

基于 MPLS VPN 构建 IPv6 电子政务网

沈俊鑫

(云南财经大学 信息学院, 云南 昆明 650208)

摘要: 基于云南省电子政务 MPLS VPN 网, 分析了 IPv6 VPN 与 IPv4 VPN 在 IPv4 MPLS VPN Core 的部署方式, 采用 6PE+ 6VPE 方式融合 MPLS VPN 与 IPv6 着重阐述了 MPLS VPN 与 IPv6 的融合机制、实现原理及过程。

关键词: 多协议标签交换; Socket API IPv6

中图分类号: TP393.02 **文献标识码:** A **文章编号:** 1007-855X(2007)03-0040-04

Construction of IPv6 Electronic Government Affairs Network Based on MPLS VPN

SHEN Jun-xin

(Information School, Yunnan University of Finance and Economics, Kunming 650208, China)

Abstract Based on Yunnan Provincial E-Gov network MPLS VPN the deployment of IPv6 VPN and IPv4 VPN in IPv4 MPLS VPN Core is analyzed combining MPLS VPN with IPv6 in the way of 6PE+ 6VPE. An emphasis is put on the combination mechanism, realization principle and process of MPLS VPN and IPv6.

Key words MPLS Socket API IPv6

1 基本方案比较

目前, 实现 IPv4 和 IPv6 互通, 主要使用双栈 Dual Stack 隧道 Tunnel 和网络地址转换 NAT 3 种技术。其中, IPv6 互通主要采用隧道技术, IPv4 与 IPv6 互通主要采用双栈或代理机制。而隧道技术 (如 GRE 隧道、6to4 隧道、ISATAP 隧道、Teredo) 要么不易维护、实现复杂, 要么使用特殊地址; 双栈技术需要升级路由器, 同时需要部分 IPv4 地址空间; 翻译技术 (NAT-PT, TRT, BIS) 对路由器性能要求高, 有些应用无法“翻译”; 而代理技术 (DSIM, SOCKS64) 通常需要对客户端进行升级。对于 L2VPN 方式, 如 kempella 和 martinj 由于这两种方式还处于 draft 阶段, 各个厂商的设备在支持 L2VPN 方面无法实现兼容。

2 系统设计与实现

通过比较分析各种 IPv4 MPLS VPN Core 承载 IPv6 的方法, 结合现有网络架构和应用需求, 以 6PE+ 6VPE 方式对现有网络进行升级。

2.1 现有网络构架及应用需求

云南省电子政务网络覆盖省、地、县三级党政机构的“王”字型, 基于专有的 ATM 交换机, 提供 FR/PVC, Ethernet-LAN 以及 VPDN 3 种接入方式。为解决 IP 地址重叠、重叠交叉应用、骨干路由复杂以及提高服务质量 QoS 等问题, 部署了 IPv4 MPLS VPN。随着中央各部委网络的不断升级, 引入 IPv6 机制, 因此在现有 IPv4 MPLS VPN 不仅要能够实现 IPv6 Sites 间、IPv6 VPN Sites 间互联互通, 同时也要提供 IPv6 VPN 与 IPv4 VPN 之间的互联互通。

2.2 系统设计

2.2.1 构建 6PE

6PE 的构建如图 1 所示, 原系统 P 路由器不需要做任何改动, 需要接入 IPv6 域的 PE 路由器升级为 Dual Stack 6PE, 并在相应接口上启用 Dual Stack 绑定 6PE 标签到 IPv6 前缀。6PE 路由器由原来 PE 路由器升级而成, 需要支持 Dual Stack 协议, 既支持 IPv4 也支持 IPv6 它必须是核心 IPv4 网络的成员, 而且接

收稿日期: 2007-01-10

作者简介: 沈俊鑫 (1978-), 男, 硕士, 高级工程师。主要研究方向: 电子政务、网络工程, 等。

入核心网的结构必须运行 MPLS 保留原来的标签交换协议 LDP 和路由交换协议例如 IPv4 OSPF.

