

# 实验三 VPN实验

---

张云鹤、肖凌、王美珍、梅松

# 主要内容

---

- 实验目的
- 实验环境
- 实验内容
- 实验要求
- 报告提交

# 1 实验目的

---

- ❑ 掌握**VPN**的网络和安全技术。为实现这一目标，要求学生实现简单的**TLS/SSL VPN**。
- ❑ 这个**TLS/SSL VPN**的设计和实现体现了许多安全原则，包括以下内容：
  - ❑ ●虚拟专用网络
  - ❑ ●**TUN/TAP**和**IP**隧道
  - ❑ ●路由
  - ❑ ●公钥加密，**PKI**和**X.509**证书
  - ❑ ●**TLS/SSL**编程
  - ❑ ●身份认证

## 2 实验内容

---

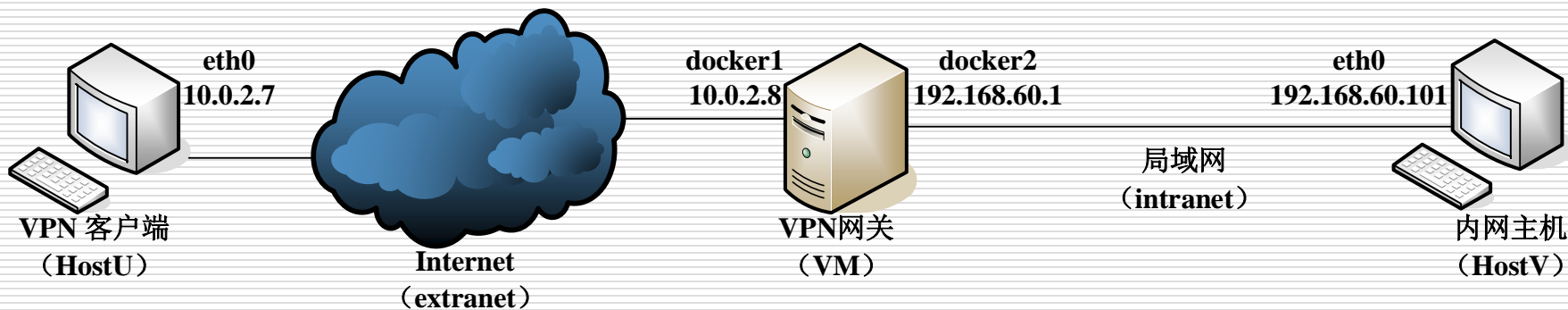
- ☐ 网络环境搭建
- ☐ 建立**VPN**隧道
- ☐ 加密隧道
- ☐ 身份认证
- ☐ 多客户端支持

## 2.1 网络环境搭建

---

- Ubuntu Seed 虚拟机下载: QQ群
- ubuntu系统的用户密码: seed: dees
- root密码: seedubuntu
- 实验需要多台虚拟机, 可以采用虚拟机 + docker容器 构建

## 2.1 网络环境搭建



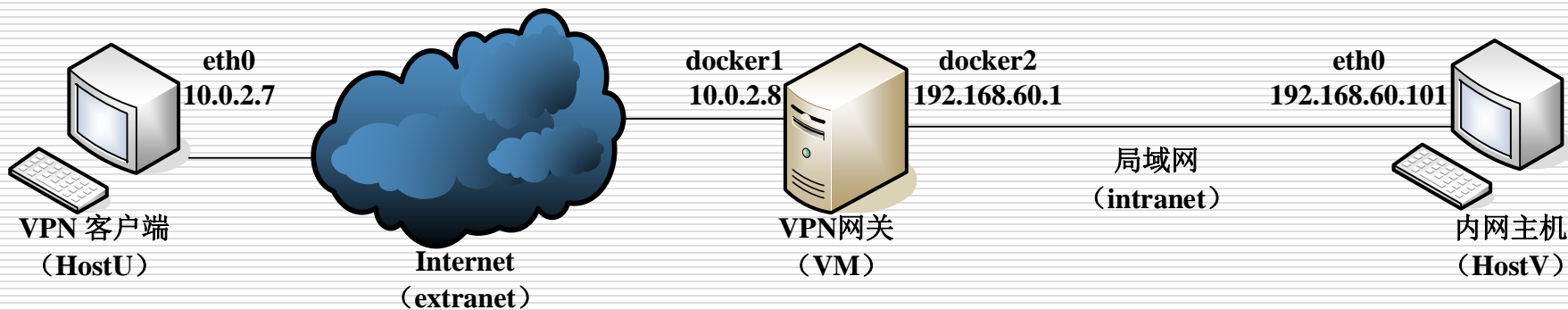
VPN客户端网络配置 (IP、掩码)

