第二章 分组网络上的语音和视频通信

本章目录

- TCP/IP和OSI参考模型
- IP网络的性能特征
- 测量IP网络性能
- 传输协议的影响
- 分组网络中音频/视频传输的要求

在深入研究RTP细节之前，你应该了解诸如Internet之类的IP网络的特性，以及它们如何影响语音和视频通信。本章回顾了Internet体系结构的基础知识，并概述了网络连接的典型行为。回顾之后，讨论音频和视频的传输需求，以及网络如何满足这些需求。

IP网络具有影响音频/视频传输应用程序和协议设计的独特特性。如果你想了解RTP设计中涉及的权衡，以及它们如何影响使用RTP的应用程序，那么理解这些特征是至关重要的。

TCP/IP和OSI参考模型

当你考虑计算机网络时，理解协议分层的概念和含义是很重要的。如图2.1所示的OSI参考模型，为分层系统的讨论和比较提供了有用的基础。

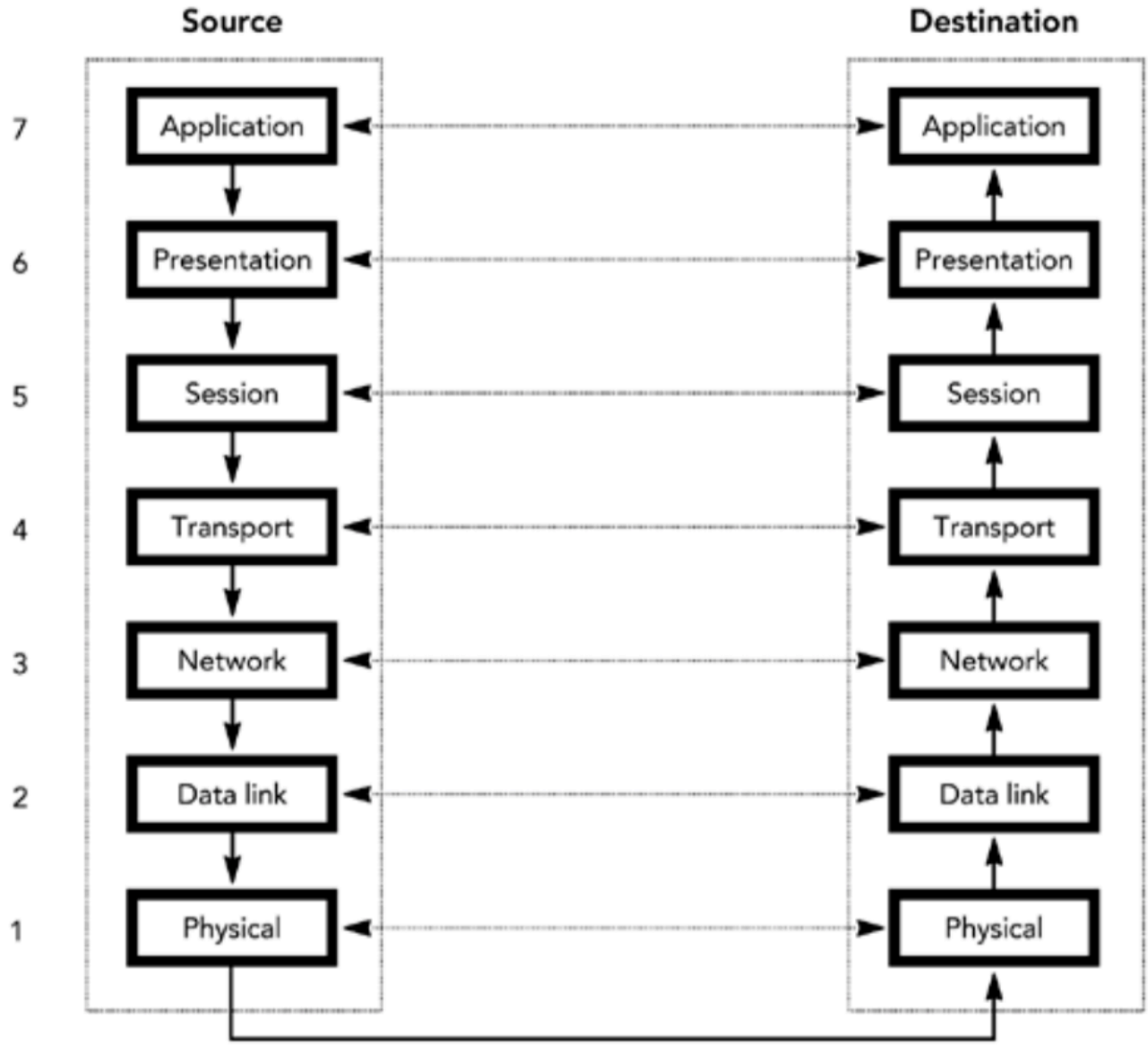

Figure2.1.The OSI Reference Model

该模型由七层组成，每一层都建立在较低层提供的服务之上，从而为上一层提供更抽象的服务。各层的功能如下:

- 物理层 最底层——物理层——包括物理网络连接设备和协议，如电缆、插头、开关和电气标准。
- 数据链路层 数据链路层建立在物理连接的基础上;例如，它将双绞线转换成以太网。这一层为数据传输单元提供帧，定义如何在多个连接的设备之间共享链接，并为每个链接上的设备提供寻址。
- 网络层 网络层连接链接，将它们统一为一个网络。它通过网络提供消息的寻址和路由。它还可以控制交换机中的拥塞、某些消息的优先级、计费等等。网络层设备处理从一个链接接收到的消息，并将其发送到另一个链接，使用与这些链接远端节点交换的路由信息。

- 传输层 传输层是第一个端到端的层。它负责使用网络层提供的服务将消息从一个系统传递到另一个系统。此职责包括在会话层需要时提供网络层没有提供的可靠性和流控制。
- 会话层 会话层以对应用程序有意义的方式管理传输连接。示例包括用于检索Web页面的超文本传输协议(HTTP)、电子邮件交换期间的简单邮件传输协议(SMTP)协商以及管理文件传输协议(FTP)中的控制和数据通道。
- 表示层 表示层描述了较低层所传递的数据的格式。示例包括用于描述Web页面表示的HTML(超文本标记语言)、描述电子邮件格式的MIME(多用途Internet邮件扩展)标准，以及更常见的问题，如FTP中的文本传输和二进制传输之间的差异。
- 应用层 应用程序本身—例如web浏览器和电子邮件客户机—构成系统的顶层，即应用层。

在模型的每一层，在一个主机上的某层与另一个主机上的等效层之间存在逻辑通信。当一个系统上的应用程序希望与另一个系统上的应用程序进行通信时，通信将向下通过源的各个层，通过物理连接的传递，然后向上到达目的地的协议堆栈。

例如，一个Web浏览器应用程序呈现一个HTML展现，它使用HTTP会话、TCP传输连接、IP网络、以太网数据链接、使用双绞线物理电缆来传递。每一步都可以看作是模型的特定层的实例化，向下通过协议栈。其结果是将Web页面从应用程序(Web服务器)转移到应用程序(Web浏览器)。

这个过程并不总是那么简单:源和目标之间可能没有直接的物理连接，在这种情况下，在中间网关系统，连接必须部分的上升协议栈。它需要上升多远?这取决于连接的是什么。以下是一些例子:

- 日益流行的IEEE 802.11b无线网络使用基站将一个物理层(通常是有线以太网)连接到另一个物理层(数据链路层的无线链路)。
- IP路由器提供了网关的一个示例，其中多个数据链路在网络级连接。在移动电话上查看Web页面通常需要连接一直上升到网关中的表示层，该层将HTML转换为无线标记语言(WML)，并将连接传递到不同的低层。
- 如上所述，我们可以使用OSI参考模型来描述互联网。这种契合并不完美:互联网的架构是随着时间的推移而演变的，在一定程度上早于OSI的模式，通常来讲，实际分层表现出的严格性比所描述的要低得多。然而，考虑互联网协议套件与OSI模型之间的关系，特别是IP作为一个通用网络层所扮演的角色，是很有意义的。

OSI参考模型的最低两层可以直接与Internet相关，Internet可以通过各种链接工作，如拨号调制解调器、DSL、以太网、光纤、无线和卫星。每个链接都可以用OSI模型的数据链路/物理层分割来描述。

在网络层，一个特定的协议将一组完全不同的私有网络转换为全球Internet。这是互联网协议(IP)。IP向上层提供的服务很简单:尽最大努力将数据报包传递到指定的目的地。由于这项服务非常简单，IP可以运行在广泛的链路层上，使得Internet能够快速传播。

简单性是有代价的:IP不能保证任何特定的传输时效性，或者数据报根本就不能传输。包含IP数据报的数据包可能会丢失、重新排序、延迟或被低层损坏。IP不会试图纠正这些问题;相反，它将数据报原封不动地传递到上层。不过，它确实提供下列服务:

- 分片，防止数据报大于底层链路层的最大传输单元。
- 一个“生存时间”字段，防止循环的包永远循环
- 一种服务类型标签，可用于为某些类型的包提供优先级
- 上层协议标识符，用于将数据包定向到正确的传输层
- 端点的寻址——包括多播来寻址一组接收端——并将数据报路由到正确的目的地

