第二部分　IPv6细节

第5章　IPv6的成型

本章介绍了IPv4的更新，描述了新的协议头中各字段及IPv6的地址空间，着重介绍了IPv6中包含的变化和新特性。IPv4拥有两个“帮助”协议：Internet控制报文协议(ICMP)和Internet组管理协议(IGMP)。主机和路由器使用这些协议来报告IP层差错及执行其他功能、诊断等。IPv6中使用的是对ICMP进行升级后的ICMPv6协议，ICMPv6中最初包含了IGMP的功能，但现在看来这些功能可能要由IGMPv2来完成。

本章第一小节描述了IPv6协议的基本框架并介绍了RFC 1883(IPv6技术规范)和其他后续标准(到1998年9月还没有分配RFC号码)中定义的IPv6头字段、选项和扩展。本章第二小节概述了RFC 1885(用于IPv6的Internet控制报文协议(ICMPv6)的技术规范)中定义的ICMPv6。IGMPv2在RFC 2236(Internet组管理协议第2版)中定义，并且与IPv4和IPv6均有关联。

IPv6的地址方案将在第6章中介绍，第7章对IPv6的选项和扩展头有更详细的介绍。第8章将探讨IPv6的选路。第9章将进一步讨论IPv6中的安全性和身份验证问题，第10章将介绍升级到IPv6对IP的上层和下层协议造成的影响。

5.1　IPv6

对IPv4的升级最早在两个RFC中进行了定义。RFC 1883中描述的是协议本身，而RFC 1884介绍的是IPv6的地址结构。现在RFC 1884已经被RFC 2373所替代，1998年夏天IETF批准了一个草案来替换RFC 1883。从32位地址到128位地址的变化代表了一个重大的转变，但如何制定和分配IPv6地址直到1998年秋天也没有定论。第6章将对于IPv6的地址有更详细的介绍。本节只介绍真正的IPv6协议中最重要的改变而不讨论地址细节。

5.1.1　变化概述

IPv6中的变化体现在以下五个重要方面：

- 扩展地址。
- 简化头格式。
- 增强对于扩展和选项的支持。
- 流标记。
- 身份验证和保密。

对于IP的这些改变对IAB于1991年制定的IPv6发展方向中的绝大部分都有所改进。IPv6的扩展地址意味着IP可以继续增长而无需考虑资源的匮乏，该地址结构对于提高路由效率有所帮助；对于包头的简化减少了路由器上所需的处理过程，从而提高了选路的效率；同时，改进对头扩展和选项的支持意味着可以在几乎不影响普通数据包和特殊包选路的前提下适应更

多的特殊需求；流标记办法为更加高效地处理包流提供了一种机制，这种办法对于实时应用尤其有用；身份验证和保密方面的改进使得 IPv6更加适用于那些要求对敏感信息和资源特别对待的商业应用。

1. 扩展地址

IPv6的地址结构中除了把32位地址空间扩展到了128位外，还对IP主机可能获得的不同类型地址作了一些调整。就像在第 6章中将要详细介绍的一样， IPv6中取消了广播地址而代之以任意点播地址。 IPv4中用于指定一个网络接口的单播地址和用于指定由一个或多个主机侦听的组播地址基本不变。

2. 简化的包头

IPv6中包括总长为40字节的8个字段(其中两个是源地址和目的地址)。它与IPv4包头的不同在于， IPv4中包含至少12个不同字段，且长度在没有选项时为 20字节，但在包含选项时可达60字节。IPv6使用了固定格式的包头并减少了需要检查和处理的字段的数量，这将使得选路的效率更高。

包头的简化使得IP的某些工作方式发生了变化。一方面，所有包头长度统一，因此不再需要包头长度字段。此外，通过修改包分段的规则可以在包头中去掉一些字段。 IPv6中的分段只能由源节点进行：该包所经过的中间路由器不能再进行任何分段。最后，去掉 IP头校验和不会影响可靠性，这主要是因为头校验和将由更高层协议 (UDP和TCP)负责。

3. 对扩展和选项支持的改进

在IPv4中可以在IP头的尾部加入选项，与此不同， IPv6中把选项加在单独的扩展头中。通过这种方法，选项头只有在必要的时候才需要检查和处理。下面和第 7章将对此有更多的讨论。