VPN服务器端网络配置 (双网卡) (IP、掩码)

服务端内网主机网络配置 (IP、掩码、网关)

**HostU能否访问到HostV?**

## 2.1 网络环境搭建



VPN服务器（网关）——双网卡，用VM自身做  
HostU、HostV分别用2个容器做  
需要建立2个docker网络extranet（模拟Internet）、intranet（模拟局域网）

**如何配置？**

## 2.1 网络环境搭建

- 在 VM 上创建 docker 网络 extranet

```
$ sudo docker network create --subnet=10.0.2.0/24 --gateway=10.0.2.8 --opt  
"com.docker.network.bridge.name"="docker1" extranet
```

- 在 VM 上创建 docker 网络 intranet

```
$ sudo docker network create --subnet=192.168.60.0/24 --gateway=192.168.60.1  
-- opt "com.docker.network.bridge.name"="docker2" intranet
```

- 在 VM 上新开一个终端，创建并运行容器 HostU

```
$sudo docker run -it --name=HostU --hostname=HostU --net=extranet --  
ip=10.0.2.7 --privileged "seedubuntu" /bin/bash
```

- 在 VM 上新开一个终端，创建并运行容器 HostV

```
$sudo docker run -it --name=HostV --hostname=HostV --net=intranet --  
ip=192.168.60.101 --privileged "seedubuntu" /bin/bash
```

