第 33 卷 第 2 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.2 Computer Engineering January 2007
·网络与通信· 文章编号：1000—3428(2007)02—0099—03 文献标识码：A 中图分类号：TP393

# IPv6 网络拓扑发现技术研究

杨国正，陆余良，夏 阳
(合肥电子工程学院网络工程系，合肥 230037)

**摘 要：**分析了 IPv6 网络的自身特性，提出了在 IPv6 网络环境中进行拓扑发现与 IPv4 网络相比存在的问题。针对这些问题，阐述了相应的解决方案，在此基础上设计了 IPv6 网络拓扑自动发现系统的整体结构，对系统主要探测模块的功能和算法的实现思想进行了分析。
**关键词：**IPv6；网络拓扑；隧道发现

# Research on Topology Discovery for IPv6 Networks

YANG Guozheng, LU Yuliang, XIA Yang
(Department of Network Engineering, Hefei Institute of Electronic Engineering, Hefei 230037)

**【Abstract】** This paper analyzes the features of IPv6 networks, and presents several problems which are different from IPv4 networks. It also presents solutions to these problems, and designs a whole system framework for IPv6 network topology discovery. The module functions and algorithmic technology are described later.
**【Key words】** IPv6; Network topology; Tunnel discovery

随着网络规模的迅速发展，IPv4 协议逐渐显示出它的弊端：IP地址空间匮乏、路由表的急剧扩张和服务质量缺乏等问题已逐渐不能满足网络发展的需要。针对这些问题，IETF早在 20 世纪 90 年代初就提出开发新的IP协议，即IPv6 协议。RFC3513[1]是IETF最近公布的IPv6 地址结构体系，它通过采用 128 位的地址空间替代IPv4 的 32 位地址空间来扩充因特网的地址容量，使得IP地址空间不再成为限制网络规模的因素。但由于IPv6 中IP地址长度以及包的格式等都发生了变化，原IPv4 网络的相关协议也必须升级到相应的v6 版本，或者被新的协议所取代。因此，IPv4 网络拓扑发现算法在IPv6 中并不能适用。鉴于IPv6 在下一代网络中的重要地位，尤其在目前由IPv4 向IPv6 网络的过渡时期，对IPv6 网络拓扑及其发展的研究具有非常深远的意义。

## 1 IPv6 拓扑发现的问题

### 1.1 子网节点探测问题

根据RFC3587[2]给出的IPv6 全球单播地址新格式(如表 1)，接口ID占了 64 位，因此IPv6 的子网中节点个数可以达到 $2^{64}$个，对于如此庞大的数量，如果还是采用ICMP回声请求的方法进行子网内主机存活率探测，结果的时效性将不能满足拓扑发现的需要。因此，在IPv6 下子网探测不能采用群ping方法。

**表 1 IPv6 全球单播地址新格式**

| n bits | 64-n bits | 64 bits |
|---|---|---|
| 全球路由前缀 | 子网ID | 接口ID |

### 1.2 路由器寻址问题

在 IPv6 中，地址是独立接口的标识符，所有的 IPv6 地址都被分配到网络接口，而非节点。而对于任何一个网络接口，可同时拥有聚集全球单播地址(Aggregatable Global Unicast Addresses)、站点本地地址(Site-Local address)和链路本地地址(Link-Local address)3 类单播地址。站点本地地址和链路本地地址的地址格式如表 2。

**表 2 站点本地地址和链路本地地址的地址格式**

| 地址类型 | 10 bit | 54 bit | 64 bits |
|---|---|---|---|
| 站点本地地址 | 1111111011 | 子网ID | 接口ID |
| 链路本地地址 | 1111111010 | 0 | 接口ID |

网络拓扑发现的过程中，得到路由器地址后，要找到其它子网信息，就需要依赖于识别网络前缀。在 IPv4 中经常采用 MIB 信息中的 ipRouteNextHop 项来发现和路由器相连的子网信息，而在 IPv6 路由表中，其 ipv6RouteNextHop 项经常是以本地链路地址来表示，即其子网前缀为 FE80∶∶/64，这在任何情况下都是不可能被寻址的，因而无法从该子网前缀得到下一跳的信息。

