ICS 33 040 40
L 78

YD

# 中华人民共和国通信行业标准

YD/T 1612-2007

# IPv4 网络向 IPv6 网络过渡中的互联互通技术要求

Interconnection and Interoperation Technique Requirements for IPv4-IPv6 Transition



2007-04-16 发布　　2007-10-01 实施

中华人民共和国信息产业部　发布

# 目　次

# 前　言

本标准主要依据国内目前IPv6网络和IPv4网络的现状，并以相关IETF RFC为技术依据编写而成。

本标准是“IPv4－IPv6网络互通”系列标准之一。该系列标准预计的结构及名称如下：

1. IPv4网络向IPv6网络过渡中的互联互通技术要求
2. 基于IPv6网络的IPv4网络互联
3. 面向网络地址翻译（NAT）用户的IPv6隧道技术要求

目前与之相关的“IPv6协议”系列标准的结构及名称如下：

1. IPv6技术要求——IPv6协议
2. IP v6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无状态地址自动配置技术要求
5. 基于IPv6的邻居发现协议
6. IPv6协议一致性测试方法

本标准附录A、附录B、附录C、附录D均为资料性附录。

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：中国电信集团公司
华为技术有限公司
中兴通讯股份有限公司
诺基亚首信通信有限公司

本标准主要起草人：张　荣　张艳芬　卢燕青　李明正　陈　丹　罗汉军　李德丰　赵　锴
王　浩　邓　辉

# IPv4 网络向 IPv6 网络过渡中的互联互通技术要求

## 1 范围

本标准规定了IPv4网络向IPv6网络过渡中采用的多种互通与过渡的技术，包括双栈技术、隧道技术和翻译转换技术等。

IPv4向IPv6过渡涉及到多方面的内容，本标准主要适用于IPv4网络和IPv6网络的互联互通。本标准不涉及目前技术不成熟的过渡技术。

## 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。

| | |
|---|---|
| IETF RFC 2765 | 无状态的 IP/ICMP 翻译算法（SIIT） |
| IETF RFC 2766 | 网络地址翻译——协议翻译（NAT-PT） |
| IETF RFC 2767 | 使用“冲击进入堆栈”（BIS）技术的双堆栈主机 |
| IETF RFC 2893 | IPv6 主机和路由的转换机制 |
| IETF RFC 3053 | IPv6 隧道代理 |
| IETF RFC 3056 | 通过 IPv4 网络连接 IPv6 域 |
| IETF RFC 3142 | IPv6 到 IPv4 传输中继翻译器（TRT） |
| IETF RFC 3338 | 使用 BIA 的双堆栈主机 |
| IETF RFC 4214 | 站内自动隧道寻址协议 |
| draft-ietf-ngtrans-dstm-08.txt | 双栈过渡机制（DSTM） |
| draft-ietf-ngtrans-dstm-overview-00.txt | 双栈过渡机制（DSTM）概述 |
| draft-ooms-v6ops-bgp-tunnel-06.txt | 利用运营商 IPv6 边缘路由器的过渡机制 |



## 3 定义和缩略语

下列定义和缩略语适用于本标准。

### 3.1 定义

**地址（address）**

一个或一组接口的 IP 层标识。

**链路范围内全部节点组播地址（link local scope all-nodes multicast address）**

到达所有节点的本地链路范围地址：FF02::1。

**链路范围内全部路由器组播地址（link local scope all-routers multicast address）**

到达所有路由器的本地链路范围地址：FF02::1。

**泛播地址（Anycast address）**

分配给一组接口的地址，该组接口可以属于不同的节点，目标地址是泛播地址的数据包将被发送到

所有由该地址标识的接口中的一个，该接口是按路由协议中的距离标准确定的路由距离最短的一个。

**自治域（Autonomous System）**

具有独立的、选路策略的和惟一的内部网关路由协议的管理区。

**BGP 设备（BGP speaker）**

保存 BGP 路由信息，运行 BGP 路由选择程序的路由器等设备。

**BGP 对等体（BGP peer）**

两个 BGP 发言者之间相互连接，完成路由信息的交互，这两个路由器就称为 BGP 对等体。

**通信（communication）**

节点之间进行的任何包交换，要求在这种包交换过程中，每个节点进行交换时采用的地址保持不变。例如，TCP 连接或 UDP 请求—响应。

**解封装（decapsulation）**

把封装过的数据包内的内容数据恢复成原来的协议数据包的过程称为解封装。

**封装（encapsulation）**

把一种协议的数据包作为内容数据，完整地放到另一协议数据包内的过程称为封装。

**主机（host）**

除路由器外的任何节点。

**接口（interface）**

节点到链路的连接点。

**接口标识符（interface identifier）**

应用于一个接口，（至少）在每条链路上惟一的基于链路的标识符。无状态地址自动配置用一个接口标识符和一个前缀合成一个地址。从地址自动配置的观点来看，接口标识符是一个已知长度的比特串。接口标识符的确切长度以及产生的方式在单独的链路类型特定文件中定义，文件中会涉及在某种链路类型（如以太网）上传送 IP 的有关问题。在很多情况下，标识符与接口的链路层地址相同。

**无效地址（invalid address）**

未分配给任何接口的地址。有效地址超过有效生存期之后成为无效地址。无效地址不应作为包的目的或源地址出现。在前一种情况下，Internet 路由系统将不能传送该包，在后一种情况下，包的接收者将不能发送响应。

**IPv4 兼容地址（IPv4-compatible IPv6 address）**

拥有前缀高 96 位为 0:0:0:0:0:0，低 32 位为 IPv4 地址的 IPv6 地址。IPv4 兼容地址用于 IPv4/IPv6 节点启动自动隧道技术时。

**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）。

**链路层地址（link-layer address）**

为一个接口分配的链路层标识符。如以太网链路的 IEEE 802 地址和 ISDN 链路的 E.164 地址。

**链路最大传输单元（link MTU）**

通过链路完整传输的数据包最大长度，以字节为单位。

**最长前缀匹配（longest prefix match）**

确定前缀集合中哪个前缀覆盖目标地址的过程。如果前缀的比特位和目标地址的比特位从左最大匹配，前缀就覆盖这个目标地址，当多个前缀覆盖一个地址时，具有最长前缀的地址将得到匹配。

**组播地址（multicast address）**

分配给一组接口的地址，该组接口可以属于不同的节点，目标地址是组播地址的包将被发送到所有由该地址标识的接口。IPv6 中没有广播地址，而是由组播地址代替。

**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。

**节点（node）**

实现 IPv6 的设备。

**数据包（packet）**

IPv6 头和有效载荷构成的数据块。

**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。

**推荐地址（preferred address）**

分配给接口的一个地址，高层协议对它的应用不受限制。推荐地址可以被用做出（或入）接口的源（或目的）地址。

**推荐生存期（preferred lifetime）**

有效地址作为推荐地址的时长（即直到成为不推荐地址之前的时间）。超过推荐生存期之后，地址变成不推荐地址。

**前缀（prefix）**

由地址的初始比特位构成的比特串。

**返回路径可达过程（return routability procedure）**

通过采用加密 Cookie 的交换，返回路径可达过程为绑定过程进行授权。

**路由器（router）**

负责转发目标地址不是它本身的 IPv6 数据包的节点。

**站点本地地址（site-local address）**

局限于本地站点范围内的地址。

**目标地址（target address）**

地址解析所解析出的地址或重定向时新的第一跳地址。

**临时地址（tentative address）**

将一个地址分配到一个接口之前要确认其在链路上的惟一性，正在进行确认的这种地址称为临时地址。临时地址不是通常意义上认为的分配到一个接口上的地址。对临时地址要进行重复地址发现操作，接口会接收与此有关的邻居发现包，但对收到的其他寻址到临时地址的包予以丢弃。

**单播地址（Unicast address）**

分配给单个接口的标示符，目标地址是一个单播地址的数据包将被发送到该地址所标识的接口。

**有效地址（valid address）**

推荐或不推荐地址。有效地址可以作为一个包的源地址或目的地址出现，Internet 路由系统会将发往有效地址的包传传送到它们希望送达的接收者。

**有效生存期（valid lifetime）**

地址保持有效状态的时长（即直到无效之前的时间）。有效生存期必须大于等于推荐生存期。超过有效生存期之后，地址变为无效。

**网络地址转换（NAT）**

本标准引用的术语网络地址转换（NAT）与 RFC 2663 中的 IPv4 网络地址转换（NAT）意义相近，但两者并不完全相同。IPv4 NAT 指的是将一个 IPv4 地址转换成另外一个 IPv4 地址，而本标准中的 NAT 指的是从 IPv4 地址到 IPv6 地址的转换，或者从 IPv6 地址到 IPv4 地址的转换。

另外，IPv4 中的 NAT 提供的是从私有 IPv4 地址域到外部 IPv4 地址域的路由，而本标准中的 NAT 提供的是 IPv6 地址域到外部 IPv4 地址域的路由。

**传统网络地址转换 – 协议转换（NAT-PT）**

传统网络地址转换 – 协议转换允许 IPv6 网络中的主机访问 IPv4 网络中的主机。在传统网络地址转换—协议转换中，只能单向的从 IPv6 网络向外建立会话，这与双向网络地址转换 – 协议转换相对，双向网络地址转换 – 协议转换允许双向建立输入和输出的会话。

**双向网络地址转换 – 协议转换**

对于双向网络地址转换 – 协议转换来说，会话既可以由 IPv4 网络中的主机发起，也可以是由 IPv6 网络中的主机发起。当任何一个方向的连接建立起来以后，IPv6 地址与 IPv4 地址被静态或动态绑定在一起。IPv4 和 IPv6 域中的主机名字空间被认为是惟一的，IPv4 域中的主机访问 IPv6 域中的主机时，通过域名服务器（DNS）进行地址解析。在应用双向网络地址转换 – 协议转换时，必须结合使用域名服务器 – 应用层网关（DNS-ALG）以方便名字到地址的映射。特别地，该域名服务器 – 应用层网关必须能够在 IPv4 和 IPv6 域之间进行传输，并且对 DNS 查询中的 IPv6/IPv4 地址进行转换，映射到其对应的 IPv4/IPv6 绑定地址，并对该查询进行响应。

**协议转换（PT）**

本标准中的协议转换指的是将 IPv4 数据包转换成语义上完全相同的 IPv6 数据包，反之亦然。

**应用层网关**

应用层网关是一个特定的应用层代理，它可以配合完成 IPv6 节点与 IPv4 节点的双向通信。有些应用会在其载荷中携带网络层地址，而网络地址转换 – 协议转换无法对应用层载荷进行处理，所以此时就需要将应用层网关与网络地址转换 – 协议转换结合起来，以便支持这些特殊的应用。

**隧道代理**

隧道代理是一种专用服务器，用来自动管理来自用户的隧道请求。隧道代理用来管理隧道的创建、修改和删除。基于可扩展性原因，隧道代理可以在几个隧道服务器之间共享网络侧隧道端点上的负载。当隧道建立、修改或删除时，隧道代理将发送配置指令到相应的隧道服务器。隧道代理还可以完成注册用户 IPv6 地址和名字到 DNS 服务器。

**隧道服务器**

隧道服务器是连接到因特网上的一个 IPv4/IPv6 双栈路由器。当它接收到隧道代理发送过来的一个配

置指令以后，它将会创建、修改或删除每一个隧道的服务器端，同时，隧道服务器也可能会对每一个活动隧道维护一些应用统计信息。

**隧道代理客户端**

隧道代理服务的客户端是连接到 IPv4 网络中的一个双栈 IPv6 节点（可以是主机，也可以是路由器）。

**6to4 伪接口**

6to4 的隧道封装是在一个逻辑上和 IPv6 接口等同的点上完成的，它的链路层地址就是 IPv4 单播地址。这个点被称为伪接口。有些实现者把该接口完全作为普通的接口来对待，而其他实现者视其为一个隧道端点。

**6to4 前缀**

根据特定规则生成的 IPv6 前缀。

**6to4 地址**

采用 6to4 前缀生成的 IPv6 地址。

**原始 IPv6 地址**

使用非 6to4 的其他前缀生成的 IPv6 地址。

**6to4 路由器（或 6to4 边界路由器）**

支持 6to4 伪接口的 IPv6 路由器。通常是一个介于 IPv6 站点和广域 IPv4 网络之间的边界路由器。

**6to4 主机**

一个拥有至少一个 6to4 地址的 IPv6 主机。在其他方面看来，则是一个标准的 IPv6 主机。

**6to4 区域**

内部采用 6to4 地址运行的 IPv6 网络区域，因此至少应该包含一个 6to4 主机和一个 6to4 路由器。

**中继路由器**

被配置成支持 6to4 地址和原始 IPv6 地址之间跨越路由的 6to4 路由器。

**6to4 外部路由域**

互联一系列的 6to4 路由器和中继路由器的路由域。它不同于 IPv6 站点内部路由域，也不同于所有的原始 IPv6 外部路由域。

**6over4**

6over4 过渡机制是利用 IPv4 的组播实现的一种自动隧道方式，它把 IPv4 作为 IPv6 的虚拟链路层处理。这种机制适用于分布在 IPv4 网络中分散的双栈节点实现互联。6over4 需要 IPv4 组播支持。

**IPv4 资质节点（IPv4 capable node）**

有 IPv4 协议栈的节点。在该协议栈可以使用之前，必须为该节点分配一个或者多个 IPv4 地址。

**IPv4 激活节点（IPv4 enabled node）**

具有 IPv4 协议栈并且已经分配了一个或者多个 IPv4 地址的节点，纯 IPv4（IPv4-only）节点和 IPv6/IPv4 的节点都是 IPv4 激活的节点。

**IPv6 资质节点（IPv6 capable node）**

有 IPv6 协议栈的节点。在该协议栈可以使用之前，必须为该节点分配一个或者多个 IPv6 地址。

**IPv6 激活节点（IPv6 enabled node）**

具有 IPv6 协议栈并且已经分配了一个或者多个 IPv6 地址的节点，纯 IPv6（IPv6-only）节点和 IPv6/IPv4 的节点都是 IPv6 激活的节点。

**IPv4 映射地址（IPv4-mapped Address）**

具有 0::ffff:a.b.c.d 形式的地址，用来标识一个非 IPv6 资质节点。

**IPv4 兼容地址（IPv4-compatible Address）**

具有形式 0::0:a.b.c.d 的地址，用来标识支持自动隧道技术的 IPv6/IPv4 双栈节点。

**IPv4 翻译地址（IPv4-translated Address）**

具有形式 0::ffff:0:a.b.c.d 的地址，用来标识一个 IPv6 激活节点。因为前缀 0::ffff:0:0:0/96 是用来校验和为 0，从而避免给传输协议的伪包头校验和带来任何变化。

## 3.2 缩略语

| | | |
|---|---|---|
| AFI | Address Family Identifier | 地址族标识符 |
| AH | Authentication Header | 认证头 |
| ARP | Address Resolution Protocol | 地址解析协议 |
| AS | Autonomous System | 自治域 |
| BGP | Border Gateway Protocol | 边界网关协议 |
| EGP | External Gateway Protocol | 外部网关协议 |
| ESP | Encapsulating Security Payload | 封装安全载荷 |
| FIFO | First In First Out | 先进先出 |
| ICMP | Internet Control Message Protocol | 互联网控制消息协议 |
| IETF | Internet Engineering Task Force | 互联网工程任务组 |
| IGP | Interior Gateway Protocol | 内部网关协议 |
| IANA | Internet Assigned Numbers Authority | 互联网编号分配机构 |
| IP | Internet Protocol | 互联网协议 |
| IPv4 | Internet Protocol Version 4 | 互联网协议版本 4 |
| 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 |
| NBMA | Non Broadcast Multi-Access | 非广播多连接 |
| ND | Neighbor Discovery | 邻居发现 |
| NLR | Network Layer Routing Information | 网络层可到达信息 |
| SAFI | Subsequent Address Family Identifiers | 子序列地址族标识符 |
| SNPA | Subnet Work Points of Attachment | 子网连接点 |
| TLV | Type-Length-Value | 类型-长度-值 |

# 4 IPv4 向 IPv6 过渡的基本技术

目前，从IPv4向IPv6过渡的基本技术主要有三类：双栈技术、隧道技术以及翻译转换技术。本标准中所讨论的过渡机制不适用于NAT穿越环境，该问题仍有待研究。

## 4.1 双栈技术

### 4.1.1 双栈

实现 IPv6 节点与 IPv4 节点互通的最直接方式是在 IPv6 节点中加入 IPv4 协议栈。具有双协议栈的节

点称作“IPv6/ IPv4 节点”，这些节点既可以收发 IPv4 数据包，也可以收发 IPv6 数据包。它们可以采用 IPv4 协议与 IPv4 节点互通，也可以直接采用 IPv6 协议与 IPv6 节点互通。

双栈节点可处于以下三种工作模式：

（1）激活 IPv4，保持 IPv6 为非工作状态；

（2）激活 IPv6，保持 IPv4 为非工作状态；

（3）同时激活 IPv4、IPv6 协议栈。

双栈节点的地址配置：

对于双栈节点，一般 IPv4 的地址配置采用原有 IPv4 的地址配置方法，IPv6 的地址采用 IPv6 的地址分配机制来分配纯 IPv6 地址；但是在结合各种隧道机制工作的时候，需要通过 IPv4 地址来生成对应的 IPv6 地址，具体细节在相关的隧道技术中进行介绍。

DNS 相关问题：

（1）域名解析结果的返回与响应

由于双栈节点既要求可以和 IPv4 节点通信，也要求可以和 IPv6 节点通信，那么 IPv4/IPv6 双栈节点就必须具有解析 IPv4 与 IPv6 地址的能力，即双栈节点上的域名解析器需要能够处理 AAAA 和 A 两种类型的记录。当域名解析器仅得到含有 IPv6 地址的 AAAA 记录或者含有 IPv4 地址的 A 记录请求答复时，只能将相应的域名解析对应地址送交应用，不存在选择问题；当域名解析器既得到含有 IPv6 地址的 AAAA 记录同时也得到含有 IPv4 地址的 A 记录请求答复时，就会存在对两个域名解析结果的过滤和排序的问题，因为这直接影响到启用 IPv4 和 IPv6 哪一个协议栈进行通信。通常域名解析器有三种选择：

a）仅向应用返回 IPv6 的地址；

b）仅向应用返回 IPv4 的地址；

c）将 IPv6 的地址与 IPv4 的地址都返回给应用，这时会涉及到排序问题。

应用层对上述结果的响应：

a）仅得到 IPv6 的地址时，应用 IPv6 协议进行通信；

b）仅得到 IPv4 的地址时，应用 IPv4 协议进行通信；

c）同时得到 IPv4 与 IPv6 的地址时，应用层决定应用哪一个地址，也就是应用层选择使用哪一个协议栈进行通信。

域名解析器在同时送交两种域名解析结果给上层应用时，可以通过选举进行 IPv6 地址与 IPv4 地址的排序，从而影响上层应用的选择；因为通常应用都是选择排在前边的结果。

域名解析器的具体过滤和排序操作与实现相关。双栈节点可以对此提供策略配置，也可以完全交给上层由应用来处理。

（2）由 DNS 通告 IPv6 地址的建议

建议仅在以下三个条件都得到满足时，可考虑把 AAAA 记录加入 DNS：

a）该地址已分配给该节点的相应接口；

b）该地址已经配置在该接口上；

c）该接口所处链路与 IPv6 的网络相连接。

当条件 c）不满足的时候，即该节点是一个孤立的 IPv6 节点，在 DNS 中就不应该有该节点的 AAAA 记录。当其他双栈节点主动与孤立的双栈节点进行通信的时候，由于 DNS 中没有 AAAA 记录，通信双方直接采用 IPv4 进行通信，而不会尝试用 IPv6 进行通信（假设通信双方在 DNS 中都存在 A 记录）。

当孤立的双栈节点主动与其他双栈节点建立通信的时候，它可能得到对端的 AAAA 记录。如果该节点本身至少有一个接口配置了 IPv6 地址，那么它会选择用 IPv6 进行通信。由于没有 IPv6 的路由（不满足条件 c），那么采用 IPv6 的通信将会失败，也就说由于 TCP 超时引起几分钟的通信时延。一旦发生 TCP 超时，希望应用会尝试 IPv4 进行通信，但是这个过程会比较缓慢；如果应用不作尝试，那么通信将彻底失败。

上述限制条件可避免一些不必要的通信延时和失败。当其他双栈节点主动与孤立的双栈节点进行通信的时候，如果DNS中有AAAA记录，通信发起端将会采用IPv6进行通信，而只能得到延时或者不可达的信息。

### 4.1.2 DSTM

#### 4.1.2.1 DSTM 概述

DSTM（Dual Stack Transition Mechanism，双栈迁移机制），需要结合隧道技术进行应用。采用 DSTM 机制的节点必须是双栈节点，因此我们称这种过渡技术为双栈技术。

DSTM 的出发点是提供 IPv6 节点一个获得 IPv4 地址的方式，从而使之能够与纯 IPv4 节点或者 IPv4 应用程序通信。基于 IPv4-over-IPv6 隧道，通过 DSTM 实现了纯 IPv6 网上传输 IPv4 流量；同时 DSTM 提供一个分配临时 IPv4 地址给 IPv6/IPv4 双栈节点的方法。DSTM 机制无需采用 NAT 技术即可实现 IPv6 节点与 IPv4 节点的互通。

DSTM 的体系结构中包括：一个 DSTM 地址服务器、一个网关或者说是隧道端点和若干 DSTM 节点。地址服务器负责为客户端节点分配 IPv4 地址。DSTM 服务器只需要保证在一定时间内 IPv4 地址的惟一性；网关或隧道终点可以被看作纯 IPv6 域和外部的 IPv4 Internet 或 Intranet 的边界路由器；节点执行封装/解封装数据包，完成收发过程。最后为了保证 IPv4 的连通性，在纯 IPv6 域中的节点必须能动态配置它们的 IPv4 栈（通过向服务器请求临时地址）而且必须能建立 4over6 隧道到隧道端点（TEP）。



DSTM 核心设想是对应用层透明，应用可以继续采用 IPv4 地址，而且对于承载网络也是透明的 IPv6 的包。能保证一些在净核中包含 IPv4 地址的应用包能够继续正确地被传送（IPSec 和 H.323 等）。

DSTM 模型假定如下：

- DSTM 域在 Intranet 而不是 Internet 中。
- IPv6 节点平时不维护 IPv4 地址（除非是与纯 IPv4 节点通信或采用 IPv4 应用而临时分配地址）。
- 临时 IPv4 地址由 DSTM 地址服务器分配，分配的协议有多种选择（如 DHCPv6）。由于服务器和客户端间的通信是纯 IPv6 方式，故地址分配协议要考虑这个限制。
- 为了减少对 IPv4 地址的需要，作为扩展 DSTM 服务器还可以提供一个端口范围给客户端使用。这样将允许不同节点同时采用相同的 IPv4 地址，从而降低了对 IPv4 地址池大小的要求。
- DSTM 域内，IPv4 路由表保留到最小，因此减少了过渡时期所需要的网管工作。
- 一旦 IPv6 节点获得 IPv4 地址，动态隧道将被用来把封装了 IPv4 数据的 IPv6 数据包转发到另一端的 TEP（隧道端点）。此 TEP 将其再解封装成 IPv4 数据包并采用 IPv4 转发。在分配 IPv4 地址的同时，DSTM 地址服务器应同时提供 TEP 的 IPv6 地址。
- 现存的 IPv4 应用于 DSTM 节点中不需要做任何修改。
- 只要 TEP 能够到达 IPv4 目的地址，DSTM 节点就可以与该 IPv4 节点通信。

DSTM 的过渡模型如图 1 所示。

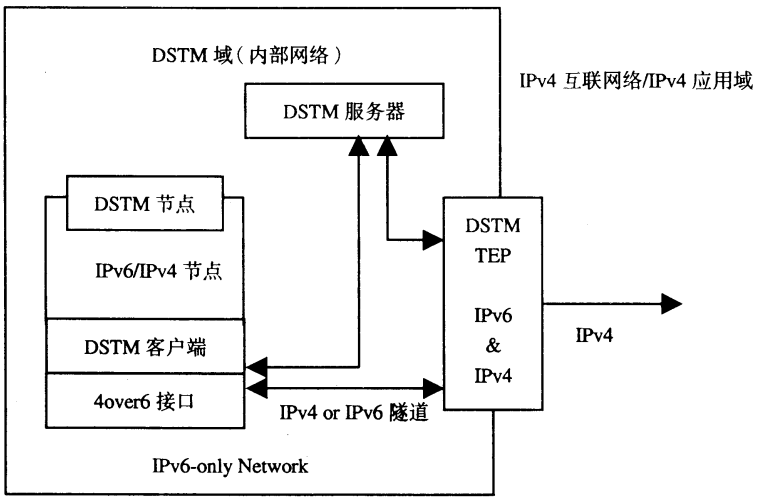



**图 1　DSTM 过渡模型**

对于参与 DSTM 的 IPv6 节点必须是双栈节点，纯 IPv6 的节点不能采用 DSTM。

#### 4.1.2.2　DSTM 节点要求

（1）IPv4 协议栈的配置

在 DSTM 节点中，本域的 IPv6 通信不需要采用 IPv4 地址。当出现以下情况时，主机需要 IPv4 地址来完成通信：

a）DNS 解析的结果表明目的地是一个 IPv4 地址的节点；

b）当应用打开一个 IPv4 Socket；

c）生成一个 IPv4 数据包，却没有接口可以转发此包。

当第一个 IPv4 包需要被转发时，DSTM 客户端必须联系 DSTM 服务器。紧跟着这个消息的交换，客户端从服务器拿到 TEP 的 IPv6 地址同时也拿到临时的 IPv4 地址。如果允许，服务器还可以提供可以被应用的端口范围。这些信息是用来配置 4over6 接口的。只有 4over6 的隧道接口配置后，DSTM 节点的 IPv4 协议栈才可以说是完整配置的。

（2）IPv4 包转发

由于缺乏 IPv4 路由结构，DSTM 节点不能直接在网络上发送 IPv4 包。节点必须将 IPv4 包封装进 IPv6 包中并且发给隧道终端 TEP，而此 TEP 解开封装并转发到 IPv4 网络上。

在 DSTM 节点上，由 4over6 接口完成封装工作。所有的 IPv4 流量可以通过一个 IPv4 路由表项指向这个接口。具体的 4over6 接口和有关路由表项，与应用相关的。

（3）DSTM 节点中 IPv4 包的处理

当 DSTM 节点需要发送 IPv4 包时，它会交给 4over6 接口。如果 4over6 接口还没有配置（无 IPv4 地址），处理被阻塞，同时节点会向 DSTM 服务器请求一个临时地址。一旦得到 IPv4 地址，所有经过该 4over6 接口转发的包都应用这个临时地址作 IPv4 源地址，其他 IPv4 域正常处理。

（4）IPv6 数据包结构

当 4over6 接口将 IPv4 包封装到 IPv6 包中时，它必须决定 IPv6 包的目的地址。通常这个地址是 TEP 地址。TEP 地址可以是静态配置或者节点获得 IPv4 地址的同时从 DSTM 服务器动态获得。

当 DSTM 节点接受临时 IPv4 地址的时候，TEP 的 IPv6 地址必须由 DSTM 服务器提供。然而 DSTM

节点在早先应用 DSTM 的时候可以手工配置 TEP。对于长期解决方式不推荐使用后者。

封装 IPv4 的包中“下一个头”域填 4。当隧道包到达 IPv6 终点时，IPv6 包头被去掉，由 IPv4 栈来处理解封后的包。此后 TEP 将采用普通 IPv4 的方法来转发剩下的 IPv4 包。TEP 应该缓存这种 IPv4 和 IPv6 源地址的对应关系。

一个封装有 IPv4 的 IPv6 包的源地址，应该是 IPv6 包被发送的物理接口的 IPv6 地址。

#### 4.1.2.3 DSTM 服务器的要求

DSTM 服务器负责为节点分配临时的 IPv4 地址。服务器仅仅需要保证在一段时间内 IPv4 地址的惟一性。为了减少对 IPv4 地址的需要，有些应用还包括了一个端口范围作为分配过程的一部分。这样将允许不同节点同时采用同一 IPv4 地址。

DTSM 服务器应记录节点的 IPv6 地址与临时 IPv4 地址之间的对应关系。临时 IPv4 地址具有一定的生命周期，经过生命周期后，客户端应重新申请租用 IPv4 地址。

在 IPv4 的路由方面，必须保证 DSTM 服务器所管理的 IPv4 地址池能被路由到一个或多个此 DSTM 域中的 TEP。当分配一个地址给 DSTM 节点时，服务器消息应该包括 TEP 的 IPv6 地址，并且 DSTM 服务器可以负责配置 TEP 中的 IPv4 – IPv6 映射表。当 TEP 不能动态建立映射或者因为安全原因取消动态映射，DSTM 的客户端和服务器的通信必须是 IPv6 方式。DSTM 服务器也可以在没有客户端请求的情况下分配临时 IPv4 地址。

DTSM 服务器应该能够对 DTSM 客户端进行认证。

#### 4.1.2.4 DSTM 的适用性

DSTM 适用于 IPv6 域内的节点需要与域外 IPv4 节点通信的情况。如果应用层网关被适当的运用，IPv4 的连通性需求将大大降低。常规的服务，比如 HTTP、SMTP 就可以利用这个特点。DSTM 在没有其他的解决方案（如应用层网关）时就可以布署使用。DSTM 允许双栈节点获得 IPv4 地址并且提供一个缺省路由（通过 4over6 隧道）到 IPv4 网关。如果采用上述机制，并且 DSTM 被配置成能分配共有 IPv4 地址，那么任何纯 IPv4 应用都可以在 IPv6 网络上运行，在域内的主机将能与 Internet 上任何其他主机进行通信。

#### 4.1.2.5 安全性

DSTM机制可以应用所有已经定义的安全规范。对于DNS可以应用DNS安全扩展/更新。而在地址分配方面，当由DSTM节点触发的连接中，对地址池的DOS危险是有限的，因为DSTM是一个Intranet环境。在Intranet DSTM中，如果布署了DHCPv6，则可以应用DHCPv6认证消息，同时TEP位于Intranet中，它们不作为开放的中继。最后对于DSTM节点的IPv4通信，一旦节点具有IPv4地址，IPSec就能被应用，因此DSTM不会破坏在任何节点的端到端通信安全。

### 4.2 隧道技术

#### 4.2.1 隧道概述

隧道机制提供了利用现有 IPv4 网络架构实现 IPv6 通信的方法，也可适用于利用 IPv6 网络架构实现 IPv4 的通信，后者的详细技术方案另题研究。

隧道机制的基本工作方法如下：

（1）隧道入口对 IPv6 数据包先进行 IPv4 封装，然后发送。

（2）隧道出口收到隧道封装的数据包后，先确认是否需要重组，如果数据包经过分段，那么需要重组；否则不必。然后去掉隧道封装（IPv4 头），更新 IPv6 头，对收到的数据包作相应处理。

（3）为了使数据包能够顺利通过隧道，隧道入口可能需要维护隧道的软状态信息，比如记录隧道

MTU 等参数。一个网络节点所采用的隧道可能会很多，相关的软状态可以被缓存等不用的时候就丢弃。

隧道机制有以下几种应用情况：

（1）路由器到路由器（R-R）：通过 IPv4 网络互联的两台 IPv6/IPv4 双栈路由器可以利用隧道方式在这两台路由器之间传递 IPv6 数据包。R-R 隧道通常用于 IPv6 端到端路径的中间段。

（2）主机到路由器（H-R）：通过 IPv4 网络互联的双栈主机与双栈路由器之间可以建立隧道进行 IPv6 通信。H-R 隧道通常用于 IPv6 端到端路径的首段。

（3）主机到主机（H-H）：通过 IPv4 网络互联的两台 IPv6/IPv4 双栈主机之间可以建立隧道进行 IPv6 通信。H-H 隧道连接 IPv6 通信的两端，即覆盖 IPv6 端到端路径全程。

（4）路由器到主机（R-H）：双栈路由器与通过 IPv4 网络与之互联双栈主机之间建立的隧道。H-H 隧道通常用于 IPv6 端到端路径的最后一段，即双栈主机为通信终点。

在 R-R、H-R 的应用中，隧道终点的路由器属于端到端通信的中间节点，即 IPv6 数据包的目的节点并非隧道终点。也就是说隧道终点的地址不能够通过 IPv6 数据包的目的地址得到，必须由隧道起点处的配置信息得到。这种隧道终点必须显式配置的隧道称为配置隧道。

在 H-H、R-H 的应用中，隧道终点与 IPv6 数据包的目的节点相同。这样就可以通过把隧道终点的 IPv4 地址信息放在 IPv6 的目的地址中，从而可以不需要特别配置隧道终点的 IPv4 地址就可以从目的 IPv6 地址中获得隧道终点的 IPv4 地址。这种利用内嵌 IPv4 地址的特殊 IPv6 地址，使隧道起点自动发现隧道终点 IPv4 地址的隧道称为自动隧道。

自动隧道与配置隧道的主要不同在于如何识别隧道终点的地址，其他原理基本相同。

#### 4.2.1.1 隧道的封装

IPv6 in IPv4 的封装如图 2 所示。



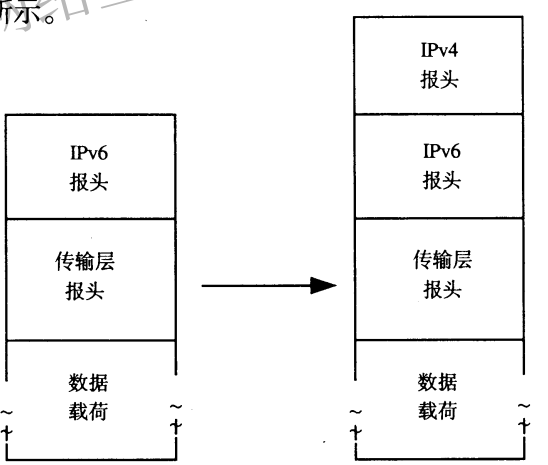



**图 2 IPv6 in IPv4 的隧道封装格式**

除了为 IPv6 的数据包加上 IPv4 的数据包头，封装节点还需要：

- 决定是否需要拆分数据包以及是否需要向源端发送“数据包过长”的 ICMP 错误消息；
- 如何将隧道路径上路由器返回给源端的IPv4错误消息映射成IPv6的ICMP消息。