- 在容器 HostU 上删除掉默认路由

```
# route del default
```



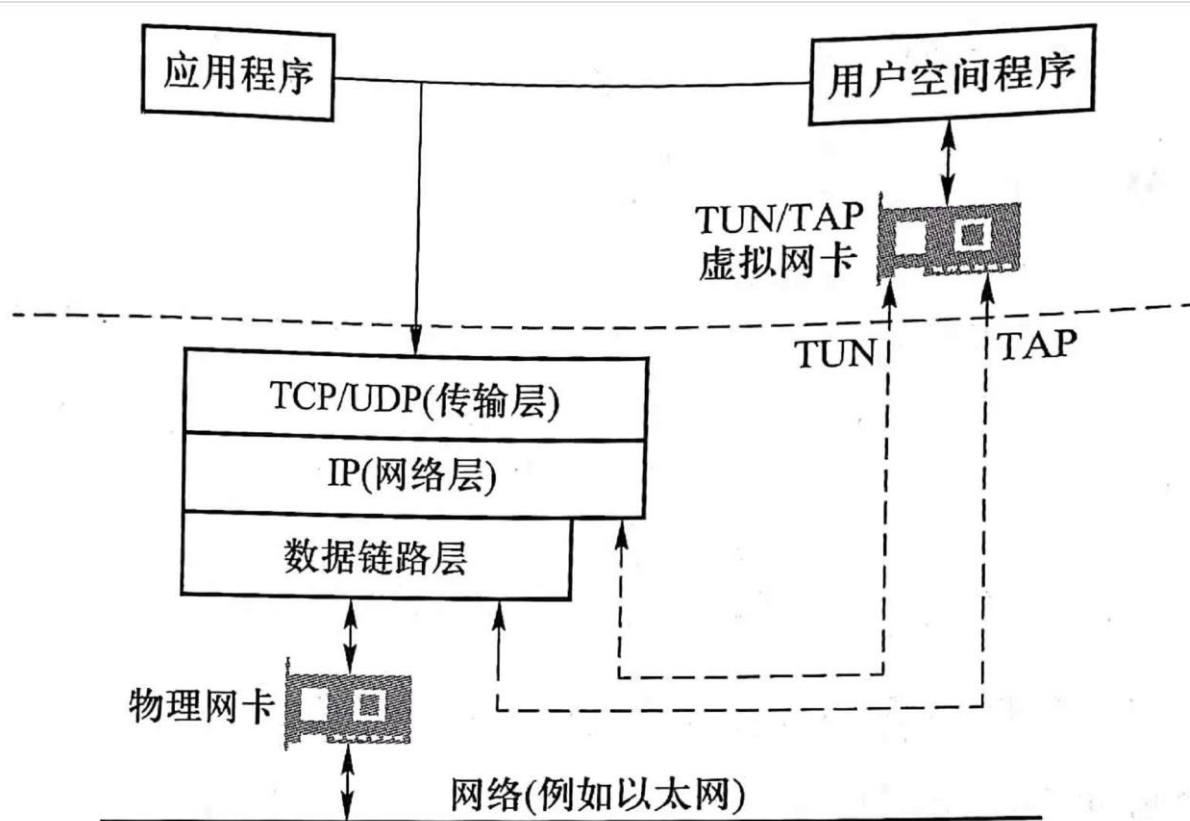
## 2.2 建立VPN隧道

---

- 使用**TUN/TAP**创建一个主机到主机的隧道
  - TLS/SSL VPN中使用了TUN/TAP技术，TUN和TAP是虚拟网络内核驱动程序，linux直接支持
  - TAP模拟以太网设备，处理的是以太网帧等二层数据包；TUN模拟网络层设备，处理的是IP等三层数据包
  - 我们可以用TAP/TUN创建虚拟网络接口。

