

从 IPv4 到 IPv6 的演进技术

邵文简* 曹争

(东南大学 计算机系, 210096 南京)

【摘要】本文首先介绍了 IPv6 演进中的基本问题, 如地址结构、域名解析、路由。然后介绍了两类演进工具, 双协议栈和隧道技术以及转换工具, 并且详细说明了几种具体的实现方法。

【关键词】IPv6 DNS Tunneling Dual Stack 隧道技术 双协议栈

中图分类号: TP393

The migration technology from Ipv4 to Ipv6

Shao Wenjian Cao Zheng

(Southeast University, Computer Science Dept., 210096 Nanjing, P.R.China)

【Abstract】In the paper, some general deployment issues such as Ipv6 addressing, Ipv6 DNS and Ipv6 routing are introduced. Then two types of transition tools, dual stack & tunneling technology and protocol translation tools are discussed. Several realization methods are discussed in detail.

【Key words】Ipv6; DNS; Tunneling; Dual stack;

1. 引言

从 IPv4 到 IPv6 的演进是一个长时期的过程, 人们对它的接受情况也不相同。一些人很快接受了 IPv6, 而有些人在 IPv4 地址耗尽时材不得不接受。所以, IPv4 与 IPv6 将在很长的时期共存。IETF 建立了专门的讨论组“ngtrans”来研究演进方面的问题。

IPv6 过渡的关键问题是允许 IPv6 主机和 IPv4 主机的互操作。其次是允许 IPv6 的主机和路由器可以自由的以扩散和增长的方式被安装配置。第三是易于端用户, 系统管理员和网络操作员操作。

IPv6 的过渡机制有以下特点:

- I 增长式升级和配置。IPv4 主机和路由器可以独立地被升级到 IPv6, 新的 IPv6 的主机和路由器可以一台一台的安装, 可以首先升级路由器, 再升级主机, 也可以首先升级主机, 再升级路由器。
- I 升级时只需先将 DNS 服务器升级为支持 IPv6 地址的即可, 没有其它的要求。
- I IPv4 主机和路由器的地址仍然可以使用, 不需要重新分配地址。
- I 将已有的 IPv4 系统升级为 IPv6 系统只需作很少的准备工作。

目前, 演进问题涉及的主要方面是如何从 IPv4 网演进到 IPv6 网, 当然 IPv6 的设备与 IPv4 设备的交互能力是考虑的重点。为了实现这个目的, 现在已经有了多种机制。

* 作者简介: 邵文简, 硕士研究生, 主要研究方向为 Ipv6 技术。

曹争, 高级工程师, 主要研究方向包括网络管理、IP 互联技术。

定稿日期: 2000-06-10

个双协议栈节点，发送端节点将 IPv6 报文作为其 IPv4 报文的数据段进行封装，当然还要做相应的分段，计算校验和等工作，然后向接收端的 IPv4 地址发送。接收端在收到报文后做相应的分段等工作，通过 IPv4 的协议类型字段发现其中封装的是 IPv6 报文，将做为数据段传输的 IPv6 报文交由其 IPv6 的协议栈处理。

隧道是单项的，所以要实现通信必须在两个方向都建立隧道。

3.2 协议转换

在从 IPv4 向 IPv6 演进的早期，希望 IPv6 的“岛屿”能与 IPv4 的“海洋”互通，在演进的后期则正好相反。协议转换的思想是很自然的，如果能够实现 IPv4 协议与 IPv6 协议之间的转换，就可以很好的解决 IPv4 与 IPv6 协议互通问题。有三类转换技术，它们是：报头转换、传输中继、应用层代理。

3.2.1 报头转换

报头转换是 IP 层的转换机制，它将 IPv6 的报头转换为 IPv4 的报头，或者将 IPv4 的报头转换为 IPv6 的报头，如有必要，对校验和做适当的调整。报头转换的速度非常快，但是它有三个缺点。

