

中华人民共和国通信行业标准

YD/T 1341-2005

IPv6 基本协议——IPv6 协议

Internet Protocol version6(IPv6) specification

2005-05-11 发布

2005-11-01 实施

中华人民共和国信息产业部 发布

目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 定义及缩略语	1
3.1 定义	1
3.2 缩略语	2
4 概述	2
5 IPv6 头格式	3
6 IPv6 扩展头	4
6.1 扩展头的顺序	5
6.2 选项	5
6.3 逐跳选项头	6
6.4 路由头	7
6.5 分段头	9
6.6 目的地选项头	11
6.7 无下一个头	12
7 IPv6 数据包的长度	12
8 流标签	13
9 业务等级	13
10 上层协议	13
10.1 上层校验和	13
10.2 数据包最大生存时间	14
10.3 最大上层载荷长度	14
10.4 响应承载路由头的数据包	14
附录 A (规范性附录) 流标签的语义及使用	15
附录 B (资料性附录) 选项格式准则	16

前 言

本标准修改采用 IETF 的 RFC 2460 (1998), 主要差异如下:

1. 按照 GB 1(2000) 系列的要求对标准格式进行了修改;
2. 将一些适用于国际标准的表述改为我国标准的表述;
3. 增加了 3.2 节;
4. 将标准附录 B 改为资料性附录。

本标准是“IPv6 协议”系列标准之一, 该系列标准预计的结构及名称如下:

1. 《IPv6 基本协议——IPv6 协议》
2. 《IPv6 技术要求——支持计算机移动部分》
3. 《IPv6 技术要求——地址、过渡及服务质量》
4. 《IPv6 地址结构协议——IPv6 无状态地址自动配置》
5. 《IPv6 邻居发现协议——基于 IPv6 的邻居发现协议》
6. 《IPv6 协议—一致性测试方法》

本标准的附录 A 为规范性附录, 附录 B 为资料性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位: 信息产业部电信研究院

本标准主要起草人: 赵 锋 赵 锴 张德华 杜和青

IPv6 基本协议——IPv6 协议

1 范围

本标准规定了 IPv6 协议，包括 IPv6 头格式、扩展头格式、数据包的长度、流标签、业务等级、对上层协议的影响等内容。有关 IPv6 的安全性特征不在本标准的讨论范围之内。

本标准适用 IPv6 设备开发、系统组建。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

- RFC 791 (1981) 互联网协议规范版本 4 (IPv4)
- RFC 1661 (1994) 点到点协议 (the Point-to-Point Protocol, PPP)
- RFC 1981 (1996) IPv6 的路径最大传输单元发现 (Path MTU Discovery for IP Version 6)
- RFC 2373 (1998) IPv6 地址结构 (IP Version 6 Addressing Architecture)
- RFC 2401 (1998) IP 协议的安全体系结构 (Security Architecture for the Internet Protocol)
- RFC 2402 (1998) IP 认证头 (IP Authentication Header)
- RFC 2406 (1998) IP 封装安全有效载荷 (IP Encapsulating Security Payload, ESP)
- RFC 2463 (1998) IPv6 的互联网控制消息协议 (ICMP for the IPv6)

3 定义及缩略语

3.1 定义

下列定义和缩略语适用于本标准。

3.1.1

节点 node
实现 IPv6 的设备。

3.1.2

路由器 router
负责转发（目标地址不是它本身）IPv6 数据包的节点。

3.1.3

主机 host
除路由器外的任何节点。

3.1.4

上层 upper layer
紧接在 IPv6 之上的协议层。例如，传输层协议（如 TCP 或 UDP），控制协议（如 ICMP），路由协议（如 OSPF）以及通过隧道方式封装在 IPv6 中的互联网层或低层协议（如 IPX、AppleTalk、IPv4 甚至 IPv6）。

3.1.5

链路 link
是通信设备或媒体。节点可以通过链路在数据链路层（紧接在 IPv6 的下层）进行通信。例如，

Ethernet、PPP 链路、帧中继、ATM 网络以及互联网（或更高）层的隧道（如在 IPv4 或 IPv6 上的隧道）。

3.1.6

邻居 neighbors

连接在同一链路上的不同节点，这些节点之间的数据包传输不通过路由器转发。

3.1.7

接口 interface

节点到链路的连接点。

3.1.8

地址 address

一个或一组接口的 IP 层标识。

3.1.9

数据包 packet

IPv6 头和有效载荷构成的数据块。

3.1.10

链路最大传输单元 link MTU

能通过链路完整传输的数据包的最大长度，以字节为单位。

3.1.11

路径最大传输单元 path MTU

源节点和目的节点之间的一条路径上所有链路最大传输单元的长度中的最小值。

注：有一些可能的情况，对于一个具有多个接口的设备作如下设置：转发来自于某些接口且不是以它自身为目的地的数据包；丢弃来自于另一些接口且不是以它自身为目的地的数据包。这时，该设备在接收来自转发接口的数据包，或在转发接口与邻居节点交互时，要遵守路由器的协议要求；而在接收来自非转发接口的数据包，或在非转发接口与邻居节点交互时，要遵守主机的协议要求。

3.2 缩略语

下列缩略语适用于本标准。

IETF	Internet Engineering Task Force	互联网工程任务组
ICMP	Internet Control Message Protocol	互联网控制消息协议
IP	Internet Protocol	互联网协议
IPv6	Internet Protocol Version 6	互联网协议版本 6
MTU	Maximum Transmission Unit	最大传输单元
MSS	Maximum Segment Size	最大分段长度
TCP	Transmission Control Protocol	传输控制协议
TLV	Type—Length—Value	类型—长度—值
UDP	User Datagram Protocol	用户数据报协议

4 概述

互联网协议第六版（IPv6）是互联网协议的一个新的版本。IPv6 相对于 IPv4 主要改变如下：

- 扩展的寻址能力

IPv6 把 IP 地址空间从 32 比特增加到了 128 比特，从而能够支持更多层次的寻址结构，更多的可寻址节点的数量，以及更为简化的地址的自动配置。IPv6 通过在组播地址中加入了一个“范围”域而提高了组播选路的扩展性。在 IPv6 中还定义了一种称为“泛播地址”的新的地址类型，它被用来向一组节点中的任一个节点发送数据包（注）。

