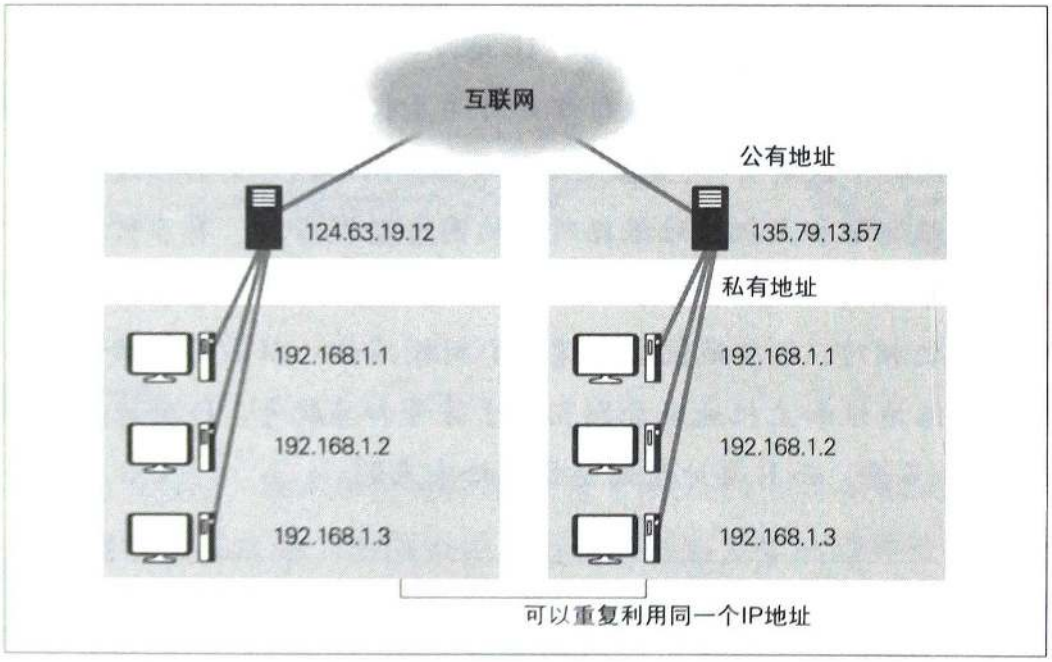
地址中私有地址的范围，在这个范围之外的就是公有地址。通过划分范围，私有地址可以被重复利用（图C1.2），我们可以在有限的地址空间中尽可能地为更多的设备分配IP地址。

表C1.1 私有地址的范围

分类	范围	网络数量
A类地址	10.0.0.0 ~ 10.255.255.255	1
B类地址	172.16.0.0 ~ 172.31.255.255	16
C类地址	192.168.0.0 ~ 192.168.255.255	256



图C1.2 IP地址的重复利用

IPv6的地址分配方法基本上与IPv4一致。由于篇幅有限，本书无法一一列举。IP地址本身涉及许多规则和用法，想要详细了解的读者请务必阅读专门的参考书学习。

[第2层]数据链路层

数据链路层为物理上互相连接的两台设备提供通信功能。例如，一台台路由器之间的通信就是通过它来实现的。此外，连接这样的两台设备并

传递信息的线路称为**链路**（link）。数据链路层将由 0 与 1 的位序列构成的数据组合成数据**帧**（frame），并按照规定的步骤完成信息的收发。

通信流程的标准有许多种。其中，常用于办公室和家庭的有线 LAN 所用的标准统称为**以太网**，它的细则详见 IEEE 802.3 标准。另外，无须 LAN 电缆，只通过电波便可以通信的无线 LAN 现在也广泛普及。这种通信方式基于 IEEE 802.11 标准。

有些标准具有一些机制，能在链路中发生信号丢失时进行一定程度的弥补。这称为**介质访问控制**（Medium Access Control，MAC）。例如，在 IEEE 802.3 标准下，如果信号接收方同时收到多台路由器的信号，就会发生信号冲突，从而导致无法准确还原数据。因此，信号发送方需要事先预测数据到达的时间点，然后调整发送时间以规避此类问题；在已经发生了冲突的情况下，则以数据帧为单位重发数据；但是，如果冲突仍然无法规避，且频繁发生，那么就会放弃在数据链路层重发数据，选择交由上层更可靠的重发协议进行处理。