1. 大量的应用层协议，如 FTP 等，在应用层中嵌入了网络层的信息，而这些信息仅通过报头转换是不能转换的。
2. 因为 IPv6 的报头比 IPv4 的报头长 20 个字节，所以一个大的 IPv4 报文将被 IPv6 重新分段。
3. ICMPv4 与 ICMPv6 不可互换。

3.2.2 传输中继

传输中继是传输层的转换机制，它中继一个 IPv4 的会话与一个 IPv6 的会话。当一个 IPv4 的 TCP 请求到达中继服务器，中继服务器就将它上传到 TCP 层。然后中继服务器与 IPv6 的目的节点建立一个 IPv6 的 TCP 连接，将前面接收到的数据通过这条连接传输到目的节点。同样，从 IPv6 到 IPv4 的中继也是如此。SOCKS 是另一种类型的传输中继，一个基于 SOCKS 的转换器要求客户端支持 SOCKS，即已经安装了 SOCKS 库。

因为是传输层的转换机制，传输中继可以很好的解决报头转换中的分段和 ICMP 不可互换的问题。但是，它不能解决应用层中嵌入了网络层的信息问题。

3.2.3 应用层代理

应用层代理是应用层的转换机制。它可以用来隐藏站点信息，并且可以通过 cache 机制提高服务性能。应用层代理如果实现了双协议栈，就可以用来提供应用层的转换功能。

因为是应用层的转换机制，所以它可以解决报头转换中的所有缺点，但是它要求服务器同时提供 IPv4 和 IPv6 服务。

4. 实际的演进工具

4.1 隧道

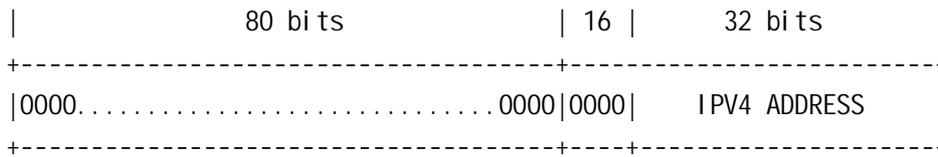
隧道是最常用的演进工具，它又分为手工配置隧道和自动隧道。

4.1.1 手工配置隧道

手工配置隧道主要用来连接两个被 IPv4 网络分隔的 IPv6 网络。它适用于网络间流量比较稳定的情况，当前 6BONE 主干就是通过手工配置隧道来连接的。它需要手工地对每个单向的隧道配置本端节点和对端节点的 IPv4 和 IPv6 地址，这样隧道才能正常工作。

4.1.2 自动隧道

自动隧道主要用来连接两个被 IPv4 网络分隔的 IPv6 主机。它适用于突发流量的情况，有流量产生时临时建立，在用完后及时拆除。自动隧道不需要手工地配置，它使用兼容 IPv4 的 IPv6 地址。兼容 IPv4 的 IPv6 地址格式如下：



由图可见，兼容 IPv4 的 IPv6 地址中包含了 IPv4 地址，所以不需要手工配置。在 IPv6 的源地址和目的地址中使用兼容 IPv4 的 IPv6 地址传输地址时，主机会自动将其封装在 IPv4 报文中传输，其中的 IPv4 信息从兼容 IPv4 的 IPv6 地址中获得。

4.1.3 隧道代理

手工配置隧道的配置和管理工作较为繁重，隧道代理可以自动地管理建立隧道的请求，并为申请方提供 IPv6 地址以及域名解析，特别适合那些想接入 IPv6 网络，但其 ISP 尚没有接入 IPv6 网络的用户。

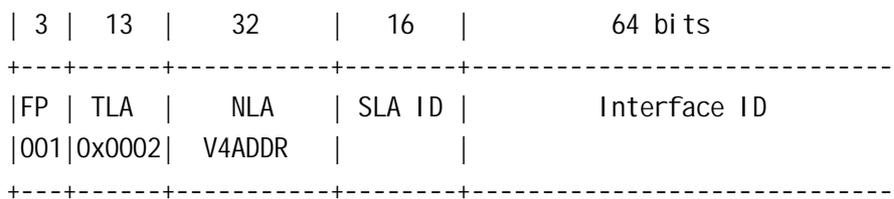
一个完整的隧道代理服务器由隧道代理、隧道服务器、域名服务器组成。隧道代理通常通过一个基于 WWW 的界面接受用户建立、修改、删除隧道的请求。用户通过界面填写必要的注册信息。隧道服务器是真正与用户建立隧道的节点。域名服务器为用户提供域名服务。