为便于说明，考虑以下两种不同类型的扩展部分：分段头和选路头。 IPv6中的分段只发生在源节点上，因此需要考虑分段扩展头的节点只有源节点和目的节点。源节点负责分段并创建扩展头，该扩展头将放在 IPv6头和下一个高层协议头之间。目的节点接收该包并使用扩展头进行重装。所有中间节点都可以安全地忽略该分段扩展头，这样就提高了包选路的效率。

另一种选择方案中，逐跳(hop-by-hop)选项扩展头要求包的路径上的每一个节点都处理该头字段。这种情况下，每个路由器必须在处理 IPv6包头的同时也处理逐跳选项。第一个逐跳选项被定义用于超长IP包(巨型净荷)。包含巨型净荷的包需要受到特别对待，因为并不是所有链路都有能力处理那样长的传输单元，且路由器希望尽量避免把它们发送到不能处理的网络上。因此，这就需要在包经过的每个节点上都对选项进行检查。

4. 流

在IPv4中，对所有包大致同等对待，这意味着每个包都是由中间路由器按照自己的方式来处理的。路由器并不跟踪任意两台主机间发送的包，因此不能“记住”如何对将来的包进行处理。IPv6实现了流概念，其定义如RFC 1883中所述：

> 流指的是从一个特定源发向一个特定(单播或者是组播)目的地的包序列，源点希望中间路由器对这些包进行特殊处理。

路由器需要对流进行跟踪并保持一定的信息，这些信息在流中的每个包中都是不变的。这种方法使路由器可以对流中的包进行高效处理。对流中的包的处理可以与其他包不同，但无论如何，对于它们的处理更快，因为路由器无需对每个包头重新处理。下一节中将对流和

流标记有更详细的讨论。

5. 身份验证和保密

RFC 1825(IP的安全性体系结构)描述了IP的安全性体系结构，包括IPv4和IPv6。它发表于在1995年8月，目前正在进行修改和更新。1998年3月发表了一个更新版Internet草案。IP安全性的基本结构仍然很坚固，且已经进行了一些显著的改变和补充。这个体系结构以及它在IPv6中如何实现，都将在第9章介绍。

IPv6使用了两种安全性扩展：IP身份验证头(AH)首先由RFC 1826(IP身份验证头)描述，而IP封装安全性净荷(ESP)首先在RFC 1827(IP封装安全性净荷(ESP))中描述。

报文摘要功能通过对包的安全可靠性的检查和计算来提供身份验证功能。发送方计算报文摘要并把结果插入到身份验证头中，接收方根据收到的报文摘要重新进行计算，并把计算结果与AH头中的数值进行比较。如果两个数值相等，接收方可以确认数据在传输过程中没有被改变；如果不相等，接受方可以推测出数据或者是在传输过程中遭到了破坏，或者是被某些人进行了故意的修改。

封装安全性提供机制，可以用来加密IP包的净荷，或者在加密整个IP包后以隧道方式在Internet上传输。其中的区别在于，如果只对包的净荷进行加密的话，包中的其他部分(包头)将公开传输。这意味着破译者可以由此确定发送主机和接收主机以及其他与该包相关的信息。使用ESP对IP进行隧道传输意味着对整个IP包进行加密，并由作为安全性网关操作的系统将其封装在另一IP包中。通过这种方法，被加密的IP包中的所有细节均被隐藏起来。这种技术是创建虚拟专用网(VPN)的基础，它允许各机构使用Internet作为其专用骨干网络来共享敏感信息。

5.1.2 包头结构

在IPv4中，所有包头以32位为单位，即基本的长度单位是4个字节。在IPv6中，包头以64位为单位，且包头的总长度是40字节。IPv6协议为对其包头定义了以下字段：

- 版本。长度为4位，对于IPv6，该字段必须为6。
- 类别。长度为8位，指明为该包提供了某种“区分服务”。RFC 1883中最初定义该字段只有4位，并命名为“优先级字段”，后来该字段的名字改为“类别”，在最新的IPv6 Internet草案中，称之为“业务流类别”。该字段的定义独立于IPv6，目前尚未在任何RFC中定义。该字段的默认值是全0。
- 流标签。长度为20位，用于标识属于同一业务流的包。一个节点可以同时作为多个业务流的发送源。流标签和源节点地址唯一标识了一个业务流。在RFC 1883中这个字段最初被设计为24位，但当类别字段的长度增加到8位后，流标签字段被迫减小长度来作补偿。
- 净荷长度。长度为16位，其中包括包净荷的字节长度，即IPv6头后的包中包含的字节数。这意味着在计算净荷长度时包含了IPv6扩展头的长度。
- 下一个头。这个字段指出了IPv6头后所跟的头字段中的协议类型。与IPv6协议字段类似，下一个头字段可以用来指出高层是TCP还是UDP，但它也可以用来指明IPv6扩展头的存在。
- 跳极限。长度为8位。每当一个节点对包进行一次转发之后，这个字段就会被减1。如果