### 1.3 路由器的匿名问题

由于 IPv6 的报文格式发生了变化，路由器的转发报文机制有了一定的改变。在正常情况下，发送 traceroute 探测包进行拓扑探测的过程中，在探测包的 TTL 值耗尽的时候，会以当前所在路由器的一个接口上的全球单播地址为源地址，返回一个 ICMP 响应包。然而，有时并非路由器上所有的接口都配置有全球单播地址，有的只配置了站点本地地址和链路本地地址。当选择了一个只配置有站点本地地址和链路本地地址的接口为源地址发送 ICMP 响应包时，大多数路由器会拷贝初始探测包的目的地址为 ICMP 响应包的源地址，返回给探测点，从而将本身的路由器信息隐藏掉，这种路由器就叫做匿名路由器。例如，对于 X-Y-Z 的路径，如果 Y 为匿名路由器，则探测结果将会是 X-Z-Z。

### 1.4 IPv6 的隧道发现问题

由于IPv6 的部署是一个长期的过程，IPv4 和IPv6 的混合

**作者简介：**杨国正(1982－)，男，硕士生，主研方向：网络安全；陆余良，博导；夏 阳，博士生
**收稿日期：**2006-01-22 **E-mail：**yangguoz@hotmail.com

网络必然会存在相当长的一段时间。隧道机制是解决在IPv4网络中传送IPv6数据包的重要解决途径。目前，主要使用的隧道方式有 6over4 隧道[3]和 6to4 隧道[4]。在IPv6 网络连入IPv4 骨干网的边界上，一般都配有双栈协议的隧道接口路由器。传输过程中，IPv6 数据包用隧道源IPv4 地址的报文进行二次封装，经过IPv4 骨干网后，由另一端双栈路由器去掉IPv4的报头后，继续向目标IPv6 站点转发。因此，如何判断IPv6隧道的存在，以及如何确定隧道的两边端点是对混合网络拓扑发现的一个重要问题。

## 2 IPv6 拓扑发现算法

### 2.1 研究现状

目前，国内外对IPv6 网络的拓扑发现研究都比较少。比较典型的IPv6 网络拓扑发现框架是由法国ORIA-INRIA实验室I. Astic等提出的分层拓扑发现结构[5]，如图 1。

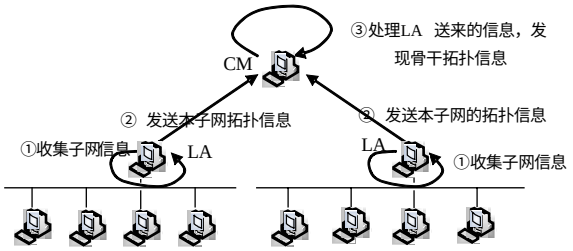


**图 1 分层拓扑发现结构**

它通过子网代理(LA)发送一组多播地址来发现子网内的所有节点的地址信息和主机信息；通过管理中心(CM)来集中处理由多个本地代理送来的子网信息，并用 traceroute6 来实现对骨干网络的结构发现，然后分析出全网的拓扑结构。

对于IPv6 隧道的发现方面，Lorenzo Colitti等人[6]采用发现最大传输单元(MTU)和注入IP欺骗包的方法提出了基于IP欺骗包的 8 条发现规则来判断隧道的存在和确定两端的节点信息。

以上方法提出了 IPv6 拓扑发现探测中一些存在问题的解决方案，但考虑的方面不同：I. Astic 等人的分层拓扑发现是在纯 IPv6 网络下进行的，因而不适用于目前的混合网络；Lorenzo Colitti 等人是从发现 IPv6 隧道的角度出发，提出了对隧道发现算法的设计，但对拓扑发现的其它问题没有涉及。本文将从当前 IPv6 网络发展的结构出发，在上述研究的基础上对 IPv6 网络拓扑发现的整体架构和实现算法进行阐述。

