ICS 33 060 40
M 33

YD

中华人民共和国通信行业标准

YD/T 1635-2007

IPv6 网络技术要求
——面向网络地址翻译（NAT）用户的 IPv6 隧道技术

Technical Requirements for IPv6 network
——IPv6 Tunneling Technology with private network addresses through NAT devices

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；
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。

与之相关的“IPv6 协议”系列标准的结构及名称目前如下：

1. IPv6 技术要求——IPv6 协议；
2. IPv6 技术要求——支持计算机移动部分；
3. IPv6 技术要求——地址，过渡及服务质量；
4. IPv6 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。

本标准的发布机构提请注意如下事实，声明符合本标准时，可以使用涉及4.3节中有关《一种可穿越网络地址翻译的自动隧道的方法》和4.3节中有关《一种利用双重隧道机制穿越NAT的方法》的相关专利。

本标准的发布机构对于专利的范围、有效性和验证资料不提出任何看法。

专利持有人已向本标准的发布机构保证，他愿意同任何申请人在合理和非歧视的条款及条件下，就使用授权许可证进行谈判。在这方面，该专利持有人的声明已在本标准的发布机构备案。有关资料可从以下地址获得：

专利持有单位名称：中国科学院计算技术研究所

地址：北京海淀科学院南路6号

请注意除上述已经识别出的专利外，本标准的某些内容有可能涉及专利。本标准的发布机构不应承担识别这些专利的责任。

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：中国科学院计算技术研究所
中国联合通信有限公司
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

本标准主要起草人：刘　敏　李忠诚　舒　童　王明会　杨　征　张　荣　陈洪飞　戴险峰　文　武

IPv6网络技术要求
——面向网络地址翻译（NAT）用户的IPv6隧道技术

1 范围

本标准规定了在IPv4网络向IPv6网络过渡中出现的面向NAT用户的IPv6隧道机制。本标准适用于IPv4网络和IPv6网络中位于NAT域内的IPv6/IPv4双栈主机。

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 RFC768（1980） 用户数据报协议（UDP）
IETF RFC791（1981） IP协议
IETF RFC1321（1992） MD5算法
IETF RFC2401（1999） IP协议的安全体系结构
IETF RFC2460（1998） IPv6协议规范
IETF RFC2461（1998） IPv6邻居发现协议
IETF RFC2572（1999） 简单网络管理协议的消息处理和分配
IETF RFC2574（1999） SNMPv3的基于用户的安全模型
IETF RFC2575（1999） SNMP的基于视图的访问控制模型
IETF RFC2663（1999） IP网络地址转换（NAT）的术语和考虑事宜
IETF RFC3315（2003） IPv6动态主机配置协议（DHCPv6）
IETF RFC3489（2003） STUN：利用UDP报文穿越NAT设备
IETF RFC3633（2003） IPv6动态主机配置协议的前缀选项

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。

通信（communication）

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

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。

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。

链路（link）

通信设备或媒介。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示符。如：以太网链路的IEEE 802地址和ISDN链路的E.164地址。

节点（node）

实现IPv6的设备。

数据包（packet）

由IP头和有效载荷构成的数据块。

网络地址转换（Network Address Translator，NAT）

本标准引用的术语网络地址转换（NAT）与RFC2663中的IPv4网络地址转换（NAT）意义相同。

Teredo 客户端（Teredo Client）

Teredo 客户端是指支持 Teredo 隧道接口的 IPv6 / IPv4 双栈节点，可通过Teredo协议与其他的Teredo客户端以及 IPv6 Internet上的其他节点（通过 Teredo 中继）建立IPv6通信。

Teredo 服务器（Teredo Server）

Teredo 服务器是具有IPv4全局地址的 IPv6 / IPv4 双栈节点，支持Teredo 隧道接口，用于帮助Teredo客户端进行地址配置，以及协助Teredo 客户端与其他的Teredo客户端或纯 IPv6 主机建立通信连接。Teredo 服务器使用 UDP 3544 端口侦听 Teredo 通信。

Teredo 中继（Teredo Relay）

Teredo中继是指能够在 IPv4 Internet 上的 Teredo 客户端之间（使用 Teredo 隧道接口）以及Teredo客户端与纯 IPv6 主机之间传送数据包的 IPv6 / IPv4 双栈路由器，负责在整个Internet上或Internet的一个子集范围内通告Teredo服务前缀的可达性。在某些情况下，Teredo 中继和 Teredo 服务器协同工作，帮助Teredo客户端与其他的Teredo客户端或纯 IPv6 主机建立通信连接。Teredo 中继使用 UDP 3544 端口侦听Teredo 通信。

Teredo 气泡包（Teredo Bubble）

Teredo气泡包是为建立或维持一个 NAT 映射而发送的最小的IPv6包，只包含一个IPv6 包头。其IPv6有效载荷为null，载荷类型为59。

Teredo 前缀（Teredo IPv6 Service Prefix）

用于组建Teredo地址的特殊IPv6前缀。

全局 Teredo 前缀（Global Teredo IPv6 Service Prefix）

取值为2001:0000:/32。

Teredo UDP 端口（Teredo UDP port）

Teredo服务器用于监听Teredo报文的UDP端口，该值为3544。

Teredo 服务端口（Teredo Service Port）

Teredo客户端发送Teredo报文的端口。

Teredo 发现地址（Teredo IPv4 Discovery Address）

用于发现同一IPv4子网中其他Teredo客户端的IPv4多播地址，其值为224.0.0.253。

Silkroad 客户端（Silkroad Client）

位于NAT域内，支持 Silkroad隧道接口的 IPv6 / IPv4 双栈节点，可通过Silkroad协议与其他的Silkroad客户端或者IPv6 Internet上的其他节点建立IPv6通信。

Silkroad 接入路由器（Silkroad Access Router）

能够通过公有IPv4地址访问IPv4 Internet的IPv6/IPv4双栈路由器，用来帮助Silkroad客户端建立IPv6连接。

Silkroad 导航器（Silkroad Navigator）

Silkroad导航器是用来帮助Silkroad接入路由器相互路由的节点，只需能够被Silkroad接入路由器访问，可以通过IPv4或IPv6来寻址，具体方式取决于特定的ISP。Silkroad导航器为推荐使用。

Silkroad UDP 端口（Silkroad UDP port）

Silkroad接入路由器用于监听Silkroad报文的UDP端口，该值为5188。

3.2 缩略语

IETF	Internet Engineering Task Force	互联网工程任务组
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
NAT	Network Address Translator	网络地址翻译
RS	Route Solicitation	路由请求
RA	Route Advertisement	路由通告
SC	Silkroad Client	Silkroad客户端
SAR	Silkroad Access Router	Silkroad接入路由器
SN	Silkroad Navigator	Silkroad导航器

4 面向 NAT 用户的 IPv6 隧道机制

NAT（即网络地址翻译）是解决IPv4地址不足的一种技术，用来实现多个用户共用一个合法IPv4地址与外部Internet进行通信。在我国，由于IPv4地址资源严重不足，NAT技术应用得非常广泛。然而，传统隧道机制均无法实现NAT域内的IPv6节点与其他IPv6节点间的通信。本标准中所涉及的两种隧道技术（Teredo和Silkroad）均采用IPv6-in-UDP隧道，也就是通过将IPv6报文封装在IPv4的UDP报文载荷中的方式实现NAT用户和IPv6网络的互连。其区别在于，Teredo是一种自动隧道技术，需要特殊的IPv6地址前缀。Teredo的IPv6地址中嵌有Teredo服务器的IPv4地址，以及客户端映射后的IPv4地址和端口。因此其IPv6地址会随内嵌IPv4地址的改变而改变。另外，Teredo隧道不能穿越对称NAT（Symmetric NAT）。与Teredo相比，Silkroad是一种配置隧道技术，可为NAT用户分配固定或动态的IPv6地址，且所分配的IPv6地址没有采用固定格式的前缀，也没有嵌入NAT映射地址或映射端口等信息，可支持包含对称NAT在内的现有NAT类型。Silkroad是一种穿越NAT域建立可控隧道的方法。这对愿意为NAT域内客户提供IPv6连接的ISP来说是一种有效的方案。但在Silkroad中进行路由优化要比Teredo复杂，因为Silkroad客户端所使用的IPv4地址和UDP端口无法通过其IPv6地址直接获得。

Teredo和Silkroad作为两种面向NAT用户的IPv6隧道机制，其适用范围不同，用户可根据需要进行选择。Teredo和Silkroad隧道机制的目的是为NAT用户提供IPv6连接，其代价是增加了UDP封装的开销。所以，对于有本地IPv6接入的用户以及NAT域外的用户，不建议使用Teredo和Silkroad隧道。

4.1 NAT 分类

NAT（Network Address Translator，网络地址翻译）是一种把IP地址从一个域映射到另一个域的方法。实现的核心是把内部网络中数据报文的地址翻译为外部合法地址，并将报文向外部网络发送；而在收到外部数据报文后，再翻译为内部地址并向内部网络发送。这有可能是因为内部地址对外网无效，也可能是因为内部地址必须对外网保密。

本标准假定读者熟悉NAT设备。NAT设备能够采用许多不同的策略来实现。[RFC3489] 将NAT分为4类：完全锥型NAT（Full Cone NAT，或简写为Cone NAT）、受限锥型NAT（Restricted Cone NAT）、端口受限锥型NAT（Port Restricted Cone NAT）和对称NAT（Symmetric NAT）。