显示这些服务如何映射到包的IP报头的格式如图2.2所示。

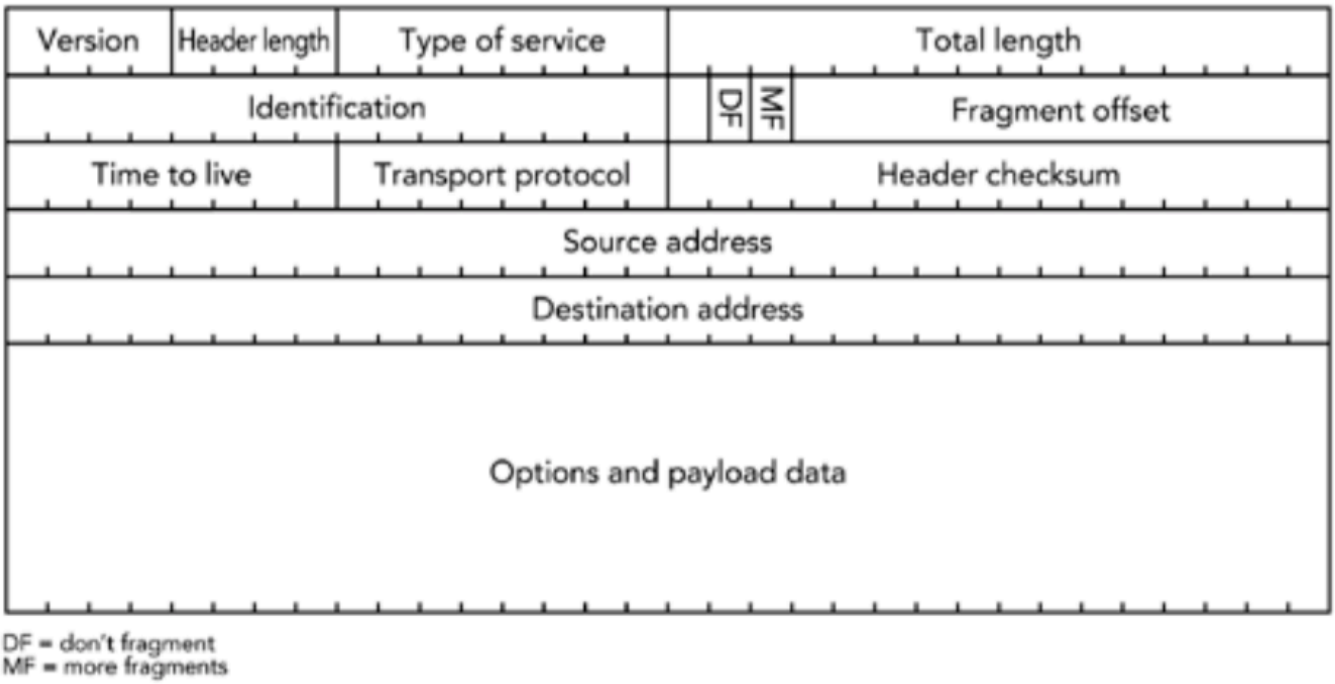

Figure 2.2. Format of an IP Header.png

图2.2中的标头是IPv4标头，是当前Internet上的标准。但现在有往IPv6过渡的举动，它提供了本质上相同的功能，但大幅增加了地址空间(128位地址，而不是32位)。如果发生这种转换——这是一种长期的前景，因为它涉及连接到互联网的每个主机和路由器的变化——它将允许更多的机器连接，从而促进网络的增长，但它不会在其他方面显著地改变服务。

Internet协议提供了单个网络的抽象，但这并不改变系统的基本性质。尽管它看起来是一个单一的网络，但实际上互联网是由许多独立的网络组成的，这些网络由网关(现在更常被称为路由器)连接，并由单一的IP服务和地址空间统一。图2.3显示了单个网络如何构成更大的Internet。不同的互联网服务提供商选择如何运行全球网络下它们自己的那部分:一些拥有高容量的网络，几乎没有拥塞，高可靠性;其他的没有。

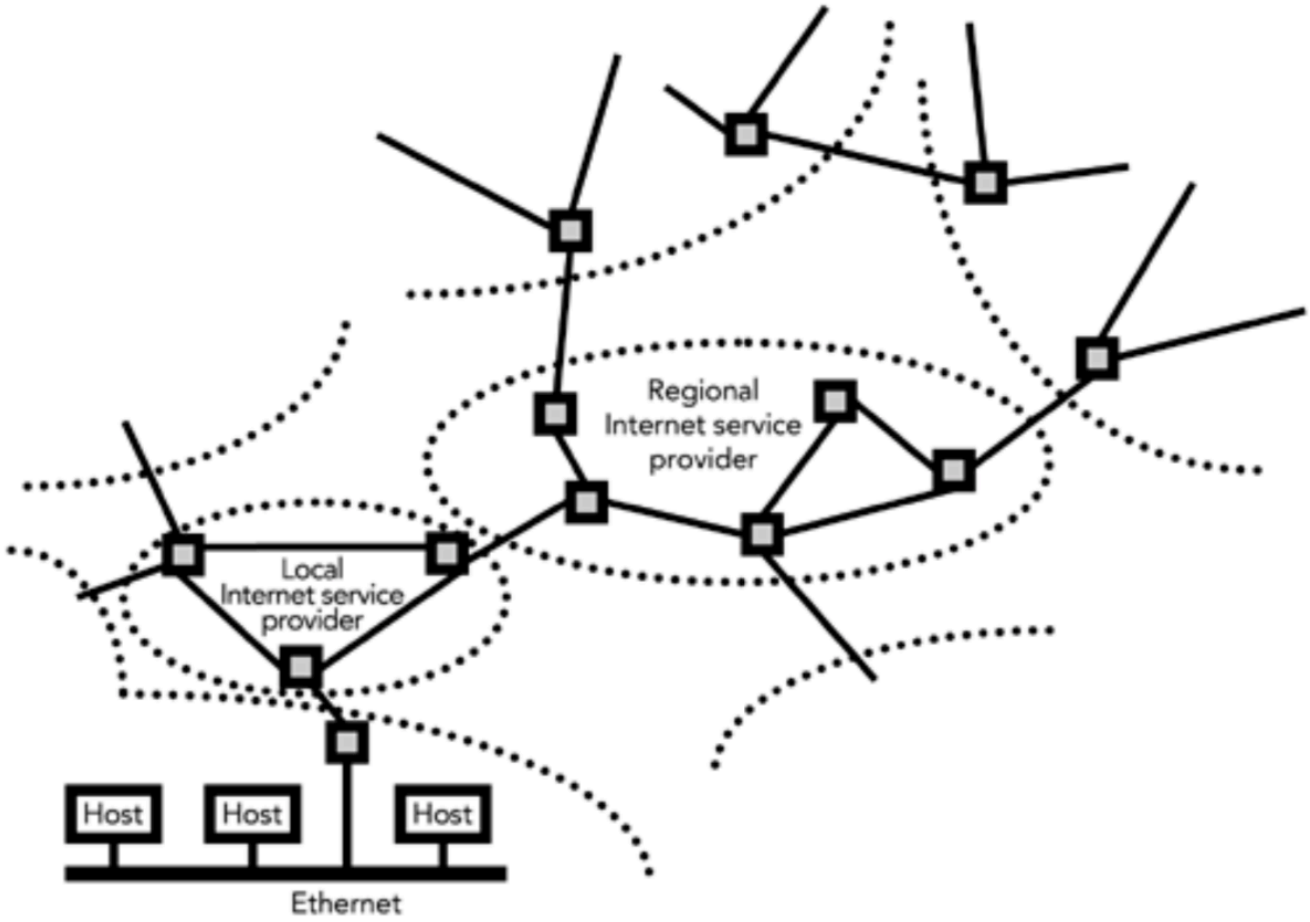

Figure2.3. An IP Inter-network

在错综复杂的互联网络中，包含IP数据报的数据包被单独路由到各自的目的地。路由器不需要立即发送数据包;如果在发送链路上正在传输另一个包，它们可以将其短暂地排队。它们还可能在拥塞时丢弃数据包。如果底层网络发生变化(例如，由于链接失败)，IP包所采取的路由可能会发生变化，这可能导致上层协议可以观察到传输质量的变化。

在Internet体系结构中，位于IP之上的是两种常见的传输协议:传输控制协议(TCP)和用户数据报协议(UDP)。TCP对原始IP服务进行调整，以便在每个主机上的服务端口之间提供可靠的、有序的传输，并根据网络的特性改变传输速率。另一方面，UDP提供与原始IP服务类似的服务，只是增加了服务端口。本章后面将更详细地讨论TCP和UDP。

在这些传输协议之上，是Internet世界中常见的会话协议，例如用于Web访问的HTTP和用于发送电子邮件的SMTP。堆栈由各种表示层(HTML、MIME)和应用程序本身完成。

从这个讨论中应该清楚的是，IP在系统中扮演着关键的角色:它提供了一个抽象层，对应用程序隐藏了底层网络链接和拓扑的细节，并将底层与应用程序的需求隔离开来。这种体系结构称为沙漏模型，如图2.4所示。