WAN（Wide Area Network，广域网）中专用网络的通信所用的 PPP（Point-to-Point Protocol，点对点协议），VPN（Virtual Private Network，虚拟专用网络）中使用的 PPTP（Point-to-Point Tunneling Protocol，点对点隧道协议）和 L2TP（Layer 2 Tunneling Protocol，第 2 层隧道协议）也在数据链路层。

——[第 1 层] 物理层

物理层负责将二进制位序列转换为电信号或者光信号，并通过同轴电缆、空间或者光纤等介质[①]完成信息的传输。

在 LAN 线缆上常见的 10 BASE-T 和 100 BASE-TX 等标识是其物理层规格的名称。如同这些名称所示，可通信数据量取决于线缆等介质。

在无线传输的情况下，信息传输的介质是“空间”，因此这种介质没有特定的名称。但是，因为信号是被编码到电波中进行传输的，所以信号要使用的电波频率（**载波**）和可传输的数据量（**带宽**）需要人们事先以法律法规的形式确定好。这是因为无线传输是以空间为介质实现的，这种介质是人类可以利用的公共资源。

① 前文介绍数据链路层时所说的“链路”就是一种介质。

TCP/IP 着眼于实现与实用性的模型

除了 OSI 参考模型之外，还有一种分层协议模型——TCP/IP（也称为 TCP/IP 分层模型、TCP/IP 模型）。TCP/IP 起源于推动了互联网发展的 DARPA（Defense Advanced Research Projects Agency，美国国防高级研究计划局），因此也有人称之为 DARPA 模型。实际上，几乎所有的应用程序都基于 TCP/IP 模型，而非 OSI 参考模型。图 1.5 描述了两者间的对应关系及相应的协议。

图 1.5 分层协议模型与协议的对应关系

OSI 参考模型中从会话层（第 5 层）到应用层（第 7 层）的这 3 层，对应于 TCP/IP 模型中的应用层，其中大部分的功能实现在各个应用程序之中。与前面描述的 HTTP 的例子一样，建立会话的功能实际上是 HTTP 协议的一部分，字符编码的转换和视频压缩格式通常也是由应用程序指定的。

OSI 参考模型如前文所述，是按照实际功能划分的，TCP/IP 则有所不同，它着眼于实现与实用性。可以说正因如此，现在它才被广泛普及和应用。但是，如果我们因为 OSI 参考模型没有得到实际应用就否定了它的价值，是有失偏颇的。学习划分得更为详细的 OSI 参考模型，可以更明确地掌握各层的具体职责与功能，因此可以说 OSI 参考模型是有助于我们掌

握基础知识的重要模型。

RFC

RFC 是由互联网技术标准化组织 IETF（The Internet Engineering Task Force，国际互联网工程任务组）发布的一系列技术文档。

RFC 中记载了 TCP/IP 相关的协议标准文档。记载各种协议的文档会被分配一个编号，并被公开到互联网上。只要阅读这些技术文档，任何人都能了解相应的技术与特性，甚至实现出来。

基础的技术文档如下：UDP 记载在 RFC 768，IP 记载在 RFC 791，TCP 则记载在 RFC 793。有时，新增功能或扩展会定义在新的 RFC 文档中。反过来，RFC 文档有时也会被废除。举例来说，关于 TCP 算法的一个版本——Reno（详见第 3 章）的记述最初是在 RFC 2581 中，但随着 RFC 5681 的更新，RFC 2581 就被废除了。不过，文档即使被废除了，也仍会被保留，所以现在也可以参阅。表 1.1 中介绍了 RFC 文档的几个具体例子。

表 1.1 RFC 示例

RFC 编号	标题	概要
RFC 768	User Datagram Protocol	UDP 基本技术
RFC 791	Internet Protocol	IPv4 基本技术
RFC 793	Transmission Control Protocol	TCP 基本技术
RFC 2001	TCP Slow Start, Congestion Avoidance, Fast Restransmit [1]	慢启动、拥塞避免和快速重传算法
RFC 2460	Internet Protocol, Version 6(IPv6) Specification [2]	IPv6
RFC 3550	RTP: A Transport Protocol for Real-Time Applications [3]	为 UDP 添加时间信息
RFC 5681	TCP Congestion Control [4]	Reno
RFC 6582	The NewReno Modification to TCP's Fast Recovery Algorithm [5]	NewReno