Teredo可以对完全锥型NAT、受限锥型NAT、端口受限锥型NAT中的用户提供IPv6连接服务；Silkroad可以为所有这四种NAT域中的用户提供IPv6连接服务。

因为TCP报文和UDP报文是仅有的两类可以通过所有常用的NAT类型的报文。而由于TCP的拥塞控制机制，利用TCP来传输IPv6报文实时性差，故此目前穿越NAT的技术通常选用UDP报文作为承载报文。

4.2 Teredo

4.2.1 Teredo 地址

Teredo地址格式如下：

```
+--------+-------------+-------+------+-------------+
| Prefix | Server IPv4 | Flags | Port | Client IPv4 |
+--------+-------------+-------+------+-------------+
```

—Prefix：32位的Teredo地址前缀。

—Server IPv4：Teredo服务器的IPv4地址。

—Flags：16位，指示地址和NAT的类型。

—Port：Teredo服务在客户端的UDP映射端口。

—Client IPv4：客户端的IPv4映射地址。

在此格式中，客户端的UDP映射端口和IPv4映射地址是隐藏的。此地址和端口的每个比特要取非，通过把16位的端口号和32位的地址分别与OxFFFF和OxFFFFFFFF进行按位异或运算来实现。

IPv6的寻址规则指出，“对以非000开头的所有单播地址，要求用改进的EUI-64格式构建长64位的接口标识”。这就对Flags域的编码做出了规定：Flags域的16位应与改进的EUI-64 ID最重要的16位有效值相对应。

```
0         00        1
0         78        5
+----+----+----+----+
|Czzz|zzUG|zzzz|zzzz|
+----+----+----+----+
```

此格式中：

——“UG”位应设为“00”，指示非全局的单播标识。

——若客户端确信它位于完全锥型NAT域中，则C（cone）位应设为1，否则设为0；这些值会在资格确认过程中确定不同的服务器行为（详见4.2.3.1节）以及客户端和中继对气泡包的不同处理。

——“z”位必须设为0，并在接收时被忽略。

因此，Flags域当前有两种指定的值：若C位为0，则Flags为0x0000；若C位为1，则Flags为0x8000。

在有些情况下，Teredo节点使用本地链路地址。这些地址包含本地链路前缀（FE80::/64）和64位标识，其构成方式与上述格式相同。本地链路地址和全局地址的区别在于全局地址中的标识必须包含全局范围的IPv4单播地址，而本地链路地址中的标识可以包括IPv4私有地址。

4.2.2 数据包格式

在Teredo隧道机制中，IPv6数据包是封装在IPv4[RFC791]的UDP[RFC768]数据包中进行传输的。源和目的节点的IP地址和UDP端口使用本节所指定的值。数据包分两种格式：简单封装（simple encapsulation）和带原始指示符的封装（encapsulation with origin indication）。

简单封装时，直接把IPv6报文封装在UDP报文的载荷里面：

```
+------+-----+-------------+
| IPv4 | UDP | IPv6 packet |
+------+-----+-------------+
```

当服务器中转来自第三方的数据包时，它可能会在UDP载荷的前几个字节中插入原始指示符（origin indication）：

```
+------+-----+------------------+-------------+
| IPv4 | UDP | Origin indication | IPv6 packet |
+------+-----+------------------+-------------+
```

原始指示符有8个字节，格式如下：

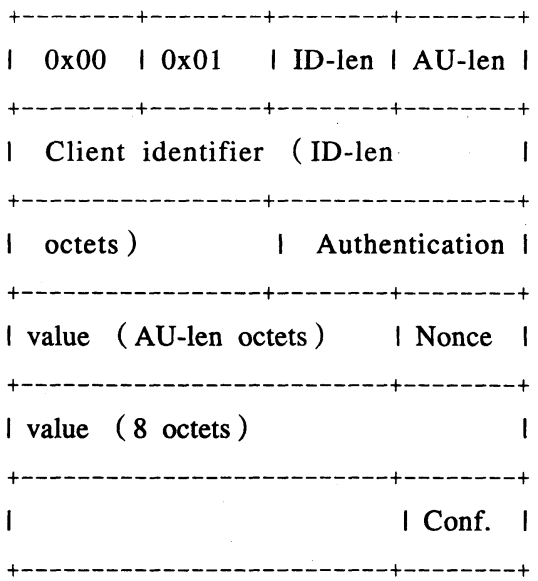
```
+-------+-------+---------------+
| 0x00  | 0x00  | Origin port # |
+-------+-------+---------------+
|   Origin IPv4 address         |
+-------------------------------+
```

原始指示符的前两个字节设为null，用来区分简单封装和带原始指示符的封装，因为简单封装的前4位包含IPv6协议的标识。接下来的16位包含隐藏的源端口号（网络字节序）。其后32位包含隐藏的源IPv4地址（网络字节序）。此格式中，客户端的源IPv4地址和端口的每个比特都要取非，这里也是通过把16位的端口号和32位的地址分别与OxFFFF和OxFFFFFFFF进行按位异或操作来实现的。

在Teredo客户端和其服务器交换路由请求（Route Solicitation，RS）和路由通告（Route Advertisement，RA）消息时，数据包可包含验证指示符：

```
+------+-----+----------------+-------------+
| IPv4 | UDP | Authentication | IPv6 packet |
+------+-----+----------------+-------------+
```

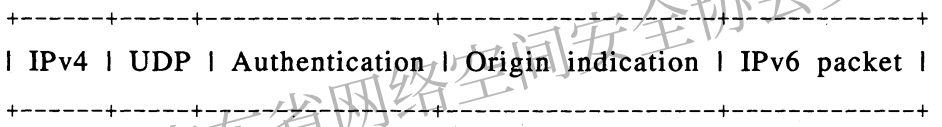
验证指示符的长度可变，包括客户端标识（Client Identifier）、验证值（Authentication Value）、当前值（Nonce）和确认字节（Confirmation Byte），具体格式如下：



验证指示符的第1个字节为null，第2个字节为1，这样就可以与IPv6报文和原始指示符区分开。第3个字节指定客户端标识的长度（以字节为单位）；第4个字节指定验证值的长度（以字节为单位）。验证值后是8字节的当前值和确认字节。

如果服务器不要求明确的客户端认证，ID-len和AU-len可设为null。

验证指示符和原始指示符可以组装在一起。在这种情况下，验证指示符必须是UDP载荷的第1个元素：



4.2.3 Teredo 客户端的描述

在使用Teredo服务之前，客户端必须配置：服务器的IPv4地址和该服务器的第二IPv4地址（secondary IPv4 address）。

如果要求安全检测，客户端必须配置：（1）客户端标识；（2）与服务器的共享密钥；（3）与服务器共享的认证算法。

为避免在端口预留（port conserving）的NAT域中进行操作所出现的问题，在同一NAT域中的不同客户端应使用不同的发送端口。这可以显示地设定；若未设定，则随机选取服务端口。

客户端要维护以下反映Teredo服务状态的变量：（1）Teredo连接状态；（2）Teredo服务端口对应的映射地址和端口号；（3）Teredo服务端口对应的Teredo前缀；（4）根据前缀生成的Teredo IPv6地址；（5）本地链路地址；（6）与Teredo服务器最近一次交互的日期和时间；（7）Teredo刷新间隔；（8）随机刷新间隔；（9）Teredo对等节点列表。

在发送数据包之前，客户端必须完成Teredo的资格确认过程，确定Teredo的连接状态、映射地址和端口以及Teredo前缀。客户端还应完成完全锥型NAT的确认过程，确定NAT的状态，此过程有可能改变前缀值。如果资格确认是成功的，那么客户端可以用Teredo服务端口发送和接收IPv6包。

上述过程使用对等节点列表。对每个节点而言，该列表包括：（1）节点的IPv6地址；（2）节点的IPv4映射地址和UDP映射端口；（3）映射地址状态，即可信或不可信；（4）最近一次向此节点发送的nonce值；（5）最近一次从此节点接收报文的日期和时间；（6）最近一次向此节点发送报文的日期和时间；（7）发送给该节点的气泡包数。

节点列表使IPv6包的传输能够使用直接路径。客户端应该实现列表的管理策略，例如：删除最近最少使用的表项。客户端应保证列表具有足够的长度，以避免不必要的气泡包交换。

客户端必须定期完成维护过程，以保证Teredo服务端口的可用性。是否使用此过程依赖于与Teredo服务器最近一次交互的时间距离当前时间的时间间隔。刷新过程使用“Teredo刷新间隔”参数，此参数初值设为30秒。随机刷新间隔是在刷新间隔的75%到100%之间选取一个随机数。

Teredo客户端可以使用4.2.3.6节中定义的“可选的本地客户端发现过程”来避免位于相同NAT域中的站点间的三角路由。当NAT不能把某一内部主机发送的包转发给另一内部主机的映射地址和端口时，使用此过程还可以增强连通性。

4.2.3.1 资格确认过程

资格确认过程的目的是确定本地IPv4连接的状态和本地Teredo接口的Teredo前缀。此过程从“initial”状态开始，若资格确认成功，则在“qualified”状态结束，否则在“off-line”状态结束。

