虚拟专用网络

N8.1. SSH 隧道与 VPN 隧道的主要区别是什么？

N8.2. Bob 使用基于 TLS 的 VPN 接入雪城大学网络（`128.230.0.0/16`）。隧道建立后，他在本机看到如下路由表：

```
---------------------------------------------------------------
  Network
Destination    Netmask          Gateway         Interface
---------------------------------------------------------------
    0.0.0.0    0.0.0.0          192.168.0.1     192.168.0.13
  127.0.0.0    255.0.0.0        On-link         127.0.0.1
  127.0.0.1    255.255.255.255  On-link         127.0.0.1
128.230.0.0    255.255.0.0      128.230.153.48  128.230.153.80
128.230.153.12 255.255.255.255  192.168.0.1     192.168.0.13
128.230.153.80 255.255.255.255  On-link         128.230.153.80
192.168.0.0    255.255.255.0    On-link         192.168.0.13
192.168.0.13   255.255.255.255  On-link         192.168.0.13
192.168.0.255  255.255.255.255  On-link         192.168.0.13
```

请根据上述路由信息回答问题并说明理由。

(a) Bob 机器上 TUN 接口的 IP 地址是多少？

(b) 雪城大学 VPN 服务器的 IP 地址是多少？

(c) 机器物理网卡的 IP 地址是多少？

(d) 假设本地防火墙阻止 Bob 访问 IP 为 `1.2.3.4` 的网站。他如何借助该 VPN 绕过限制？若需修改路由表，请明确说明改动。

N8.3. 图 1 中，机器 X 与机器 Y 建立了 VPN；Y 是连接到专用网络 `10.0.20.0/24` 的 VPN 服务器。X 上的用户现在可访问该专用网络，并运行 `"telnet 10.0.20.100"`。图中给出了相应报文流。请回答以下问题：

(a) 报文❶ 和❷ 之间有什么关系？

(b) 报文❸ 和❹ 之间有什么关系？

(c) 报文 ❶、❷、❸ 和 ❹ 的源 IP 与目的 IP 分别是什么？

(d) 机器 X 上需要哪些路由条目？

(e) 机器 Y 上需要哪些路由条目？

(f) 机器 `10.0.20.100` 上需要哪些路由条目？

(g) VPN 隧道中断时，`telnet` 连接会立即断开吗？几秒后重建 X 与 Y 之间的隧道，又会发生什么？

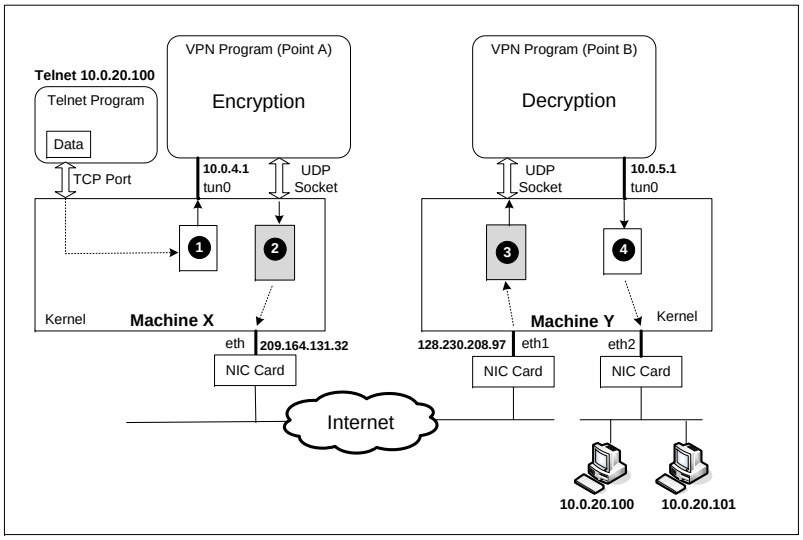

图 1: VPN 上的数据包流（针对问题 N8.3.）

N8.4. 主机 U 位于专用网络 `192.168.60.0/24`。

主机 V 位于专用网络 `192.168.80.0/24`。VPN 使两台主机能够通信。网络结构见图 2。请描述以下内容：

(a) 主机 U、VPN 客户端、VPN 服务器和主机 V 分别需要添加哪些路由？无需写出具体命令，但要说明路由条目。

(b) 主机 V 收到主机 U 发来的报文时，源 IP 地址是什么？

(c) VPN 服务器通过隧道收到主机 U 发往主机 V 的外层报文时，该报文的源 IP 与目的 IP 是什么？

(d) VPN 隧道建立后，从主机 U ping 主机 V。请详细说明 ICMP Echo Request 如何从 U 到达 V，以及 Echo Reply 如何返回 U。

N8.5. 使用 VPN 访问被本地防火墙阻止的 Facebook 时，我们把发往 Facebook 的报文路由到 TUN 接口，经隧道交给 VPN 服务器，再由服务器转发到互联网。Facebook 的响应会直接发给客户端，还是先发到 VPN 服务器再经隧道返回？为什么？