① TCP 慢启动、拥塞避免和快速重传。
② IP 协议（第 6 版）规范。
③ RTP：实时应用程序传输协议。
④ TCP 拥塞控制。
⑤ TCP 快速恢复算法的优化版 NewReno。

分层模型下的数据格式

要想使用各层的协议进行数据通信，需要给数据增加附加信息，主要是首部（header）。首部中存储的并非应用程序间传输的实际数据，而是各个协议层所需的控制信息。IP 协议层所需的是地址信息，TCP 协议层所需的则是数据顺序号和重传控制（retransmission control）信息等。也就是说，首部中存储的数据是便于各个协议层完成自身的职责而事先定义好格式的内容。

图 1.6 展示了 TCP/IP 分层模型中的数据格式。原始的待发送数据，会从上层开始被依序添加上首部。首部并不是待传输的数据，因此会导致数据传输效率降低。这称为系统开销（overhead）。例如，在 1500 字节的以太网帧中，TCP 首部占了 60 字节，IP 首部占了 20 字节，因此应用层以上的有效数据只有 1420 字节。不仅如此，再加上以太网首部的 14 字节，和以太网帧尾部用于错误检查的 FCS（Frame Check Sequence，帧校验序列）的 4 字节，实际的传输效率只有 $1420/(1500+18)\times 100\approx 93.5\%$。

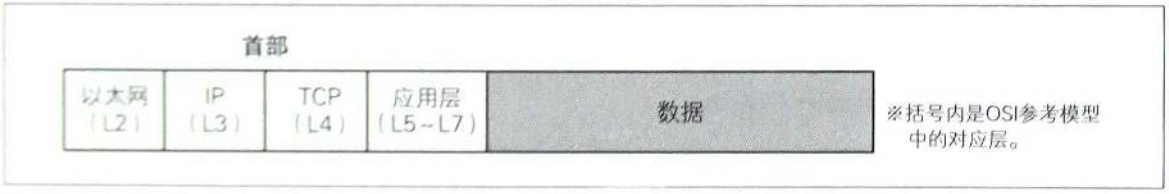


图 1.6 TCP/IP 分层模型中的数据格式示例（各类首部 + 数据）

如上所述，定义首部时应该只保留必要的信息，尽最大可能减小首部长度，而不应该为了增加各种功能而不断增加首部内容。

各个协议层一般只能使用当前层的首部信息。因此，为了与上一层或下一层协议进行通信，需要定义相应的通信协议。

协议分层结构下的通信过程

图 1.7 展示的是分层结构模型下的通信过程。图中，数据由服务器之类的通信主机（host）发出，经由互联网最终到达家庭内部 LAN 中的客

户端计算机。下面将以此为例介绍一下整个过程。

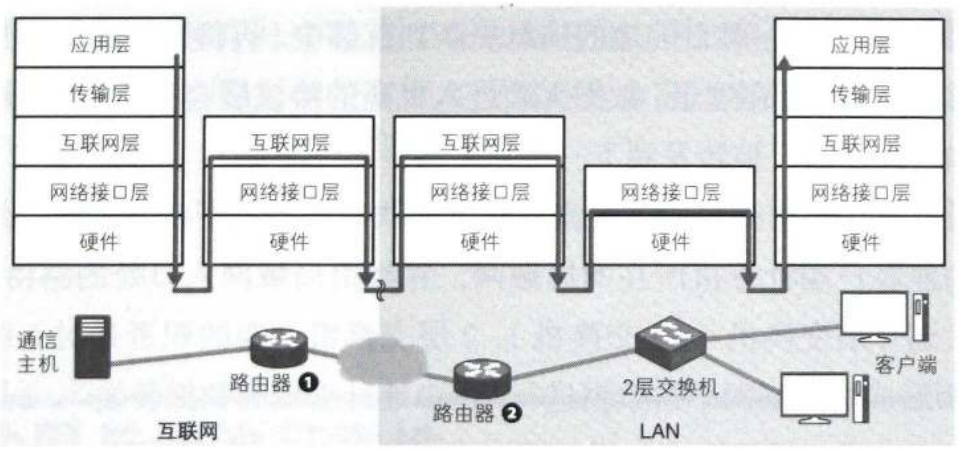


图 1.7 协议分层结构下的通信过程

当服务器端收到来自客户端的通信请求之后，服务器端应用会作为发送方开始发送数据。待发送的数据被依序加上各个协议层的首部，然后通过传输介质被转发到下一跳目的地——路由器。在数据发送方的处理流程中，应用层会先将数据格式的相关信息等写入首部，并将其附加到数据上，然后将数据交给下一层即传输层。