隧道代理的工作流程是：当隧道代理接到一个建立隧道的请求后，通过一定的负载均衡算法从一组隧道服务器中挑选一个，然后根据用户的信息为用户分配一段地址，并且设定隧道的生存期。动态地向域名服务器注册。自动配置隧道的服务器方，可以自动配置隧道的用户端，也可以要求用户手动地配置隧道的用户端。

4.2 6TO4

6TO4 是另一种非显式隧道的技术，但是它不是使用自动隧道的兼容 IPv4 的 IPv6 地址，而是使用一种有特殊前缀的 IPv6 地址。自动隧道是传输两个主机间的突发流量，而 6TO4 是在两个 IPv6 网的边界路由器上实现的，用来传输稳定的流量。

6TO4 使用的是一种特殊的可聚类的全球唯一地址，IANA 为 6TO4 分配了一个永久的顶级聚类标识 0x0002，它的次层聚类标识的值为 IPv4 地址。地址格式如下：



因为 6TO4 使用的 IPv6 地址中有 IPv4 的地址信息，所以实现了 6TO4 的边界路由器就可以将 IPv6 报文封装在 IPv4 的报文中传输，而将 IPv4 网络视为一条虚拟链路。

4.3 6OVER4

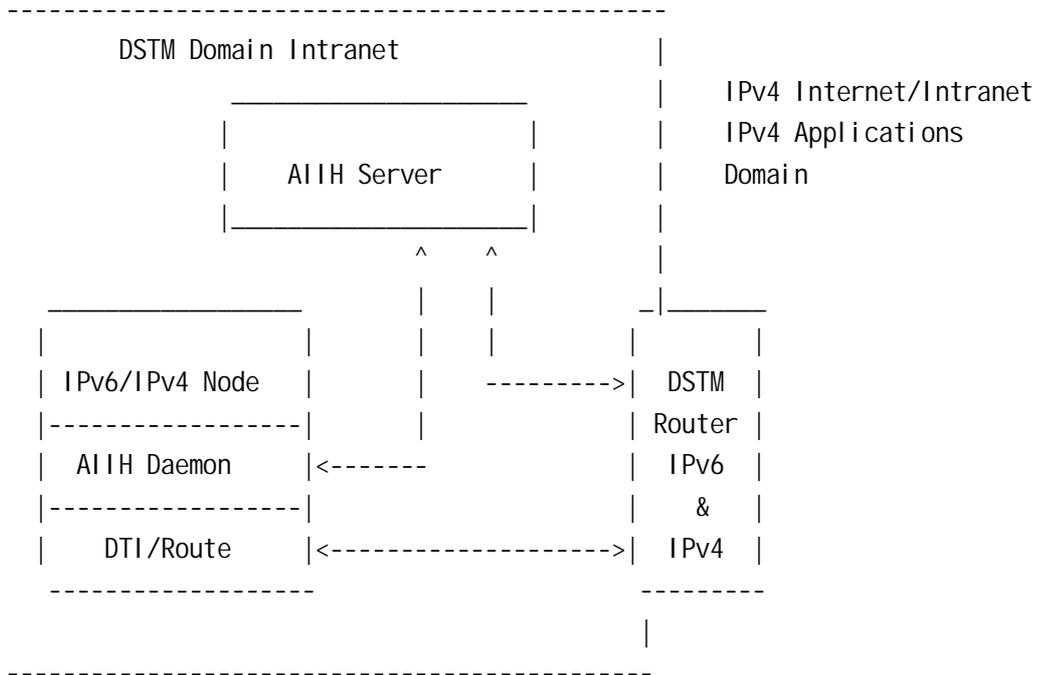
6OVER4 也是一种非显式隧道的技术，它的设计目的是允许孤立的 IPv6 主机将一个 IPv4 的组播(Mul ti cast)域做为它们的虚拟链路与这个组播域中的 IPv6 路由器连接，实现完全的 IPv6 功能。因此在这个组播域中一定要有一个实现 6OVER4 的路由器。使用这种方法连接的主机不需要兼容 IPv4 的 IPv6 地址或手工配置的隧道，但是要求 IPv4 网络必须支持组播功能。

