



网络与系统安全

第七章 网络与系统防御

第一节 防火墙

黄 玮



- 网络与系统渗透无孔不入
 - 人
 - 应用程序
 - 网络设备
 - 主机/服务器操作系统
 - 物理设备
- 掌握网络与系统渗透方法
 - 知己知彼，百战不殆



- 防火墙在网络与系统防御中的作用和地位
- 防火墙实现的关键技术
- 防火墙实例及应用



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



防火墙概述

- 防火墙的定义
- 防火墙的发展简史
- 设置防火墙的目的和功能
- 防火墙的局限性
- 防火墙技术发展的动态和趋势



防火墙定义

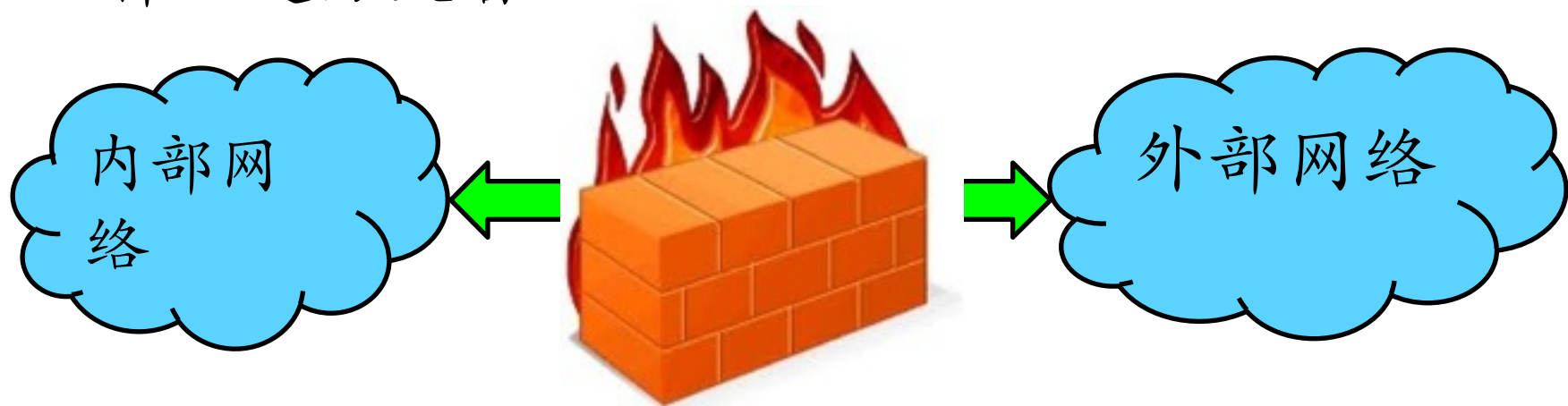
- 什么是防火墙





防火墙定义

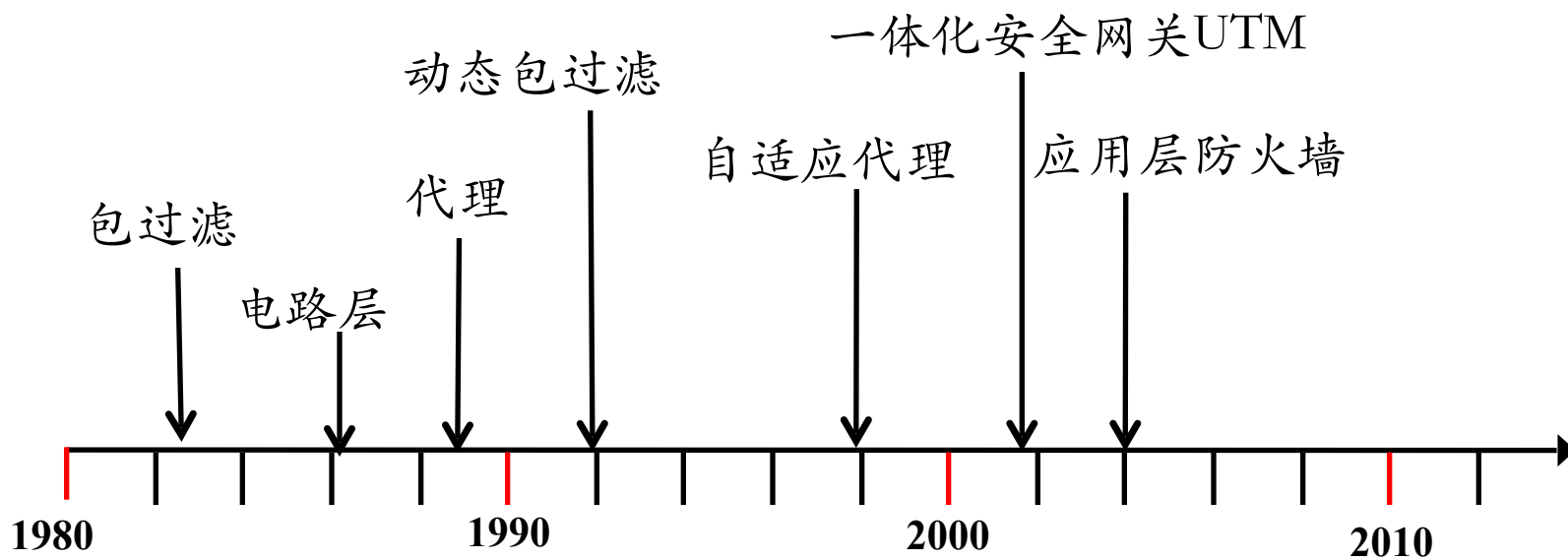
- 什么是防火墙



防火墙：在两个信任程度不同的**网络**之间设置的、用于加强**访问控制**的软硬件保护措施



防火墙发展简史(1/3)





