doi:10.3969/j.issn.1000-436x.2012.z2.043

适用全类型 NAT 用户的隧道过渡机制

陆年锋，王振兴，刘慧生
(解放军信息工程大学 数学工程与先进计算国家重点实验室，河南 郑州 450002)

摘 要：深入分析当前隧道机制的优缺点，在 TSP 机制的基础上，面向不同的场景，提出一种更高效、适用性更广的 TrNAT 过渡隧道机制。TrNAT 借助 MIPv6 通信最优化思想实现部分 NAT 用户间的通信优化，同时利用 Shim6 机制实现隧道机制的路径冗余及改善 6 to 4 隧道路由聚集。最后在 linux 系统上实现 TrNAT 隧道机制并进行了实验验证。

关键词：NAT 穿透；Shim6 机制；隧道过渡机制；通信优化
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2012)Z2-0294-07

Tunnel transition mechanism support all types of NAT users

LU Nian-feng, WANG Zhen-xing, LIU Hui-sheng
(State Key Laboratory of Mathematic Engineering and Advanced Computing, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: Through the analied of current tunneling mechanism, a more efficient and applicable tunneling mechanism called TrNAT based on TSP to face different scenarios was proposed. Optimization of the communications between NAT users was achieved with ideology of MIPv6. Then, Shim6 was utilized to realize route redundancy and also improved the route aggregation of 6 to 4 tunnel. Finally, TrNAT tunneling mechanism was implemented on Linux kernel and validated by experiments.

Key words: NAT traversal; Shim6 mechanism; tunnel transition mechanism; communication optimization

1 引言

随着物联网、云计算、移动互联网概念的提出，IP 地址的需求量明显上涨，然而在 2011 年 2 月 3 日，ICANN 宣布全球 IPv4 地址已经分配完毕；同年 4 月 APNIC 宣布亚太地区 IPv4 地址资源仅剩下最后一组。IP 地址的匮乏已经严重制约着互联网的发展，目前，欧、美、日本等国正按照既定计划积极部署 IPv6，中国于 2012 年 5 月 4 日正式发布了《互联网行业“十二五”发展规划》将 IPv6 过渡正式提上发展日程。但是由于 IPv4 覆盖范围广、网络结构复杂以及 IPv4 和 IPv6 的不兼容等特点，使得从 IPv4 过渡到 IPv6 是一个长期、复杂的过程，当前正处于从 IPv4 网络向 IPv6 网络过渡初期。

为了实现平滑过渡，IETF 提出了双栈机制、隧道机制和协议翻译机制这 3 大类过渡机制。其中，双栈机制是基础；协议翻译机制最具有部署情景，但需要升级现有的网络设施；隧道机制由于不需要更改现有的网络设施，在当前过渡阶段中广泛使用。

但是现有的隧道机制在当前过渡初期的应用中存在一些不足，主要表现为不能穿透 NAT[1]设备。由于 IPv4 地址的严重匮乏，NAT 设备在当前的网络中广泛部署，绝大多数 NAT 设备只支持 TCP、UDP 和 ICMP 等常见协议的转发，不支持 IPv6-in-IPv4 协议转发，因此，基于该类协议的隧道机制应用场景十分有限，比如 6 to 4[2]和 ISATAP[3]。在现阶段，如何有效解决 NAT 用户与 IPv6 网络的

收稿日期：2012-10-23

互联互通显得格外重要。

本文的目的是在复杂过渡环境中，为 NAT 用户通过隧道机制接入 IPv6 网络提供一种更加高效、适用性更广的 TrNAT 隧道机制。

2　相关研究

2.1　Shim6 协议及研究情况

Shim6[4]（Level 3 Shim for IPv6）协议是实现多宿机制最有应用情景的多宿协议，其关键思想是修改终端网络协议栈，在 IP 路由子层与 IP 终端子层之间插入 Shim6 层（如图 1 所示），通过该层建立 IP 地址标识符和定位符的映射关系，解决 IP 语义过载问题。在通信过程中 IP 路由子层的变化对应用层透明，通过 Shim6 层映射一直保持上层标识符的不变，因此即使 IP 路由子层发生变化，原有通信能够继续维持。目前 Shim6 实现协议主要有 LinShim6 和 MipShim6 2 种[5]。

