6_Operation07VPN 和 NTP

一 VNP

1.1 VPN:Virtual Private Network

1.1.1 在公共网络的基础上建立专用私有网络,进行加密通讯

多用于为集团公司的各地子公司建立连接

连接完成后,各个地区的子公司可以想局域网一样通讯

在企业网络中有广泛的应用

偶尔可以用于翻墙

1.1.2 目前主流的 VPN 技术:GRE,PPTP,L2TP+IPSec,SSL

搭建难度从左到右依次增加,安全性从左到右依次增加

lsmod 查看内核激活的功能

lsmod | wc -l 统计内核激活的程序数

modprobe ip_gre 内核激活 ip_gre

rmmod 程序名 关闭激活的程序

lsmod | grep pre

modinfo ip_gre

1.2 案例：配置 GRE VPN

要求:

搭建一个 GRE VPN 环境，并测试该 VPN 网络是否能够正常通讯，要求如下：

启用内核模块 ip_gre

创建一个虚拟 VPN 隧道(10.10.10.0/24)

实现两台主机点到点的隧道通讯

方案:

使用 lsmod 查看当前计算机已经加载的模块，使用 modprobe 加载 Linux 内核模块，使用 modinfo 可以查看内核模块的信息。

准备实验所需的虚拟机环境，实验环境所需要的主机及对应的 IP 设置列表如表-1 所示，正确配置 IP 地址、主机名称，并且为每台主机配置 YUM 源。

表－1 主机列表

主机名	IP 地址
client	eth3(201.1.2.10/24)
proxy	eth0(192.168.4.5/24) eth3(201.1.2.5/24)

实验拓扑如图-1 所示。

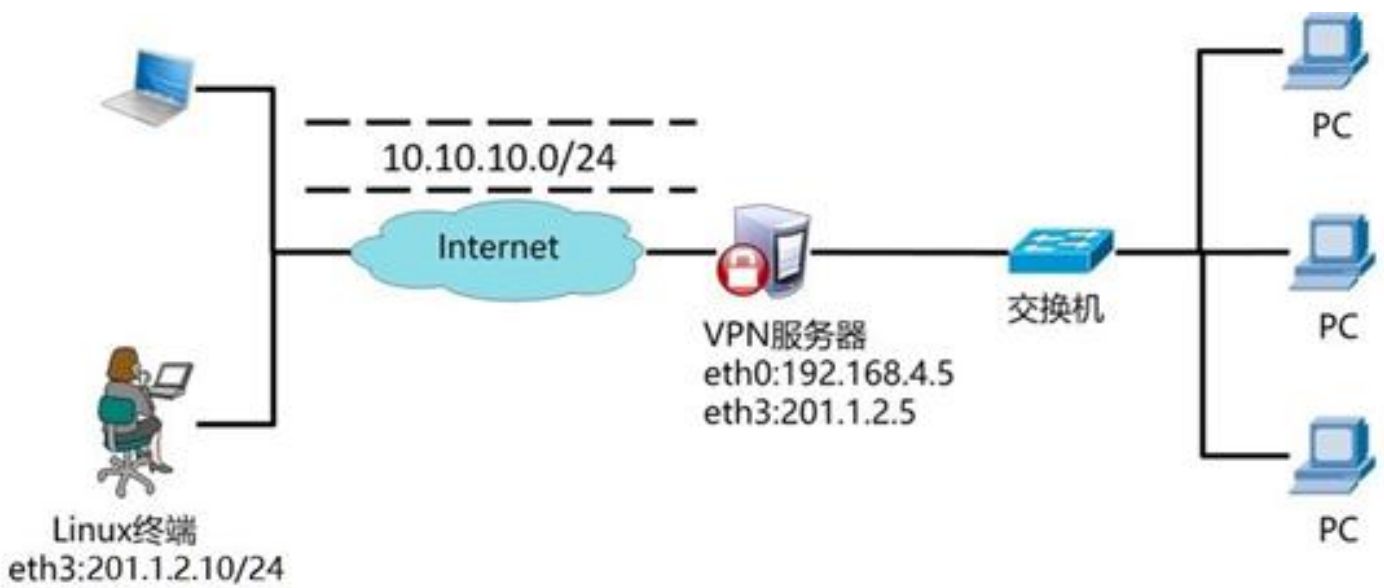

图-1

步骤一：启用 GRE 模块（client 和 proxy 都需要操作）

1）查看计算机当前加载的模块

```
client ~]# lsmod  #示模块列表

client ~]# lsmod  | grep ip_gre  #确定是否加载了 gre 模块
```