- 简化的头格式

IPv6 省略了一些 IPv4 头中的域或将其改成了可选项，从而减少了数据包的公共处理开销，并减少了

IPv6 头所占的带宽开销。

- 对扩展和选项的增强支持

IPv6 在 IP 头选项的编码方式上作了一些变化,其目的是更有效地进行转发,并放宽了对选项长度的严格限制,为将来加入新选项提供更大的灵活性。

- 流标签能力

为了满足发送者所要求的特殊处理,IPv6 增加了一个新的域来标记属于特殊传输数据流的数据包,例如,非缺省的服务质量或实时业务等。

- 认证和保密能力

IPv6 规定了包括认证、数据完整性和数据加密(可选)在内的扩展功能。

本标准规定了基本的 IPv6 头和最初定义的 IPv6 扩展头及选项。本标准还讨论了数据包的长度问题,流标签和业务等级的语义,以及 IPv6 对上层协议的影响。IPv6 地址的格式和语义在 RFC 2373 中分别进行了规定。在 RFC 2463 中规定了 IPv6 版的 ICMP,它是所有 IPv6 实现所必须包括的。

注:泛播地址(Anycast Address)是分配给一组接口的地址,该组接口可以属于不同的节点,以泛播地址为目的地址的数据包会被转发到根据路由协议测量的距离最近的一个接口上。

5 IPv6 头格式

IPv6 的头格式如下所示。

0	4	12	31
版本	业务等级	流标签	
载荷长度		下一个头	跳数限制
源地址			
目的地址			

各域含义如下:

- 版本 (Version): 该域长度为 4 比特, IPv6 版本号 6。
- 业务等级 (Traffic Class): 该域长度为 8 比特, 参见第 9 章。
- 流标签 (Flow Label): 该域长度为 20 比特, 参见第 8 章。
- 载荷长度 (Payload Length): 该域为 16 比特的无符号整数, 表示 IPv6 载荷长度, 即数据包中 IPv6 头之后其余部分的长度, 以字节为单位 (注 1: 任何的扩展头都被认为是载荷的一部分, 其长度应被计算在内)。
- 下一个头 (Next Header): 该域长度为 8 比特, 表示紧接在 IPv6 头后面的下一个头的类型。这个域取不同的值, 对应的扩展头类型不同, 值与扩展头类型之间的对应关系见“IANA 协议值与指定服务网页”(注 2)。
- 跳数限制 (Hop Limit): 该域为 8 比特无符号整数, 数据包每向前经过一个转发节点, 跳数限制减 1, 当跳数限制减至 0 时, 该数据包被丢弃。
- 源地址 (Source Address): 该域长度为 128 比特, 表示产生数据包的节点的 IPv6 地址。
- 目的地址 (Destination Address): 该域长度为 128 比特, 表示期望数据包到达的 IPv6 地址, 如果出现路由头, 这个地址可能不是最终的接收数据包的 IPv6 地址。

注 2: IANA (Internet Assigned Numbers Authority, 互联网地址分配机构) 管理互联网运行所必须的许多特殊参数和协议值, 包括特殊端口号的指派和字符集的注册, 下面我们称这些特殊参数和协议值为特殊号码。以前, IANA 通过系列 RFC 文档发布这些特殊号码, 最后的文档 RFC 1700 也已经过时; 现在 IANA 将这些特殊号码的最新集合列在名为 IANA 协议值与指定服务 (Protocol Numbers and Assignment Services) 的网页上供查阅, 网址是 <http://www.iana.org/numbers.html>, 并在

批准新的特殊号码和指定服务时随时更新这个网页的内容。

6 IPv6 扩展头

在 IPv6 中, 可选的互联网层信息被编码在单独的头中, 并放在一个数据包内的 IPv6 头和上一层头之间。这种扩展头的数量不多, 每个扩展头都被一个明确的“下一个头”域的值所确定, 如下所示, 每个 IPv6 数据包可带有 0 个、1 个或多个扩展头, 每个扩展头由前一个头的“下一个头”域所确定。

IPv6 头 下一个头=TCP	TCP 头+数据
--------------------	----------

IPv6 头 下一个头=路由	路由头 下一个头=TCP	TCP 头+数据
-------------------	-----------------	----------

IPv6 头 下一个头=路由	路由头 下一个头=分段	分段头 下一个头=TCP	TCP 头+数据
-------------------	----------------	-----------------	----------

除了逐跳选项头之外的其他扩展头不被数据包发送路径上的任何一个节点检查或处理, 除非是数据包到达了 IPv6 头中“目的地址”域所指明的节点 (或在组播路由的情况下, 节点组中的任一个节点)。在对 IPv6 头中“下一个头”域正常解复用时首先要处理第一个扩展头 (当没有扩展头时直接处理上层头)。每一个扩展头的内容和语义决定了是否要继续处理下一个头。因此, 必须严格按照扩展头在数据包中出现的顺序对它们进行处理。接收者不能在数据包中搜索一个特定的扩展头, 并且不能在处理完所有排在它前面的头之前处理它。

逐跳选项头中携带的信息必须被数据包传送路径上包括源节点和目的节点在内的每一个节点检查和处理。逐跳选项头如果存在, 则它必须紧随在 IPv6 头之后。当 IPv6 头中“下一个头”域的值不为 0 时, 说明后面有逐跳选项头存在。

如果节点处理一个头的结果是要进行下一个头的处理, 但这个头的“下一个头”域的值不能被节点所识别, 则节点将丢弃这个数据包并向数据包的源节点发送一个 ICMP “参数错误”消息, ICMP 代码值为 1 (不能识别下一个头的类型), ICMP 指针域包含源数据包中不能被识别的域的偏移量。若一个节点遇到除 IPv6 头外的任一个头的“下一个头”域为 0, 则节点对这个数据包也应按上面的方法进行处理。