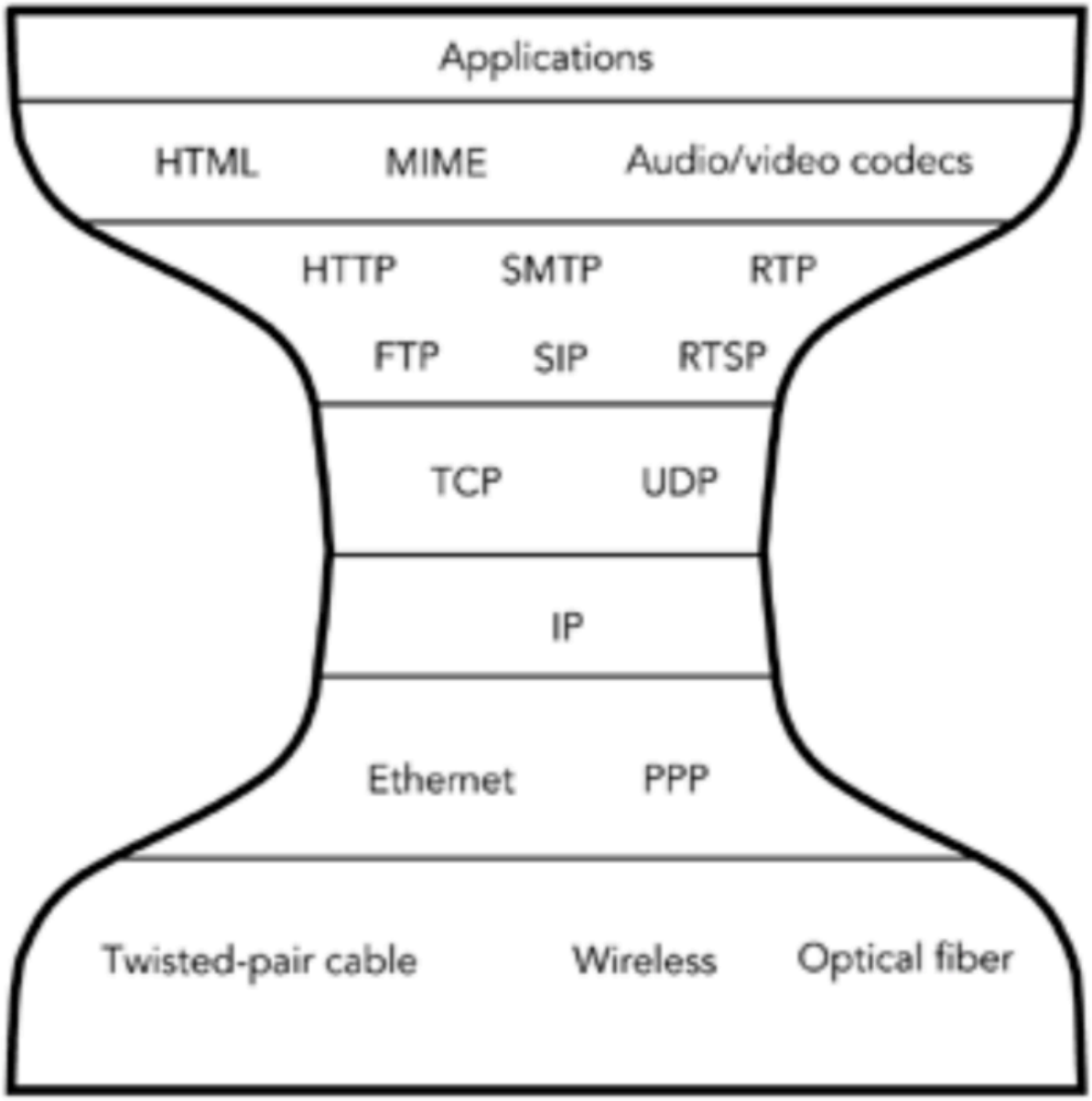

Figure2.4. The Hourglass Model of the Internet Architecture

决定跨Internet通信系统性能的主要因素是IP层。较高层协议可以在一定程度上适应和补偿IP层的行为，但若IP层性能较差会导致整个系统性能较差。接下来的两个部分将详细讨论IP层的性能，指出它的独特特性以及它带来的潜在问题和好处。

IP网络的性能特征

从Internet体系结构的沙漏模型可以明显看出，应用程序通过抽象IP而隐藏了较低层的细节。这意味着应用程序无法直接确定一个IP包所经过的网络类型——它可以是任何东西，从14.4千比特的蜂窝无线电连接到一个千兆比特的光纤——或者该网络的拥塞程度。获取网络性能的唯一方法是观察和测量。

那么我们需要测量什么，如何测量呢?幸运的是，IP层的设计意味着参数的数量是有限的，而且这个数量通常可以根据应用程序的需要进一步加以限制。我们可以问的最重要的问题是:

- 数据包在网络中丢失的概率是多少?
- 数据包在网络中被破坏的概率是多少?
- 数据包通过网络需要多长时间?传输时间是常数还是变量?
- 可容纳多大的包?
- 我们发送信息包的最大速率是多少?

下一节将介绍上述列出的前四个参数的一些样例测量。最大速率与数据包在网络中丢失的概率密切相关，如第10章拥塞控制中讨论的那样。

什么影响这样的测量?最明显的因素是测量站的位置。在局域网上两个系统之间进行的测量将清楚地显示与跨大西洋连接不同的属性!但地理因素并不是唯一的因素;遍历链接的数量(通常称为跃点的数量)、经过运营商的数量以及进行度量的时间都是因素。Internet是一个大型的、复杂的、动态的系统，因此必须小心确保任何测量都能代表要使用应用程序的网络部分。

我们还必须考虑所使用的网络类型、其他流量以及其他流量的大小。到目前为止，绝大多数网络路径是固定的、有线的(铜或光纤)连接，绝大多数流量(96%的字节，62%的流量，根据最近的估计)是基于TCP的。这些流量模式的影响如下:

- 由于基础设施主要是有线和固定的，所以链路非常可靠，而损耗主要是由路由器的拥塞造成的。
- TCP传输假定包丢失是一个信号，表明瓶颈带宽已经达到，拥塞正在发生，应该降低它的发送速率。TCP流将增加它的发送速率，直到观察到丢失，然后返回，这是一种确定特定连接可以支持最大速率的方法。当然，其结果是瓶颈链接临时超载，这可能会影响其他流量。

如果网络基础设施或流量的组成发生变化，其他丢包来源可能变得重要。例如，无线用户数量的大量增加可能会增加丢包比例，这是由于包损坏和对无线链路的干扰而造成的。在另一个例子中，如果使用TCP以外的传输的多媒体流量的比例增加了，而这些传输对丢失的反应与TCP不同，那么丢失模式可能会因为拥塞控制动态的变化而改变。

当我们开发在IP上运行的新应用程序时，我们必须意识到我们给网络带来的变化，以确保我们不会给其他用户带来问题。第10章，拥塞控制，更详细地讨论了这个问题。

测量IP网络性能

本节概述有关IP网络性能的一些可用数据，包括平均包丢失的公布结果、丢失模式、包损坏和重复、传输时间和多播影响的结果。

有几项研究测量了公共互联网上各种条件下的网络行为。例如，Paxson报告了9个国家35个站点之间的20,000例转移行为;Handley和Bolot对多播会话行为的研究, Yajnik、Moon、Kurose和Towsley报告了包丢失统计中的时间依赖性。其他数据来源包括CAIDA(互联网数据分析合作协会)、NLANR(应用网络研究国家实验室)和ACM(计算机械协会)维护的流量档案。

平均丢包

各种包丢失度量能够被研究。例如，平均丢包率提供了对网络拥塞的一般度量，而丢包模式和相关性提供了对网络动态的观察。

报告的平均丢包率测量显示了一系列情况。例如,由帕克森在1994年和1995年做的TCP / IP流量的测量显示,根据路线和日期,30%到70%的流量显示没有包丢失,但那些显示有丢包的流量,平均丢包范围从3%到17%(这些结果总结在表2.1)。来自Bolot的使用64 kb的pcm编码音频的数据，显示了类似的模式，丢包率在4%到16%之间，这取决于一天的时间，尽管这些数据也可以追溯到1995年。Yajnik等人在1997-1998年使用模拟音频流量的最新结果显示，丢包率较低，为1.38%至11.03%。Handley的结果——1996年5月和9月的两组大约350万包数据和多播视频的接收报告统计数据——显示，根据接收位置和时间的不同，每五秒的平均丢包在0%到100%之间变化。1996年5月29日，一个特定接收器在10小时内的样本，如图2.5所示，显示了在5秒间隔内采样的平均丢失率在0%到20%之间变化。