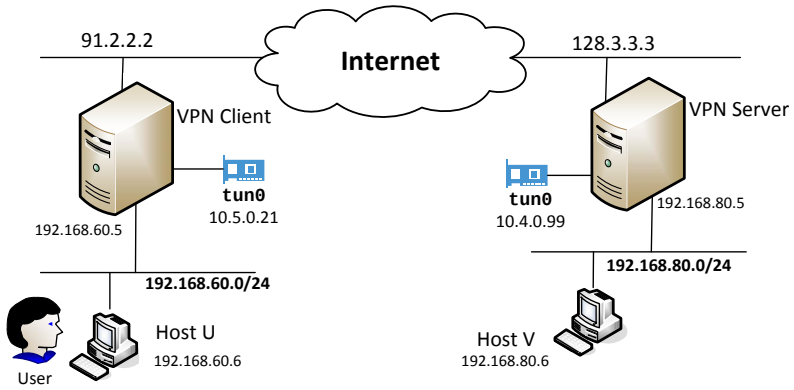

图 2: 问题 N8.4. 的网络拓扑

N8.6. 某网站根据访问者 IP 的地理位置，只允许来自加利福尼亚州的主机访问。你位于纽约州锡拉丘兹，如何访问该网站？

N8.7. 以下代码创建 TUN 接口。该接口可能的名称是什么？

```
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'abc%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)
```

N8.8. 教材示例使用 `os.read(tun, 2048)`，每次最多从 TUN 接口读取 2048 字节。IP 数据报理论上可能更大，是否需要增大这个值？若需要，应增大到多少？接口属性如下。

```
$ ifconfig
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>
    mtu 1500 inet 192.168.53.99
    netmask 255.255.255.0  destination 192.168.53.99
    ...
```

N8.9. 从配置了 `IFF_NO_PI` 的 TUN 接口读取 IPv4 报文时，数据的第一个字节最可能是哪一个：（A）`0x12`，（B）`0x23`，（C）`0x34`，（D）`0x45`，（E）`0x56`？

N8.10. `"ip route"` 命令显示以下结果。

```
$ ip route
default via 10.0.5.1 dev enp0s3 proto dhcp
10.0.5.0/24 dev enp0s3 proto kernel scope link src 10.0.5.5
```

```
192.168.53.0/24 dev tun0 proto ... src 192.168.53.99
10.0.6.0/24 dev tun0 scope link
```

分别向以下地址发送报文时，每个报文的源 IP 是什么？

1. 1.1.1.1
2. 10.0.5.9
3. 192.168.53.7
4. 10.0.6.6

N8.11. 为什么 VPN 服务器需要启用 IP 转发？

N8.12. 建立 VPN 后，在所有接口（lo、tun0 和 eth0）上运行 tcpdump，观察到以下报文。请指出每行来自哪个接口。

```
root@client:# tcpdump -n -i any
IP 10.0.53.99 > 10.0.8.5: ICMP echo request, id 94, seq 282    ①
IP 10.0.7.5.38018 > 10.0.7.11.9090: UDP                        ②
IP 10.0.7.11.9090 > 10.0.7.5.38018: UDP                        ③
IP 10.0.8.5 > 10.0.53.99: ICMP echo reply, id 94, seq 282      ④
```

N8.13. A 与 B 之间通过 VPN 建立了 TCP 连接。由于硬件故障，VPN 客户端短暂宕机数秒，随后恢复并重新连接 VPN 服务器。这会影响 A 与 B 之间的 TCP 连接吗？

N8.14. 在未使用 host 网络模式的 Docker 容器内嗅探时，通常只能捕获进出本容器的报文，无法看到其他容器之间的通信。为什么？

N8.15. TUN 依靠三层路由接收 IP 报文。二层 TAP 接口如何获得以太网帧？

N8.16. 教材示例为什么使用 select() 系统调用？

N8.17. 学生实现了以下简单 TUN 客户端，用于打印接口收到的报文，但把 IFF_NO_PI 的值误写成 0x100；正确值应为 0x1000。

```
TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_NO_PI = 0x100

tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)
```

```
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
   packet = os.read(tun, 2048)
   if packet:
      pkt = IP(packet)
      print(pkt.summary())
```

当学生从同一台机器运行 "ping 10.0.53.1" 时，程序打印出如下异常结果。请说明 IFF_NO_PI 的作用，并解释输出为何会被错误解析。

```
root@aed05427c19d:/volumes# ./tun.py
Interface Name: tun0
64.1.81.52 > 10.0.53.99 udp frag:84 / Raw
64.1.80.77 > 10.0.53.99 248 frag:84 / Raw
64.1.80.43 > 10.0.53.99 26 frag:84 / Raw
64.1.79.139 > 10.0.53.99 186 frag:84 / Raw
```