每个扩展头的长度应为 8 的整数倍 (以字节为单位), 以保证下面的头也按 8 个字节对齐。每个扩展头内的多字节域按它们的自然分界来对齐。

IPv6 的完整实现包括下面扩展头的实现:

- 逐跳选项;
- 路由 (类型 0);
- 分段;
- 目的地选项;
- 认证 (注 1, 注 3);
- 封装安全载荷 (注 2, 注 3)。

前 4 个头在本标准中规定, 后两个分别在 RFC 2402 和 RFC 2406 中规定, 不属于本标准的讨论范围。

注 1: 认证头用于为 IP 数据报提供无连接完整性和数据初始认证, 此外还能防止重发攻击的发生。

注 2: 封装安全载荷头用于提供机密性、数据初始认证、无连接完整性、防止重发攻击以及受限的数据流机密性。

注 3: 认证头和封装安全载荷头可以结合使用, 也可以通过使用隧道模式嵌套使用。它们可以在主机之间、安全网管之间或安全网管与主机之间提供安全服务。封装安全载荷头还可以提供机密性 (加密) 服务。它们之间的主要区别在于覆

盖的范围不同。此外，如果 IP 头不通过封装安全载荷头来封装（以隧道模式），那么封装安全载荷头将不会保护任何的 IP 头中的域。

6.1 扩展头的顺序

当一个数据包中使用多个扩展头时，这些头应按照下面的顺序出现：

- IPv6 头；
- 逐跳选项头；
- 目的地选项头（注 1）；
- 路由头；
- 分段头；
- 认证头（注 2）；
- 封装安全载荷头（注 2）；
- 目的地选项头（注 3）；
- 上层头。

注 1：这些选项要在 IPv6 目的地址域所列出的第一个目的地进行处理，也要在路由头所列出的后续目的地进行处理。

注 2：在 RFC 1827 中给出了有关认证头和封装安全载荷头之间的相对顺序的附加建议。

注 3：这些选项只在数据包的最目的地进行处理。

同一类型的扩展头最多只能出现一次（如果有多个同种扩展头，它们应顺次、连续地排列在一起），惟一例外是目的地选项头可以出现两次，一次在路由头前出现，另一次在上层头前出现。

如果上层头是另一个 IPv6 头（即 IPv6 通过隧道方式封装在 IPv6 中），那么接在它后面的是它自己的扩展头，这些扩展头也应按照上面规定的顺序来排列。

如果要定义其他的扩展头，则必须要说明它们同以上所列的头的顺序约束关系。

IPv6 的节点必须接受并处理同一个数据包中以任何顺序、任何次数出现的扩展头，只有逐跳选项头才必须严格地接在 IPv6 头之后。然而，我们强烈建议数据包的发送者严格遵守上面建议的顺序，除非以后的规范推翻这一顺序。

6.2 选项

在前面介绍的扩展头中，逐跳选项头和目的地选项头可携带不定数量的选项。这些选项采用 TLV 编码方式，格式如下：

选项类型	选项数据长度	选项数据
------	--------	------

选项类型（Option Type）：无符号的 8 位整数，说明选项的类型。

选项数据长度（Option Data Length）：无符号的 8 位整数，以字节为单位，表示选项数据的长度。

选项数据（Option Data）：可变量长度域，包含“选项类型”的数据。

接收者在处理一个头时，必须严格按照每个选项在头中出现的顺序来处理。例如，不能在头中搜索出一个选项并在处理排在它前面的选项之前处理它。

在内部编码时，“选项类型”域的最高位两比特指明了当 IPv6 节点不能识别该选项类型时，所必须采取的动作：

00— 跳过这个选项并继续处理该头；

01— 丢弃这个数据包；

10— 丢弃这个数据包，并且无论这个数据包的目的地址是否是组播地址，都向该数据包的源地址发送一个 ICMP 数据包，指出不能识别的选项类型；

11— 丢弃这个数据包，并且只有当目的地址不是组播地址时，向该数据包的源地址发送一个 ICMP “参数错误”消息，代码值为 2，指针域指向不能识别的选项类型。

“选项类型”域的第三位指明这个选项的数据是否能改变数据包到达最终目的地的路由。当一个数据包中有认证头时，对于选项数据可能改变选路的任何选项，在计算或验证数据包的认证值时，这个选项

的整个数据域必须当作 0 值来处理。

0— 选项数据不改变选路；

1— 选项数据可能改变选路。

上述的 3 个高位比特应被视为选项类型的一部分，而不应独立于选项类型。也就是说，应由一个完整的 8 比特选项类型来标识一个特别的选项，而不能仅由选项类型的低位 5 个比特来标识。

逐跳选项头和目的地选项头都使用相同的选项类型编号空间。然而，一个特别的选项可能会受到限制只能用于这两个头中的一个。

个别的选项有特殊的对齐要求，以确保选项数据域中的多字节值符合自然分界。一个选项的对齐要求是用 $xn+y$ 表示的。也就是说，选项类型必须是在从该扩展头开始算的 x 字节的整数倍加上 y 个字节的位置出现。

例如：

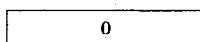
$2n$ 指从该扩展头开始的任何 2 字节偏移。

$8n+2$ 指从该扩展头开始的任何 8 字节偏移，加上 2 字节。

有两类填充选项，当需要时用于后续选项的排列，以填充该头使其长度为 8 字节的整数倍。所有 IPv6 节点必须能识别这些填充选项：

Pad1 选项结构（对齐要求：无）

0 7

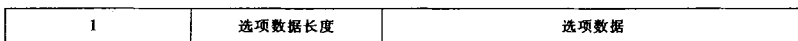


注：Pad1 选项是一个特例，它没有长度域和值域，只是一个 8 比特的 0 值码。

Pad1 选项用于在扩展头的选项域中填充一个字节的长度。当填充量多于一个字节时，要用 PadN 选项，而不是多个 Pad1 选项。

PadN 选项结构（对齐要求：无）

0 7 15



PadN 选项用于在扩展头的选项域中填充 2 个或多个字节。如填充 n 个字节，则“选项数据长度”域的值为 $n-2$ ，“选项数据”域是 $n-2$ 个 0 值字节。