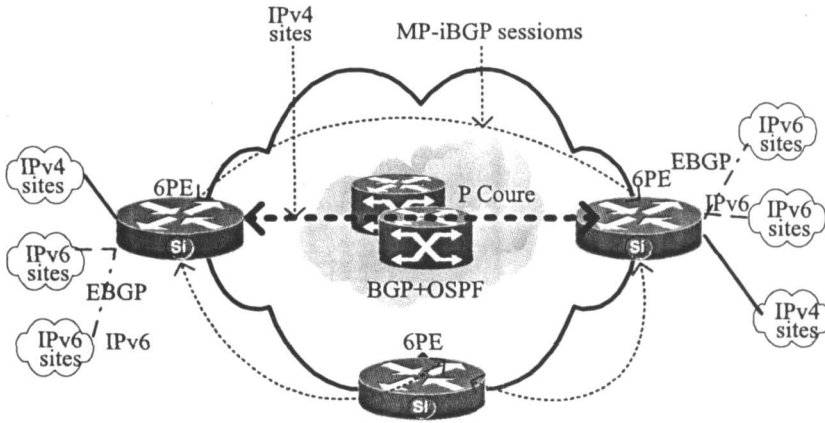


图1 6PE拓扑图
Fig.1 6PE topology

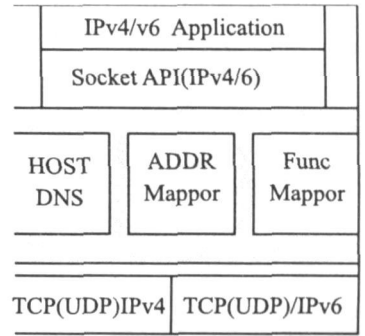


图2 6PE功能逻辑图
Fig.2 6PE logical function

对于不需要 IPv6 VPN 的用户,可以直接采用 6PE 方式. 过程如下:原 IPv4 MPLS VPN Core 利用 MPLS 的标签建立的 LSP 实现数据的传递和转发, 6PE 路由器通过 BGP 或 EBGP 学习 IPv6 路由, 然后封装为 IPv4 MBGP 路由, 利用 LSP 建立的 MPLS 隧道转发到对端 6PE. 6PE 路由器需要在原来支持 PE 的路由器上增加模块以实现 IPv4 和 IPv6 地址的解析: 如图 2 所示.

Socket API (IPv4 IPv6): 向应用层提供 IPv4 和 IPv6 两种套接字服务, 屏蔽 IPv4 和 IPv6 在 TCP/UDP 等方面具体的实现细节;

Address M apper; F unction M apper 向 Socket API 提供具体的地址映射和功能映射;

TCP(UDP) IPv4/IPv6 提供 IPv4 IPv6 两个版本的 TCP(UDP).

具体需要实现的函数(功能)

1) 核心 Socket 这些核心 Socket 需要完成包括:

a 建立、释放 TCP 连接, 接收、发送 UDP 包, 核心函数有: socket(), bind(), connect(), accept(), sendmsg(), sendto(), recvfrom(), 及 recvmsg(); 函数原型如下:

```
# include < sys/socket.h >
```

```
int socket( int family, int type, int protocol); //返回: 非负描述符—成功, -1—失败
int connect( int sockfd, const struct sockaddr* servaddr, socklen_t addrlen); //返回: 0—成功, -1—失败
int bind( int sockfd, const struct sockaddr* myaddr, socklen_t addrlen); //返回: 0—成功, -1—失败
int accept( int sockfd, struct sockaddr* cliaddr, socklen_t* addrlen); //返回: 0—成功, -1—失败
```

b 设计传输独立性: 主要实现特权端口、不透明地址和广播功能的独立性;

c 对于 IPv6 通过指针方式传递协议地址等功能:

```
s = socket(AF_INET6, SOCK_STREAM, 0)[11]; //定义 IPv6 字节流套接口
s = socket(AF_INET6, SOCK_DGRAM, 0)[11]; //定义 IPv6 数据报套接口
```

2) IPv4 地址数据结构的变化, 对于 Structure sockaddr_in 结构无法承载 16 位的 IPv6 地址信息, 在 IPv4 Address Structure 增加设计数据结构, 存储 IPv6 结构信息, 结构如下:

```
struct sockaddr_in { /* 地址结构 for IPv4 */
sa_family_t sin_family; /* 套接口地址结构的地址簇, 这里为 AF_INET */
in_port_t sin_port; /* TCP 或 UDP 端口 */
struct in_addr sin_addr; /* IPv4 地址 */
}
struct sockaddr_in6 { /* 地址结构 for IPv6 */
sa_family_t sin6_family; /* 套接口地址结构的地址簇, 这里为 AF_INET6 */
in_port_t sin6_port; /* TCP 或 UDP 端口 */
uint32_t sin6_flowinfo; /* priority & flow label */
struct in6_addr sin6_addr; /* IPv6 address */
}
```

```
uint32_t sin6_scope_it; /* set of interfaces for a scope */
}
```

3) DNS 解析功能: 增加函数以同时支持 IPv4 和 IPv6 主要有 `gethostbyname()`, `gethostbyaddr()`;

4) 地址转换功能: 增加函数以同时支持 IPv4 和 IPv6 主要有 `inet_ntoa()`, `inet_addr()`;