防火墙发展简史(2/3)

- 第一代防火墙
—采用了包过滤技术
- 第二代、第三代防火墙
—1989年，推出了电路层防火墙和应用层防火墙的初步结构
- 第四代防火墙
—1992年，开发出了基于动态包过滤技术的第四代防火墙
- 第五代防火墙
—1998年，NAI公司推出一种自适应代理技术，可以称之为第五代防火墙



防火墙发展简史(3/3)

- 一体化安全网关UTM
 - 统一威胁管理
 - 整合防火墙、入侵检测、入侵保护、防病毒、防垃圾邮件等综合功能
- 应用防火墙
 - 又可以称为IPS：入侵保护
 - 病毒防火墙
 - Web防火墙
 - VoIP防火墙
 - 。 。 。



防火墙的目的和功能

- 防火墙能够强化安全策略
- 防火墙能够有效记录因特网上的活动
- 防火墙限制暴露用户点
- 防火墙是一个安全策略检查站



防火墙的局限性

- 防外不防内
- 管理和配置复杂度高
 - 配置不当易导致安全漏洞
- 很难为用户在防火墙内外提供一致的安全策略
- 粗粒度的访问控制
 - 应用层防火墙和UTM产生的需求驱动力



已有的防火墙产品

- 开源产品

- Endian
- ModSecurity
- SmoothWall
- pfSense
- iptables
- m0n0wall

- IPCop

- 商业产品

- Juniper
- 华为
- 思科
- 联想网御神州

- 绿盟
- Safe3

- 按吞吐能力

- 百兆 / 千兆 / 万兆

- 按并发处理能力

- 少于5000
- 5000~十万
- 十万~五十万
- 五十万以上

- 按防护类型

- 传统防火墙
- 应用层防火墙
- 防DDoS
- 垃圾信息过滤



防火墙技术发展动态和趋势

- 更强的性能
- 可扩展的结构和功能
 - 缓存加速 / 统一认证接入 / 防DDoS / 路由器 …
- 简化的安装和管理
- 积极适应持续变化的网络安全环境
 - 防病毒与防黑客
 - 反垃圾信息
 - 垃圾邮件 / 垃圾短信 / 垃圾电话等



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



防火墙关键技术

- 网络防火墙
 - 包过滤技术
 - 状态检测技术
 - 代理服务技术
- 应用防火墙
 - 见下一节《入侵检测》的技术原理



包过滤技术(1/3)

应用层

TCP层

IP层

链路层



包过滤技术(1/3)