传输层依据应用层的指示，选择使用 TCP 或 UDP 协议，并将所用协议对应的控制信息写入首部，再将其附加到应用层发来的数据上，然后一起交给下一层即互联网层。互联网层获取最终的目的地和下一跳目的地——路由器 ❶ 的信息后，会将这些信息写入首部，并将其附加到传输层发来的数据上，再交给下一层即网络接口层。

网络接口层会将收到的数据调整成依据通信方法事先定义好的帧格式，然后把物理地址和控制信息写入首部，并将其附加到互联网层发来的数据上，生成最终的数据包。数据包通过硬件被转换为电信号，发送给接收方。这个硬件称为**网卡**（Network Interface Card，NIC），每个网卡都会被分配一个独一无二的 **MAC 地址**（Media Access Control address）。数据的传输则是通过电缆等物理介质完成的。

下面介绍数据接收方——路由器 ❶ 的处理流程。电信号通过硬件被

转换为数据后，路由器会从数据中读取网络接口层的首部信息，确认自己是否就是接收方。如果确认无误，路由器会去掉首部，将剩下的数据转发给上一层即互联网层。互联网层读取首部信息，确认下一跳的目的地是路由器❷[1]，并将下一跳目的地的信息更新到首部中，再将首部附加到数据上转发给下一层[2]。在这里，数据无须进入更高的协议层，它会在经过网络接口层和硬件之后，被转发到下一跳路由器。

接下来，数据将以同样的流程经过互联网中的若干台路由器，最终到达目的地客户端计算机所在的局域网，并经由局域网入口处的路由器❷，被转发到2层交换机（L2交换机）。2层交换机主要的职责是对上级路由器❷所形成的局域网中的数据进行MAC地址管理和数据转发[3]。2层交换机只在第2层，即网络接口层就可以完成所有的工作，可谓名副其实。因此，想要知道数据应转发到哪一个MAC地址，就必须确定IP地址与MAC地址之间的对应关系。

路由器等工作在第3层协议以上的设备都有一张对照表，用来记录第2层的MAC地址与第3层的IP地址之间的对应关系。这张表是使用ARP（Address Resolution Protocol，地址解析协议）事先生成的。路由器❷通过查询这张对照表，获取最终目的地即计算机的MAC地址，并将数据转发到2层交换机。然后，2层交换机查询转发目的地的MAC地址，将数据转发给最终的接收方设备。

在最终的目的地即客户端计算机中，每一个协议层都会从收到的数据中解析自己协议层的首部信息，确认自己是否是数据接收方，并在确认无误后将数据交由上一层处理。另外，传输层（在TCP的情况下）还会检查数据是否存在乱序、丢失等问题，并返回确认结果，也就是确认应答（Acknowledgement，ACK）。之后，数据被发送到上一层即应用层，并被

---

① 具体过程是，先读取目的地IP地址，然后查询路由表获取下一跳（即路由器）的IP地址。——译者注

② 原文的描述容易引起误解，其实在互联网层，IP首部的目的地IP字段并不会被修改。这里修改的应该是IP首部记录的路由信息，它用于记录IP数据包经过的设备的IP地址。——译者注

③ 原文比较笼统，实际上2层交换机的作用是处理收到的数据，根据其目的地MAC地址，选择合适的端口转发出去。——译者注

转换为合适的格式。

从上面的例子可以看出，在数据传输中，只需根据设备的不同在必要的协议层进行处理即可。数据的传输线路则会依据网络结构的不同而发生变化。例如，数据有时会经过工作在物理层、用以延长网络的中继器（repeater），有时会经过用于实现多台服务器负载均衡的 4 层到 7 层（L4 ～ L7）交换机。