2)加载模块 ip_gre

```
client ~]# modprobe  ip_gre
```

3）查看模块信息

```
client ~]# modinfo ip_gre

filename:/lib/modules/3.10.0-693.el7.x86_64/kernel/net/ipv4/

ip_gre.ko.xz

alias:          netdev-gretap0

alias:          netdev-gre0

alias:          rtnl-link-gretap

alias:          rtnl-link-gre

license:        GPL

rhelversion:    7.4

srcversion:     F37A2BF90692F86E3A8BD15

depends:        ip_tunnel,gre

intree:         Y

vermagic:       3.10.0-693.el7.x86_64 SMP mod_unload modversions
```

signer: CentOS Linux kernel signing key

sig_key:

DA:18:7D:CA:7D:BE:53:AB:05:BD:13:BD:0C:4E:21:F4:22:B6:A4:9C

sig_hashalgo: sha256

parm: log_ecn_error:Log packets received with corrupted ECN (bool)

步骤二：Client 主机创建 VPN 隧道

ip 命令帮助

ip help;ip tunnel help;等

1）创建隧道

client ~]# **ip tunnel add tun0 mode gre **

> remote 201.1.2.5 local 201.1.2.10

ip tunnel add 创建隧道（隧道名称为 tun0）

ip tunnel help 可以查看帮助

#mode 设置隧道使用 gre 模式

#local 后为本机的 IP 地址，remote 后为与本机建立隧道的对方主机的 IP 地址

2）启用该隧道（类似与设置网卡 up）

client ~]# ip link show

client ~]# **ip link set tun0 up** #启用隧道 tun0

client ~]# **ip link show(ip a s)**

3）为 VPN 配置隧道 IP 地址

client ~]# **ip addr add 10.10.10.10/24 peer 10.10.10.5/24 **

> dev tun0

#为隧道 tun0 设置本机隧道 IP 地址为:10.10.10.10/24

#为隧道 tun0 对面主机隧道 IP 地址为:10.10.10.5/24

client ~]# ip a s #查看 IP 地址

4）关闭防火墙

client ~]# firewall-cmd --set-default-zone=trusted

步骤三：Proxy 主机创建 VPN 隧道

1）查看计算机当前加载的模块

client ~]# **lsmod** #显示模块列表

client ~]# **lsmod | grep ip_gre** #确定是否加载了 gre 模块

2)加载模块 ip_gre

client ~]# **modprobe ip_gre**

3）创建隧道

proxy ~]# ~]# **ip tunnel add tun0 mode gre **

> remote 201.1.2.10 local 201.1.2.5

#ip tunnel add 创建隧道（隧道名称为 tun0），ip tunnel help 可以查看帮助

#mode 设置隧道使用 gre 模式

#local 后为本机的 IP 地址，remote 后为与本机建立隧道的对方主机的 IP 地址

4）启用该隧道（类似与设置网卡 up）

```
proxy ~]# ip link show
proxy ~]# ip link set tun0 up          #启用隧道 tun0
proxy ~]# ip link show
```

5）为 VPN 配置隧道 IP 地址

```
proxy ~]# ip addr add 10.10.10.5/24 peer 10.10.10.10/24 \
>  dev tun0
```

#为隧道 tun0 设置本地 IP 地址（10.10.10.10.5/24）

#隧道对面的主机 IP 的隧道 IP 为 10.10.10.10/24

```
proxy ~]# ip a s                       #查看 IP 地址
```

6）开启路由转发、关闭防火墙

```
proxy ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

#只能用重 echo+重定向修改值为 0 或 1;中继的主机必须开启此设置,作为软路由,连接公网,并在软路由上设置网络地址转换(NAT)[实验中 proxy 为中继的主机]