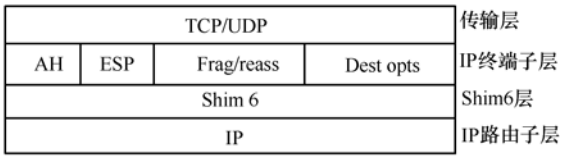


图 1　Shim6 层在协议栈中的位置

2.2　典型的隧道机制及研究情况

当前典型的隧道机制有：Teredo 隧道机制[6]、TSP 机制[7]和 6 to 4 隧道机制。

Teredo 隧道机制是当前唯一成为 RFC 标准，面向 NAT 用户的隧道机制。虽然 Teredo 支持穿透 NAT，解决了部分 NAT 用户通过隧道访问 IPv6 的问题，但是 Teredo 同时存在不足：1）不支持穿透对称 NAT；2）不能为用户分配固定 IPv6 地址，不支持端到端的通信；3）不能有效阻止非法用户利用 Teredo 中继器与 IPv6 网络互联，存在安全问题。

TSP 机制并不是隧道机制，而是用来协商建立隧道参数的协商机制。TSP 支持多种隧道封装协议，如 IPv6-any-IPv4、IPv6-over-IPv4、IPv4-over-IPv6 和 IPv6-over-udp-in-IPv4，封装协议可以由客户指定或通过自动协商完成；同时 TSP 还支持判断路径上 NAT 设备的存在。但是由于采用 C/S 模式，使得所有的数据分组都通过服务器端进行转发，无疑加重了服务器的负荷，成为网络通信的瓶颈。

6 to 4 隧道机制使用全球统一的 6 to 4 地址前缀在 6 to 4 路由器之间建立隧道，实现跨越 IPv4 网络的 6 to 4 孤岛之间通信，运营商在未获得 IPv6 前缀的情况下也可以应用 6 to 4 技术。6 to 4 隧道机制使用特殊格式的 IPv6 地址，隧道端点可以直接从目的地址提取得到 IPv4 地址，因此 6 to 4 主机间的通信，不需要像 TSP、Teredo 机制需要经过服务器解封后进行二次封装。但是 6 to 4 使用特殊的地址格式，与 IPv6 地址可聚集的机制相违背，给路由聚集造成了很大困难；而 6 to 4 最大的缺点是要求 6 to 4 主机拥有全球可路由 IPv4 地址，这也是 6 to 4 机制不能被广泛部署的主要原因。

目前隧道优化大多针对单一隧道机制进行，通过隧道结合达到优化目的的文献和资料并不多见。解利忠在文献[8]中提出了一种利用精简的 Shim6 协议，结合 TB 机制和 6 to 4 隧道机制来达到隧道优化的目的。文中利用裁剪的 Shim6 协议，融合 2 种机制。但是文中提到端到端双栈节点使用 6 to 4 隧道进行通信，由于 6 to 4 隧道采用 IPv6-in-IPv4 的封装协议且要求隧道端点具有独立的全球可路由 IPv4 地址，加上 NAT 的广泛部署，因此文献[8]中提出的隧道优化机制应用场景十分有限。

吴贤国在文献[9]中提出了面向所有 NAT 用户的 Silkroad 隧道机制。但是该机制同样基于 C/S 模型，虽然对部分 NAT 用户之间进行了通信优化，但是服务器端难免还会成为网络通信的瓶颈。

Joseph Davies 在文献[10]中对 Teredo 隧道具体的通信机制进行了优化，其方法具有一定的参考价值。

3　TrNAT 隧道方案设计

6 to 4 隧道机制不需要经过二次封装，具有更高的通信效率，但是不支持穿透 NAT；TSP 隧道支持穿透 NAT，但是需要经过 TSP 服务器二次封装。本节以 TSP 机制为基础，综合 TSP 机制和 6 to 4 机制优点，提出 TrNAT 的方案，并对 NAT 客户端间的通信进行一定的优化。

3.1　TrNAT 总体设计

TrNAT 定义了与 TSP 机制相对应的 3 种通信实体：客户端、服务器和导航器。客户端与导航器的功能与 TSP 类似，其中，区别较大的是 TrNAT 服务器实现 TSP 服务器和 6 to 4 服务器 2 种服务器功能。

在 TrNAT 隧道机制中，如果客户端位于 NAT 后，建立 TSP 隧道，采用 IPv6-in-UDP-over-IPv4 隧道封装协议穿透 NAT；否则同时建立 6 to 4 隧道和 TSP 隧道，TSP 隧道采用 IPv6-over-IPv4 隧道封

装协议。因此，如果不是 NAT 主机，将同时分配到 6 to 4 地址及 TSP 地址；其他主机由于无法构建 6 to 4 隧道，只分配 TSP 地址。其基本框架如图 2 所示，此时图中导航器与服务器实现同一个功能。