1.2 传输层与传输可靠性

将数据无乱序、无丢失地发送给接收方

本节将针对传输层进行一些简单的探秘。

传输可靠性

前文所述的传输层的“传输可靠性”，其指代的意义较为模糊。本书将**传输可靠性**定义为“将发送方待发送的数据无乱序、无丢失地发送给接收方”。

网络拥塞 发生在无法从外部观察内部情况的大型网络中的问题

组成网络的各台通信设备是通过线缆等介质在物理上连接在一起的。这些传输介质在单位时间内能传输的数据量有上限。另外，路由器等设备在单位时间内能处理的数据量也有上限。所以，并非想向网络传输多少数据量就能传输多少。假如大量设备同时开始通信，发送大量的数据，就会导致网络瘫痪，无法正常运作。

路由器等设备都有称为**缓冲区**（buffer）的内存区域，用来缓存数据包，以便进行排队处理。如果收到超过缓冲区容量的数据，设备就会无法

处理[1]。

此外，同时从多台路由器收到数据（**冲突**）也可能导致一部分数据出现缺损。这种情况通常称为**拥塞**。

虽然我们可以找到出现拥塞的设备，并通过设备更新等方式直接处理问题，但由于网络规模十分庞大，所以想要直接找到问题并解决无异于大海捞针。

互联网仿佛云层，我们无法从外部看穿其内部的情况（图1.8）。要想知道是否发生了网络拥塞，就必须采取间接的手段。

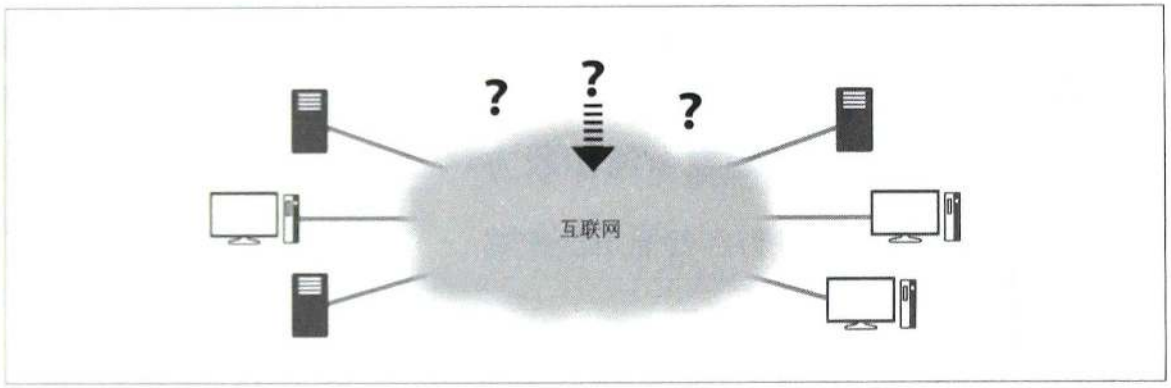


图1.8 互联网

通信对网络的要求

Web、视频、游戏……不同的应用程序对网络有不同的要求

随着网络终端性能的不断提升和智能手机的普及，互联网逐渐地渗透到我们生活的方方面面，已是我们生活中不可或缺的一部分。电子邮件、SNS（Social Network Software，社交网络服务）、音乐、视频，还有各种数据处理，全都是通过互联网实现的。近年来，一些需要保持互联网连接才能进行通信的应用程序也越来越多，例如在线游戏、Web摄像头视频直播等。

在这种背景下，网络流量也在逐年增长（详见第58页的图2.13）。如前文所述，由于出现拥塞，网络会时不时地发生数据丢失问题。但与此同

1. 这通常称为缓冲区溢出。

时，我们却总能成功地收到信息，浏览网络内容。这一切，都要归功于传输层实现了网络拥塞控制和丢失数据补偿的机制。

表 1.2 展示了应用程序对网络的一些要求。其中，通话对实时性要求很高，网页浏览和文件下载则对传输可靠性有要求。提到流媒体，人们通常会想到视频流媒体观看服务，不过近些年来，网络广播和音乐流媒体播放服务也逐渐增多。此外，也有像网络游戏那样对实时性和可靠性都有要求的应用程序。网络中连接的通信设备都需要使用符合这些要求的数据通信方法。

表 1.2 应用程序对网络的要求

应用程序	可靠性	实时性	特征
网页浏览	√	—	耗时稍久一点也无妨，但数据一旦缺失就无法获得信息
文件下载	√	—	耗时也无妨，但数据哪怕缺失一部分也会导致获取失败
语音或视频通话	—	√	双向，即使偶尔网络断线也不影响交流
流媒体	△	√	单向，故障会导致加载视频卡顿
网络游戏	√	√	时延、断线都会带来致命的影响