```
proxy ~]# firewall-cmd --set-default-zone=trusted
```

7)测试连通性

```
client ~]#  ping 10.10.10.5
proxy ~]#   ping 10.10.10.10
```

1.3 案例：创建 PPTP VPN

1.3.1 要求

搭建一个 PPTP VPN 环境，并测试该 VPN 网络是否能够正常通讯，要求如下：

使用 PPTP 协议创建一个支持身份验证的隧道连接

使用 MPPE 对数据进行加密

为客户端分配 192.168.3.0/24 的地址池

客户端连接的用户名为 jacob，密码为 123456

1.3.2 方案

准备实验所需的虚拟机环境，实验环境所需要的主机及对应的 IP 设置列表如表-2 所示，正确配置 IP 地址、主机名称，并且为每台主机配置 YUM 源。

表一2 主机列表

主机名	IP 地址
windows 主机	网卡桥接 public2(201.1.2.20/24)
proxy	eth0(192.168.4.5/24) eth3(201.1.2.5/24)

实验拓扑如图-2 所示。

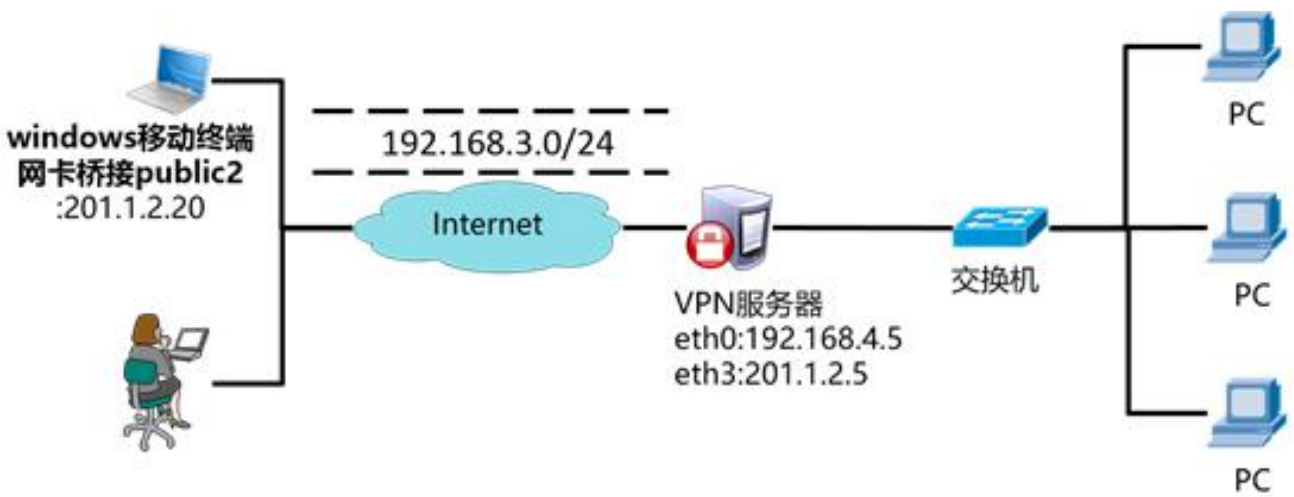

图-2

1.3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署 VPN 服务器

1）安装软件包（软件包参考 lnmp_soft）

```
proxy ~]# yum localinstall pptpd-1.4.0-2.el7.x86_64.rpm
proxy ~]# rpm -qc pptpd
/etc/ppp/options.pptpd
/etc/pptpd.conf
/etc/sysconfig/pptpd
```

2)修改配置文件(设置 IP 地址池\加密和 DNS)

```
proxy ~]# vim /etc/pptpd.conf    #设置 IP 地址池
```

最后 2 行,解除注释并修改

```
localip 201.1.2.5              #服务器本地 IP
remoteip 192.168.3.1-50,192.168.3.100-150
      #分配给客户端的 IP 池(VPN 的 IP 地址池),多段 ip 用,号分隔

proxy ~]# vim /etc/ppp/options.pptpd  #设置加密和 DNS
require-mppe-128               #使用 MPPE 加密数据
ms-dns 8.8.8.8                 #设置 DNS 服务器,给客户机域名解析使用
proxy ~]# vim /etc/ppp/chap-secrets   #修改配置文件设置账户和密码
```

```
jacob       *           123456      *
用户名        服务器标记    密码          客户端
proxy ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
      #开启路由转发
```

3）启动服务

```
proxy ~]# systemctl start pptpd
proxy ~]# systemctl enable pptpd
proxy ~]# firewall-cmd --set-default-zone=trusted
```

4）翻墙设置（非必需操作）