应用层

TCP层

检查IP,TCP,UDP包报头, 与过滤规则比较

IP层

链路层

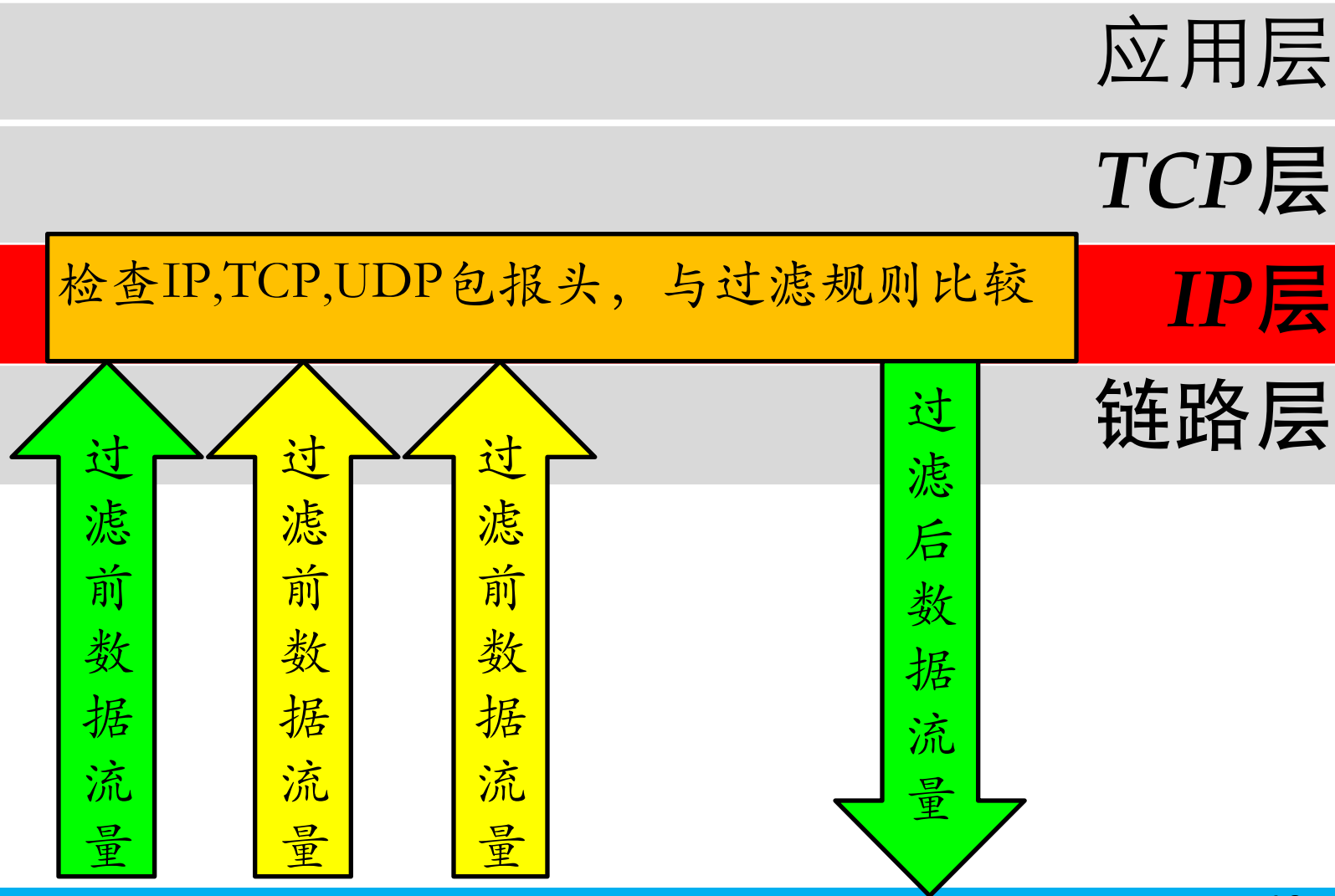
过滤前数据流量

过滤前数据流量

过滤前数据流量



包过滤技术(1/3)





包过滤技术(2/3)

- 包过滤技术检查的数据包报头信息
 - IP数据报的源IP地址、目的IP地址、协议类型，选项字段等
 - TCP数据包的源端口、目标端口、标志段等
 - UDP数据包的源端口、目标端口
 - ICMP类型



包过滤技术(3/3)

- 优点

- 不需要内部网络用户做任何配置，对用户来说是完全透明的
- 过滤速度快，效率高

- 缺点

- 不能进行数据内容级别的访问控制
- 一些应用协议不适合用数据报过滤
- 过滤规则的配置复杂，容易产生冲突和漏洞



状态检测技术(1/3)