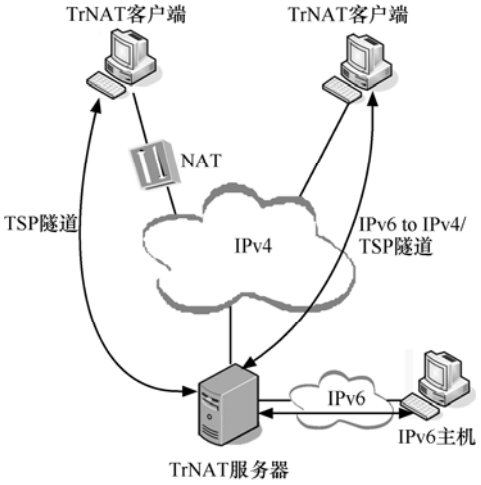


图 2 TrNAT 隧道基本示意

TSP 机制不支持分配固定的 IPv6 地址，用户每次和服务器初始化通信时往往被 NAT 转换成不同的外部地址或外部端口，因此丧失了端到端通信的优势。端到端通信的基础是双方拥有固定的 IP 地址，本文通过在 TrNAT 服务器端增加一个有状态的客户端 IPv6 地址与隧道参数的映射，实现给客户端分配固定的 IP 地址。映射关系的创建在 TrNAT 客户端初始化时完成。映射关系通过如下四元关系表示：{IPv6↔(IPv4，UDP_port，|NAT|)}，其中，IPv6 表示分配给 TrNAT 客户端 A 的 IPv6 地址；IPv4 表示客户端 A 的外部地址，即 NAT 公有地址；UDP_port 表示客户端外部端口，即 NAT 连接公有地址一侧接口上的端口号；|NAT|表示 A 位于哪种 NAT 类型后。当服务器收到 IPv6 数据分组后，根据 IPv6_dst 查找映射表，如果存在对应的映射关系，就用得到的隧道参数封装数据分组发给客户端，若下次隧道参数发生变化，只需要更改映射关系，客户端的 IPv6 地址保持不变。

TrNAT 服务器的有状态特性可以提高安全性。TrNAT 服务器通过检查数据分组与映射关系可以判定数据分组是否合法。当 TrNAT 服务器收到 IPv6 数据分组时，查看是否存在该 IPv6 数据分组目的地址的映射关系，如果不存在，可判定该数据分组为非法数据分组，将其丢弃；当 TrNAT 服务器收到 IPv4 数据分组时，查看是否存在该数据分组 IPv4 源地址的隧道参数映射，如果不存在，可判定该数据分组为非法数据分组，将其丢弃。而映射关系的合法性，通过初始化过程中客户端与导航器进行相互的身份认证来保证，只有身份认证通过的客户端才能建立有效的映射关系。

3.2 TrNAT 中 NAT 设备类型的判定

根据 NAT 地址及端口映射关系建立的不同，NAT 可以分为全克隆 NAT、限制性 NAT 和对称 NAT。由于 NAT 的存在会影响隧道参数的选择，因此在客户端初始化过程中需要完成 NAT 设备类型的判定。虽然 TSP 机制支持探测路径上 NAT 设备的存在性，但是不支持判定 NAT 设备的类型，需要在 TrNAT 机制中增加 NAT 类型的判定功能。

TSP 机制的一个优势是支持通过增加或删除 XML 项实现信令协议的灵活扩展，为了支持判明路径上 NAT 设备类型，本文在信令协议的 TSP client 项中增加一个新的 NAT 属性，它包含 4 个值（用|NAT|表示 NAT 属性的值），其中，|NAT|=00 表示未设置 NAT 类型；|NAT|=01 表示全克隆 NAT；|NAT|=10 表示限制性 NAT；|NAT|=11 表示对称 NAT。NAT 设备类型判定的基本示意如图 3 所示。

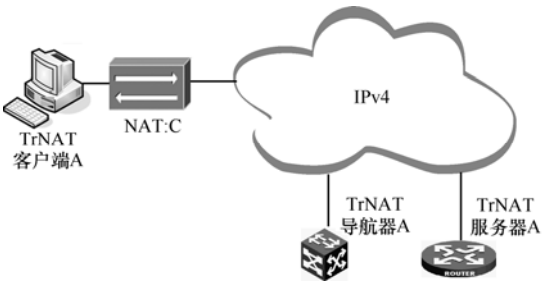


图 3 路径上 NAT 类型判断示意

1) 首先比较 TSP 机制中源地址 IPv4_src 和 TSP 消息中的地址来判明路径上是否存在 NAT。如果两者地址相同，说明路径上不存在 NAT，探测结束，A 将|NAT|值设为 00，同时 A 将建立 6 to 4 隧道及 TSP 隧道，分配 6 to 4 地址及 TSP 地址；如果不同，则说明存在 NAT，需要进一步判断 NAT 类型。