```
proxy ~]# iptables -t nat -A POSTROUTING -s 192.168.3.0/24 \
>  -j SNAT --to-source 201.1.2.5
```

步骤二：客户端设置

启动一台 Windows 虚拟机，将虚拟机网卡桥接到 public2，配置 IP 地址为 201.1.2.20。

新建网络连接（具体操作如图-3 所示），输入 VPN 服务器账户与密码（具体操作如图-4 所示），连接 VPN 并测试网络连通性（如图-5 所示）。


网络和共享中心
控制面板 ▸ 所有控制面板项 ▸ 网络和共享中心
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
控制面板主页
更改适配器设置
更改高级共享设置
查看基本网络信息并设置连接
(此计算机)
查看活动网络
未识别的网络
公用网络
更改网络设置
设置新的连接或网络
连接到网络
连接到 Internet
设置无线、宽带或拨号连接，连接到 Internet。
连接到工作区
设置到您的工作区的拨号或 VPN 连接。
您想如何连接?
使用我的 Internet 连接(VPN)(I)
通过 Internet 使用虚拟专用网络(VPN)来连接
设置 Internet 连接(S)
我将稍后设置 Internet 连接(I)

图-3


连接到工作区
键入要连接的 Internet 地址
网络管理员可提供此地址。
Internet 地址(I):
201.1.2.5
目标名称(E):
VPN 连接
连接到工作区
键入您的 Internet 服务提供商(ISP)提供的信息
用户名(U):
jacob
密码(P):
显示字符(S)
记住此密码(R)
连接名称(N):
宽带连接
允许其他人使用此连接(A)
这个选项允许可以访问这台计算机的人使用此连接。

图-4

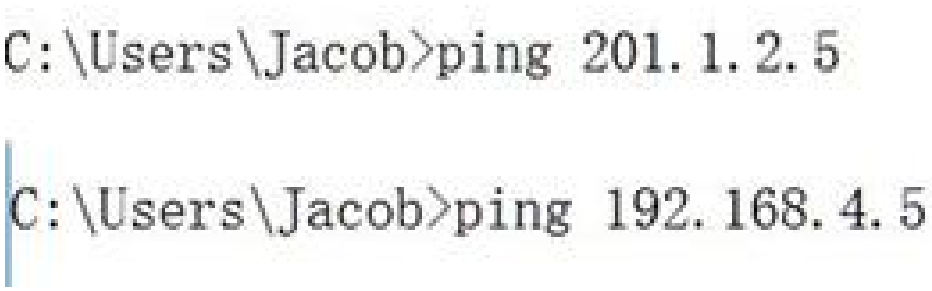

图-5

1.4 案例:创建 L2TP+IPSec VPN

1.4.1 要求

搭建一个 L2TP+IPSec VPN 环境，并测试该 VPN 网络是否能够正常通讯，具体要求如下：

使用 L2TP 协议创建一个支持身份验证与加密的隧道连接

使用 IPSec 对数据进行加密

为客户端分配 192.168.3.0/24 的地址池

客户端连接的用户名为：jacob，密码为：123456

预共享密钥为：randpass PSK:PRE SHARED KEY

1.4.2 方案

准备实验所需的虚拟机环境，实验环境所需要的主机及对应的 IP 设置列表如表-3 所示，正确配置 IP 地址、主机名称，并且为每台主机配置 YUM 源。

表一3 主机列表

主机名	IP 地址
windows 主机	网卡桥接 public2(201.1.2.20/24)
client(作为 vpn 服务器)	eth0(192.168.4.10/24) eth3(201.1.2.10/24)

实验拓扑如图-6 所示。

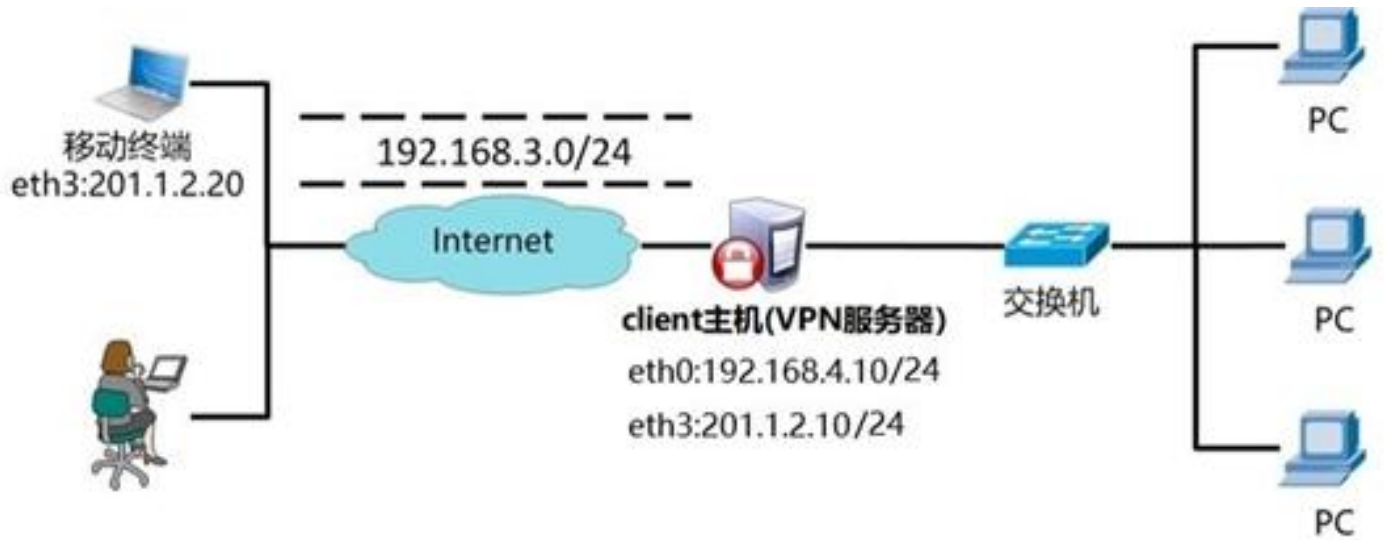

图-6

1.4.3 步骤

步骤一：部署 IPSec 服务

1）安装软件包