该字段达到0，这个包就将被丢弃。 IPv4中有一个具有类似功能的生存期字段，但与IPv4不同，人们不愿意在IPv6中由协议定义一个关于包生存时间的上限。这意味着对过期包进行超时判断的功能可以由高层协议完成。

- 源地址。长度为128位，指出了IPv6包的发送方地址。
- 目的地址。长度为128位，指出了IPv6包的接收方地址。这个地址可以是一个单播、组播或任意点播地址。如果使用了选路扩展头（其中定义了一个包必须经过的特殊路由），其目的地址可以是其中某一个中间节点的地址而不必是最终地址。

图5-1中显示了IPv6头的格式。下一节中提供了IPv6头与IPv4头字段间更加详细的比较。

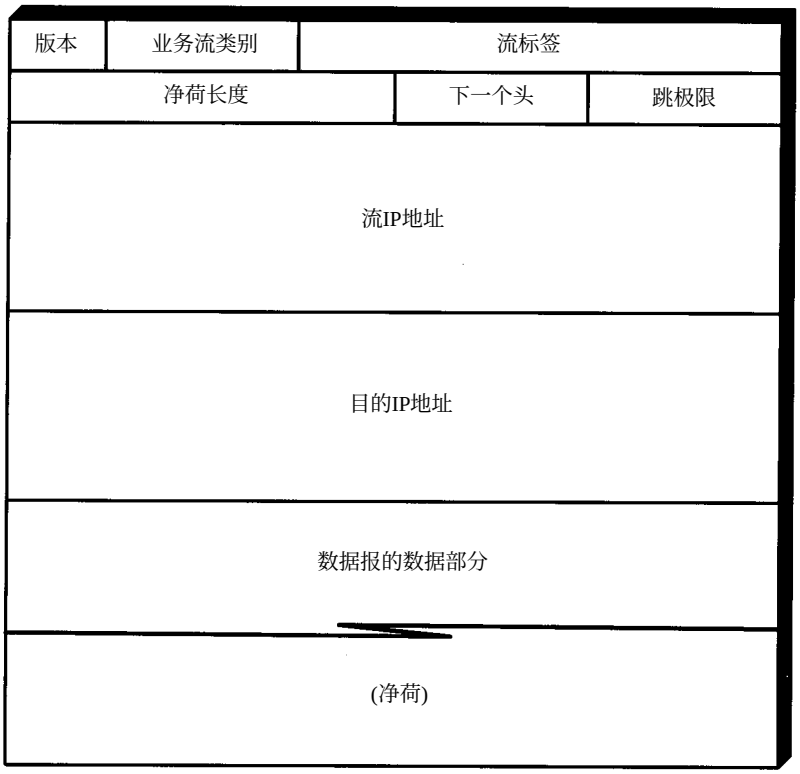

图5-1 Ipv6头比IPv4头(参见图2-3)简单得多

5.1.3 IPv4与IPv6的比较

先回顾一下图2-3中定义的IPv4头。尽管这些头字段中有一些与IPv6头类似，但其中真正完全保持不变的只有第一个字段，即版本字段，因为在同一条线路上传输时，必须保证IPv4和IPv6的兼容性。下一个字段，即包头长度，则与IPv6无关，因为IPv6头是固定长度，IPv4中需要这个字段是因为它的包头可能在20字节到40字节间变化。

服务类型字段与IPv6的流类别字段相似，但TOS的位置比该字段要靠后一些，而且在具体实现中也没有广泛应用。下一个字段是数据报长度，后来发展成了IPv6中的净荷长度。IPv6的净荷长度中包含了扩展头，而IPv4数据报长度字段中则指明包含包头在内的整个数据报的长度。这样一来，在IPv4中，路由器可以通过将数据报长度减去包头长度来计算包的净荷长度，而在IPv6中则无须这种计算。