应用层

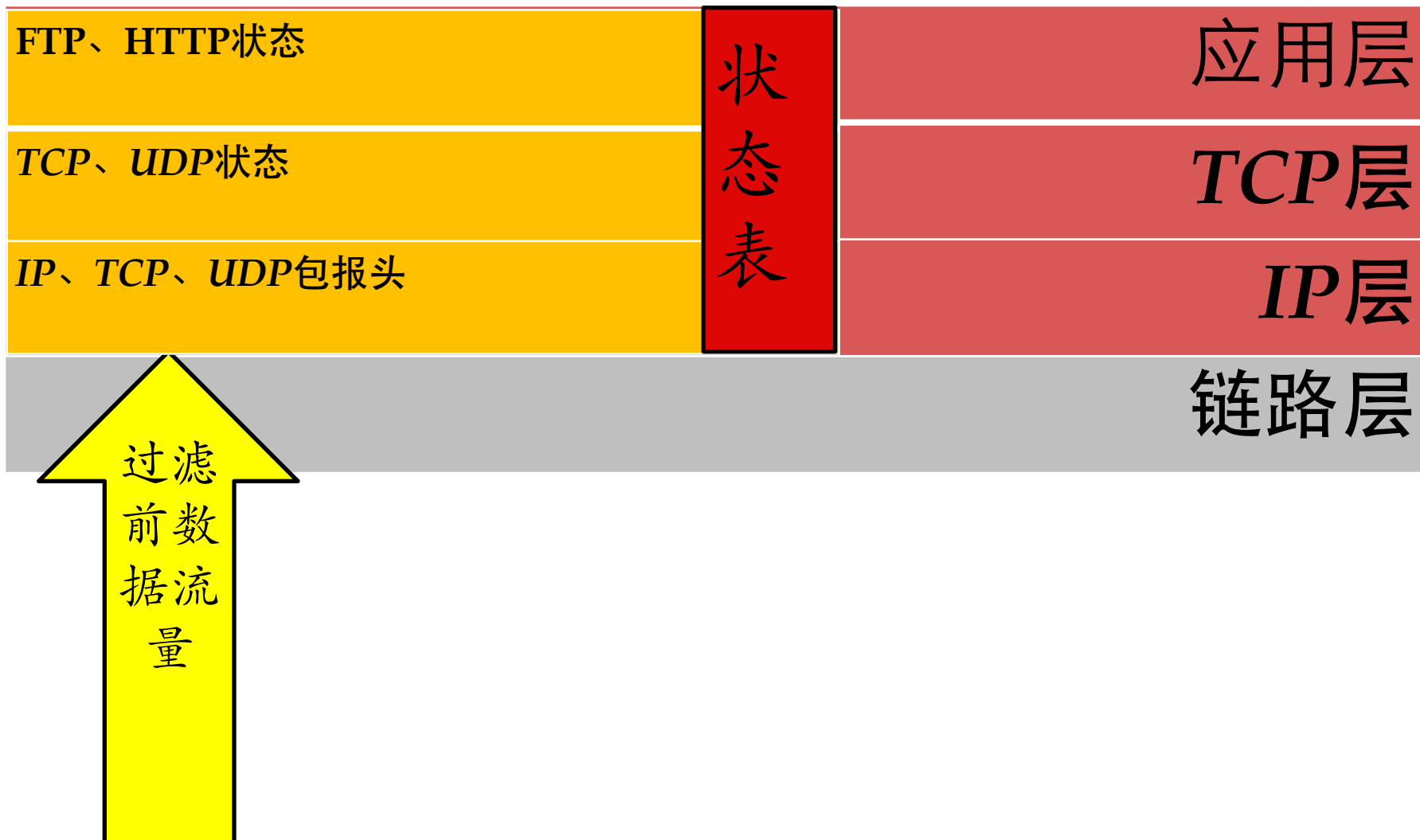
*TCP*层

*IP*层

链路层

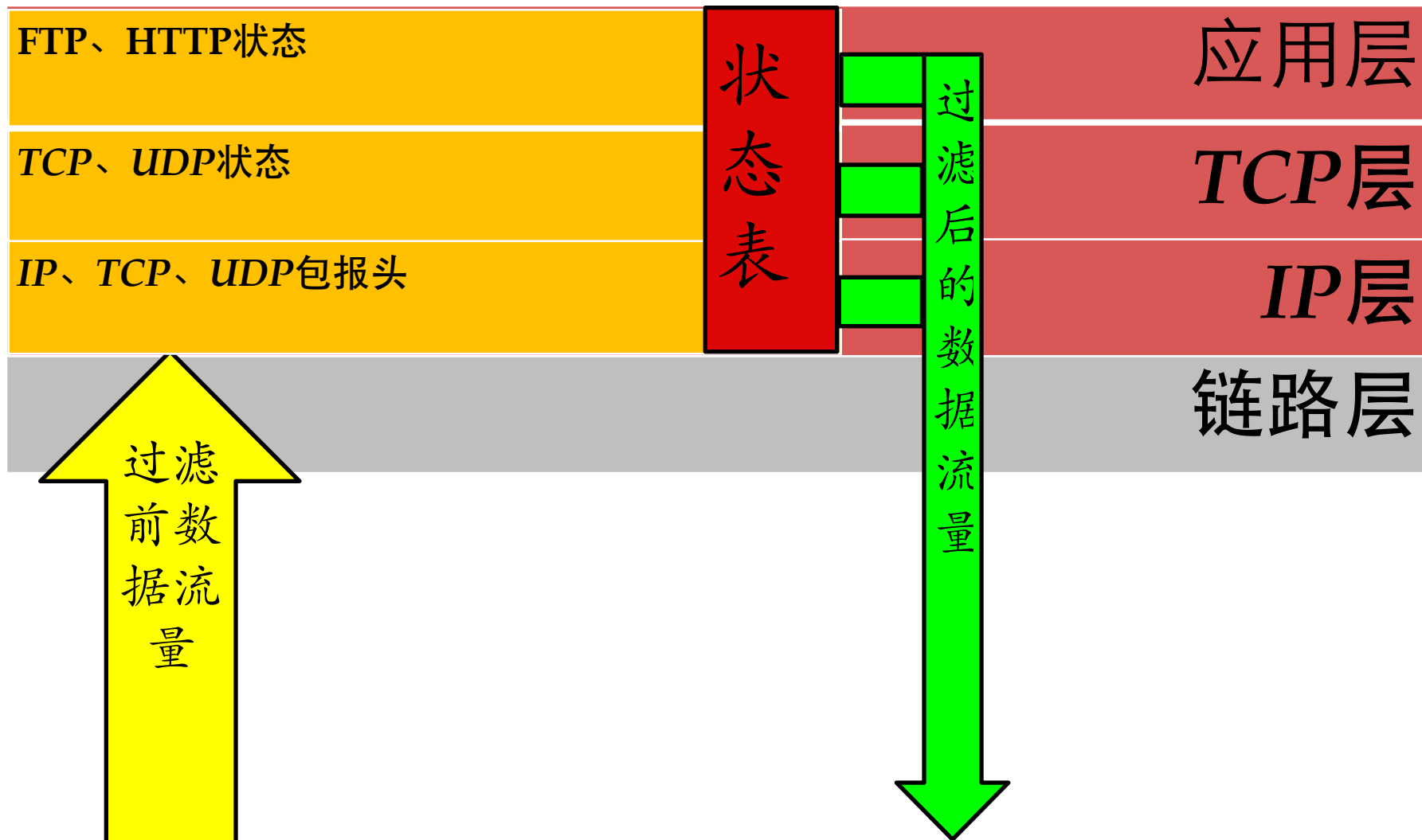


状态检测技术(1/3)



