

# 实验三 **VPN**实验

---

# 1 实验目的

---

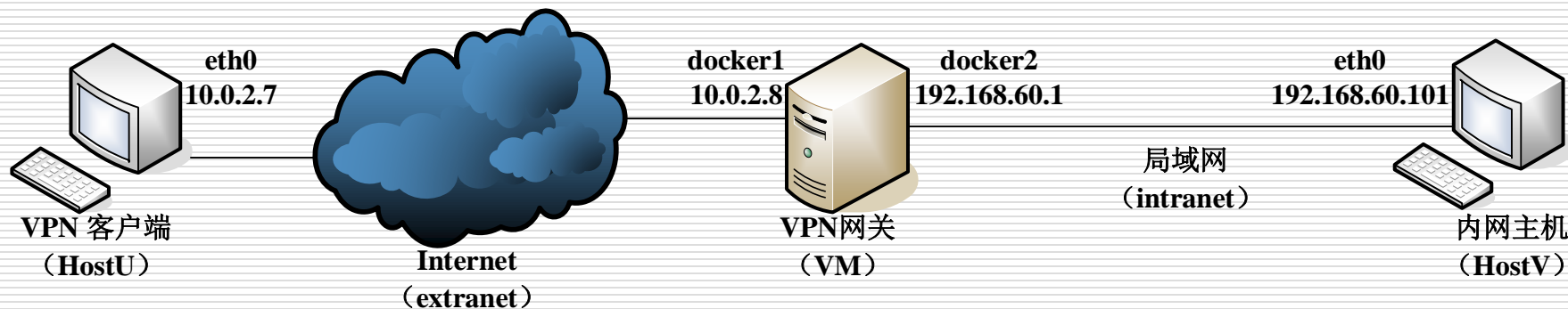
- 掌握**VPN**的网络和安全技术，实现简单的**TLS/SSL VPN**
- **TLS/SSL VPN**的设计和实现体现了许多安全原则，包括以下内容：
  - 虚拟专用网络
  - TUN/TAP和IP隧道
  - 路由
  - 公钥加密，PKI和X.509证书
  - TLS/SSL编程
  - 身份认证

## 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



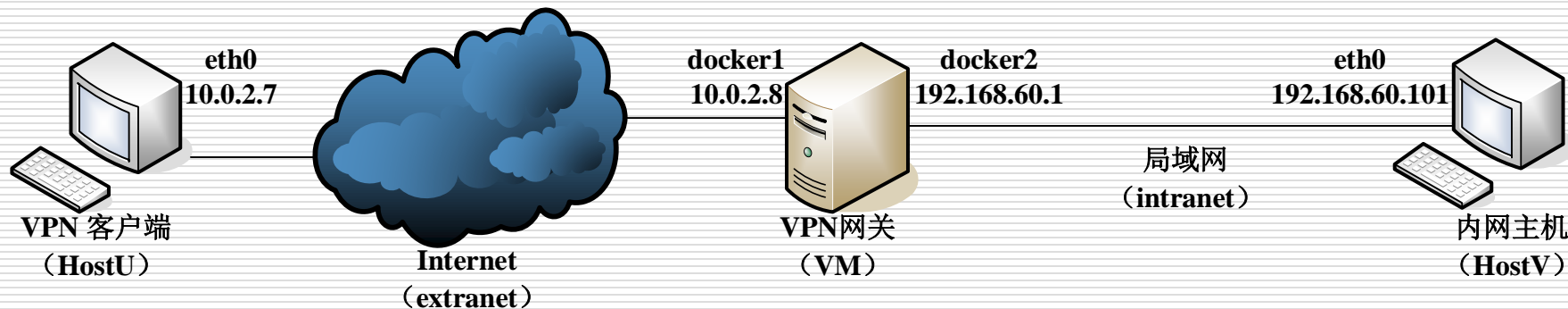
VPN客户端网络配置 (IP、掩码)