传输层的职责

那么，确保传输可靠性的机制设置在哪里呢？TCP 定义在 OSI 参考模型的传输层（第 4 层）中。传输层规定了一系列用来完成数据通信和确保通信可靠性的通信步骤。

如前文所述，应用程序的需求千变万化。每一位程序开发者，如果想要一个个地满足所有的需求，绝对压力巨大。因此人们制定了若干个协议，用来规定所有通信设备必备的标准功能。满足所有的需求，必然要实现一大堆传输协议，这显然是不现实的。因此 TCP 和 UDP 两个协议作为满足需求的最低标准被制定出来。

1.3 UDP的基本情况

无连接的简单特性

在介绍TCP之前，我们先来介绍一下只有一些简单特性的UDP协议。UDP虽然特性比较简单，但它并没有因此而无人问津，反而在许多应用程序之中得到了应用。

UDP的基础知识 无连接的通信

UDP记载于标准文档RFC 768。它不具备任何确保传输可靠性的复杂机制，提供的是**无连接的通信**（connectionless communication）。

UDP的示意图如图1.9所示，应用程序发送出去的数据被直接发送到网络之中。UDP实际执行的功能只是“把数据转发到目的地”和“使用校验和（详见第3章）检测数据是否损坏”。假如进行后文所述的TCP的重传控制，数据传输就一定会出现时延（latency），因此UDP不会进行重传控制。

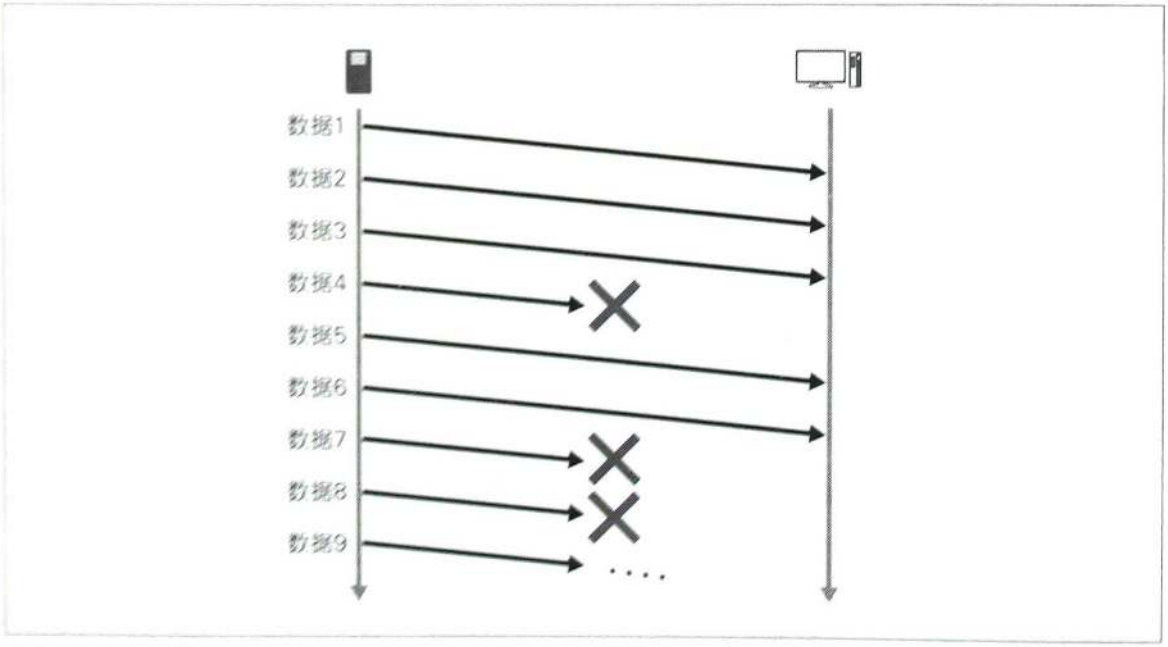


图1.9 UDP

UDP 更看重实时数据传输，而非数据可靠性。首部也只有 8 个字节，非常简单（具体的首部结构见 3.1 节的介绍），所以 UDP 的系统开销较少，比 TCP 的传输效率要高。