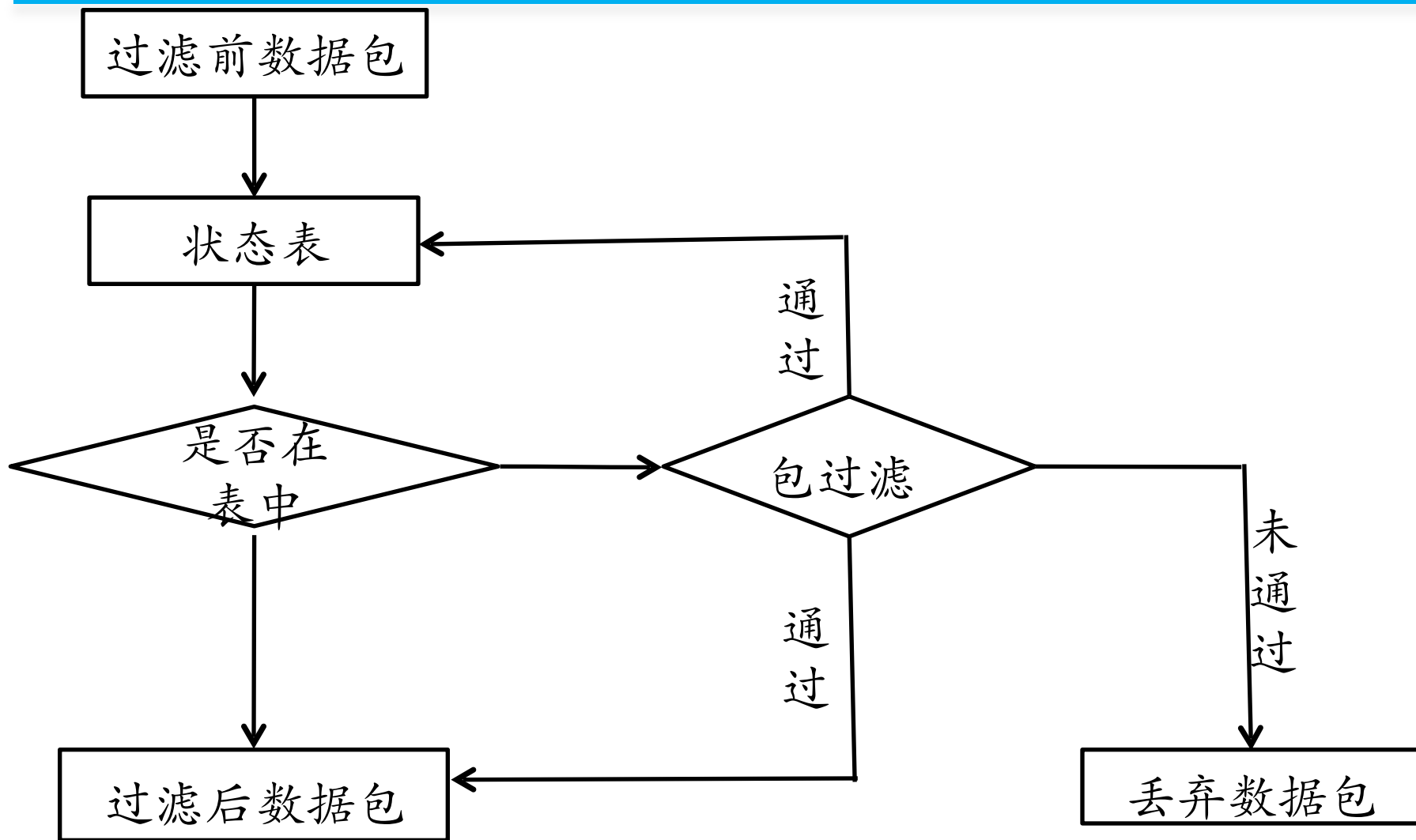


状态检测技术(1/3)





状态检测技术(2/3)





状态检测技术(3/3)

- 优点

- 状态表是动态建立的，可以实现对一些复杂协议建立的临时端口进行有效的管理
- 状态检测技术是为每一个会话连接建立、维护其状态信息，并利用这些状态信息对数据包进行过滤
- 动态状态表是状态检测防火墙的核心，利用其可以实现比包过滤防火墙更强的控制访问能力

- 缺点

- 没有对数据包内容进行检测，不能进行数据内容级别的控制
- 允许外部主机与内部主机直接连接，容易遭受黑客攻击



应用级代理(1/4)





应用级代理(1/4)



外部网络主机



应用级代理(1/4)



外部网络主机



应用代理型防火墙





应用级代理(1/4)