#### 4.2.1.2 隧道 MTU 与数据包分段

封装节点可以将 IPv6 in IPv4 的封装形式看作是 IPv6 用 IPv4 作为数据链路层。如果仅考虑 IPv4 对数据包的处理，该链路层的 MTU 可为（65535 – 20）字节（20 个字节是 IPv4 头的长度）。封装节点仅当 IPv6 数据包长度超过这个 MTU 时向源端返回“数据包过长”的 ICMP 错误消息。但是这样大的 MTU 会带来一些缺点：

（1）这样会引起更多的数据包分段；由于一旦出现数据包丢失，重传的数据包将多于丢失的数据包，因此会导致性能下降。应尽量避免 IPv4 层过多的数据包分段。

（2）隧道中经过 IPv4 拆分的数据包在隧道终点都需要重组，如果隧道终点是路由器，那么将消耗路由器上更多的内存进行数据包整合，重组成完整的 IPv6 数据包后才能继续转发。

通过封装节点记录隧道路径上的 IPv4 路径 MTU，可以将拆分数据包的次数减为最低。具体方法参考 IPv4 路径 MTU 发现协议。封装节点的 IPv6 层视 IPv4 层为链路层，其 MTU 等于（IPv4 路径 MTU－20）字节。但是当 IPv4 路径 MTU－20 小于 1280 字节时，仍会存在数据包分段，因为 IPv6 协议规范中规定 IPv6 的任何链路层的 MTU 都必须大于等于 1280 字节。

封装节点可采用以下算法决定数据包的拆分：

```
if （IPv4 路径 MTU－20）不大于 1280
        if  IPv6 数据包长度大于 1280 bytes
                发送“数据包过长”ICMP 消息，其中 MTU 参数的值等于 1280
                丢弃数据包。
        else
                封装，同时 IPv4 头中不设置“不可拆分”标志；
                （封装后的数据包可能被封装节点或者隧道路径上的路由器进行拆分）
        endif
    else
        if  IPv6 数据包长度大于（IPv4 路径 MTU－20）
                发送“数据包过长”ICMP 消息，其中 MTU 参数的值等于（IPv4 路径 MTU－20）
                丢弃数据包
        else
                封装，同时 IPv4 头中设置“不可拆分”标志；
        endif
    endif
```

如果封装节点包含多条隧道，那么该节点可能不能够同时保存所有隧道的路径 MTU 值。对于封装节点没有路径 MTU 信息的隧道，不必采用 IPv4 路径 MTU 算法，直接采用 IPv4 下边链路层的 MTU 作为 IPv4 路径 MTU 的值实现上述算法即可，这可能导致 IPv4 层对数据包进行拆分，此时“不可拆分”标志应清零。

#### 4.2.1.3 跳数限制（Hop Limit）

IPv6-over-IPv4 的隧道被视为单跳，即在隧道终点 IPv6 数据包头中的 Hop Limit 值减 1，而隧道起点对 IPv6 的 Hop Limit 域不进行操作。单跳模型使隧道对于网络用户不可见，用 TRACEROUTE 等网络诊断工具也不会发现隧道的存在。

### 4.2.2 手工配置隧道

在配置隧道里，配置隧道终点的IPv4地址是从封装节点的配置信息里获得的。对于每一个隧道，封装节点都必须保存该隧道的终点IPv4地址。当IPv6数据包通过这个隧道传输时，配置的隧道终点IPv4地址将作为封装数据包的目的地址。

数据包是否通过隧道传输，通常由封装节点通过路由信息确定。一般通过路由表直接应用数据包的目标地址经过前缀掩码和匹配技术决定是否采用隧道。

#### 4.2.2.1 缺省配置隧道

IPv6/IPv4节点如果没有数据链路连接到一个IPv6路由器的话，可以通过一个配置隧道到达一个IPv6路由器。隧道允许一台主机跟IPv6的互联网络进行通信。如果知道一个IPv4地址是IPv6骨干网边界IPv6/IPv4路由器的接口地址之一，那么它就能用作隧道的终点地址。隧道就可以作为IPv6的“缺省路由”配置到路由表。也就是说，所有的IPv6目的地址将可以匹配这条路由而穿过这条隧道。因为缺省路由的掩码长度是零，如果没有更长一些的掩码匹配这个目的地址，那么它将惟一的被应用。缺省配置隧道可以与本标准规定的自动隧道一起使用。

#### 4.2.2.2 入口验证

去封装节点必须在转发IPv6数据包前验证隧道的源地址是否可以接受，以防止恶意攻击。但需注意，如果去封装数据包是向传输层协议递送，则不应该检查。对于双向的配置隧道，还要检查源地址是否属于隧道另一端的。如果是单向配置隧道，去封装节点必须维护一个可以接收的源IPv4地址前缀列表，并且这个列表缺省应该是空的，也就是说，节点如果要转发通过单向隧道收到的数据包，必须经过明确的配置。

### 4.2.3 兼容地址自动隧道

自动隧道允许IPv6/IPv4节点通过不用预先配置隧道的IPv4路由网络进行通信。在自动隧道里，隧道的终点地址由通过隧道的IPv6数据包里的IPv4兼容目的地址获得。

#### 4.2.3.1 IPv4 兼容地址格式

具有自动隧道功能的IPv6/IPv4节点应该被分配一个IPv4兼容地址。IPv4兼容地址是由96位的全零前缀和后32位的IPv4地址组成。其结构如图3所示。

| 96-bits | 32-bits |
|---|---|
| 0 : 0 : 0 : 0 : 0 : 0 | IPv4 Address |

**图 3 IPv4 兼容的 IPv6 地址格式**

IPv4兼容地址是专门分配给支持自动隧道节点的。只有在准备接收封装在IPv4数据包里的目的地址内嵌着IPv4地址的IPv6数据包时候，该节点才应该配置IPv4兼容地址，IPv4封装数据包头中的目的地址等于IPv6数据包头中IPv4兼容目的地址的低32位。

如果IPv4地址不是私有地址，那么由这个IPv4地址构成的IPv4兼容地址也应该是全球惟一的。实现时就像把这个IPv4兼容地址分配到自动隧道接口上一样，即使某些自动隧道实现时没有用到接口的概念。因此，IPv4兼容地址不应该被看作分配给类似以太网的接口，也就是说，以太网上的邻居发现机制在这类隧道上并不适用。

#### 4.2.3.2 IPv4 兼容地址配置

一个IPv6/IPv4节点的IPv4兼容地址作为自己的IPv6地址之一来应用，而嵌入在低32位的IPv4地址是自己某个接口上的IPv4地址。

一个节点可以通过IPv4地址配置协议生成IPv4兼容地址。它可以用任何的IPv4配置机制去获得IPv4地址，然后用96位的前缀0:0:0:0:0:0“映射”成IPv4兼容地址。这种配置模式允许IPv6/IPv4节点“利用”已有的基于IPv4的地址配置服务。

具体的获取IPv4兼容地址的算法可以基于以下的IPv4地址配置协议。

（1）IPv6/IPv4 节点用标准的 IPv4 机制或协议去为一个接口获取一个 IPv4 地址，包括：

- 动态主机配置协议（DHCP）；
- 引导程序协议（BOOTP）；
- 逆向地址解析协议（RARP）；
- 手动配置；
- 其他正确的生成节点自己的 IPv4 地址机制。

（2）节点将获得的地址分配到接口上。

（3）然后将 96 位前缀 0:0:0:0:0:0 与第 1 步获取的 IPv4 组合起来，形成了低 32 位是 IPv4 地址的 IPv4 兼容地址。节点就像采用其他自己的 IPv6 地址一样应用这个 IPv4 兼容地址。

#### 4.2.3.3 自动隧道的操作

自动隧道终点地址是根据经过隧道的数据包确定的。如果这个目的IPv6地址是IPv4兼容地址，这个数据包就能够通过自动隧道；如果目的IPv6地址是普通IPv6地址，就不能够通过自动隧道发送。

路由表项能够指导数据包的自动隧道。一种实现是前缀0:0:0:0:0:0/96有一个专门的静态路由表项，就是说，这个路由将全零前缀来用作96位掩码。匹配这个前缀的数据包将被发送到能完成数据包自动隧道传递的伪接口。因为所有的IPv4兼容地址都匹配这个前缀，因此所有的数据包都会通过自动隧道发送到目的地。

一旦数据包通过伪接口被传送到自动隧道模块，这个数据包将用IPv4数据包头进行封装，封装数据包的源地址和目的地址将按下面描述的进行分配：

目的IPv4地址：

IPv6目的地址的低32位；

起源IPv4地址：

发送该数据包的接口IPv4地址。

自动隧道模块将永远发送被封装的数据包，不管目的地址在不在数据链路上。

自动隧道模块必须禁止发送广播或多播数据包。如果IPv6数据包的目的IPv4兼容地址里的IPv4地址是广播地址、组播地址、未定义地址和环回地址，这个数据包必须被丢弃。

#### 4.2.3.4 利用缺省配置隧道

自动隧道经常和缺省配置隧道联合应用。“孤立”IPv6/IPv4主机配置了自动隧道能力和IPv4兼容地址，并且至少配置了一条连接到IPv6路由器的配置隧道，这个IPv6路由器也配置了自动隧道能力。这些IPv6/IPv4主机发送的数据包如果目的地址是IPv4兼容地址，就采用自动隧道，如果目的地址是普通IPv6地址，就采用配置隧道。在路由查找中，目的地址为IPv4兼容地址的数据包将匹配96位全零前缀，目的地址为普通IPv6地址的数据包将匹配与配置隧道关联的缺省路由。这些数据包的响应数据包在IPv6网络中先被路由到IPv6/IPv4路由器，路由器再通过自动隧道返回给源主机。

#### 4.2.3.5 源地址的选择

当一个IPv6/IPv4节点构造发送一个IPv6数据包时，它必须选择一个源IPv6地址。配置了自动隧道的IPv6/IPv4节点可能配置一个普通的全局IPv6地址，也可能配置IPv4兼容地址。源地址的选择将决定对端返回数据包时通过哪种隧道。如果选择IPv4兼容地址作为源地址，那么返回数据包将通过自动隧道；如果选用普通IPv6地址，返回数据包将通过配置隧道。通常最好是两端对称，以下是推荐的选择方法：

目的地址是IPv4兼容地址：

用分配给IPv4出口的IPv4兼容地址作为源地址；

目的地址是纯IPv6地址：

用出接口的普通IPv6地址。

如果一个IPv6/IPv4节点没有普通的全局IPv6地址，但要发送一个数据包到具有普通IPv6地址的节点，这时可以采用IPv4兼容地址作为源地址。

#### 4.2.3.6 入口过滤

为了避免恶意攻击，去封装节点必须在转发前验证封装数据包是否可以接受。需要交付给传输层协议的数据包不应该进行这种检查。

#### 4.2.3.7 安全方面的考虑

除了防止DoS攻击外，目前还没有其他关于隧道安全漏洞方面的介绍。防止DoS攻击可以通过源地址过滤技术来避免：在去封装节点明确配置了可以接收哪些IPv4地址为隧道源地址的遂传数据包。

### 4.2.4 6to4 隧道技术

#### 4.2.4.1 基本概念

6to4 过渡机制可以使连接到不支持纯 IPv6 的 IPv4 网络中孤立的 IPv6 子网或 IPv6 站点与其他同类站点在尚未能获得纯 IPv6 连接时彼此间进行通信。

采用这种机制连接的 IPv6 站点或主机不需要 IPv4 兼容的 IPv6 地址或已配置好的隧道。通过这种方式，IPv6 可以获得相对于广域网络很高的独立性，可以跨越许多 IPv4 子网。实现完整的 6to4 机制只需要在边界路由器上增加配置，而对于主机，除了增加一个默认地址选择以外不需要其他修改。



##### 4.2.4.1.1 IPv6 前缀分配

IANA 为 6to4 过渡方案永久地分配了一个具有 IPv6 格式前缀 001[AARCH,AGGR]的 13 比特 IPv6 TLA（Top Level Aggregator）标识符。数值为 0x0002，表示成 IPv6 地址前缀格式为 2002::/16。如果一个用户站点拥有至少一个有效的全球惟一的 32 位 IPv4 地址，本标准称为 V4ADDR，注意该 IPv4 地址必须是由一个地址注册机构分配的（通常通过一个服务提供商），而且不能是一个私有地址[RFC 1918]。那么该用户站点将不需要任何分配申请即可拥有如下的 IPv6 地址前缀。

前缀长度：48 比特

格式前缀：001

TLA 值：0x0002

NLA 值：V4ADDR

6to4 的 IPv6 地址格式如图 4 所示。

| 3 | 13 | 32 | 16 | 64 |
|---|---|---|---|---|
| FP<br>001 | TLA<br>0x0002 | V4ADDR | SLA ID | 接口 ID |

图 4 6to4 的 IPv6 地址格式

这样，该前缀便具有了根据[AGGR]分配常规/48 地址前缀相同的格式。简写为 2002:V4ADDR::/48。在该用户站点范围内，它可以作为有效的 IPv6 前缀使用。

##### 4.2.4.1.2 地址选择

为了在复杂拓扑结构的网络中确保 6to4 的正常运行，必须正确实现源地址和目地地址的选择。目前地址选择机制还在研究中。但是要保证 6to4 能够正确运作的基本选择原则为：

如果一个主机只有一个 6to4 地址，另外一个主机既有一个 6to4 地址也有一个纯 IPv6 地址，那么两者都采用 6to4 地址。

如果两个主机都拥有一个 6to4 地址和一个纯 IPv6 地址，那么两者要么一起采用 6to4 地址，要么一起采用纯 IPv6 地址。优先采用哪种选择应该是可配置的。默认配置应该是两者都采用纯 IPv6 地址。

#### 4.2.4.1.3 IPv4 封装

从 6to4 站点发出的 IPv6 数据包在离开站点时被封装到 IPv4 数据包中，然后进入外部的 IPv4 网络。注意转发 6to4 数据包的 IPv4 接口在概念上等同于一个 IPv6 接口，我们称其为伪接口。V4ADDR 必须在 IPv4 接口上进行配置。

IPv6 数据包被封装到协议类型为 41 的 IPv4 数据包中进行传输。IPv4 头中包含有源和目的 IPv4 地址，其中之一与上述方式构造的 IPv6 前缀中的 V4ADDR 域相同。IPv4 数据包的载荷包含整个被封装的 IPv6 包的内容。其数据包格式如图 5 所示。

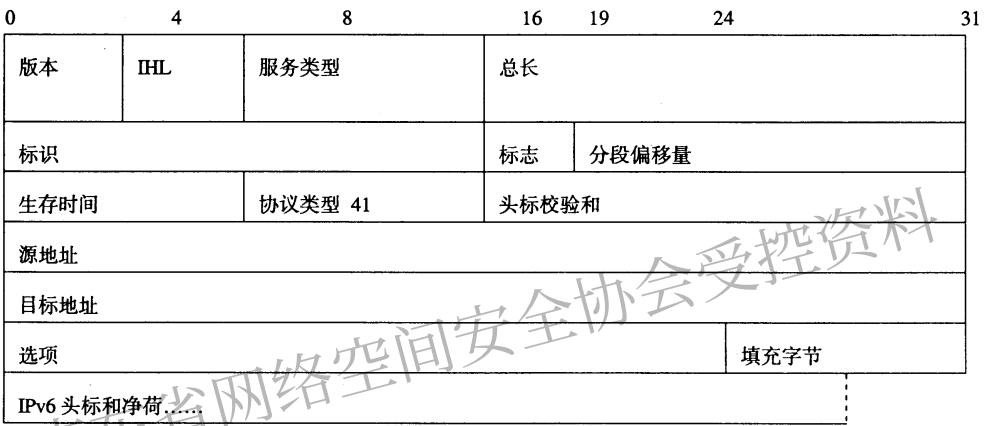





**图 5 IPv6 数据包在 IPv4 中的封装格式**

#### 4.2.4.1.4 链路域地址及 NUD

进行 6to4 封装的 6to4 伪接口所采用（如果需要采用）的链路域地址将按照通用 IPv4 封装 IPv6 的隧道过渡机制中的规则产生。然而，由于一个 6to4 网关无法决定应当发送给哪个适当的链路层（IPv4）地址，所以目前还没有场合需要用到这种链路域地址。

邻居不可达检测 NUD 的处理同样遵照通用的 IPv4 封装 IPv6 的隧道过渡机制中说明的方法。

#### 4.2.4.1.5 最大传输单元（MTU）

MTU 值的处理遵照通用的 IPv4 封装 IPv6 的隧道过渡机制中说明的方法。

如果一个 IPv6 MTU 大小对于某些中间 IPv4 子网而言过大时，将会发生 IPv4 分段。在封装的 IPv4 头中，不应当设置 IPv4“不允许分段”标志。

### 4.2.4.2 6to4 的运行原理

#### 4.2.4.2.1 简单情况——所有站点都以相同的方式工作

6to4 最简单的配置情形是在一些至少具有一个公共 IPv4 互联网连接的站点之间应用。其中的 IPv4 互联网可以是全球互联网，也可以是一个内部的 IP 网络。当连接到全球互联网时，并不要求这些 6to4 站点全都连接到同一个互联网服务供应商。对它们惟一的要求是这些站点中每一个都可以发送协议类型为 41 的 IP 数据包给其他的任何一个互通的 6to4 站点。按照定义，每一个站点都具有一个 6to4 格式的 IPv6 前缀。从而可以为这些地址创建 DNS 记录。如图 6 所示。具有 IPv4 地址 192.1.2.3 的 A 站点将创建一条带有 IPv6 前缀为{FP=001，TLA=0x0002，NLA=192.1.2.3}/48（即 2002:c001:0203::/48）的 DNS 记录。具

有地址9.254.253.252的B站点将创建一条带有IPv6前缀为{FP=001,TLA=0x0002,NLA=9.254.253.252}/48（即2002:09fe:fdfc::/48）的DNS记录。

当B站点中的一个IPv6主机查询A站点中一个主机的DNS表项时，它获得一个带有前缀{FP=001，TLA=0x0002，NLA=192.1.2.3}/48的地址。在两个站点的内部，IPv6的数据包都以常规方式产生并发送。

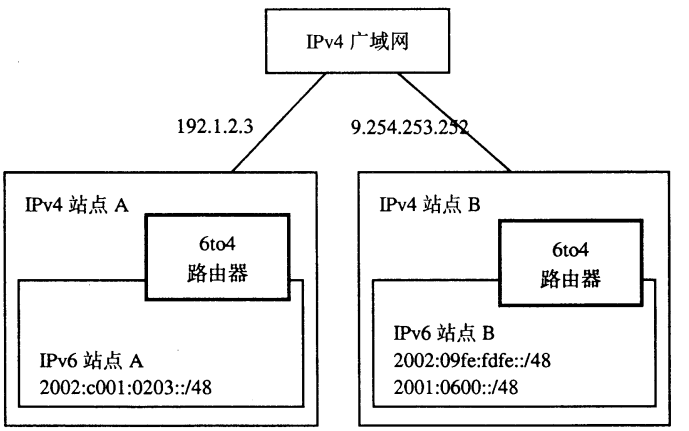



**图6　6to4隧道的网络应用（不带中继路由器）**

在一个6to4站点内，除了带有本地2002:V4ADDR::/48地址前缀以外的6to4数据包将按照处理其他目的地址是非本地IPv6地址的数据包一样处理——即通过默认路由或明确的路由方式路由到6to4边界路由器。

当一个向外发送的数据包到达6to4路由器时，它被封装到IPv4中，并依照6to4的发送规则发送。接收到的数据包则按6to4的解封装规则进行解封装。在整个IPv6转发过程中，只有发送和解封装规则和正常的IPv6转发不同，而且这些不同只发生在边界路由器上。没有任何IPv4路由信息被引入到IPv6路由中，也没有任何IPv6的路由信息被引入到IPv4路由中。

在这种情况下，任意数目的6to4站点都可以不需要任何预先的隧道配置进行互联，而且也没有对IPv4网络有任何特殊的服务要求。所需要的就是在6to4路由器上配置正确的DNS表项以及发送和解封装规则。当然，该路由器应当可以发出正确的IPv6前缀公告[CONF，DISC]。

尽管A站点和B站点的内部都需要运行IPv6路由，但是在这种简单的运行方案下，它们并不需要运行IPv6外部路由协议。IPv4的域间外部协议可以完成全部的外部路由工作。

建议无论在任何情况下，每一个站点的每一个6to4路由器只采用一个IPv4地址，而且该IPv4地址应当是分配给6to4路由器外部通信接口的IPv4地址。因此，单穴（single-homed）6to4站点的路由应当仅采用一个IPv4地址。多穴（Multi-homed）站点的情形将在后面作简单的讨论。

#### 4.2.4.2.2　与带有到IPv6网络的中继组合应用

在过渡到IPv6的过程期间，我们当然希望站点都符合刚才描述的模型（孤立的只连接到IPv4互联网站点），但实际上有很多站点同时也是一些巨大的应用常规IPv6 TLA地址IPv6岛的一部分。6to4站点也要实现与这些纯IPv6岛互联。在6to4模型中，这种互通连接是通过同时拥有6to4和纯IPv6地址的IPv6路由器实现的。尽管它们的工作方式与标准IPv6路由器没什么不同，但本标准为了使它区分于仅支持6to4或仅支持纯IPv6的路由器，把它称为中继路由器（relay routers）。

在6to4域和一个给定的纯IPv6域之间，至少需要一个路由器充当中继节点。该中继节点没有任何特殊，仅仅是一个拥有至少一个逻辑上的6to4伪接口和至少一个IPv6接口的路由器。作为6to4路由器，也应当能够完成6to4的发送和解封装规则。

现在有三种不同类型的路由域需要考虑：

（1）每个 6to4 站点内部的 IPv6 路由，6to4 站点的内部路由按照上述的简单情形所述方式工作。

（2）一个互联了一系列 6to4 边界路由器包括中继路由器的外部 IPv6 路由域。6to4 外部路由域有以下两种实现方式：

不采用任何 IPv6 外部路由协议。所有采用同一个给定的中继路由器的 6to4 路由器都设有一个默认的 IPv6 路由指向该中继路由器。中继路由器可以采用源地址过滤方式选择接收哪些特定的 6to4 通信。

用一种 IPv6 外部路由协议。采用同一给定的中继路由器的一组 6to4 路由器通过采用诸如 BGP4+[RFC 2283，BGP4+]等路由协议从中继路由器处获得通往纯 IPv6 的路由。中继路由器在其 6to4 伪接口上通告它拥有的任何一个纯 IPv6 路由前缀。这些前缀会给出中继路由器能够转发哪些纯 IPv6 区域。对它们的选取需要由路由策略决定。当选择路由公告的传播范围时，网络运营商需要仔细考虑所期望的传输模式和网络拓扑结构。中继路由器会和该中继路由器希望接受的特定的 6to4 路由器建立 BGP 连接关系。尽管这个方案较为复杂，它却提供了比较有效的策略控制，即 BGP4+策略可以决定哪些 6to4 路由器可以采用哪个中继路由器。

（3）每个纯 IPv6 岛的外部 IPv6 路由域。中继路由器必须向纯 IPv6 外部路由域通告一个到 2002::/16 的路由。这个到 2002::/16 的路由通告能在纯 IPv6 路由系统中传播多远则是一个关系到路由策略（routing policy）的问题。既然通常会有多个中继路由器通告该路由，那么网络运营商必须按一定规则对之过滤。选择不正确的策略将导致这个区域内存在潜在的不可达问题或糟糕的传输性能。为防止 IPv4 路由表成分混入 IPv6 路由表，不能将比 2002::/16 更精确的 6to4 前缀能通告进入纯 IPv6 路由域中。因此，拥有纯 IPv6 连接的 6to4 站点不允许在该连接上通告有到 2002::/48 的路由，并且所有的纯 IPv6 网络运营商必须过滤掉所有前缀长度大于/16 的 2002::路由公告。

除了具有一个 6to4 连接外还拥有至少一个纯 IPv6 连接的站点至少有一个非 2002::前缀的 IPv6 前缀。这些站点的 DNS 表项将反映这一点，DNS 查询时将返回多个地址。如果有两个这样的站点需要互相通信，具体是采用 6to4 路由还是采用纯 IPv6 路由将由主机本身（甚至应用程序）的 IPv6 地址选择来决定。

现在回过头来再考虑一下前一节的例子。在图 7 中，假定 B 站点中的一台 IPv6 主机查询 A 站点中一台主机的 DNS 表项，DNS 返回具有不同前缀的多个 IPv6 地址。

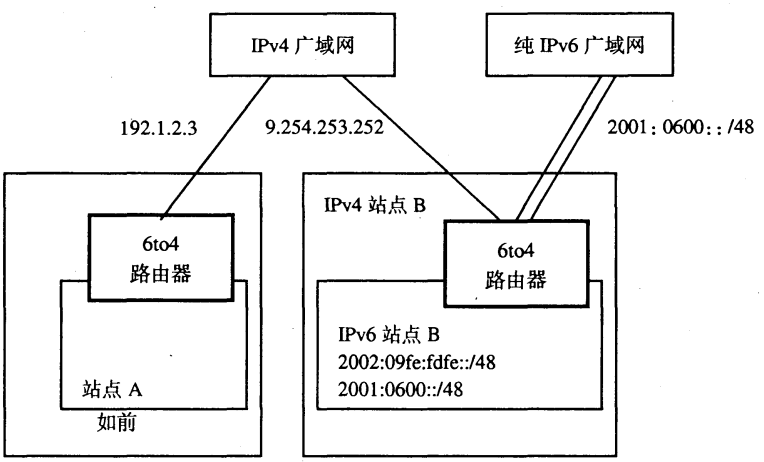



**图 7　6to4 隧道的网络应用（带中继路由器）**

如果主机按照多前缀的某些规则选取了 6to4 前缀，那么它将以前缀{FP=001，TLA=0x0002，NLA=192.1.2.3}/48 构造 IPv6 目的地址发送数据包。数据包的源地址需要采用前缀为{FP=001，

TLA=0x0002，NLA=9.254.253.252}/48 的地址，这样才能保证双向连接的建立。

（1）使用 ISP 中继

以上情况都是假定了中继路由器是由协商好的 6to4 用户站点提供的。实际上已经提供了纯 IPv6 连接的 ISP 也可以运营中继路由器。从技术角度上，这与前面的情况完全相同——站点 B 只不过是 ISP 的一个内部 6to4 站点而已，而且很有可能只包含一个系统（即中继路由器本身）。

（2）中继路由器配置概要

中继路由器在其纯 IPv6 接口上参与 IPv6 单播路由协议，也可以同时在 6to4 伪接口上参与 IPv6 单播路由协议，但它们工作于相对独立的不同路由域内。

中继路由器同样在其用来支持 6to4 的 IPv4 接口上参与 IPv4 单播路由协议，这里不多做讨论。

在其连接纯 IPv6 网络的接口上，中继路由器必须通告一个到 2002::/16 的路由。在该接口上，不允许通告很长的 2002::路由前缀。通告的传播范围由纯 IPv6 路由域中的路由策略决定，从而限定了该域中中继路由器的可见性。

当中继路由器收到的下一跳（next hop）IPv6 地址能匹配 2002::/16 的 IPv6 数据包时，中继路由器将把它发给 6to4 伪接口。

（3）不使用 BGP4+

如果在 6to4 外部路由域中没有运行 BGP4+,那么中继路由器将被配置成为只接收和转发其 6to4 客户站点的 IPv6 通信量。中继服务器为之服务的每一个 6to4 路由器将被配置成为具有一个默认的 IPv6 路由指向该中继路由器（例如，A 站点的默认 IPv6 路由::/0 将指向前缀为 2002:09f4:fdfc::/48 的中继路由器地址）。

（4）使用 BGP4+

如果在 6to4 外部路由域中采用了 BGP4+，那么中继路由器将在其 6to4 伪接口上通告纯 IPv6 路由，同时只与它所服务的 6to4 路由器建立连接。选择哪些路由进行通告是由路由策略决定的，但这些路由必须选自中继路由器通过其纯 IPv6 接口可达的路由之中。在最简单的情形下，就是通告一个通往全体 IPv6 地址的默认路由。当采用了多个 6to4 中继路由器时，将根据期望的路由策略通告更为详尽的路由前缀。BGP4+的使用是完全标准化的，因此本标准不再做规定。

（5）不中继

这种情况发生在一个站点拥有一个既有 6to4 伪接口同时也有纯 IPv6 接口然而却并不想充当中继的路由器时。这种站点不允许向纯 IPv6 域通告有到 2002::前缀的路由，也不允许向 6to4 域通告任何纯 IPv6 路由前缀或默认的 IPv6 路由。在 6to4 域内部，它的运行和简单情况下的基本 6to4 完全一致。

#### 4.2.4.3 其他细节

##### 4.2.4.3.1 发送及解封装规则

与标准的 IPv6 转发惟一的区别就是每一个 6to4 路由器（且仅仅是 6to4 路由器）必须实现下述的附加发送和解封装规则。

发送规则中，“下一跳”指的是数据包将发往的下一个 IPv6 节点——并不一定是最终节点而更可能是由 IPv6 路由机制（mechanism）指出的下一个 IPv6 邻机。如果最终目的地址是一个 6to4 地址，遵照此规则，它将被视作下一跳。如果最终目的地址不是一个 6to4 地址同时也不是一个本地地址，那么路由决定的下一跳将是一个中继路由器的 6to4 地址。

6to4 路由器的附加发送规则：

如果某个 IPv6 包的下一跳 IPv6 地址与前缀 2002::/16 相匹配，并且不与任何本地站点地址前缀相匹配，那么选用一种安全机制做安全检查，将 IPv6 数据包封装入 IPv4 包中，该 IPv4 数据包的目的地址=下一跳 IPv6 地址中提取出的 NLA 值 V4ADDR；送入待进行 IPv4 转发的数据包队列中。

另外还要求能够实现一个简单的对协议类型为 41 的 IPv4 数据包进行解封装的规则。

6to4 路由器的附加解封装规则：

采用一种安全机制做安全检查；去掉 IPv4 头标；将数据包递交给本地 IPv6 路由。

#### 4.2.4.3.2 使用隧道连接到 IPv6 域

没有纯 IPv6 网络连接的 6to4 站点可以通过一个“已配置好的隧道”连接到一个具有 IPv6 通路的路由器（不一定非得是 6to4 路由器），从而获得到 IPv6 连接。这类隧道可以通过采用一个 IPv4 泛播地址获得自动地址配置，也可以采用隧道代理（tunnel broker）。但这已越出本标准的范围。

#### 4.2.4.3.3 被分割的情形

如果在纯 IPv6 和 6to4 域之间存在多个中继路由器，6to4 域中不同的部分将选用不同的中继。纯 IPv6 域中引入的惟一复杂就是 2002::/16 路由前缀通告的传播范围。对于任意的 BGP4+通告来说，路由策略必须正确地定义它们的传播范围，从而才能保证发往 2002::/16 地址的通信信息沿着所期望的途径传输。

如果多个 IPv6 末稍区域（stub）全都通过全球 IPv4 互联网实现 6to4 互联，那么这只是基本情形的一个简单推广，并没有产生任何新的问题。如图 8 所示。



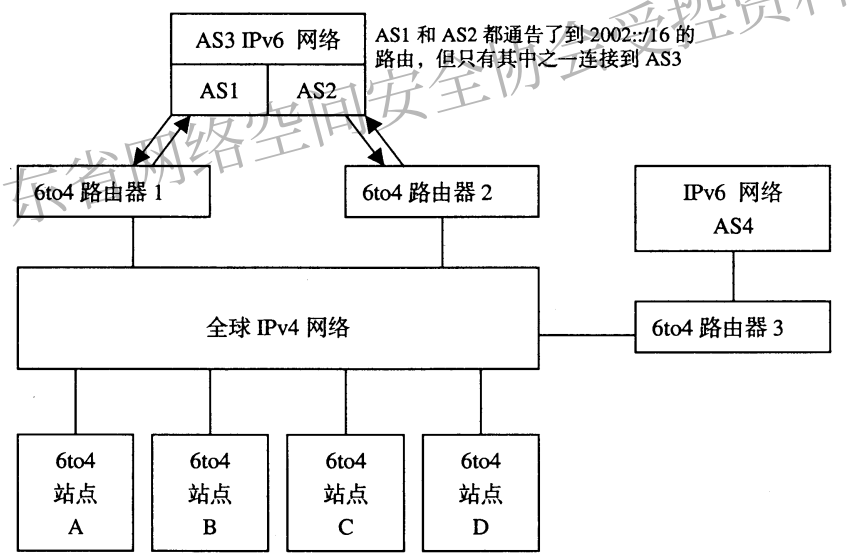



**图 8 IPv4 域未被分割情况下的 6to4 应用**

如果多个 IPv6 末稍区域通过多个互不相联的 IPv4 网络（即被分割了的 IPv4 域）连接在一起，那么整个 6to4 域也是被分割的，这是一种必须避免发生的情形。图 9 说明了为什么这种情形下无法工作。由于中继 2 无法看到中继 1 发出的 2002::/16 通告（反之亦然），因此 A 站点和 B 站点将无从获得到 C 站点和 D 站点的连接。