2) 客户端 A 向服务器 S 发送|NAT|=01 的 RS 报文。

3) 由于|NAT|=01，S 选择从其他 IPv4 地址发送 RA。如果 A 收到报文说明其位于全克隆 NAT 后，探测结束，A 将 NAT 值设为 01。如果 A 在超时时间内未收到 RA 报文，A 重新发送|NAT|=00 的 RS

报文。

4) 由于|NAT|=00，S 用收到数据分组的 IPv4 目的地址作为 IPv4_src 发送 RA 报文，如果 A 收到报文，则位于限制性 NAT 后，探测结束，A 将 NAT 值设为 10。

5) A 为了确定是否在对称 NAT 后，A 会发送另一个 RS 报文到 TrNAT 代理 B。

6) B 用收到报文的 IPv4 目的地址回复 RA 报文，A 通过比较 S 和 B 返回的 RA 报文，如果包含的隧道参数相同，则说明不在对称 NAT 后，|NAT|=10；否者位于对称 NAT 后，探测结束，A 将 NAT 值设为 11。

此时客户端已经完成初始化工作，服务器端保存了一份客户端隧道参数的有状态映射关系。

3.3 TrNAT 中客户端的通信优化

以 TSP 机制为基础的 TrNAT 隧道机制中节点间的通信类似于移动 IPv6 节点间的通信，当离开家乡网络的移动节点跟通信节点通信时，需要经过家乡代理的转发，移动 IPv6 协议为了达到通信的最优化，减少通信的延迟，提出了通信最优化方案，在家乡代理的协助下完成移动节点和通信节点之间的可达性检测后，移动节点跟通信节点就不再需要经过家乡代理的二次转发，直接进行通信。

在本文的场景中，为了使得 NAT 主机间能够直接进行通信，NAT 主机之间首先判断能否进行通信优化，其次需要进行可达性检测，即 NAT 主机获取对方的隧道参数。因此路由可达性过程相当于在 TrNAT 服务器参与下，NAT 主机交换隧道参数并进行可达性验证的过程。由于 NAT 设备对于外网数据分组的转发具有一定的限制，可达性验证的过程就是在双方 NAT 中建立一条映射关系，使得对方的数据分组能够顺利转发到己方内网主机。虽然在节点初始化过程中，在服务器端保存了一份隧道参数，但是该隧道参数是针对目的地址及端口，是服务器的 NAT 映射，而不是通信对端的 NAT 映射，初始化过程中创建的隧道参数并不能用于 NAT 主机通信优化中，因此需要获取针对通信对端的隧道参数。

上文提到的 NAT 可以分为 3 种类型，由于不同 NAT 类型对于转发到内网数据分组限制的不同，并不是所有的 NAT 类型的客户端都能进行通信优化。尤其是对称 NAT，内部地址/端口、通信目的地址/端口，四者中的任何一个发生变化，NAT 都会产生一条不同的映射，现有的技术并没有完全解决穿透对称 NAT 的问题，大多方法是基于端口的预测和端口遍历，消耗资源的同时也不能保证 100%的成功，因此本文对于限制最严的对称 NAT 并不进行通信优化。通过排列组合，上文提到的 3 种 NAT 类型可以产生 9 种通信情况，分别是全克隆 NAT→全克隆 NAT、全克隆 NAT↔限制性 NAT、全克隆 NAT↔对称 NAT、限制性 NAT→限制性 NAT、限制性 NAT↔对称 NAT、对称 NAT→对称 NAT。其中，限制性 NAT↔对称 NAT、对称 NAT→对称 NAT 这 3 种类型不能进行通信优化。通信优化示意如图 4 所示，其中，A 表示通信优化发起端，B 表示通信优化响应端，S 表示 TrNAT 服务器，IPv6A、IPv6B 表示 A、B 的 IPv6 地址，IPv4A、PortA 表示 A 内侧 IP 及端口，IPv4A'、PortA'表示 C 公网侧 IP 及端口，IPv4B、PortB 表示 B 内侧 IP 及端口，IPv4B'、PortB'表示 D 公网侧的 IP 及端口。