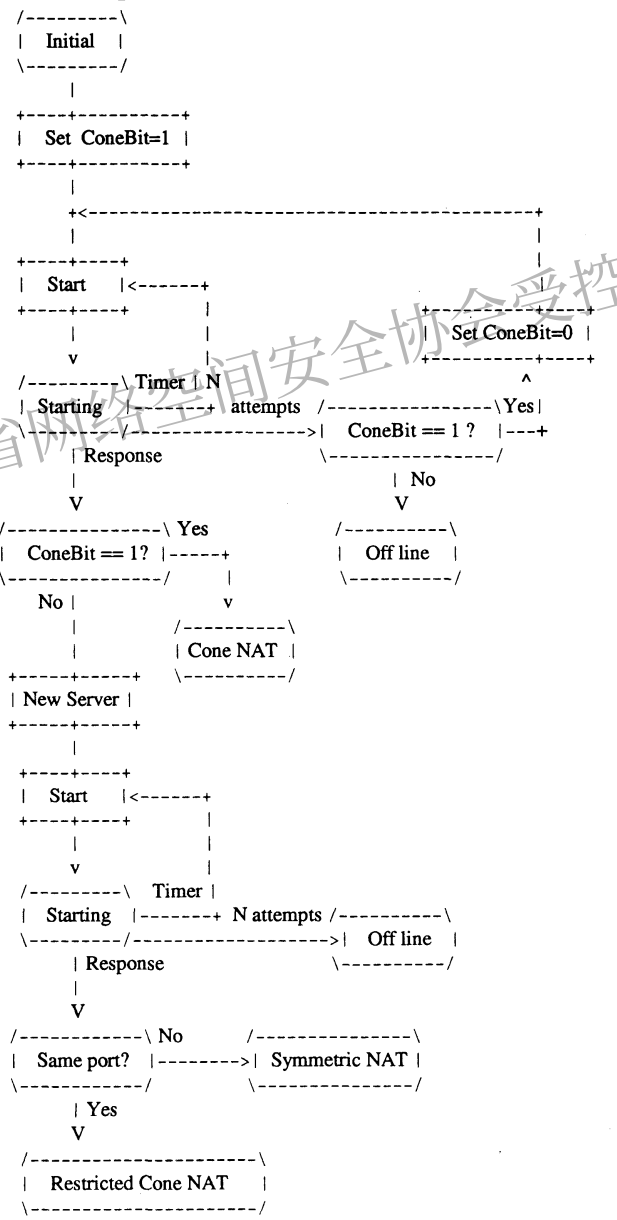


图1 资格确认过程

起初，Teredo连接状态被设为“initial”。当接口初始化后，正如[RFC2461]中所定义的，系统首先发送路由请求消息RS来执行起始操作。客户端选取一个本地链路地址并把它作为消息的IPv6源地址，地址中的C位设为1（参见第4.2.1部分中的地址格式）。RS的IPv6目的地址是所有路由器多播地址（all-routers multicast address）。数据包将通过UDP报文从Teredo服务端口发送到Teredo服务器的IPv4地址和Teredo UDP端口。此后，连接状态将变为“starting”。

在“starting”状态，客户端监听由Teredo服务器发送的路由通告。如果在时间T内，未收到响应，客户端会重新发送路由请求消息。如果重复N次均无回应，那么客户端就认为自己不在完全锥型NAT域中，因此把C位设为0，并重复此过程。如果N次超时和重发后仍没有回应，则客户端就断定自身不能使用UDP，Teredo服务不可用；并把状态设为“off-line”。根据[RFC2461]，默认的超时时间T为4秒，最大重复数N=3。

如果收到响应，那么客户端检查响应中是否包含原始指示符和在[RFC2461]中定义的有效的RA，检查IPv6目的地址是否与RS中的本地链路地址相同，检查RA是否包含前缀信息选项。此前缀应是有效的Teredo前缀：前32位应包含全局Teredo前缀，紧接着的32位应包含Teredo服务器的IPv4地址。在这种情况下，客户端将从原始指示符中获知Teredo映射地址和Teredo映射端口。RA的IPv6源地址是Teredo服务器的本地服务器地址（link-local server address）。若响应不是有效的RA，将直接丢弃。

如果客户端收到一个RA，并且这个RA的IPv6目的地址中的C位为1，则该客户端位于完全锥型NAT域中，并完全符合资格确认的要求。若所收到的RA的C位为0，则该客户端无法知道本地NAT是受限NAT还是对称NAT。此时客户端选用服务器的第二IPv4地址，并重复此过程，其中C位仍设为0。如果客户端收不到响应，那么可探知此服务不可用。如果客户端收到响应，那么它将把第2次响应中的映射地址和映射端口与第1次收到的值进行比较。若值不同，则客户端位于对称NAT域中，不能使用Teredo服务。若值相同，则客户端位于端口受限锥型NAT或受限锥型NAT中，可以使用Teredo服务（对受限锥型NAT和端口受限锥型NAT，Teredo的操作相同。）

如果客户端符合资格确认的要求，那么它将使用从RA中获知的Teredo前缀，以及原始指示符中的隐藏UDP映射端口和IPv4映射地址来创建Teredo IPv6地址。若客户端位于完全锥型NAT域内，C位设为1，否则为0。此后，客户端便可以开始使用Teredo服务。

4.2.3.2 数据包的接收

Teredo客户端从Teredo接口接收数据包。包接收的过程不仅包括接收数据包，而且还要维护与Teredo服务器的最新交互日期和时间，以及当前节点列表。

当从Teredo服务端口收到UDP包时，Teredo客户端根据4.2.2节定义的包格式检查它的编码格式，判断其中是只包含有效的IPv6包，还是有效的原始指示符和IPv6包的组合。相关报文可能会通过验证指示符来进行安全防护。如果数据包不符合要求，就会被直接丢弃。

如果IPv6包遵循[RFC2460]，则被认为是有效的：应有协议标识符指示是IPv6包，载荷长度应与封装它的UDP报文长度相一致。此外，客户端应检测该包的IPv6目的地址是否与其Teredo地址相对应。

然后，Teredo客户端检查所收包的IPv4源地址和UDP端口。如果这些值与服务器的IPv4地址和Teredo端口相匹配，那么客户端把“与Teredo服务器交互的最新日期和时间”更新为当前日期和时间。如果出现原始指示符，客户端应进行4.2.3.7节描述的“直接IPv6连接测试”。

如果IPv4源地址和UDP端口与服务器的IPv4地址和Teredo端口不同，那么客户端检查包的IPv6源地址：

（1）如果可信的对等节点列表中有该IPv6源地址对应的表项，那么客户端把表项中的IPv4映射地址和映射端口与数据包的源IPv4地址和源端口比较。如果值匹配，那么该包被接收；同时更新最近一次从对等节点接收报文的日期和时间。

（2）如果不可信的对等节点列表中有IPv6源地址对应的表项，那么客户端查看该数据包是否是ICMPv6 echo reply。在此情况下，若响应中的ICMPv6数据与对等节点表项中所存的nonce值相匹配，则接收此包，并把表项的状态改为“可信”。表项中的IPv4映射地址和映射端口应设为所收包的源IPv4地址和源端口，并且应更新最近一次从对等节点接收报文的日期和时间。

（3）如果源IPv6地址是Teredo地址，那么客户端把源IPv6地址中隐藏的IPv4映射地址和映射端口与数据包的源IPv4地址和源端口进行比较。如果匹配，那么客户端在对等节点列表中为该IPv6源地址创建一个对等节点表项；如果表项已经存在，那么只需更新表项。表项中IPv4映射地址和映射端口应设为所收包的相应值，并把状态设为“可信”。如果这个表项是新创建的，那么应把最近一次向此节点发送报文的日期和时间设为当前时刻的30秒之前，并把气泡包数设为0。若数据包为气泡包，则处理后即可丢弃；否则数据包应被接收。在这两种情况下，客户端都将把队列中等待此目的地址的数据包移出队列并进行转发。

（4）如果收到此包的IPv4地址是Teredo发现地址（Teredo IPv4 Discovery Address），IPv6源地址是有效的Teredo地址，并且目的地址是所有节点多播地址（all nodes on link multicast address），那么该数据包应该被当作本地发现气泡包（local discovery bubble）。如果源地址不存在本地表项，那么就创建一个表项，但要把状态设为“不可信”。客户端应该向此气泡包的IPv4源地址和端口回复单播的Teredo气泡包，并把气泡包的IPv6源地址设为本地的Teredo地址，把气泡包的IPv6目的地址设为所收气泡包的IPv6源地址。（未实现可选的本地发现过程的客户端将不能处理本地发现气泡包）。

（5）如果IPv6源地址是Teredo地址并且源地址中的IPv4映射地址和映射端口与数据包的IPv4源地址和源端口不匹配，那么客户端会查看此IPv6地址是否存在本地表项。如果存在，并且该表项中的本地IPv4地址和本地端口与数据包的IPv4源地址和源端口匹配，那么客户端更新本地表项，将其状态设为“可信”。若数据包为气泡包，则处理后即可丢弃，否则应接收。在这两种情况下，客户端都将把队列中等待此目的地址的数据包移出队列并进行转发。

（6）在其他情况下，数据包可以被接收，但客户端应该意识到可能存在源地址欺骗。在数据包处理前，客户端应进行4.2.3.7节所述的“直接IPv6连接测试”。

无论IPv4源地址和UDP源端口是什么，收到IPv6包的客户端都可以向该目标发送Teredo气泡包，参见4.2.3.5节。

4.2.3.3 数据包的发送

当Teredo客户端从Teredo接口发送数据包时，它会检测IPv6目的地址。客户端首先检查当前Teredo对等节点列表中是否有此IPv6地址对应的表项，以及此表项是否仍然有效：如果此表项（本地/非本地对等节点）对应的最近一次接收日期和时间距当前时间在30秒之内，那么此表项有效。（只有在客户端使用4.2.3.6中所述的本地客户端发现过程的情况下，本地对等节点表项才存在。）