外部网络主机



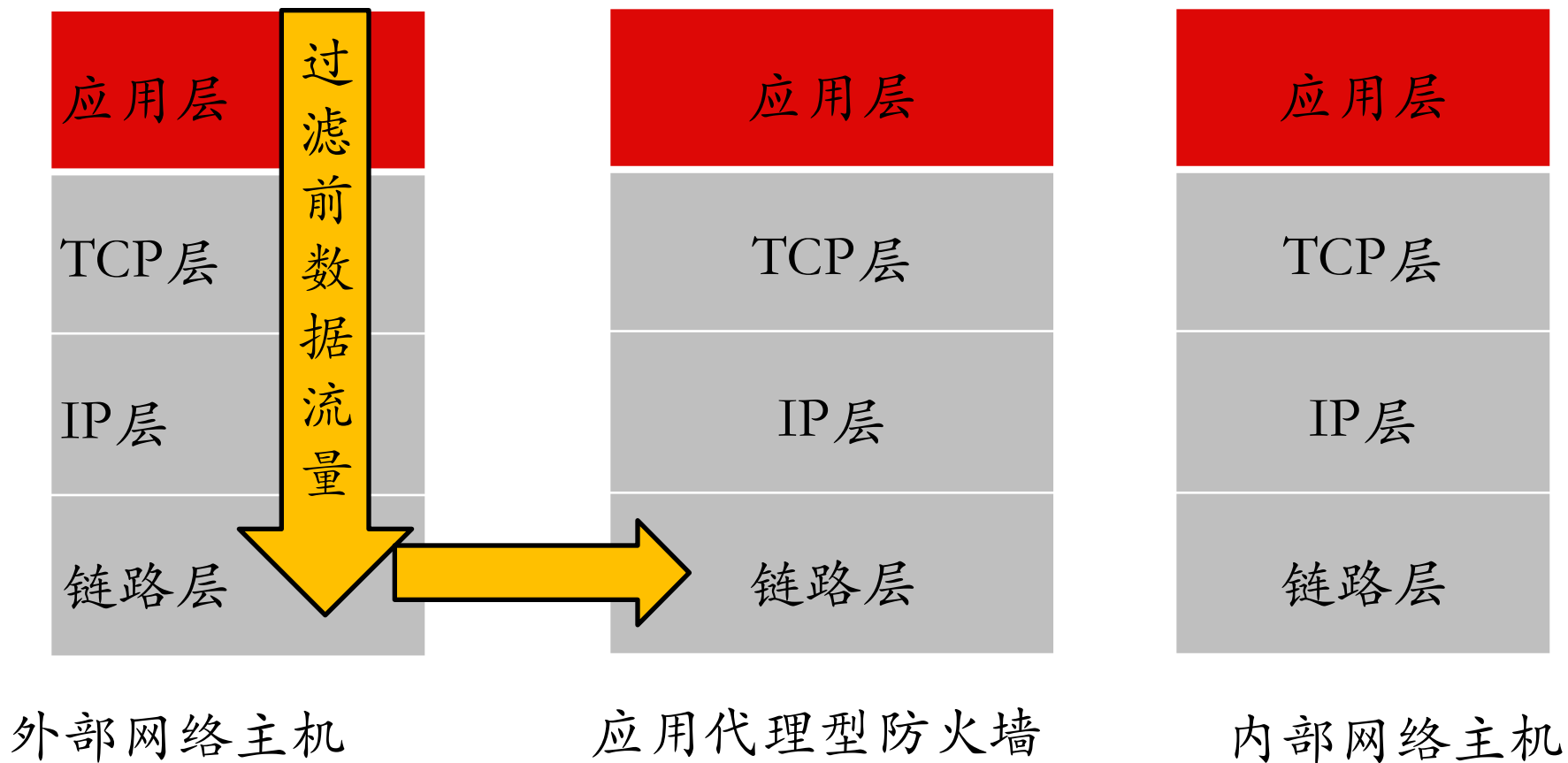
应用代理型防火墙



内部网络主机

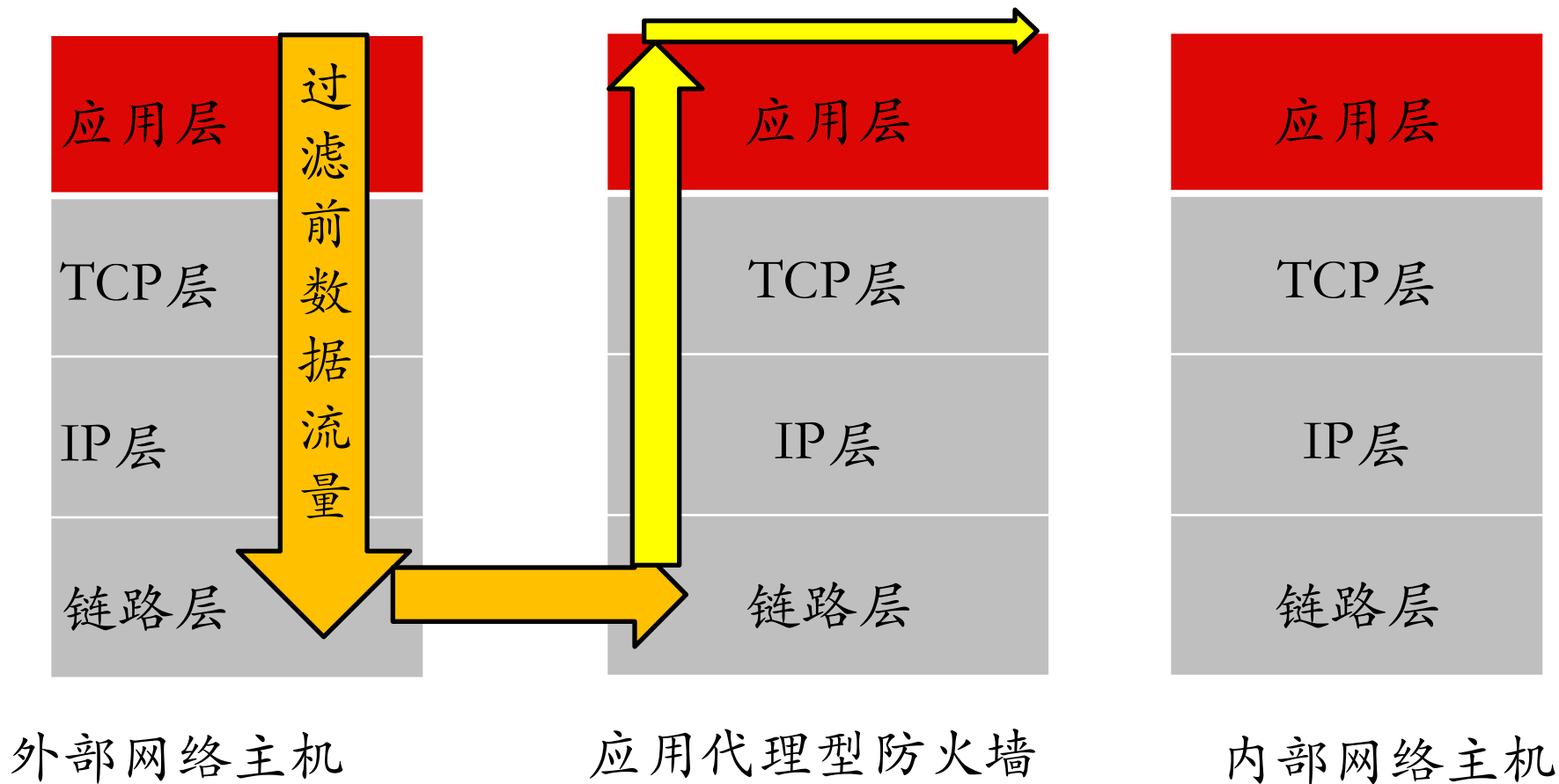


应用级代理(1/4)