单播、多播、广播

此外，UDP 支持同时向多个目的地发送数据。数据传输方式的分类如图 1.10 所示。

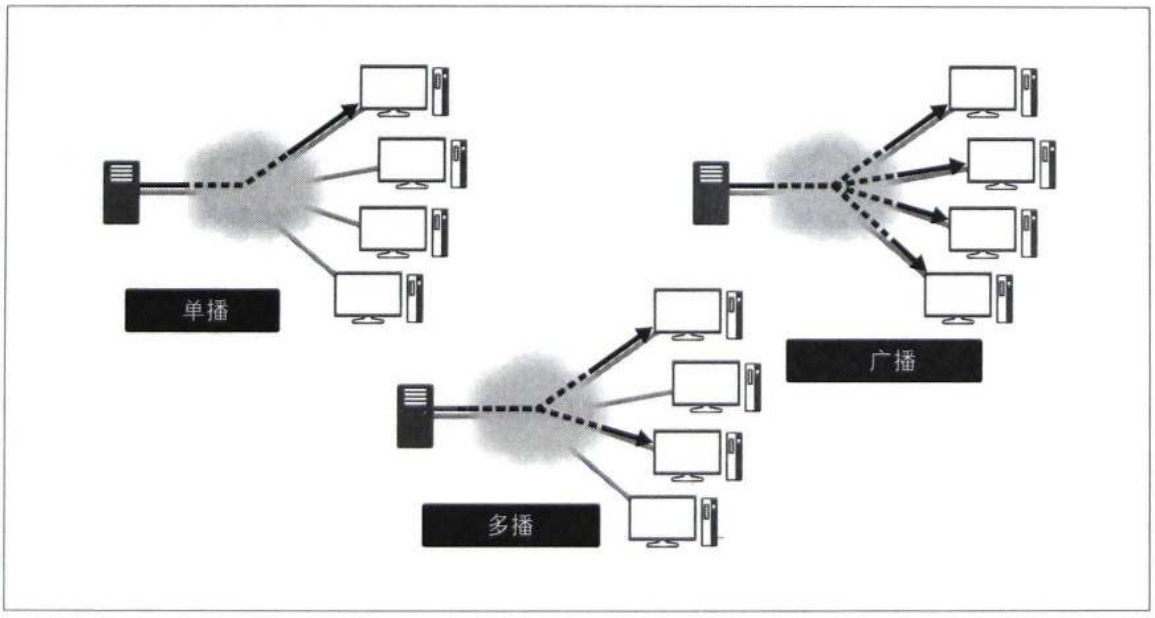


图 1.10 数据传输方式的分类

单播（unicast）是面向单个通信对象的数据传输方式；**多播**（multicast）是面向事先确定好的多个通信对象的数据传输方式；**广播**（broadcast）是面向非特定的多个通信对象同时通信的数据传输方式。

TCP 支持单播的传输方式，而 UDP 支持所有传输方式，即支持单播、多播和广播。

UDP 和应用程序可靠性的确保 RTP 和 RUDP

另外，如果既想用 UDP 又要确保传输的可靠性，上层的应用程序就必须进行控制处理。例如，采用增加了序列号和时间戳机制、能够保证实

时性的RTP（Real-time Transport Protocol，实时传输协议，RFC 3550），以及增加了序列号、确认应答和重传机制的RUDP（Reliable User Datagram Protocol，可靠用户数据报协议，详见1.6节）等。

适合UDP的应用程序 视频串流、VoIP和DNS等

适合UDP的应用程序通常有哪些？举例来说，视频串流、VoIP（Voice over IP，基于IP的语音传输）之类的语音通话和视频通话就是典型的例子。另外，每次请求都需要尽快得到响应的应用，例如DNS、RIP、DHCP和NTP等也都是基于UDP协议运作的。

近些年来，随着视频的高清化和语音通话的IP化，UDP流量不断地增加，进一步挤占了TCP流量的规模。因此，TFRC（TCP Friendly Rate Control，TCP友好速率控制算法，RFC 5348）在2003年被提出，它是考虑了对TCP流量的影响、基于UDP协议改良得到的算法。