附录 B 包括了设计新选项的格式准则。

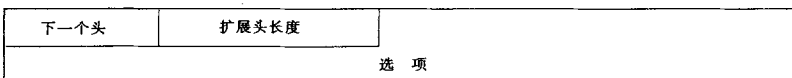
6.3 逐跳选项头

逐跳选项头用来携带那些在数据包发送路径上必须由每个节点检查的信息。如果 IPv6 头的“下一个头”域的值 0，则说明紧接着 IPv6 头的下一个头是逐跳选项头。

逐跳选项头的格式为：

0

31



下一个头（Next Header）：该域长度为 8 比特，定义紧接在逐跳选项头之后的头的类型。这个域取不同的值，对应的扩展头类型不同，值与扩展头类型之间的对应关系见“IANA 协议值与指定服务网页”。

扩展头长度（Hdr Ext Len）：该域是一个 8 比特无符号整数，以 8 个字节为单位表示逐跳选项头的长度，它不包括起始的 8 个字节。

选项域（Options）：该域长度可变，其长度应保证整个逐跳选项头的长度为 8 个字节的整数倍长，包含有一个或多个 TLV 编码的选项。

在本标准中只为逐跳选项头规定了 5.2 节中定义的 Pad1 和 PadN 选项。

6.4 路由头

IPv6 源数据包使用路由头来列出数据包从源地址到目的地址之间需要访问的一个或多个中间节点。该功能非常类似于 IPv4 的松散源选项和记录路由选项。如果某一个头的“下一个头”域值为 43，则说明紧接着它的下一个头是路由头。

路由头的格式如下：

0

31

下一个头	扩展头长度	路由类型	剩余段
与类型相关的数据			

下一个头 (Next Header)：该域长度为 8 比特，定义紧接在路由头之后的头的类型。这个域取不同的值，对应的扩展头类型不同，值与扩展头类型之间的对应关系见“IANA 协议值与指定服务网页”。

扩展头长度 (Hdr Ext Len)：该域为 8 比特无符号整数，以 8 个字节为单位表示路由头的长度，不包括起始的 8 个字节。

路由类型 (Routing Type)：该域长度为 8 比特，标识不同类型的路由头。

剩余段 (Segments Left)：该域为 8 比特无符号整数，表示剩余的路由段的数量，即在到达最终目的节点之前已经列出的但尚未访问的中间节点的数目。

与类型相关的数据 (Type-Specific Data)：该域长度可变，格式由“路由类型”决定，长度应保证整个路由头的长度是 8 个字节的整数倍。

节点在处理接收到的数据包时，如果遇到一个路由头包含有不能识别的“路由类型”值，则节点要依据“剩余段”域的值采取措施。具体方法如下所述：

a) 如果“剩余段”的值为 0，则节点忽略这个路由头，继续处理数据包中的下一个头（其类型由路由头的“下一个头”域的值标识）。

b) 如果“剩余段”的值不为 0，则节点必须丢弃这个数据包，并且向数据包的源地址发送一个 ICMP“参数错误”消息（代码值为 0），ICMP 指针指向不能识别的“路由类型”。

如果一个中间节点在处理完接收数据包的路由头后，决定应将该数据包转发到一条链路 MTU 小于该包长度的链路上，那么该节点必须丢弃此数据包并向该包的源地址发送一个 ICMP“数据包过大”消息。

下面是 0 型路由头的格式：

0

31

下一个头	扩展头长度	路由类型=0	剩余段
保留域			
地址 [1]			
地址 [2]			
⋮			
地址 [n]			

下一个头 (Next Header): 该域长度为 8 比特, 定义紧接在路由头之后的头的类型。这个域取不同的值, 对应的扩展头类型不同, 值与扩展头类型之间的对应关系见“IANA 协议值与指定服务网页”。

扩展头长度 (Hdr Ext Len): 该域为 8 比特无符号整数, 以 8 个字节为单位表示路由头的长度, 不包括起始的 8 个字节。对于 0 型路由头, 扩展头长度是头内地址数的两倍。

路由类型 (Routing Type): 该域长度为 8 比特, 值为 0。

剩余段 (Segments Left): 该域为 8 比特无符号整数, 它表示剩余的路由段的数量, 即在到达最终目的节点之前已经列出的但尚未访问的中间节点的数目。

保留域 (Reserved): 该域为 32 比特保留域, 传输时初始值设为 0, 接收方可忽略此域。

地址 [1..n] (Address): 编号从 1~n 的每一个地址域均为 128 位地址矢量。

0 型路由头内不能有组播地址, 如果一个 IPv6 数据包带有 0 型路由头, 则其最终目的地址域不能是组播地址。

一个路由头只有当它到达 IPv6 的“目的地址”域所指明的节点时, 才能被检查或处理。对于 0 型路由头, 节点应按如下算法进行处理:

```

if 剩余段=0 {
    继续处理数据包的下一个头, 其类型由路由头的“下一个头”域的值所标识。
}
else if 扩展头长度为奇数 {
    向源地址发送一个 ICMP “参数错误”消息 (代码为 0), ICMP 消息指针指向“扩展头长度”域, 并
    丢弃该数据包。
}
else {
    计算路由头内的地址数 n, n=扩展头长度/2。
    if 剩余段>n {
        向源地址发送一个 ICMP “参数错误”消息 (代码为 0), ICMP 消息指针指向“剩余段”
        域, 并丢弃该数据包。
    }
    else {
        “剩余段”减 1,
        计算地址段中下一个将要被访问的地址标号 i,
        i=n-剩余段,
        if 地址 [i] 或 IPv6 最终地址是组播地址 {
            丢弃该数据包。
        }
        else {
            交换 IPv6 目的地址和地址 [i],
            if IPv6 “跳数限制” <=1 {
                向源地址发送一个 ICMP “超时-传输中超过‘跳数限制’”消息, 并丢弃数据包。
            }
            else {
                “跳数限制”减 1,
                重新把数据包交给 IPv6 模块, 以传输到新的目的地。
            }
        }
    }
}
}