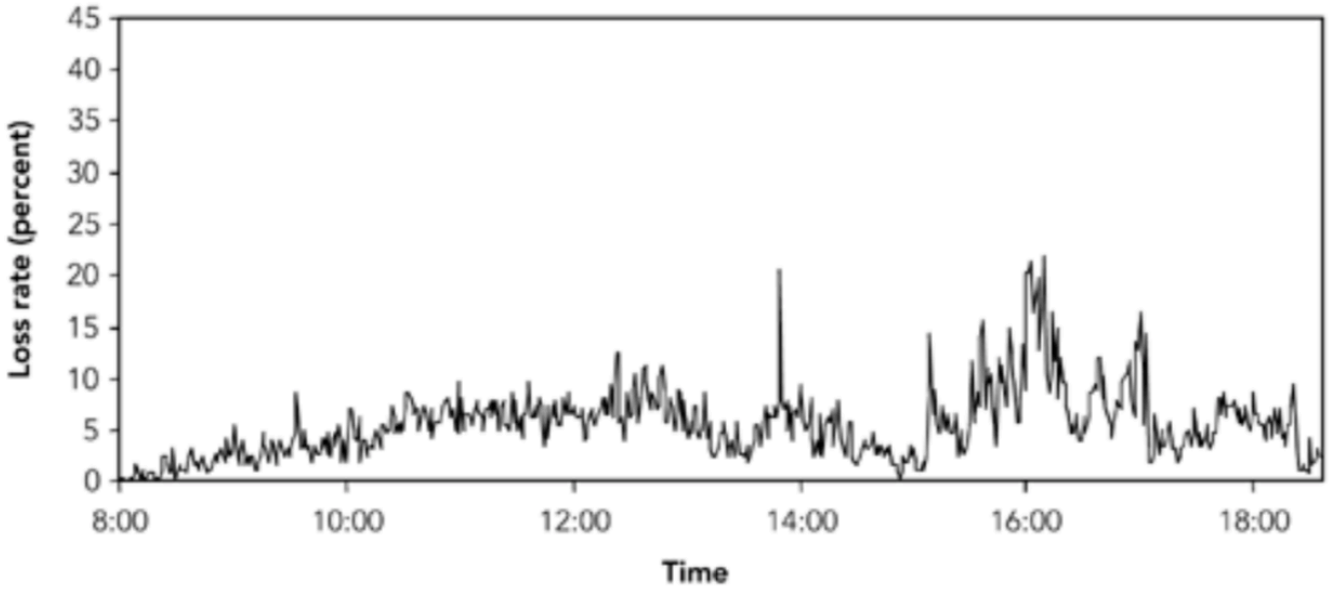

图2.5 丢包率随时间的分布

	Fraction of Flows Showing No Loss		Average Loss Rate for Flows with Loss	
Region	**Dec. 1994**	**Dec. 1995**	**Dec. 1994**	**Dec. 1995**
Within Europe	48%	58%	5.3%	5.9%
Within U.S.	66%	69%	3.6%	4.4%
U.S. to Europe	40%	31%	9.8%	16.9%
Europe to U.S.	35%	52%	4.9%	6.0%

表格2.1. 不同地区的丢包率

观测到的平均丢包率不一定是恒定的，也不一定是平稳变化的。该样本显示了一个丢包率，尽管在某些点上发生了突然的变化，但总体而言，变化相对平稳。来自Yajnik等人的另一个示例如图2.6所示。这个案例显示了丢包率的一个更显著的变化:在一个小时的过程中，丢包率从2.5%缓慢下降到1%，10分钟后，丢包率上升到25%，然后恢复正常——这个过程几分钟后重复。

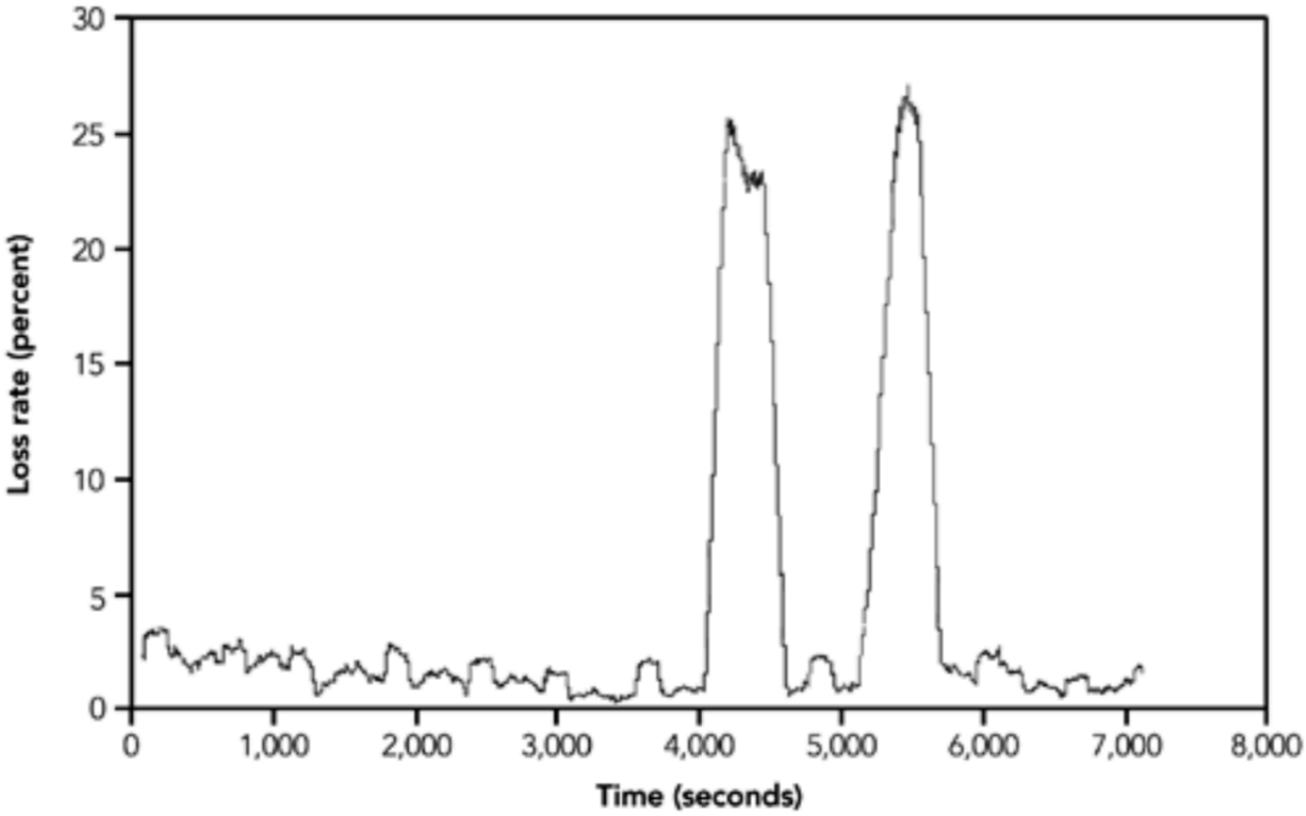

图2.6 丢包率随时间的分布

这些丢包率与目前的网络相比如何?在写这篇文章的时候，传统的观点是，可以对网络主干进行设计，这样就不会发生包丢失，所以人们可以期待最近的数据来说明这一点。在某种程度上这是真的;然而，即使有可能使网络的一部分免于丢包，这种可能性并不意味着整个网络将以同样的方式运行。今天，许多网络路径都出现了丢失，即使丢失的只是一小部分数据包。

《互联网天气报告》(Internet Weather Report)是对互联网上一系列路由的丢包率进行的月度调查。该报告显示，截至2001年5月，根据ISP的不同，美国境内的平均丢包率从0%到16%不等。在美国，每月的平均丢包率约为2%，但就整个互联网而言，平均丢包率略高，约为3%。

我们能从中学到什么?即使网络的某些部分得到了很好的设计，其他部分也会有很大的丢包。请记住，如表2.1所示，美国境内70%的网络路径在1995年没有丢包，而其他网络的平均丢包率几乎为5%，这个速率足以导致音频/视频质量的显著下降。

丢包模式

除了研究平均丢包率的变化外，考虑短期的丢包模式也很有意义。如果我们的目标是恢复丢包，那么我们需要了解丢包是随机分布在一个媒体流中，还是突发的。

如果丢包在时间上是均匀分布的，那么我们应该期望特定包丢失的概率与前一个包丢失的概率相同。这意味着丢包通常是孤立事件，这是一个理想的结果，因为单个丢包比连续丢包更容易恢复。然而，不幸的是，如果前面的包丢失了，那么丢失特定包的概率通常会增加。也就是说，丢包往往是连续发生的。Vern Paxson的测量表明，在某些情况下，如果之前的包丢失，特定包的丢失概率会增加5到10倍，这显然意味着包丢失不是均匀分布的。

其他一些研究——例如，Bolot在1995年、Handley和Yajnik等人在1996年和我在1999年收集的测量数据——证实了包丢失概率不是独立的。这些研究表明，在所有情况下，绝大多数丢包都是单个数据包，约占丢包的90%。如图2.7所示，较长的突发丢包概率降低;很明显，如果丢包是独立的，较长时间的突发丢包会发生得更频繁。

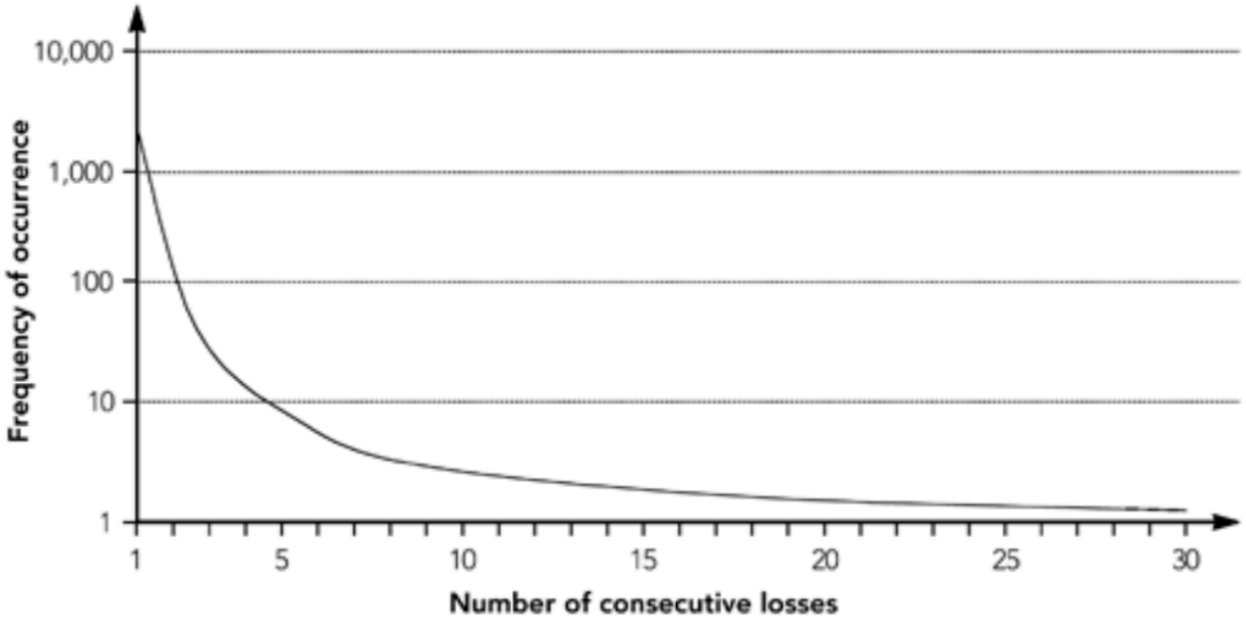

图2.7 连续丢失的数据包数目的量化分布

观察到的丢包模式在某些情况下也显示出明显的周期性。例如，Handley报告说，在1996年的测量中，大约每30秒就会发生一次突发丢包(见图2.8)，2001年4月也报告了类似的问题。这样的报告并不是普遍的，许多迹象表明没有这样的影响。据推测,周期性是由于某些系统路由更新引起的过载导致的,但不确定。