6OVER4 是将 IPv6 报文封装在 IPv4 的组播报文中在组播，这样组播域中的所以主机都可以收到，就好象所有主机都在一条物理链路上一样。每个主机在收到组播的报文后将 IPv4 的报头去除，再根据 IPv6 报文的信息来判断是否是发给自己的报文。

4.4 DSTM

DSTM 是双协议栈转换机制的缩写，它是两种技术的组合，为 IPv6 主机分配 IPv4 地址(AI IH)和动态隧道接口(DTI)。它用于双协议栈主机(但是没有 IPv4 地址)与 IPv4 主机之间

的交互。但是它只能在 IPv6 的 Intranet 中使用，而不能用于 Internet。一个实现了 DSTM 的 IPv6 Intranet 称为 DSTM 域。DSTM 的结构如下图所示：



AIH 服务器包括域名服务器、DHCPv6 服务器以及它们之间的通信。AIH 服务器接受分配地址的请求，通过 DHCPv6 动态的分配给请求者一个 IPv4 地址，并且在域名服务器中对请求者的信息做相应的修改。

因为 DSTM 域是一个 IPv6 的 Intranet，这就需要有一种机制将 IPv4 的报文封装在 IPv6 的报头中。DTI 处于在 IPv4 API 与 IPv6 之间，它将 IPv4 报文封装的 IPv6 报文中在 DSTM 域中传输。

当一个双协议栈的主机 A 需要与 IPv4 主机 B 交互时，它向域名服务器请求 B 的 IPv6 地址，在收到错误的信息后，它请求 B 的 IPv4 地址。这时 A 向 AIH 服务器请求分配一个暂时的 IPv4 地址，AIH 服务器通过 DHCPv6 分配给 A 一个 IPv4 地址，同时动态地修改了 A 的域名解析信息。DTI 将 IPv4 报文封装在 IPv6 中在 DSTM 域中传输，DSTM 域的边界路由器将报文解封后传输给 B。B 到 A 的过程恰好相反。

4.5 SIIT

SIIT 是动态 IP/ICMP 转换算法的缩写，它是一种网络层的报头转换机制，在 IPv4 与 IPv6 的报头之间进行转换。但是为了能够在 ICMPv4 与 ICMPv6 之间进行转换，对 ICMPv4 与 ICMPv6 的能够转换的错误信息进行了转换，对于 IPv4 中没有的错误信息，则只能做舍弃处理。转换是在动态进行的，对每一个报文都要进行转换。

处理 SIIT 算法的服务器处于 IPv4 和 IPv6 网络之间，对 IPv4 和 IPv6 网络之间的报文进行转换。要使 IPv6 主机与 IPv4 主机之间顺利的交互，该服务器还需要有一个为 IPv6 主机分配暂时的 IPv4 地址，并动态修改域名服务器中配置的功能模块。但是 SIIT 算法本身并没有定义这个模块，可以通过 DHCPv6 或其他方法实现。当然，这个 IPv4 的地址不是真实的分配给 IPv6 主机，而是在该服务器中有一个 IPv6 与 IPv4 的地址转换表。当 IPv6 的报文到达该服务器时，它用相应的 IPv4 地址来转换报头，而当 IPv4 报文到达该服务器时，它用相应的 IPv6 地址来转换报头。