```

}

例如，假定从源节点 S 向目的节点 D 发送一个数据包，使用一个路由头，以使该数据包的路由经过中间节点 I1、I2 和 I3。在每一段传输路径上 IPv6 头和路由头中相关域的值如下：

当数据包从 S 传送到 I1 时：

源地址=S 扩展头长度=6
 目的地址=I1 剩余段=3
 地址 [1] =I2
 地址 [2] =I3
 地址 [3] =D

当数据包从 I1 传送到 I2 时：

源地址=S 扩展头长度=6
 目的地址=I2 剩余段=2
 地址 [1] =I1
 地址 [2] =I3
 地址 [3] =D

当数据包从 I2 传送到 I3 时：

源地址=S 扩展头长度=6
 目的地址=I3 剩余段=1
 地址 [1] =I1
 地址 [2] =I2
 地址 [3] =D

当数据包从 I3 传送到 D 时：

源地址=S 扩展头长度=6
 目的地址=D 剩余段=0
 地址 [1] =I1
 地址 [2] =I2
 地址 [3] =I3

6.5 分段头

IPv6 源节点使用分段头来发送数据包长度比路径 MTU 大的数据包（注：与 IPv4 不同，IPv6 的分段由源节点完成，而不是由数据包发送路径上的路由器完成。参见第 7 章）。如果一个头的“下一个头”域的值 44，则说明紧接在它后面的一个头是分段头。

分段头的格式如下：

0

31

下一个头	保留域	分段偏移	保留	M
标识				

下一个头 (Next Header)：该域长度为 8 比特，定义紧接在分段头之后的头的类型。这个域取不同的值，对应的扩展头类型不同，值与扩展头类型之间的对应关系见“IANA 协议值与指定服务网页”。

保留域 (Reserved)：该域长度为 8 比特，传输时初始值设为 0，接收方忽略此域。

分段偏移 (Fragment Offset)：该域长度为 13 比特，以 8 个字节为单位，表示该头后面的数据相对于原始数据包可分段部分的起始位置的数据的偏移量。

保留 (Reserved)：该域长度为 2 比特，传输时初始值设为 0，接收方忽略此域。

标志位 M (M Flag)：该域长度为 1 比特，M=1 表示还有更多的分段，M=0 表示这是最后一段。

标识 (Identification): 该域长度为 32 比特, 详述如下:

为了从源节点传送一个大于路径 MTU 的数据包到目的节点, 源节点可将该数据分段, 并将每个分段作为一个独立的数据包传送, 由接收者重新进行组装。

源节点为每个要分段的数据包生成一个标识值。该标识值必须不同于从相同源地址到相同目的地址的最近 (注) 发出的任何其他数据包的标识值。如果有路由头存在, 则上述目的地址指的是最终目的地址。

注: “最近”是指在一个包的最大可能生存时间内, 包括从源到目的地的传输时间和等待同一个数据包的其他分段重新组装的时间。然而, 源节点并不需要知道数据包的最大生存时间。可以假定满足这种方法的是把该标识值作为一个 32 位的循环计数器使用, 一旦有数据包被分段, 该计数器就增加 1, 并填入“标识”域。用这种方法来保证“标识”的唯一性。对于每个 IPv6 实现, 它可以自己决定是配置单独的节点计数器, 还是配置多个计数器, 即给每个可能的源地址配置一个计数器, 或是给每一对源地址和目的地址配置一个计数器。

原始数据包是指最初的、没有分段的数据包, 它由下面两部分组成:

原始数据包:

不可分段部分	可分段部分
--------	-------

不可分段部分包括 IPv6 头以及到达目的地路径上由节点处理的所有扩展头, 例如, 所有头包括路由头, 或者逐跳选项头 (如果有的话), 不然就是没有扩展头。

可分段部分包括该数据包余下的部分, 其中包括只能由目的节点处理的扩展头、上层头和数据。

原始数据包的可分段部分被分成段, 除了最后一段外, 每一段的长度都应是 8 字节的整数倍。每一个分段数据包按如下方式传输:

原始数据包

不可分段部分	分段 1	分段 2	分段 n
--------	------	------	-------	------

分段数据包

不可分段部分	分段头	分段 1
--------	-----	------

不可分段部分	分段头	分段 2
--------	-----	------

不可分段部分	分段头	分段 n
--------	-----	------

每个分段数据包的组成如下:

a) 源数据包的不可分段部分, 其中源 IPv6 头内的载荷长度改为该分段数据包的长度 (不包括 IPv6 头本身的长度), 并且将不可分段部分的最后一个头的下一个头域值改变为 44。

b) 分段头部分。包括:

- 1) 下一个头 (Next Header): 标识原始数据包可分段部分的第一个头;
- 2) 段偏移 (Segments Left): 以 8 个字节为单位表示分段相对于原始数据包可分段部分开始位置的偏移量, 第一个分段的段偏移域的值 0;
- 3) 标志位 M (M Flag): M=0 表示该分段是最后一段, M=1 表示该分段不是最后一段;
- 4) 标识 (Identification): 用来标识原始数据包。

c) 分段本身

分段数据包的长度必须不能超过到数据包目的地路径的路径 MTU。

在目的节点要对分段数据包重新进行组装以恢复成原始未分段时的形式。

重组的原始数据包：

不可分段部分	可分段部分
--------	-------

以下是重组时应遵循的原则：

一个原始数据包只能由具有相同源地址、目的地址和分段标识的分段数据包来重组。

重组数据包的不可分段部分包括所有头直到但不包括第一个分段数据包的分段头，它主要做如下两个改动：

- 不可分段部分最后一个头的“下一个头”域的值取自第一个分段的分段头的“下一个头”域；
- 重组数据包的载荷长度从不可分段部分的长度、最后一个分段的长度和偏移量计算得来。计算重组的源数据包长度的公式为：

$$PL_{orig} = PL_{first} - FL_{first} - 8 + (8 * FO_{last}) + FL_{last}$$

其中

PL_{orig} = 重组数据包的载荷长度值；

PL_{first} = 第一个分段数据包的载荷长度值；

FL_{first} = 第一个分段数据包中分段头后面的分段的长度；

FO_{last} = 最后一个分段数据包中分段头的“段偏移”的值；

FL_{last} = 最后一个分段数据包中分段头后面的分段的长度。

重组数据包的可分段部分是由每个分段数据包的分段头后面的分段组成。每个分段的长度等于数据包载荷长度减去 IPv6 头与分段本身之间的头的长度，每个分段在数据包“可分段部分”中的相对位置由“段偏移”的值计算而来。