后面的三个字段是数据报 ID、分段标志和分段偏移值，它们都用于 IPv4数据报的分段。由于IPv6中由源结点取代中间路由器来进行分段 (后面将有更多关于分段的内容)，这些字段在IPv6中变得不重要，并被IPv6从包头中去掉了。

而生存期字段，正如上面所述，变成了跳极限字段。生存期字段最初表示的是一个包穿越Internet时以秒为单位的存在时间的上限。如果生存期计数值变为 0，该包将被丢弃。其原因是包可能会存在于循环路由中，如果没有方法让它消失，它可能会一直选路 (或者直到网络崩溃为止)。在最初的规范中要求路由器根据转发包的时间与收到包的时间的差值 (以秒为单位)来减小生存期的值。在实际情况中，大部分路由器都设计为每次对该值减 1，而不是计算路由器上真正的处理时间。

协议字段，如前所述，指出在 IPv4包中封装的高层协议类型。各协议对应的数值在最新版本的RFC(现在是RFC 1700)中可以查到。这个字段后来发展成为 IPv6中的下一个头字段，其中定义了下一个头是一个扩展头字段还是另一层的协议头。

由于如TCP和UDP等高层协议均计算头的校验和， IPv4头校验显得有些多余，因此这个字段在IPv6中已消失。对于那些真的需要对内容进行身份验证的应用， IPv6中提供了身份验证头。

IPv6中仍然保留了32位的IPv4源地址和目的地址，但将它们扩展为 128位。而IP选项字段则已经彻底消失，取而代之的是IPv6扩展头。

5.1.4 流标签

IPv4通常被描述为无连接协议。就像任何一个包交换网络一样， IPv4设计为让每个包找到自己的路径以到达其目的地。每个包都分别处理，而结果是两个从相同数据源发往相同目的地的包可以采用完全不同的路由来穿越整个网络。这对于适应网络突发事件来说是个好办法，因为突发事件意味着任何一条路由都可能在任何时间出现故障，但只要两主机间存在某些路由则可以进行数据的交互。

但是，这种方法的效率可能不太高，尤其是当包并不是孤立的，且实际上是两个通信系统间的业务流的一部分时。进一步考虑一个包流从一台主机发往另一主机时在它所经过的路径上将发生的事情：每个中间路由器对每个包的处理将导致在链路上轻微地增加延时。对于类似文件传输或终端仿真之类的大部分传统 Internet应用，延时只会带来一点不方便而已，但对于一些提供互操作的音频和视频应用而言，即使只是增加一点点延时也会显著降低服务质量。

对每个IPv4包均进行单独处理带来的另一个问题在于难以把特定的业务流指定到较低代价的链路上。例如，电子邮件的传输优先级不高，并且不是实时应用，但 IPv4网络管理员却没有简单的办法来标识这些包，把它们传输到较低开销的 Internet链路，并为实时应用保留较高开销的链路。

IPv6中定义的流的概念将有助于解决类似问题。 IPv6头字段中的流标签把单个包作为一系列源地址和目的地址相同的包流的一部分。同一个流中的所有包具有相同的流标签。

5.1.5 业务流类别

最早有关IPv6的RFC(1883)中定义了4位优先级字段，这意味着每个包可能具备 16个优先级中的一个。但是，经过多次讨论后这个字段的名字改为“ 类别 ”，且长度也扩大到了1字节。

在最新的关于RFC 1883的Internet修订草案中，名字又被改为“业务流类别”。

IPv6类别字段的数值及如何正确使用还有待定义。使用IPv4服务类型字段和使用IPv6类别的实验最终必将为此带来有用的结果。使用业务流类别的目的在于允许发送业务流的源节点和转发业务流的路由器在包上加上标记，并进行除默认处理方法之外的不同处理。一般来说，在所选择的链路上，可以根据开销、带宽、延时或其他特性而对包进行特殊的处理。

虽然在IPv6的实现中很可能需要并建议高层协议为它们的数据指定一个特定的业务流等级，但这些实现中可能也允许中间路由器根据实际情况修改这个值。

5.1.6 分段

IPv6的分段只能由源节点和目的节点进行，这样就简化了包头并减少了用于选路的开销。逐跳分段被认为是一种有害的方法。首先，它在端到端的分段中将产生更多的分段。此外在传输中，一个分段的丢失将导致所有分段重传。 IPv6的确可以通过其扩展头来支持分段，但是如下所述，了解IPv4分段如何工作将有助于了解IPv6中为什么要进行改变。