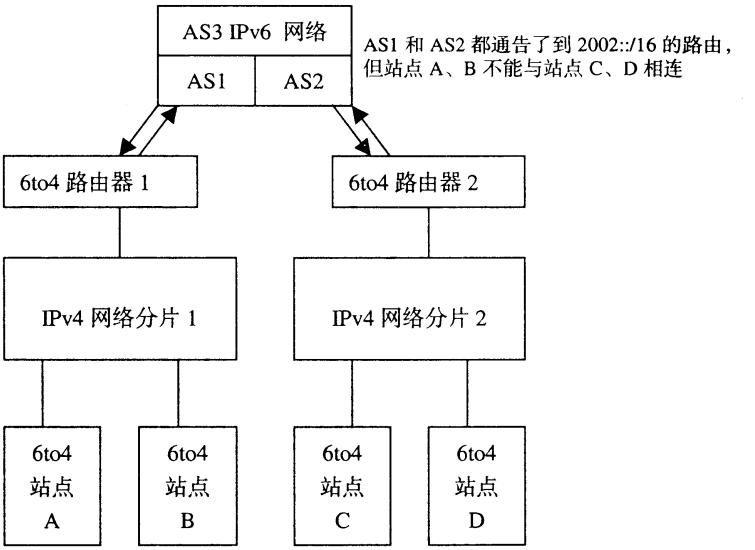



**图 9　IPv4 域被分割情况下的 6to4 应用**

#### 4.2.4.3.4　多穴（multihoming）

具有 IPv4 多穴的 6to4 站点可以通过为每一个 IPv4 边界路由器采用一个 2002::前缀的方法扩展 6to4，从而通过同时采用多个 IPv6 前缀和多个中继路由器来获得一个简单的 IPv6 多穴模式。

#### 4.2.4.3.5　过渡方面的考虑

站点如果已经遵循了上述路由公告和地址选择规则，则可以通过一段长期共存阶段从 6to4 过渡成为纯 IPv6 连接，在其间涉及到的过程如下：

采用适当的实现方式在站点内运行 IPv6，包括完全的纯 IPv6、[6over4]、隧道均可。

配置一个连接 IPv4 网络的边界路由器（或带有 IPv4 NAT 的路由器），使之能够支持 6to4，包括在本地通告适当的 2002::路由前缀。使用该路由前缀配置 IPv6 DNS 表项。此时，6to4 已经可以使用，同时站点也获得了一个“免费的”IPv6 前缀。

连接一个中继服务器节点与纯 IPv6 域互联。该中继可以是属于另外的一个 6to4 协作站点，也可以是一个 ISP 服务。如果在 6to4 外部路由域中没有采用外部路由协议，那么该站点的 6to4 路由器将被配置成为具有一个默认的 IPv6 路由指向中继路由器的 6to4 地址。如果采用了如 BGP4+的外部路由协议，那么站点的 6to4 路由器将被配置成可以建立适当 BGP 互联关系。

当可以获得纯 IPv6 外部连接时，在两个边界路由器的配置中都再加入一个纯 IPv6 前缀和 DNS 配置。此时地址选择规则将决定何时采用 6to4，何时采用纯 IPv6。

当决定终止采用 6to4 时（可能是几年以后），去掉 6to4 配置。

#### 4.2.4.3.6　与防火墙、NAT 或 RSIP 共存

边界路由器上运行防火墙并不能影响 6to4 机制运行。

如果站点只有很有限的全球 IPv4 地址空间，并且运行了 IPv4 网络地址翻译（NAT），上面提到的所有机制仍然有效。NAT 网关（box）同样必须包含一个具备完整功能的 6to4 IPv6 路由器。V4ADDR 采用的地址就是一个分配给 NAT 的全球惟一 IPv4 地址。如图 6 所示，6to4 路由器同样也可以是站点的 IPv4 NAT 网关，两个网络的网关分别拥有全球惟一的 IPv4 地址 192.1.2.3 和 9.254.253.252。

这种 6to4 路由器结合 IPv4 NAT 的方式可以自动地提供给 6to4 站点一个全球惟一的 IPv6 /48 前缀。这样 NAT 后的所有主机将成为 IPv6 主机，不需要为之分配额外的地址空间，也不需要 Internet 服务商的

干预。这些 IPv6 主机也无需地址翻译。

当主机距全球惟一的 IPv4 地址之间存在多个 NAT 时，由于只有最外层的 NAT 才有惟一的 IPv4 地址，所以这将出现更为复杂的情形。这种情形下，所有的 IPv6 主机必须采用源自 2002::前缀和最外层 NAT 的 IPv4 地址所构建成的地址。内层 NAT 的 IPv4 地址不是全球惟一的，在 6to4 机制中不起任何作用，6to4 封装和解封装只能发生在最外层 NAT 处。

专用领域 IP 机制（RSIP）也可以和 6to4 共存。如果某个 6to4 边界路由器与 RSIP 边界路由器已结合起来，那么它可以支持 IPv6 主机采用 6to4 地址，IPv4 主机采用 RSIP，或双栈主机同时采用 6to4 地址和 RSIP。RSIP 功能为动态全局 IPv4 地址分配提供了精密控制，而 6to4 功能则为每一个主机提供了稳定的 IPv6 全局地址。正如与 NAT 共存的情形一样，用以构建站点的 2002::前缀的 IPv4 地址就是 RSIP 边界路由器的全球地址之一。

#### 4.2.4.3.7 对路由带来的影响

IGP（站点）将对本地站点的 2002::/48 前缀按照其纯 IPv6 站点前缀一样处理。如果 2002::/16 不是按照默认路由来处理，则应该有一个 IGP 路由指向 2002::/16 前缀。

EGP（即 BGP）将从中继路由器向纯 IPv6 域发路由通告 2002::/16，该通告的传播范围由路由策略限定。这是 BGP 通告的惟一非纯 IPv6 前缀。

为访问纯 IPv6 域，6to4 路由器有必要获得中继路由器的路由。在最为简单的情形下，将有一个手工配置的默认 IPv6 路由指向中继路由器。这样的路由可用来建立交换一些 IPv6 路由的 BGP 会话。

#### 4.2.4.3.8 防止路由环路

鉴于 6to4 对于 IPv4 路由没有任何影响，所以它不可能在 IPv4 中引入路由环路。又由于 2002::前缀与标准 IPv6 前缀具有一样的作用，所以除非配置错误，否则它们不会产生新的路由环路问题。一种非常危险的错误配置是在 6to4 外部路由域中通告 2002::/16 前缀，因为这将使得所有的 6to4 通信信息都被发给通告该前缀的站点。如果之后 6to4 路由器将遣返非本地的 6to4 通信信息，那么则会形成路由环路。

#### 4.2.4.3.9 组播和泛播

由于广域的 IPv4 组播支持并不能随时获得，所以（与[6OVER4]不同）6to4 机制只能假设其下层的 IPv4 网络的只有单播的支持。所以需要有一种 IPv6 组播路由协议[MULTI]。

泛播地址空间的分配是与 2002::前缀相兼容的，也就是说，可以在 6to4 站点内部使用由这种前缀构成的泛播地址。

#### 4.2.4.3.10 ICMP 数据包

ICMP“不可达”及其他由 IPv4 路由系统返回的信息将返回给生成封装的 2002::数据包的 6to4 路由器。然而，由于在“不可达”数据包中缺乏足够的信息，该路由器通常无法返回一个 ICMPv6 数据包给源 IPv6 节点。这意味着对于 IPv6 而言，6to4 之间的 IPv4 连接是一个不可诊断的链路。

#### 4.2.4.3.11 IANA 考虑

除了已经分配的特殊 TLA 值 0x0002 外，不需要其他 IANA 号分配。

#### 4.2.4.3.12 安全性考虑

使用者应当明白一点，除了应当考虑可能来自对 IPv6 的攻击以外，还应当考虑对 IPv4 的安全性攻击。然而，为高效率起见，却应当避免同时在 IPv4 和 IPv6 层上应用 IP 安全。例如，如果 IPv6 是加密运行的，除非通信信息分析被认为是危险的，否则 IPv4 加密运行将是多余的。而如果 IPv6 是加密运行的，那么对 IPv4 实施加密运行几乎不能增加任何安全性。相反，IPv4 安全并不能保护 IPv6 通信信息一旦它离开 6to4

域。因此，即使可以获得 IPv4 安全，仍然应当实现 IPv6 安全。

默认情况下，6to4 通信信息可被任何可以接收常规 IPv4 通信信息的节点接收和解封装。如果可能存在安全性攻击的风险，那么就应该采用额外的基于信源地址的过滤。一种可能加入的检查就是核对封装的 IPv4 地址是否与 2002::前缀中的 IPv4 地址相一致。如果采用了这种检测方法，就必须配置它对例外情况的处理使之能够接受中继路由器发出的通信信息。

无论何种情况下，封装者和解封装者都必须丢弃源地址或目的地址被嵌入一个非全局单播地址格式的 V4ADDR 数据包。明确地说，[RFC 1918]中定义的 IPv4 地址、广播地址、子网广播地址、组播地址和环回地址都是不可被接收的。

#### 4.2.5 6over4 机制

6over4 使得没有直接与 IPv6 路由器相连的孤立的 IPv6 主机通过 IPv4 组播域作为它们虚拟链路层形成 IPv6 的互联。如果需要实现 IPv6 路由时，就需要至少有一个采用 6over4 的 IPv6 路由器和该 6over4 主机连接在同一个 IPv4 的组播域中。

采用该机制互联的主机并不需要 IPv4 兼容地址或者是配置隧道，通过这种机制，IPv6 可以独立于底层的链路而且可以跨越 IPv4 子网。

以下描述了传输 IPv6 数据包的帧格式以及在 IPv4 组播域上如何形成 IPv6 链路地址。

##### 4.2.5.1 MTU

默认的通过 IPv4 域传输的 IPv6 数据包 MTU 大小是 1480 字节，也可以调整路由器通告[DISC]中携带的 MTU 选项值或者直接在节点上手工配置 MTU。如果 IPv6 的 MTU 大小对于一些中间 IPv4 子网太大的话，将会引起 IPv4 的分段。值得注意的是，在这种情况下，不要设置 IPv4 的“不分段”字段。

##### 4.2.5.2 数据包格式

IPv6 数据包被封装在协议类型为 41 的 IPv4 数据包[RFC791]中传输。IPv4 头标中包含了源和目的 IPv4 地址。IPv6 头标和载荷紧跟在 IPv4 数据头标之后。如果 IPv4 头标包含扩展选项，则填充应该添加在 IPv4 头标，这样可以保证 IPv6 头的起始位置与数据链路头末端有 32 位的偏移。TTL 字段应该设置成一个较小的值，以防止数据包从 IPv4 域漏出。TTL 字段必须是一个可配置的参数，推荐的值是 8。帧格式如图 10 所示。

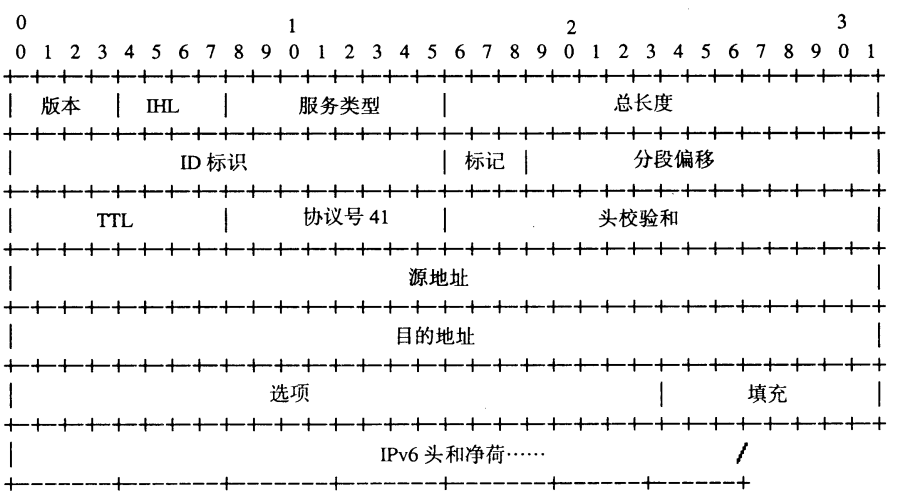



图 10 IPv6 数据包在 IPv4 中的封装格式

##### 4.2.5.3 无状态地址自动配置（Stateless Autoconfigurationg）和链路域地址（Link-local Address）

IPv4 接口的接口标识符（interface identifier）[AARCH]是该接口的 32 位 IPv4 地址（与它在 IPv4 头

标中的顺序一致）经左边填充后形成的 64 位标识。注意，“广域/局域位”置 0，表示接口标识符并不是全局惟一的，当主机在物理接口上有多个 IPv4 地址时，需要进行决策选择一个地址。

除了在后面扩展性和过渡问题中提到情况中需要用到 128 位地址前缀外，用于 IPv4 接口的无状态自动配置[CONF]的 IPv6 地址前缀必须是 64 位。

用于 IPv4 虚拟接口（virtual interface）的 IPv6 链路域地址是通过在前缀 FE80::/64 后添加接口标识符形成的，如图 11 所示。

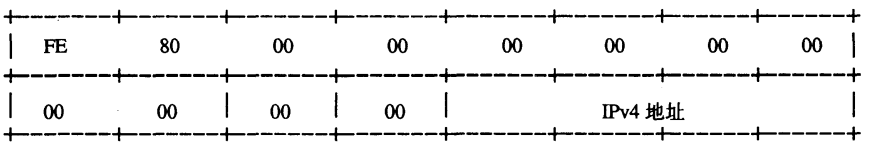



**图 11　6over4 的 IPv6 地址格式**

#### 4.2.5.4　地址映射 – 单播

映射 IPv6 地址到 IPv4 虚链路层的过程在邻机发现机制中做过描述。当链路层是 IPv4，源/目的链路层地址选项形式如图 12 所示。由于长度字段是 8 字节为单位，因此值为 1。


**图 12　源/目的链路层地址格式**

类型：1　用于源链路层地址；

2　用于目的链路层地址。

长度：1（以 8 个字节为单元）；

IPv4 地址：32 位的 IPv4 地址采用网络字节顺序。

#### 4.2.5.5　地址映射 – 组播

具有组播目的地址 DST 的 IPv6 数据包必须利用图 13 所示的映射传送到组织域（Organization-Local Scope）的 IPv4 组播地址。这些 IPv4 组播地址应该来自于 239.192.0.0/16 这样一个组织域地址块（Organization-Local Scope address）的子块。如果这些地址都不可用，则从扩展块中寻找。

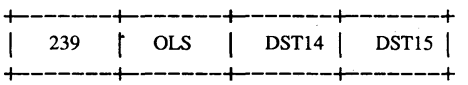



**图 13　IPv6 组播地址到 IPv4 组播地址的映射**

DST14、DST15：IPv6 组播地址的最后两个字节

OLS：来自于组织域地址块，通常应该是 192。

#### 4.2.5.6　扩展性和过渡问题

除了上面的组播机制中 MTU 值略为减少之外，6over4 与纯 IPv6 有相同的可扩展属性。在 ATM 网上，IPv4 组播采用了相对复杂的机制，因此在 IPv6 over IPv4 over ATM 将没有纯 IPv6 over ATM 效果好。

IPv6 over IPv4 机制允许一个站点同时运行 IPv4 和 IPv6，而不需要将 IPv6 主机配置成具有 IPv4 兼容的地址或具有隧道。IPv6 路由器和主机的接口需要支持 6over4 模式。

一个站点在 IPv6 过渡初期，可以将 IPv6 边界路由器的连接 IPv4 域的接口配置成支持 IPv6 over IPv4，在另一个连接 IPv6 域的接口上配置 IPv6。任何一个在 IPv4 域支持 6over4 的主机都可以与这些路由器或 IPv6 域进行自由通信，而不需要手动配置隧道，也不需要 IPv4 兼容地址的主机。

在过渡过程中，路由器可能需要通告至少两个 IPv6 前缀，一个用于 Native LAN（如以太网），一个

用于 6over4。与分配给 IPv6 子网的任何前缀一样，后者（即 6over4，它不是真正的 IPv6 子网）在其地址作用域内必须是惟一的，不论采用的是站点域还是广域的寻址方法）。

注意：当路由器可以在同一个物理接口上同时处理 native LAN 和 6over4，在无状态自动配置时，会有一个时期用到 IPv6 的链路域地址，这两种情况都用到前缀 FE80::/64。为了区别，链路域地址前缀长度必须是 128。

#### 4.2.6 隧道代理

##### 4.2.6.1 隧道代理概述

隧道代理（TB）提供一种简化配置隧道的方法，可以减少繁重的隧道配置工作。隧道代理的思想就是通过提供专用的服务器作为隧道代理，自动地管理用户发出的隧道请求。用户通过 Tunnel Broker 能够方便的和 IPv6 ISP 建立隧道连接，从而访问外部可用的 IPv6 资源。隧道代理这种过渡机制对于在 IPv6 早期吸引更多的 IPv6 使用者方便快捷的实现 IPv6 连接有很大的益处。同时也为早期的 IPv6 提供商提供了一种非常简捷的接入方式。

隧道代理和 6to4 的不同就在于它们分别服务于两种不同的 IPv6 区域。

隧道代理非常适合于独立的小型 IPv6 站点，特别是独立的分布在 IPv4 互联网中的 IPv6 主机需要连接到已有 IPv6 网的情况。

6to4 的设计初衷是在 IPv4 ISP 提供完全的 IPv6 服务以前，就可以使一些分布的、独立的 IPv6 站点通过 IPv4 的网络实现互联互通。



##### 4.2.6.2 隧道代理机制模型

隧道代理可以看作是一个虚拟的 IPv6 的 ISP，为已经连入 IPv4 互联网的用户提供 IPv6 连接。理想的情况是出现很多的隧道代理提供隧道代理的服务，用户只需要从中挑选一个使用。

TB 机制的模型是基于图 14 中的几个功能单元。

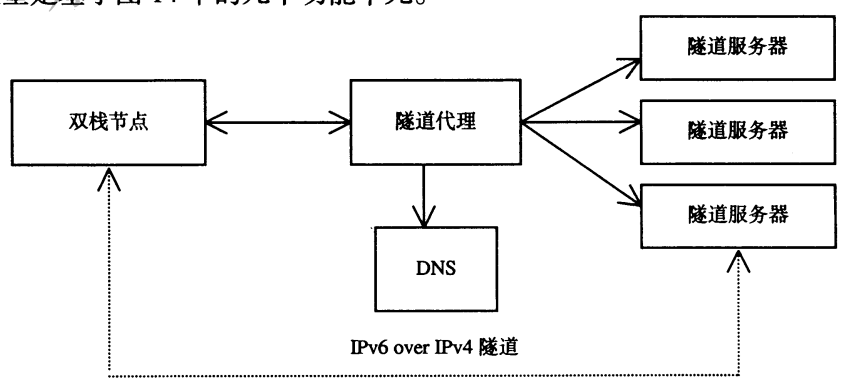



图 14 隧道代理机制模型

###### 4.2.6.2.1 隧道代理（TB）

用户在 TB 上进行注册和启动隧道。TB 根据用户的需求，代表用户负责隧道的建立、修改和删除。

基于可扩展性原因，隧道代理可以在几个隧道服务器之间分配网络侧隧道端点上的负载。当隧道建立、修改或删除时，隧道代理将发送配置指令到相应的隧道服务器。隧道代理还可以完成注册用户 IPv6 地址和名字到 DNS 服务器。

隧道代理必须是 IPv4 可编址的，它也可以被 IPv6 编址，但这不是强制性的，隧道代理和隧道服务器之间的通信既可以采用 IPv4 也可以采用 IPv6。

###### 4.2.6.2.2 隧道服务器（TS）

隧道服务器是连接到因特网上的一个 IPv4/IPv6 双栈路由器。当它接收到隧道代理发送过来的一个配置指令以后，它将会创建、修改或删除每一个隧道的服务器端，同时，隧道服务器也可能会对每一个活动隧道维护一些应用统计信息。

#### 4.2.6.2.3 隧道代理的工作过程

为了连接到 TB，客户端应该首先提供相应的身份证明。为了防止非法的使用，客户端与 TB 之间必须有预先配置或自动配置的安全措施。因此 TB 也可以看作是 IPv4 连接到 IPv6 用户的访问控制服务器。一旦客户的接入请求得到允许，客户需要提供至少以下的信息：

- 隧道客户端的 IPv4 地址；
- 客户端登记到 DNS 使用的域名；
- 客户端的功能（例如，主机或路由器）。

如果客户端是一个 IPv6 路由器，而且要为多个 IPv6 主机提供连接，客户端必须提供需要 IPv6 地址的数目。从而使隧道代理可以分配所需的地址前缀，而不是只分配一个单一的 IPv6 地址。

TB 接到客户端的请求后做如下的操作：

- 指定一个 TS 作为网络侧的隧道端点；
- 选择 IPv6 前缀分配给用户，前缀的长度可以是 0~128 之间；
- 设定隧道的生存期；
- 在 DNS 上自动登记分配给隧道端点的全球 IPv6 地址；
- 配置隧道的服务器端；
- 通知客户相关的配置信息，包括隧道的参数和 DNS 域名。

上述的配置步骤执行后，客户端与一个 TS 之间的 IPv4 封装 IPv6 的隧道就建立好并开始工作了，TB 用户就可以访问 6bone 或者任何连在 TS 上的 IPv6 网络。



#### 4.2.6.2.4 IPv6 地址分配

隧道两端的 IPv6 地址必须是属于 TB 所管理的 IPv6 地址空间中的全局 IPv6 地址。这些 IPv6 地址的生存期比用户间的 IPv4 连接的生存期要长。这样使得那些通过拨号接入因特网而动态获得 IPv4 地址的用户可以获得较为稳定的 IPv6 地址和 DNS。

#### 4.2.6.2.5 隧道管理

活动（active）的隧道会消耗隧道服务器上的内存资源和处理时间，因此在设计隧道管理机制时应尽量使得当前建立而未被使用的隧道数量越少越好。

每个由 TB 生成的隧道至少要分配一个生存期，在过期之后，如果没有得到明确的延长生存期的请求，可自动被删除。然而这种方法对于通过动态地址访问因特网的用户并不适用，因为用户每次建立连接都将得到新的 IPv4 地址，因此只好重新建立一个新的隧道或者更新原有隧道的配置。这样导致一个新建的隧道在使用很短的时间后将不再被使用。这种情况需要一种更有效的隧道管理机制，一种方法是 TS 向 TB 周期性地发送每个活跃隧道的 IPv6 业务量统计信息等数据，这样 TB 在一段时间没有收到这种周期性的数据之后就可以删除隧道，而不需要等待过了生存期。另一种方法是在客户和 TS 之间（或 TB 和客户之间）执行一种隧道管理协议或保活（keep-alive）机制。这样当用户断开连接时，可以迅速释放隧道。这种方式的缺点是要求升级用户侧的软件用于支持 Ad Hoc 保活机制。

另外，在客户断开连接之后，跟踪隧道的配置也可能有一定的价值。这样用户重新上线后，仍可以重用之前分配的 IPv6 地址。

#### 4.2.6.2.6 客户机、TB、TS 以及 DNS 的之间的交互

下面概要的介绍客户机 TB、TB-TS、TB-DNS 之间交互时的情况。

TB 和用户的交互可以基于 HTTP。例如，用户可以在隧道代理上运行的 Web 服务器上的申请表上填写自己的配置信息，服务器会回应一个 HTML 的网页通知用户服务器端已经建立了隧道连接，并同时将该隧道的配置信息显示出来。

然后留给用户的工作就是在用户端配置隧道端点，如果这部分工作可以自动实现最好。

对于隧道代理和隧道服务器之间的交互也有几种方式实现。例如，可以采用一系列的基于 IPSec 之上的简单 RSH 命令。还可以采用 SNMP 或采用其他的网络管理解决方案。

最后，TB 控制的自动 DNS 更新（就是在隧道代理用户预留的 DNS 区内添加和删除 AAAA，A6 以及 PTR 记录）需要采用动态 DNS 更新协议。TB 可以采用一系列简单的 RSH 命令动态的更新在 DNS 服务器上的直接和反向数据库。

#### 4.2.6.3 约束条件

这种机制不适用于用户使用 NAT 与外界通信的情况。

#### 4.2.6.4 安全性考虑

在隧道代理体系中，所有功能单元之间的交互都需要采用安全机制保护。

- 客户和 TB 的交互；
- TB 和隧道服务器的交互；
- TB 和 DNS 的交互。

以上各种交互中采用哪一种安全技术要在具体的实现过程中选择。

对于客户与 TB 的交互中，因为 HTTP 的应用使得广泛的安全机制可以利用。例如 SSL（Secure Socket Layer）在 Web 服务器上对发送和下载进行加密。还可以采用简单的用户名密码程序来实现访问控制。

对于 TB-TS 之间的交互，可以采用 SNMP。如果在 TB-DNS 交互中采用的是动态 DNS 更新程序，安全性考虑和 RFC 2575 中讨论的完全相同。另外，如果采用基于 RSH 命令的简便方法，也可以采用标准的 IPSec。

如果客户端的配置是通过 TB 提供的脚本实现的话，在执行这些脚本时必须给这些脚本很高的权限，因为要实现对一些接口的配置管理。这样作显然是有安全漏洞的，应该只是在隧道代理的使用初期才使用的方案。利用 MIME 在 HTTP 上传输隧道配置参数会提高安全性能。

另外，一个拨号上网的用户在停用通过 TB 建立的隧道之前就挂断和因特网的连接时，会发生机密泄漏。因为隧道服务器会继续发送 IPv6 隧道包到老的 IPv4 地址，该地址可能已经被分配给另一个主机使用。这个问题可以通过保活机制（keep-alive）解决，就是在每个隧道上都使 TB 立即停止向断开连接的用户发送 IPv6 数据。

最后，TB 必须防范可能出现的恶意攻击。它们可能会同时申请大量的隧道连接从而耗尽隧道服务器的资源，可以采用简单的机制来限制单一个用户一次申请隧道连接的个数来实现对上述情况的保护。

### 4.2.7 ISATAP

#### 4.2.7.1 介绍

ISATAP（the Intra-site Automatic Tunnel Addressing Protocol，站内自动隧道寻址协议）可以使IPv4站点内的双栈节点通过自动隧道接入到IPv6路由器，允许与IPv6路由器不共享同一物理链路的双栈节点通过IPv4自动隧道将数据包送达IPv6下一跳。

ISATAP过渡机制采用一个内嵌IPv4地址的IPv6地址，无论站点采用的是全球或是私有的IPv4地址，都可以在站点内采用IPv6-in-IPv4自动隧道技术。ISATAP地址格式既可以采用站点单播IPv6地址前缀也可以采用全局单播IPv6地址前缀，即可以能够支持站点和全局的IPv6路由。

#### 4.2.7.2 适用性声明

ISATAP具有以下特征：

- 利用自动 IPv6-in-IPv4 隧道技术，将站点的 IPv4 体系结构视为 IPv6 的一个 NBMA 链路层。
- 在边界网关上不影响聚合范围的情况下，能够在 IPv4 站点内部署新的 IPv6 主机。
- 不需要站点提供特殊的 IPv4 服务（例如，多播等）。
- 支持无状态地址自动配置和手工配置两种方式。
- 支持采用非全局惟一 IPv4 地址的网络（如采用私有地址的时候）。
- 和其他过渡机制相互兼容（如 6to4 机制）。

#### 4.2.7.3 基本 IPv6 操作

ISATAP链路把IPv4自动隧道当作IPv6链路层传输数据包，也就是说，IPv6把站点的IPv4体系结构当成NBMA链路。

##### 4.2.7.3.1 接口标识和单播地址

ISATAP接口标识采用修改的EUI-64格式，它是由32bit字符串“00-00-5E-FE”后加上ISATAP链路上的IPv4地址组成的。

因此，全局和本地ISATAP地址格式如图15所示。

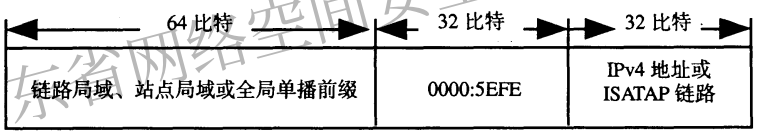



图 15 ISATAP 地址格式



##### 4.2.7.3.2 ISATAP 链路/接口配置

在站点内，ISATAP链路包括一个或若干个下层链路（underlying link），这些链路利用IPv4隧道传送IPv6包文。

ISATAP接口是基于ISATAP链路配置的，每一个IPv4地址被当作ISATAP链路层地址分配给一个下层链路。

ISATAP链路上的邻居发现提供了等价于其他NBMA媒体类型的单播虚电路（VCs）功能。邻居状态信息可能被保存在邻居缓存中。

##### 4.2.7.3.3 链路层地址选项

在ISATAP链路层地址选项中，[NTL]和[STL]域（引自RFC 2491）编码是0，[NBMA Number]域编码为4字节的IPv4地址。

##### 4.2.7.3.4 多播和泛播

ISATAP链路上的多播/泛播仿真机制（如MLD适配、PIM-SM、MARS等）目前没有规定，以后另文处理。

#### 4.2.7.4 ISATAP 自动隧道要求

普通隧道机制一般都能应用在ISATAP中，但要注意如下几点：

（1）支持双IP层操作

ISATAP节点提供完整的IPv4和IPv6实现，并且能够收发IPv4和IPv6包。

（2）封装/解封

ISATAP采用自动隧道封装技术。另外，封装在ISATAP链路上的数据包的IPv6下一跳地址必须是一个ISATAP地址。其他包被丢弃，同时给源节点发一个错误码为3（目标地址不可达）的ICMPv6包。

（3）隧道MTU和分段

ISATAP自动隧道接口可以配置在具有不同最大传输单元（MTU）的底层链路上。IPv6接口的最小MTU是1280字节，但是ISATAP接口需要考虑下面的情况：

- 几乎所有的 IPv4 节点连接的物理链路的 MTU 都是 1500 或者更大（如以太网）；
- IPv4 子网的链路封装（如 VPN）可以出现在若干路径上；
- 一般的 VPN 接口采用 1400 字节的 MTU。

为了最大可能地减少IPv4分段，最好的部署情况是，ISATAP接口MTU或者叫链路MTU应该不超过1380字节（1400减20字节的IPv4封装）。当采用动态MTU发现机制或者配置静态MTU并且在IPv4站点内分段被认为是可以接受的，那么链路MTU可以被设置为更大的值。

当没有采用动态MTU发现机制时，ISATAP链路层在封装IPv6包时，在IPv4头中应该关闭不可分段标志。

（4）IPv4 ICMP错误处理

IPv4的ICMP错误和ARP失败均被当作一个链路层错误处理。

（5）本地采用的IPv6单播地址

本地的单播地址格式请参见接口标识和单播地址一节。

（6）入口过滤

采用通用的隧道入口过滤机制，特别强调的是，ISATAP节点转发解封装的数据包前必须证明隧道源地址是可以接受的。



#### 4.2.7.5 ISATAP 链路的邻居发现

ISATAP 链路应该实现 IPv6 邻居发现中的重定向、邻居不可达检测和路由下一跳选择。下面重点描述 ISATAP 的地址解析相关机制。

##### 4.2.7.5.1 地址解析和邻居不可达检测

ISATAP地址是由链路层地址（IPv4）通过静态计算得到的，也就是，最后4个字节被看作IPv4地址。

在静态地址解析之后，主机应该通过发送邻居请求（NS）和接收邻居宣告（NA）消息，确认邻居是否可达。当ISATAP接口提供了多播仿真机制时，请求消息的目的地址就是目标节点被请求节点多播地址。否则，目的地址被填写为目标节点的单播地址。

另外，主机还应该执行邻居不可达检测（NUD）。

路由器可以执行上面描述的可达性检测和NUD过程，但这可能并不适用于所有环境。

所有ISATAP节点都必须发送被请求的邻居宣告消息。

##### 4.2.7.5.2 重复地址检测

ISATAP地址不需要进行重复地址检测，因为假设IPv4地址是不重复的。

##### 4.2.7.5.3 路由器和前缀发现

下面描述了ISATAP链路上支持路由器和前缀发现的机制。

###### 4.2.7.5.3.1 数据结构

ISATAP节点除了采用邻居发现中的数据结构前缀列表和默认路由器列表外，ISATAP链路还增加了一个新的数据结构“潜在路由器列表”（PRL）和新的配置变量。

PrlRefreshInterval：

初始化之后连续两次PRL重新刷新的时间间隔，以s为单位。应该不少于3600s。

默认：3600s。

PRL与每条ISATAP链路相关。在PRL中，每个表项（“PRL（i）”）包含了一个ISATAP接口的IPv4地址（“V4ADDR（i）”）和一个相关的定时器（“TIMER（i）”）。下面描述了PRL的初始化和刷新处理。

当一个节点使能了ISATAP链路时，它用IPv4地址初始化PRL。这个IPv4地址可能通过下面方法获得：ISATAP的DHCPv4选项（选项编码TBD）、手工配置或一种未指明的备用方案（如DHCPv4的vendor-specific选项）。

在没有其他机制可用时，可以采用通过带外方法（如DHCPv4、静态配置等）建立的DNS FQDN（fully-qualified domain name）。FQDN通过以下方法为PRL确定IPv4地址：静态主机文件、站点的名字服务、在站点内查询DNS服务器或者一种未指明的备用方案。FQDN选择没有强制的规则，但是手工配置必须支持。当采用DNS时，客户端必须用IPv4通信。

初始化之后，每隔PrlRefreshInterval时间，节点周期性地刷新PRL（也就是用上面描述的一种或多种方案）。

#### 4.2.7.5.3.2 路由器通告消息的有效性检查

必须执行IPv6邻居发现的RA消息正常有效性检查。

另外，接收的RA消息包括了前缀信息选项和/或在当前跳数限制、路由器生存时间、可达时间、重传时间域中的非0的编码值，RA消息必须满足下面的ISATAP有效性检测。

网络层（IPv6）源地址是一个ISATAP地址，并且这个地址是某个PRL（I）的内嵌V4ADDR（i）。

#### 4.2.7.5.3.3 路由器规格

除了正常的IPv6路由器所必须具有的规格要求外，当请求数据包的源地址不是一个未指定地址时，ISATAP接口应该响应单播RA消息给这个请求主机。

### 4.2.7.5.4 主机规格

当主机收到的一个未请求的RA消息，消息中包含前缀信息选项和/或者路由器生存时间值非零时，主机可以发送RS消息。本节主要描述RS消息发送和RA消息的处理流程。

（1）发送路由器请求

假设所有的PRL（i）项都表示站点内激活的ISATAP接口，也就是，这是PRL的信任基础，不需要可达性检测。主机根据一个或更多的PRL（i）周期性地发送RS消息。