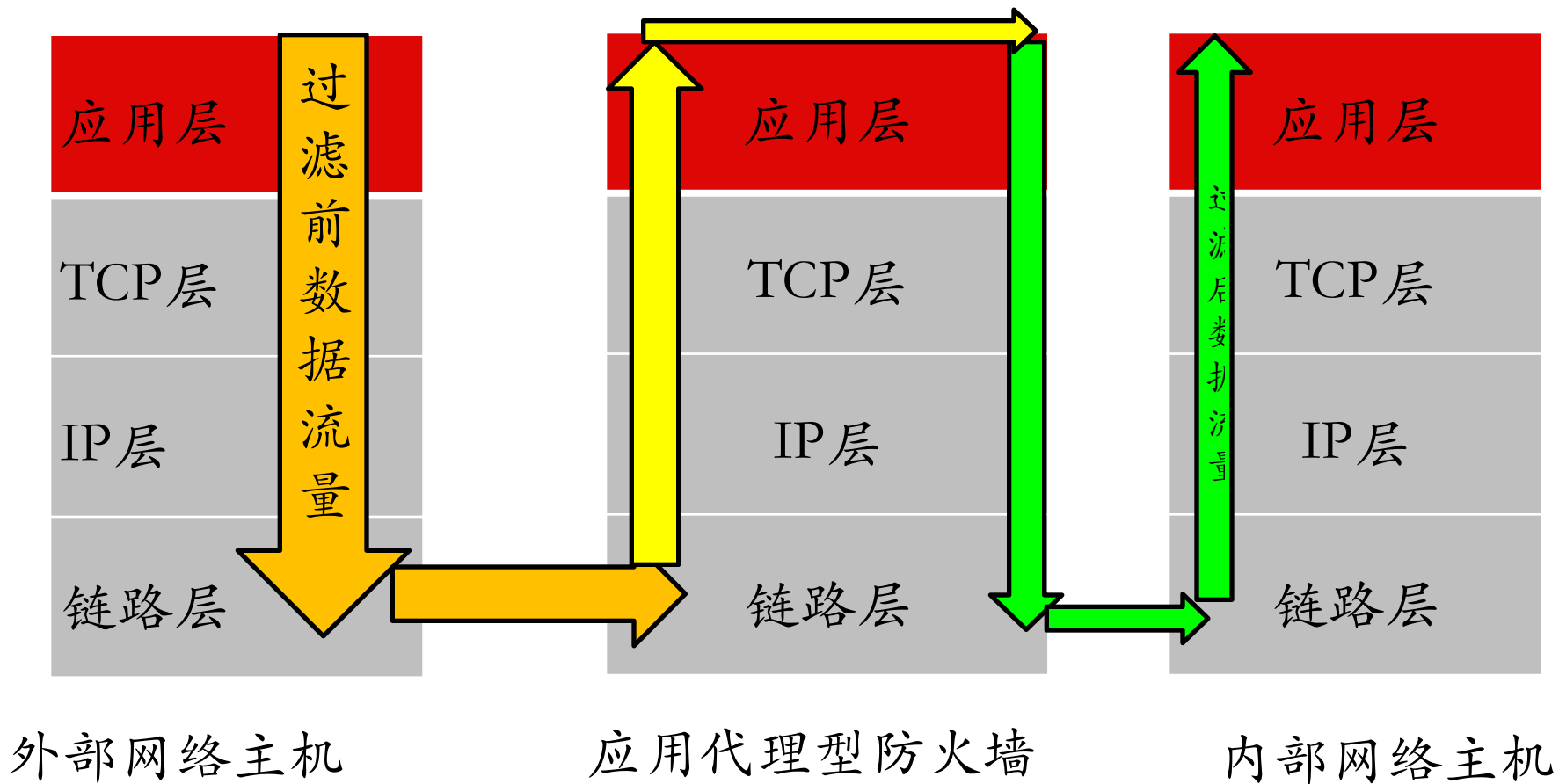


应用级代理(1/4)