在IPv4中，当一个没有分段的包由于太长而无法沿着发送源到目的地的网络链路进行传输时，就需要进行包的分段。举例来说，一个源节点可以创建一个长度为 1500字节的包，并把它向Internet上的某个远端目的地发送。这个包通过源节点的本地以太网到达该节点的默认路由器。然后路由器通过其链路把数据发到Internet上，这条链路可能是到一个ISP的点到点连接。在Internet中的某处或离目的节点较近的某处，可能有条网络链路无法处理这样一大块的数据。在这种情况下，使用该网络链路的路由器将不得不把 1500字节的数据报分割成许多不超过下一个网络的最大传输单元 (MTU)的分段。因此，如果假设下一个链路可以处理的包长度不能超过1280 字节的话，路由器将把最初的一个包分割为两个。第一个包的长度为 1260字节，留下的20字节用于IPv4头。第二段的长度就是剩余数据的长度， 240字节，另外再用20字节作为另一个IPv4头。

IPv4中的分段由包沿途的中间路由器根据需要进行。进行分段的路由器根据需要修改包头并在其中包含进最初的包的数据报标识，同时还将正确地设置分段标志和分段偏移值。当目的节点收到由此产生的分段包之后，该系统必须根据每个分段包的 IPv4头后的分段数据重组最初的包。

在使用了分段之后，不论中间的网络是什么类型，不同类型网络上的节点都可以互操作，源节点无需了解任何有关目的节点网络的信息，同时也无需了解它们之间的网络信息。这一直被认为是一个不错的特性，由于不需要节点或路由器存储信息或记录整个 Internet的结构，从而Internet可以获得很好的扩展性。但另一方面，它也为路由器带来了性能方面的问题，对IP包进行分段消耗了沿途路由器和目的地的处理能力和时间。了解 IP数据报标识、计算分段偏移值、真正把数据分段以及在目的地进行重装都会带来额外的开销。

问题在于对于任何一个指定的路由器，虽然源节点能够了解链路的 MTU是多大，但却没有办法事先知道整个路径的MTU。路径MTU是源节点和目的节点之间在不分段时可以沿着该路由穿越任何网络的最大包长。

然而，目前有两种方法可以减少或消除对于分段的需求。第一种方法可用在 IPv4中，它使用一种叫做“路径MTU发现”的方法。通过这种方法，路由器可以向目的地发送一个包来报告该路由器上链路的MTU值。如果包到达了一条必须对其进行分段的链路，负责分段的路

由器将使用ICMP回送一个报文来指出分段路由器上链路的MTU值。这种过程可以重复进行直到路由器确定路径MTU为止。(后面将有对ICMP的进一步讨论。)

另一种减少分段需求的方法是要求所有支持IP的链路必须能够处理一些合理的最小长度的包。换句话说，如果一个链路的MTU超过20字节，那么所有的节点都必须准备产生可观数量的分段包。另一方面，如果能够提出所有网络链路都可以适应的某个合理的长度，并把它设置为允许包长度的绝对最小值，那么就可以消灭分段。

IPv6中实际上同时使用了上面两种方法。在最初的RFC中，IPv6规定每个链路支持的MTU最小为576字节。那么这些包的净荷长度将是536字节，另外40字节用于IPv6头。由于RFC 1883发表于1995年，后来产生了很多关于更大的MTU的争论。在Huitema提出的报告(参见《IPv6：新的IP》第2版，Prentice-Hall)中，建议值为1997，Steve Deering则正在促使将MTU值改为1500字节。在最新的于1997年11月发表的Internet草案中，MTU值被设为1280字节。很明显，关注的焦点在于：倡导较短MTU的人希望那些不能支持较长MTU的网络不会被完全丢弃，而倡导较长MTU的人不希望为照顾小部分接近于废弃的网络而使得整个Internet的性能下降。

为了对较短的MTU进行一些弥补，IPv6标准中强烈推荐所有IPv6节点都支持路径MTU发现。路径MTU发现最早出现在RFC 1191中，其中使用了分段标志中的“不能分段”来要求中间路由器在发现包太长时返回一个ICMP出错报文。