主机增加了下面变量来支持请求处理：

MinRouterSolicitInterval

相同的 ISATAP 接口连续请求的最小时间间隔。应该小于 900s。

默认：900s。

RS消息采用接口的链路本地地址作为源地址。当ISATAP接口提供多播仿真机制时，RS消息被发送给所有路由器多播地址。否则，RS消息被发送到的某个PRL（i）的链路本地ISATAP地址，这个地址由PRL（i）的V4ADDR（i）构成。RS消息的后续发送处理与正常IPv6的RS发送流程没有区别。

（2）处理路由器应答

RA消息的处理首先应该遵从IPv6正常的RA处理流程。另外，如果RA消息的源地址是为某个PRL（i）的内嵌V4ADDR（i）的ISATAP地址时，主机重新启动定时器TIMER（i）。设置“MIN_LIFETIME”为路由器生存时间的最小值或在RA消息中生存时间选项值。则TIMER（i）被设置为：

MAX（（0.5 * MIN_LIFETIME），MinRouterSolicitInterval）

#### 4.2.7.6 部署考虑

##### 4.2.7.6.1 主机和路由器部署考虑

对于主机，如果一个底层链路同时支持IPv4（其上实现了ISATAP）和IPv6，若IPv6层没有收到路由器宣告消息（比如，没有直接连接到IPv6路由器），ISATAP功能可能被启动。当配置了一个非本地链路地址，且在链路上获得一个默认路由器之后，主机要停止ISATAP路由器请求过程，允许ISATAP地址配置过期，为该主机添加到DNS的ISATAP地址记录也将被删除。通过这种方法，随着站点内IPv6路由器的增加，ISATAP使用范围将逐步变小。

路由器在同一物理链路上同时配置普通的IPv6接口和ISATAP接口。两个域之间的路由按照原有方式进行。注意：ISATAP接口和普通IPv6接口的前缀是截然不同的。在路由器的ISATAP接口上配置的IPv4地址应该人工或自动添加到站点的地址记录中。

##### 4.2.7.6.2 站点内管理考虑

- ISATAP 链路由一系列公告的 ISATAP 接口，以及发现这些接口地址的一系列节点所组成，因此ISATAP 链路是一个管理上（非物理）的概念。
- 主机和路由器采用特殊的方式进行 ISATAP 部署。特别地，主机部署时可以不知道已存在的路由器，或只知道少量已存在路由器，路由器的部署不需要主机重新配置。
- 站点管理员维护一批代表公告 ISATAP 接口的 IPv4 地址，节点能够通过前面描述的机制访问这个地址列表。ISATAP 节点利用该列表初始化 PRL，并周期刷新。可靠的站点管理能减少控制信息量，系统管理员应该保证站点的 PRL 信息被很好地维护。

#### 4.2.7.7 安全考虑

ISATAP站点边界路由器和防火墙必须执行IPv6入口过滤功能，不能向站点外转发源地址和/或目的地址是站点本地地址的数据包。

另外，对IPv6和IPv4的攻击也必须被考虑。特别地，边界路由器和防火墙必须执行IPv4入口过滤和协议号为41的数据包过滤。

为了防止发自ISATAP站点内的源IPv6地址欺诈攻击，路由器在ISATAP接口上应启用缓解欺诈攻击的安全机制。至少，ISATAP站点的边界网关必须记录欺诈源地址的来源。

IPv6邻居发现信任模型也适用于ISATAP。但是启用了安全机制的企业网的危险实际上是很小的。

ISATAP 地址不支持无状态自动配置的私有扩展。

ISATAP（the Intra-site Automatic Tunnel Addressing Protocol，站内自动隧道寻址协议）用来将IPv4站点内的双栈节点通过自动隧道接入到IPv6路由器，它允许那些和IPv6路由器不共享同一物理链路的双栈节点通过IPv4自动隧道将数据包送达IPv6下一跳。从这点上看，站点的IPv4体系结构被当作是一个NBMA（Non-Broadcast Multiple Access link layer）链路层。

ISATAP过渡机制采用一个内嵌IPv4地址的IPv6地址，这样一来，不管站点采用的是全球或是私有的IPv4地址，都可以在站点内采用IPv6-in-IPv4的自动隧道技术。ISATAP地址格式既可以采用站点单播IPv6

地址前缀也可以采用全局单播IPv6地址前缀，这样就能够支持站点和全局的IPv6路由。

#### 4.2.7.8 适应性声明

ISATAP具有以下特征：

- 利用自动 IPv6-in-IPv4 隧道技术，将站点的 IPv4 体系结构视为 IPv6 的一个 NBMA 链路层。
- 在边界网关上不影响聚合范围的情况下，能够在 IPv4 站点内部署新的 IPv6 主机。
- 不需要站点提供特殊的 IPv4 服务（例如，多播等）。
- 支持无状态地址自动配置和手工配置两种方式。
- 支持使用非全局惟一 IPv4 地址的网络（如使用私有地址的时候）。
- 和其他过渡机制相互兼容（如 6to4 机制）。

### 4.2.8 BGP 隧道

#### 4.2.8.1 概述

BGP 隧道实现 IPv6 岛屿互联的方式尤其适合于已经开展了 BGF/MPLS VPN 业务的运营商。这种过渡方式可以使运营商暂时不必将现有核心网络升级为 IPv6 网络就可以实现对外提供 IPv6 业务。

PE 与 CE 之间必须有纯 IPv6 连接，这个连接可以是物理连接也可以是逻辑连接。在 CE 与 PE 之间必须有 IPv6 路由通路，可以通过域间路由协议交换 IPv6 路由可达信息实现，也可以通过 PE 和 CE 上配置的静态/缺省路由信息进行路由控制。

其中，利用 MPLS LSP 路径传送 IPv6 数据包不仅可以解决 IPv6 网络互联，同时也有利于实施移动 IP 业务。图 16 所示为利用 MPLS 机制传输 IPv6 数据包的一种实现方案，该方案需要 PE 支持 IPv6，称为 6PE 方案。6PE 通过 MPLS 与 MP-BGP+的配合实现控制平面与数据平面分离，IPv6 的可达信息与标签分配通过 MP-BGP+传送，而 IPv6 数据流通过 MPLS LSP 传送。运行在运营商网络边界的 PE 路由器需要具备双栈协议。具体 6PE 方案不在本标准中具体规定。



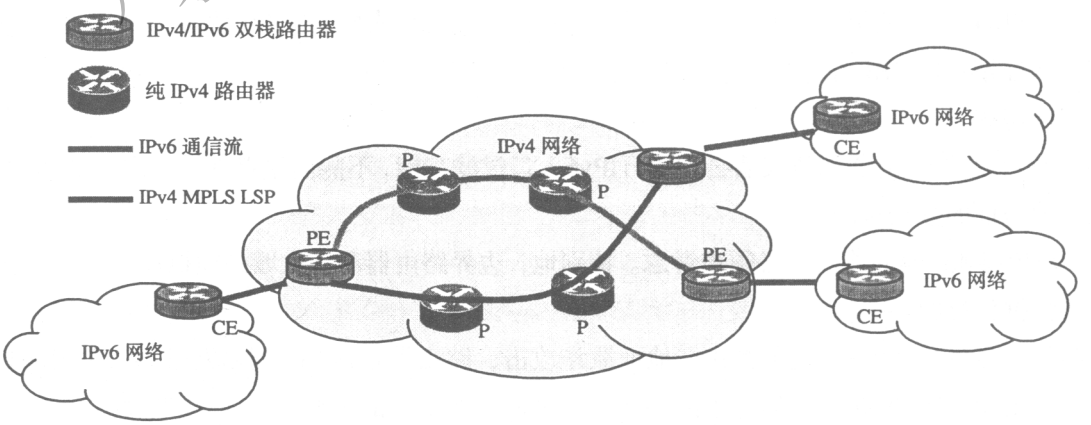



图 16 PE-to-PE MPLS 隧道方式

#### 4.2.8.2 工作原理

本节中，IPv6 网络的边缘路由器即 CE，IPv4 网络中与 IPv6 网络直接相连接的路由器称为 PE。

IPv6 站点必须通过 CE 连接到一个或多个运行 MP-BGP 的双栈 PE（DS-BGP-R）上。DS-BGP 必须至少配置一个 IPv4 地址和一个 IPv6 地址。IPv4 地址用于连接 IPv4 网络，IPv6 地址用于连接 IPv6 网络。

接收从 IPv6 网络发送的 IPv6 数据包的 DS-BGP 设备，称之为入口双栈 BGP 设备，即 I-DS-BGP-R；向 IPv6 网络中发送 IPv6 数据包的 DS-BGP-R，称之为出口双栈路由器，即 E-DS-BGP-R。

跨越 IPv4 网络互联 IPv6 网络步骤如下：

（1）通过 DS-BGP-R 交换 IPv6 可达信息：

（1.a）DS-BGP-R 通过 MP-BGP 与对端的 DS-BGP-R 交换 IPv6 的路由可达信息。

（1.b）为此，E-DS-BGP-R 需要宣称其为 BGP 下一跳节点。

（2）I-DS-BGP-R 与 E-DS-BGP-R 之间通过隧道传送 IPv6 数据包。

有两种方法可以实现通过 BGP 互联跨越 IPv4 网络的 IPv6 网络，分别是：

- MP-BGP OVER IPv4
- MP-BGP OVER IPv6

两种方法在 1.a）的实现上是相同的，但是在 1.b）与（2）的实现上是有差别的。

#### 4.2.8.2.1 MP-BGP OVER IPv4

MP-BGP over IPv4 必须在 IPv4 的协议栈上运行 MP-BGP（MP-BGP/TCP/IPv4），即 DS-BGP-R 采用 IPv4 地址作为 BGP 下一跳地址，并将此信息传送给隧道对端。MP-BGP 要求下一跳的地址族与 NLRI 地址族相同，因此 IPv4 地址信息必须嵌入 IPv6 地址形式中，这里必须采用 IPv4 映射地址格式。这种地址格式也可以使 DS-BGP 自动地建立隧道。封装可以采用 IPv4、MPLS 或者 GRE 形式。

I-DS-BGP-R 必须建立通向 E-DS-BGP-R 的隧道；隧道的目的地址根据 BGP 信息中相应的下一跳域中的 IPv4 映射地址获得。

#### 4.2.8.2.2 MP-BGP over IPv6

MP-BGP over IPv6 必须在 IPv6 协议栈上运行 MP-BGP（MP-BGP/TCP/IPv6），即 DS-BGP-R 采用 IPv6 地址作为 BGP 下一跳地址，并将此信息传送给隧道对端。MP－BGP 消息以及 IPv6 数据包跨越 IPv4 网络在 IPv6 网络之间传递必须依赖于现有的各种隧道过渡技术，如 6to4、ISATAP 等。

I-DS-BGP-R 必须建立通向 E-DS-BGP-R 的隧道，采用的隧道技术与 BGP 信息中相应的下一跳域中的 IPv6 地址相一致。

#### 4.2.8.2.3 两种方法的共同点

两种方法在以下方面相同：

- MP-BGP 的 AFI 必须等于 2，即表示 IPv6 地址族；SAFI 的取值与具体的隧道技术有关。
- 如果 PE 数量不多，可以采用 FULL-MESH 方式；否则可以采用路由反射的方法。
- IPv6 网络中的主机可以采用纯 IPv6 地址，不需要特殊的 IPv6 地址。

### 4.2.8.3 隧道的方式

#### 4.2.8.3.1 MP-BGP over IPv4

在 MP-BGP over IPv4 的方法中，可以选用 IPv4、MPLS 或者 GRE 隧道方式；具体的隧道技术请参见相关文档。

（1）IPv4 或者 GRE 隧道方式

当采用 IPv4 或者 GRE 隧道时，MP-BGP 中 SAFI 的取值必须在此范围内：单播（1），组播（2），两者共同（2）。

I-DS-BGP-R 必须采用 BGP 下一跳中含有的 IPv4 地址作为隧道头的目的地址，同时采用自身的一个 IPv4 地址作为隧道头的源地址。

（2）MPLS LSP 隧道方式

如果 IPv4 骨干网络支持 MPLS，可以采用 MPLS LSP 作为隧道技术。LSP 的建立可以采用已有的各种协议，包括 LDP、RSVP；具体过程参见相关文档。

如果 MP-BGP over IPv4 方法采用了 MPLS LSP 作为隧道实现方式，I-DS-BGP-R 可以直接根据 IPv6 地址进行打标签，并不需要先抽出 IPv4 地址然后根据 IPv4 地址来打标签。出口标签指向从 I-DS-BGP-R 到 E-DS-BGP-R 的 LSP。只采用单级标签可以完成 MP-BGP over IPv4 中的隧道操作，但是也可以选择采用二级隧道。

只采用单级标签时，MP-BGP 不通告标签信息，SAFI 取值必须为单播，组播或者两者都有（1，2，3）。

采用双级标签时，MP-BGP 通告标签信息，SAFI 必须为标签 SAFI（4）或者 VPN（128），具体应该是哪一个与标签分配过程有关。参考相关文档。

#### 4.2.8.3.2 MP-BGP over IPv6

MP-BGP over IPv6 依赖于现有的 IPv6 网络互联的各种隧道方式，隧道终点 IPv4 地址的提取方法与具体的隧道技术有关。

SAFI 取值必须为其中之一：单播、组播或者两者都有（1，2，3）。

### 4.2.8.4 安全考虑

此技术可以采用 BGP 中的安全特性以及 ISP 域中的各种安全策略，不会引入新的安全问题。

## 4.3 翻译转换机制

### 4.3.1 SIIT

#### 4.3.1.1 简介

##### 4.3.1.1.1 概述

这个算法适用于在过渡初期纯 IPv6 节点或者没有配置 IPv4 地址的 IPv4/IPv6 双栈节点，即 IPv4 资质节点。需要与纯 IPv4 节点通信的情况。该算法仅描述为完成 IPv6 节点与 IPv4 节点通信的完整解决方案中的一项机制，不涉及临时 IPv4 地址获取与路由等。

图 17 显示了无状态 IP/ICMP 翻译算法（SIIT）应用的情况：

（1）小规模网络中（如一个简单的子网）

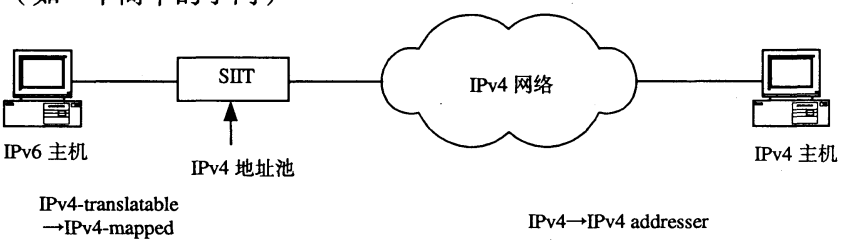



**图 17 在单一 IPv6 子网中的 SIIT 应用**

（2）包含纯 IPv6 的节点的双栈 IPv4/IPv6 网络站点。在 IPv6 与 IPv4 共存的网络中采用 SIIT 机制如图 18 所示。

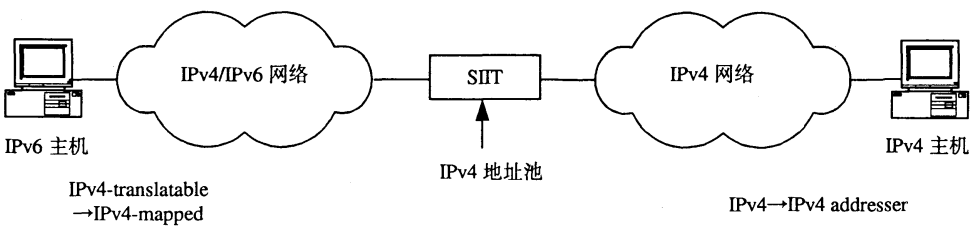



**图 18 在 IPv6 与 IPv4 共存的网络中采用 SIIT 机制**

上述两种应用情况直接对应的是 IPv6 主机的数量，因此间接地可以对应到过渡初期的时间轴上，那么应用一早于应用二。

为了保证数据包都能被翻译，在每条用来路由“翻译包”进出网络拓扑的路径上都必须有翻译器。这并不要求在该网络和互联网之间的每个物理连接点都要有翻译器，因为路由可以将这些数据包传送到翻译器。

通过翻译器与IPv4节点通信的纯IPv6节点采用IPv4映射地址形式来标识对端IPv4节点，而采用IPv4翻译地址作为本地通信的地址。当纯IPv6的节点发送数据包到IPv4节点时，翻译器根据其IPv4映射地址的目的地址形式作出判断需要进行翻译。当IPv4节点发送数据包到达翻译器时，翻译器会把它们翻译成有IPv4翻译地址作为目的地址。当数据包需要进行翻译器通信时，翻译器会将收到的IPv4/IPv6数据包翻译成IPv6/IPv4数据包。

#### 4.3.1.1.2 技术前提

（1）采用翻译器与纯IPv4节点通信的IPv6节点必须有一个IPv4翻译地址形式的IPv6地址，也就是有一个IPv4地址池可以用来产生IPv4翻译地址。

（2）不管数据包是在翻译器的“内部”还是“外部”产生的，只要其目标地址是这个翻译器地址池里地址的IPv4数据包，那么都应该找到正确路由，即地址池内的地址必须和翻译器IPv4侧的地址分属不同网段。

（3）没有UDP校验和的分段IPv4 UDP数据包基本上不属于正常包，故翻译器对这类数据包不进行翻译。

（4）对于应用程序来说，如果是双栈上的应用程序不需要任何改动，即可以采用SIIT机制；对于纯IPv6节点上的应用，需要改动应用程序使其可以识别通信对端的类型（IPv4还是IPv6节点）等，具有双栈通信的能力。



### 4.3.1.2 从IPv4到IPv6的翻译

当IPv4-to-IPv6翻译器收到了一个IPv4数据包，如果其目的地址不属于翻译器所连IPv4网络，翻译器就将数据包的IPv4头翻译成IPv6头，然后根据其IPv6目的地址进行转发。原先数据包中的IPv4头被IPv6头取代。除ICMP包之外，这种数据包的传输层头和数据部分都保持不变。

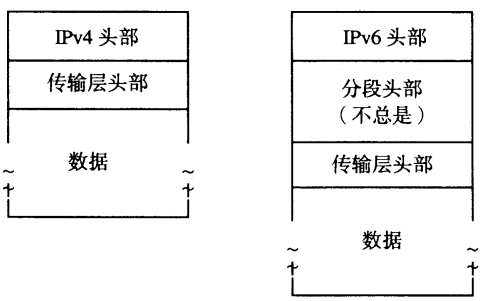



图19 IPv4到IPv6翻译

IPv4和IPv6的区别之一是在IPv6中路径MTU发现是必须的，但在IPv4中这是可选的，这意味着IPv6路由器不会分解数据包，只有发送方才会做分段。

如果IPv4节点做路径MTU发现时（这通过对包头DF位进行设置可以实现），路径MTU发现就可以进行端到端操作，也就是要通过翻译器。在这种情况下，IPv4和IPv6路由器都会发送ICMP包“数据包过大”消息回给发送方。当IPv6路由器发送的ICMP错误消息将通过翻译器时，翻译器就会将这些ICMP错误消息翻译成IPv4发送方可以理解的形式。在这种情况下，只有到达翻译器的IPv4数据包已经分段，对应的翻译数据包才会有IPv6 分段头。

如果 IPv4 发送方不进行路径 MTU 发现操作时，翻译器就不得不确保数据包不会超过 IPv6 侧的路径最大传输单元，由于 IPv6 保证了 1280 字节的数据包不需要分段，所以也可以通过分解 IPv4 数据包，让它匹配 IPv6 数据包的 1280 字节来实现。这也是说，当 IPv4 发送方不进行路径最大传输单元发现操作时，翻译器就必须一直带有 IPv6 分段头来表明发送方允许分段操作。

以上规则确保了数据包可以被发送方或者 IPv4 路由器分段，分段标志的低阶 16 位能自始至终地保留，从而确保数据包能被正确重组。除此之外，规则用 IPv6 分段头的存在来表明发送方可能没有要求路径最大传输单元发现，这也是说，当数据包以后被翻译回 IPv4 地址时，将不应设置 DF 标志位。

除了有专门的规则来处理分段和路径 MTU 发现操作，实际的翻译是由下面定义的简单映射组成。为了翻译 ICMP 误差消息的内容，ICMP 数据包要进行专门的处理，同时加上 ICMP 伪头校验和。

#### 4.3.1.2.1 IPv4 头翻译成 IPv6 头

如果没有设置 DF 标志，由 IPv4 产生的 IPv6 数据包可能会大于 1280 字节。因此在这种情况下，IPv4 数据包必须在翻译之前就被分段。由于分段后对应的翻译数据包中会有一个 8 字节的分段头，故 IPv4 净荷的长度不能超过 1232（1280 – 40 – 8）字节。分段后的结果再用下面介绍的方法进行翻译。

如果设置了 DF 位，数据包就不是分段包（也就是说，MF 标志位没有设置，分段的偏移 Offset 是 0），这种情况也就没必要再加分段头到数据包中，IPv6 头域设置如下：

版本：

6

业务流分类：

缺省情况下，直接拷贝 TOS 与优先级域中的值过来（所有的 8 位都拷过来）。根据[DIFFSERV]，各比特的定义在 IPv4 和 IPv6 中是一样的。但是，在某些 IPv4 环境里，这些位可以和老的语意“业务和优先类别”一起使用。这时，翻译器应该具有忽略 IPv4“TOS”的能力，并且总是设置 IPv6 业务流分类为 0。

流标志：

0（所有位都是 0）

载荷长度：

IPv4 头中的总长度减去 IPv4 头和 IPv4 选项长度。

头下一个头：

IPv4 头中协议域的值。

跳数限制：

从 IPv4 头中拷贝过来的 TTL 值，因为翻译器是路由器，作为路由器转发数据包功能的一部分，需要递减 IPv6 跳数限制值同时检查结果是否为 0。如果结果为 0，那么就需要并发送 ICMPv4“ttl exceeded”错误消息。

源地址：

低阶 32 位是 IPv4 源地址，高阶 96 位是 IPv4 映射前缀（::ffff:0:0/96）

目的地址：

低阶 32 位是 IPv4 目的地址，高 96 位是 IPv4 翻译前缀（0::ffff:0:0:0/96）

IPv4 数据包里的 IPv4 选项一般都要忽略掉，也就是说不翻译这些选项。无论如何，如果有一个没有过期的源路由选项，那么数据包就必须被扔掉，同时发送 ICMPv4 错误消息“目的地不可达/源路由失败”

（Type 3/Code 5）给发送方。

如果有必要加一个分段头（没有设置 DF 位或者数据包就是分段包），数据包头域设置和上面的一致，除了以下特殊情况。

（1）在 IPv6 头中

载荷长度：IPv4 头里的总长值，加上分段头的 8 字节，减去 IPv4 头和 IPv4 选项的长度。

头下一个头：分段头（44）。

（2）在分段头中

头下一个头：从 IPv4 头拷贝过来的协议域。

分段偏移：从 IPv4 头拷贝过来的分段偏移。

标志 M：从 IPv4 头拷贝过来的更多分段域。

标识：低阶 16 位从 IPv4 头里的标识位直接拷贝过来，高阶 16 位都设置为 0。

#### 4.3.1.2.2 翻译 IPv4 上的 UDP

如果 UDP 数据包有零 UDP 校验和，为了翻译这个数据包就必须计算有效的校验和。无状态翻译器不能给分段的数据包做这些，但是[MILLER]指出具有 0 校验和分段的 UDP 数据包仅用于恶意的目的，因此并不认为这是一个限制。

当翻译器收到了分段 UDP IPv4 数据包的第一个分段部分，并且校验和位是零，翻译器应该扔掉数据包并产生一个系统管理事件，来记录数据包里的 IP 地址和端口号。当翻译器收到的分段不是第一个分段，它应该只扔掉数据包，因为没必要再记录它的端口号。

当翻译器受到了一个没有被分段的 UDP IPv4 数据包，并且其校验和位是零，翻译器在翻译时必须计算丢失的 UDP 校验和。同时，翻译器也应该用一个计数器来统计通过这种方式产生了多少个 UDP 校验和。



#### 4.3.1.2.3 ICMPv4 头翻译成 ICMPv6 头

所有需要翻译的 ICMP 消息在翻译的时候需要更新 ICMP 校验和位，因为 ICMPv6 不同于 ICMPv4，它有一个类似 UDP、TCP 的伪头校验和。

所有的 ICMP 数据包都需要翻译 Type 值，此外，对于 ICMP 错误消息来说，其中包含的 IP 头也要翻译。

必须翻译的各类 ICMPv4 消息如下。

（1）ICMPv4 查询消息

Echo 和 Echo 应答（Type 8 和 Type 0）

分别调节 type 为 128 和 129，更新 ICMP 校验和。

信息请求/应答（Type 15 和 Type 16）

在 ICMPv4 中已经过时了，扔掉。

Timestamp 和 Timestamp 应答（Type 13 和 Type 14）

在 ICMPv6 中已经过时了，扔掉。

地址掩码请求/应答（Type 17 和 Type 18）

在 ICMPv6 中已经过时了，扔掉。

ICMP 路由广播（Type 9）

单跳消息，扔掉。

ICMP 路由请求（Type 10）

单跳消息，扔掉。

未知的 ICMPv4 类型

扔掉。

（2）IGMP 消息

当 MLD 消息[MLD]是针对 IPv4 IGMP 消息的逻辑 IPv6 的补充，由于所有“正常”的 IGMP 消息都是单跳消息，翻译器应该只扔掉它们。其他的 IGMP 消息可能用于组播路由协议。因为试图让路由器毗邻关系穿过 IPv4/IPv6 翻译器，将产生一个配置错误，这些数据包也将被扔掉。

（3）ICMPv4 错误消息

目的地不可达（type 3）

对所有那些没有在下面列出的都设置 Type 为 1。

（4）Code 位翻译如下

Code 0，1（网络，主机不可达）：

设置 Code 为 0（没有路由到目的地）。

Code 2（协议不可达）：

翻译成 ICMPv6 参数问题（Type 4，Code 1），并且让指针指向下一个头域。

Code 3（端口不可达）：

设置 Code 为 4（端口不可达）。

Code 4（需要分段和设置 DF 位）：

设置 type 2，code 0 翻译成一个 ICMPv6 数据包过大的消息（Type 2）。由于 IPv4 和 IPv6 数据包头长度不同，所以需要调节 MTU 域。如果 IPv4 路由器没有设置 MTU 域，也就是说，路由器没有实现[PMTUv4]，则翻译器必须采用在[PMTUv4]指定的 plateau 优先值来决定一个可能的路径 MTU，并且在 ICMPv6 数据包里包含进路径 MTU（使用最大的优先 plateau 值，其小于返回的总长域值）。

Code 5（源路由失败）：

设置 Code 为 0（没有路由到目的地）。由于源路由不会被翻译，所以这个错误一般是不会发生的。

Code 6，7：

设置 Code 为 0（没有路由到目的地）。

Code 8：

设置 Code 为 0（没有路由到目的地）。

Code 9，10（和目标主机通信被禁止）：

设置 Code 为 1（和目标主机通信被禁止）

Code 11，12：

设置 Code 为 0（没有路由到目的地）。

重定向（Type 5）

单跳消息，只扔掉。

Source Quench（Type 4）

在 ICMPv6 中已经过时。只扔掉。

超时（Type 11）

设置 Type 位为 3。Code 位保持不变。

参数问题（Type 12）

设置 Type 位为 4。需要更新指针让其指向在翻译后内 IP 头。

#### 4.3.1.2.4 ICMPv4 错误消息翻译成 ICMPv6

以上已经介绍了 IPv4 和 IPv6 ICMP 错误消息格式的一些不同。除此之外，错误数据包中的 ICMP 错误消息所包含 IP 头需要像正常 IP 头一样需要翻译。翻译“包出错”很可能会改变数据包的长度，因此在外部 IPv6 头中的载荷长度域可能需要更新。内部 IP 头的翻译能要递归调用那些翻译外部 IP 头的功能函数。

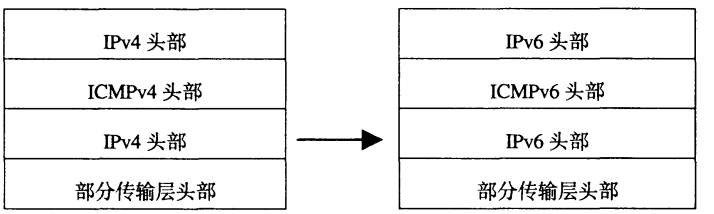



图 20 从 IPv4 到 IPv6 的 ICMP 错误消息翻译

#### 4.3.1.2.5 翻译判断

翻译器必须知道用来表示内部纯 IPv6 节点的 IPv4 地址池。因此当翻译器发现 IPv4 目的地址正好属于该地址池时，那么就会对这个数据包进行翻译。

### 4.3.1.3 从 IPv6 到 IPv4 的翻译

当一个 IPv6-to-IPv4 翻译器收到一个 IPv6 数据包，其目的地址是一个 IPv4 映射地址时，它就把数据包的 IPv6 头翻译成 IPv4 头，然后根据 IPv4 目的地址进行转发。原来数据包里的 IPv6 头被丢掉，用 IPv4 头取而代之。除了 ICMP 数据包以外，传输层头和数据包的数据保持不变。



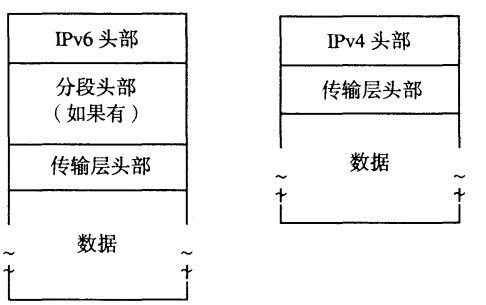



图 21 IPv6 到 IPv4 翻译

IPv6 和 IPv4 在分段和最小链路 MTU 上有些不同，这些都会影响翻译。IPv6 链路必须有 1280 字节的 MTU 或者比这更大。相对应的 IPv4 限制是 68 字节。因此，如果路径上有一个 IPv6 到 IPv4 的翻译器，若没有特殊的办法，是不可能做端到端的路径 MTU 发现，这是因为 IPv6 节点很可能收到一个由 IPv4 路由器送来的 ICMP“包太大”错误消息，报告 MTU 小于 1280。无论如何，[IPv6]要求 IPv6 节点通过减少路径 MTU 到 1280，并在每个数据包里带 IPv6 分段头，来处理这种“数据包过大”的 ICMP 错误消息。这就可以实现只要路径 MTU 是 1280 字节或者更大，端到端路径 MTU 发现就可以穿过翻译器。当路径 MTU 下降到低于 1280 的限制时，IPv6 发送方将生成 1280 字节的数据包，翻译成 IPv4 以后，IPv4 路由器在对它进行分段。

这个方法仅有的缺点是不能在发送方用 PMTU 来对 UDP 分段进行优化（因为完全反对避免分段）。因为 IPv6 分段头存在就意味着询问“可以在 IPv4 一侧做数据包的分段吗?”。因此，如果 UDP 应用想发

送独立于 PMTU 之外的大数据包，发送方就只能够决定翻译器 IPv6 一侧的路径 MTU。如果路由器的 IPv4 一侧的路径 MTU 比较小，则 IPv6 发送方将收不到“包过大”的 ICMP 错误消息，因此也无法调节它正在发送的数据包分段的尺寸。

除了专门用于处理分段和路经 MTU 发现的规则以外，实际的头翻译由下面定义的小映射组成。为了翻译 ICMP 错误消息内容和 ICMP 伪头校验和，ICMP 数据包的翻译要进行专门处理。

#### 4.3.1.3.1 IPv6 头翻译成 IPv4 头

如果没有 IPv6 分段头，则 IPv4 头域设置如下：

版本：

4

互联网头长度：

5（没有 IPv4 选项）

TOS 和优先级：

缺省情况下，直接从 IPv6 流量类别（所有的 8 个比特）拷贝来的。在 IPv4 和 IPv6 中，各比特的定义是一样的。翻译器应该具有忽略 IPv6 流量类别和设置 IPv4“TOS”到 0 的能力。

总长：

IPv6 头里面的载荷长度值加上 IPv4 头的长度。

标识：

全零。

标志：

更多的分段标志位设置为零。不分段标志位设置为 1。

分段偏移：

全零。

生存时间：

从 IPv6 头直接拷贝来的跳限制值。因为翻译器就是路由器，作为转发数据包功能的一部分，需要将 IPv4 TTL 减一，然后检查是否结果为 0。如果结果为 0，那么需要发送 ICMPv6“ttl 过期”错误。

协议：

从 IPv6 头里面直接拷贝来的下一个头域。

头校验和：

IPv4 头建立起来后，经过计算得出。

源地址：

如果 IPv6 源地址是一个 IPv4 翻译地址，则把 IPv6 源地址的低阶 32 位当成 IPv4 源地址。否则把源地址设置成 0.0.0.0，用 0.0.0.0 可以避免被扔掉。

目的地址：

被翻译的 IPv6 数据包有一个 IPv4 映射目的地址。因此将 IPv6 目的地址的低阶 32 位直接拷贝到 IPv4 目的地址。

如果 IPv6 数据包中有 IPv6 逐跳选项头、目的地选项头或者具有剩余段域等于零的路由头出现，翻译器对之不进行翻译。因此，当忽略这些域时，总长域和协议域都将作相应的调整。