此外，HTTP协议在传输层一般使用TCP协议，但是基于UDP协议改良后的QUIC（Quick UDP Internet Connections，快速UDP互联网连接）协议，和以QUIC协议作为传输层协议的HTTP/3也逐渐普及开来。下面的专栏将简单介绍QUIC。到目前为止，TCP各个版本和IPv6等技术都是基于协议层的概念不断发展的。然而，也许不久以后，我们就可以像QUIC一样不再拘泥于协议层次，而是在应用程序层次上单独实现能够确保通信可靠性的功能。

专 栏

QUIC：以UDP为基础快速建立高可靠性通信链路

截至笔者执笔时（2019年5月），传输层协议QUIC正在由IETF进行标准制定。HTTP-over-QUIC技术在2018年末由IETF宣布正式更名为HTTP/3，因此QUIC有望成为HTTP协议的下层协议。除此之外，QUIC对IP协议的版本也没有要求。另外，为了确保安

全性，它使用 TLS[①] 1.3，所有的连接都会被加密。QUIC 目前仍处于标准化流程中，具体的标准规范未来很有可能发生各种变化，因此本书不再详细涉及。如果想要了解细节，请通过互联网查阅最新的信息。

提升 HTTP 的通信速度是人们开发 QUIC 的主要目的。在通常情况下，TCP 协议需要通过 3 次握手建立连接，之后由于要进行 TLS 连接，所以从发出 TCP 报文段开始到收到对应的 ACK（详见后文）为止所花费的往返时延（*RTT*，详见后文）越长，那么到开始发送数据所需要花费的时间也越长。与之相对，QUIC 基于 UDP 协议，连接的建立与 TLS 的建立是同时进行的，因此只需要 0 或 1 个往返时延便可以开始发送数据。另外，TCP 在出现丢包时，如果没有完成丢包数据的重传，是无法进行后续逻辑处理的，这称为队头阻塞（head-of-line blocking）。在这一点上，QUIC 无须关心丢包情况，可以直接按照送达数据的顺序依序处理，因此效率更高。

1.4 TCP 的基本情况

可靠性的确保与实时性

终于要切入正题，开始介绍 TCP 了。首先，我们通过与 UDP 进行对比，来看一下 TCP 的特征吧。

TCP 的基础知识 面向连接的通信

TCP 的基本标准记载于 RFC 793。

TCP 提供**面向连接型的通信**（connection-oriented communication），会

① 即传输层安全协议（Transport Layer Security）。它是一个基于通信对象认证、通信内容加密和数据篡改检测技术来实现安全通信的协议。2018 年 3 月由 IETF 正式承认的 TLS 1.3 是当前（2019 年 5 月）的最新版本（RFC 8446）。

确认通信设备间连接的开始和结束。在传输数据的过程中，发送方发送数据，而接收方在收到数据后返回对应的ACK。通过这种方式，双方设备便可以确认数据是否发送成功，进而**确保数据传输准确无误**。此外，TCP支持通信双方同时进行数据的收发，即**全双工通信**。TCP中数据传输的基本单位是TCP报文段。

网络的状况可能会导致到达接收方的TCP报文段的顺序前后颠倒，但是TCP会对**顺序**进行管理。

TCP会采用一系列措施，如在预测网络的拥堵情况的同时**控制发送的TCP报文段的数量**，**重传**丢失的TCP报文段数据等，实现端到端的高可靠性通信。

TCP与UDP的功能与特点

表1.3列出了TCP和UDP各自的功能。从表中的内容我们可以看出，UDP功能比较简单，而TCP拥有多个功能，相对比较复杂。

表1.3 TCP与UDP的功能对比

名称	功能
TCP	连接管理、序列号、重传控制、顺序控制、拥塞控制和校验和
UDP	校验和

表1.4列出了TCP和UDP因功能不同而表现出的不同特点。

表1.4 TCP与UDP的特点对比

名称	转发类型	可靠性	实时性	通信对象数量	拥塞控制
TCP	流模式	有	低	一对一	有
UDP	数据报模式	无	高	一对一、一对多	无

TCP只有在收到ACK之后，才会发送新的数据，而且具有重传机制，所以能够确保可靠性。可以说，TCP正是牺牲了实时性才保证了可靠性。

UDP则无视数据丢失和数据乱序的问题，持续发送数据包。而数据接收方只要收到数据就会第一时间交给上一层即应用层。换句话说，UDP