填满，它就会发送一个有关最后收到数据的应答报文，其中新的窗口大小为0。这样会有效地使发送停止，直到拥塞的机器能清理掉其缓冲。它所处理的每一段必须被应答，使用应答，可以通过重新设置大于 0的窗口尺寸来启动发送。

虽然这个简单的机制能有效地调整两台机器之间的数据流，但是它只能保证通信的端系统不会被接收的数据所湮没。窗口尺寸自身不会考虑网络上存在的拥塞情况。网络拥塞意味着报文到达目的地的时间比通常情况长。因此拥塞管理一定是网络上时间的函数。 TCP通过计时器的使用实现拥塞管理。

5. 计时机制

TCP为几个关键功能使用计时控制。每次传输一个数据段时，设置一个计时器。假如计时器在接到应答之前停止 (就是说，减少到0)，数据段就被认为已丢失。因此，会重传。计时器可以间接地管理网络拥塞，其方法是当超时出现时减慢传输率。理论上讲，当超时出现时才减小发送速率，因此， TCP不能很好地管理网络拥塞，但它会减小自身对拥塞的影响。

源机器会使用一个坚持 (Persist)计时器周期性地查询目的主机的最大窗口尺寸。在理想世界中，从不需要 Persist计时器，因为每个应答会包含窗口尺寸。然而，有时网络确实会丢失数据。如果一台机器发生了缓冲上溢问题，并发回一个 0窗口尺寸的应答，传输节点会中止发送。但是，如果后序非 0窗口尺寸的应答丢失，发送会话会处于危险境地。 Persist计时器通过周期性的询问窗口大小来保证这种情况不会发生。如果查询仍不能得到窗口大小， TCP协议会重新设置连接。

另一个计时机制称为最大段生存时间 (Maximum Segment Lifetime，MSL)。MSL使TCP机

器识别已经在网络中传输了很长时间的因此已被替换了的数据报，接收到 MSL中止的数据报被简单抛弃。

6. 应答接收

如果ACK被设置，目的TCP机器必须要对接收到的特定数据作出应答。考虑到 TCP几乎总是用于可靠模式，因此ACK不被设置的情况很少见。

没被应答的数据段被认为在传输过程中已丢失，并被重传。重传必须在源和目的机器之间配合进行。

9.4 理解用户数据报协议(UDP)

用户数据报协议是IP的另一个主机到主机层协议 (对应于OSI参考模型的传输层)。UDP提供了一种基本的、低延时的称为数据报的传输。为了理解 UDP是如此简单的一种协议，读者只需把RFC768(UDP功能，数据结构和机制的最初规范描述)和其他RFC比较一下就可看出。RFC768内容简短：长度仅有3页纸。其他的RFC文档中3页纸只能够装下内容表！

UDP的简单性使UDP不适合于一些应用，但对另一些更复杂的、自身提供面向链接功能的应用却很适合。其他可能使用 UDP的情况包括：转发路由表数据交换、系统信息、网络监控数据等的交换。这些类型的交换不需要流控、应答、重排序或任何 TCP提供的功能。

9.4.1 UDP头结构

图9-5示出了UDP头结构和各域的大小：

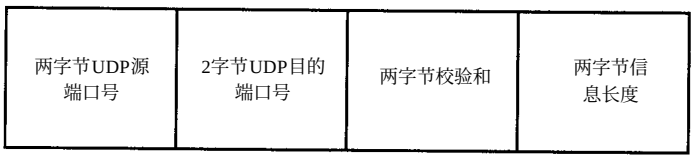

图9-5 UDP头结构显示了UDP形式和功能的简单性

UDP协议头有以下结构：

- UDP源端口号 —— 16位的源端口是源计算机上的连接号。源端口和源 IP地址作为报文的返回地址之用。
- UDP目的端口号 —— 16位的目的端口号是目的主机上的连接号。目的端口号用于把到达目的机的报文转发到正确的应用。
- UDP校验和 —— 校验和是一个16位的错误检查域，基于报文的内容计算得到。目的计算机执行和源主机上相同的数学计算。两个计算值的不同表明报文在传输过程中出现了错误。
- UDP信息长度 —— 信息长度域16位长，告诉目的计算机信息的大小。这一域为目的计算机提供了另一机制，验证信息的有效性。

9.4.2 UDP能做什么

很少！UDP被设计成一个有效的和最小的传输协议。这一点直接反映在其头结构中。它只包括用于转发数据报至合适应用 (端口号)的足够信息，并且执行一定的错误检查。

UDP不提供任何TCP支持的更先进的功能。没有计时机制、流控或拥塞管理机制、应答、

紧急数据的加速传送，或其他任何功能。 UDP使用尽力方式传送数据报。由于某种原因传输失败，数据报被丢弃并且不试图作重传。

9.4.3 TCP和UDP

TCP和UDP是迥异的传输层协议，被设计为做不同的事情。二者的共性是都使用 IP作为其网络层协议。 TCP和UDP之间的主要差别在于可靠性。 TCP是高度可用的，而 UDP是一个简单的、尽力数据报转发协议。这个基本的差别暗示 TCP更复杂，需要大量功能开销，然而UDP是简单和高效的。

UDP经常被认为是不可靠的，因为它不具有任何 TCP的可靠性机制。 UDP不可靠，是因为其不具有TCP的接收应答机制、乱序到达数据的顺序化，甚至不具有对接收到损坏报文的重传机制。也就是说 UDP不保证数据不受损害地到达目的端！因此， UDP最适合于小的发送(也就是单独的报文)，对于数据分成多个报文且需要对数据流进行调节的情况， TCP更适合。

有必要对UDP的不可靠性和UDP的优点作一折衷。 UDP是小的、节约资源的传输层协议。它的操作执行比TCP快得多。因此，它适合于不断出现的、和时间相关的应用如 IP上传输语音和实时的可视会议。

UDP也能很好地适用于其他的网络功能，如在路由器之间传输路由表更新，或传输网络管理/监控数据。这些功能，虽然对网络的可操作性很关键，但是，如果使用可靠的 TCP传输机制会对网络造成负面影响。不可靠的协议并不意味着 UDP是无用协议，它只意味着设计用于支持不同的应用类型。

9.5 小结

TCP/IP协议组件(包括UDP和ICMP)适用于快速增长的用户和应用通信需要近 20年了。那时，这些协议一直被不断更新，以跟上技术不断发展的步伐以及满足 Internet从半私人研究机构到公共商业设施不断演化的需要。

Internet商业化带来了Internet用户前所未有地增长。反过来，这也造成了对更多地址和新的Internet服务类型的需要。因此， IPv4的不足促使人们开发全新的协议版本。新版本称为 IP版本6(IPv6)，但经常也称为网际协议：下一代(IPng)。IPv6将在第10章中详细讨论。