```
client ~]# yum -y install libreswan
```

2)新建 IPSec 密钥验证配置文件(lnmp/php 内 myipsec.conf)

```
client ~]# cat /etc/ipsec.conf  #仅查看一下该主配置文件
.. ..
include /etc/ipsec.d/*.conf  #加载该目录下的所有配置文件
```

```
[root@client ~]# vim /etc/ipsec.d/myipsec.conf
      #新建该文件并修改，参考 lnmp_soft/vpn/myipsec.conf
conn IDC-PSK-NAT
    rightsubnet=vhost:%priv  #允许建立的 VPN 虚拟网络
    also=IDC-PSK-noNAT

conn IDC-PSK-noNAT
    authby=secret                                #加密认证
        ike=3des-sha1;modp1024                   #算法
        phase2alg=aes256-sha1;modp2048           #算法
    pfs=no
    auto=add
    keyingtries=3
    rekey=no
    ikelifetime=8h
    keylife=3h
    type=transport
    left=201.1.2.10          #重要，服务器本机的外网 IP
    leftprotoport=17/1701
    right=%any               #允许任何客户端连接
```

rightprotoport=17/%any

3)创建 **IPSec** 预定义共享密钥

client ~]# cat /etc/ipsec.secrets　　#仅查看，不要修改该文件

include **/etc/ipsec.d/*.secrets**　**#预共享秘钥的存放路径和命名方式**

[root@client ~]# **vim /etc/ipsec.d/mypass.secrets**

　　#创建预定义共享密钥并修改

201.1.2.10　%any:　PSK　"randpass"　#randpass 为预共享密钥

　　#201.1.2.10 是 VPN 服务器的 IP

4)启动 **IPSec** 服务

client ~]# systemctl start ipsec

client ~]# netstat -ntulp |grep pluto

udp　0　0　127.0.0.1:4500　0.0.0.0:*　3148/pluto

udp　0　0　192.168.4.10:4500　0.0.0.0:*　3148/pluto

udp　0　0 201.1.2.10:4500　0.0.0.0:*　3148/pluto

udp　0　0 127.0.0.1:500　0.0.0.0:*　3148/pluto

udp　0　0 192.168.4.10:500　0.0.0.0:*　3148/pluto

udp　0　0　201.1.2.10:500　0.0.0.0:*　3148/pluto

udp6　0　0 ::1:500　:::*　3148/pluto

步骤二：部署 **XL2TP** 服务

1）安装软件包（软件包参考 lnmp_soft/php）

client ~]# yum localinstall xl2tpd-1.3.8-2.el7.x86_64.rpm

2) 修改 xl2tp 配置文件（修改 3 个配置文件的内容）

client ~]# **vim /etc/xl2tpd/xl2tpd.conf** #修改主配置文件

[global]

.. ..

[lns default]

ip range = 192.168.3.128-192.168.3.254 #分配给客户端的 IP 池

local ip = 201.1.2.10 #VPN 服务器的 IP 地址

client ~]# **vim /etc/ppp/options.xl2tpd #认证配置**

require-mschap-v2 #添加一行，强制要求认证

#crtscts #注释或删除该行

#lock #注释或删除该行

root@client ~]# **vim /etc/ppp/chap-secrets #修改用户和密码文件**

jacob * 123456 *

#账户名称 服务器标记 密码 客户端 IP

3）启动服务

client ~]# systemctl start xl2tpd