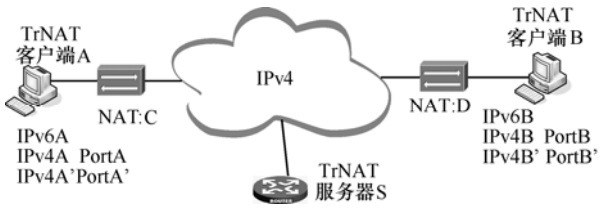


图 4　通信优化示意

客户端 A、B 完成初始化后，服务器端保存了客户端 A、B 的隧道参数，NAT:C，NAT:D 允许转发来自服务器 S 的数据分组到相应的客户端 A、B。根据客户发起端 NAT 类型的不同，通信优化过程分为如下 3 种。

① 全克隆 NAT→其他类型 NAT

1) A 向 B 发送空数据分组，其中，IPv6_src=IPv6A，IPv6_dst=IPv6B。

2) S 截获 A 的数据分组后通过 IPv6_src、IPv6_dst 查找映射表，得到 A、B 的隧道参数，同时 S 判断得到 A 的|NAT|=01，S 将包含 A 隧道参数的数据分组用 UDP 封装后转发给 B，其中，IPv4_dst=IPv4B'，UDP_dst=PortB'。

3) B 收到步骤 2)的数据分组得到 A 的隧道参数，向 A 发送空数据分组，其中，IPv4_dst=IPv4A'，UDP_dst=PortA'，IPv6_src=IPv6B，IPv6_dst=IPv6A。此时 D 产生一条转发来自 A 数据分组到 B 的映射。

4) 由于 C 是全锥型 NAT，因此 C 将步骤 3)的数据分组转发给 A，A 从 IPv4_src 和 UDP_src 中得

到 B 的隧道参数。隧道参数交换完成。

② 非全克隆 NAT→全克隆 NAT

1) 首先，A 向 B 发送空的数据分组，IPv6_src= IPv6A，IPv6_dst=IPv6B。

2) S 截获 A 的数据分组后通过 IPv6_src、IPv6_dst 查找映射表，得到 A、B 的隧道参数，S 判断得到 A 的|NAT|≠01，且 B 的|NAT|=01。S 向 A 发送包含 B 隧道参数的 UDP 数据分组，其中，IPv4_dst=IPv4A'，UDP_dst=PortA'。

3) A 收到步骤 2)的数据分组得到 B 的隧道参数，向 B 发送包含自身隧道的 UDP 数据分组，其中，IPv4_dst=IPv4B'，UDP_dst=PortB'，IPv6_src= IPv6A，IPv6_dst=IPv6B。此时 C 产生一条转发来自 B 数据分组到 A 的映射。

4) 由于 D 是全克隆 NAT，将步骤 3)的数据分组转发给 B，B 得到 A 的隧道参数，向 A 发送 UDP 数据分组，其中，IPv4_dst=IPv4A'，UDP_dst=PortA'，IPv6_src=IPv6B，IPv6_dst=IPv6A。

5) 经过步骤 3)后，C 将步骤 4)的数据分组转发给 A，A 确认 B 已经获得自己的隧道参数。隧道参数交换完成。

③ 限制性 NAT→限制性 NAT

1) 首先 A 向 B 发送空的数据分组，其中，IPv6_ src=IPv6A，IPv6_dst=IPv6B。

2) S 截获 A 的数据分组后通过 IPv6_src、IPv6_dst 查找映射表，得到 A、B 的隧道参数，S 判断得到 A 的|NAT|=10，B 的|NAT|=10。

3) S 将 B 的隧道参数用 UDP 封装后发给 A，其中，IPv4_dst=IPv4A'，UDP_dst=PortA'。

4) 同时 S 将 A 的隧道参数用 UDP 封装后发给 B，其中，IPv4_dst=IPv4B'，UDP_dst=PortB'。

5) A 收到来自 S 的数据分组，得到 B 的隧道参数，向 B 发送 UDP 数据分组，其中，IPv4_dst= IPv4B'，UDP_dst=PortB'，IPv6_src=IPv6A，IPv6_dst= IPv6B。此时 C 产生一条转发来自 B 数据分组到 A 的映射。但是由于在 D 上没有转发 A 数据分组到 B 的映射，因此 D 将丢弃该数据分组。