5) 协议无关功能: 实现协议无关; 主要程序段如下:

```
ConnSock= accept( ServSock[ i], (LPSOCKADDR)&From, &FromLen);
```

```
ReVal= sendto( ServSock[ i], Buffer AmountRead Q (LPSOCKADDR)&From, FromLen);
```

```
/* 协议无关节点名和服务名转换 */
```

```
int getnameinfo( const struct sockaddr* sa socklen_t salen, char* node, socklen_t nodelen, char* service,
socklen_t servicelen int flags); /* 查找 IP 地址和端口号, 返回: 0—成功, 非 0—失败 */
```

```
int getaddrinfo( const char* nodename const char* servname const struct addrinfo* hints struct addrinfo
* * res); /* 获取地址信息, 实现节点名和地址的转换 */
```

```
struct addrinfo{ // addrinfo 结构定义于 netdb.h 头文件
```

```
.....
```

```
};
```

6) 其他: 为 IPv6 增加跳数限制头, 控制 IPv6 多播包的转发。

```
/* IPV6_UNICAST_HOPS */
```

```
int hoplimit= 10
```

```
if ( setsockopt( $ IPPROTO_IPV6, IPV6_UNICAST_HOPS ( char* ) & hoplimit sizeof( hoplimit ) ) =
= - 1) /* 获取跳数限制 */
```

```
 perror( "setsockopt IPV6_UNICAST_HOPS");
```

```
int hoplimit
```

```
socklen_t len= sizeof( hoplimit);
```

```
if ( getsockopt( $ IPPROTO_IPV6, IPV6_UNICAST_HOPS ( char* ) & hoplimit & len ) == - 1)
```

```
 perror( "getsockopt IPV6_UNICAST_HOPS");
```

```
else
```

```
 printf( "Using %d for hop limit \n", hoplimit);
```

采用 6PE 方式使 MPLS 核心可以承载 IPv6 核心路由器只支持 MPLS and IPv4, 所以不能转发 IPv6 LMP, 也无法实现 IPv6 VPN.^[19]

IPv4 VPN 可以采用 M-BGP 配合 Spoke HUB 方式实现。^[8]

2.2.2 构建 6VPE

虽然 L2 MPLS VPN 可以很好的承载 IPv6 但由于各个厂商在 L2 MPLS VPN 上无法实现兼容性^[5], 所以原 MPLS VPN 骨干只实现 L3 MPLS VPN, 为了实现 IPv6 sites 之间的 VPN, 可以部署 6VPE 或者 MPLS VPN over GRE 方式, MPLS VPN over GRE 方式对 IPv6 接入用户的路由器有特殊要求。如图 3 所示, 对于需要接入 IPv6 VPN 的 PE 路由器, 需要同时启 IPv4 VRF 和 IPv6 VRF, 对于每个 IPv4 或 IPv6 VPN Sites 都在 6VPE 上对应一个虚拟子接口、路由表、转发表、路由实例以及路由策略, 6VPE 通过 RD+ IPv4+ IPv6 BGP NH 和 RD+ IPv6: FFFF: IPv4 BGP NH 进行路由交换。6VPE 只在需要实现 IPv6 VPN 业务时才启用^[10]。

2.3 PE 路由器配置

在云南省电子政务 MPLS VPN Core 上部署 6PE+6VPE, 需要对 PE 路由器进行重新升级、设置, P 路由器不需要进行任何改动, P 路由器仍然只运行原网络的路由协议和标签交换协议。在整个 MPLS VPN 主干中提供 IPv6 支持服务需要进行如下步骤^[3]:

● 指定源地址接口:

```
ipv6 address 2001:0DB8:FFFF::2/64 mpls ipv6 source-interface Loopback 0
```

● 指定并绑定 6PE 标签到 IPv6 前缀

```
no bgp default ipv4-unicast
```

```
neighbor 192.168.1.10 remote-as 65001
```

```
neighbor 192.168.1.10 update-source Loopback 0
```

```
address-family ipv6
```

```
neighbor 192.168.1.10 activate
```

neighbor 192.168.1.10 send-label

●配置 BGP 多路径共享 : maximum-paths bgp 5

●验证 6PE 配置 :

show ipv6 cef 2001:0DB8:DDDD: /64 show bgp ipv6 2001:0DB8:DDDD: /48

3 总结

3.1 应用效果

通过在 IPv4 MPLS VPN Core 上对 PE 路由器的改造, 不仅保持原网络中 PE 路由器完成标签交换、器维护 IPv4-VRF、承担路由反射器等基本功能, 还可以实现以下效果:

●同时支持 IPv4 和 IPv6 能够进行 IPv4 和 IPv6 路由存储与转发, 实现 IPv6 sites 与 IPv6 sites IPv4 sites 与 IPv6 sites 的连接

●同时支持 IPv4 VPN 和 IPv6 VPN, 而且增加 IPv6 VPN 的配置和 IPv4 VPN 的配置相同;

● IPv4 MPLS VPN Core 对 IPv6 Sites 透明, 即 IPv6 Sites 不需要关心 MPLS 标签交换;

●对于 IPv6 用户端而言, 不需要关心网络中其他 IPv6 Sites 的变化 (由 PE 路由器进行维护)。

3.2 不足及展望

论文未对在 IPv4 MPLS VPN 部署 6PE /6VPE 以承载 IPv6 带来的网络流量和 6PE 路由性能问题进行研究和阐述^[6]。同时, 6VPE 方式对于 PE 路由器的性能, 特别是信令开销比价大。

参考文献:

- [1] 华为 - 3Com 技术有限公司. IPv6 技术 [M/OL]. 北京: 清华大学出版社, 2004
- [2] van Peepeljak MPLS 和 VPN 体系结构 CCIP 版 [M]. 北京: 人民邮电出版社, 2003
- [3] Cisco Systems Inc. Implementing IPv6 over MPLS [M]. January, 2004
- [4] Hinden R. RFC 2460 IPv6 Addressing Architecture [M]. April 2003
- [5] Conta A, Deering S. RFC 2473 Generic Packet Tunneling in IPv6 Specification [M]. December 1998
- [6] Raghunathan R. Ed. RFC 4022 IP Version 6 Management Information Base for the Transmission Control Protocol [M] March 2005
- [7] Carpenter B, Moore K. RFC 3056 Connection of IPv6 Domains via IPv4 Clouds [M]. February 2001
- [8] 沈俊鑫. MPLS VPN 与 IPsec VPN 融合在电子政务中的应用 [D]. 昆明: 云南大学出版社, 2006
- [9] 沈俊鑫. MPLS VPN 技术及产品探究 [D]. 昆明: 云南大学出版社, 2006
- [10] Jeremy D C. Cisco Systems draft-ietf-l3vpn-bgp-ipv6-06 [M]. USA: Cisco Press February 2005
- [11] 吴晟, 苏庆堂, 罗斌. 基于 Socket 和多线程技术的并发服务器的研究 [D]. 昆明: 昆明理工大学出版社, 2006

(上接第 39 页)

4 结束语

基于 Pocket PC 数据库应用软件与基于桌面的数据库应用软件有所不同, 它要受到 Pocket PC 资源的限制, 要做到简单实用. 本文所讨论的有关方法对基于 Windows CE 应用平台的数据库应用程序开发具有一定的参考意义。

参考文献:

- [1] Chris M. Windows CE 权威指南 [M]. 北京: 中国电力出版社, 2001
- [2] Douglas B. Windows CE 程序设计 [M]. 北京: 北京大学出版社, 2001
- [3] 黄聪明. Pocket PC 数据库应用程序设计 [M]. 北京: 清华大学出版社, 2002
- [4] 田东风. Windows CE 应用程序设计 [M]. 北京: 机械工业出版社, 2003
- [5] 范盛荣. Windows Mobile 应用程序开发实践 [M]. 北京: 科学出版社, 2006
- [6] 孟宪国, 胡建旺, 王丙坤, 等. ADOCE 技术在 eVC++ 中的应用研究 [J]. 微计算机信息 (测控自动化), 2004 20(5): 64-65
- [7] 张新房, 吕跃刚, 徐大平, 等. 使用 Platform Builder 配置 Windows CE 操作系统 [J]. 单片机与嵌入式系统应用, 2002 (7): 8-11

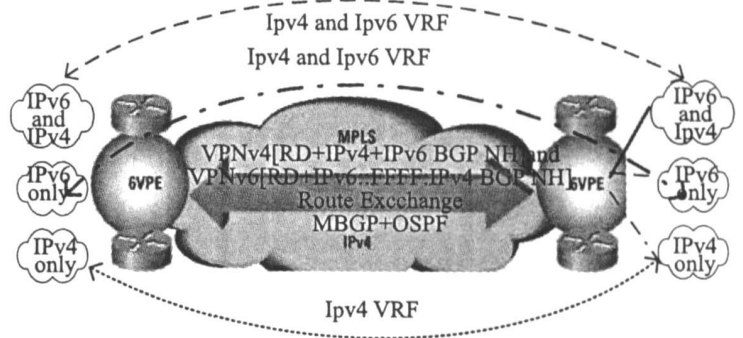


图3 6VPE逻辑图
Fig.3 6VPE logical fig