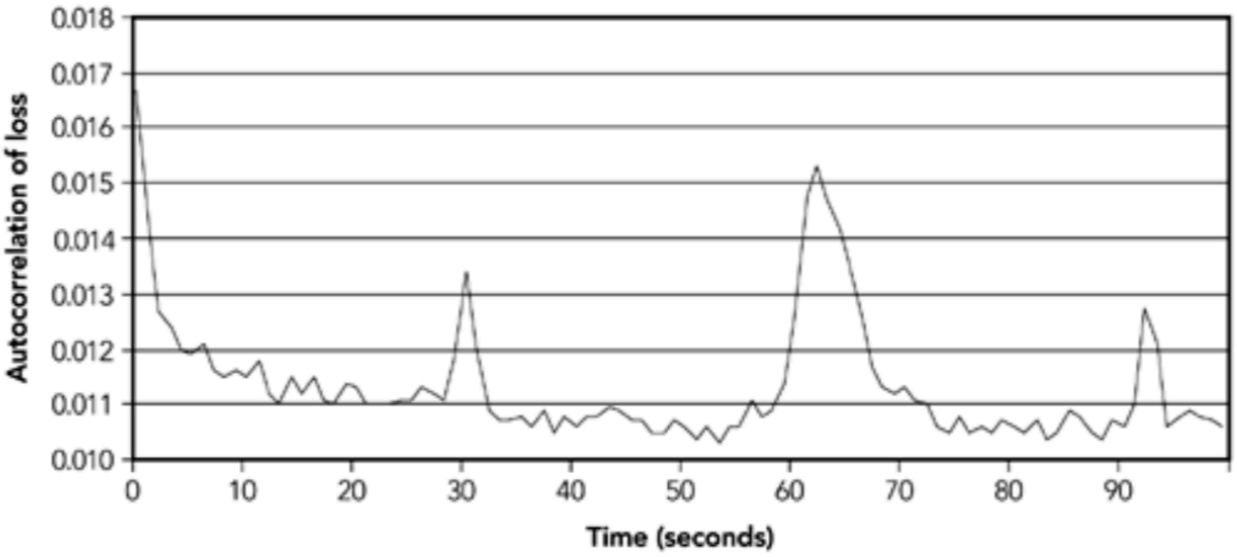

图2.8 丢包的自相关

数据包重复

如果数据包在网络中会丢失，那么它们会出现重复吗?也许令人惊讶的是，答案是肯定的!一个源可能发送一个包，而接收者可能获得该包的多个副本。重复的最可能的原因是网络中的路由/交换元素出现故障;正常操作中不应出现重复。

重复包有多普遍?Paxson的测量显示了连续丢包的趋势，也显示了少量的包重复。在测量20,000个流时，他发现了66个重复的包，但他也指出，“我们已经观察到了一些痕迹……其中超过10%的包重复。问题是由于桥接设备配置不当引起的。”我在1999年8月进行的跟踪显示了来自大约125万个包有131个重复。

只要应用程序知道问题并丢弃了重复包(RTP为此包含一个序列号)，那么重复的包不应该引起问题。重复的包过多会浪费带宽，同时这标志着网络配置错误或设备故障。

包损坏

包可能丢失或重复，也可能被破坏。每个IP包都包含一个校验码，该校验码保护包报头的完整性，但在IP层不保护有效负载。但是，链路层可能提供校验码，TCP和UDP都支持整个数据包的校验码。理论上，这些协议将检测到大多数损坏的数据包，并使它们在到达应用程序之前被丢弃。

关于数据包损坏频率的统计数据很少被报道。Stone引用了Paxson的观察结果，即大约每7500个数据包中就有一个未能通过TCP或UDP校验，这表明数据包已经损坏。该工作中的其他测量显示平均校验失败率从1100分之一到31900分之一不等。注意，这个结果是针对有线网络的;无线网络可能会有显著不同的特性，因为无线电干扰造成的损坏可能比电线噪

音造成的损坏更严重。

当校验失败时，该包被认为已损坏并被丢弃。应用程序不会看到损坏的数据包;这些包对应用程序来说已经丢了。包损坏会导致测量的丢包率小幅增加。

在某些情况下，应用程序可能需要接收损坏的包，或者获得包损坏的明确指示。UDP为这些情况提供了一种禁用校验码的方法。第8章《错误隐藏》和第10章《拥塞控制》，更详细地讨论了这个主题。

网络传输时间

数据包通过网络需要多长时间?答案取决于所走的路线，虽然短路线比长路线花费的时间要少，但是我们需要注意对“短”的定义。

影响传输时间的因素包括链路的速度、数据包必须通过的路由器数量，以及每个路由器造成的排队延迟。在物理距离较短的路径中，数据包的跳数可能较长，而在每一跳路由器中的排队延迟往往是主要因素。在网络术语中，短路径通常是跳数最少的路径，即使它覆盖了较长的物理距离。卫星链路是一个明显的例外，它的距离会带来无线电显著的传播延迟。表2.2提供了2001年5月平均往返时间的量度数据，以供比较。对存在各种往返延迟的电话交谈的研究表明，人们不会注意到少于300毫秒的延迟。虽然这显然是一个取决于人和任务的主观度量，但关键是所测量的网络往返时间大多在这个限制之内。(从伦敦到悉尼是一个例外，但这里的显著增长可能是由于传输路径上有一跳是卫星。)

Source	Destination	Hop Count	Round-Trip Time (ms)
Washington, DC	London	20	94

Source	Destination	Hop Count	Round-Trip Time (ms)
Washington, DC	Berkeley, CA	21	88
London	Berkeley, CA	22	136
London	Sydney	28	540

表2.2 样本往返时间测量

对延迟的度量本身并不是很有趣，因为很显然它们取决于源和目的地的位置。更有趣的是网络传输时间是如何随着数据包而变化的:对于应用程序来说，工作在具有固定传输延迟的网络上比工作在传输延迟不断变化的网络上更容易，特别是当应用程序传输时间敏感的媒体数据时。

传输时间变化(jitter)的粗略度量是包的到达率。例如，图2.9显示了以恒定速率发送的流的测量到达率;很明显，到达率变化很大，这表明网络上的传输时间不是恒定的。

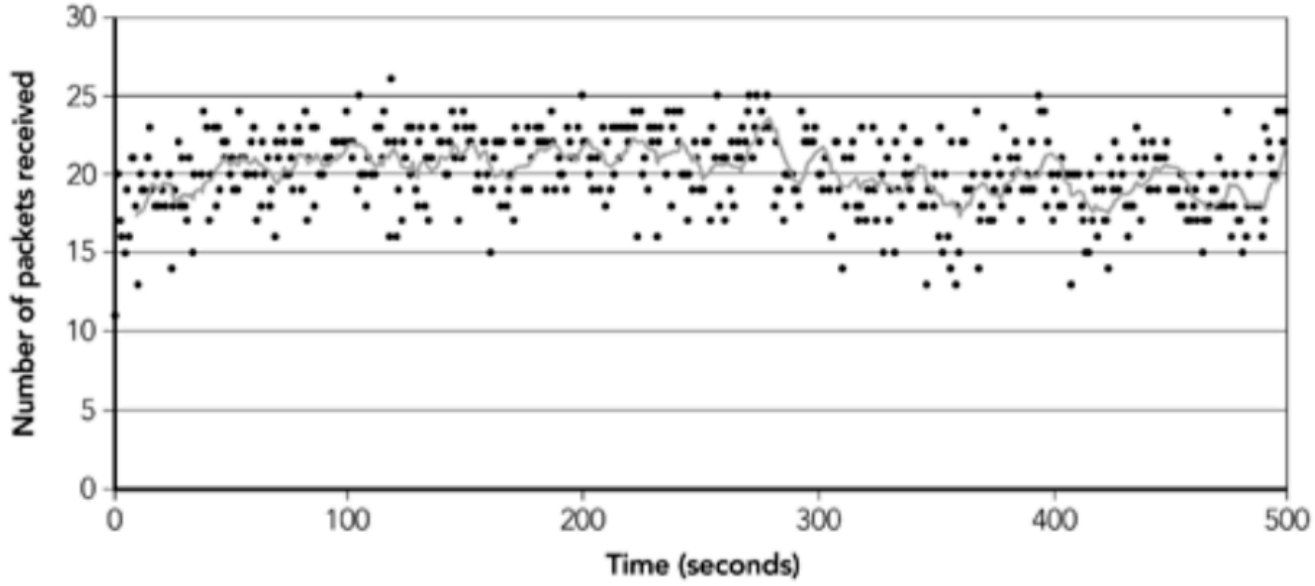

图2.9 包到达率

一个更好的测量方法是通过测量每个包的到达时间和离开时间的差值来求出传输时间，而不是假设速率不变。不幸的是，测量绝对传输时间是困难的，因为它需要源和目的地的时钟精确同步，而这通常是不可能的。因此，大多数网络传输时间的追踪都包括时钟偏移，而且除了延迟的变化之外，不可能研究其他任何东西(因为不可能确定有多少偏移是由未同步时钟造成的，有多少是由网络造成的)。