分段头不出现在最终的重组数据包中。

下面的错误情况可能会在重组分段数据包时发生：

如果在接收到第一个到达的分段之后的 60s 内一个数据包所有要重组的分段没有全部到达，需放弃重组该数据包，并且所有已接受的分段都要丢弃。在这种情况下，如果第一个分段数据包已接收到，则要向那个分段数据包的源地址发送一个 ICMP “超时-段重组超时” 消息。

如果从分段数据包“载荷长度”域中得到的分段长度不是 8 个字节的整数倍，并且这个段的 M 标志位是 1，则这个段就必须丢弃，并且要向段的源地址发送一个 ICMP “参数错误” 消息（代码为 0），ICMP 指针指向分段数据包的“载荷长度”域。

如果一个分段的长度和偏移导致出现这种情况，即由这个分段重组的数据包“载荷长度”超过 65535 个字节，则必须丢弃这个分段，并且向分段的源地址发送一个 ICMP “参数错误” 消息（代码为 0），ICMP 指针指向分段数据包的“段偏移”域。

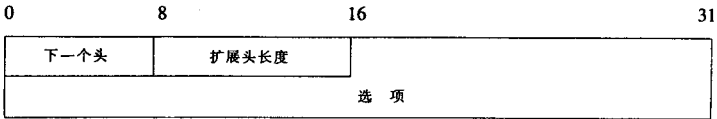
下面的几种情况，虽然不希望在重组时出现，但在发生时不认为是错误：

同一个数据包的不同分段的分段头，在其前面的头的个数和内容可以不同。无论什么头，只要出现在每个分段数据包的分段头前，则当数据包到达时，都要在重组排队之前被处理。只有那些偏移量为 0 的分段数据包的头才保留在重组数据包中。

同一个原始数据包的不同分段的分段头的“下一个头”的值可以不同。因为只有偏移量为 0 的分段数据包的相应值才在重组时有用。

6.6 目的地选项头

目的地选项头用来携带只需由目的节点处理的选项信息。当某一个头的“下一个头”域的值为 60 时，说明紧接着这个头后面的头是目的地选项头。它的格式如下所示：



下一个头 (Next Header): 该域长度为 8 比特, 定义紧接在目的地选项头之后的头的类型。这个域取不同的值, 对应的扩展头类型不同, 值与扩展头类型之间的对应关系见 “IANA 协议值与指定服务网页”。

扩展头长度 (Hdr Ext Len): 该域是一个 8 比特无符号整数, 以 8 个字节为单位表示目的地选项头的长度, 它不包括起始的 8 个字节。

选项 (Options): 该域长度可变, 其长度应保证整个目的地选项头的长度为 8 个字节的整数倍长, 包含有一个或多个 TLV 编码的选项。具体在 5.2 节规定。

在本标准中只为 “目的地址选项” 头规定了 5.2 节中定义的 Pad1 和 PadN 选项。

注意, 在 IPv6 数据包中可以用两种方式来把可选择的目的地信息进行编码: 作为目的地选项头的一个选项, 或是作为单独的扩展头。分段头和认证头是后一种方式的例子。具体采用哪种方式要根据不能识别可选信息的目的节点希望采取的动作而定。

a) 如果目的地节点希望采取的动作是丢弃数据包, 并且只有当数据包目的地地址不是组播地址时, 向数据包的源地址发送一个 ICMP “不能识别类型” 消息, 然后信息可以作为单独的或是作为目的地选项头的一个选项 (“选项类型” 域的高位两比特的值为 11) 来进行编码。具体选用哪种方式的准则是使用的字节数少, 或是排列容易, 或者语义解析时更有效。

b) 如果希望采取其他动作, 则信息必须作为目的地选项头的一个选项 (“选项类型” 域的高位两比特的值为 00、01 或 10) 来进行编码。

6.7 无下一个头

IPv6 头或任何扩展头中 “下一个头” 域的值 59 表示这个头后面没有任何数据了。如果 IPv6 头的 “载荷长度” 域指出头后还有字节, 而这些字节跟在一个包含 59 值的头后面, 则必须忽略这些字节, 同时如果该数据包是转发的则保持不变继续传送。

7 IPv6 数据包的长度

IPv6 要求互联网中任何一条链路的 MTU 不小于 1280 个字节。在任何一条不支持 1280 字节数据包的路径上, 必须在 IPv6 层之下的最近一层提供与链路相关的分段和重组功能。

可配置 MTU 的链路必须将 MTU 配置为至少 1280 字节, 建议这样的链路将 MTU 配置为 1500 字节或更高, 以便在不进行分段的情况下适应可能出现的数据封装。

在与某一节点直接相连的任一条链路上, 这个节点必须要能接收同这些链路的 MTU 一样大的数据包。

IPv6 强烈建议节点实现 “路径 MTU 发现” 以便发现和利用具有大于 1280 个字节的 MTU 的路径。然而, 一个最简单的 IPv6 实现可以简单地限制自己不发送大于 1280 个字节的数据包从而省去了 “路径 MTU 发现”。

为了能发送大于路径 MTU 的数据包, 节点必须在数据源用 IPv6 分段头给数据包分段, 并且在目的地进行重组。然而, 如果应用能调节它的数据包使其适应路径 MTU, 则不鼓励使用分段 (即最小为 1280 字节)。

一个节点必须能接收重组后大小为 1500 字节的分段数据包。一个节点允许接收重组后大小为 1500 字节以上的分段数据包。一个依赖于 IPv6 分段来发送长度超过路径 MTU 的数据包的上层协议或应用不应该发送长度超过 1500 字节的数据包, 除非它明确知道目的地节点有能力重组这么大的数据包。