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和HostV内分别删除掉默认路由

```
# 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

---

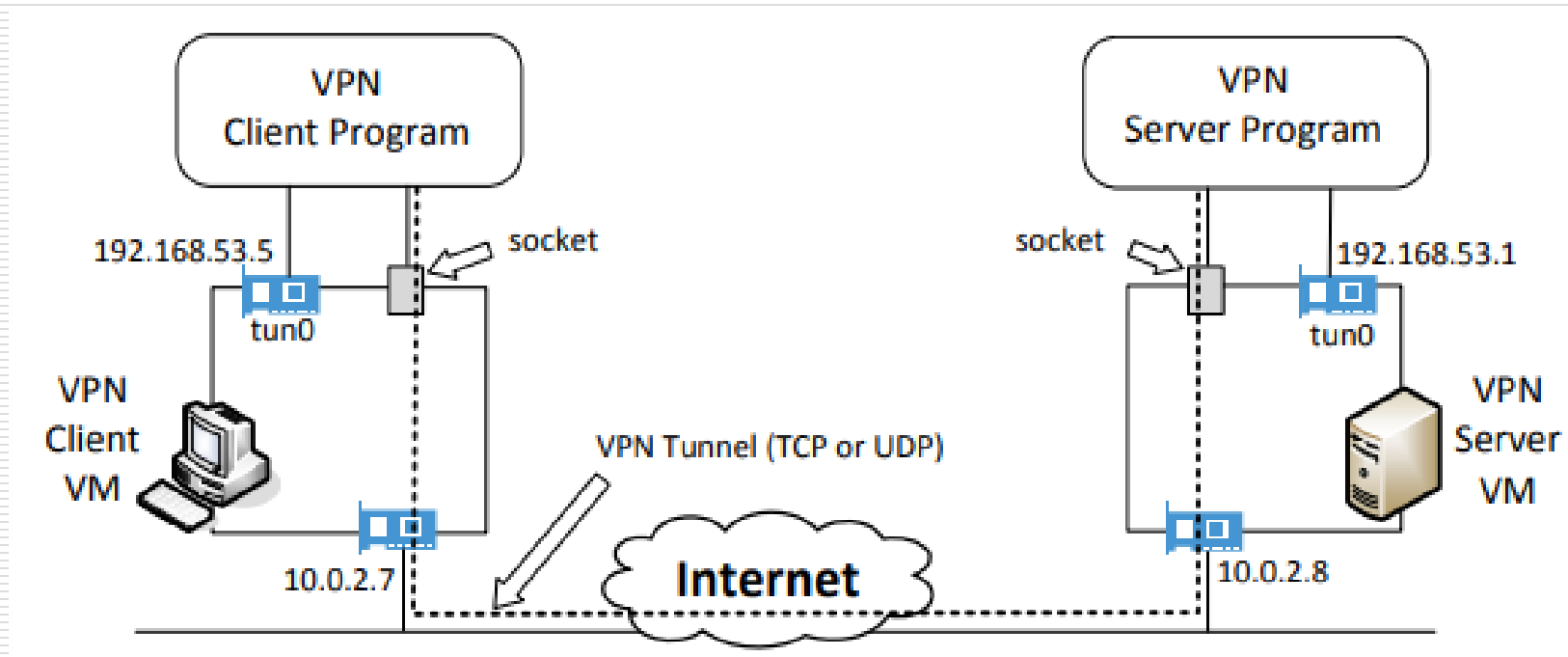
### □ TUN/TAP接口的使用

- 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-2023.zip**



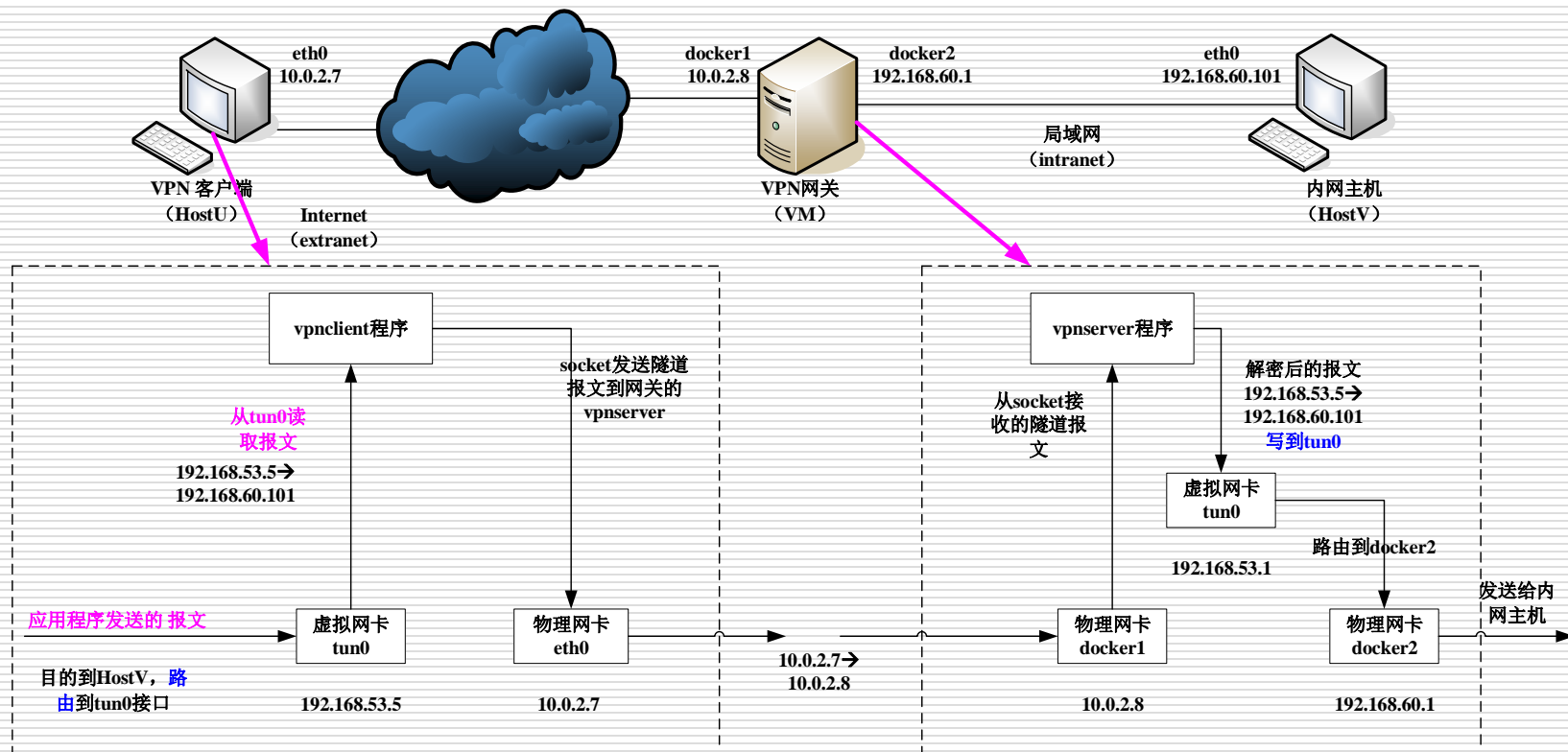
## 2.2 建立VPN隧道

### □ TUN隧道原理



## 2.2 建立VPN隧道

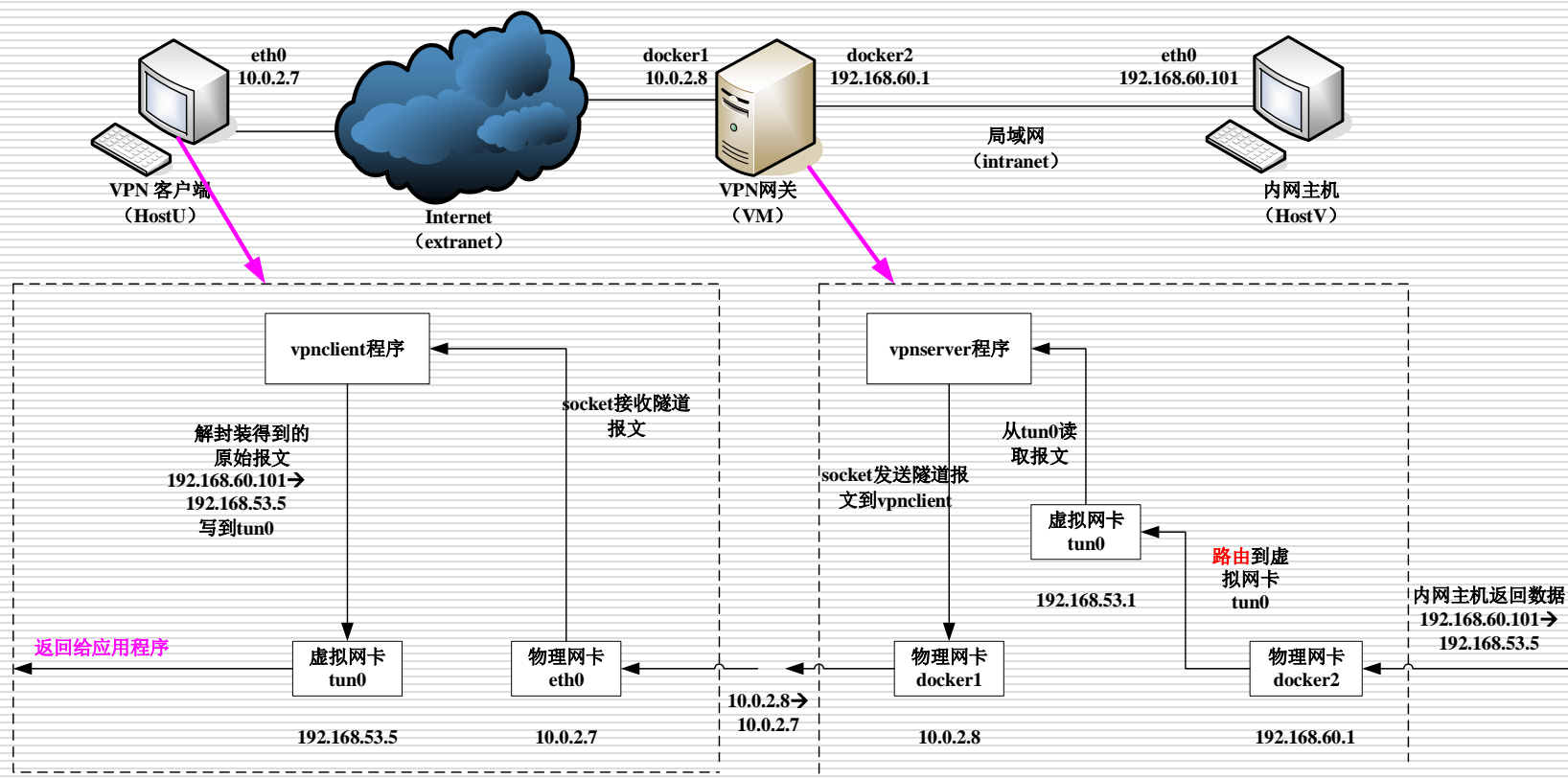
### ■ 客户端主机应用程序发送数据给内网主机



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

### ■ 内网主机返回数据给客户端主机应用程序



注意：目的到192.168.53.0/24的路由需要事先配置好

## 2.3 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: **tls2023.zip**

## 2.4 认证服务器

---

### □ 服务器公钥证书身份认证

- (1) 验证服务器证书是否有效
- (2) 验证服务器是证书的所有者
- (3) 验证服务器是否是目标服务器

### □ 在客户端调用

- `SSL_CTX_load_verify_location`指定CA证书及路径
- `SSL_CTX_set_verify`指定证书验证方式

## 2.5 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5.5节**）

## 2.6 支持多客户端

---

□ 多进程、多线程

### 3 实验要求

---

- 按照实验指导手册，在提供的虚拟机中完成实验内容，所有功能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。
- 检查验收，按照检查单的检查点演示系统，由老师或助教检查评分。
- 按照给定的模板要求，撰写提交实验报告。
- 注意保存实验过程中的截包数据和屏幕截屏