路径MTU发现的IPv6版本在RFC 1981(IPv6的路径MTU发现)中描述。这是对原有的RFC 1191的升级，但其中加入了一些改变使之可以工作在IPv6中。其中最重要的是，由于IPv6头中不支持分段，因此也就没有“不能分段”位。正在执行路径MTU发现的节点只是简单地在自己的网络链路上向目的地发送允许的最长包。如果一条中间链路无法处理该长度的包，尝试转发路径MTU发现包的路由器将向源节点回送一个ICMPv6出错报文。然后源节点将发送另一个较小的包。这个过程将一直重复，直到不再收到ICMPv6出错报文为止，然后源节点就可以使用最新的MTU作为路径MTU。

这里需要注意，有一些实例并没有实现路径MTU发现。例如，使用最小IPv6实现来进行远程网络启动的终端只是简单地使用576字节的路径MTU。从源节点到目的节点的IPv6分段，作为一个扩展头来实现，将在下一节中讨论。

5.1.7 扩展头

IPv4选项的问题在于改变了IP头的大小，因此更像一个“特例”，即需要特别的处理。路由器必须优化其性能，这意味着将为最普遍的包进行最佳性能的优化。这使得IPv4选项引发一个路由器把包含该选项的包搁置一边，等到有时间的时候再进行处理。

IPv6中实现的扩展头可以消灭或至少大量减少选项带来的对性能的冲击。通过把选项从IP头中搬到净荷中，路由器可以像转发无选项包一样来转发包含选项的包。除了规定必须由每个转发路由器进行处理的逐跳选项之外，IPv6包中的选项对于中间路由器而言是不可见的。

可用的选项

除了减少IPv6包转发时选项的影响外，IPv6规范使得对于新的扩展和选项的定义变得更加简单。在需要的时候可能还会定义其他的选项和扩展。本节仅列出已定义的扩展，而对于扩展头和选项的使用在第7章中将有更详细的讨论，安全性头将在第9章讨论。RFC 1883中为

IPv6 定义了如下选项扩展：

- 逐跳选项头。此扩展头必须紧随在 IPv6头之后。它包含包所经路径上的每个节点都必须检查的选项数据。由于它需要每个中间路由器进行处理，逐跳选项只有在绝对必要的时候才会出现。到目前为止，已经定义了两个选项：巨型净荷选项和路由器提示选项。巨型净荷选项指明包的净荷长度超过 IPv6的16位净荷长度字段。只要包的净荷超过 65 535字节(其中包括逐跳选项头)，就必须包含该选项。如果节点不能转发该包，则必须回送一个ICMPv6出错报文。路由器提示选项用来通知路由器， IPv6数据报中的信息希望能够得到中间路由器的查看和处理，即使这个包是发给其他某个节点的 (例如，包含带宽预留协议信息的控制数据报)。
- 选路头。此扩展头指明包在到达目的地途中将经过哪些节点。它包含包沿途经过的各节点的地址列表。IPv6头的最初目的地址是路由头的一系列地址中的第一个地址，而不是包的最终目的地址。此地址对应的节点接收到该包之后，对 IPv6头和选路头进行处理，并把包发送到选路头列表中的第二个地址。如此继续，直到包到达其最终目的地。
- 分段头。此扩展头包含一个分段偏移值、一个“更多段”标志和一个标识符字段。用于源节点对长度超出源端和目的端路径 MTU的包进行分段。
- 目的地选项头。此扩展头代替了 IPv4选项字段。目前，唯一定义的目的地选项是在需要时把选项填充为64位的整数倍。此扩展头可以用来携带由目的地节点检查的信息。
- 身份验证头(AH)。此扩展头提供了一种机制，对 IPv6头、扩展头和净荷的某些部分进行加密的校验和的计算。
- 封装安全性净荷 (ESP)头。这是最后一个扩展头，不进行加密。它指明剩余的净荷已经加密，并为已获得授权的目的节点提供足够的解密信息。

5.2 ICMPv6

IP节点需要一个特殊的协议来交换报文以了解与 IP相关的情况。ICMP正好适用于这种需求。在IPv4升级到IPv6的过程中，ICMP也经历了一定的修改。ICMPv6在RFC 1885中定义。ICMP报文可以用来报告错误和信息状态，以及类似于包的 Internet探询(Ping)和跟踪路由的功能。

IGMP一开始就包含在ICMPv6规范中，并且在1997年11月发表的RFC 2236中得到更新，1998年初秋，IGMP第3版也开始了讨论。IGMP可以用来支持组播传输，它为主机提供了向本地路由器报告其属于某个组播组的方法。