4.6 NAT-PT

NAT-PT 是网络地址转换-协议转换的缩写，它也是一种 IPv4 与 IPv6 之间通信的工具。NAT-PT 是由两部分：网络地址转换和协议转换构成。NAT-PT 的协议转换就是使用了 SIIT 的算法，而网络地址转换与 IPv4 的 NAT 是很相似的，不过前者是 IPv6 与 IPv4 地址之间的转换，后者是 IPv4 的私有地址与正式地址之间的转换。

NAT 会记录每一条会话的信息，报文是一个会话的初始化报文时，NAT 就会为它分配一个暂时的 IPv4 地址，并进行记录。而如果报文不是一个会话的初始化报文时，NAT 就会在它的记录中找到这条会话对应的信息，并使用信息中的地址进行转换。

NAPT-PT 是网络地址、端口转换-协议转换的缩写，NAPT-PT 与 NAT-PT 的不同在于，它可以只使用一个 IPv4 地址，而不需要为每个 IPv6 地址分配一个暂时的 IPv4 地址。它采用的方法是将 IPv6 的地址和传输层的端口号转换成 IPv4 地址和传输层的端口号。这样，每一个 IPv4 地址最多可以转换 6 万多个 TCP 与 UDP 会话。

4.7 BITSV6

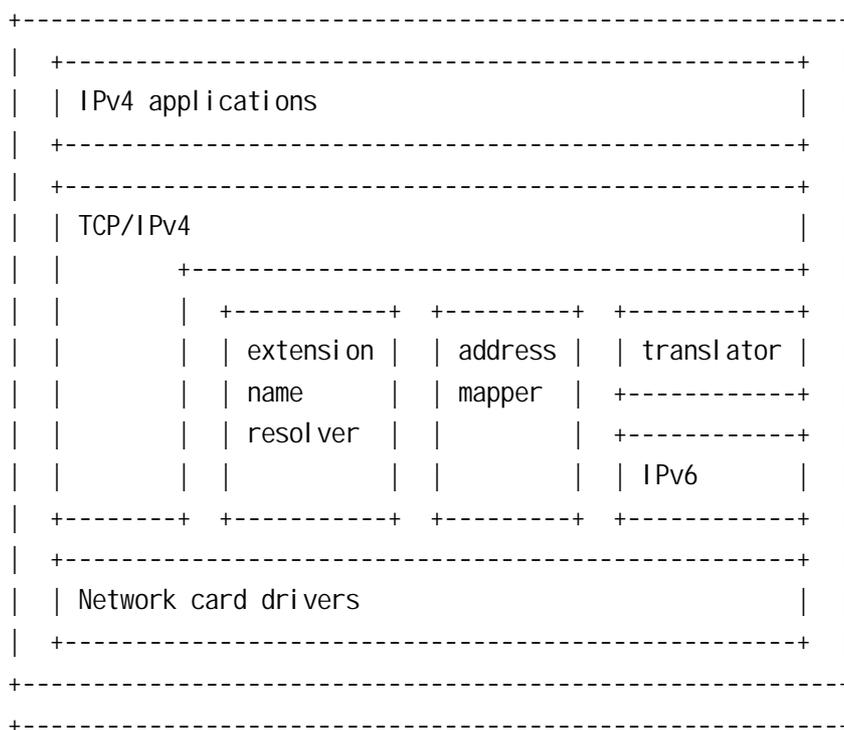
Bump-In-The-Stack 是一个在 IPsec 中使用的技术。BITSV6 使用类似的概念，使得双协议栈主机可以使用 IPv4 的应用程序与 IPv6 主机通信。

BITv6 中有三个重要的模块，协议转换器，扩展名称解释器和地址映射器。它们位于 TCP/IPv4 模块和网卡驱动模块间，监听 IPv4 数据并进行处理。

协议转换器使用 SIIT 算法在 IPv4 和 IPv6 报头间进行转换。扩展名称解释器响应 IPv4 应用程序的请求，返回一个恰当的 IPv4 地址。当 IPv4 应用程序向域名服务器请求解析一个 IPv4 地址，扩展名称解释器监听到这个请求后，向域名服务器请求解析 IPv4 和 IPv6 地址。如果得到一个 IPv4 地址，说明目的主机是一个 IPv4 主机，不需要协议转换。如果得到一个 IPv6 地址，说明目的主机是一个 IPv6 主机，需要地址映射器为它分配一个 IPv4 地址，并将 IPv4 地址返回给应用程序。地址映射器接收扩展名称解释器或协议转换器的请求，为 IPv6 主机分配 IPv4 地址，并记录地址间的映射关系。