```
client ~]# netstat  -ntulp |grep xl2tpd
udp  0  0 0.0.0.0:1701  0.0.0.0:*  3580/xl2tpd
```

4）设置路由转发，防火墙

```
client ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
client ~]# firewall-cmd --set-default-zone=trusted
```

5）翻墙设置（非必需操作）

```
client ~]# iptables -t nat -A POSTROUTING -s 192.168.3.0/24 \
>  -j SNAT --to-source 201.1.2.10
```

步骤三：客户端设置

启动一台 Windows 虚拟机，将虚拟机网卡桥接到 public2，配置 IP 地址为 201.1.2.20。

1. 新建网络连接（参考案例 2），输入 VPN 服务器账户与密码（参考案例 2）。

设置 VPN 连接的属性，预共享密钥是 IPSec 配置文件中填写的 randpass，具体操作如图-7 所示。

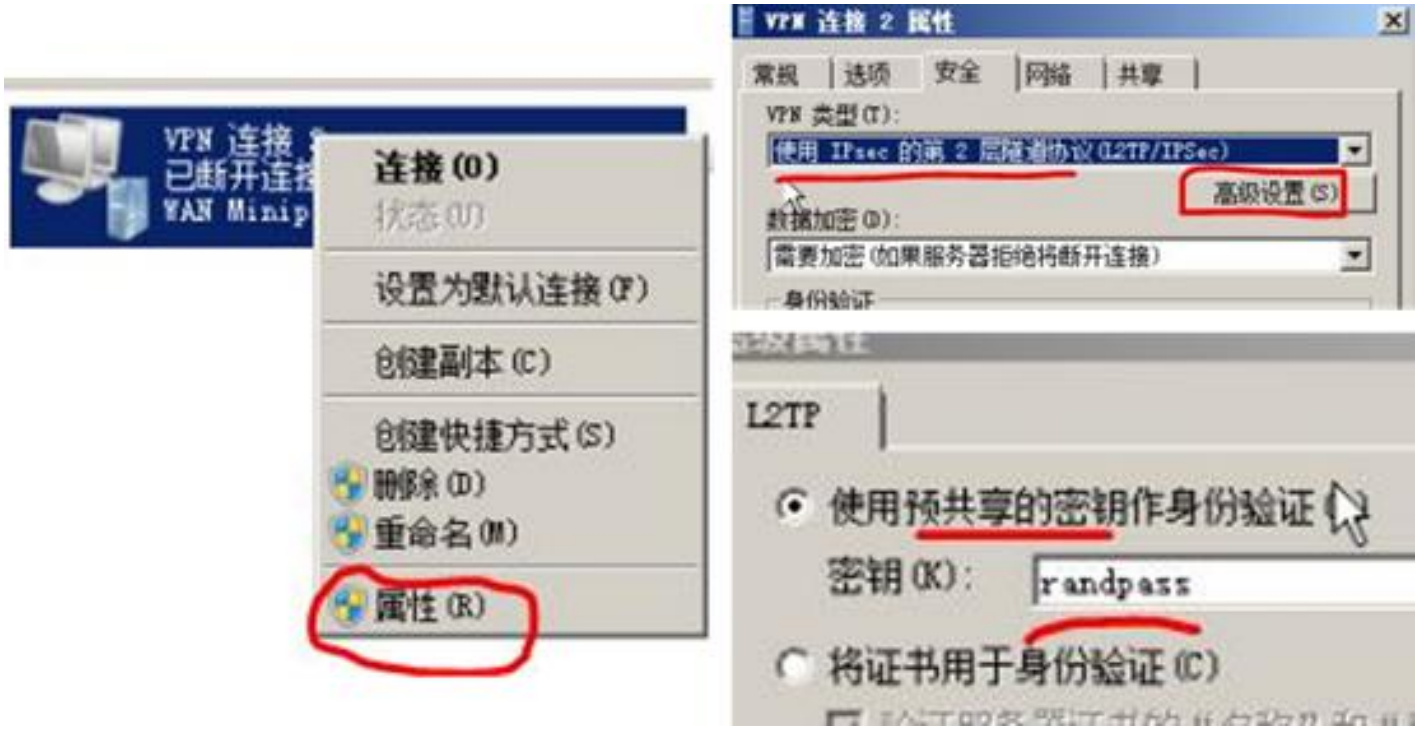

图-7

2．设置 Windows 注册表（不修改注册表，连接 VPN 默认会报 789 错误），具体操作如下：

单击"开始"，单击"运行"，键入"regedit"，然后单击"确定"

找到下面的注册表子项，然后单击它：

HKEY_LOCAL_MACHINE\

System\CurrentControlSet\Services\Rasman\Parameters

在"编辑"菜单上，单击"新建"->"DWORD 值"

在"名称"框中，键入"ProhibitIpSec"