当一个 IPv6 数据包被发送到 IPv4 目的地时, 起始的 IPv6 节点可能会收到一个 ICMP “数据包过大” 消息来报告下一跳的 MTU 小于 1280 字节。在这种情况下, 并不要求 IPv6 节点把以后发送的数据包长度

减少到 1280 字节以下，但必须在这些数据包中加入分段头，使承担 IPv6-IPv4 协议翻译的路由器能得到一个合适的标记值来进行 IPv4 的分段。注意，这意味着载荷长度可能会减少到 1232 个字节（1280 减去 IPv6 头的 40 个字节和分段头的 8 个字节），如果有附加的扩展头存在，载荷长度可能还会更少。

8 流标签

IPv6 头内的流标签域占 20 比特，源节点用它来标记那些需要 IPv6 路由器特殊处理的一系列数据包，这些特殊处理包括“非缺省的服务质量”或“实时服务”。有关这方面内容在 IPv6 制定时还处于试验阶段，随着互联网的发展不断明朗，支持流的要求会使这部分内容得到更改。对于那些不支持流标签域功能的主机和路由器来说，当发送一个数据包时，在这个域填入 0 值；当转发数据包时，对这个域不作任何改动；当接收数据包时，忽略这个域。

附录 A 描述了现有语义和流标签域的使用方法。

9 业务等级

IPv6 头中的 8 比特业务等级域被源节点和/或路由器用于确定 IPv6 数据包的等级或优先权。目前在 IPv4 中正在试验使用业务类型域为 IP 数据包提供不同形式的区分服务，IPv6 头中的业务等级域与此具有类似的功能。

希望目前的这些试验能最终确定最适合于 IP 数据包的业务流分类。IPv6 业务等级域的语法和语义的详细定义在另外的标准中提供，不属于本标准讨论范围。

下面列出了一些对业务等级域的通用要求：

- IPv6 服务接口必须要能为上层协议提供一种方法，使之在生成数据包时可以修改业务等级域的值。该域的缺省值为 0。
- 支持业务等级域的特殊应用的节点可以在生成、转发或接收数据包时根据特殊应用的需要改变这一域的值。不具备此能力的节点应忽略此域并且不能对其进行修改。
- 一个上层协议接收到的数据包中的业务等级域的值与源节点发送的数据包中该域的值可能不同。

10 上层协议

10.1 上层校验和

任何传输协议和其他上层协议，若在校验和计算中包含了从 IP 头得到的地址，则必须用 IPv6 的 128 比特地址替代 IPv4 的 32 比特地址。基于 IPv6 的 TCP 和 UDP 的“伪头”，如下所示。

源地址	
目的地址	
上层数据包长度	
0	下一个头

- IPv6 数据包含有路由头，则伪头中的目的地址就是最终目的地址。在源节点，这个地址是路由头的最后一个路由选项，在最终接收方，这个地址是接收数据包中 IPv6 头的目的地址域。
- 在伪头中，“下一个头”的值表示上层协议，例如，TCP 为 6，UDP 为 17。如果在 IPv6 头与上层头间存在扩展头，则伪头中的“下一个头”的值不同于 IPv6 头中的“下一个头”的值。
- 伪头上层数据包长度是上层头和数据的长度。某些上层协议带有自己的长度信息，对于这样

的上层协议，伪头中的长度就是该长度信息。对于没有长度信息的上层协议，伪头中的长度是 IPv6 头中的载荷长度值减去 IPv6 头与上层头之间所有扩展头的长度之和。

- 与 IPv4 不同之处是，当 UDP 数据包是由 IPv6 节点发出时，则 UDP 校验不作为选项。即无论谁发出 UDP 数据包，IPv6 节点都必须计算数据包和伪头上的 UDP 校验和。如果计算结果为 0，则必须改为十六进制 0xFFFF 放在 UDP 头中，IPv6 接收方必须丢弃包含 0 校验和的 UDP 数据包，并记录下错误。

- IPv6 的 ICMP 在它的校验和计算中包括了伪头。这与 IPv4 版的 ICMP 不同。IPv4 版的 ICMP 在它的校验和中不包括伪头。这种变化的原因是为了防止 ICMP 的误投送，或是保护 IPv6 头中的相关域不受损害。这一点与 IPv4 不同，在 IPv4 中是由一个互联网层的校验和来完成这个工作的。ICMP 的伪头的“下一个头”域的值 58 时，标识这是一个 IPv6 的 ICMP。

10.2 数据包最大生存时间

与 IPv4 不同，IPv6 节点并不要求必须实现“数据包最大生存时间”。这就是为什么 IPv4 的“生存时间”域在 IPv6 中被改成了“跳数限制”域的原因。在实际应用中，IPv4 设备很少遵守限制数据包最大生存时间的要求，从这个意义上讲实际上没有多少改变。任何一个上层协议，如果它依赖于互联网层来限制数据包生存时间，那么它就应该进行升级以提供检查并丢弃过时数据包的机制。

10.3 最大上层载荷长度

在计算适合于上层数据的最大载荷长度时，上层协议必须要考虑到 IPv6 头和扩展头的总和比 IPv4 头大。例如，在 IPv4 中，TCP 的 MSS 选项是由最大数据包长度减去 40 个字节得到的。其中，最大数据包长度是默认值或是从“路径 MTU 发现”得到的。40 个字节中的 20 个字节是 IPv4 头的最小长度，另外 20 个字节是 TCP 头的最小长度。当在 IPv6 上使用 TCP 时，MSS 必须是最大数据包长度减去 60 个字节，这是因为 IPv6 头的长度比 IPv4 头的最小长度多 20 个字节。

10.4 响应承载路由头的数据包

当一个上层协议发送一个或多个数据包以响应接收到的包含路由头的数据包时，响应数据包中不能包含直接将接收到路由头进行“反转”而得到的“路由头”，除非接收到的源地址和路由的完整性和真实性得到验证（例如，通过使用接收到的数据包中认证头实现）。在响应一个接收到的包含有路由头的数据包时，响应数据包只能是以下的几种情况：