然后，客户端进行以下操作：

（1）如果在对等节点列表中，有此IPv6地址对应的表项并且该表项状态为“可信”，那么应该利用UDP封装把IPv6包发送到表项中指定的IPv4地址和UDP端口。客户端更新对等节点表项中的最近发送日期和时间。

（2）如果目的地址不是Teredo 地址，那么数据包进入队列，并且客户端进行4.2.3.7节中所述的“直接IPv6连接测试”。如果此过程成功，那么数据包出队并转发。如果在2秒内，直接IPv6连接测试未完成，应该重复此过程，且最多重复3次。

（3）如果目的地址是本地对等节点的Teredo地址（即在最近600秒内收到过该Teredo地址发来的本地发现气泡包），那么数据包进入队列。客户端向表项中指定的本地IPv4地址和本地端口发送一个单播Teredo气泡包，并向Teredo发现地址发送一个本地Teredo气泡包。

（4）如果目的地址是Teredo地址并且它的C位为1，那么可以从IPv6地址中抽取出IPv4映射地址和UDP映射端口，把IPv6包封装在UDP中发向该地址。

（5）如果目的地址是Teredo地址，并且C位为0，那么数据包进入队列。如果客户端位于非完全锥型NAT域中，它就向目的地址的IPv4映射地址和映射端口，发送直接气泡包。在任何情况下，客户端都将通过UDP隧道把一个间接气泡包发向服务器地址和Teredo端口，此气泡包将被转发到目的地址。当客户端收到直接来自Teredo对等节点的气泡包或数据包时，缓存的数据包将出队并转发。如果在2s的时限内未收到气泡包，应重复此过程，且最多重复3次。

在第4和第5种情况下，用UDP发送数据包前，客户端必须查看IPv4目的地址是否是全局单播地址格式；如不是就直接丢弃。（注意：如果成功地执行了本地发现过程，那么在第1种情况下，数据包可以正确地发送到非全局单播地址。）

全局单播地址检查是为预防许多可能的攻击而设计的，因为攻击者可能使用Teredo主机对单个本地IPv4目标或一组目标进行攻击。在Teredo的实现中，IPv4地址如果不属于以下地址，就被认为是全局单播地址：

—本地子网0.0.0.0/8；

—环回子网127.0.0.0/8；

—本地地址范围10.0.0.0/8；

—本地地址范围172.16.0.0/12；

—本地地址范围192.168.0.0/16；

—本地链路地址块169.254.0.0/16；

—6to4任播地址的保留块192.88.99.0/24；

—多播地址块224.0.0.0/4；

—受限的广播地址255.255.255.255；

—主机所在子网的直接广播地址。

由于可靠性的原因，客户端可以决定忽略flag中C位的值，跳过第4种情况的测试，总是按第5种情况进行处理，即认为所有Teredo对等节点都位于非完全锥型NAT中。这将引起流量的增加，但是在对NAT状态的确定因某种原因而出错的情况下，却可以避免出现可靠性方面的问题。同理，即使客户端认为它们位于完全锥型NAT域中，也可以总是用第5种情况中的方法发送直接气泡包。

4.2.3.4 维护

Teredo客户端必须保证它所用的映射是有效的，可以通过检查在规定的时间内是否有来自Teredo服务器的数据包来保证有效性。

在一定的时间间隔内，客户端必须查看“与Teredo服务器最近一次交互的日期和时间”，从而确保在最后一个随机的Teredo刷新时间间隔内至少收到一个数据包。若未收到，则客户端应向服务器发送RS消息，参见4.2.3.1。客户端需要使用在资格确认过程中收到RA时使用的C位值。

当客户端收到RA时，应该按4.2.3.1所述，来检查它的有效性，丢弃无效的RA。如果RA是有效的，那么客户端必须检查映射地址和端口是否与当前Teredo地址匹配。若不匹配，则映射已经改变，客户端必须把旧地址标记为无效，并开始使用新地址。

4.2.3.5 发送 Teredo 气泡包

Teredo客户端接收到数据包后，或者进行了一次发送尝试后，可能需要向另一个Teredo客户端发送气泡包。有两种气泡包：被直接发向对等节点的IPv4映射地址和UDP映射端口的气泡包称为直接气泡包；由对等节点的Teredo服务器转发的气泡包称为间接气泡包。

当Teredo客户端要发送直接气泡包时，它从目标Teredo地址中取得IPv4映射地址和映射端口，然后查看当前对等节点列表中是否已有该IPv6地址的对应表项。如果没有，那么客户端必须为该地址创建一个新表项，把最新接收日期和最新发送日期均设为当前时刻的30秒之前，并把气泡包数设为0。

当Teredo客户端要发送间接气泡包时，它从目标IPv6地址的Teredo前缀（不同的客户端可以用不同的服务器）中提取出Teredo服务器的IPv4地址。气泡包将被发送至该IPv4地址和Teredo UDP端口。气泡包可能在传输过程中丢失，多次发送可以增强Teredo服务的可靠性。然而，气泡包在某种NAT配置下，也会因系统原因而丢失。为了在可靠性与不必要的重传之间进行权衡，作出如下规定：

—如果最近一次向某节点发送报文的日期和时间距当前时刻小于2s，那么客户端不能向此节点发送气泡包。

—如果客户端在最近300s内已经向该对等节点发送过4个气泡包，而未收到直接的响应，那么客户端不能再发送气泡包。

在其他情况下，客户端可以发送气泡包。当它发送气泡包时，必须更新最近一次向该对等节点发送报文的日期和时间，并把气泡包数加1。

4.2.3.6 可选的本地客户端发现过程

此过程的设计是为了使位于同一NAT域中的Teredo客户端之间能够直接通信而不必通过Teredo服务器中转。

希望使用本地发现过程的Teredo客户端应该加入到Teredo发现地址指定的IPv4多播组中。客户端应该等待本地发现气泡包在Teredo 发现地址上被接收。客户端必须以在200~300s间服从均匀分布的随机时间间隔向Teredo发现地址发送本地发现气泡包，该气泡包有以下特征：

—IPv4源地址：发送端的IPv4地址

—IPv4目的地址：Teredo发现地址

—IPv4 ttl：1

—UDP源端口：发送端的Teredo服务端口

—UDP目的端口：Teredo UDP端口

—UDP载荷：最小IPv6包，其内容如下：

—IPv6源地址：发送端的全局Teredo地址

—IPv6目的地址：所有节点多播地址（all-nodes on-link multicast address）

—IPv6载荷类型：59（无Next Header，如[RFC2460]所定义）

—IPv6载荷长度：0

—IPv6跳数限制：1

本地发现过程有可能带来拒绝服务攻击，这是因为恶意节点可以向其他节点发送虚假气泡包，从而捕获来自这些节点的流量。4.2.3.4节中描述的过滤规则可以削弱这种攻击。另外，Teredo发现地址的“link only”多播范围，同样可以削弱这种攻击，因为这意味着发送给这些地址的数据包将不被路由器转发。

为了从“link only multicast”保护中受益，客户端应该直接丢弃所有通过单播地址收到的本地发现气泡包。为进一步削弱拒绝服务攻击，如果本地发现气泡包的IPv6源地址不是正确的Teredo格式的IPv6地址，或者其IPv4源地址不属于本地IPv4子网，则客户端必须直接丢弃该气泡包。若本地发现气泡包的Teredo地址包含的IPv4映射地址与客户端本身的映射地址不相同，客户端可以决定直接丢弃这种本地发现气泡包。

如果气泡包被接受，那么客户端要查看当前对等节点列表中是否存在与该气泡包的源IPv6地址所对应的表项。若存在，客户端必须更新本地对等节点地址和本地对等节点端口，来反映气泡包的IPv4源地址和UDP源端口。若不存在，客户端必须创建一个新的表项，并把本地对等节点地址和本地对等节点端口设为气泡包的IPv4源地址和UDP源端口，把最近一次从此节点接收报文的日期和时间设为当前时刻，把最近一次向此节点发送报文的日期和时间设为当前时刻的30s之前，把气泡包数设为0。把此表项的状态设为“不可信”。

接收到本地发现气泡包后，客户端用4.2.3.2节中所述的单播气泡包回复。

4.2.3.7 直接 IPv6 连接测试

设计此过程是为了使Teredo主机和Teredo中继之间能够直接通信。位于完全锥型NAT域中的Teredo主机会直接收到来自Teredo中继的数据包；其他Teredo主机会通过本地Teredo服务器知道第三方的初始地址和UDP端口。所有这些情况都存在IPv6源地址被欺骗的危险。Teredo主机必须做出两个决定：是否接收数据包并进行本地处理，以及是否通过新获知的IPv4地址和UDP端口继续向该IPv6地址发送数据包。基本原则是大胆接收，小心发送。为防止欺骗而拒绝接收数据包会降低连通性，只有当基本上确定源地址被欺骗时才应这样做。另一方面，应避免向错误地址发送数据包。

当客户端想要向IPv6节点或6to4节点发送数据包时，它应该检查与IPv6目的地址对应的有效对等节点表项是否已经存在。若不存在，则客户端会选取一个随机数（nonce）（建议使用至少64位长的随机数）并组成一个ICMPv6 Echo Request报文，其源地址是本地Teredo地址，目的地址是目标IPv6节点的地址，数据域设为nonce。该nonce值和发送日期会被记录在IPv6目的地址对应的临时对等节点表项中。然后，把这个ICMPv6数据包封装在UDP包中，发向Teredo服务器的IPv4地址和Teredo UDP端口。4.2.3.2节规定了如何处理此报文的回复。