## 2.2 建立VPN隧道

### □ TUN接口和物理接口



到达TUN接口的IP报文，用户空间程序可以直接通过/dev/tun设备读取；

因此，IP报文在用户空间程序中作为数据在套接字通信中传输。

## 2.2 建立VPN隧道

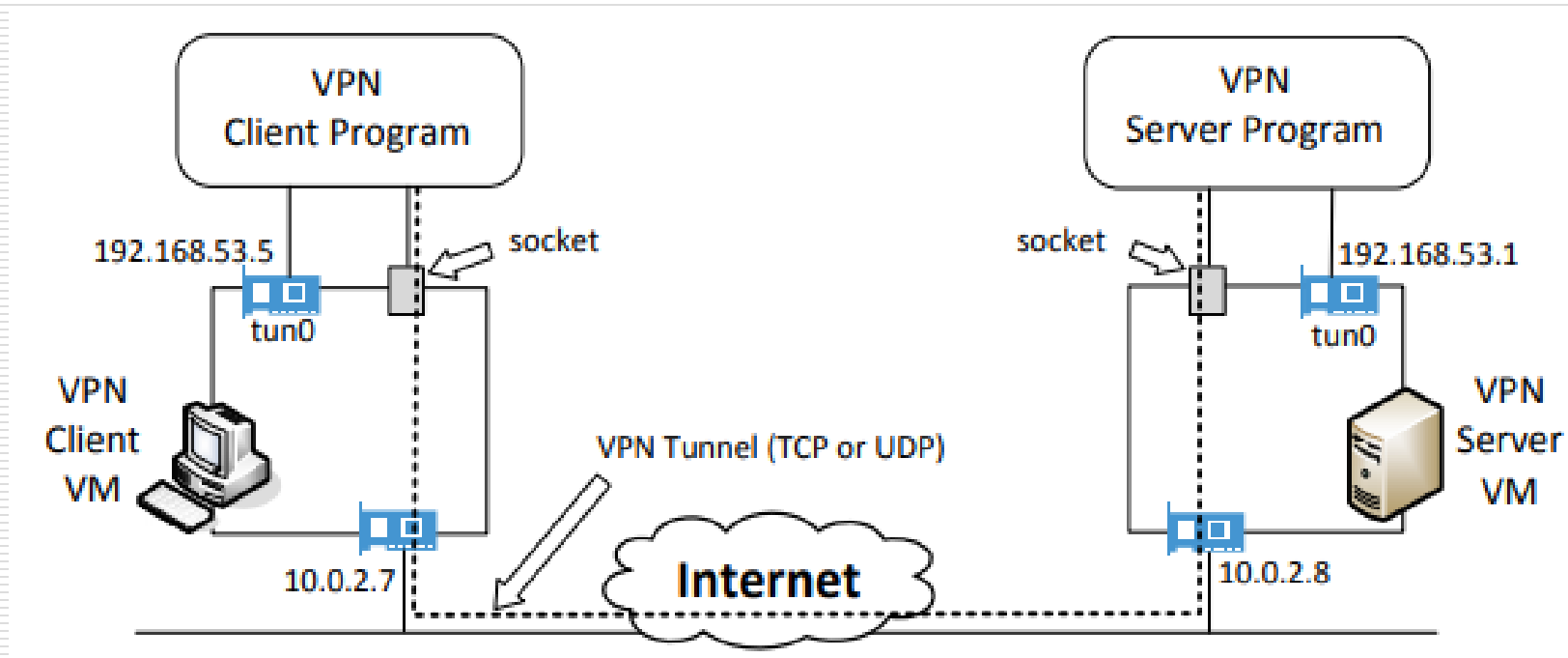
---

### □ TUN/TAP接口的使用

- 用户空间程序通过设备节点/dev/net/tun或/dev/net/tap访问TUN/TAP虚拟网络接口
- 当程序从TUN/TAP接口**读取**数据时，计算机发送到此接口的IP数据包将被传送给程序；
- 程序向tun/tap接口**写入**数据时，发送到接口的IP数据包将被传送到计算机中
- 程序可以使用标准的read()和write()系统调用来接收或发送数据包到虚拟接口。
- 例程：**vpn.tgz**