6) B 收到来自 S 的数据分组，得到 A 的隧道参数，向 A 发送 UDP 数据分组，其中，IPv4_dst= IPv4A'，UDP_dst=PortA'，IPv6_src=I Pv6A，IPv6_dst =IPv6B。此时 D 产生转发来自 A 的数据分组到 B 的映射。

7) 经过步骤 5)，C 将步骤 6)的数据分组转发给 A，此时 A 可以确认 B 已经收到自己的隧道参数。隧道参数优化完成。

3.4 TrNAT 中实现路径冗余及改善 6 to 4 路由聚集

对于位于 NAT 后的主机，由于不能建立 6 to 4 隧道，仅使用 TSP 地址通过 TSP 建立隧道收发数据。连接到不同的 TrNAT 服务器，TrNAT 主机将分配到不同的 TSP 地址，因此双栈主机可以使用 Shim6 多宿协议实现路径冗余，将一个 TSP 地址作为标识符保持不变，使用不同的 TSP 地址作为定位符，通过不同的 TrNAT 服务器转发数据从而维持上层通信。举例说明如图 5 所示（本节的举例说明只针对上层应用程序，如果考虑完整的通信过程，需要使用 IPv4 报文封装原有的 IPv6 报文），假设 TrNAT 主机连接到 2 个 TrNAT 服务器，分配到 2001:1001::1:10 和 2001:1002::1:10 2 个地址，通信对端 IPv6 主机地址为 2001:1011::1:10。两主机通过 Shim6 4 次握手协议将 2001:1001::1:10 与 2001:1011::1:10 作为标识符对。假设 TrNAT 服务器 1 出现问题，Shim6 层检测到链路失效，将使用 2001:1002::1:10 作为定位符发送数据分组，而对于上层通信其标识符没有改变，原有通信继续维持。

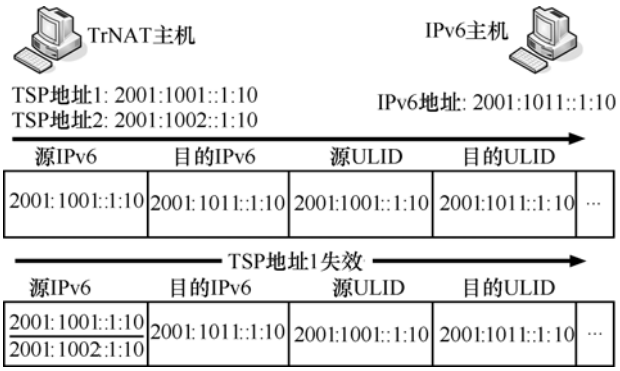


图 5 NAT 主机使用 Shim6 协议通信过程

由于 6 to 4 隧道不需要经过二次封装，具有更好的数据转发效率，但是 6 to 4 地址不支持地址聚合，通过利用 Shim6 多宿机制，将 TSP 地址作为 6 to 4 主机的标识符，将 6 to 4 地址作为 6 to 4 主机的定位符。对于 Shim6 层以上的通信，由于标识符保持不变，原有通信不会中断；而 Shim6 层以下的 IP 路由层，将 6 to 4 地址作为定位符，通过 6 to 4 隧道转发数据，提高效率的同时改善了路由聚集。

首先通过 Shim6 建立标识符与定位符映射的 4 次握手协议，建立双方主机地址对的映射关系。如

果双方主机都支持 Shim6 多宿协议，则握手成功，6 to 4 主机使用 6 to 4 地址作为定位符，TSP 地址作为标识符进行通信，通过 6 to 4 隧道收发数据；否则使用 6 to 4 地址作为标识符，使用 TSP 地址作为定位符，通过 TSP 建立的隧道收发数据，由于不存在 NAT，可以用 3.3 节中非全克隆 NAT→全克隆 NAT 进行通信优化。举例说明如图 6 所示，假设 TrNAT 主机配置了公共 IPv4 地址 157.54.0.1，该主机自动配置生成 6 to 4 地址为 2002:9D36:1::9D36:1；TrNAT 服务器为客户端分配的 TSP 地址为 2001:1001::1:10。通信对端是 IPv6 多宿主机，其地址为 2001:1010::1:10。此时 IPv6 数据分组的源地址为 6 to 4 地址，源 ULID 为 TSP 地址，该数据分组通过 6 to 4 隧道转发。