4.2.4 Teredo 服务器的描述

Teredo服务器是无状态的。Teredo服务器使用为Teredo服务选定的IPv4地址，在Teredo UDP端口监听UDP包。此外，服务器能用不同的IPv4地址和端口发送和接收数据包。

Teredo服务器将起到IPv6路由器的作用。它接收路由请求消息，回复路由通告消息。它还接收其他数据包，例如ICMPv6消息和Teredo气泡包，并根据IPv6协议的规定来处理它们。

在默认的情况下，Teredo服务器的路由功能是有限的。Teredo服务器用于中转Teredo气泡包、ICMPv6 echo request和ICMPv6 echo reply，但是它不用于中转其他类型的IPv6数据包。不过，操作者可以决定是否把Teredo服务器和Teredo中继的功能组装在一起，参见4.2.5节。

4.2.4.1 Teredo 包的处理

在处理数据包之前，Teredo服务器必须检查封装的IPv6源地址，以及IPv4源地址和UDP源端口是否有效。

（1）如果UDP包的内容不符合4.2.2节所定义的Teredo包的格式，则直接丢弃该数据包。

（2）如果UDP包既不是Teredo气泡包，也不是ICMPv6消息，则应丢弃该包（如果Teredo服务器同时作为Teredo中继，那么可以处理该数据包，参见4.2.5）。

（3）如果IPv4源地址不是全局单播地址格式，那么直接丢弃此包。

（4）如果IPv6源地址是IPv6本地链路地址，IPv6目的地址是本地链路范围内的多播地址（FF02::2），并且数据包包含ICMPv6路由请求消息，那么此数据包必须被接收。如果服务器要求安全资格确认，但没有验证指示符或验证失败，那么丢弃该包。

（5）如果IPv6源地址是Teredo地址，并且其内嵌的IPv4地址和UDP端口与当前的IPv4源地址和UDP源端口匹配，那么应该接收此数据包。

（6）如果IPv6源地址不是Teredo地址，但IPv6目的地址是该服务器所分配出去的Teredo地址，那么应接收此包。

（7）其他任何情况下，直接丢弃此包。

然后，Teredo服务器查看被封装的IPv6目的地址：

（1）如果IPv6目的地址是本地链路范围内的多播地址（FF02::2）或是服务器的本地链路地址，那么Teredo服务器处理此包。服务器可能需要处理路由请求消息和ICMPv6 echo request消息。

（2）如果IPv6目的地址不是全局范围的IPv6地址，不能转发此包。

（3）如果目的地址不是Teredo地址，那么应该用常规的IPv6路由方式把此包转发到IPv6 Internet。

（4）如果IPv6目的地址是有效的Teredo地址，那么Teredo服务器必须检查此IPv6地址中内嵌的IPv4地址是否符合全局单播地址格式。若不符合，则直接丢弃。

如果地址有效，那么Teredo服务器用新的UDP数据包封装此IPv6包，参数设置如下：

—从IPv6目的地址中获得IPv4目的地址；

—IPv4源地址是Teredo服务器的IPv4地址；

—从IPv6目的地址中获得UDP目的端口；

—把UDP源端口设为Teredo UDP端口。

如果IPv6目的地址是Teredo客户端并且该客户端使用本服务器提供的服务，那么服务器应在UDP载荷的第1个字节中插入原始指示符，详见4.2.2节。（为了验证客户端是否由该服务器服务，服务器把客户端Teredo地址的32~63位与自己的IPv4地址进行比较）

4.2.4.2 路由请求的处理

当Teredo服务器收到RS消息时，它记录下源IPv4地址和UDP端口，将其作为客户端的Teredo映射地址和Teredo映射端口。服务器利用这些值来组装响应报文中的原始指示符。

Teredo服务器通过发送RA来响应RS。RA中必须包含由服务前缀和服务器的IPv4地址所组成的Teredo前缀。应该把IPv6源地址设为Teredo服务器的本地链路地址。此地址源自服务器的IPv4地址和Teredo端口，详见第4.2.1部分，C位设为1。IPv6目的地址设为RS的IPv6源地址。RA用UDP封装后发送给客户端的Teredo映射地址和Teredo映射端口。IPv4源地址和UDP源端口应设为服务器的IPv4地址和Teredo端口。如果客户

端的IPv6地址的C位为1，则服务器在发送RA的时候必须使用另一个地址，而不是接收RS的服务器地址。如果C位为0，那么服务器用相同的地址发回响应。

在发包前，Teredo服务器必须检查IPv4目的地址是否符合全局单播地址格式。若不符合，则直接丢弃（参见4.2.3.3节定义的全局单播地址）。

如果服务器要求安全认证，它就要在承载RA的UDP包中插入有效的验证指示符。

4.2.5 Teredo 中继的描述

Teredo中继是一种IPv6路由器，可通过IPv6路由协议通告Teredo前缀的可达性（最小的Teredo 中继可以只为一台本地主机服务，只把前缀告知此主机）。Teredo中继接收发往Teredo客户端的IPv6包，也应能接收Teredo客户端通过IPv4的UDP封装发来的数据包。中继可以应用过滤规则，例如：只接收自己曾经向其发送过报文的Teredo客户端所发来的数据包。

Teredo中继的收发规则与Teredo客户端相似：必须使用Teredo服务端口向Teredo客户端发送数据包，必须维护对等节点列表。此列表与Teredo客户端维护的对等节点列表相同。

4.2.5.1 向 Teredo 客户端发送报文

当Teredo中继要向Teredo客户端发送数据包时，将检测IPv6目的地址。按定义，Teredo中继只发送IPv6目的地址是有效的Teredo地址的IPv6包。

在处理数据包之前，Teredo中继必须检查Teredo地址中内嵌的IPv4目的地址是否符合全局单播地址格式。若不符合，则直接丢弃。（参见4.2.3.3节定义的全局单播地址）

然后，中继检查当前Teredo对等节点列表中是否存在与此IPv6地址对应的表项，以及该表项是否有效。接下来，中继要进行如下操作：

（1）如果在对等节点列表中有该IPv6地址对应的表项，并且该表项的状态为“可信”，那么把IPv6包用UDP封装发向此表项中的IPv4映射地址和UDP映射端口。更新最近一次向此节点发送报文的日期。

（2）如果对等节点列表中没有可信表项，并且目的地址是C位为1的Teredo地址，那么从该IPv6地址中抽取IPv4映射地址和UDP映射端口，把数据包用UDP封装后发向该地址和端口。

（3）如果对等节点列表中没有可信表项，并且目的地址是C位为0的Teredo地址，那么Teredo中继创建一个气泡包，源地址是本地IPv6地址，目的地址是数据包目的节点的Teredo地址。中继把气泡包发送给Teredo目的节点对应的服务器地址。当中继接收到来自此IPv6地址的气泡包或数据包时，将对应的表项设为可信。如果在2秒内未收到这样的报文，则Teredo中继重发气泡包，最多重复3次。如果中继多次重发后仍未收到气泡包，那么从对等节点列表中删除该表项。中继可以让不可信表项对应的数据包进入队列。当表项可信时，再让队列中的数据包出队，并转发包。如果表项被删除，则数据包也要丢弃。为了避免拒绝服务攻击，中继应限制队列中的数据包个数。

在第2和第3种情况下，Teredo 中继应为IPv6地址创建对等节点列表。表项状态在第2种情况下（完全锥型NAT域中）标记为可信，在第3种情况下为不可信。在第3种情况下，如果Teredo中继恰巧位于非完全锥型NAT域中，它还要向Teredo目的节点的IPv4映射地址和映射端口发送直接气泡包。这可以为Teredo客户端返回气泡包打开一条通路。

4.2.5.2 从 Teredo 客户端接收报文

Teredo中继可以接收来自Teredo客户端的数据包。一般情况下，只有中继先向该客户端发送过数据包后，即中继的对等节点列表中存在客户端的IPv6地址，该客户端才可以向中继发送数据包。中继和客户端一样，使用包接收过程来维护与Teredo服务器交互的最新日期和时间，以及当前对等节点列表。

当从Teredo端口接收到UDP包时，Teredo中继检查包中是否包含有效的IPv6包（参见[RFC2460]）。若无效，则直接丢弃。

然后，Teredo中继检查IPv6源地址是否是有效的Teredo地址，以及内嵌的IPv4映射地址和映射端口与接收报文的IPv4源地址和端口是否匹配。若不匹配，则直接丢弃。之后，Teredo中继检查当前对等节点列表中是否存在IPv6源地址所对应的表项。若不存在，则直接丢弃；若存在，则把表项设为“可信”。中继把最新交互日期和时间更新为当前日期和时间。

最后，中继检查IPv6目的地址。若目的地址在中继所服务的IPv6地址范围内，则接收此包，并向目的节点转发；否则，丢弃。

4.2.5.3 Teredo 中继和 Teredo 服务器的区别

因为Teredo服务器能转发Teredo包，因此所有的Teredo服务器必须拥有Teredo中继的功能，但不要求Teredo中继拥有Teredo服务器的功能。

如果把服务器和中继组合在一起，将给服务器设计增加额外的复杂度。其收包过程应是4.2.4.1中定义的服务器处理规则和4.2.5.2节中定义的中继处理规则的组合。

4.3 Silkroad

Silkroad通信过程的基础是假设NAT可以对UDP报文进行处理。

Silkroad隧道机制的目的是为NAT用户提供IPv6连接，其代价是增加了UDP封装的开销。所以，对于有本地IPv6接入的用户以及NAT域外的用户，不建议使用Silkroad隧道。