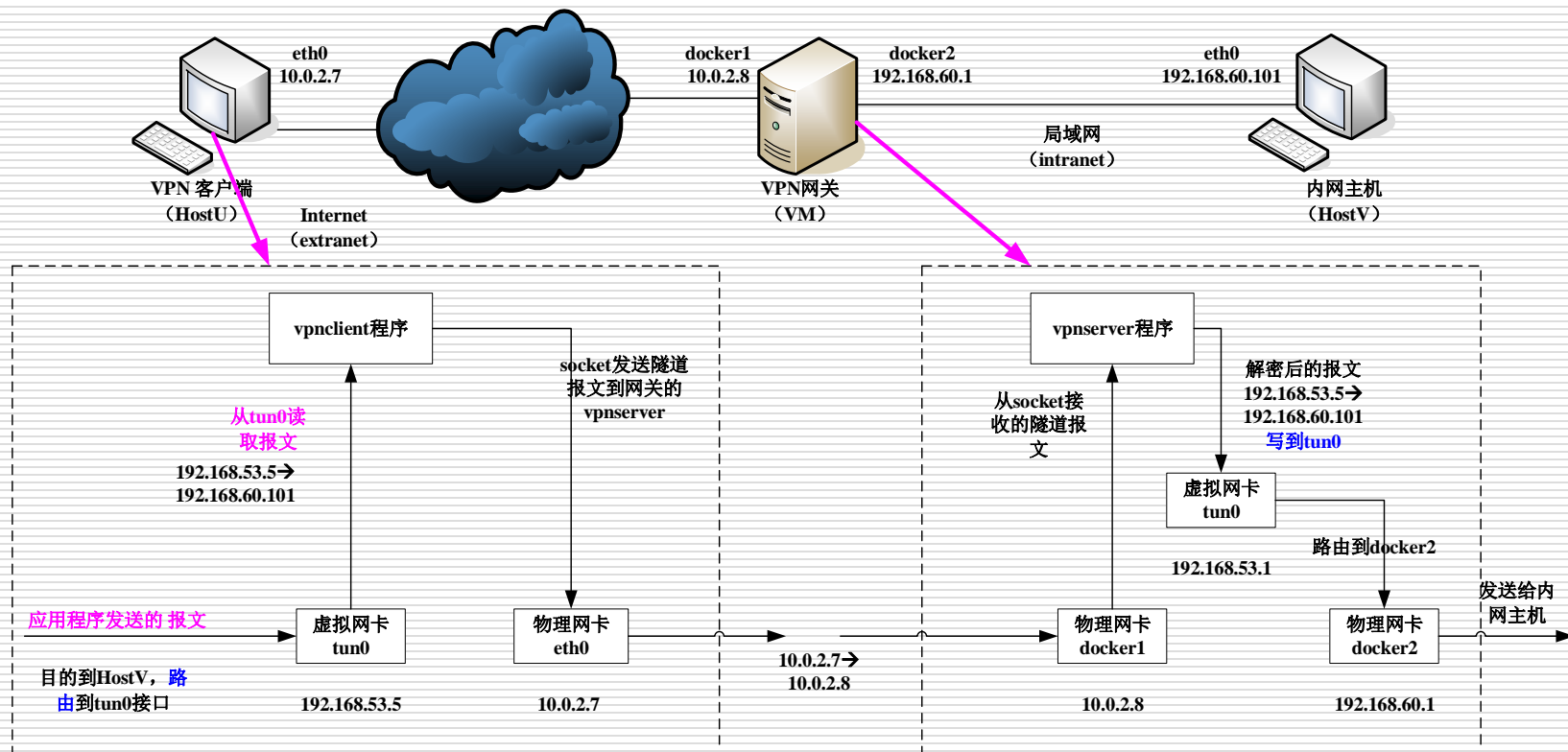
## 2.2 建立VPN隧道

### □ TUN隧道原理



## 2.2 建立VPN隧道

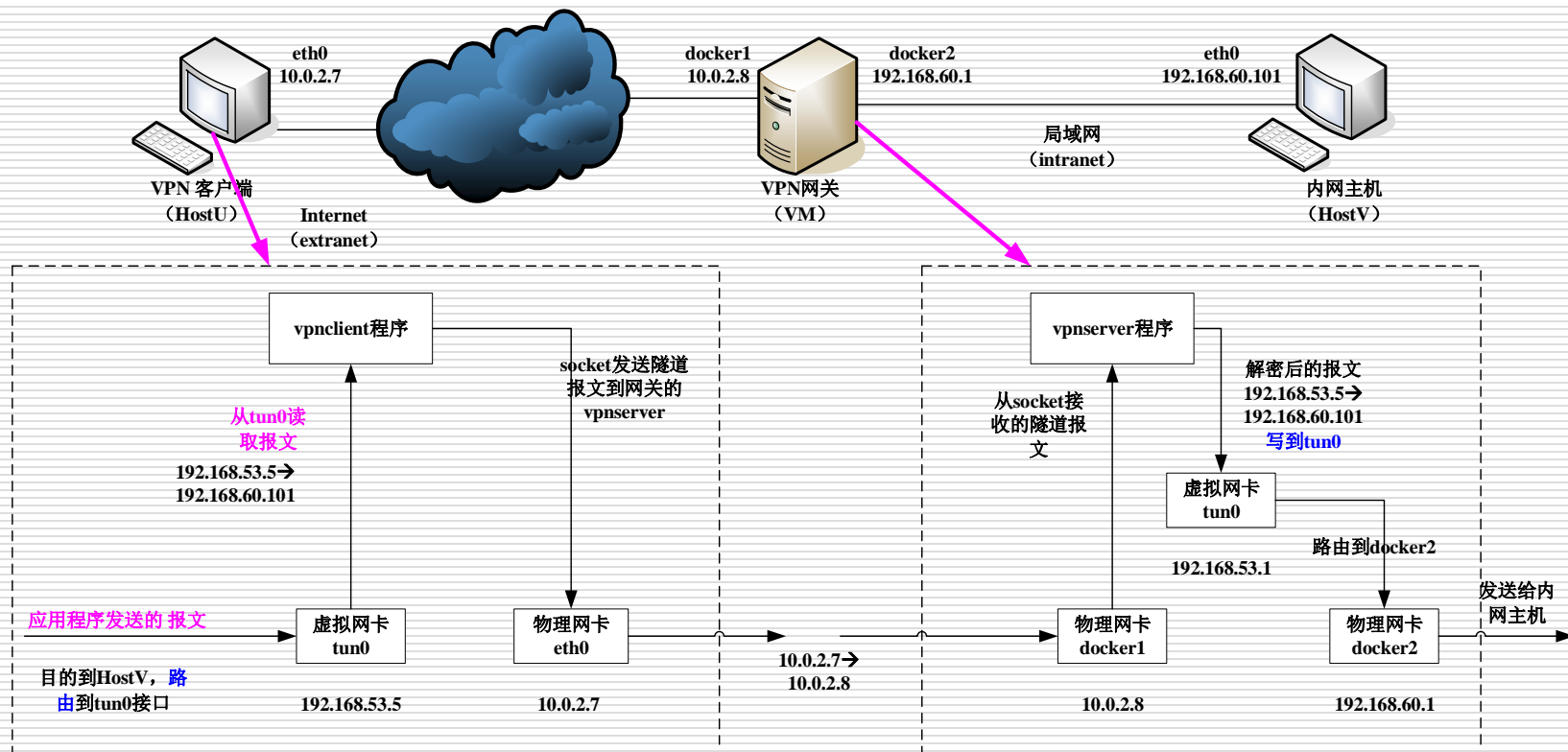
### ■ 客户端主机应用程序发送数据给内网主机



注意：目的到内网主机V的路由走tun0接口的路由需要提前加好

## 2.2 建立VPN隧道

### ■ 客户端主机应用程序发送数据给内网主机



注意：目的到内网主机V的路由走tun0接口的路由需要提前加好

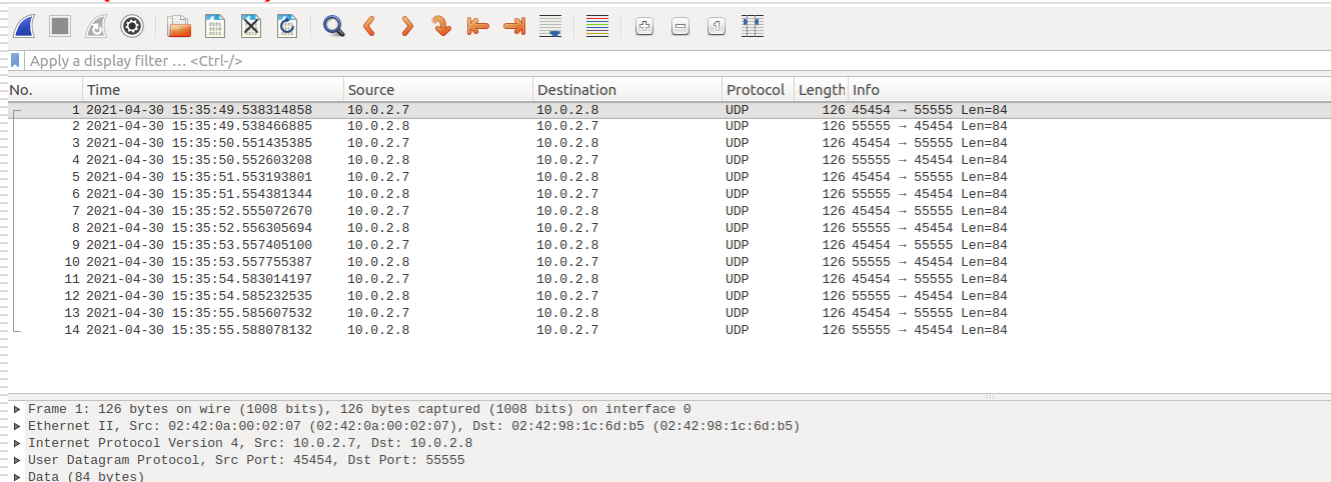
# VPN隧道程序流程

---

- ❑ **VPN**客户端和服务端之间建立套接字通信