协议转换是在主机内进行的，地址映射器分配的 IPv4 地址不会在主机外出现，而且对应用程序而言，目的主机始终是 IPv4 的主机。

BITSV6 的结构如下图：



当双协议栈主机用 IPv4 应用程序与 IPv6 主机通信时，经过上面的描述的过程，应用程序向目的地址发送 IPv4 报文。当 IPv4 报文到达协议转换器，转换器从地址映射器中查得 IPv4 地址对应的 IPv6 地址，在进行转换后发送到 IPv6 主机。当 IPv6 报文到达协议转换器，协议转换器将 IPv6 报文转换为 IPv4 报文。

4.8 SOCKS64

SOCKS64 网关是一种传输中继，是传输层的转换工具。它接收 IPv4 主机的 SOCKS 连接，将其中继到 IPv4 或 IPv6 主机。对那些已经使用 SOCKS 技术的站点，SOCKS64 网关是最快捷的演进方法。同样使用 SOCKS64 网关可以实现 IPv6 主机访问 IPv4 主机，也可以用于 IPv6 网络中的 IPv4 主机间通信或 IPv4 网络中的 IPv6 主机间通信。

5. 结束语

IPv6 对 IPv4 做了重大的改进，解决了 IPv4 中的许多问题。同时，它又提供的许多新的特性，支持其它网络协议(如 IPX 和 NSAP)，满足了新的市场的要求(如个人的移动计算设备，网络娱乐，设备扩展等等)，是适应下一世纪的网络发展的网络协议。因此，IPv6 技术为今后网络技术的研究和发展开辟了广阔的空间。尽早得从 IPv4 演进到 IPv6 就可以在今后的发展中取得主动，这也是中国网络技术赶超世界水平的一个机遇。

当然，IPv6 还是一个新生的事物，本文中涉及的算法和方法很多还处于发展阶段，远没有达到成熟的地步，可能存在着不足之处，恳请专家不吝指正。

【参考文献】

- [1] S.Deering, R.Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) Specification, December 1998.
- [2] R. Hinden, S. Deering. RFC 2373: IP Version 6 Addressing Architecture, July 1998.
- [3] R. Hinden, M. O'Dell, S. Deering. RFC 2374: An IPv6 Aggregatable Global Unicast Address Format, July 1998.
- [4] W. Biemolt, T. Larder, H. Steenman. Internet Draft: A Guide to the Introduction of IPv6 in the IPv4 World, October 1999.
- [5] George Tsirtsis, Pyda Srishuresh, "Network Address Translation - Protocol Translation (NAT-PT)", draft-ietf-ngtrans-natpt-06.txt (work in progress).
- [6] Erik Nordmark, "Stateless IP/ICMP Translator", draft-ietf-ngtrans-siit-06.txt (work in progress).
- [7] B. Carpenter, K Moore, "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels", draft-ietf-ngtrans-6to4-03.txt (work in progress).
- [8] K. Tsuchiya, H. Higuchi, Y. Atarashi, "Dual Stack Hosts using the Bump-in-the-Stack technique", draft-ietf-ngtrans-dual-stack-hosts-02.txt (work in progress).
- [9] A. Durand, P. Fasano, I. Guardini, D. Lento, "IPv6 Tunnel Broker", draft-ietf-ngtrans-broker-00.txt (work in progress).
- [10] J. Bound, L. Toutain, H. Affifi, "Dual Stack Transition Mechanism (DSTM)", draft-ietf-ngtrans-dstm-00.txt (work in progress).