4.3.1 Silkroad 的体系结构

Silkroad隧道的体系结构如图2所示。

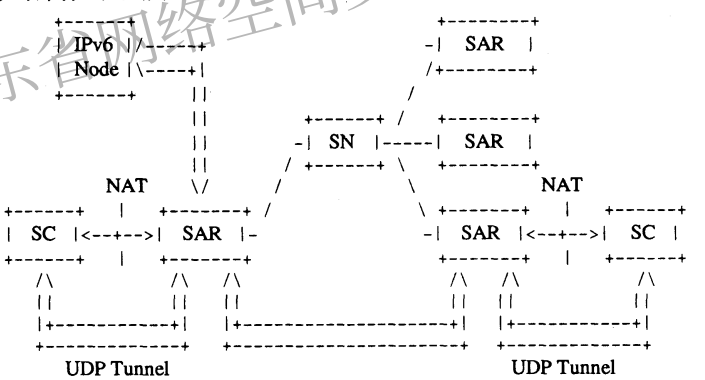


图2 Silkroad 的体系结构

如图2所示，Silkroad服务要求3种实体相互协作：（1）Silkroad客户端（SC），它位于NAT域内，但想要使用IPv6连接；（2）Silkroad接入路由器（SAR），它协助Silkroad客户端和现有IPv6网络互连；（3）Silkroad导航器（SN），它帮助Silkroad接入路由器传送数据包。

Silkroad服务是通过SC与SAR的交互，利用Silkroad服务协议发送和接收IPv6包来实现的。在SC和SN间没有直接的交互。SC和SAR部署量相对较少的ISP可以在它的SAR之间配置静态隧道。在这种情况下，可以不使用SN。拥有大量SAR和SC的ISP要实现N^2个信任模型是很困难甚至是不可行的，因此建议部署SN来协助SAR完成报文传输过程中相互间的路由。在此过程中，SN的角色类似于DNS（域名服务系统）。ISP中的SN可以分层，每个SN分管不同的SAR域。各ISP内部的SAR和SN的认证和一致性维护，将依赖于特定ISP对安全与路由的考虑和其现有的实现机制。如果没有事先约定的话，不同ISP的SAR和SN之间不

会进行自动的同步和交互。一般来说，不同ISP的SAR之间的包传递将遵循标准的IPv6规则，并且不进行路由优化。

Silkroad模型基于两点假设：（1）所有SC和SAR可以识别Silkroad端口。（2）所有SAR均知道其所属SN的地址。

4.3.1.1 Silkroad 客户端（SC）

SC是一个位于NAT之后，既要访问IPv4网络又要访问IPv6网络的节点。它主要有以下功能：（1）管理SC与SAR间的UDP隧道：创建、修改、删除隧道，确定NAT的类型，优化路由，发送keep-alive数据包；（2）和SAR协同实现安全控制；（3）入口过滤；（4）为应用程序提供透明的Silkroad服务。

4.3.1.2 Silkroad 接入路由器（SAR）

SAR是一个可通过全局可路由地址访问IPv4网络的IPv4/IPv6双栈路由器。它主要有以下功能：（1）管理与SC间的UDP隧道：创建、修改、删除隧道，确定NAT的类型，统计每个活跃隧道的使用情况；（2）IPv6地址的分配和管理；（3）维护当前通信对等节点列表；（4）借助SN来更新路由cache；（5）登录SN去查找某一IPv6目的地址是否是由另一SAR所服务的SC，若是则获得此SAR的IPv4地址；（6）SC的安全资格确认；（7）转发客户端和目的IPv6站点之间的通信数据流；（8）访问控制和安全控制。

ISP应该保证每个SAR能够处理它所管辖的用户数量和相应的传输流量。

4.3.1.3 Silkroad 导航器（SN）

对SN的唯一要求是能够被SAR访问。它可以是IPv4地址可达，也可以是IPv6地址可达，这取决于特定的ISP。SN主要有以下功能：（1）管理和控制本域中的SAR；（2）帮助SAR更新它们的路由cache；（3）帮助SAR查找某一特定SC的SAR，并提供它的IPv4地址；（4）如果有下层SN，则对它们进行管理并协助其进行地址解析。

正如我们上面所提到的，拥有许多SAR和SC的大型ISP建议部署SN，以便在包传输过程中帮助SAR进行相互间的路由。在此过程中，SN的角色类似于DNS。ISP的SN可以分层。每个SN分管不同的SAR域。但是在IPv6连接需求很大时，建议提供本地IPv6连接，而不是部署过多的SN。

4.3.2 Silkroad 的操作规程

4.3.2.1 SAR 的确定

作为Silkroad隧道机制的起始阶段，SC选择适当的SAR来为自己提供Silkroad隧道服务。换句话说，SC必须知道它的SAR的IPv4地址。SC获取SAR网络信息的方式有多种，例如SC可以从系统管理员处获得SAR的信息，也可以用DNS去查找服务名称，还可以用预先配置或预先设定的IPv4任播地址查找SAR。当然，ISP也可通过其他方式把SAR的地址或域名通告给SC。无论如何，SAR的信息一旦确定，将自动保存到SC的配置文件中。如果SC没有明确地改变它的配置，此SAR将成为SC的默认SAR。

4.3.2.2 IPv6 地址/前缀的获取

为了获得IPv6地址/前缀，SC通过UDP隧道向SAR发送一个常规的邻居发现（Neighbor Discovery）路由请求（Route Solicitation：RS）[RFC 2461]，或DHCPv6 SOLICIT[RFC 3315]，或前缀授权请求消息[RFC 3633]。

SAR通过UDP隧道向SC回复一个常规的邻居发现路由通告或是相应的DHCPv6消息。默认前缀长度是64位。是否提供任意长度的前缀取决于特定ISP的策略和业务流程。ISP可以规定特殊要求需进行事先协商和离线申请。

SAR返回的响应报文还包含一个控制选项（Control Option），可用于存放SAR收到路由请求时所看到的SC的地址信息（经NAT映射后的IPv4地址和端口）以及其他控制信息。同时，SAR为SC创建IPv6地址和IPv4映射地址与端口号之间的映射。事实上，当一个地址请求报文到达SAR时，可能经过了多级NAT设备，此时SAR看到的地址请求报文的源地址和端口应该是最靠近SAR的那个NAT所映射的地址和端口。SAR将会把该地址和端口写入响应报文的Control Option字段，和IPv6的地址响应信息一起返回给SC。

当SAR返回RA时，相应的报文格式如下：

```
+------+-----+---------------+--------+
| IPv4 | UDP | Control Option | RA     |
+------+-----+---------------+--------+
```

由SAR分配给SC的IPv6地址应是属于该SAR所管理的地址空间的全局IPv6地址。该地址可以永久分配给SC。这样即使SC的IPv4地址是动态分配的，也可保持一个固定的IPv6地址。

类似地，SC也可以要求分配临时的IPv6地址。这样的地址会有一个生存期，超过生存期后，地址将被收回，除非SC提交延长地址生存期的请求。

一旦一个SAR给一个SC分配了IPv6地址/前缀，这个SAR就会在自己的映射表里面维护一个地址映射，将SC的IPv6地址和SC的IPv4映射地址和映射端口绑定在一起。该绑定可以随着SC的IPv4地址的改变而动态更新。此外，SAR要维护这个UDP隧道的管理信息。

4.3.2.3 NAT 类型的确定

前一部分介绍了一个简单的地址分配过程。当SC收到地址响应后，将把Control Option域中的IP地址和端口与自己发送RS时绑定的本地IP地址和端口进行比较。如果不匹配，则SC位于NAT域内；否则SC不需要使用Silkroad服务。

为了获得更高的传输效率，SC可以选择在SAR的帮助下确定它所使用的NAT类型。为此，SC发送“测试请求”，而SAR返回“测试应答”。基本过程可参照[RFC 3489]中描述（当然，具体实现非常灵活）。

SC可从同一个源地址和端口向SAR的另一个IPv4地址发送“测试请求”报文。如果应答报文内嵌的Control Option域中的地址和端口与首次应答的不同，SC就知道它位于对称NAT之后。

为了确定SC是否位于完全锥型NAT之后，SC可以发送一个带标志的“测试请求”，该标志是让SAR从不同于它收到“测试请求”的IP地址和端口发送应答。如果SC接到该应答，证明它是位于完全锥型NAT之后。

当然，SC发送的“测试请求”中的标志也可以让SAR从收到“测试请求”的相同IP地址和不同端口发送“测试应答”。此法用于探测SC是位于一个端口受限锥型NAT后，还是位于受限锥型NAT后。