  - ❑ 创建**TUN**接口，给**TUN**接口配置ip
  - ❑ 增加路由，将数据包路由到**TUN**接口
  - ❑ 在套接字和**TUN**接口之间转发数据包
  - ❑ 注：**VPN**网关需要
    - 打开路由转发开关  
echo 1 > /proc/sys/net/ipv4/ip\_forward
    - 清空iptables防火墙的规则：  
**sudo iptables -F**
-

# VPN隧道通信截包

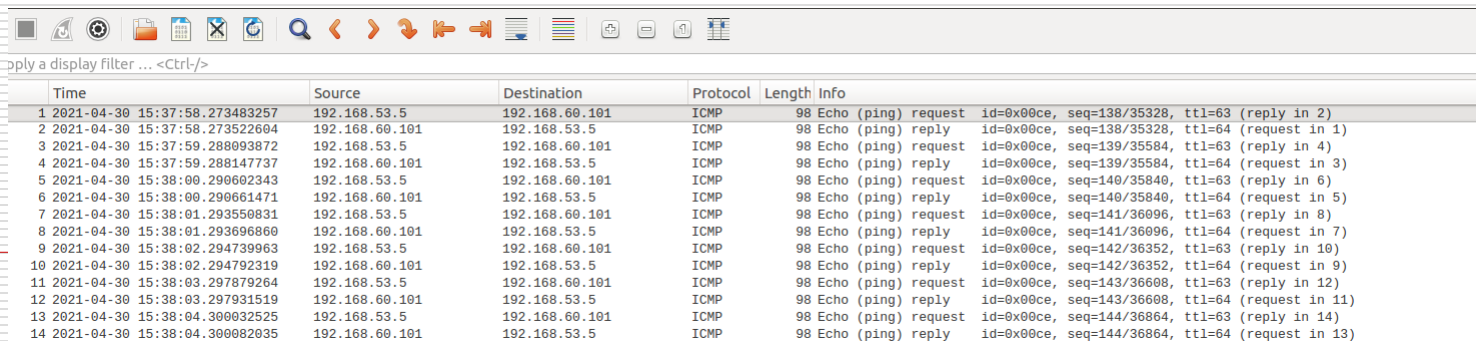
## □ 从HostU(192.168.53.5) ping HostV(192.168.60.101) extranet(docker1)截包：隧道封装报文