如果一个带非 0 剩余段域的路由头出现，该数据包是不能被翻译的，同时发送一个 ICMPv6“碰到一

个参数问题/错误的头域”（Type 4/Code 0）的错误消息返回发送方，其指针域标明剩余段域的第一个字节。

如果 IPv6 数据包含一个分段头，各域设置如上面所述，除了以下例外：

总长：

从 IPv6 头里得到的载荷长度值，减去分段头的 8 个字节，再加上 IPv4 头的长度。

标识：

从分段头里面标识位的低阶 16 位拷贝过来。

标志：

更多的分段标志从分段头的 M 标志拷贝过。不分段标志设置为 0 来允许 IPv4 路由器对这个数据包进行分段操作。

分段偏移：

在分段头里面的分段偏移域拷贝过来。

协议：

从分段头拷贝过来的下一个头值。

#### 4.3.1.3.2 ICMPv6 头翻译成 ICMPv4 头

作为翻译功能的一部分，所有的将被翻译的 ICMP 消息都要求更新 ICMP 校验和位。因为 ICMPv6 和 UDP、TCP 一样，有伪头校验和，这和 ICMPv4 不一样。

除此以外，所有的 ICMP 数据包需要翻译 Type 值，对于 ICMP 错误消息而言，所包含的 IP 头的也需要翻译。

各类需要翻译的 ICMPv6 消息如下。

（1）ICMPv6 信息类消息

Echo 请求和 Echo 应答（Type 128 和 129）

分别调节 type 为 0 和 8，同时更新 ICMP 校验和。

MLD 组播监听查询/报告/结束（Type 130，131，132）

单跳消息，扔掉。

邻居发现消息（Type 从 133 到 137）

单跳消息，只扔掉。

未知的信息消息

扔掉。

（2）ICMPv6 错误消息

目的不可达（type 1）

设置 Type 位为 3。翻译 code 位如下：

Code 0（没有到目的地路由）：

设置 Code 为 1（主机不可达）。

Code 1（和目的地通信被禁止）：

设置 Code 为 10（和目的地通信被禁止）。

Code 2（在源地址范围之上）：

设置 Code 为 1（主机不可达）。因为 IPv4 翻译源地址是全求地址，所以这个错误是几乎不可能发生的。

Code 3（地址不可达）：

设置 Code 为 1（主机不可达）。

Code 4（端口不可达）：

设置 Code 为 3（端口不可达）。

数据包过大（Type 2）

通过设置 code 为 4 把它翻译为 ICMPv4 目的地不可达错误信息。由于 IPv4 和 IPv6 包头尺寸的不同，考虑数据包错误是否包含一个分段头，来通过调节 MTU 域。

超时（Type 3）

设置 Type 为 11。Code 保持不变。

参数问题（Type 4）

如果 Code 为 1，翻译这个消息为 ICMPv4 协议不可达（Type 3，Code 2）。否则设置 Type 为 12 和 Code 为 0。在被翻译的包括 IP 头里更新指针让其指向对应的域。

未知错误消息

扔掉。

#### 4.3.1.3.3 ICMPv6 错误消息翻译成 ICMPv4

前面已经解释了 IPv4 和 IPv6 ICMP 错误消息格式的不同。除此之外，ICMP 错误消息包含的 IP 头的必须像普通 IP 头一样被翻译。翻译这个“错误的数据包”很可能会改变数据包的长度，因此外部的 IPv4 包头的总长度可能需要更新。

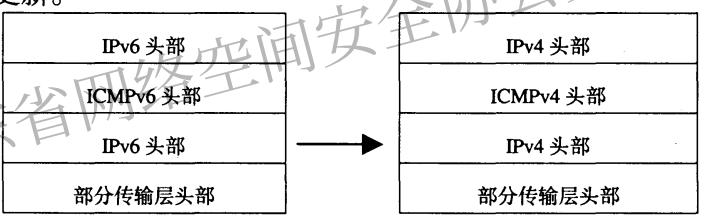



图 22 从 IPv6 到 IPv4 ICMP 错误报告消息的翻译

内部 IP 头的翻译可以通过递归调用翻译外部 IP 头的功能函数来完成。

#### 4.3.1.3.4 翻译判断

当翻译器接收到一个 IPv6 数据包，如果其目的地址是 IPv4 映射地址，翻译器就会将这个数据包翻译成 IPv4。

### 4.3.1.4 对 IPv6 节点的要求

（1）对于纯 IPv6 节点上的应用，需要改动应用程序使其可以识别通信对端的类型（IPv4 还是 IPv6 节点）等，具有双栈通信的能力。

（2）应该具有判断何时分配和更新 IPv4 翻译地址的能力。由于处理发生在 Socket API 下面，通常不需要涉及应用程序就可以实现。例如，当 Connect 或者 Sent to Socket 调用被激发时，需要检查目的地址是否是 IPv4 映射地址，如果是，则需要分配/刷新 IPv4 翻译地址。

（3）作为源地址选择机制的一部分，确保当目的地址是 IPv4 映射地址时，源地址必须是 IPv4 翻译的地址。IPv4 翻译地址不能和其他 IPv6 目的地址一起使用。

（4）如果通信对端有 AAAA DNS 记录，即使应用程序（或解析器）用这个 AAAA 记录进行通信失败了，也不可以再用通信对端的 A DNS 记录再次通信。这个局限性的原因是为了防止在两个 IPv6 节点（它们的 AAAA 记录在 DNS 里面）之间的通信意外的两次通过 SIIT 翻译器；先从 IPv6 到 IPv4，然后再次回

到 IPv6。最好发送一个失败的信号给应用程序。

#### 4.3.1.5 安全性考虑

无状态 IP/ICMP 翻译器不会引入新的安全问题。

由于认证头 [IPv6-AUTH]是涉及到 IPv4 标识域而翻译函数不能够总是保留这个标识域，故 IPv6 端点对于收到的经过 IPv4-to-IPv6 翻译的数据包不能计算 AH 值。因此 AH 不可以通过翻译器。

因为 ESP 不依赖 ESP 头前面的头域，带有 ESP 的数据包可以被翻译。ESP 传输模式比 ESP 隧道模式容易处理；为了采用 ESP 隧道模式，IPv6 节点在发送数据包的时候需要产生一个包含在 ESP 中的 IPv4 头；当接收到这样的数据包时，需要去掉包含在 ESP 内部的 IPv4 头。

### 4.3.2 NAT-PT

#### 4.3.2.1 NAT-PT 简介

SIIT[SIIT]描述了一种不需要为通信节点维护会话状态信息的协议翻译机制，该方法通过对 IPv6 和 IPv4 数据包的协议翻译，使得纯 IPv6 节点和纯 IPv4 节点能相互通信。SIIT 假定 IPv6 节点在和 IPv4 节点进行通信时分配了一个 IPv4 地址，但没有指明分配该 IPv4 地址的机制。

NAT-PT 过渡机制为 IPv6 节点规定了其在访问 IPv4 节点时的 IPv4 地址分配方法，当 IPv6 节点开始访问 IPv4 节点时，NAT-PT 从其 IPv4 地址缓冲池中为 IPv6 节点动态地分配一个 IPv4 地址。以上地址缓冲池中的地址是全局惟一的，而对于其为私有地址的情况不在本标准中考虑，有待于进一步的研究。NAT-PT 通过对 IPv6 和 IPv4 地址的翻译、绑定，在无需对节点做改动的情况下，为 IPv6 和 IPv4 域中节点的相互通信提供了透明的路由[NAT-TERM]。但是这要求 NAT-PT 能跟踪所建立的通信会话，并且属于该会话的数据包要通过同一个 NAT-PT 路由器。NAT-PT 对网络拓扑结构上的要求与 IPv4 中的 NAT 相同[NAT-TERM]。协议翻译的细节详见 SIIT[SIIT]，其在协议格式等方面的翻译是地址翻译机制功能的扩充。

通过结合 SIIT 协议翻译功能、NAT 的动态地址翻译能力以及相应的应用层网关（ALG），NAT-PT 为实现大量常用的应用程序能在纯 IPv6 节点和纯 IPv4 节点之间互通提供了完全的解决方案。

关于应用 NAT-PT 的一个基本假设是 IPv6 域和 IPv4 域之间没有其他的方法（如本地 IPv6 连接、隧道）来为 IPv6 和 IPv4 节点提供相互通信时，才采用 NAT-PT 过渡机制。也就是说 NAT-PT 过渡机制用于纯 IPv6 节点和纯 IPv4 节点之间的相互通信，应该避免在纯 IPv6 节点和一个双栈节点的 IPv4 部分之间采用协议翻译。

#### 4.3.2.2 传统 NAT-PT 的工作机制（IPv6 到 IPv4）

下面将描述传统 NAT-PT 的工作过程。

##### 4.3.2.2.1 基本 NAT-PT 的工作机制

图 23 用来描述 IPv6 域中的节点通过 NAT-PT 设备来访问 IPv4 域中的节点。其中：

节点 IPv6 – A 的 IPv6 地址为 FEDC:BA98::7654:3210；

节点 IPv6 – B 的 IPv6 地址为 FEDC:BA98::7654:3211；

节点 IPv4 – C 的 IPv4 地址为 132:146:243:30。

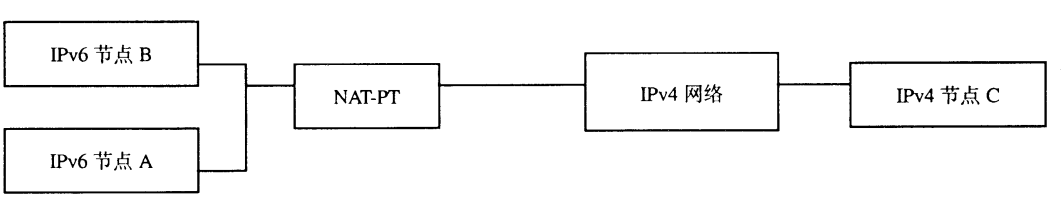



图 23 IPv6 节点访问 IPv4 节点

图 23 中的 NAT-PT 有一个 IPv4 地址缓冲池，缓冲池中 IPv4 的地址属于子网 120:130:26/24。

IPv4 地址缓冲池中的地址可以一对一的分配给 IPv6 节点。一一对应的情况要求缓冲池中的 IPv4 地址数等于或者多于 IPv6 域中的节点数。在本标准中，我们假定缓冲池中的 IPv4 地址数少于 IPv6 域中的节点数，所以对 IPv4 地址要进行动态分配。

假设 IPv6 – A 节点想和 IPv4 节点通信，节点 A 发出如下数据包：

源地址 SA= FEDC：BA98::7654:3210

目的地址 DA=PREFIX::132:146:243:30

注：PREFIX::/96 是 NAT-PT 在 IPv6 域中通告的网络前缀，目的地址带有该前缀的数据包将被路由到 NAT-PT 网关。预先配置好的前缀 PREFIX 只要在 IPv6 的末端区域内部是可路由的即可。

上述数据包路由到 NAT-PT，在那里被翻译成 IPv4 数据包。如果该数据包不是会话起始包，NAT-PT 应该已经保存了该会话的状态信息（如已分配的 IPv4 地址和其他一些翻译中要用到的参数）。若这些状态信息不存在，那么数据包将被丢弃。

如果该数据包是会话起始包，NAT-PT 从地址缓冲池中分配一个 IPv4 地址（如 120:130:26:10）给 IPv6 节点，并把数据包翻译成 IPv4 数据包。在会话期间，NAT-PT 保存翻译的参数和 IPv6 到 IPv4 的映射关系。

经过翻译形成的 IPv4 数据包具有源和目的地址分别为 SA=120:130:26:10，DA=132:146:243:30。任何从 C 节点回应的 IPv4 数据包会被 NAT-PT 通过保存的会话信息识别出是属于前面的同一个会话，并将其翻译成有如下参数的 IPv6 数据包：SA = PREFIX::132:146:243:30，DA=FEDC:BA98:7654:3210。这时翻译后的数据包在 IPv6 域内可以像正常的数据包那样传送。



#### 4.3.2.2.2 NAPT-PT 工作机制

NAPT-PT 通过网络地址、端口翻译和协议翻译，能使多个 IPv6 节点只采用地址缓冲池中的一个 IPv4 地址和外部的 IPv4 节点进行透明的通信。IPv6 节点的 TCP/UDP 端口号被翻译成已分配的 IPv4 地址的 TCP/UDP 端口号。

NAT-PT 支持 TCP、UDP 和其他端口复用的应用程序，但当缓冲池中的 IPv4 地址分配完时，NAT-PT 不能再为另外的 IPv6 节点提供对 IPv4 节点的访问。NAPT-PT 解决了 NAT-PT 存在的这个内在缺陷。通过对传输层端口的复用，NAPT-PT 能在一个 IPv4 地址上支持 63k 个 TCP 和 63k 个 UDP 连接。

对于如图 1 所示的例子，当用 NAPT-PT 代替 NAT-PT 时，所有的 IPv6 节点的 IPv6 地址可以翻译成一个 IPv4 地址 120:130:26:10。图 1 中的 IPv6 节点 A 与 IPv4 节点 C 建立了一个 TCP 连接。节点 A 发送了如下的数据包：

源地址 SA = FEDC：BA98:7654:3210

源 TCP 端口号 = 3017

目的地址 DA = PREFIX::132:146:243:30

目的 TCP 端口号 = 23

当数据包到达 NAPT-PT 设备时，NAPT-PT 为其在已经分配的 IPv4 地址上再分配一个 TCP 端口，并把 IPv6 数据包翻译成如下的 IPv4 数据包：

SA = 120:130:26:10

源 TCP 端口号 = 1025

DA=132:146:243:30

目的 TCP 端口 = 23

从地址为132:146:243:30,TCP端口号为23返回的数据包会被NAPT-PT辨识出,并被翻译成如下IPv6数据包:

SA = PREFIX::132:146:243:30

源TCP端口号 = 23

DA = FEDC:BA98:7654:3210

目的TCP端口号 = 3017

对于通过NAPT-PT进来的会话数据包，IPv6域中的服务器上通过对TCP/UDP端口的静态绑定只提供一种特定的服务。例如，图1中的节点A只能是IPv6域中的HTTP（端口号为80）服务器。IPv4域中的节点C发送如下数据包:

SA = 132:146:243:30

源TCP端口号 = 1025

DA = 120:130:26:10

目的TCP端口号 = 80

NAPT-PT把它翻译成如下IPv6数据包:

SA = PREFIX::132:146:243:30

源TCP端口号 = 1025

DA = FEDC:BA98:7654:3210

目的TCP端口号 = 80

在上述示例中，所有目的端口号为80的会话通信都转发到同一个节点A。



#### 4.3.2.3 DNS-ALG用于地址分配

当NAT-PT判断出有出去或进来的初始会话时，就从IPv4地址缓冲池中为IPv6节点分配一个地址。出去或进来的初始会话处理不同，但分配的IPv4地址属于同一个地址缓冲池。

在DNS中，IPv4域名和地址之间的映射关系保存为“A”记录。IPv6域名和地址之间的映射关系保存为“AAAA”或“A6”记录。本节中对DNS-ALG对“AAAA”或“A6”记录处理的规则是一样的。设计NAT-PT的目的是在IPv6节点和IPv4节点没有其他的相互通信手段的情况下提供它们之间的互通机制。在以下对NAT-PT的讨论中，假设IPv6域除有IPv4连接外，还可以有全局的IPv6连接或通过隧道与其他IPv6域相连。

##### 4.3.2.3.1 IPv4节点访问IPv6节点时IPv4地址分配

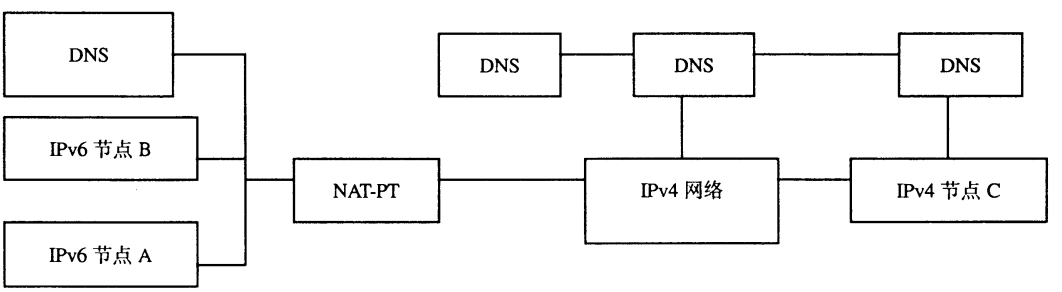



图24 IPv4节点访问IPv6节点

图24中节点A有一个IPv6地址:FEDC:BA98:7654:3210;节点B有一个IPv6地址:FEDC:BA98:7654:3211。节点C有一个IPv4地址：132:146:243:30。NAT-PT有一个IPv4地址缓冲池，缓充池中IPv4的地址属于子网：120:130:26/24。

当节点C发出对节点A进行域名解析的请求数据包时，解析请求被传送到IPv6域的DNS上。考虑NAT-PT在IPv4域和IPv6域的边界路由器上，请求数据包会穿越这个边界路由器。NAT-PT设备上的DNS-ALG会把进入IPv6域的A记录DNS请求数据包（DNS数据包的TCP/UDP源和目的端口号均为53）翻译如下：

对域名到地址解析的请求数据包，请求类型“A”改为“AAAA”或“A6”。

对地址到域名解析的请求数据包，用字符串“IP6.INT”代替“IN.ADDR.ARPA”。用IPv6地址代替字符串“IN-ADDR.ARPA”前的IPv4地址。

同时，当DNS应答数据包从IPv6域中的DNS服务器上返回给IPv4域中的请求节点时，DNS-ALG翻译DNS应答数据包的格式如下：

把DNS应答中的“AAAA”或“A6”记录翻译成“A”记录（当域名被完整地解析出时，只翻译“A6”记录）。

用NAT-PT从地址缓冲池中分配的IPv4地址代替IPv6域中DNS服务器解析出并返回的IPv6地址。

若IPv4还没有分配给IPv6节点，则在这时候从地址缓冲池中分配。以图24中的节点访问节点A为例，若C开始一个与A的会话时，它首先通过本地的DNS服务器查询节点A的“A”记录。本地DNS服务器把这个DNS请求转发给IPv6域中的DNS服务器。DNS-ALG截获并翻译该请求中的“A”记录为“AAAA”或“A6”记录，再将翻译后的DNS请求转发给IPv6域中的DNS服务器。节点A返回DNS应答如下：

节点A　　AAAA　　FEDC:BA98:7654:3210

DNS-ALG截获并翻译该数据包应答如下：

节点A　　A　　120.130.26.1

DNS-ALG在NAT-PT中保存FEDC：BA98::7654:3210和120.130.26.1之间的映射关系。包含“A”记录的DNS应答数据包返回给节点C。于是节点C发起会话如下：

SA=132.146.243.30

源TCP端口号 = 1025

DA=120.130.26.1

目的TCP端口号 = 80

该数据包被路由到保存有FEDC：BA98::7654：3210和120.130.26.1之间映射关系的NAT-PT，NAT-PT把数据包翻译成：

SA=PREFIX::132.146.243.30，源TCP端口号 = 1025

DA=FEDC:BA98::7654：3210，目的TCP端口号 = 80

这时数据包能在IPv6域内正确地传送。

所有经过NAT-PT DNS数据包的资源记录（RR）中的TTL值应设置为0，这样DNS服务器/客户端不缓存临时分配的RR。但是一些有缺陷的DNS客户端程序把TTL值设置为1，在某些情况下，这也能正常工作。

因为IPv4域中的一个节点可以同时对IPv6域中的多个节点进行地址解析并与之通信，导致DNS-ALG把NAT-PT地址缓冲池中的地址都耗完，从而阻塞了其他合法的通信，所以上述对进入IPv6域的会话地址分配方法易受到拒绝服务攻击。为降低拒绝服务攻击的可能性，对缓存在NAT-PT进入IPv6域会话的地址映射关系应该进行定时过期清除。同时，应该保证始终有一个IPv4地址可供从IPv6域发起的会话采

用（NAPT-PT），以避免如上所述的拒绝服务攻击。

#### 4.3.2.3.2 IPv6 节点访问 IPv4 节点时的 IPv4 地址分配

建议在 IPv6 域的 DNS 服务器在维护域内节点映射关系的同时也尽可能的缓存域外节点的域名与地址间的映射关系。在 IPv6 域中的 DNS 服务器缓存有域外 IPv4 节点映射关系的情况下，DNS 请求不会转发到 IPv6 域外，这可减少 DNS-ALG 的处理。同时，我们还建议 IPv6 域中的外部 DNS 服务器只缓存外部节点的域名和地址的映射关系，应尽量避免 DNS 解析时请求 – 应答数据包穿越 IPv6 和 IPv4 边界。

在采用 NAPT-PT 的情况下，当探测到一个新的连接请求时，NAPT-PT 从已分配的地址的 TCP/UDP 端口中分配一个新的端口给这个连接。同时，当 IPv6 节点和 IPv4 节点通信时要在 IPv4 地址前加一个特殊的前缀 PREFIX::/96。该前缀不用进行任何的配置。

我们可通过图 23 来描述另一个示例。假设节点 A 要访问节点 C，节点 A 通过 DNS 解析节点 C 的“AAAA”记录。因为节点 C 可能拥有 IPv6 和 IPv4 地址，NAT-PT 设备上的 DNS-ALG 将对 AAAA 记录解析的请求数据包转发给外部 DNS 服务器，同时生成一个对 C 节点的 A 记录解析的请求数据包并发送出去。如果外部 DNS 服务器上有 C 节点的 AAAA 记录，该记录将返回到 NAT-PT，在那里在原文转发给节点 A。若外部 DNS 服务器中还有 C 节点的 A 记录，则 A 记录返回 NAT-PT，DNS – ALG 将对该记录进行翻译，若 A 记录为：

节点 C　　A　　132.146.243.30

其被翻译成：

节点 C　　AAAA　　PREFIX::132.146.243.30　或

节点 C　　A6　　PREFIX::132.146.243.30

在 PREFIX 不改变的情况下，现在节点 A 可以像其他 IPv6 地址一样采用上面的 IPv6 地址，IPv6 域内的 DNS 服务器可以缓存上述记录。

因为这里没有双栈机制，所以存在一个 IPv6 域中的 DNS 服务器如何和 IPv4 域中的 DNS 服务器通信的问题。为解决这个问题，IPv4 域中的外部 DNS 服务器需要知道 NAT-PT 中地址缓冲池中的一个 IPv4 地址，NAT-PT 保存有该 IPv4 地址和 IPv6 地址域中的 DNS 服务器映射关系。同时 IPv6 域中的 DNS 服务器需要知道 IPv4 域中的外部 DNS 服务器不是 IPv6 节点，并且通过由 IPv4 域中的外部 DNS 服务器的 IPv4 地址与前缀 PREFIX::/96 形成的 IPv6 地址指向该外部 DNS 服务器。这种 DNS 服务器间能相互通信的机制具有良好的扩充性，很容易扩展到第二个 DNS 服务器。

#### 4.3.2.4 协议翻译细节

IPv6 和 ICMPv6 的头标和它们的 IPv4 版本非常相近，只是有一些项或者被省略，或者已经变成其他的意义，或者长度不同。要想实现端到端的 IPv4 和 IPv6 之间的通信，NAT-PT 必须能将全部的 IP/ICMP 头标从 IPv4 翻译到 IPv6，也能从 IPv6 翻译成 IPv4。另外，NAT-PT 还应该对上层的协议头标作相应的调整。本标准会有单独的 FTP-ALG 部分介绍当 FTP 的数据包从 IPv4 穿越到 IPv6 或从 IPv6 穿越到 IPv4 时 FTP-ALG 会对 FTP 载荷所做的变动。

协议翻译的细节已经在[SIIT]中已经有了详细的说明，但是因为 NAT-PT 同时还要做地址翻译，所以要对 SIIT 做一些变动。

#### 4.3.2.4.1 IPv4 头标翻译成 IPv6 头标

除了下面将提到的一些不同外，该翻译过程和 SIIT 中描述的完全一致。

源地址：低 32 位是 IPv4 源地址，高 96 位是为全部的 IPv4 通信指派的 PREFIX，带有该前缀的地址

将被路由到 NAT-PT 网关。（PREFIX::/96）

目的地址：NAT-PT 保留了 IPv4 目的地址和目的节点的 IPv6 地址的映射关系，IPv4 目的地址将被其映射的 IPv6 地址替换。

#### 4.3.2.4.2 IPv6 头标翻译成 IPv4 头标

除了下面将提到的一些不同外，该翻译过程和 SIIT 中描述的完全一致。

源地址：NAT-PT 保留了 IPv6 源地址到 IPv4 地址池中可用的 IPv4 地址映射关系，IPv6 的源地址根据保存的映射关系被相应的 IPv4 地址替换。

目的地址：被翻译的 IPv6 数据包的目的地址应该具有 PREFIX::IPv4/96 格式，只要将其低 32 位 IPv4 地址取出作为 IPv4 目的地址即可。

#### 4.3.2.4.3 TCP/UDP/ICMP 校验和的更新

以下描述了数据包通过 NAT-PT 时 TCP/UDP/ICMP 校验和在通过 NAT-PT 时的更新过程。

##### 4.3.2.4.3.1 TCP/UDP/ICMP 校验和从 IPv4 到 IPv6 的更新

为了能够反映出 IPv4 到 IPv6 的地址发生了变化，TCP 校验和与被设置为非零值的 UDP 校验和应该被重新计算，递增校验和调整算法可以沿用[NAT]中的使用。

在 NAPT-PT 情况下，TCP/UDP 的校验和也应该做些调整才可以反映出从 IPv4 到 IPv6 时引起的地址和 TCP/UDP 端口的变化。

当 IPv4 的 UDP 数据包中的校验和被设置为零时，NAT-PT 必须完整的为翻译后的 IPv6 数据包计算校验和。如果一个校验和为零的 IPv4 的 UDP 数据包以分段方式到达 NAT-PT，NAT-PT 必须等待所有的分段都到达后才将它们重装为单个分段的数据包，并在执行翻译到 IPv6 的 UDP 数据包的操作之前完成计算其校验和。



ICMPv6 和 ICMPv4 不同的是它采用了伪头标，所以在计算校验和的过程中与 UDP 和 TCP 相似。这就要求在计算 ICMPv6 校验和时，要对校验和作调整才能考虑到附加的伪头标。注意，由 ICMP 的载荷中所携带的源地址和目的地址的变化（在 NAPT-PT 的情况下还有 TCP/UDP/ICMP 标识的变化）也会引起对校验和的调整。

##### 4.3.2.4.3.2 TCP/UDP/ICMP 校验和从 IPv6 到 IPv4 时的更新

为了能够反映出地址从 IPv6 到 IPv4 的变换，TCP 和 UDP 的校验和应该被重新计算，使用的递增校验和调整算法可以沿用[NAT]的说明。在 NAPT-PT 情况下，应该调整 TCP/UDP 校验和使其能够顾及到从 IPv6 到 IPv4 地址时的地址和端口号变化。对于 UDP 数据包，校验和可以简化修改为零。

对 IPv4 的 ICMP 头标的校验和计算需要通过运行校验和调整算法[NAT]从 IPv6 的 ICMP 头标获得，运行该算法的目的是从计算中去除 IPv6 伪头标。注意，这种调整必须增加考虑对 ICMP 内部携带的载荷中所作的源地址目的地址（还包括在 NAPT-PT 情况下的传输层端口号）的更新操作所带来的变化。

#### 4.3.2.5 FTP 应用层网关（FTP-ALG）的支持

因为在 FTP 的控制会话中，其载荷携带了 IP 地址和 TCP 端口号等关于数据会话的信息。这就需要增加一个 FTP-ALG 来为这种使用广泛的互联网应用提供应用层的透明。

在原始的 IPv4 节点上运行的 FTP 应用中，FTP PORT 命令的参数中与 PASV 回应（成功）的参数中包含 IPv4 的地址和 TCP 端口号。表示为 ASCII 就是 h1，h2，h3，h4，p1，p2。然而在 RFC 2428 中建议 EPRT 和 EPSV 命令是 FTP 的扩展，既可以用在 IPv4 也可以用在 IPv6 的节点，它们最终会彻底替代 PORT 和 PASV 命令的使用。下面是 FTP-ALG 的工作原理。

#### 4.3.2.5.1 对于 IPv4 发起的 FTP 会话中的载荷修改

IPv4 主机在它的 FTP 应用程序中可以采用也可以不采用 ERPT 和 EPSV 的扩展命令。如果一个 IPv4 主机发起 FTP 会话，并且采用 PORT 或者是 PASV 命令，FTP-ALG 在转发到 IPv6 节点之前会将这些命令对应的翻译成 EPRT 或 EPSV 命令。同样，在转发回应到 IPv4 节点之前，来自 IPv6 节点的 EPSV 回应会被翻译成为 PASV 回应。EPRT 和 EPSV 命令以及 EPSV 回应的格式如下[FTP-IPv6]：

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>

EPSV<space><net-prt>

（or）

EPSV<space>ALL

Format of EPSV response（Positive）：229 <text indicating

extended passive mode>（<d><d><d><tcp-port><d>）

从 IPv4 节点来的 PORT 命令通过下面的操作被翻译成 EPRT 命令：设置协议的<net-prt>项为 AF #2（IPv6），<net-addr>项中的 IPv4 主机地址（表示为 h1，h2，h3，h4）翻译成 NAT-PT 分配的 IPv6 地址。PORT 命令中的 TCP 端口号 p1，p2 必须在 EPRT 命令中以十进制<tcp-port>给出。另外，在 NAPT-PT 情况下还需要<tcp-port>的翻译。来自 IPv4 节点的 PASV 命令通过将<net-prt>项设置为 AF #2 而翻译成一个 EPSV 命令。从 IPv6 节点返回的 EPSV 回应在被转发到 IPv4 主机之前要被翻译成 PASV 回应。

如果一个 IPv4 的主机发起 FTP 的会话并且采用 EPRT 和 EPSV 命令，FTP-ALG 将只是简单地对这些命令中的参数进行翻译，而并不需要改变这些命令本身。协议号<net-prt>从 AF #1 变到 AF #2。<net-addr>从 ASCII IPv4 地址翻译成 NAT-PT 分配的以字符串表示的 IPv6 地址。只有在 NAPT-PT 情况下才需要对 EPSV 回应中的<tcp-port>项做翻译。

#### 4.3.2.5.2 对于 IPv6 发起的 FTP 会话中的载荷作修改

如果一个 IPv6 的主机发起 FTP 会话，FTP-ALG 会有两种实现方式可以遵从。在第一种方式中，FTP-ALG 可以保持“EPRT”和“EPSV”的命令字符不变，只需要简单地将<net-prt>、<net-addr>和<tcp-port>项从 IPv6 翻译成它们的 NAT-PT 分配信息。<tcp-port>只有在 NAPT-PT 情况下才做翻译。以上的实现方式同样用在对来自 IPv4 节点的 EPSV 回应的处理中。

为了保证对 RFC 2428 的支持，推荐采用这种方式。然而，采用这种方式需要强制 IPv4 的主机升级它们的 FTP 应用以支持 EPRT 和 EPSV 扩展，从而允许以类似的方式访问 IPv4 和 IPv6 的主机。

在第二种方式中，FTP-ALG 将把“EPRT”和“EPSV”的命令字符和它们的参数从 IPv6 节点翻译成它们等同的 NAT-PT 分配的 IPv4 节点信息，同时加入到“PORT”和“PASV”命令中。类似的，来自 IPv4 节点的 PASV 回应在被转发到目标 IPv6 节点之前被翻译成 EPSV 回应。然而，FTP-ALG 将无法翻译 IPv6 节点发出的“EPSV<space>ALL”命令。这时，收到命令的 IPv4 主机可能会返回一个错误的信息显示功能并不被支持。因为 EPSV 是 RFC 2428 强制要求支持的，所以该报错回应会导致很多 RFC 2428 兼容的 FTP 应用失败。

#### 4.3.2.5.3 FTP 控制数据包的头标更新

前面所考虑的载荷翻译都是以数据基于 ASCII 编码为前提的。所以这些翻译可能会导致数据包大小的变化。

如果大小不变，载荷翻译所带来的只是 TCP 校验和的调整。如果数据包的大小前后不一致，TCP 的序列号也需要被变动以反映出 FTP 控制会话中载荷的长度变化。IPv4 头标中的 IP 数据包长度项或者是

IPv6 头标中的 IP 载荷长度项同样也需要被变动以反映出当前新的载荷长度。FTP-ALG 采用了一个表格用来为两个方向的控制数据包纠正 TCP 头标中的序列号和 ACK 号。

表项中应该有会话的 IPv4 与 IPv6 部分的源地址、源数据端口、目的地址和目的数据端口，以及向外发送控制数据包时的序列号增量和从外面接收到的控制数据包的序列号增量。

向外发出的控制数据包的序列号应该增加向外的序列号增量，而且针对该数据包的回应确认序号也同样减去向外的序列号增量。反过来，从外面接收得到控制数据包序列号增加向内的序号增量，相应的回应序号也减去向内的序号增量。

#### 4.3.2.6 NAT-PT 的约束条件和今后的工作

所有限制 NAT[NAT]应用的因素同样也限制 NAT-PT。下面是其中的几个较为重要的限制因素和 NAT-PT 独有的几个约束条件。

##### 4.3.2.6.1 网络拓扑的约束

NAT-PT 过渡机制的应用是有约束条件的。它强制同一个会话相关的所有请求和回应都应通过同一个 NAT-PT 路由器。满足这种要求的一个办法就是使 NAT-PT 工作在一个末稍区域惟一的一个边缘路由器上，这样 IP 数据包或者是在域内产生，或者是发往该域内。这个问题在 NAT 中就已经存在。[NAT-TERM]

注意，这个约束条件不适用于从双栈节点产生的或者发往双栈节点的并不需要翻译的数据包。这是因为在一个双栈节点的配置中，IPv6 的地址中可以暗示该节点的 IPv4 地址，其格式为 PREFIX::x.y.z.w。并且一个双栈路由器可以相应的在 IPv4 和双栈节点之间路由数据包。