在"数值数据"框中，键入"1"，然后单击"确定"

退出注册表编辑器，然后重新启动计算机

连接 VPN 并测试网络连通性（参考案例 2）。

二 案例：NTP 时间同步

2.1 问题

本案例要求搭建一个 NTP 服务器，为整个网络环境中的所有主机提供时间校准服务，

具体要求如下：

部署一台 NTP 时间服务器

设置时间服务器上层与 0.centos.pool.ntp.org 同步

设置本地服务器层级数量为 10

允许 192.168.4.0/24 网络的主机同步时间

客户端验证时间是否同步

2.2 方案

准备实验所需的虚拟机环境，实验环境所需要的主机及对应的 IP 设置列表如表-4 所

示，正确配置 IP 地址、主机名称，并且为每台主机配置 YUM 源。

表一4 主机列表

主机名	IP 地址
client	eth0 (192.168.4.10/24)
proxy	eth0(192.168.4.5/24) eth1(192.168.2.5/24)

实验拓扑如图-8 所示。

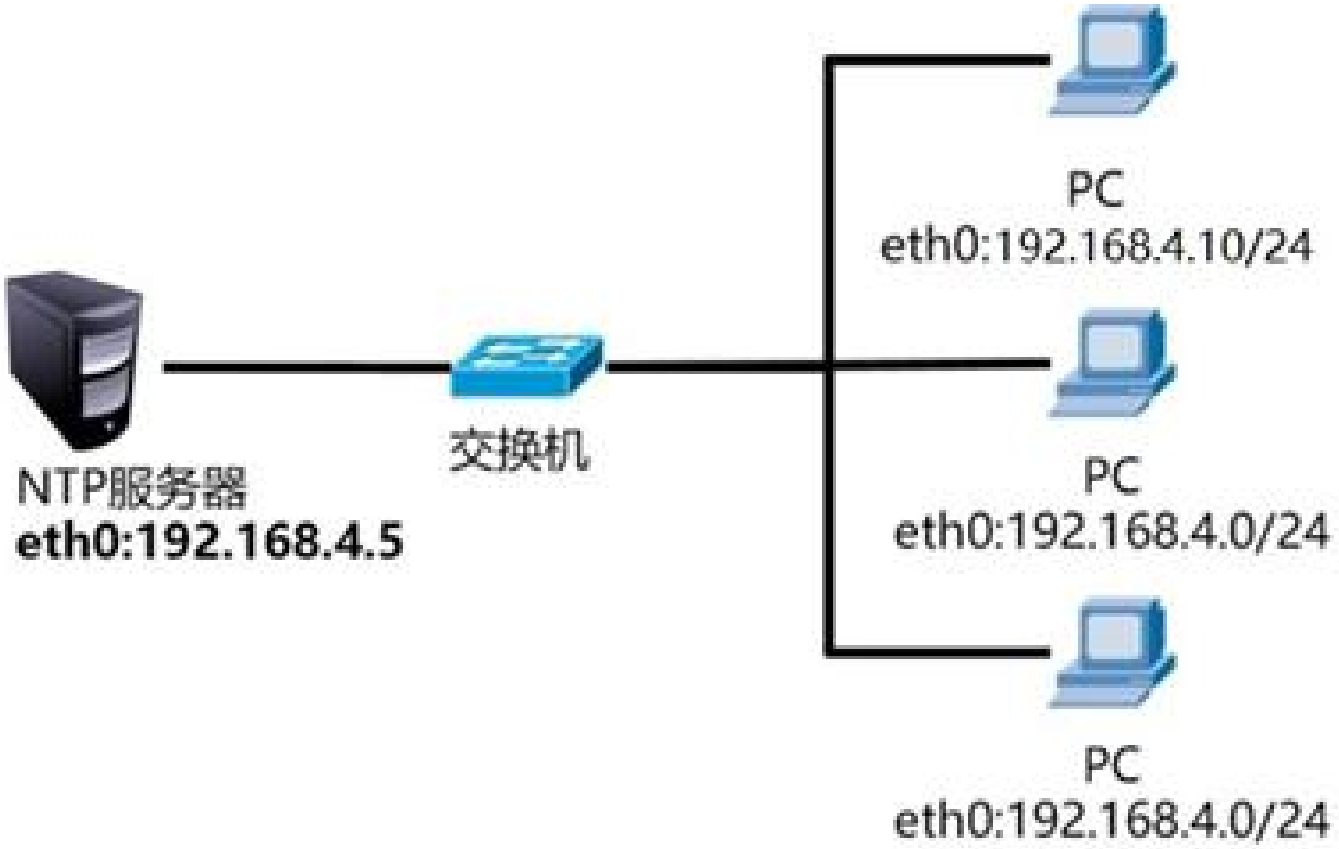

图-8

Network Time Protocol 网络时间协议)采用的是分层设计，如图-9所示，**Stratum层的总数限制在 15 以内（包括 15）**。

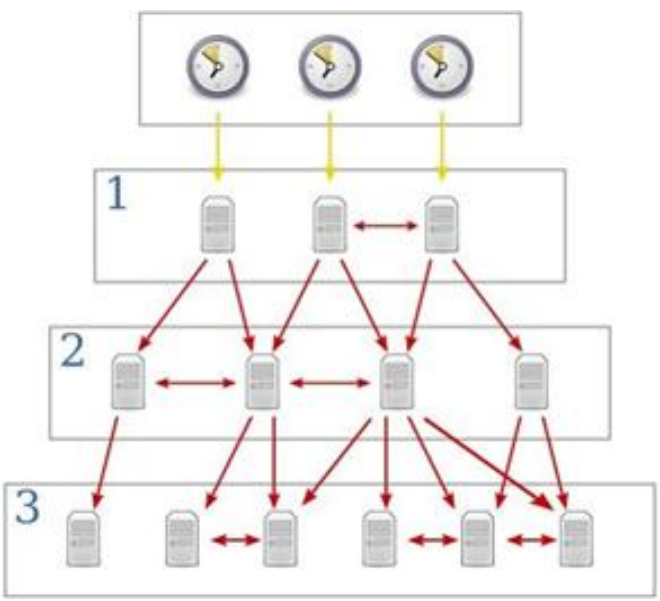

图-9

2.3 步骤

步骤一：部署 **NTP** 服务

1）安装软件包

proxy ~]# **yum -y install chrony**

proxy ~]# rpm -qc chrony #查看配置文件列表

/etc/chrony.conf

/etc/chrony.keys

.. ..

2)修改配置文件

proxy ~]# cat **/etc/chrony.conf**