### 2.2 IPv6 拓扑发现框架

目前，IPv6 网络尚处于起步发展阶段，在 Internet 上其网络结构如图 2 所示。

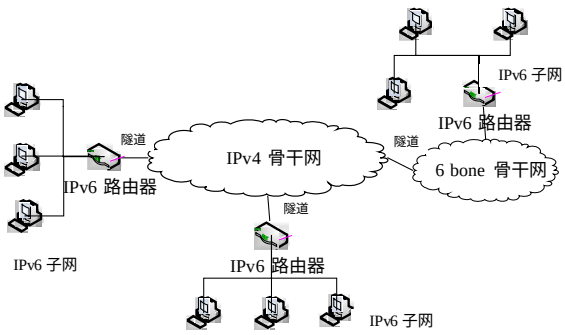


**图 2 IPv6 网络示意图**

IPv6 子网孤立地存在于 IPv4 骨干网络之间，通过隧道机制完成与 6bone 骨干网(1996 年由 IETF 创建的与现有 Internet 相连的 IPv6 示范网)的连接。因此，对 IPv6 拓扑发现应该分为 3 个部分：子网发现模块，隧道发现模块和骨干网发现模块。子网发现模块中包括节点发现子模块、地址处理子模块、本地存储模块和信息发送子模块；隧道发现模块包括隧道判定子模块和双栈识别子模块；骨干网发现模块包括路径探测子模块模块和匿名接口处理子模块。此外，还应包括拓扑存取模块、拓扑数据融合模块、拓扑显示模块。其总体框架如图 3。

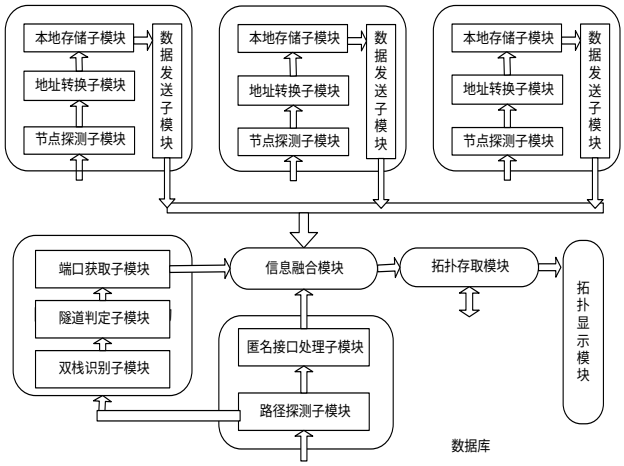


**图 3 IPv6 网络拓扑发现系统结构**

### 2.3 子网发现模块

子网发现模块的主要功能是要发现本地链路内存在的路由器和主机信息，进行本地地址和可路由地址之间的相互转换工作，以相应的格式存储起来并转发到相应的控制节点。

在 RFC3513 中预定义了一组多播地址可以解决子网节点的探测问题。IPv6 中预定义的组播地址如表 3 所示。

**表 3 预定义多播地址类型**

| 多播地址 | 地址类型 |
|---|---|
| FF01:0:0:0:0:0:0:1 | 所有本地接口地址 |
| FF02:0:0:0:0:0:0:1 | 所有本地链路地址 |
| FF01:0:0:0:0:0:0:2 | 本地接口所有路由器地址 |
| FF02:0:0:0:0:0:0:2 | 本地链路所有路由器地址 |
| FF05:0:0:0:0:0:0:2 | 本地站点所有路由器地址 |

对于目标地址为 FF02::1 的包，本地链路内所有节点(包括路由器、主机)都会接收并返回响应包；对于目标地址为 FF02::2 的包，本地链路内所有路由器都会接收并处理。因此，利用这两个多播地址便可以发现本地链路内存在的节点，并对主机和路由器进行区分。