该约束也不会影响正常的 IPv6 到 IPv6 的通信，而且实际上只有在没有其他办法可以实现直接通信的时候才去做翻译。例如，NAT-PT 还可以有纯 IPv6 的连接或者是一些通过隧道的 IPv6 连接，如果有以上这两种连接可以完成通信的情况下就应该首选它们。这样就可以保证 NAT-PT 只是作为纯 IPv6 通信的一种辅助工具。



##### 4.3.2.6.2 协议翻译的约束

许多以前 IPv4 头标中的字段的意义在 IPv6 中都已经改变。所以翻译并不是直接完成的。例如，选项头的语义和语法已经在 IPv6 中起了巨大的变化。IPv4 到 IPv6 的协议翻译细节可以参见[SIIT]。

##### 4.3.2.6.3 地址翻译带来的后果

因为 NAT-PT 做了地址翻译，在上层就已经携带了 IP 地址的一些应用将不能正常工作。在这种情况下需要应用层网关（ALG）的配合使用。这也是 NAT 本身带来的问题，详见[NAT-TERM]。

##### 4.3.2.6.4 缺少端到端的安全保证

NAT-PT 一个很重要的约束条件就是端到端的网络层的安全性很难实现，而且如果在应用层就包含了 IP 地址的情况下，其传输层安全和应用层的安全也同样无法实现。这也是网络地址翻译本身产生的约束条件。

##### 4.3.2.6.5 DNS 过渡和 DNS SEC

在前面曾经描述的方案中包括了 DNS 信息的翻译，很显然，这种方案不能应用在联合采用安全 DNS（DNS SEC）的情形下。就是说，一个在 IPv6 域中经过授权的域名服务器不能完成为返回给 IPv4 世界的应答签名。这样的结果是，一个等待要求被签发的 DNS 回应的 IPv4 末端节点会拒收这个已经被 NAT-PT 篡改过的回应。

然而值得庆幸的是，只有在 IPv6 域中的需要被 IPv4 世界访问的服务器才会因为以上的约束条件付出代价，因为 IPv4 的末端节点可能因为 DNS 回应没有被签名而无法访问 IPv6 的服务器。

很明显，在 DNS 服务器和末端主机的解析中采用了 DNS SEC 后，本标准中规定的方法将无法工作。

#### 4.3.2.6.6 适用性声明

NAT-PT 是适合于一个只安装了 IPv6 末稍网络再连接到 IPv4 网络或 IPv4 IPv6 混合网络时用在边缘的非常有用的过渡工具。

NAT-PT 运行在最简单的形式下，即没有 DNS-ALG 的支持时，只能提供 IPv6 末稍区域和 IPv4 区域的单向连接。就是说只有 IPv6 末稍区域的 IPv6 节点发起的会话才可以被翻译，而由 IPv4 节点发起的会话将不会被处理而丢弃。这就使 NAT-PT 在不需要提供 IPv4 网络可见的服务器情况下就可以保持和 IPv4 网络的连接。

NAT-PT 结合了 DNS-ALG 从而提供了 IPv6 末稍区域和 IPv4 世界的双向连接。也就是说允许 IPv4 的节点发起到 IPv6 末稍区域的连接。

有一些应用对地址的稳定性要求很高，NAT-PT 中采用的动态地址重复将满足不了这些应用的要求。对于运行类似应用的主机，NAT－PT 可以按照预先的配置提供静态的 IPv6 地址和 IPv4 地址的映射。这就防止了 NAT-PT 地址变化引起的操作失败。

#### 4.3.2.6.7 安全性的考虑

NAT-PT 端到端的网络层的安全性很难实现，而且在应用层就包含了 IP 地址的传输层和应用层的安全也同样无法实现。

DNS-ALG 不能同时和安全 DNS 使用。最后所有 NAT[NAT-TERM]基于安全性的考虑都适用于本标准。



### 4.3.3 TRT

#### 4.3.3.1 介绍

传输中继转换器简称“TRT”（transport relay translator）适用于纯IPv6网络与纯IPv4网络通信的环境。TRT系统位于纯IPv6主机和纯IPv4主机之间，可以实现{TCP，UDP}/IPv6与{TCP，UDP}/IPv4的数据的对译。

TRT的优点主要表现在：

（1）TRT不需要修改纯IPv6主机和纯IPv4主机。

（2）TRT不需要考虑PMTU和数据包分段的问题。

TRT的不足如下所述：

（1）TRT只支持双向传送，不支持如单向的多播数据包的转换。

（2）TRT是一个位于两个通信实体中间的有状态的传输中继转换系统。即使在同一个区域内部署多个TRT系统时，一个传输层会话必须通过同一个TRT系统。

（3）TRT系统本身无法进行非NAT友好协议的转化，例如，IPSec等。

以下描述假定所有业务是由IPv6主机发起的，目的是IPv4主机。如果采用合适的地址映射机制，TRT也可以支持IPv4到IPv6的数据业务。

#### 4.3.3.2 IPv4-to-IPv4 传输中继

传输中继可以分为 TCP 中继和 UDP 中继两类。

##### 4.3.3.2.1 TCP 中继

TCP中继主要用于防火墙有关的产品中，需要实现以下功能：

（1）阻止IP数据包通过中继系统；

（2）允许{TCP，UDP}数据流经过中继后通过系统。

如图25所示，“TCP中继系统”不会转发内部的IP数据包到外部网络，反之亦然。它只对网络之间特定端口的TCP数据流进行中继传输。

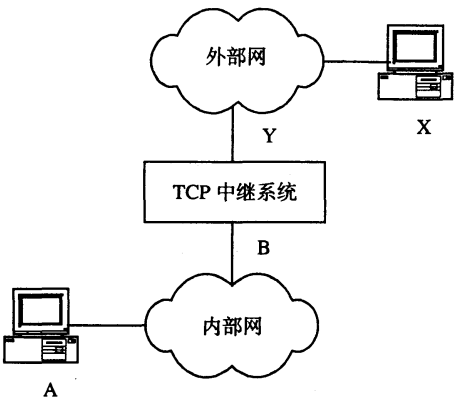



图 25　TCP 中继

当源主机（IP地址是A）试图用TCP去连接目的主机（IP地址是X），TCP数据包被路由器转发到基于路由器设计的TCP中继系统。这个中继系统不是数据包的目的地，但仍然收到并接受这个数据包。这个TCP中继系统假装有这个目的IP地址（X），建立一条同A的TCP连接，然后，TCP中继系统建立另一条Y到X的TCP连接，进行从A到X的中继传输。因此，在这个图中，有两个TCP连接建立，从A到B（如同到X）和Y到X，如下所述：

TCP/IPv4：源主机（A）→TCP中继系统（如同X），在IPv4数据包头中的地址：A→X；

TCP/IPv4：TCP中继系统（Y）→目的主机（X），在IPv4数据包头中的地址：Y→X；

TCP中继系统需要去捕获不是发往自己的TCP数据包，具体的实现方法不在本标准的范围之内。

#### 4.3.3.2.2　UDP 中继

UDP中继的实现原理类似于TCP中继。UDP中继可以识别双向UDP数据流，记录地址/端口对，并对表项进行超时处理。

#### 4.3.3.3　IPv6-to-IPv4 的传输中继转换器

以下TRT过渡技术原理的描述主要是针对TCP会话的，对于UDP也可以用类似的方式实现。

为传输中继转换机制保留了一个IPv6前缀C6::/64，C6::/64应该属于站点中的单播地址空间中的一部分。路由必须被配置成将C6::/64的数据包转发到TRT系统。图26表示了这个网络配置，用“伪前缀”标记的子网实际上是不存在的。同样，我们假定数据包源主机是IPv6节点，目的主机是IPv4节点。

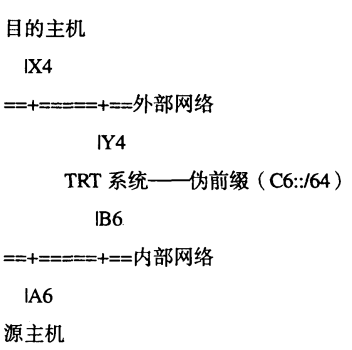



图 26　UDP 中继

当源主机（IPv6地址是A6）向目的主机（IPv4地址是X4）发起通信时，需要建立一条面向C6::X4的TCP/IPv6连接。举例来说，如果C6::/64等于fec0:0:0:1::/64，X4等于10.1.1.1，那么这个目的地址就应该是

fec0:0:0:1::10.1.1.1。数据包被路由转发到TRT系统，并且被它捕获。这个TRT系统接受了位于A6和C6::X4之间的TCP/IPv6连接。然后，这个TRT系统根据目标地址的低32位（IPv6地址是C6::X4）去得到一个真实的IPv4目标地址（IPv4地址是X4）。于是就建立一条从Y4到X4的TCP/IPv4连接，然后在这两个TCP连接之间转发数据流。

因此TRT系统为一个TCP会话维护两条TCP连接，一条是TCP/IPv6，另一条是TCP/IPv4。

#### 4.3.3.4 地址映射

TRT中IPv6源节点需要将目标IPv4的地址映射成前缀为C6::/64的IPv6地址，有三种地址映射方式：

- 从源主机的静态地址映射表（像 UNIX 中的/etc/host）中解析出来；
- 指定专门的 DNS 服务器实现；
- 修改源节点的 DNS 解析实现。

#### 4.3.3.5 实现的注意事项

用于UDP的TRT必须注意UDP/IPv6侧的PMTU。由于TRT不直接转发IPv6/IPv4间IP的数据包，因此TRT是否执行IPv6 PMTU功能与IPv4数据业务无关，一个简单的方法是，TRT总是按IPv6最小MTU（1280字节）分段向UDP/IPv6侧发送数据包。这就省去了IPv6的PMTU探测。

虽然TRT系统只能中继传输{TCP，UDP}数据流，它也需要去检查目的为C6::X4的ICMPv6数据包，这样它就能够识别PMTU发现消息以及其他A6与C6::X4之间的通知消息。

当转发TCP数据流时，TRT系统应当谨慎地处理紧急数据，可参见RFC 793。

当中继传输非NAT协议数据包时，TRT系统可能需要去修改上层数据的内容，就像其他需要修改IP地址的转换器一样。

当把一个TRT系统部署到一个大型的IPv6网络区域的时候，必须考虑到它的升级能力。可升级能力参数包括：

（1）操作系统核心接受的连接数目；

（2）一个用户进程能够转发的连接数（等于每个进程的文件句柄数）；

（3）一个TRT系统的传输中继进程数。设计时必须考虑用适当用户数量的进程去支持适当数量的连接。

在一个大型网络区域，可能部署多个TRT系统。可以用下面的步骤去实现：

（1）配置多个TRT系统；

（2）给它们配置不同的伪前缀；

（3）为了均衡负载，所有的源主机从这些前缀中随机挑选一个。

如果网络配置了DNS，（3）可以实现如下：

- 配置 DNS 服务器，使不同的 DNS 服务器返回不同的伪前缀，然后向不同的源主机通告不同 DNS 服务器；
- 让 DNS 服务器随机的分配这些前缀。

一旦一个TCP连接建立，这个目标地址就不会改变，负载均衡就成为可能。

对于地址映射，对于一个大型的网络，建议采用专门的DNS服务器，静态地址映射则用于小型的网络。

- 传输方面：与转换 DNS 请求/应答数据包相比，提供递归 DNS 服务更容易一些。如果要采用 TRT 传输中继 DNS 数据包，那么要把 C6::X（C6 是 TRT 的保留前缀，X 是一个 DNS 服务器的 IPv4 地址）放

进/etc/resolv.conf。这个配置比起通常想像的要复杂的多。

• 净荷方面：在某些部署中，无需修改传输 DNS 请求/应答更有优势；而在某些部署中，通过 TRT 系统将 IPv6DNS 请求（如对一条 AAAA 记录的请求）转换成对 A 记录的请求更好些，反之亦然。这可以根据具体用户场所的安装/设置情况决定。

#### 4.3.3.6 安全方面的考虑

恶意攻击者可能会像利用SMTP一样利用TRT，公开将数据中继到一个特定的IPv4目的地。这类似于入口过滤，或者其他一些不正当情况。TRT应当实现一些访问控制机制来预防这类事情。

一个TRT系统实现不好的话可能会发生堆栈缓冲溢出，这类问题与具体的实现相关。

由于TCP/UDP中继服务的特点，不推荐TRT用于基于源IP地址的认证类协议的中继传输（也就是rsh/rlogin）。

IPSec数据包不能通过TRT中继。

如果采用了修改RRs的DNS代理，将使解析者无法验证DNS SEC签名。

### 4.3.4 BIS

#### 4.3.4.1 引言

RFC 1933 为过渡的起始阶段制定了包括双栈和隧道技术的过渡机制，相应的主机和路由器已经问世。但是与大量基于 IPv4 的应用相比，IPv6 应用的数量却微乎其微。为了促进从 IPv4 到 IPv6 的平滑过渡，迫切需要与 IPv4 应用数量相当的基于 IPv6 的应用。然而，这可能需要很长一段时间。

本标准规定了一种在 IP 安全领域里采用 BIS 技术的双栈主机机制。这一机制是在主机中添加若干个模块，用于监测 TCP/IP 模块与网卡驱动程序之间的数据流，并进行相应 IPv4 与 IPv6 数据包之间的相互翻译。当与其他 IPv6 主机进行通信时，在这台主机内部给对应 IPv6 主机分配一些 IPv4 地址，这些地址只在这台主机内部使用，而且这种分配过程是通过 DNS 协议自动来完成的。因此，用户不用关心与其通信的对应主机是不是 IPv6 主机。也就是说，主机可以采用现有的 IPv4 应用和其他 IPv6 主机进行通信，使其成为能够既支持 IPv4 应用又同时支持 IPv6 应用的双栈主机，从而扩大了双栈主机的应用领域。甚至由于各自的角色和功能不同，这种双栈主机机制可以和其他的转换机制共存。

#### 4.3.4.2 组成模块

在 RFC 1933 中定义的双栈主机中需要有应用程序、TCP/IP 模块以及 IPv4 和 IPv6 的地址。本标准所建议的主机系统用三个模快支持主机上的 IPv4 应用，来代替 IPv6 的应用和其他 IPv6 主机进行通信。这三个模块是翻译器、扩展域名解析器、地址映射器，图 27 说明了各种模块在主机中的位置。

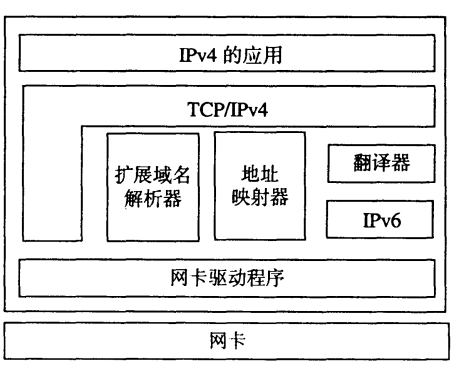



图 27 采用 BIS 机制的双栈主机的系统模块组成

##### 4.3.4.2.1 翻译器

利用 RFC 2765[SIIT]中定义的 IP 转换机制进行 IPv4 和 IPv6 间的转换。

当翻译器收到来自 IPv4 应用的数据包时，将 IPv4 头转换为 IPv6 头，然后对转换后的数据包进行适当的分段处理( 因为 IPv6 头至少比 IPv4 头大 20 字节 ),发送到 IPv6 网络中。当从 IPv6 网络中接收到 IPv6 数据包时，翻译器做相反的转换，但此时不对数据包进行分段处理。

#### 4.3.4.2.2 扩展域名解析器

扩展域名解析器用于对来自 IPv4 应用的请求返回一个正确的响应。

应用通常会向域名服务器发出一个解析目标主机名的 A 型记录查询请求。扩展域名解析器收到这种查询请求后，按所查询的目标主机名生成另外的包含 A 型和 AAAA 型两种记录的查询请求，并发向域名服务器。如果解析出 A 型记录，解析器就把此记录原封不动地返回给应用。在这种情况下，翻译器不进行 IP 转换。如果只能解析出 AAAA 型记录，解析器就会要求地址映射器给这个 IPv6 地址分配一个 IPv4 地址，然后把这个 IPv4 地址作为 A 型记录返回给应用。

注：这种操作类似于[NAT-PT]中采用的 DNS 应用网关。

#### 4.3.4.2.3 地址映射器

地址映射器负责管理一个 IPv4 地址池，这个地址池里也可以包含私用地址。同时，地址映射器维护一张包含有 IPv4 和 IPv6 地址对的映射表。当解析器和翻译器需要为一个 IPv6 地址分配一个 IPv4 地址时，地址映射器从其管理的地址池中选出一个 IPv4 地址，并在映射表中动态地记录下地址之间的映射关系。在以下两种情况中，需要在映射表中记录地址映射关系：

（1）解析器只得到目标主机名字的 AAAA 型记录，并且在映射表中没有相应的 IPv6 地址记录。

（2）翻译器收到一个 IPv6 数据包，映射表中没有对应于数据包源地址的映射记录。

注：仅在初始化这张映射表时有一个例外，地址映射器要静态地把自己的 IPv4 地址和 IPv6 地址对登记到映射表里。



### 4.3.4.3 通信示例

本节描述了名为“dual stack”的双栈主机用它的 IPv4 应用与一台名为“host6”的 IPv6 主机相互通信的过程。

#### 4.3.4.3.1 发送过程

本小节描述了“dual stack”作为发送方所执行的一系列操作，通信由“dual stack”发起。

应用向域名服务器发出一个查询“host6”主机 A 型记录的请求。

域名解析器捕获这个请求，生成另一个查询“host6”主机 A 型和 AAAA 型两种记录的请求发给域名服务器。在本次通信中，只解析得到 AAAA 型记录，于是域名解析器要求地址映射器给得到的 IPv6 地址分配一个 IPv4 地址。

注：在与 IPv4 主机通信的情况下，只解析得到 A 型记录，域名解析器把记录返回给应用，不进行以下的 IP 转换操作。

地址映射器从地址池中选择一个 IPv4 地址并返回给解析器。

解析器为分配的 IPv4 地址产生 A 型记录并返回给应用。

应用向“host6”发送 IPv4 数据包。

IPv4 数据包到达翻译器。翻译器试图把 IPv4 数据包转换成 IPv6 数据包，但是不知道如何对 IPv4 数据包中的源地址和目的地址进行翻译。于是翻译器请求地址映射器为其提供地址映射记录。

地址翻译器在映射表中进行搜索，返回相应的 IPv6 源地址和目的地址给翻译器。

注：地址映射器事先在表中注册了自己的 IPv4 和 IPv6 地址。

翻译器把 IPv4 的数据包翻译成 IPv6 的数据包，并根据需要对 IPv6 数据包进行分段处理，再发送到

IPv6 的网络上。

IPv6 数据包到达“host6”。

然后，“host6”发送一个新的 IPv6 数据包给“dual stack”。

IPv6 数据包抵达“dual stack”上的翻译器。

翻译器从地址映射器得到前面 IPv6 源地址和目的地址的映射记录。

然后，翻译器把 IPv6 的数据包翻译成 IPv4 数据包，上交给应用。

图 28 说明了上述过程。

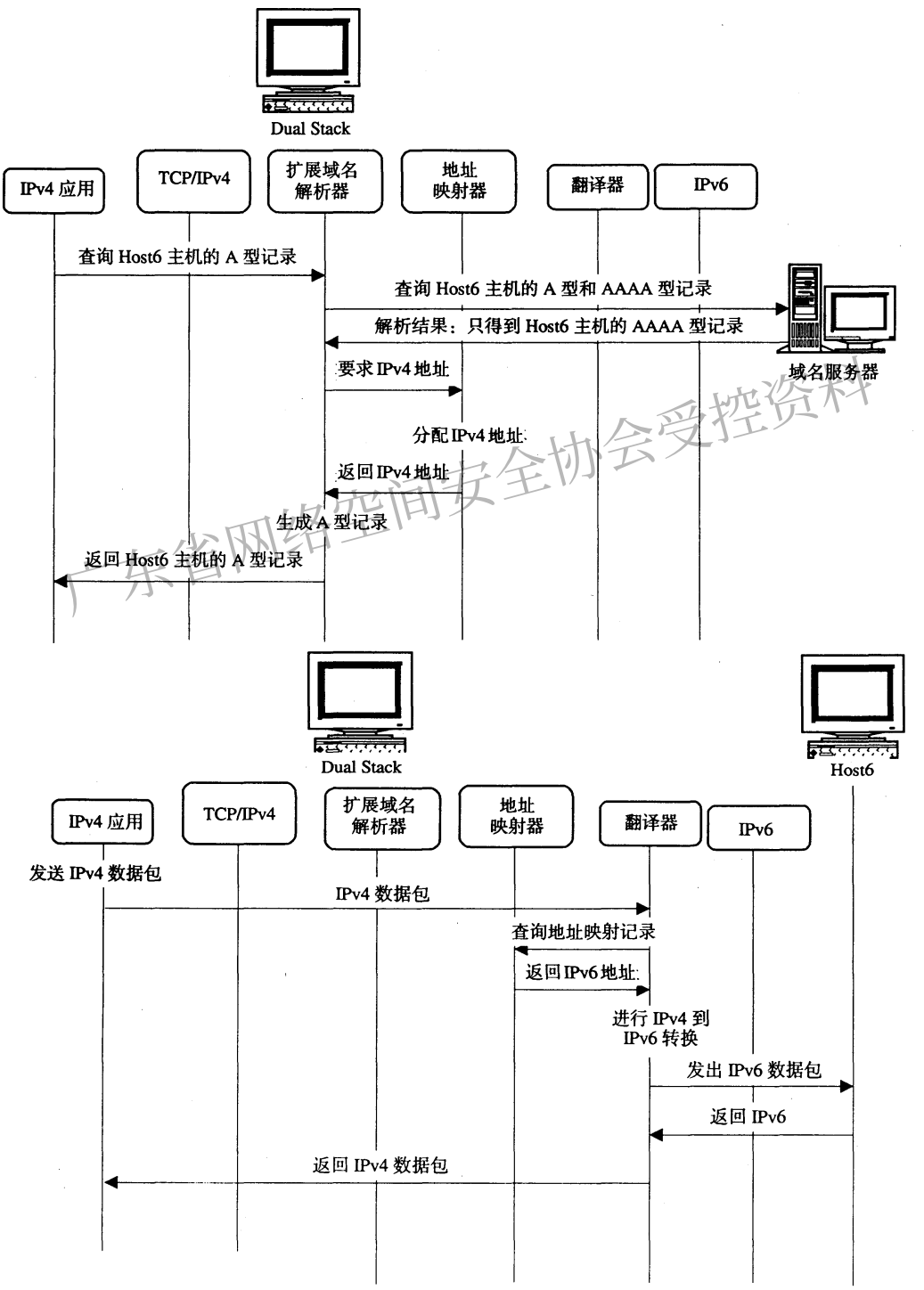





图 28　采用 BIS 机制的双栈主机向 IPv6 主机发起通信的过程

#### 4.3.4.3.2 接收过程

本小节描述了主机“dual stack”的接收过程，通信由主机“host6”发起。

“host6”通过域名服务器解析主机“dual stack”的 AAAA 记录，然后向得到的 IPv6 地址发送 IPv6 数据包。

IPv6 数据包到达“dual stack”的翻译器。

翻译器试图将 IPv6 数据包翻译成 IPv4 数据包，但不知道如何对数据包中 IPv6 源地址和目的地址进行翻译。于是翻译器请求地址映射器提供地址映射记录。

地址映射器在映射表中搜索源地址和目的地址各自的记录，并找到 IPv6 目的地址的记录。

但是如果在地址映射表中没有 IPv6 源地址的记录，那么地址映射器就从其管理的地址池中选一个 IPv4 地址，作为 IPv4 的源地址，然后把源地址和目的地址返回给翻译器。

翻译器把 IPv6 数据包翻译成 IPv4 数据包，然后上交给应用。

应用发送出新的 IPv4 数据包给“host6”。接下来的操作见 4.3.4.3.1 节。

图 29 描述了上述操作过程。

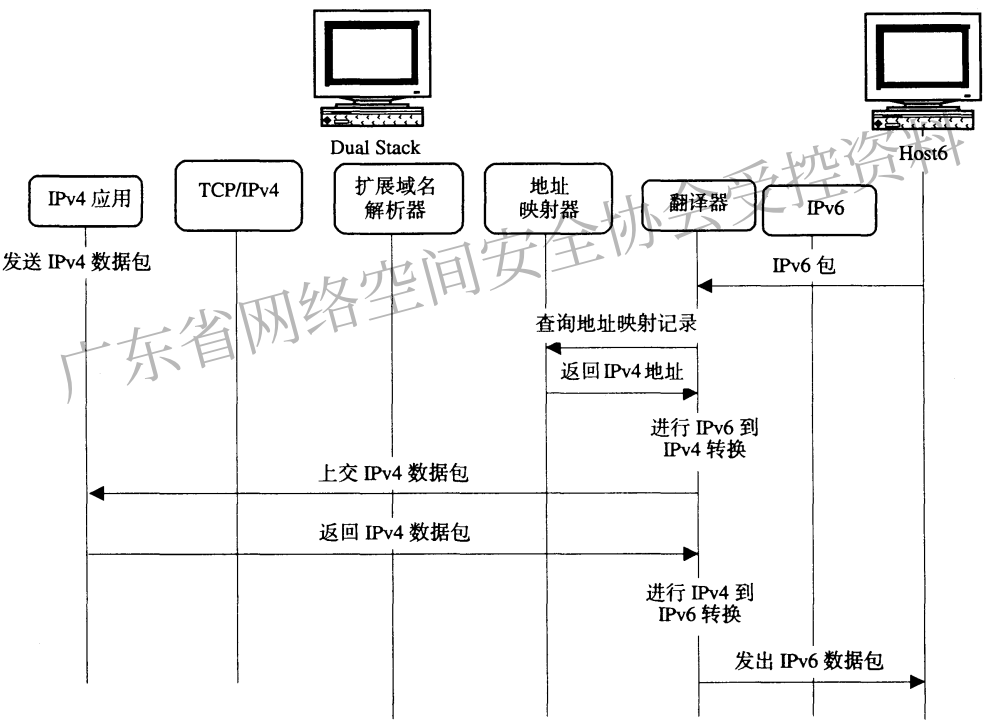





**图 29 采用 BIS 机制的双栈主机从 IPv6 主机接收通信的过程**

### 4.3.4.4 注意事项

#### 4.3.4.4.1 IP 转换

类似于 NAT，IP 转换需要对嵌在应用层协议中的 IP 地址进行翻译，FTP 是一种典型情况。所以在这种情况下很难实现完整的转换。

#### 4.3.4.4.2 IPv4 地址池和映射表

地址池有可能采用私用地址，所以是很大的地址空间。然而，如果这台主机和大量的 IPv6 主机通信，而地址映射器对映射表中已注册了的表项不加清除的话，有可能耗尽地址池中的地址，导致没有空闲的 IPv4 地址分配给 IPv6 主机。为了解决这个问题，可以让映射器释放表中过时的记录，重用 IPv4 地址生成新记录。

#### 4.3.4.4.3 内部 IPv4 地址分配。

在主机内部，分配给 IPv6 目的主机的 IPv4 地址只在主机内部有效。所以并不会对其他主机产生负面影响。

#### 4.3.4.5 可用性与限制

#### 4.3.4.5.1 可用性

这种机制适用于在过渡的最初始阶段，即还有许多应用没有做适应 IPv6 修改的那个阶段。在大部分的应用都已经进行了修改后，也可以支持那些还有某些应用尚未修改的用户。因为这种机制允许主机利用已有的 IPv4 应用与 IPv6 主机进行通信。那么，即使主机没有 IPv6 应用，也能够和 IPv4 和 IPv6 网络保持连通。

请注意这种机制也可以和完整的 IPv6 协议栈协同工作。采用这种机制的主机可以运行 IPv4 应用同 IPv4 主机和 IPv6 主机通信，也可以用 IPv6 应用通过 IPv6 协议栈同 IPv6 主机通信。

#### 4.3.4.5.2 限制

这种机制只适用于单播通信，不适用于组播通信。组播通信需要其他的机制。

尽管这种机制允许主机采用已有的 IPv4 应用和 IPv6 主机进行通信，但是由于翻译器不能把 IPv4 的参数都转换成 IPv6 相应的参数，所以不适用于使用了某个 IPv4 参数的应用。同样，除了分段头和路由头外，翻译器也不能进行 IPv6 到 IPv4 的参数转换。所以接收到的带有参数的 IPv6 数据包都会被丢弃。

就像 NAT 一样，IP 转换需要对嵌在应用层协议中的 IP 地址进行翻译，FTP 是典型的应用情形。所以很难完整转换所有这样的应用。

由于在数据中含有 IP 地址，所以网络层之上的安全策略不能在采用这种机制的主机上使用。

最后，由于扩展域名解析器不能处理安全域名服务（Secure DNS）协议，所以也不能与其协同工作。

#### 4.3.4.6 安全性考虑

本节考虑了本标准所建议的双栈主机的安全性问题。

当采用 BIS 机制的双栈主机上的 IPv4 应用和其他 IPv4 主机通信时，可以实施所有层面上的安全策略。同样，当运行 IPv6 应用的主机通过完整的 IPv6 协议栈和其他 IPv6 主机通信时，像其他普通的 IPv6 间通信一样，也可以在所有层面上实施安全策略。然而，当主机上的 IPv4 应用利用 BIS 机制和其他的 IPv6 主机通信时，不能实施网络层以上的安全策略。因为当携带有 IP 地址的协议数据被加密时，或者协议数据以 IP 地址作为键值进行加密后，这种机制不能在 IPv4 和 IPv6 的数据之间进行转换。因此建议需要采用安全机制与其他 IPv6 主机进行通信时，最好把现有的应用升级为 IPv6 的应用。

### 4.3.5 BIA

#### 4.3.5.1 介绍

BIA 技术在双栈主机的 Socket API 模块与 TCP/IP 模块之间加入一个 API 翻译器，所以它能够在 IPv4 的 Socket API 函数和 IPv6 的 Socket API 函数间进行互译。这种机制简化了 IPv4 和 IPv6 间的转换，但无需进行 IP 头的翻译。

采用 BIA 的双栈主机假定在本地节点上同时存在 TCP/IPv4 和 TCP/IPv6 两种协议栈。

当双栈主机上的 IPv4 应用程序与其他 IPv6 主机通信时，API 翻译器检测到 IPv4 应用程序中的 Socket API 函数，并调用 IPv6 的 Socket API 函数与 IPv6 主机通信，反之亦然。为了支持 IPv4 应用程序与目标 IPv6 主机间的通信，在 API 翻译器中，IPv4 地址池由域名解析器进行分配。

#### 4.3.5.2 可用性与不适用性

（1）可用性

BIA 的主要目的与 BIS 一样，使 IPv4 应用在不作任何修改的情况下，可以和 IPv6 主机通信。BIS 适用于没有 IPv6 协议栈的系统，而 BIA 适用于具有 IPv6 协议栈，但是一些应用在 IPv6 上仍不可用的系统；并且这些应用的源代码不可用，从而阻碍移植这些应用程序。BIA 对那些还没有将所有应用程序准备好的早期用户有效，但不适于主流产品的使用。

当运行 BIA 的客户节点试图通过一个只与 IPv4 应用程序关联的端口与一个双栈节点通信时（见 5.5 节），有两种解决方法：

- 客户端应用程序应该轮询所有的相关地址，最后尝试 IPv4 地址。
- 通过 BIA 来进行通信。

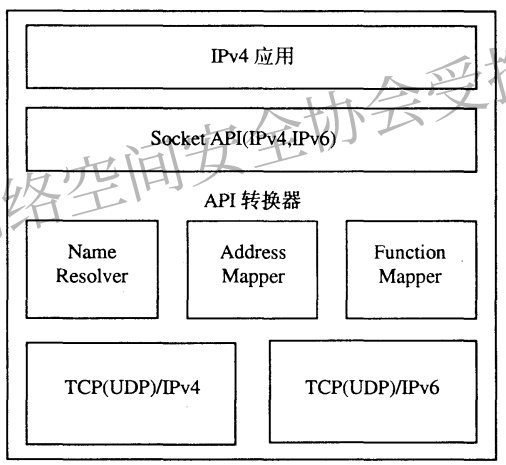
到本标准完成为止，仍不清楚哪种方法更可取（可能依赖于应用），需要从试验中得到相关信息。

不适用性

BIA 不应当用于其源代码可用的应用。我们强烈建议应用程序源代码可用时，程序开发者不应当使用这种机制，它也不应当成为拒绝或拖延移植软件的借口。

#### 4.3.5.3 采用 BIA 双栈主机的结构

图 30 显示了装有 BIA 的双栈主机结构。






Name Resolver——域名解析器；Address Mapper——地址映射器；Function Mapper——函数映射器

**图 30 采用 BIA 机制的双栈主机的结构模型**

在 RFC 2893 中定义的双栈主机需要应用程序，TCP/IP 模块以及 IPv4 和 IPv6 地址。本标准建议的主机用一个 API 翻译器，使现存的 IPv4 应用程序能与其他 IPv6 主机通信。API 翻译器包含三个模块：域名解析器、地址映射器和函数映射器。

##### 4.3.5.3.1 函数映射器

在 IPv4 的 Socket API 函数与 IPv6 的 Socket API 函数间相互翻译。当从 IPv4 应用中检测到 IPv4 的 Socket API 函数时，函数映射器拦截函数调用，并采用一个对应于该 IPv4 API 函数的 IPv6 API 函数同 IPv6 目标主机进行通信。当从来自 IPv6 主机的数据中检测到 IPv6 的 Socket API 调用时，则做相反的转换。