No.	Time	Source	Destination	Protocol	Length	Info
1	2021-04-30 15:35:49.538314858	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
2	2021-04-30 15:35:49.538466885	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
3	2021-04-30 15:35:50.551435385	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
4	2021-04-30 15:35:50.552603208	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
5	2021-04-30 15:35:51.553193801	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
6	2021-04-30 15:35:51.554381344	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
7	2021-04-30 15:35:52.555072670	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
8	2021-04-30 15:35:52.556305694	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
9	2021-04-30 15:35:53.557405100	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
10	2021-04-30 15:35:53.557755387	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
11	2021-04-30 15:35:54.583014197	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
12	2021-04-30 15:35:54.585232535	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84
13	2021-04-30 15:35:55.585607532	10.0.2.7	10.0.2.8	UDP	126	45454 → 55555 Len=84
14	2021-04-30 15:35:55.588078132	10.0.2.8	10.0.2.7	UDP	126	55555 → 45454 Len=84

▶ Frame 1: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0  
▶ Ethernet II, Src: 02:42:0a:00:02:07 (02:42:0a:00:02:07), Dst: 02:42:98:1c:6d:b5 (02:42:98:1c:6d:b5)  
▶ Internet Protocol Version 4, Src: 10.0.2.7, Dst: 10.0.2.8  
▶ User Datagram Protocol, Src Port: 45454, Dst Port: 55555  
▶ Data (84 bytes)

## intranet(docker2)截包：原始报文



No.	Time	Source	Destination	Protocol	Length	Info
1	2021-04-30 15:37:58.273483257	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=138/35328, ttl=63 (reply in 2)
2	2021-04-30 15:37:58.273522604	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=138/35328, ttl=64 (request in 1)
3	2021-04-30 15:37:59.288093872	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=139/35584, ttl=63 (reply in 4)
4	2021-04-30 15:37:59.288147737	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=139/35584, ttl=64 (request in 3)
5	2021-04-30 15:38:00.290602343	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=140/35840, ttl=63 (reply in 6)
6	2021-04-30 15:38:00.290661471	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=140/35840, ttl=64 (request in 5)
7	2021-04-30 15:38:01.293550831	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=141/36096, ttl=63 (reply in 8)
8	2021-04-30 15:38:01.293696860	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=141/36096, ttl=64 (request in 7)
9	2021-04-30 15:38:02.294739963	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=142/36352, ttl=63 (reply in 10)
10	2021-04-30 15:38:02.294792319	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=142/36352, ttl=64 (request in 9)
11	2021-04-30 15:38:03.297879264	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=143/36608, ttl=63 (reply in 12)
12	2021-04-30 15:38:03.297931519	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=143/36608, ttl=64 (request in 11)
13	2021-04-30 15:38:04.300032525	192.168.53.5	192.168.60.101	ICMP	98	Echo (ping) request id=0x00ce, seq=144/36864, ttl=63 (reply in 14)
14	2021-04-30 15:38:04.300082035	192.168.60.101	192.168.53.5	ICMP	98	Echo (ping) reply id=0x00ce, seq=144/36864, ttl=64 (request in 13)