图2.10和图2.11给出了传输时间变化的一些测量样本，包括由于时钟不同步造成的偏移。我是在1999年8月测量的;Ramjee等人(1994年)和Moon等人也提出了类似的测量方法。请注意以下几点:

- 测量值的缓慢向下倾斜是由于源和目标之间的时钟倾斜造成的。一台机器的时钟比另一台的稍微快一点，导致感知到的传输时间逐渐改变。
- 可以观察到平均传输时间的几个较大的改变，这可能是由于网络中的路由改变所致。
- 传输时间不是常数;相反，它在整个过程中会有显著的变化。

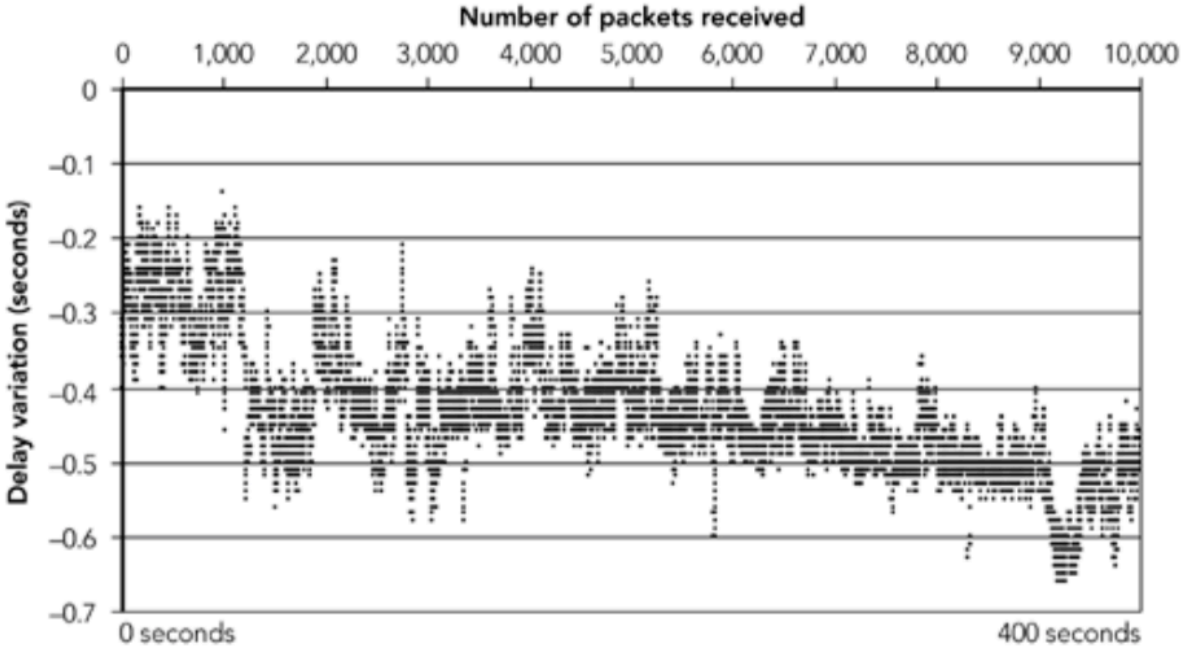

图2.10-网络传输时间变化-400秒样本

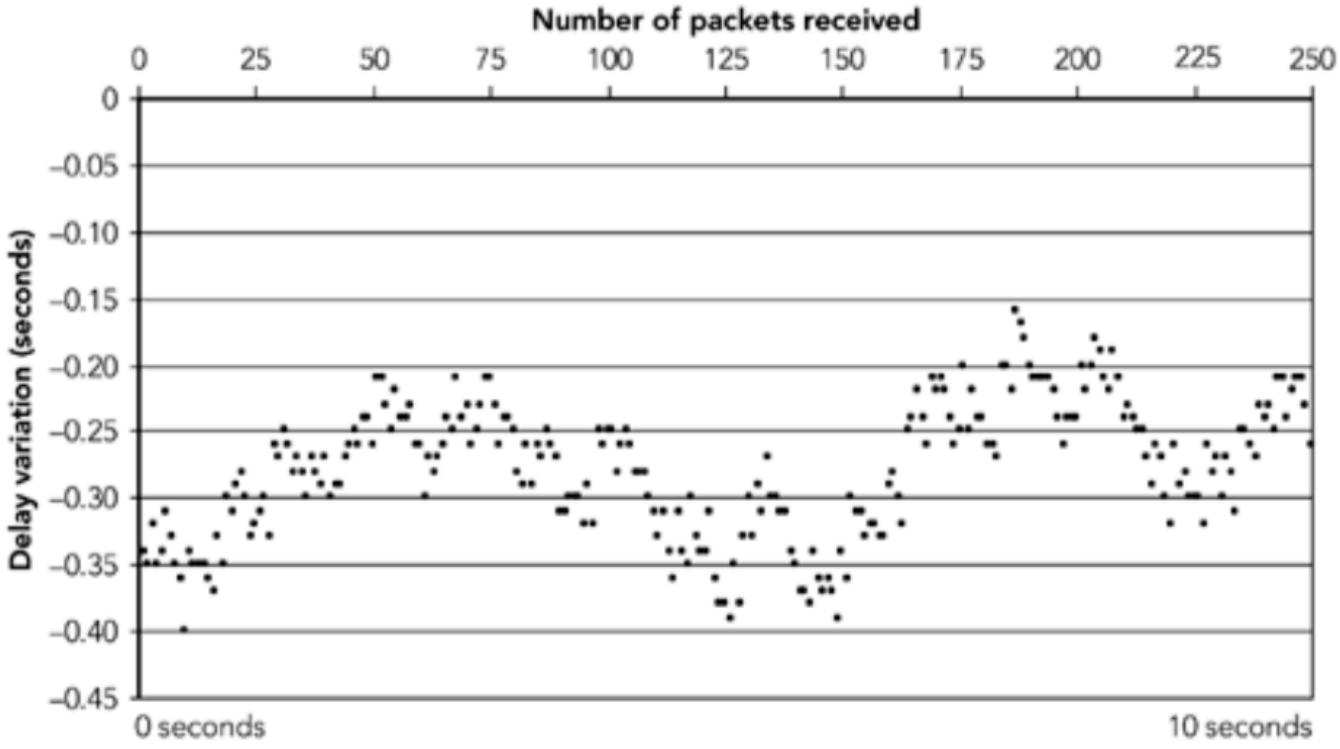

图2.11网络传输时间变化-10秒采样

这些都是应用程序或更高层协议必须准备处理的问题，如果需要，还必须纠正。

在网络中对数据包进行重新排序也是可能的——例如，如果路由发生了更改并且新路由更短的话。Paxon观察到，总共有2%的TCP数据包是无序传输的，但是在不同的追踪之间，无序传输数据包的比例有很大的差异，其中一条追踪显示15%的数据包是无序传输的。

网络传输时间中的“峰值”是另一个可以被观察到的特征，如图2.12所示。目前还不清楚这些峰值是由于网络内的缓冲还是由于发送系统中的缓冲，但是如果试图将数据包序列的到达时间变得平滑，那么这些峰值将是一个有趣的挑战。

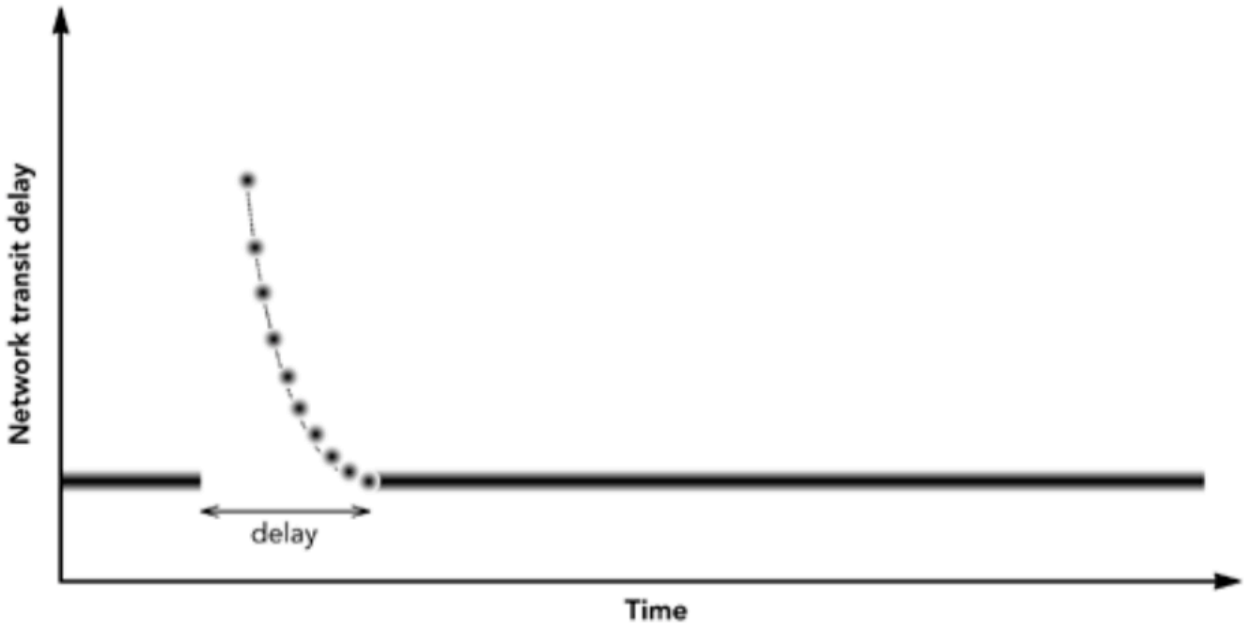

图2.12 网络传输时间峰值

最后，网络传输时间可以显示周期性(如Moon et al. 1998中提到的)，尽管这似乎是一种次要的影响。我们期望这种周期性与前面提到的丢包周期性有相似的原因，除非这些事件不那么严重，只导致路由器中的队列堆积，而不是队列溢出导致丢包。