##### 4.3.5.3.2 域名解析器

对 IPv4 应用程序的请求返回一个正确的应答。

当某一 IPv4 应用调用解析库函数来解析某一主机名字时，BIA 拦截函数调用，用同样功能的 IPv6 函数调用来替代解析 A 和 AAAA 型记录。

若可以得到 AAAA 型记录，那么就需要地址映射器给相应的 IPv6 地址分配一个 IPv4 地址，然后为其产生一条 A 型记录，返回给应用。

#### 4.3.5.3.3 地址映射器

在主机内部维护一张 IPv4 与 IPv6 地址对的表格，分配的 IPv4 地址来自 IPv4 地址池中，采用未采用的 IPv4 地址（如 0.0.0.1 ~ 0.0.0.255）。

当域名解析器和函数映射器需要地址映射器为 IPv6 地址分配一个 IPv4 地址时，地址映射器从 IPv4 的地址池中选择一个未用的 IPv4 地址，并动态在映射表中注册。在下面两种情况下需要进行注册：

（1）域名解析器得到目标主机的一个 AAAA 型记录，而在地址映射表中没有此 IPv6 地址的记录；

（2）函数映射器从接收到的数据中得到一个 Socket API 函数调用，而在地址映射表中并无数据包中 IPv6 源地址的记录。

### 4.3.5.4 举例

本节描述了运行 IPv4 应用程序的双栈主机“dual stack”与 IPv6 主机“host6”通信的过程。

在本节中，各箭头指示有如下含义：

→由应用和 API 翻译器里的域名解析器生成的用于域名解析的 DNS 消息；

+++>为了域名解析器和函数映射器而进行的对地址映射器的 IPv4 地址请求与回复操作；

===> 由应用和 API 翻译器中函数映射器产生的 socket API 函数数据流。

#### 4.3.5.4.1 发送过程

本节描述了“dual stack”向“host6”发送数据的过程。

当 IPv4 应用向它的域名服务器发送 DNS 查询请求时，域名解析器拦截了这个请求，并产生一个新的查询请求来解析 A 和 AAAA 两种记录。当只有 AAAA 记录被解析时，域名解析器会要求地址映射器为 IPv6 地址分配一个 IPv4 地址。

域名解析器为分配的 IPv4 地址产生一条 A 记录，返回给 IPv4 应用程序。

为了使 IPv4 应用程序能够向“host6”发送 IPv4 数据包，它调用 IPv4 的 Socket API 函数。

函数映射器检测到来自于应用的 Socket API 函数，若其来自于 IPv6 的应用，则跳过翻译程序。若是来自于 IPv4 的应用，则需要一个 IPv6 地址来重新调用 IPv6 的 Socket API，函数映射器向地址映射器请求一个 IPv6 地址，地址映射器从表中查找到 IPv4 地址对应的 IPv6 地址，函数映射器采用这个地址调用相应的 IPv6 socket API 函数。

当函数映射器接收到一个 IPv6 函数调用时，为了把 IPv6 的 Socket API 调用转换成 IPv4 的 Socket API 调用，它向地址映射器发送一个 IPv4 地址请求。然后函数映射器利用此 IPv4 地址进行 Socket API 函数调用。

图 31 描述了上述发送过程。

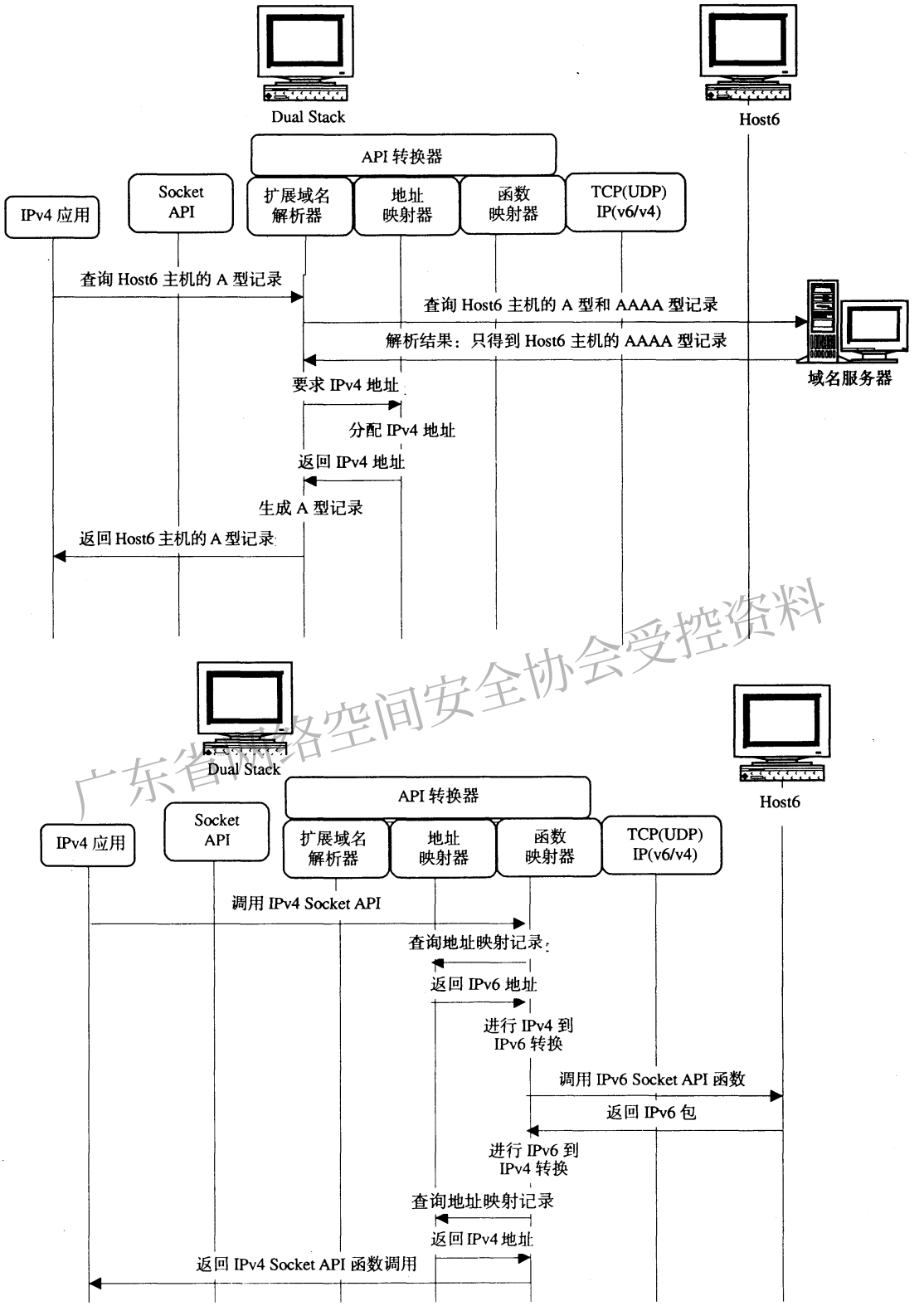





图 31　采用 BIA 机制的双栈主机向 IPv6 主机发起通信的过程

#### 4.3.5.4.2　接收过程

本节描述了“dual stack”的接收过程，通信过程由“host6”发起。

“host6”通过它的域名服务器解析“dual stack”的 AAAA 记录，然后向“dual stack”发送一个 IPv6

的数据包。IPv6 数据包到达“dual stack”，函数映射器检测到它。为了通过调用 IPv4 的 API 函数和 IPv4 应用通信，函数映射器向地址解析器发送一个 IPv4 地址请求。然后，函数映射器用返回的 IPv4 地址发起一个 IPv4 的 Socket API 调用。

图 32 描述了上述过程。

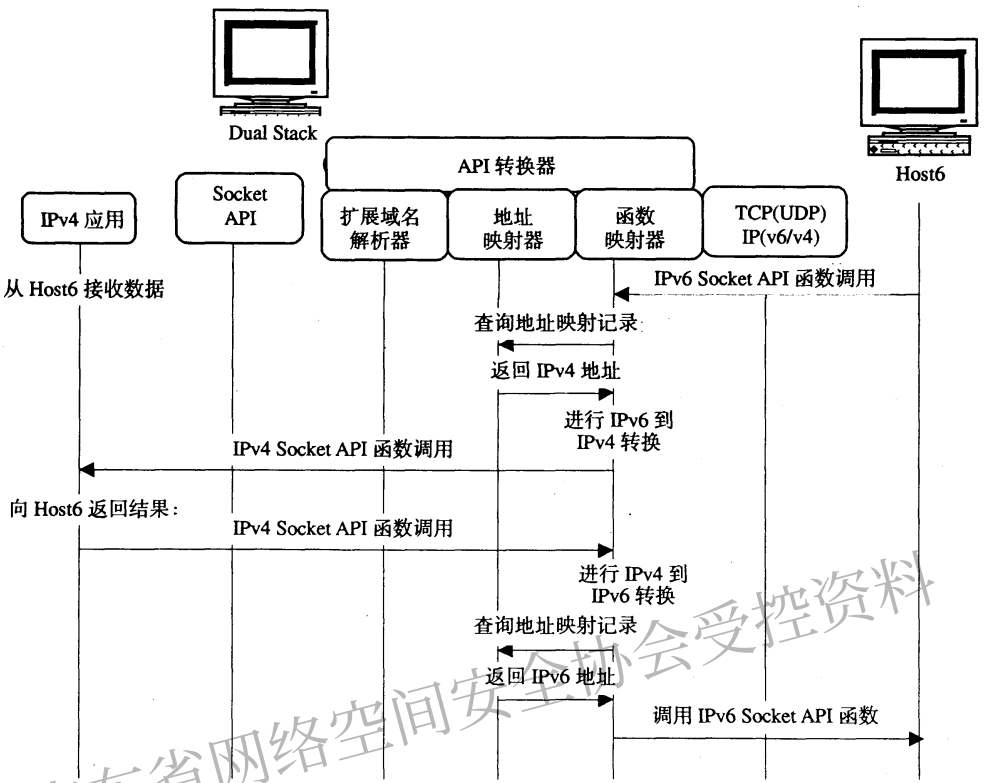





图 32　采用 BIA 机制的双栈主机从 IPv6 主机接收通信的过程

#### 4.3.5.5　注意事项

##### 4.3.5.5.1　Socket API 转换

IPv4 的 Socket API 函数从语义上转换成相应的 IPv6 Socket API 函数。附录 A 列出了能够被 BIA 所拦截的 API 函数。在 API 函数中转换内嵌于应用层协议的 IP 地址，其实现有赖于操作系统。

##### 4.3.5.5.2　ICMP 消息处理

当某一应用需要来自网络层 ICMP 消息的相关值( 如类型、代码等 )时，依据[SIIT]这类值可以在 ICMP4 和 ICMP6 之间进行互译，可以通过 raw rocket 来实现。

##### 4.3.5.5.3　IPv4 地址池和映射表

地址池由未使用的 IPv4 地址组成。这个地址池可以在节点中以不同的粒度来实现，即可以每个节点一个池，也可以更精细的粒度：每个用户、每个进程一个地址池。然而，如果大量的 IPv4 应用和 IPv6 的主机通信时，可能耗尽可用地址，导致 IPv4 应用不能和 IPv6 的主机通信，所以需要对地址池内的地址进行有效管理。例如，可以让地址映射器清除旧的记录，使得释放出的 IPv4 地址能被重新分配利用。这个问题同 BIS 中所遇到的相同。在每个节点一个地址池的情况下，更有可能出现地址被耗尽的情况。

##### 4.3.5.5.4　在内部分配 IPv4 地址

由于从地址池中分配给 IPv6 目的主机的 IPv4 地址是未使用的 IPv4 地址（如 0.0.0.1~0.0.0.255 )。所以在主机之外这些 IPv4 地址也不会与其他私用地址出现冲突。

##### 4.3.5.5.5　DNS 查询结果（AAAA）与另一端应用程序版本不匹配问题

若正在使用的服务器端应用不支持 IPv6，但是它又运行在一台支持其他 IPv6 服务的主机上，而且这台主机还在 DNS 中以 AAAA 型记录出现。那么采用 BIA 的 IPv4 客户端应用可能连接不到这个服务器端的应用上，因为在 DNS 的查询结果（AAAA）和服务器应用的版本之间出现了不匹配的情况。解决方法是尝试所有在 DNS 中列出的地址，而不应只尝试一次即宣告失败。有两种实现手段：客户端应用应该轮询所有的相关地址，最后尝试 IPv4 地址；或者通过扩展 BIA 中的域名解析器和 API 翻译器功能来实现。为了在扩展域名解析器返回的地址中找到对端应用所能用的地址，BIA 应不断重复尝试，这也与应用密切相关。由于 BIA 能够发现连接呼叫失败，所以 BIA 对于 TCP Sockets 可以重复尝试各种可能的地址。但是对于 UDP Sockets，即便可能，BIA 也很难发现可工作的 IP 地址，因此应用必须重复尝试各种可能的地址，直到发现一个可用地址为止。

另一种避免这种问题的方法是仅当通信对端的 A 型记录不存在时，BIA 才发生作用。这样一台采用 BIA 的双栈主机上的应用到另一台双栈主机的数据流只采用 IPv4 协议。

#### 4.3.5.5.6 实现的相关问题

一些操作系统支持预载库函数，因此很容易实现 API 翻译器。例如，用户可以定义自己的 API 函数来替代系统已有的函数，以实现 API 函数间的转换。这样每一个 IPv4 应用都用预载函数库作为转换库，在运行前，库函数会动态的绑定到应用上。

其他一些操作系统支持用户定义的分层协议，允许用户开发一些附加的协议附加到已存在的协议栈上。在这种情况下，API 翻译器可以作为一个分层协议模块来实现。

在以上两种实现方法中，均假设系统中存在 TCP/IPv4 和 TCP/IPv6 两种协议栈，而不需要修改或增加 TCP-UDP/IPv6 协议栈。

#### 4.3.5.6 限制

类似于[NAT-PT]，BIA 需要转换嵌在应用层协议中的 IP 地址，例如，FTP。所以这种机制可能不适用于那些其负载中包含地址的新应用。

这种机制仅支持单播通信。为了支持组播通信，需要在函数映射模块中增加新的功能。

由于 IPv6 的 API 具有新的高级参数，所以转换带有这种参数的 IPv6 API 是很困难的。因此，接收到的带有高级参数的 IPv6 数据包会被丢弃。

#### 4.3.5.7 安全性考虑

BIA 的安全性考虑主要依赖于[NAT-PT]的安全性。差别在于 BIA 的地址翻译发生在 API 层而不是在网络层。由于这种翻译机制发生在 Socket API 层，所以采用这种机制运行 IPv4 应用的主机和其他 IPv6 主机通信时，可以利用网络层的安全策略（例如，IPSec）。因为不存在 NAT-PT 中的 DNS ALG，所以也不会和 DNS SEC 产生冲突。

# 附 录 A
## （资料性附录）
## IPv6 地址申请与分配

### A.1 地址分配机构

RFC 1881 规定，IPv6 地址空间的管理必须符合 Internet 的公众利益，并且要求至少要有一个中心权威机构对地址进行分配，才能保证地址的使用效率。早期主要由 IANA（Internet Assigned Numbers Authority，Internet 号码分配权威机构）来承担这个中心分配机构的功能。IANA 同时承担着 IP 地址的分配、域名注册、协议编号分配、根服务器的运行 4 项互联网络管理任务。随着互联网向全球化、商业化的演变，ICANN（the Internet Corporation for Assigned Names and Numbers）逐步取代了 IANA 的互联网络管理任务。

IANA/ICANN 下设有地址支持组织 ASO（Address Supporting Organization），由区域性互联网络注册机构 RIR（Regional Internet Registry）组成，配合 IANA/ICANN 进行网络地址的分配，目前，ASO 下设 4 个 RIR：即 APNIC（Asia Pacific Network Information Center）、RIPE NCC（Reseaux Internet Protocol Europeans，Network Coordination Center）、ARIN（the American Registry for Internet Numbers）和 LACNIC（Latin American and Carribean Network Information Center），计划不久将增加 AfriNIC（African Network Information Center）。其中 APNIC 负责进行亚太区的地址分配管理。

在 RIR 下还可以依次设下级分配机构，如 RIR 下设国家级互联网注册机构 NIR（National Internet Registry），NIR 下设地区互联网注册机构 LIR（Local Internet Registry）等。地址分配机构组织架构可参考图 A.1。



中国的 ISP 和用户可以通过以下三种方法获得 IP 地址和 AS 号码：

- 向 APNIC 申请成为其会员，直接从 APNIC 获得 IP 地址和 AS 号码。
- 通过已成为 APNIC 会员的 ISP 向 APNIC 申请 IP 地址和 AS 号码。
- 加入 CNNIC 互联网络地址资源分配联盟，通过 CNNIC 向 APNIC 申请 IP 地址和 AS 号码。

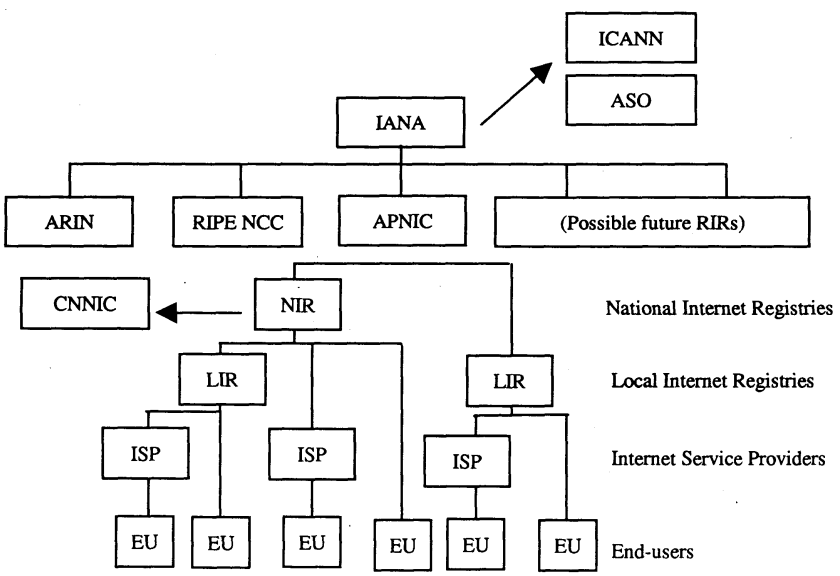



图 A.1 地址分配机构组织架构

## A.2 IPv6 地址等级结构与分配

IPv6 地址长度为 128 位，地址空间非常巨大。整体上，IPv6 分为三类：单播地址、多播地址以及任播地址。其中，单播地址按照地址的传输范围分为可聚类全局单播地址（Aggregatable Global Unicast Addresses）、NSAP 地址（Network Service Access Point）、IPX 层次地址、站点本地地址（Site-Local address）和链路本地地址（link-Local address）等。IPv6 采用长度可变的前缀进行地址空间的分配，便于满足多种类型地址提供者的需要和实现分层结构，从而减少路由器路由表的大小。

依照 RFC 2373 以 001/3 起始的 IPv6 地址划分为全球可聚类单播地址，占整个 IPv6 地址空间的 1/8。这部分地址是当前主要由地址分配机构进行分配的地址。RFC 2374 提出了全球可聚类单播地址的分级地址格式，RFC 3587 中重新定义了 IPv6 地址结构，参见图 A.2。

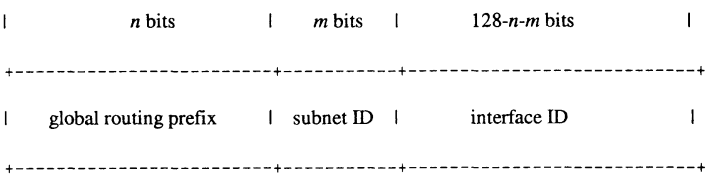



图 A.2 IPv6 地址格式

在 RFC 3177 中，IAB/IESG 对 IPv6 的地址分配方式给出了一些建议：

（1）当用户数目非常大时，建议使用/48 的前缀；

（2）有且只有一个子网存在时，建议使用/64 的前缀；

（3）有且只有一个设备存在时，建议使用/128 的前缀；

（4）对于家乡网络的用户，不论是拨号用户还是专线用户，应获得一个/48 的前缀；

（5）中小型企业应分配一个/48 的前缀；

（6）大型企业可以分配一个/47 或比/47 稍短的前缀，或分配多个/48 的前缀；

（7）在移动网络中，具有无线接口（如蓝牙、802.11b）的移动终端应具备一个静态的/64 的前缀；

（8）在宾馆等公共场所，单个 PC 终端通过拨号上网，应获得一个/128 的 IPv6 地址。



## A.3 IPv6 地址申请政策

需要申请 IPv6 地址的企业，运营商或者其他机构，应向所在地对应的地址分配分支机构提出申请。如前所述，中国的 ISP 或者其他机构可以根据对地址的需求量通过三种途径申请 IPv6 地址。下边将分别简单说明。

（1）直接从 APNIC 申请到/32 的 IPv6 地址块

APNIC 是由 ICANN 授权的亚太地区的地址管理机构。它于 1999 年开始受理 IPV6 地址申请。目前 APNIC 拥有的 IPV6 地址范围为 2001:0200::/23。APNIC 在 2002 年 7 月发布的分配政策中，规定向 APNIC 首次申请地址，需满足以下要求：

- 必须为成员机构，LIR/或者 NIR;
- 不可以是最终用户；
- 有计划向一些组织分配/48 的地址块，并为之提供 IPv6 的连接；
- 有计划在两年内向其他机构或组织分配至少 200 个/48 地址块。

APNIC 地址分配策略如下：

- 首次分配将获得/32 的地址空间；

- 如果技术允许，可以分配更大的空间。

此政策之前，APNIC 曾经以 35/为地址分配的单元空间，并为已分配的 35/地址块保留相应的后续 6 个 bit 的地址空间，为其未来发展作好预留同时保持地址的连续性有助于简化路由等。新的分配政策发布后，持有/35 地址块的机构可以扩展到 32/。

目前的地址分配政策仍属于初始阶段的政策。随着 IPv6 的应用普及，地址分配机构的分配策略将会发生更严格地变化。

（2）通过 APNIC 的成员单位进行申请

目前，中国大陆地区已有大约 6 家企业或者组织向 APNIC 申请到 IPv6 地址。国内机构也可以通过这几家成员单位获得 IPv6 地址。教育科研单位一般可以向 CERNIC 提出申请。目前 CERNIC 向其成员单位分配 /35 的地址空间。

（3）通过 CNNIC 申请 IPv6 地址

CNNIC 对于我国 IPv4 的发展起了重要的作用。目前，CNNIC 可以代理其成员单位，向 APNIC 申请 IPv6 地址。

## A.4 IPv6 地址分配现状

现在在分配和使用中的 16 比特 TLA IPv6 地址主要有三类：

- 由 RIR 分配的商用地址前缀（2001::/16）；
- 6to4 前缀（2002::/16）；
- 6bone 分配的实验地址前缀（3FFE::/16），包括/24、/28 以及/32。



6to4 地址是无需经过申请/分配的，可以自由使用的地址资源。3FFE::/16 地址在 IPv6 前期发展中起到了重要作用，我国 CERNET IPv6 试验床于 1998 年加入 6bone 网络，获得 3FFE:3200::/24 的地址空间。目前，以 2001::/16 为前缀的商用地址空间也获得了大量申请和分配。

据 2004 年 4 月 4 日 APNIC 统计数据，4 个 RIR 已经向全球分配 561 个/32 的地址空间以及 60 个/48 或者/64 的地址用于互联网络交换节点。其中 APNIC、ARIN、LACNIC、RIPE NCC 向所辖区域已分配/32 地址空间数量分别为 147、95、6、313；所分配的互联交换节点地址数量分别为 10、10、0、40。详细信息可以参见 http://www.ripe.net/cgi-bin/ipv6allocs。

我国大陆地区商用 IPv6 地址总数为 9/32+/48；主要分布在 6 家单位，具体分布见表 A.1。实验地址集中在科研教育单位，其中 BII 集团也获得了 3FFE:81B0::/28 的地址空间。此外，台湾地区 IPv6 地址总数为 13/32 /48；香港特区 IPv6 地址总数为/32 /33 /34 2/35 /64。

**表 A.1 中国大陆地区 IPv6 地址分配状况**

| 单位名称 | 地址数量 | 地址空间 |
|---|---|---|
| 中国教育和科研计算机网 | 商用地址：3 个 32/ + 48/<br>试验地址：1 个 24/ | 2001:0250::/32;2001:0DA8::/32;2001:0251::/32+2001:07FA:0005::/48;<br>3FFE:3200::/24 |
| 北京英纳特科技网 | 商用地址：2 个 32/<br>试验地址：1 个 28/ | 2001:03F8::/32; 2001:0D60::/32<br>3FFE:81B0::/28 |
| 中国科技网 | 商用地址：1 个 32/<br>试验地址：1 个 28/ | 2001:0CC0::/32<br>3FFE:8330::/28 |
| 中国电信集团公司 | 商用地址：1 个 32/ | 2001:0C68::/32 |
| 铁道通信信息有限责任公司 | 商用地址：1 个 32/ | 2001:0E08/32 |
| 中国网络通信集团公司 | 商用地址：1 个 32/ | 2001:0E18::/32 |

# 附 录 B
## （资料性附录）
## 运营商 IPv4 向 IPv6 过渡与互通方案建议

### B.1 过渡的内容

网络厂商和开发者逐渐将 IPv6 引入不同的平台，网络运营者逐步确定自己的 IPv6 网络功能，向 IPv6 过渡也将是一个相对缓慢的过程，预计 IPv4 和 IPv6 将长期共存。在现有的 IPv4 网络上逐步融合 IPv6，从运营商的角度首先需要明确过渡的内涵，即过渡内容包括：

（1）网络的过渡；

（2）用户（主机）的过渡；

（3）应用程序的过渡；

（4）IPv4/IPv6 网络互通；

（5）IPv6 网络之间的互通。

本标准主要针对网络的过渡与互通问题。

### B.2 过渡方案与要求

从 IPv4 到 IPv6 的过渡必然是一个平滑的渐进过程，根据 IPv4 网络与 IPv6 网络的共存状况，此过程可以划分为以下三个阶段。

（1）初始阶段：引入 IPv6 Intranet，提供 IPv6 接入等服务。IPv4 网络中出现若干个 IPv6 孤岛，不同的 IPv6 孤岛采用自动或人工配置的隧道通过 IPv4 网络连接起来。此阶段主要为以后积累组网经验并从技术层面探索开发基于 IPv6 的可运营业务模式。

（2）共存阶段：IPv6 得到较大规模的应用，出现了骨干的 IPv6 Internet 网络，在 IPv6 平台上引入了大量的业务。IPv6 业务可以通过 IPv6 Internet 网络与 IPv6 Intranet 网络，从而可以充分利用 IPv6 的诸多优势，如 QoS 保证。但由于 IPv6 网络之间有可能不是相互联通的，因此还会采用隧道。在 IPv6 平台上实现丰富的业务加快了 IPv6 的实施。但仍将有大量的传统 IPv4 业务存在，许多节点也仍然是双栈节点。

（3）主导阶段：IPv6 占据主导地位，具备全球范围内的连通性，所有的业务都运行在 IPv6 平台上。网络结构得以简化，维护也更加容易。

本附录中论述主要针对初始阶段与共存阶段。过渡的初始阶段与共存阶段，对于网络的过渡和互通要求如下。

#### B.2.1 网络过渡

网络过渡的内容主要涉及到设备升级、地址规划、互联互通方案的选择。

（1）骨干网络

a）设备要求： 路由器应具有双栈能力以及 IPv6 的硬件转发能力。

b）地址分配： 各运营商自行制定地址分配方案。

c）路由：所有骨干网络中的路由器应具有 IPv4 可达性；核心节点之间可暂时采用静态隧道，当流量到达一定程度时，IPv6 路由采用纯 IPv6 链路。

内部网关协议采用 OSPFv3 或者 IS-ISv6;

边界网关协议采用静态路由或者 BGP4+;

对外广播路由之前应进行路由汇聚；

IPv6 流量工程（过渡中后期实施）。

d）网管：带外监控，SNMP。

e）适合的过渡技术：双栈、静态配置隧道、BGP/MPLS 隧道。

（2）接入网络

a）设备要求：接入服务器应具有双栈协议以及相应的支持 IPv6 的各种接入服务器、网络功能等，如 PPPv6 等。

b）地址分配：接入服务器由于只分配 IPv6 地址块，可以不需要每次都动态分配地址前缀。（待讨论）

c）路由：运营范围内，一般采用静态路由，应同时具有 IPv4 与 IPv6 的路由可达性。

d）网管： 带内方式和带外方式。

e）适合的过渡技术：隧道代理、NAT-PT、6TO4、静态隧道等。

### B.2.2 IPv6 网络互通方案

（1）手工配置隧道

用于连接两个被 IPv4 网络分割的 IPv6 网络，用于网络间流量比较稳定的情况。在 IPv6 网络建设的初期阶段，当两个网络节点之间的流量较小时，且需要配置的隧道数量较少时，则手工配置隧道具有实际意义。目前世界上几乎所有的 IPv6 网络（包括 6bone 主干）都采用了手工隧道。随着过渡过程的深入，这种隧道连接以后可能被专线连接所取代。

（2）兼容地址的自动隧道

这种隧道的建立和拆除是动态的，它的端点根据数据包的目的地址来确定，适用于单独的主机之间或不经常通信的站点之间。这些站点之间必须有可用的 IPv4 连接。这种隧道的两个端点都必须支持双栈。在隧道要经过 NAT 设施的情况下这种机制不可用。

（3）隧道代理

隧道代理非常适合于独立的小型 IPv6 站点，特别是独立的分布在 IPv4 互联网中的 IPv6 主机需要连接到已有的 IPv6 网的情况。

（4）6to4 隧道

这种隧道的目的是使分布在 IPv4 网络中的 IPv6 网络实现互联，虽然采用了特殊的地址前缀，但对于互联纯 IPv6 网络的 IPv4 地址消耗很少，一个网络只需要一个公有的 IPv4 地址，对于地址缺乏的环境值得推广。

（5）6over4

6over4 隧道要求 IPv4 网络对组播的支持能力，因而也限制了该技术的应用范围，对于运营商的网络过渡方案中，不推荐采纳。

（6）BGP 隧道

适合于域间互联的解决方案，对于骨干网络的设备应该考虑支持 BGP 隧道功能。

（7）ISATAP 隧道

适用于企业等 Intranet 的情况，对于运营网络不做要求。

在 IPv4-IPv6 过渡时期，跨越 IPv4 网络进行 IPv6 网络之间的互通，骨干网可选手工配置隧道、MPLS/BGP 隧道等方案；接入网络适用手工隧道、隧道代理、6to4 隧道等方案。

### B.2.3 IPv4/IPv6 网络互通方案

（1）双栈

单独采用双栈的方案不适合 IPv4/IPv6 网络之间的互通，双栈与其他过渡技术结合使用。

（2）SIIT

仅是一种翻译算法，需要结合 IPv4 地址分配机制使用，不适合骨干网络。

（3）NAT-PT

适合于纯 IPv6 网络与 IPv4 网络之间的互通，对于 IPv4/IPv6 网络互通是一个较完整的解决方案。

（4）DSTM

适合于 IPv6 边缘网络与 IPv4 网络的互通，对于骨干设备不需要支持。

（5）BIS

主机内的过渡方案，这里不作要求。

（6）BIA

主机内的过渡方案，这里不作要求。

（7）TRT

待讨论。

在 IPv4-IPv6 过渡时期,对于运营网络中适用的 IPv4/IPv6 互通方案主要可选 NAT-PT、TRT 与 DSTM。

# 附 录 C
## （资料性附录）
## 主要电信设备厂商在 IPv6 领域的发展

从 20 世纪 90 年代起，IPv6 协议日趋成熟，越来越多实验与研究在国内外研究机构开展。2000 年 1 月，由 HITACHI 设计制造的 IPv4/IPv6 双栈路由器 GR2000 系列成为全世界第一台硬件转发支持 IPv6 的商用路由器。近一年后，各知名设备制造商如 Nokia、NEC、Cisco、Juniper 等纷纷推出了支持 IPv6 的路由器等设备。国内设备制造商以华为、中兴为首也相继推出双栈路由器。下面简单介绍部分厂商设备对 IPv6 的支持情况。

### C.1 Cisco

Cisco 的 IOS 对 IPv6 各功能的支持（Cisco IOS IPv6）分为三个阶段，见表 C.1。

表 C.1 Cisco IOS IPv6 Roadmap

| Cisco IOS Release | 市场定位 |
|---|---|
| Phase I / Release 12.2（2）T / 已完成 | 早期部署应用 |
| Phase II / 已完成 | 骨干网络部署应用 |
| Phase III / 进行中 | IPv6 高级业务 |

其中，2001 年 5 月 Cisco IOS 12.2（2）T 的发布标志着 Cisco 完成了第一阶段的目标，12.2（2）T 也是 Cisco 第一个支持 IPv6 的 IOS 版本。表 C.2 列出了第一阶段 Cisco IOS IPv6 支持的 IPv6 功能特性。第一阶段的 IOS 所适用的硬件平台比较广泛，包括从 800 系列到 12000 系列的产品。

第二阶段的目标是针对骨干网络的产品，增加了快速转发，IS-ISv6 与 IPv6 MPLS 等功能特性。表 C.3 给出了第二阶段要实现的功能集，这部分工作也已经实现。

Cisco 现在正在进行第三阶段的功能实现，第三阶段主要这对增强型的 IPv6 功能集，包括 QoS、组播、OSPFv3 的支持等。

Cisco 相信，随着将来 IPv6 流量的大量增长，多协议标记交换技术（MPLS）将成为 ISP 转向 IPv6 的一个最佳选择。这是因为它的开发成本更低、VPN 配置更简单，而且还具有流量管理的能力。自 2001 年中期起，一半以上已配置了 Cisco12000 Internet 路由器的客户正在考虑或者已经配置了 MPLS。

利用 MPLS，客户就能更容易地扩展流量。由于 MPLS 路由器交换数据包是基于标记而不是基于地址查找的，因而带给硬件的负担就更小。利用 MPLS 进行转发并利用 IPv6 来传递应用，ISP 就能在 MPLS 路径上发送 IPv6。这种方法将使 ISP 得到与如今的 ASIC 交换相似的强大性能，却无须对硬件基础设施进行成本高昂的升级。