.. ..

server 0.centos.pool.ntp.org iburst

#server 用户客户端指向上层 NTP 服务器

allow 192.168.4.0/24

#解除注释,激活本机为时间服务器;允许哪个 IP 或网络访问 NTP

#deny 192.168.4.1 #拒绝哪个 IP 或网络访问 NTP

local stratum 10 #解除注释;设置 NTP 服务器的层数量

.. ..

3)启动 **NTP** 服务

proxy ~]# systemctl restart chronyd

```
proxy ~]# systemctl  enable   chronyd
```

4)设置防火墙

```
proxy ~]# firewall-cmd --set-default-zone=trusted
```

步骤二：配置客户端

1）安装软件包

```
client ~]# yum -y install chrony
```

2) 修改配置文件

```
client ~]# vim /etc/chrony.conf
```

```
server 192.168.4.5
```

```
      #设置与哪台服务器同步数据
     #iburst 参数设置重启服务后尽快同步时间
```

3) 将客户端时间修改为错误的时间

```
client ~]#  date  -s  "hour:minute"  #调整时间（小时：分钟）
client ~]#  date  #查看修改后的时间
```

4) 重启 chrony 与服务器同步时间

```
client ~]# systemctl  restart  chronyd
```

5) 确认时间是否已经同步

```
client ~]# date  #多执行几次查看结果
```