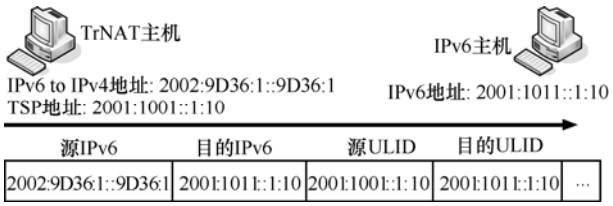


图 6　非 NAT 主机使用 Shim6 协议通信过程

4　TrNAT 实现及验证

本文 TrNAT 方案中涉及到的客户端及服务器端在 ubuntu 8.04 linux 内核版本为 2.6.24-22 上实现，为了融合 2 种隧道机制，通过修改 LinShim6 0.9 版本的源代码实现 Shim6 协议。TrNAT 实验在实验室环境中模拟互联网正常流量下进行，其基本架构如图 7 所示。

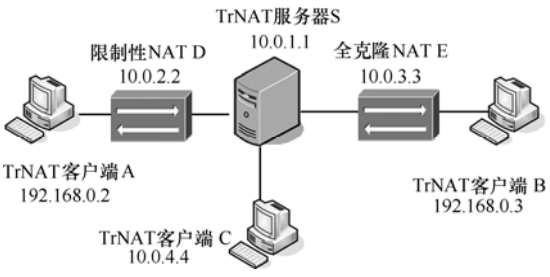


图 7　TrNAT 实现架构

为了简化操作，客户端地址由手动配置，客户端 C 由于不存在 NAT，分配到 6 to 4 地址为 2002:A00: 404::A00:404，TSP 地址为 2001:1001::4；客户端 A 分配到 TSP 地址为 2001:1001::2；客户端 B 分配到 TSP 地址为 2001:1001::3；服务器 IPv6 地址为 2001:1001::1。客户端 A 跟 C 都支持 Shim6 协议，因此 A 与 C 之间将会建立 IP 地址定位符与标识符的 Shim6 上下文状态（context），其中，定位符对为(2002:A00:404::A00:404↔2001:1001::2)，如图 8 所示。此时通过人为断开 6 to 4 隧道，10~15s 之后，Shim6 层通过检测到链路失效，将定位符切换到 2001:1001::4，说明 Shim6 机制正常发挥作用，如图 9 所示。图 10 通过 ping 客户端 A 的地址，查看返回的信息再次验证 Shim6 机制发挥了作用。