Cisco 设计了一种称为 6PE（IPv6 Provider Edge Router over MPLS）的特性来支持 MPLS，这种特性将允许服务提供商在任何时间、任何地点将 IPv6 与 MPLS 网络相结合。目前 Cisco 设备已经支持 6PE 方案。

表 C.2 第一阶段 Cisco IOS IPv6 支持的 IPv6 功能特性一览

| Feature name | RFC | IOS | | |
|---|---|---|---|---|
| IPv6 Services | | 12.2T/12.3M | 12.0S/ST（*） | 12.2 S |
| Internet Protocol version 6 | RFC 2460 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 Addressing Architecture | RFC 2373 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| ICMPv6 | RFC 2463 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |

表C.2（续）

| Feature name | RFC | IOS | | |
|---|---|---|---|---|
| Neigbour Discovery | RFC 2461 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 Stateless Auto-configuration | RFC 2462 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| MTU Path Discovery for IPv6 | RFC 1981 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| ICMPv6 Redirect | RFC 2463 | 12.2（4）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 Duplicate Address Detection | | 12.2（4）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 Standard Access Control List（ACL） | | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| Manual Configured Tunnel | RFC 2893 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| Automatic IPv4 Compatible Tunnels | RFC 2893 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| 6to4 Tunnels | RFC 3056 | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 over IPv4 GRE Tunnels | | 12.2（4）T | N/A | 12.2（14）S |
| IPv6 Routing | | | | |
| Static Routes | N/A | 12.2（2）T | | 12.2（14）S |
| RIPng | RFC 2080 | 12.2（2）T | | 12.2（14）S |
| MP-BGP4 | RFC 2545 and 2858 | 12.2（2）T | | 12.2（14）S |
| Link-local address to do MP-BGP4 peering | | 12.2（4）T | ? | ? |
| Encapsulation | | | | |
| Loopback | N/A | 12.2（2）T | | |
| Ethernet 10 Mb/s | RFC 2464 | 12.2（2）T | | |
| Ethernet 100 Mb/s | RFC 2464 | 12.2（2）T | | |
| Ethernet 1000 Mb/s | RFC 2464 | 12.2（2）T | | |
| IPv6 over ISL | N/A | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 over IEEE 802.1Q | N/A | 12.2（2）T | | |
| FDDI | RFC 2467 | 12.2（2）T | | |
| ATM PVC | RFC 2492 | 12.2（2）T | | |
| ATM Ethernet LAN-E Using packet format | RFC 2464 | 12.2（2）T | | |
| Cisco HDLC | N/A | 12.2（2）T | | |
| PPP | RFC 2472 | 12.2（2）T | | |
| Frame Relay PVC | RFC 2590 | 12.2（2）T | | |
| Switching | | | | |
| Process Switched | | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| IPv6 Management & Applications | | | | |
| Ping | | 12.2（2）T | 12.0（22）S/（21）ST | 12.2（14）S |
| Traceroute | | | | |
| Telnet | | | | |
| TFTP | | | | |
| DNS Client AAAA record over IPv4 transport | RFC 1886 | | | |



表 C.3　第二阶段 Cisco IOS IPv6 所支持的功能集一览

| 功能特性 | 12.2T/12.3M | 12.0S/ST（*） | 12.2 S |
|---|---|---|---|
| IS-IS FOR IPv6 | 12.2（8）T | 12.0（22）S/（21）ST | 12.2（14）S |
| CEFv6/dCEFv6 | 12.2（13）T | 12.0（22）S/（21）ST | 12.2（14）S |
| 扩展访问控制列表 | 12.2（13）T | 12.0（23）S | 12.2（14）S |
| IPv6 over MPLS 6PE | 12.2（15）T | 12.0（22）S | 12.2（14）S |

表 C.3（续）

| 功能特性 | 12.2T/12.3M | 12.0S/ST（*） | 12.2 S |
|---|---|---|---|
| NAT-PT（RFC 2766） | 12.2（13）T | N/A | TBD |
| IPv6 MIBs | 12.2（15）T | 12.0（22）S | 12.2（14）S |
| IPv6 CDP | 12.2（8）T | N/A | 12.2（14）S |
| 静态 ND Cache 记录 | 12.2（8）T | 12.0（22）S/（21）ST | 12.2（14）S |
| 链路本地的 BGP4+ | 12.2（4）T | 12.0（22）S/（21）ST | 12.2（14）S |
| 宽带接入（封装，AAA，前缀池） | 12.2（13）T | N/A | TBD |
| DNS AAAA over IPv6 | 12.2（8）T | 12.0（22）S/（21）ST | 12.2（14）S |
| SSH over IPv6 | 12.2（8）T | 12.0（22）S | 12.2（14）S |

## C.2 HITACHI

日立公司一直致力于 IP 网络设备的研究开发，特别是在下一代网络技术——IPv6 领域。日立从 IPv6 标准化初期就积极进行研发和推进产品化。

- 早在 1997 年日立就研制成功了世界上最早的支持 IPv6 的路由器 NR60；
- 在 2001 年，日立推出了世界上最早的硬件支持 IPv6 千兆极路由器 GR2000 系列和世界上最早的 IPv6 宽带接入服务器 AG8100 系列；
- GR2000 被日本吉比特网（JGN）、NTT 公司、欧洲 Skanova 公司（瑞典）等采用。在全球范围内已经销售了 8000 多台，其中被作为 IPv6 路由器使用的就有 600 台以上，在全球 IPv6 市场所占份额最大；
- 在 2003 年，日立推出了 GR4000 系列 IPv6 系统容量达到 320Gbit/s 高可靠性核心路由器产品。

在日立 IPv6 产品中，以 AG8100 为接入产品的代表，GR2000 为骨干路由器的代表。

（1）IPv6 宽带接入服务器 AG8100

AG8100 提供宽带接入服务，目前已通过信息产业部入网测试，应用于湖南电信 IPv6 试验网，6TNet 网络的主要功能特点如下：

- 基于网络处理器（NP）的高速性和灵活性，实现硬件高速转发；
- 高吞吐量——利用最大带宽；
- 超小型机壳（2U）内容纳 6 条千兆比特以太和 16 条快速以太接口；
- 支持 ADSL、LAN、WLAN、FTTH、CMTS 等多种宽带接入技术；
- 支持 PPPoE、PPPoA、PPPoEoA；
- 支持标准的 RADIUS 协议（IPv6 接入需要 RADIUS 协议支持 IPv6 扩展属性）；
- 支持按照时长、流量、包月等多种计费方式，资费灵活，计费准确；
- IPv6（PPP for IPv6、DHCPv6 PD）；
- L2TP 服务器重定向功能。

（2）GR2000

日立 GR2000 系列路由器采用控制与业务相分离的分布式架构设计，将路由计算等控制与高速转发分离，保证设备的高性能。

GR2000 路由器由交换矩阵、路由管理模块 RM、高速转发模块 RP 组成。RM 负责路由计算、管理等控制功能，将路由表下发给 RP 模块，RP 根据转发表进行高速路由表查找和数据转发。RM 和 RP 通过高速交换矩阵连接起来。

每个 RP 都设置有专用 ASIC 芯片，实现基于硬件处理的 IPv4/IPv6 线速转发、基于硬件处理的 QoS 控制和 DiffServ 能力、基于硬件处理的 IP 过滤、基于硬件处理的 IP 组播（日立第一个实现 IPv6 组播）、基于硬件处理的 IP 多路传送（负载均衡），满足电信级骨干网路由器的高性能需求。

## C.3 FUJISTU

GeoStreamR920和GeoStreamR980是富士通支持IPv6 技术的运营级核心路由器产品，分别提供20Gbit/s和80~160Gbit/s的系统容量。

GeoStreamR920和GeoStreamR980通过采用全线速的交换矩阵（cross-bar switch），可以实现最大20/80Gbit/s（即将扩充到106Gbit/s）的吞吐能力。它们提供高速的WAN接口，如ATM、POS在每个层二和层三通过基于MPLS的标签交换，集成了VPN（BGP/BGP/MPLS-VPN和VLAN VPN mapped MPSL），为将来IP流量大规模的增加提供了足够的扩展能力。基于在开发和制造运营级的WAN设备和交换节点设备方面的经验，GeoStream R900系列被设计成拥有全冗余配置和在线配置更改的功能，因此提供了更高的可靠性和实用性。

GeoStream R900系列中先进的IP 技术使得它能够支持IPv6、IP组播、宽带远程接入服务（BRAS）以及在下一代IP网络中的先进技术，如移动通信技术等。因此，GeoStreamR900系列可以实现和支持高质量的网络，应用在不同的场合，如在运营网络的边缘节点或公众网或企业间通信的IP骨干网交换节点。

GeoStream R900系列不仅提供IPv4，还能够提供IPv6的硬件路由。IPv6也同样可以实现全线速传送。同时支持Diffserv，可以在用户网内进行话务的差别化传送、带宽管理等优先控制。此外，通过IP组播，能对录像等动画内容的流信息进行高效率的传送。表C.4列出了GeoStreamR900系列的基本功能，表C.5是GeoStreamR900系列的性能描述。



表 C.4 GeoStreamR900 系列的基本功能

| | IPv4 | IPv6 |
|---|---|---|
| 基本功能 | 硬件路由、过滤、ICMPv4 | 硬件路由、过滤、ICMPv6 |
| 路由 | 静态、RIPv1/RIPv2、OSPFv2、BGP4 | 静态、RIPng、OSPFv3、BGP4+ |
| MPLS | LDP、RSVP-TE | |
| 多播 | IGMP、PIM-SM | |
| 流量控制 | DIFFSERV（EF/AF/CS/BE） | DIFFSERV（EF/AF/CS/BE） |
| IPv6 过渡措施 | | 双栈、IPv6 over IPv4 隧道 |
| 高级功能 | Bras、IP-VPN、VLAN VPN mapped MPLS | |
| 其他 | Ping、Traceroute、telnet、rlogin、ftp | Ping6 Traceroute6、telnet、rlogin、ftp，自动地址设置 |

表 C.5 GeoStreamR900 系列的性能描述

| | | R980 | | R920 | |
|---|---|---|---|---|---|
| 系统性能 | | 96Mpps（80Gbit/s） | | 24Mpps（24Gbit/s） | |
| | | 192Mpps（160Gbit/s） | | | |
| 路由表容量 | | | | | |
| IPv6 | 静态路由 | 10000 | IPv4 | 静态路由 | 65000 |
| | RIPng | 10000 | | RIPng | 10000 |
| | OSPFv3 | 10000 | | OSPFv3 | 10000 |
| | ISISv6 | FUTURE PLAN | | ISISv6 | 20000 |
| | BGP4+ | 120000 | | BGP4+ | 240000 |

富士通的产品参加了国内外多个 IPv6 试验网的组建，包括日本的 G 比特网络（Japan Gigabit

Network）、IPv6 中日合作项目、东京 CATV IPv6 试验网等。图 C.1 为应用富士通产品组建的 JGN 网络示意。

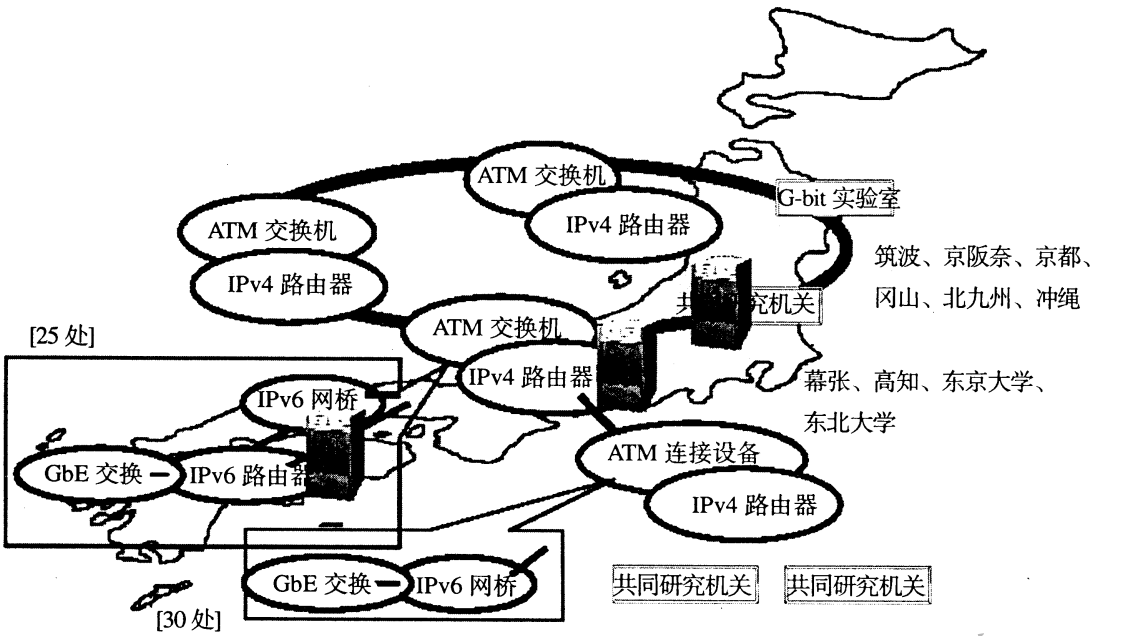



图 C.1 JGN 网络示意

## C.4 Nokia

Nokia 很早就活跃在 IPv6 领域，这与 Nokia 的发展策略不无关系，归结为技术原因就是 3G 的发展与 IP 的密切关系和对 IP 移动性的要求。Nokia 首先向 3GPP 建议将 IPv6 作为未来多媒体业务的核心协议。目前，3G 的标准也已经确定了 IPv6 作为 3G 核心网络的标准协议。Nokia 也是第一家宣布实现基于 IPv6 的 3G 全 IP 核心网制造商。就 IP 设备而言，Nokia 的优势在于对 MIPv6 的支持和移动数据网络设备。

Nokia 是在移动 Internet 中使用 IPv6 的倡导者。对 IPv6 的热衷是出于 3G 对 IPv6 的需求，其关于 MIPv6 的观点将对 IPv6、3G 的标准化工作和实现产生重要的影响。Nokia 积极推动在中国的 IPv6 研发工作，包括与中国的高校、研究所和 CERNET 的合作。

Nokia 路由器产品的软件平台为 IPSO，它是 Nokia IPRG 工程组开发的路由器原型软件，自 3.3 版之后，开始支持 IPv6。目前，最新版本为 IPSO 3.4。IPSO 运行在 Nokia 的 IP 110、IP 330、IP 440 和 IP 650 等产品上，Nokia 的 3G 产品，包括 GPRS、SGSN、GGSN 也以其为平台。

IPSO 支持的主要的功能特性包括：

（1）核心功能

- IPv6（RFC 2460）；
- ICMPv6（RFC 2463）；
- 邻居发现（RFC 2461）；
- IPv6 TCP support；
- IPv6 over IPv4隧道；
- IPv6链路层承载：

IPv6 over Ethernet（RFC 2464）；

IPv6 over FDDI（RFC 2467）；

IPv6 over PPP（RFC 2472）；

IPv6 over ATM（RFC 2492）；

IPv6 over ARCNET（RFC 2497）；

IPv6 over Token Ring（RFC 2470）。

- IPv4/IPv6互通

6over4（RFC 2529）；

6to4（RFC 3056）；

通用分组隧道（Generic Packet Tunneling，RFC 2473）；

基本的IPv6 Socket端口（RFC 2553）。

（2）路由功能

- 路由器发现—邻居发现的一个子集；
- RIPng for IPv6；
- 静态路由；
- 路由汇聚；
- 路由重定向。

（3）应用相关

- IPv6 inetd；
- IPv6 telnet—客户端和服务器；
- IPv6 ftp—客户端和服务器；
- ping、netstat、tcpdump、ndp；
- NMS—Voyager web configuration。



诺基亚公司作为CERNET IPv6 Testbed的合作伙伴也在积极开展IPv6的研究。目前试验床正式使用部分的八大地区网络中心采用的路由器是诺基亚公司捐赠的IP650。诺基亚公司的IPv6站点有两个：http://www.internet6.com.cn；http://www.ipv6.com.cn。

## C.5 华为

华为从1996年就对IPv6进行跟踪预研。目前，华为、华为3Com的平台及设备对IPv6有着良好的支持：其网络操作系统台VRP5已全面支持IPv6；全线路由器产品均已支持IPv6特性，以太网交换机产品也将在2004年下半年全面支持IPv6。其中NetEngine 16E/08E/05系列高端路由器是国内首家通过产品入网检测并在实验网上实际运行的IPv6路由器。

在Quidway NetEngine系列高端产品大规模商用的背景下，华为公司已经与中国科学院一起组建了全国产化的IPv6网络测试、网络监控、Web、视频、VoIP、E-mail等多种IPv6业务的流量，取得了良好的应用效果。华为的IPv6产品及解决方案还在重庆网通实验网、华为公司内部实验网得到了成功应用。华为IPv6硬件产品见表C.6。

VRPv5 是华为最新的通用路由平台，提供了对 IPv6 全方面的支持。基于该平台，华为公司从高端到低端的全系列路由器产品可以实现面向 IPv6 的平滑升级。华为 VRPv5 平台对 IPv6 的特性支持 1 参见表 C.7。

表 C.6　华为 IPv6 硬件产品表

| | 产品 | 华为 VRP 平台 |
|---|---|---|
| 中低端路由器系列 | | |
| | Quidway 2600 系列 | VRPv5 R001 |
| | 3600 系列 | VRPv5 R001 |
| | 4600 系列 | VRPv5 R001 |
| | NE05 系列 | VRPv5 R001 |
| | NE08 系列 | VRPv5 R001 |
| | NE16 系列 | VRPv5 R001 |
| 中高端路由器系列 | | |
| | NE20 系列 | VRPv5 R001 |
| | NE40 系列 | VRPv5 R001 |
| | NE80 系列 | VRPv5 R001 |
| | NE5000 系列 | VRPv5 R001 |

表 C.7　VRPv5 平台支持 IPv6 的特性表

| | IPv6 特性 | VRPv5R001 | VRPv5R002 |
|---|---|---|---|
| IPv6 协议 | | | |
| | IPv6 协议栈 | 支持 | 支持 |
| | IPv6 单播地址 | 支持 | 支持 |
| | IPv6 任播地址 | | 支持 |
| | IPv6 ICMPv6 | 支持 | 支持 |
| | IPv6 ICMPv6 Redirect | 支持 | 支持 |
| | IPv6 ICMP Error Limit | 支持 | 支持 |
| | IPv6 邻居发现 | 支持 | 支持 |
| | IPv6 反向邻居发现（Inverse ND） | | 支持 |
| | IPv6 静态邻居缓存 | 支持 | 支持 |
| | IPv6 无状态自动配置 | 支持 | 支持 |
| | IPv6 Path MTU 发现 | 支持 | 支持 |
| | IPv6 软件高速缓存转发 | 支持 | 支持 |
| | IPv6 ND Proxy | | 支持 |
| IPv6 链路层 | | | |
| | IPv6 Over Ethernet | 支持 | 支持 |
| | IPv6 Over PPP， | 支持 | 支持 |
| | IPv6 Over Vlan | 支持 | 支持 |
| | IPv6 Over FrameRelay | | 支持 |
| | IPv6 Over ATM | | 支持 |
| | IPv6 Over GE | 支持 | 支持 |
| | IPv6 Over HDLC | | 支持 |
| | IPv6 Over Pos | 支持 | 支持 |
| IPv6 路由 | | | |
| | IPv6 静态路由 | 支持 | 支持 |
| | IPv6 RIPng | 支持 | 支持 |
| | IPv6 ISIS | 支持 | 支持 |

表 C.7（续）

| | IPv6 特性 | VRPv5R001 | VRPv5R002 |
|---|---|---|---|
| | IPv6 OSPFv3 | | 支持 |
| | IPv6 BGP | 支持 | 支持 |
| | IPv6 路由策略 | 支持 | 支持 |
| IPv6 安全 | | | |
| | IPv6 基本 ACL | 支持 | 支持 |
| | IPv6 扩展 ACL | 支持 | 支持 |
| | IPv6 包过滤防火墙 | 支持 | 支持 |
| | IPv6 IPSec | | 支持 |
| | IPv6 IKE | | 支持 |
| IPv6 应用 | | | |
| | IPv6 Ping & Traceroute | 支持 | |
| | IPv6 TFTP | 支持 | |
| | IPv6 Telnet Client & Server | 支持 | |
| | IPv6 DNS Client | 支持 | |
| | IPv6 FTP | | 支持 |
| | IPv6 DHCPv6 | | 支持 |
| | IPv6 SNMP | | 支持 |
| | IPv6 SSH | | 支持 |
| | Mobile IPv6 通用支持 | | 支持 |
| | Mobile IPv6 家乡代理支持 | | 支持 |
| | IPV6 组播 MLDv2 | | 支持 |
| | IPV6 组播 PIM | | 支持 |
| IPv6 MIB | | | |
| | IPv6 MIB | 支持 | 支持 |
| | ICMPv6 MIB | 支持 | 支持 |
| | IPv6 TCP MIB | 支持 | 支持 |
| | IPv6 UDP MIB | 支持 | 支持 |
| IPv6 过渡技术 | | | |
| | 双栈 | 支持 | 支持 |
| | IPv6 over IPv4 手工隧道 | 支持 | 支持 |
| | IPv6 自动兼容隧道 | 支持 | 支持 |
| | IPv6 6to4 隧道 | 支持 | 支持 |
| | IPv6 6to4 中继隧道 | 支持 | 支持 |
| | IPv6 ISATAP 隧道 | 支持 | 支持 |
| | IPv6 over IPv4 GRE 隧道 | 支持 | 支持 |
| | NAT-PT | 支持 | 支持 |
| | NAT-PT ICMP-ALG | 支持 | 支持 |
| | NAT-PT DNS-ALG | 支持 | 支持 |
| | NAT-PT FTP-ALG | 支持 | 支持 |
| | NAT-PT 端口转换 | 支持 | 支持 |
| | NAT-PT 分片报文转换 | 支持 | 支持 |
| | IPv6 Over MPLS（6PE） | 支持 | 支持 |

## C.6 中兴通讯

中兴通讯从2000年开始进行IPv6领域的研究、开发。2002年成功承担国家“863”项目“高性能IPv6路由器平台系统”的工作，并在此基础上将IPv6全部功能成功集成到数据产品操作系统平台ZXROS中。ZXROS操作系统平台的成功推出使中兴通讯全系列路由器、交换机等产品全部支持IPv6/IPv4功能，从而可以通过ZXR10系列路由器和交换机设备来构建完善的IPv6网络解决方案。

中兴通讯基于ZXROS操作系统平台的全系列路由器和交换机设备均支持完善的IPv6功能，可以用于下一代IPv6网络的构建。设备主要包括：

（1）路由器

ZXR10 T128/64E核心/汇聚路由器：ZXR10 T128是中兴通讯自主研发的电信级核心路由器产品，产品采用了模块化、分布式和可扩展的设计理念和先进的CROSSBAR交换结构，系统具有良好的可扩展性，交换容量可达640/320Gbit/s；支持10GPOS、10GE、2.5G、GE、ATM、FE等丰富接口类型；采用先进的网络处理器技术，可以方便地实现对各种新业务的快速支持；支持基于MPLS技术的流量工程和VPN业务；支持基于硬件的线速NAT，支持组播业务；支持策略路由、负荷分担、统计等功能；支持完善的QoS机制。

ZXR10 GER汇聚路由器：根据对设备能力的不同需求，交换能力从8 Gbit/s到32 Gbit/s可灵活配置。支持2.5GPOS、GE/FE、622Mbit/s、155Mbit/s POS等高速接口以及E1等中低速接口；系统硬件具有QoS、组播和IPSec的支持能力，除了具有通用路由器的功能外，还有较强的VPN、ACL、MPLS功能；在网络的安全性方面，硬软件防火墙技术、防DOS攻击等功能突出。

ZXR10 GAR边缘/接入路由器：采用了模块化设计思想，通过灵活的硬件、软件配置和版本升级，可形成三款独立的路由器；设备支持的接口速率从低速64kbit/s到高速1000Mbit/s，提供V.35/V.24、信道化和非信道化E1、同步/异步串口、10/100Mbit/s、信道化和非信道化POS 155Mbit/s和GE等接入；支持VoIP业务的实现；支持MPLS，使得流量工程、QoS及L2/3VPN服务能顺利实施。

（2）交换机

ZXR10 T64C－XG/T160C－XG是中兴通讯自主研发的新一代大容量、高性能核心万兆MPLS路由交换机产品，交换容量为576/1152Gbit/s，包转发速率为360Mpps/576Mpps。ZXR10T64C－XG/T160C－XG万兆MPLS路由交换机基于先进的模块化理念进行设计，采用了基于多处理器并行处理机制和Crossbar空分交换的体系结构，关键模块均可以1：1冗余备份。ZXR10T64C－XG具备10GE、GE、FE、POS和ATM等各种丰富的接口模块，并全面支持MPLS、NAT、组播、QoS、带宽控制等业务功能。主要定位在运营商IP城域网、校园网、电子政务网和大型企事业网等网络的核心层。

ZXR10 16C/T32C/64CMPLS接入/汇聚/核心路由交换机提供完全基于硬件的L2/L3/L4线路路由交换能力，整机具有64/128Gbit/s的系统交换容量以及48Mpps/96Mpps的线速L3包转发能力。系统集成了丰富的业务接口类型和业务功能，可以将IP网络的QoS服务延伸到城域网的边缘接入层，使运营商能够充分利用城域网资源经营各种基本业务和增值服务。ZXR10 T32C/64C提供8/16个插槽，支持各种传输技术的接口模块。这一灵活性使运营商能够充分融合现有的TDM/ATM网络基础资源，逐渐统一到宽带IP城域网以及MPLS VPN的架构之中。

作为较早介入IPv6技术研究和产品开发的国内设备商，中兴通讯已经掌握了核心的IPv6及其相关技术并在2003年的IPv6高峰论坛上展出了IPv6双栈路由器。中兴通讯双栈路由器在中国科学院组织的IPv6

试验中成功实现了与国际上其他主流厂商 IPv6 设备的互通和应用。目前中兴通讯在 IPv6 设备研发方面已经取得长足的进步和丰硕的成果，可以为我国即将开展的 IPv6 网络建设起到重要的作用，保证我国在下一代网络技术上处于国际领先地位。

# 附 录 D
## （资料性附录）
## 国内外 IPv6 发展状况

从 1994 年在 IETF 的会议上被推荐讨论至今不足 10 年的时间里，IPv6 作为下一代网络协议，无论从协议完善还是应用上都有了巨大的发展。由于 IPv6 技术的发展与 IPv4 资源分配状况息息相关，因此也就决定了 IPv6 技术在世界各地发展的不平衡状况。就实验网络的开展，以欧洲和日本最为积极，而中国的相关实验网项目近年来也相继铺开。从 6Bone 开始到今天各地的运营试验网络，IPv6 正在从研究试验走向运营试验。

### D.1 国外 IPv6 发展简介

作为 Internet 的发源地，美国拥有全球 70%IPv4 的地址，几乎平均每个美国公民就拥有 10 个地址，他们几乎感觉不到地址匮乏的压力，因此美国对 IPv6 的发展一直都采取比较保守的态度；只是近年来，随着其他国家地区对 IPv6 的研究和应用的升温，美国的一些制造商和开发商开始注意到其中蕴涵的巨大商机而开始致力于 IPv6 相关的研发工作。Cisco 已经从 800 系列到 7500 系列的 12.2T 版本开始，对 IPv6 各种协议和服务提供支持。

相比之下，欧洲各国家和地区对待 IPv6 的态度都比较积极。欧洲发展高性能互联网通信技术的预算已经提交给欧洲部长会议审议表决。欧委会迄今已经提供5500万欧元的资金，其中部分资金用于两个IPv6的试验项目。欧委会希望欧洲国家尽早采用 IPv6。

欧洲国家对 IPv6 的重视根本的原因在于：欧洲在移动通信领域已经掌握了先机，即将到来的 3G 时代更让他们看到了在未来网络经济中与美国并驾齐驱的希望，要将这一希望变为现实，IPv6 统治地位的早日确立就成为关键中的关键。专家认为 3G 将在两三年内步入实际应用阶段，为了抓住这一发展的契机，欧洲的各大厂商和运营商都对 IPv6 寄予了厚望并竭尽全力对它进行推广和研究，如诺基亚、爱立信、BT 等公司一直都是 IPv6 研究方向的主要引导者。

为了推动 IPv6 的发展，欧洲设立了许多研究 IPv6 的项目，且在国际上参与了不少合作研究项目，如韩国的 TEIN、日本的 NTTv6net、美国的 Internet2 和 Esnet、世界范围的 6bone 等。

目前亚洲 IPv6 发展最领先的国家是日本，日本是全球拥有 IPv6 地址最多的国家。ICANN 于 1999 年 7 月 14 日开始正式分配 IPv6 地址。截至 2003 年 2 月 26 日，商用 IPv6 地址的获取情况为：APNIC（亚洲/大洋洲）分配了 98 个；ARIN（北美/南美）分配了 49 个；RIPE NCC（欧洲）分配了 57 个。位居亚洲的日本获得了 50 个 IPv6 地址，按国别是最多的。在日本，NTT Com、Japan Telecom 和 KDDI 等主要运营商现在几乎都提供了 IPv6 业务，它们于 1999 年 12 月开始提供试验服务，2001 年 4 月开始提供商用服务。此外，IIJ、BIGLOBE、@nifty 等大型 ISP 也已提供 IPv6 业务。1999 年 8 月有线电视运营商、ADSL 运营商开始提供试验服务。有些企业如电通国际信息服务公司也于 2001 年 3 月在内部网上全面实现了 IPv6。Global.Centre.Japan 也于 2001 年 3 月建立了 IPv6 数据中心，开始提供试运行服务。此外，日本的厂商像日立、富士通、NEC 公司已经推出了基于硬件转发的 IPv6 路由器，IPv6 之所以在日本得到重视和发展，是和日本的信息化国策紧密相关的。

KDDI 于 2003 年 2 月 17 日宣布，利用“IPv6 over IPv4 隧道技术”，从即日起开始在 KDDI 的 IPv4 互联网上提供能传送 IPv6 数据包的“IPv6 隧道”服务。该服务将作为 KDDI 正在提供的现有互联网接入

服务的附加服务来提供给客户。预计利用该服务的客户为领先导入 IPv6 网络的家电制造商，以及开发 IPv6 网络设备和软件的供应商。附加服务的使用费按照所采用的互联网接入速度的不同，接入速度越快费用越贵。例如，当 64kbit/s 的“标准 2”用户使用该附加服务时，每月使用费为 2000 日元（约合人民币 125 元）。当 ISP 将该附加服务用于 1.5～155Mbit/s 的互联网网关时，每月使用费为 10 万日元（约合人民币 6300 元）。

## D.2 国内 IPv6 发展概况

在国内，最早进入 IPv6 领域的是 CERNET，CERNET 国家网络中心于 1998 年 6 月加入 6bone，同年 11 月成为其骨干网成员。1999 年，CERNET 在国内教育网范围内组建了 IPv6 试验床，八大地区网络中心全部加入，CERNET 在试验床的 p-TLA 地址范围内开始分配地址，同时进行了有关 IPv6 各种特性的研究与开发。另外，CERNET 也从 APNIC 申请到了正式的 IPv6 地址。国内的另外一个 IPv6 试验网络是 6TNET，其由信息产业部电信研究院于 2002 年负责建设。

目前我国政府相当重视 IPv6 的发展。2003 年，以国家战略项目——中国下一代互联网示范工程（CNGI）启动为标志，我国 IPv6 商用化进程进入了实质性发展阶段。中国的五大运营商及两个学术网络全面加入 IPv6 规模部署阵营。CNGI 核心网络建设目标是在 2003~2005 年期间，采用 IPv6 技术，完成 CNGI 主干网（覆盖 20 个城市 39 个核心节点）以及国内与国际互联中心的建设，并实现与国际下一代互联网的高速连接。以 CNGI 项目为标志，我国 IPv6 商用化进程进入了实质性发展阶段。

中国 IPv6 发展大事记

| | |
|---|---|
| 1998 年 | 中国教育科研网（CERNET）建立 IPv6 试验床 |
| 1999 年 | 国家自然科学基金联合项目“中国高速互联研究试验网 NSFCNET”启动 |
| 2000 年 | 天地互连公司建立 IPv6 商用试验床 |
| 2001 年 | 中国电信启动《IPv6 总体技术方案》项目研究工作 |
| | 中科院知识创新工程“IPv6 网络关键技术研究和城域示范系统” |
| 2002 年 | 信息产业部“下一代 IP 电信试验网（6Tnet）”项目启动 |
| | 湖南电信 IPv6 试验网项目、重庆网通信息港 IPv6 城域示范网项目启动 |
| 2003 年 | 6Tnet 启动 |
| | IPv6 城域网建设 |
| | 协和医院等 SARS 定点医院采用“IPv6 新一代网络远程医疗、探视系统” |
| | 中国 IPv6 网络与应用演示中心建立中国下一代互联网示范项目（CNGI）全面启动 |
| | CERBNET2 网络建设启动 |
| 2004年 | 中国第一个下一代互联网主干网——CERNET2 试验网在北京宣布开通 |

中华人民共和国
通信行业标准
IPv4网络向IPv6网络过渡中的互联互通技术要求
YD/T 1612-2007
*
人民邮电出版社出版发行
北京市崇文区夕照寺街14号A座
邮政编码：100061
北京新瑞铭印刷有限公司

*
开本：880×1230　1/16　　2007年6月第1版
印张：5.5　　2007年6月北京第1次印刷
字数：168千字
ISBN 978-7-115-1412/07-75
定价：40.00元
本书如有印装质量问题，请与本社联系　电话：(010)67114922