具体的算法步骤描述如下：

(1)通过 ping6 FF02::1 发送回显应答请求包；

(2)根据返回应答包的源地址得到本地链路内的所有节点，存入节点表中；

(3)通过 ping6 FF02::2 发送回显应答请求包；

(4)根据返回应答包的源地址发现本地链路内的所有路由器，存入路由器表中；

(5)从节点表中取出不属于路由器表中的项，加入到主机表中。

由于通过上述方式获得的地址一般是本地链路地址，而 IPv6 中一个节点往往还具有一个和其它网络通信的 IPv6 全局地址，因此必须完成两种地址之间的转换功能。

全局地址的配置方式有 3 种：无状态地址自动配置，有

状态地址自动配置和手工配置。无状态地址自动配置是通过路由器定期发出包含子网前缀的宣告，主机收到路由器宣告以后根据子网前缀和本身的接口标识完成全局地址的配置过程。这里可以根据无状态自动配置的工作原理来发现节点的全局地址。当获得节点的本地链路地址时，模拟路由器请求报文，并发送至本地链路内所有路由器的多播地址，路由器就会发出路由器宣告。接收路由器宣告报文并解析就可以获得该本地链路的子网前缀，然后根据本地链路地址获得节点的全局地址，见图 4。

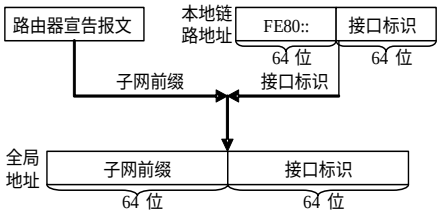


**图 4 地址类型转换**

目前使用最为广泛的是无状态地址自动配置方式，有状态地址自动配置的工作原理和 IPv4 网络中的 DHCP 服务器基本一样，手动配置的方式比较复杂，仍属于有待研究的领域。

### 2.4 隧道发现模块

隧道发现模块建立在骨干网络发现模块的基础上，对于骨干网络的每一条路径，首先必须判断路径中的每一跳节点是否为双栈节点，如果存在双栈节点，则判断是否有隧道的存在，从而进一步获得隧道两端的端口信息。

对于骨干网络中双栈节点的识别，RFC2553[7]定义了一系列的函数来实现这个功能。例如：getaddrinfo()函数完成节点名称到地址的转换；getnameinfo()完成节点地址到名称的转换。

当判断某一节点是否为双栈节点时，首先通过已知路由器节点地址信息调用 getnameinfo()函数获得该节点的名称，再针对该节点名称调用 getaddrinfo()函数，查看该节点上的所有 IP 地址，判断返回结果中是否包含 IPv4 地址。getaddrinfo()函数返回一个 addrinfo 结构体，通过对 addrinfo 结构体中的 ai_family 值来确定其地址类型，如果 ai_family 的值为 PF_INET，则该地址为 IPv4 地址。由此可以判断出该路由器节点是双栈路由器节点还是一般的路由器节点。

对于一条路径，当两端节点都为双栈路由器时，对隧道存在的判断才有必要。隧道是否存在的判断可以通过注入大的 IPv6 包，致使在传送过程中经过隧道时产生分片，来获取路径中的 MTU 值来完成。如果隧道存在，在通过隧道时必须经过 IPv4 包头的封装作为 IPv4 包来传输，因此路径上的最大 MTU 值应是 IPv4 的 MTU 值减去 IPv4 包头的址。目前使用的 IPv4 报文 MTU 值一般为 1 500B，一般的 IPv6 隧道附加 20B 的 IPv4 报头，GRE 隧道附加 24B 或 28B 的报头，再加上 IPv6 中 MTU 的最小值 1 280，当获得的路径 MTU 值属于{1 480，1 476，1 472，1 280}中的某一个值时，即可判定隧道的存在。