## 3 实验内容（2）——加密隧道

---

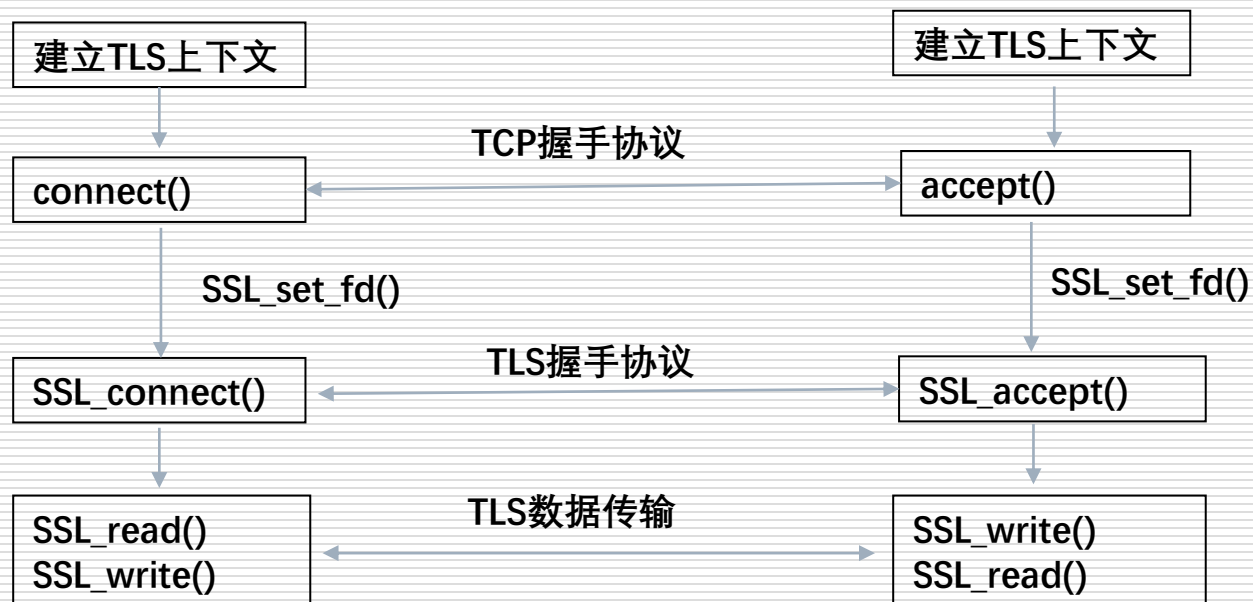
- 加密隧道——**TLS**
  - 隧道协议由**UDP--->TCP**
  - 在客户端和服务端之间的**TCP**连接上建立**TLS/SSL**会话
  - 例程: **tls.zip**
-

# TLS编程的主要步骤

---

## TLS客户端

## TLS服务器端



# TLS/SSL VPN 隧道

1:外网访问SSLVPN服务器,  
发送数据报到

192.168.60.101,80

原始报文:

S:192.168.53.5:2045

D: 192.168.60.101: 80

2:SSLVPN解密、解封装后,  
得到原始报文

②

S:192.168.53.5:2045

D:192.168.60.101:80

1.登录SSLVPN, SSL协商



公网IP: 10.0.2.7

TUN接口IP: 192.168.53.5

10.0.2.8

192.168.60.1

S: 192.168.60.101, 80

D: 192.168.53.5, 2045

③

3:内部服务器返回响应报文

192.168.60.101

192.168.60.102

192.168.60.103

④

S: 10.0.2.8,443

D: 10.0.2.7,1035

4:SSLVPN加密封装

还原报文:

S: 192.168.60.101: 80

D:192.168.53.5:2045

# 3 实验内容（3）——身份认证

---

## □ 认证服务器

- TLS/SSL协议的证书认证（SSL协议中，对服务器的证书认证是必须的）（tls例程中有对服务器的验证）

## □ 认证客户端

- TLS/SSL协议中对客户端的认证不是必须的，也可以选择用客户端证书进行认证
  - 用户名、口令认证方式，用户名直接用服务器端的账号（指导书5.4节）
-

## 3 实验内容（4）——支持多客户端

---

□ 多进程、多线程

---

## 4 实验要求

---

- 按照实验指导手册，使用本实验提供的虚拟机完成实验内容，所有的实验内容最后需要融合在一起。**最后检查的程序只有一个客户端程序、一个服务器端程序。**
  - 通过实验课的上机实验，在线演示给实验指导教师和助教检查，并提交详细的实验报告
-

## 5 报告提交

---

- 所有实验集中在一个实验报告，按照实验报告模板提交，需要包含实验指导手册中提到的证据
  - 注意保存实验过程中的截包数据和屏幕截屏
  - 学期结束前提交
-

# 常见错误

---

1. tcpdump: error while loading shared libraries: libcrypto.so.1.0.0: cannot stat shared object: Permission denied

解决:

```
mv /usr/sbin/tcpdump /usr/bin/tcpdump  
ln -s /usr/bin/tcpdump /usr/sbin/tcpdump
```

2. HostU访问不了HostV:

- 1) 检查路由, HostU是否有到HostV网络的路由, 并且走虚拟接口;
- 2) HostV是否有回到虚拟网卡网络(192.168.53.0/24)的路由或者设置了

默认网关;

- 3) 网关上的转发开关是否打开, `cat /proc/sys/net/ipv4/ip_forward`
  - 4) 网关上的iptables规则是否清空
-