可接受的数据包大小

IP层可以接受大小不等的数据包，长度最多可达65,535字节，或者由所传输链路的最大传输单元(MTU)所限制。MTU是一个链路可以容纳的最大数据包的大小。一个常见的值是1500字节，这是以太网可以传输的最大数据包。许多应用程序假设它们可以最大发送到这个大小的数据包，但是一些链接的MTU较低。例如，拨号调制解调器链接的MTU普遍为576字节。

在大多数情况下，瓶颈链接靠近发送方或接收方。几乎所有的骨干网链路都有一个1500字节或更多的MTU。

IPv4支持分段，当一个大的数据包超过一个链路的MTU时，就会被分割成更小的片段。然而，依靠这个通常是一个坏主意，因为任何一个片段的丢失将使接收者不可能重建数据包。由此产生的丢包乘数效应是我们希望避免的。

在几乎所有情况下，音频包大小都落在网络MTU内。视频帧更大，应用程序应该把它们分成多个包进行传输，这样每个包都适合网络的MTU。

多播的影响

IP多播允许发送方同时向多个接收方传输数据。它有一个有用的特性，即网络根据需要创建包的副本，这样只有一个包的副本遍历对应的一个链接。IP多播提供了非常高效的组通信，前提是网络支持它，这使得向一组接收器发送数据的成本与该组的大小无关。

支持多播是IP网络的一个可选的、相对较新的特性。在撰写本文时，它相对广泛地部署在研究和教育环境以及网络主干中，但在许多商业环境和服务提供商中并不常见。

发送到一个组意味着更多的事情可能出错:接收质量不再受到通过网络的单一路径的影响，而是受到从源到每个单独接收者的路径的影响。在测量组播会话的损耗和延迟特性时，定义因素是均匀性。图2.13演示了这个概念，显示了我测量的多播会话中每个接收器的平均丢包率。

图2.13 多播会话中的丢包率

多播不会改变网络中丢失或延迟的根本原因。相反，它使每个接收器都能经历这些影响，而源只传输每个包的一个副本。网络的异构性使得源很难满足所有的接收方:有些发送太快，有些发送太慢是很常见的。我们将在后面的章节中进一步讨论这些问题。现在，只需注意多点传送为系统增加了更多的异构性就足够了。

网络技术的影响

到目前为止提出的测量方法是公共的、大范围的、互联网的。许多应用程序将在这种环境中运行，但大量应用程序将用于私有内部网、无线网络或支持增强服务质量的网络。这些情况如何影响IP层的性能?

许多私有IP网络(通常称为内部网)具有与公共互联网非常相似的特性:流量组合通常非常相似，许多内部网覆盖范围很广，链接速度和拥塞程度各不相同。在这种情况下，测试结果很可能与公共互联网上的测试结果相似。然而，如果网络是专门为实时多媒体流量而设计的，就有可能避免许多已经讨论过的问题，并构建一个没有丢包和最小抖动的IP网络。

一些网络使用集成服务/RSVP或差异化服务来支持增强的服务质量(QoS)。使用增强的QoS可以减少应用程序对丢包和/或抖动恢复的需求，因为它为满足某些性能限制提供了强有力的保证。然而，请注意，在许多情况下，QoS方案提供的保证本质上是统计意义的，通常它不能完全消除数据包丢失，或者传输时间的变化。

在无线网络中可以观察到显著的性能差异。例如，蜂窝网络可以在短时间内表现出显著的性能变化，包括非阻塞丢包、突发丢包和高误码率。另外，一些蜂窝系统具有高延迟，因为它们在数据链路层使用交织来隐藏突发的丢包或包损坏。

不同网络技术的主要影响是增加了网络的异构性。如果你正在设计一个应用程序来处理这些技术的一个有限子集，那么你可以利用底层网络的功能来提高应用程序所看到的连接的质量。在其他情况下，底层网络可能会给健壮应用程序的设计者带来额外的挑战。

明智的应用程序开发人员会选择健壮的设计，这样当应用程序从最初设想的网络转移到新网络时，它仍然可以正确地运行。设计可在IP上运行的音视频应用程序的挑战是使它们在面对网络问题和意外情况时仍旧可靠。

关于测量特性的结论

测量、预测和建模网络行为是有许多微妙之处的复杂的问题。这一讨论只涉及这些问题，但一些重要的结论是显而易见的。

第一点，网络可以而且经常表现得很糟糕。如果一个工程师设计了一个应用程序，他希望所有的包都能及时到达，那么当这个应用程序被部署到Internet上时，他一定会大吃一惊。虽然更高层的协议(如tcp)可以隐藏一些这种缺点，但总有一些方面对应用程序是可见的。

另一个需要认识的要点是网络中的异构性。网络中某一点的测量结果不能代表另一点的情况，甚至“不寻常”的事件也一直在发生。到2000年底，网络上大约有1亿个系统，因此，即使发生在不到1%的主机上的事件也会影响成千上万台机器。作为应用程序设计人员，你需要了解这种异构性及其可能的影响。

尽管存在这种异构性，试图总结丢包和丢包模式的讨论揭示了几个“典型”特征:

- 虽然有些网络路径可能不会丢包，但这些路径在公共网络中并不常见。一个应用程序应该被设计来处理少量的数据包丢失——比如说，达到5%。
- 孤立的丢包组成了大多数观察到的丢包事件。
- 丢包的概率不是均匀的:即使大多数丢包是孤立的包，连续丢包的突发概率也比随机事件更常见。丢包的突发通常是短暂的;一个应用程序，处理两到三个连续丢失的包将足以满足大多数突发丢包。

- 很少出现长时间的突发丢包。一秒甚至更长的故障时间并不是未知的。
- 包重复很少见，但也可能发生。
- 类似地，在极少数情况下，数据包可能被破坏。其中绝大多数是由TCP或UDP校验码(如果启用)检测到的，包在到达应用程序之前会被丢弃。

传输时间变化的特征可以总结如下:

- 网络上的传输时间不是均匀的，而且会观察到抖动。
- 绝大多数抖动是合理有界的，但分布的长尾效应比较明显。
- 虽然重新排序相对较少，但在传输过程中可能会重新排序数据包。应用程序不应该假定接收数据包的顺序与发送数据包的顺序一致。

这些并不是通用的规则，每一个规则都会有一个网络路径作为反例。然而，它们确实提供了一些我们在设计高层协议和应用程序时需要注意的一些概念。

传输协议的影响

到目前为止，我们对网络特性的考虑主要集中在IP上。当然，程序员几乎从不使用原始IP服务。相反，它们在较高层传输协议(通常是UDP或TCP)的基础上构建应用程序。这些协议提供了IP协议之外的其他特性。这些添加的特性如何影响应用程序所看到的网络行为?

UDP/IP

用户数据报协议(UDP)提供了一组最小的IP扩展。UDP报头如图2.14所示。它包含64位附加头，代表源和目标端口标识符、长度字段和校验码。

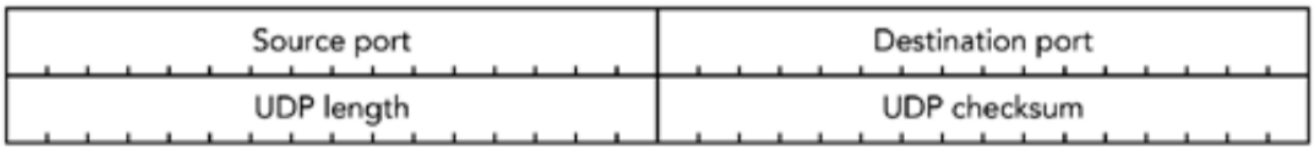

图2.14 UDP报头格式

源端口和目标端口标识通信主机中的端点，从而允许将不同的服务多路复用到不同的端口。一些服务运行在众所周知的端口上;另一些则使用在调用设置期间动态协商的端口。长度字段与IP报头中的长度字段是冗余的。校验码用于检测有效负载的损坏，是可选的(对于不需要校验码的应用程序，校验和设置为0)。

除了增加端口和校验和外，UDP还提供原始的IP服务。它没有增强传输的可靠性(尽管校验码可以检测到IP没有检测到的负载错误)，也不影响包传输的时间。使用UDP的应用程序向传输层提供数据包，传输层将数据包发送到目标机器上的一个端口(如果使用多播，则发送到一组机器)。这些包可能在传输过程中丢失、延迟或乱序，这与原始IP服务的情况完全相同。

TCP/IP

Internet上最常用的传输协议是TCP。虽然UDP只向IP服务提供了一小部分附加功能，但TCP添加了大量附加功能:它抽象了不可靠的IP包传递服务，从而在源端口和单个目标主机之间提供可靠的、连续的字节流传输。