一旦确定，NAT类型将被SC保存。

4.3.2.4 从 SC 到普通 IPv6 节点的报文传输

下图显示了一个SC和一个普通的IPv6节点之间的报文传输，其中A是NAT域中的SC。SAR1是A所选择的SAR。B是一个普通IPv6节点。

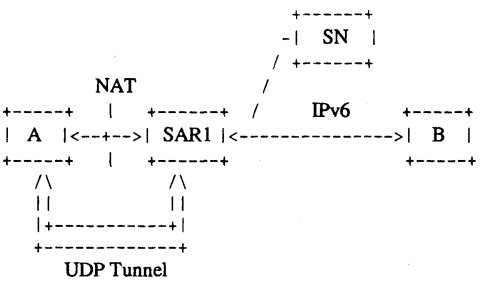


图3　从 SC 到普通 IPv6 节点的报文传输

假设A已经从SAR1处得到了它的IPv6地址，图3中每个实体的地址如下：

—A的私有IPv4地址和端口是1.0.0.1:1234

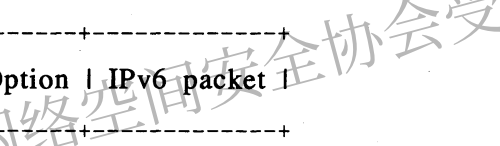
—A的IPv4映射地址和端口是2.0.0.2:5678

—A的IPv6地址是3ffe:8330:1::1234

—SAR1的IPv4地址和端口是3.0.0.3:5188

—B的IPv6地址是2001:250:f006:1::5678

A若要向B发送IPv6包，A首先要把这个IPv6包封装在IPv4的UDP数据包中，并把它用源地址和端口1.0.0.1:1234发送给SAR1的地址3.0.0.3:5188。此报文称作Silkrad包（4.3.2.2中所述的来自SAR的地址响应包以及4.3.2.3中所述的“测试请求”和“测试响应”包也是Silkroad包，详细说明见4.3.4），这种包的格式如下：



NAT会接收此数据包，用映射地址和端口2.0.0.2:5678替换UDP的源地址和端口1.0.0.1:1234，然后把包转发出去。

此包通过IPv4 UDP隧道被SAR1接收。SAR1查看IPv6目的地址并在当前的通信对等节点列表中检查这个IPv6地址对应的表项是否存在以及该表项是否有效。如果这个IPv6地址对应的表项不存在或无效，SAR1将登录它的SN去查找是否B是由另一SAR提供服务的SC。此查找过程与DNS的处理过程类似。SN将在自己的管理域中查找SAR的信息。如果没找到对应表项，它将查询上层SN。此地址解析过程只局限于一个ISP的网络中，这意味如果没有事先的约定，属于不同ISP的SN将不能相互协作。在图3所示的例子中，查询结果是B并非本ISP网络中的一个SC。所以，SAR1将丢掉IPv4和UDP报头以及Control Option域，将其余内容通过IPv6路由发向B的IPv6地址2001:250:f006:1::5678。在这种情况下，无法进行路由优化。

SAR1在SN的帮助下通过地址解析过程获得了B的路由信息后，它将更新通信对等节点列表：若B的表项在该表中不存在，则创建B的表项，并指明B为普通IPv6节点；否则，更新相关信息，并将状态设置为有效。

4.3.2.5　从普通 IPv6 节点到 SC 的报文传输

同样考虑图3中的场景。当B想要向A发送IPv6包时，B只需要遵循IPv6规则。因为A的IPv6地址属于SAR1的地址空间，并且SAR1是IPv6可达的，所以数据包可以通过IPv6传送给SAR1。此过程中无法进行路由优化。SAR1将查看其映射表中的地址绑定。在映射表中可以看到此报文的IPv6目的地址与A的IPv4映射地址和端口绑定。所以，SAR1将通过UDP隧道向A转发数据包。同时SAR1将查看通信对等节点列表。

如果存在B的表项，SAR1将更新其有效期，否则SAR1将为B添加一个表项。默认的有效期是30s。30s后此表项将被设为无效。SAR每次接到来自对等节点的数据包时，该对等节点的有效期会被重新设为30s。

4.3.2.6 两个 SC 间的报文传输

图4显示了两个SC（A和B）通过两个NAT（NAT1和NAT2）进行通信。SAR1是A所选择的SAR，而SAR2是B所选择的SAR。

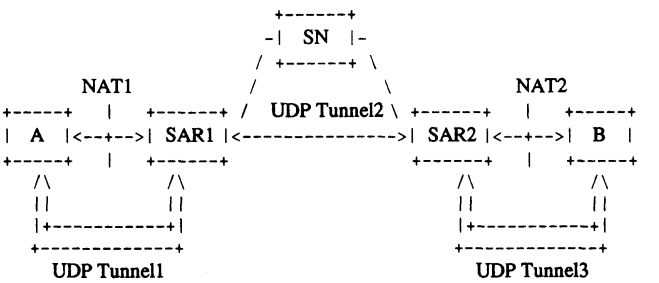


图4 两个 SC 之间的报文传输

假设A和B已经分别从SAR1和SAR2处获得了各自的IPv6地址。此例中，我们不考虑基于不同NAT类型的任何路由优化，路由优化将在第4.3.3部分介绍。我们还假设每个实体的地址如下：

—A的私有IPv4地址和端口是1.0.0.1:1234

—A的IPv4映射地址和端口是2.0.0.2:5678

—A的IPv6地址是3ffe:8330:1::1234

—SAR1的IPv4地址和端口是3.0.0.3:5188

—SAR2的IPv4地址和端口是4.0.0.4:5188

—B的私有IPv4地址和端口是5.0.0.5:1234

—B的IPv4映射地址和端口是6.0.0.6:5678

—B的IPv6地址是2001:250:f006:1::5678

A已经知道了B的IPv6地址（比如通过DNS），并开始与B通信。数据包将从A的IPv6地址（3ffe:8330:1::1234）发送到B的IPv6地址（2001:250:f006:1::5678）。从A到SAR1的传输过程与4.3.2.4节的说明相同。

SAR1检查IPv6目的地址，将发现B是SAR2所服务的一个SC。SAR1可以从通信对等节点列表中或在SN的帮助下通过地址解析过程获得相关路由信息。如果SAR1在SN的帮助下通过地址解析过程获得了B的路由信息，那么它将更新通信节点列表：若列表中不存在B的表项，则为B创建一个表项，否则更新表项内容并将状态设为有效。如果SAR1是从通信对等节点列表中获得B的路由信息，则需重置该表项的有效期。

确定了SAR2的地址后，SAR1将把IPv6包和Control Option域一起通过隧道发送给SAR2的IPv4地址4.0.0.4:5188。SAR1和SAR2的传输可以遵循不同的方式。它们可以用另一条UDP隧道传输IPv6包，如图4中的UDP隧道2。这种方法的优点是转发之前SAR1的处理比较简单，只需用3.0.0.3:5188替换UDP源地址和端口，并用4.0.0.4:5188替换UDP目的地址和端口。然而，UDP封装将带来一定的开销。Silkroad还可以像传统的隧道机制那样，使用IPv6-in-IPv4封装，把IPv6包作为IPv4包的载荷在SAR之间发送。但是这种方法不支持Control Option域，而且SAR1在转发之前必须拆包并重新封装。ISP可以指定SAR之间的传输方式。默认的处理是当包中有Control Option域时，SAR通过UDP隧道转发包；否则，使用传统隧道转发。

事实上，Control Option域一般只出现于一次会话的前几个报文中。在Silkroad隧道机制中，SAR需要监听UDP和IP两种封装。但是无论在SAR之间采用哪种传输方法，SC只需要支持UDP封装。

为简单起见，在下面的描述中，我们假设SAR之间都使用UDP隧道转发报文。

当通过UDP隧道2接收到数据包时，SAR2会发现B是自己所服务的SC，并且映射表中显示出IPv6目的地址2001:250:f006:1::5678与B的IPv4映射地址和端口6.0.0.6:5678相关联。所以，SAR2将用4.0.0.4:5188替换UDP的源地址和端口，并用6.0.0.6:5678替换UDP的目的地址和端口，然后通过UDP隧道3转发数据包。同时SAR2将查看当前通信对等节点列表。如果有A的表项，SAR2将更新表项内容，否则SAR2将为A新添一个表项。

NAT2会接收到这个消息，并利用现有的映射把6.0.0.6:5678重置为5.0.0.5:1234，并把数据包转发给B。然后，数据包将通过IPv4 UDP隧道3被B接收。

注意SAR1和SAR2可能是同一个SAR，此时，UDP隧道2将不存在。

4.3.3 路由优化

为了达到更高的传输效率，Silkroad机制在某些情况下可以进行路由优化。路由优化是可选的，ISP可以决定是否为其Silkroad用户提供路由优化。通常，如果不同ISP间无事先约定的话，路由优化只在同一ISP所辖的SC之间进行。

4.3.3.1 两个 SC 间的路由优化

如图5所示，假设A和B是两个SC，分别位于NAT1、NAT2的后面。SAR1、SAR2分别是A和B所选择的SAR。如4.3.2.6所述，SAR1和SAR2可能是同一个SAR。此时，UDP隧道2将不存在。

事实上，A和B都能识别Silkroad报文。因此，A和B可以直接互通，条件是NAT1和 NAT2不能同为对称NAT。当然，在直接通信前，A、B间需要借助SAR1、SAR2交换必要的参数信息。