- 响应数据包中不包含路由头；
- 响应数据包中包含的路由头不是通过“反转”接收到的数据包中的路由头而得到的；
- 响应数据包中包含的路由头通过“反转”接收到的数据包中的路由头而得到，当且仅当接收到的数据包中的源地址和路由头的完整性和真实性已经被响应方所验证。

附录 A

(规范性附录)

流标签的语义及使用

一个流是指从一个特定的源地址到特定的目的地址（单播或组播地址）发送的一组数据包，并且源节点希望中间的路由器对流进行特殊的处理。这种特殊处理的属性可以通过控制协议传到路由器，例如，资源预留协议，或者通过流数据包本身所携带的信息传到路由器，如逐跳选项。这样的控制协议或选项的详细内容不在本标准讨论范围。

一对源和目的之间有可能有多个激活的流，也可能有不属于任何一个流的流量。一个流由源地址和非 0 流标签的组合惟一确定。不属于任何一个流的数据包的流标签为 0。

一个流的流标签由流的源节点指定。新的流标签必须从 1 到 0xFFFFF（十六进制）范围内（伪）随机并且惟一地选择。随机分配的目的是使所产生的流标签的任何一组比特都能作为路由器中哈希表的键值，这个键值用于查找流对应的状态。

所有属于同一个流的数据包发送时必须具有相同的源地址、目的地址和流标签。如果其中任何一个数据包包含逐跳选项头，那么流的每一个包都必须包含相同的逐跳选项头（逐跳选项头中的下一个头域除外）。如果其中任何一个数据包包含路由头，那么流的每一个包的包头中的路由头之前的扩展头内容必需相同（路由头中的下一个头域除外）。允许但并不要求路由器或目的节点验证这些条件是否满足。如果检测到条件不满足的数据包，应当向源节点发送 ICMP 参数错误消息，消息代码为 0，消息指针指向流标签的高位字节（即 IPv6 包的第二个字节）。

沿着流路径建立的流处理状态的最大生存周期必须在状态建立机制中说明，例如，资源预留协议或建立流的逐跳选项。源节点不允许在任何流处理状态的最大生存周期内把该流标签分配给新的流，因为在使用流标签前，可能状态已经建立起来了。

当一个节点重启时（例如，死机后的恢复运行），必须小心使用流标签，因为该流标签有可能在前面的仍处于最大生存周期内的流中使用。这可以通过在静态存储上记录流标签的使用情况来实现，从而在死机恢复后仍然保存该信息，或者避免在任何先前可能建立的流的最大生存周期过期之前使用任何流标签。如果节点的最小重启时间已知，实际重启时间可以从等待分配流标签所需的时间中推算得到。

不要求所有或至少大多数数据包属于某一个流，即都带有非 0 的流标签。这条规则提醒协议设计者和实现者不要做相反的假设。例如，只有在大部分数据包都属于流时路由性能良好；或者路由器的包头压缩机制只处理属于流的数据包，这种设计路由器的方法都是不合理的。

附录 B
(资料性附录)
选项格式准则

本附录对设计新的选项中各字段的排列方式作出建议，这些选项用于逐跳选项头和目的地选项头。这些建议基于如下的假定：

- 一个期望的特征是在一个选项的选项数据字段中多字节域应该与自然边界对齐，即对于有 n 个字节长度的域，其起点应该在逐跳选项头或目的地选项头起始位置开始的 n 字节的整数倍处， n 为 1, 2, 4 或 8。
- 另一个期望的特征是逐跳选项头或目的地选项头长度越小越好，同时满足整个扩展头长度为 8 的整数倍的要求。
- 可以假定一个逐跳选项头或目的地选项头带有较少的选项，通常只有一个。

在这些假定下，建议如下方法来排列选项字段：字段按从小到大排序，同时不加内部填充字段，然后再按最大字段对齐要求（最多为 8 字节）来满足整个选项的对齐要求。举例如下。

例 1：

如果选项 X 要求有 2 个数据域，一个长度为 8 字节，另一个长度为 4 字节，其格式可以如下：

0	16	24	31
		选项类型=X	选项数据长度=12
4 字节域			
8 字节域			

其对齐要求为 $8n + 2$ ，以满足 8 字节域开始于头起始位置的 8 的整数倍。包含这样一个选项的逐跳选项头或目的地选项头可以是如下格式：

0	8	16	24	31
下一个头	扩展头长度=1	选项类型=X	选项数据长度=12	
4 字节域				
8 字节域				

例 2：

如果选项 Y 要求有 3 个数据域，一个长度为 4 字节，一个长度为 2 字节，另一个长度为 1 字节，其格式可以如下：

		24	31
选项类型=Y			
选项数据长度=7	1 字节域	2 字节域	
4 字节域			

其对齐要求为 $4n + 3$ ，以满足 4 字节域开始于头起始位置的 4 的整数倍。包含这样一个选项的逐跳选项头或目的地选项头可以是如下格式：

0	8	16	24	31
下一个头	扩展头长度=1	Pad1 选项=0	选项类型=Y	
选项数据长度=7	1 字节域	2 字节域		
4 字节域				
PadN 选项=1	选项数据长度=2	0	0	

例 3:

如果一个逐跳选项头或目的地选项头包含如例 1 和例 2 所示的选项 X 和选项 Y，可以有如下两种格式，具体采用哪一种格式取决于哪一种选项出现在前面。

0	8	16	24	31
下一个头	扩展头长度=3	选项类型=X	选项数据长度=12	
4 字节域				
8 字节域				
PadN 选项=1	选项数据长度=1	0	选项类型=Y	
选项数据长度=7	1 字节域	2 字节域		
4 字节域				
PadN 选项=1	选项数据长度=2	0	0	

0	8	16	24	31
下一个头	扩展头长度=3	Pad1 选项=0	选项类型=Y	
选项数据长度=7	1 字节域	2 字节域		
4 字节域				
PadN 选项=1	选项数据长度=4	0	0	
0	0	选项类型=X	选项数据长度=12	
4 字节域				
8 字节域				