### 2.5 骨干网发现模块

骨干网发现模块主要是发现骨干网中的路由器以及这些路由器之间的连接关系，然后对网络路径中存在匿名路由器的情况进行拓扑信息合并，从而形成最接近实际的骨干网络结构。

虽然 RFC2465 和 RFC2466 等文档已经定义了 IPv6 网络中扩展的 MIB 标准，但由于目前的主流操作系统并不能完全支持这些标准，因此，对 IPv6 骨干网络的探测主要还是采用 ICMPv6 通用协议来完成。在 IPv6 报文中，由于要求使用源路由机制，因此在用 traceroute 程序进行拓扑探测的时候大大增加了路径发现的概率。

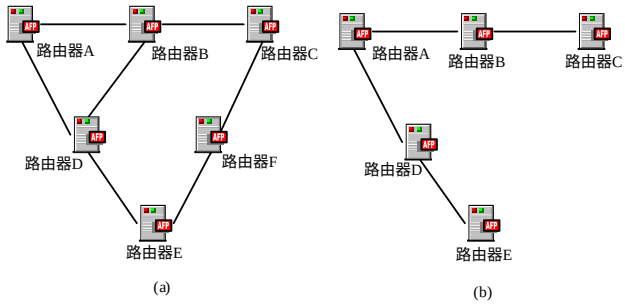


**图 5 路由路径**

例如对于网络路径如图 5(a)所示的情况，以路由器 A 为起始探测点，若用普通的 traceroute 程序进行探测的时候网络只能发现 A-B-C 和 A-D-E 的路径，如图 5(b)所示；而采用源路由机制时，在一般发现的基础上，不仅可以发现 B-D 的链路，而且可以发现额外路由器 F 以及链路 C-F 和 E-F 的情况。

因此对骨干网的发现，采用基于源路由机制的 traceroute6 程序，首先根据获得的在 6bone 上注册的 IPv6 的站点地址列表，并将其作为 traceroute6 的目标地址来获得骨干网络的路径。再根据 RFC2553 中定义的函数 getaddrinfo()和 getnameinfo()来识别同一路由器的多接口问题，合并成一个初始的骨干拓扑网络。

由于匿名节点的存在，上述探测的骨干网络将与实际情况相差很大。当一个匿名节点在多条路径上同时出现时，它将会被区分成不同的节点对待，此时将得出一个比实际情况更加复杂的拓扑图，如图 6 所示，其中图 6(a)是实际拓扑图形，图 6(b)是探测所得的图形。

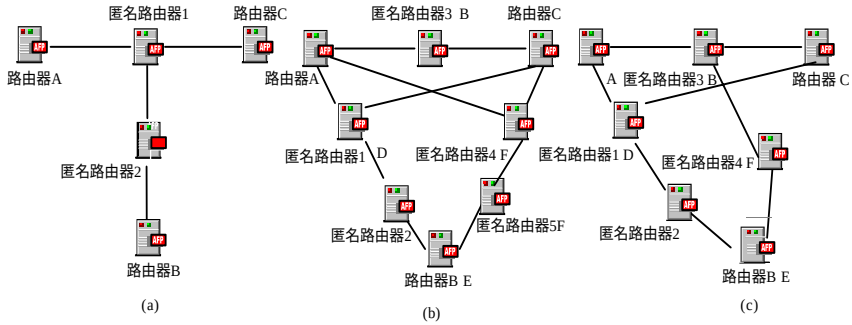


**图 6 匿名路由器**

因此，要得到符合实际情况的拓扑图，必须对匿名路由器的存在进行处理。由图可知，仅根据路径的信息，要想由图 6(b)推导到图 6(a)的情况是非常困难的，因为匿名路由器的合并情况有很多种，不同的合并方式可以得到不同的拓扑图形，如图 6(c)。这里采用最可能简化匿名路由器的方式进行合并，即最大程度的减少匿名路由器的数量和路径条数。合并准则有两点：(1)合并后各路由器节点之间的跳数不变(匿名节点除外)；(2)无环路出现的路径中的匿名路由器之间不进

(下转第 104 页)