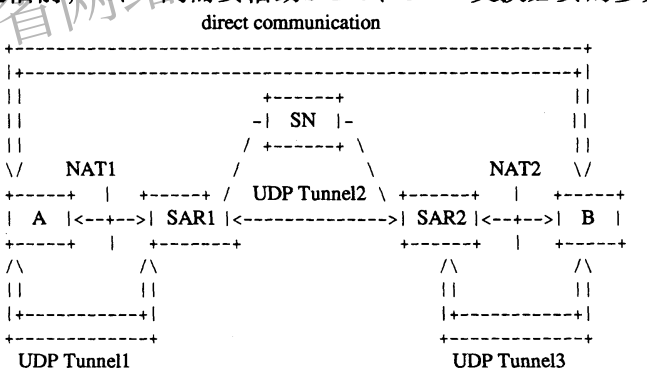


图5 两个 SC 间的路由优化

假设A想要与B通信。如果A想进行路由优化，A首先应在SAR1的协助下确定NAT1的NAT类型（如4.3.2.3所述）。然后，A将自己的映射地址和映射端口以及NAT类型封装到Silkroad报文的Control Option域内发给B。如果NAT1和NAT2都是完全锥型NAT，则B在响应报文的Control Option字段内签入自己的映射地址和映射端口，通过UDP封装直接返回给A，而后A、B即可通过UDP隧道直接通信。如果NAT1和NAT2都是对称NAT，则A、B无法在无SAR协助的情况下进行直接通信。在其他的情况下（A、B不都是完全锥型NAT，也不都是对称NAT），则A、B需要发送更多的测试报文，在非完全锥型的NAT设备上建立对应的映射关系，而后才能直接通信。在这种情况下，建议用户只在需要传输大量数据时进行路由优化，如果交互信息量比较少，则不需要发送更多的测试报文来进行路由优化。

4.3.3.2 同一链路上两个 SC 间的路由优化

如图6所示，假设A、B为同一链路上的两个SC，均通过 NAT1连入Internet， 默认SAR为SAR1。这里我们不考虑NAT1的类型，只介绍A、B间如何进行路由优化，以避免所有的报文都经过SAR1、NAT1转发。

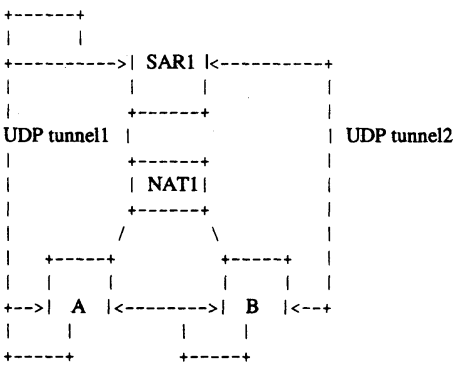


图6 同一链路上两个 SC 间的路由优化

我们假设A和B已经分别从SAR1获得了它们的IPv6地址。

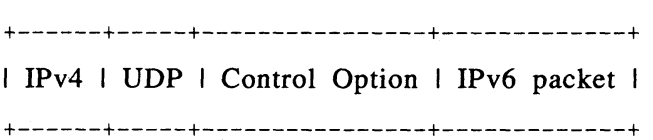
当然，如果A和B都用相同的IPv4映射地址，那么它们可能位于同一个NAT域中。然而，即使IPv4映射地址不同，它们也可能位于同一个NAT域中。另一方面，即使位于同一个NAT域内，相互之间也不一定能直接通信。所以，必须采取一些措施来确定它们之间是否是直接可达的。有许多方法可以达到这一目的，这里只是给出一种可行方案。

如果A想要判断它和B是否在同一链路上，A可以把自己的私有地址放在数据包的Control Option域中，通过NAT1和SAR1发送给B。当B接收到此包时，同时向SAR1以及A的私有地址返回报文，内含自己的私有地址。如果A能够收到这两个报文，则A可以确认它和B在同一链路上，可以直接互通。后继报文将利用A、B的私有地址封装，只在本地链路上传播，而不必再经过SAR1和NAT1转发。

4.3.4 数据包格式

本节将详细介绍Silkroad包的格式。

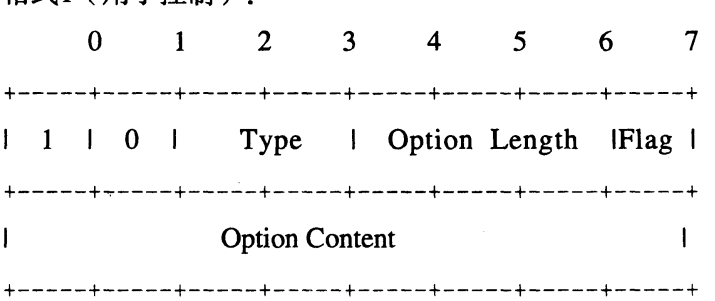
Silkroad包是作为IPv4的UDP报文的载荷进行发送的。为了控制和优化传输，在UDP载荷的前几个字节中插入了Control Option域：



Control Option域的第一个比特被设为1，这是用来区别带Control Option域的UDP封装和简单的UDP封装。在简单的UDP封装中，头4位应该是IPv6协议的标志“0110”。

Control Option有两种不同的格式：一种用于控制，一种用于测试。

格式1（用于控制）：



此格式中的Type占2比特，用于指示此控制选项的内容类型。我们目前定义了3种类型：

11——映射地址和端口

10——私有地址和端口

01——NAT的类型

00——未使用

Option Length是3比特，指示选项内容的长度，以字节为单位。

1比特的标志位用来指示本报文中在该选项后是否还有其他选项。这意味着一个报文的Control Option域可以携带多个选项。

格式2（用于测试）：

```
      0     1     2     3     4     5     6     7
+-----+-----+-----+-----+-----+-----+-----+-----+
|  1  |  1  |        Type       |   Unused  |Flag |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

此格式中的Type占3比特，指示选项的内容类型。我们目前定义了5种类型：

111——测试请求，无需应答；

110——测试请求，需要应答；

101——测试请求，需要使用不同于接收到“测试请求”的IP地址和端口返回应答；

100——测试请求，需要使用和收到“测试请求”相同的IP地址和不同端口返回应答；

011——测试应答。

010 / 001 / 000——未使用

（111类型的报文可用于发送keep-alive报文，维持连接的有效性）

1比特的标志位用来指示本报文中在该选项后是否还有其他选项。

4.3.5 隧道生存期

由于NAT将删除长时间没有数据流的会话，可能导致SC和SAR之间的UDP隧道无法使用。因此，SC必须周期性的通过IPv6-in-UDP隧道向SAR发送数据包，以保证NAT继续维持UDP会话，从而避免SC和SAR之间的IPv6-in-UDP数据流在NAT处被中断。

同时，SAR为维护隧道需要消耗一定的资源。如4.3.2.2中所提到的，IPv6地址可以永久地分配给某个SC。用这种方法，尽管SC的IPv4地址是动态的，但仍能保持一个静态的IPv6地址。临时IPv6地址的生存期应长于SC的动态IPv4连接的生存期。对于临时分配的IPv6地址而言，如果仅根据IPv6地址生命期进行管理，会导致SAR维护过多的过期隧道。SAR和SC上应该有keep-alive机制，SC周期性向SAR发送数据包，在维持NAT映射的同时，也用于维持UDP隧道。如果在超过刷新间隔的一段时间内被分配的地址上没有“刷新”数据流，SAR将自动删除相应的隧道，地址绑定也将被删除或设为休眠。

4.3.6 Silkroad 对移动的支持

Silkroad能够支持在同一ISP网络内的移动。假设A是一个SC，SAR1是A所选择的家乡SAR，A从SAR1处获得它的家乡地址。当A移动到SAR2的访问域中时，A将在SAR2上进行注册，并从SAR2处获得一个转交地址。如果A想要保持它的IPv6地址不变，它可以在SAR1上注册它的新地址，并让SAR1作为家乡代理，把所有发往A的数据包传送给到A的当前位置。当然，A和SAR1必须共享一个安全的绑定，并能使用MD5（RFC1321）为注册请求创建不可伪造的数字签名。

如无事先的约定，Silkroad将不支持不同ISP之间的移动。

4.3.7 安全考虑

通常，Silkroad服务被局限在一个ISP的范围里，ISP应该在它的边界上实行IPv4入口过滤，应该阻止SAR的地址在ISP外部被用作源地址。ISP也应该对它的用户实行IPv4入口过滤，尤其是SC，使得它们不能伪造包的IPv4源地址。

在SN和SAR的交互中，可采用安全的SNMP[RFC2572，RFC2574，RFC 2575]。此外，若要使用更简单的基于RSH指令的方法，则可以采用标准的IPsec机制[RFC 2401]。

需要指出的是SC和SAR间通讯过程的安全问题。由于两者间的隧道不是面向连接的，因此如果SC单方面强行释放了隧道，SAR会继续转发隧道数据，导致数据包被错误地发到原来的IPv4地址。Silkroad机制采用了keep-alive机制来解决这个问题。当检测到隧道失效后，SAR立即停止向离线用户转发数据包。另一方面，SC必须确保它使用的Silkroad隧道是有效的，这可以通过检查是否定期收到来自SAR的数据包来验证。

NAT使SC失去了端到端的连接，但却可以获得更高的安全性，因为NAT可以被看作是一种防火墙。在Silkroad的帮助下，SC在IPv6网络中是可达的，这就使其成为攻击者的潜在目标。为了确保通信的安全性，SC可使用IP安全服务，如使用全局IPv6连接的AH或ESP。

与Teredo相同的是，Silkroad很难发现中间人攻击，因为这种攻击与NAT的行为类似。SC和SAR之间认证过程的实现将有效地阻止中间人攻击。一种攻破这种保护的方法是离线的字典攻击（dictionary attack）。如果认证过程经过对称或非对称加密系统加密，中间人攻击就很难实现。

中 华 人 民 共 和 国

通 信 行 业 标 准

IPv6 网络技术要求

——面向网络地址翻译（NAT）用户的 IPv6 隧道技术

YD/T 1635-2007

*

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：2　　2007 年 6 月北京第 1 次印刷

字数：52 千字

ISBN 978-7-115-1390/07-53

定价：15 元