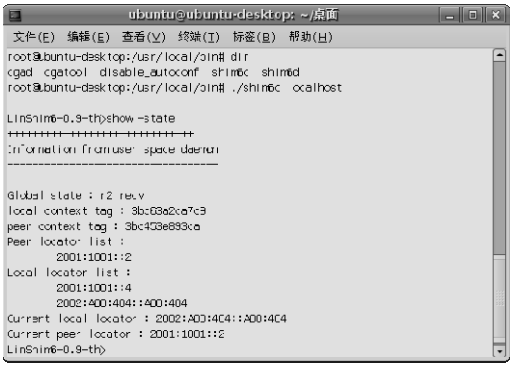


图 8　客户端 C 与 A 之间建立的 Shim6 context 情况

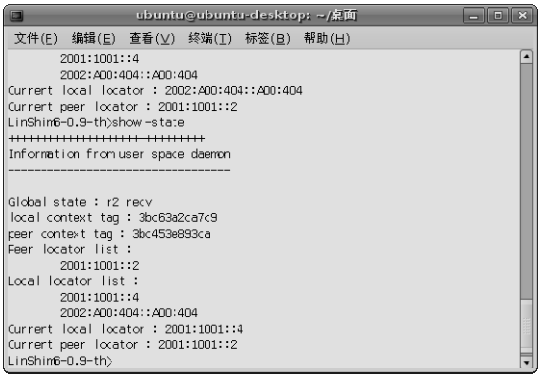


图 9　6 to 4 隧道失效后的 Shim6 context 情况

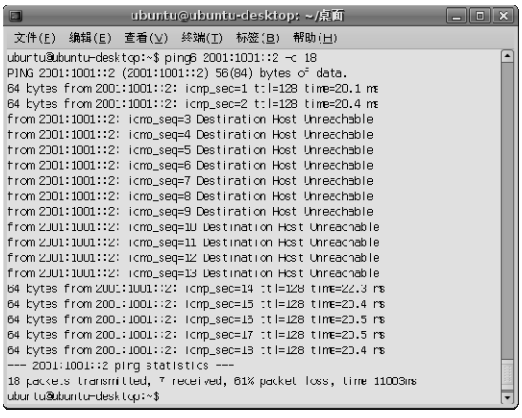


图 10　6 to 4 隧道失效后 ping 回显信息

为了测试 NAT 客户端之间的通信优化效果，客户端 B 通过 FTP 协议下载客户端 A 上不同大小

的文件，比较优化前与优化后的时间花销来测试效果，结果如图 11 所示。

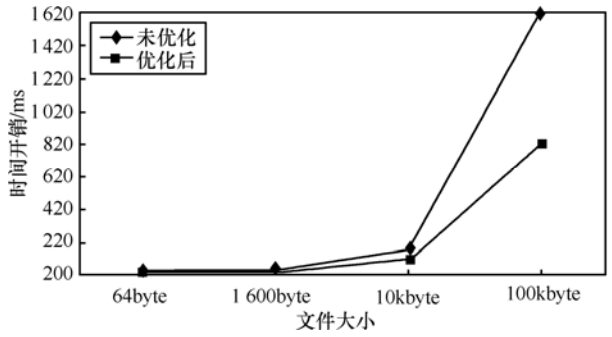


图 11 未优化与优化后时间开销对比

从图 11 可以分析得到，当下载文件很小时，未优化时间开销更小，这是由于优化过程中首先要进行客户端间的隧道参数交换，这部分时间占据了大部分开销时间；随着下载文件的增大，优化后的时间开销相比未优化情况明显减少，隧道参数交换的时间开销占总开销的比重越来越低，而对于未优化的情况，隧道数据分组还得经过服务器解封后进行二次封装，整个开销时间相对稳定，从下载 100kbyte 大小的文件来看，优化后的时间比未优化缩短了大约 50%，证明客户端间的通信优化取得了一定的效果。

5 结束语

本文首先介绍了 IPv6/IPv4 共存环境主流的过渡机制及 Shim6 多宿协议，分析了 Teredo、6 to 4 和 TSP 3 种主流隧道机制的优缺点。针对当前过渡阶段复杂的环境，设计实现基于 TSP 机制的 TrNAT 隧道方案，实现了全类型 NAT 用户访问 IPv6 网络。同时本文通过借助 MIPv6 通信最优化思想，对部分 NAT 用户进行了通信优化，实现了部分 NAT 用户间直接进行通信，减轻了 TrNAT 服务器的负担，提高了通信效率；对于非 NAT 用户则使用数据转发效率更高的 6 to 4 隧道。本文同时借助 Shim6 多宿协议实现了 NAT 用户的通信冗余，改善了非 NAT 用户使用 6 to 4 隧道路由聚集问题。

参考文献：

[1] EGEVANG K, FRANCIS P. The IP Network Address Translator (NAT)[S]. 1994.

[2] CARPENTER B, MOORE K. Connection of IPv6 Domains Via IPv4 Clouds[S]. 2001.

[3] TEMPLIN F, GLEESON T, THALER D. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)[S]. 2008.

[4] NORDMARK E, BAGNULO M. Shim6: Level 3 Multihoming Shim Protocol for IPv6[S]. 2009.

[5] IP networking lab(LinShim6|INL)[EB/OL]. http://inl.info.ucl.ac.be/LinShim6, 2007.

[6] HUITEMA C. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) [S]. 2006.

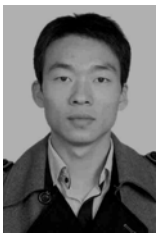
[7] BLANCHET M, PARENT F. IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)[S]. 2010.

[8] HUITEMA C. Teredo: Tunneling IPv6 over UDP through Network Address Translations(NATs)[S]. 2006.

[9] 吴贤国, 刘敏. 支持 NAT 用户的 IPv6 隧道代理设计和实现[J]. 计算机工程, 2006,32(23):97-99.
WU X G, LIU M. Design and implementation of the IPv6 tunnel broker to support NAT users[J]. Computer Engineering, 2006, 32(23):97-99.

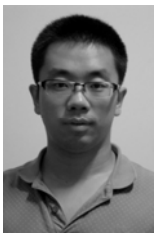
[10] DAVIES J. Understanding IPv6 (second edition)[M]. Redmond, Washington, USA: Microsoft Press, 2008.

作者简介：



陆年锋（1987-），男，浙江东阳人，解放军信息工程大学硕士生，主要研究方向为 IPv6 及网络安全。

王振兴（1959-），男，河北晋州人，解放军信息工程大学教授、博士生导师，主要研究方向为 IPv6 及网络安全。



刘慧生（1985-），男，山西忻州人，解放军信息工程大学博士生，主要研究方向为 IPv6 及网络安全。