ICMPv6报文

ICMP报文的产生来源于一些错误情况。例如，如果一个路由器由于某些原因不能处理一个IP包，它就可能会产生某种类型的 ICMP报文，并直接回送到包的源节点，然后源节点将采取一些办法来纠正所报告的错误状态。例如，如果路由器无法处理一个 IP包的原因是由于包太长而无法将其发送到网络链路上，则路由器将产生一个 ICMP错误报文来指出包太长，源节点在收到该报文后可以用它来确定一个更加合适的包长度，并通过一系列新的 IP包来重新发送该数据。

RFC 1885中定义了以下报文类型(没有包括该文档中定义的有关组的报文)：

- 目的地不可达。
- 包太长。
- 超时。
- 参数问题。
- 回声请求。
- 回声应答。

下面将详细介绍这些报文。

1. 目的地不可达

这个报文由路由器或源主机在由于除业务流拥塞之外的原因而无法转发一个包的时候产生。这种错误报文有五个代码，包括：

- 0：没有到达目的地的路由。这个报文在路由器没有定义 IP包的目的地路由时产生，路由器将采用默认路由来发送无法利用路由器的路由表进行转发的包。
- 1：与目的地的通信被管理员禁止。当被禁止的某类业务流欲到达防火墙内部的一个主机时，包过滤防火墙将产生该报文。
- 2：不是邻居。当使用 IPv6选路扩展头并严格限定路由时，将使用这个代码。当列表中的下一个目的地与当前正执行转发的节点不能共享一个网络链路时，将会产生该报文。
- 3：地址不可达。这个代码指出在把高层地址解析到链路层 (网络)地址时遇到了一些问题，或者在目的地网络的链路层上去往其目的地时遇到了问题。
- 4：端口不可达。这种情况发生在高层协议 (如DP)没有侦听包目的端口的业务量，且传输层协议又没有其他办法把这个问题通知源节点时。

2. 包太长

当接收某包的路由器由于包长度大于将要转发到的链路的 MTU，而无法对其进行转发时，将会产生包太长报文。该ICMPv6错误报文中有一个字段指出导致该问题的链路的 MTU值。在路径MTU发现过程中这是一个有用的错误报文。

3. 超时

当路由器收到一个跳极限为 1的包时，它必须在转发该包之前减小这个数值。如果在路由器减小该数值后，跳极限字段的值变为 0(或者是路由器收到一个跳限制字段为 0的包)，那么路由器必须丢弃该包，并向源节点发送 ICMPv6超时报文。源节点在收到该报文后，可以认为最初的跳限制设置得太小(包的真实路由比源节点想象的要长)，也可以认为有一个选路循环导致包无法交付。

在“跟踪路由”功能中这个报文非常有用。这个功能使得一个节点可以标识一个包在从源节点到目的节点的路径上的所有路由器。它的工作方式如下：首先，一个去往目的地的包的跳极限被设置为1。它所到达的第一个路由器将跳减少极限，并回送一个超时报文，这样一来源节点就标识了路径上的第一个路由器。然后如果该包必须经过第二个路由器的话，源节点会再发送一个跳极限为 2的包，该路由器将把跳极限减小到 0，并产生另一个超时报文。这将持续到包最终到达其目的地为止。同时源节点也获得了从每个中间路由器发来的超时报文。

4. 参数问题

当IPv6头或扩展头中的某些部分有问题时，路由器由于无法处理该包而会将其丢弃。路由器的实现中应该可以产生一个 ICMP参数错误报文来指出问题的类型 (如错误的头字段、无

法识别的下一个头类型或无法识别的 IPv6选项)，并通过一个指针值指出在第几个字节遇到这种错误情况。

5. ICMPv6回声功能

ICMPv6中包含了一个与错误情况无关的功能。所有 IPv6节点都需要支持两种报文：回声请求和回声应答。回声请求报文可以向任何一个正确的 IPv6地址发送，并在其中包含一个回声请求标识符、一个顺序号和一些数据。尽管二者都是可选项，但回声请求标识符和顺序号可以用来区分对应不同请求的响应。回声请求的数据也是一个选项，并可用于诊断。

当一个IPv6节点收到一个回声请求报文后，它必须回送一个回声应答报文。在应答中包含相同的请求标识符、顺序号和在最初的请求报文中携带的数据。

ICMP回声请求/应答报文对是ping功能的基础。ping是一个重要的诊断功能，因为它提供了一种方法来决定一个特定的主机是否与其他一些主机连接在相同的网络上。