使用TCP的应用程序向传输层提供一个数据流，传输层将其分割成适当大小的数据包，并以适合网络的速率进行传输。数据包由接收方确认，在传输过程中丢失的数据包由源重新传输。当数据到达时，在接收端进行缓冲，以便按顺序传递。这个过程对应用程序是透明的，应用程序只看到一个数据流经网络的“管道”。

只要应用程序提供足够的数据，TCP传输层就会增加它的发送速率，直到网络出现数据包丢失。丢包视为已超过瓶颈链路带宽的信号，该连接应降低其发送速率以匹配。相应地，TCP降低了丢包发生时的发送速率。这个过程会继续下去，TCP会不断探测整个网络的传输速率;结果是一个如图2.15所示的发送速率。

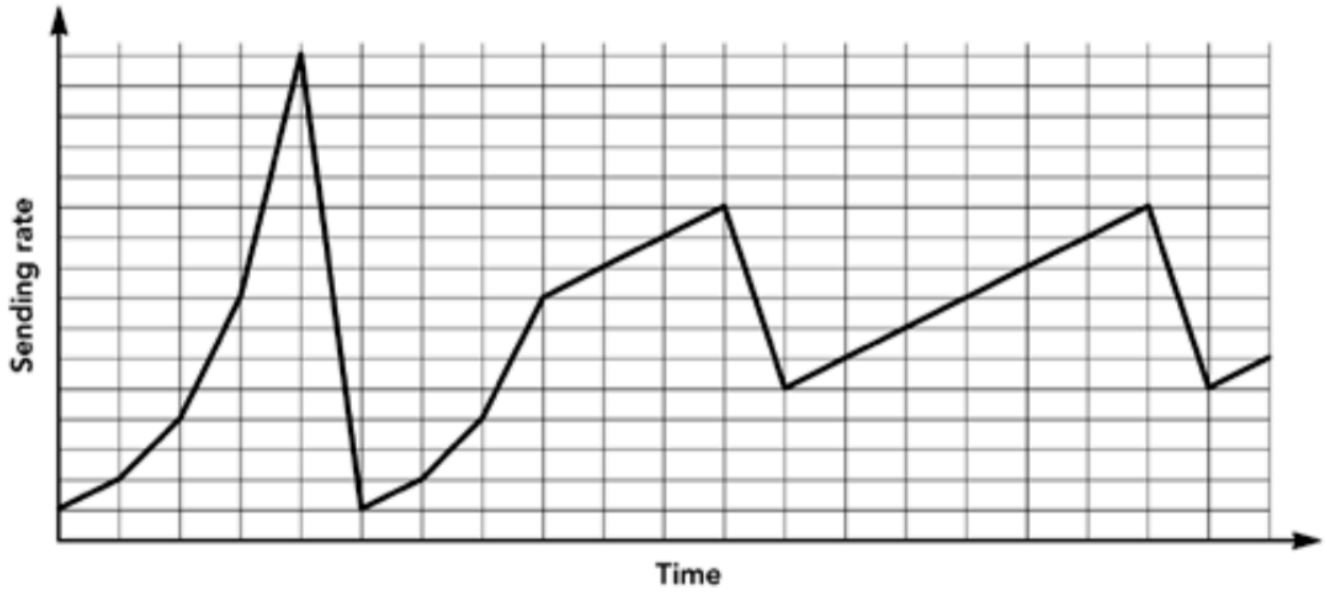

图2.15 样本TCP发送速率

这种重新传输、缓冲和探测可用带宽的组合有以下几个效果:

- TCP传输是可靠的，如果连接保持打开，所有数据最终都将被传递。如果连接失败，则通知连接端失败。这与UDP形成了对比，UDP不向发送方提供关于数据传输状态的信息。
- 应用程序对包传输的时间几乎没有控制，因为在源发送数据的时间和接收数据的时间之间没有必然的关系。这种变化与原始IP服务显示的传输时间变化不同，因为TCP层还必须考虑重新传输和发送速率的变化。发送方可以知道是否所有数据都已发送，这可能使它能够估计平均传输速率。
- 带宽探测可能导致瓶颈链路的短期过载，从而导致数据包丢失。当这种重载导致TCP流的丢失，该流将降低其速率;但是这个过程中它也可能给其他流造成损失。

当然，TCP的行为也有一些微妙之处，关于这个主题已经写了很多。还有一些特性是本讨论还没有涉及到的，比如推送模式和紧急交付，但是这些特性并不影响基本行为。对于我们的目的来说，重要的是注意TCP和UDP之间的根本区别:可靠性(TCP)和及时性(UDP)之间的权衡。

分组网络中音频/视频传输的要求

到目前为止，本章已经详细地探讨了IP网络的特性，并简要地研究了位于它们之上的传输协议的行为。我们现在可以将此讨论与实时音视频传输联系起来，考虑通过IP网络传输媒体流的需求，并确定网络在多大程度上满足这些需求。

当我们将媒体描述为实时的时候，简单讲，我们的意思是接收方在接收到媒体流时就播放它，而不是简单地将完整的媒体流存储在一个文件中以供以后回放。在理想的情况下，在接收端的播放是即时和同步的，尽管在实践中网络会造成一些不可避免的传输延迟。

实时媒体对传输协议的主要要求是网络传输时间的可预测变化。例如，考虑一个以20毫秒帧传输编码语音的IP电话系统:源将每20毫秒传输一个数据包，理想情况下，我们希望这些数据包以相同的间隔到达，这样它们包含的语音可以立即播放出来。传输时间的一些变化可以通过在接收端插入额外的缓冲延迟来调节，但是这只有在变化可以被描述并且接收端能够适应变化的情况下才有可能实现(这个过程在第6章《媒体采集、播放和时序》中有详细的描述)。

一个较低的要求是通过网络可靠地传递所有数据包。显然，可靠的传输是我们期待的，但许多音频和视频应用程序可以容忍一些丢包:在我们的IP电话示例中，单个数据包的丢失将导致1 / 50秒的丢失，如果采用适当的错误隐藏，则几乎无法察觉。由于媒体流的时变特性，一些丢包通常是可以接受的，因为它的影响会随着新数据的到来而迅速得到纠正。可接受的丢包数量取决于应用程序、使用的编码方法和丢包模式。第8章《错误隐藏》，和第

9章《错误恢复》，讨论丢包容错。

这些需求驱动传输协议的选择。很明显，TCP/IP是不合适的，因为它更看重可靠性而不是及时性，而且我们的应用程序需要实时交付。UDP/ ip传输应该是合适的，只要网络的传输时间变化可以被描述，并且丢包率是可以接受的。

标准实时传输协议(RTP)建立在UDP/IP上，提供实时恢复和丢包检测，以支持健壮系统的开发。RTP和相关标准将在本书的其余部分详细讨论。

尽管TCP对实时应用程序有限制，但一些音频/视频应用程序将其用于传输。这样的应用程序尝试估计TCP连接的平均吞吐量，并调整它们的发送速率以匹配。当没有严格的端到端延迟限制，并且应用程序有几秒钟的缓冲时间来处理由TCP重传和拥塞控制引起的传输时间变化时，可以使用这种方法。它对于需要端到端低延迟的交互式应用程序不可靠，因为TCP引起的传输时间变化太大。

使用TCP/IP传输的主要理由是许多防火墙传递TCP连接，但阻塞UDP。随着基于RTP的系统变得更加流行，防火墙变得更加智能，这种情况正在迅速改变。我强烈建议新的应用程序基于UDP/IP的RTP。RTP可以通过允许应用程序调整以适应实时媒体的方式和通过促进互操作性(因为它是开放标准)，来提供更高的质量。

基于分组的音频/视频的好处

在这个阶段，你可能想知道为什么有人会考虑IP网络上的基于分组的音频或视频应用程序。这样的网络显然对实时媒体流的可靠传输提出了挑战。尽管这些挑战是真实存在的，但IP网络具有一些独特的优势，可以在效率和灵活性方面获得显著的收益，这可能会超过其缺点。

使用IP作为实时音频和视频承载服务的主要优点是，它可以提供一个统一的、聚合的网络。这个网络可以用于语音、音乐和视频，也可以用于电子邮件、Web访问、文件和文档传输、游戏等等。因此可以显著节省在基础设施、部署、支持和管理方面的成本。

统一的分组网络使流量的统计复用成为可能。例如，语音活动检测可用于防止分组语音应用程序在静默期间进行传输，而使用TCP/IP作为其传输的流量将适应可用容量的变化。只要谨慎地设计音频和视频应用程序，减少这种适应的影响，应用程序将不会受到不利影响。因此，我们可以实现更高的链路利用率，这在资源有限的系统中是很重要的。

另一个重要的好处是IP多播，它允许将数据以低成本传递给可能很大的一组接收方。通过使传送的成本不受受众多少的影响，IP多播使人们能够负担得起Internet广播和电视以及其他组通信服务。

最后，也许是最引人注目的，基于IP的音视频的情况下，IP支持新的服务。这种融合允许实时音视频和其他应用程序之间进行丰富的交互，使我们能够开发以前不可能开发的系统。

总结

IP网络的特性与传统的电话、音视频分发网络有很大的不同。在设计基于IP的应用程序时，你需要了解这些独特的特性，以使你的系统在这些特性的影响下仍旧保持健壮性。

本书的其余部分将描述这种系统的体系结构，解释RTP及其模型，这个模型用于时间戳恢复和音视频同步、错误纠正和隐藏、拥塞控制、报头压缩、多路复用和隧道以及安全性。