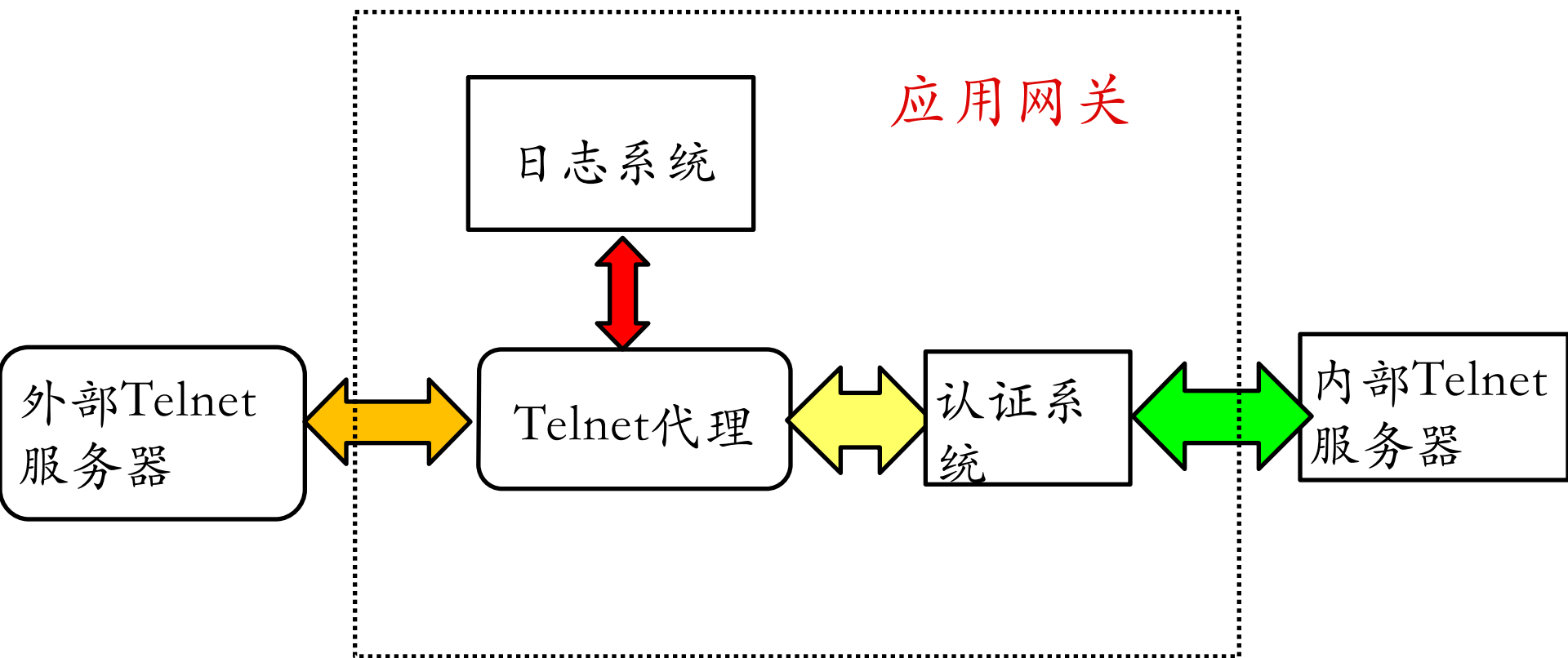


应用级代理(1/4)



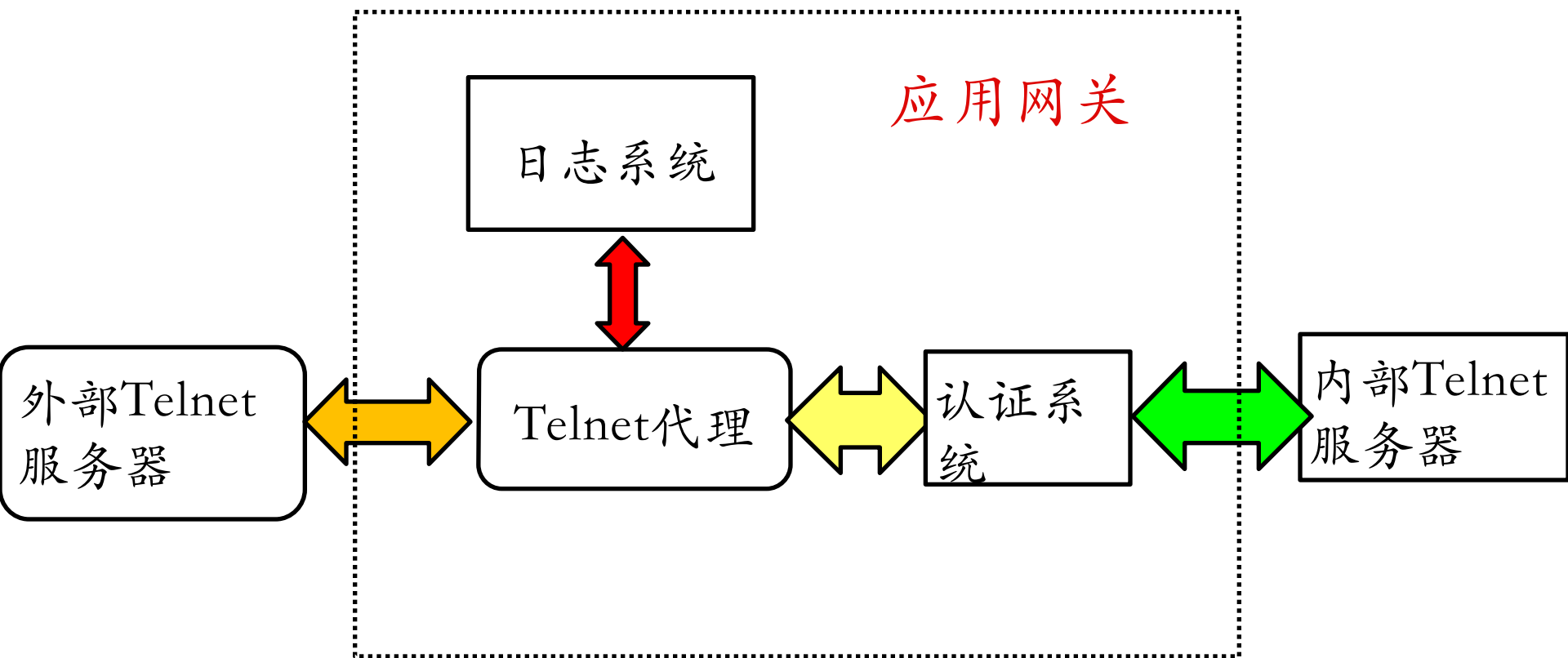


应用级代理(2/4)





应用级代理(2/4)



一个Telnet例子



- 应用代理原理

- 当接收到客户方发出的连接请求后，应用代理检查客户的源和目的IP地址，并依据事先设定的过滤规则决定是否允许该连接请求
- 如果允许该连接请求，进行客户身份识别。否则，则阻断该连接请求
- 通过身份识别后，应用代理建立该连接请求的连接，并根据过滤规则传递和过滤该连接之间的通信数据
- 当一方关闭连接后，应用代理关闭对应的另一方连接，并将这次的连接记录在日志内



应用级代理(4/4)

- 优点

- 内部网络的拓扑、IP地址等被代理防火墙屏蔽，能有效实现内外网络的隔离
- 具有强鉴别和日志能力，支持用户身份识别，实现用户级的安全
- 能进行数据内容的检查，实现基于内容的过滤，对通信进行严密的监控
- 过滤规则比数据包过滤规则简单

- 缺点

- 代理服务的额外处理请求降低了过滤性能，其过滤速度比包过滤器速度慢
- 需要为每一种应用服务编写代理软件模块，提供的服务数目有限
- 对操作系统的依赖程度高，容易因操作系统和应用软件的缺陷而受到攻击



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



- 以Linux操作系统上的Netfilter/iptables机制为例



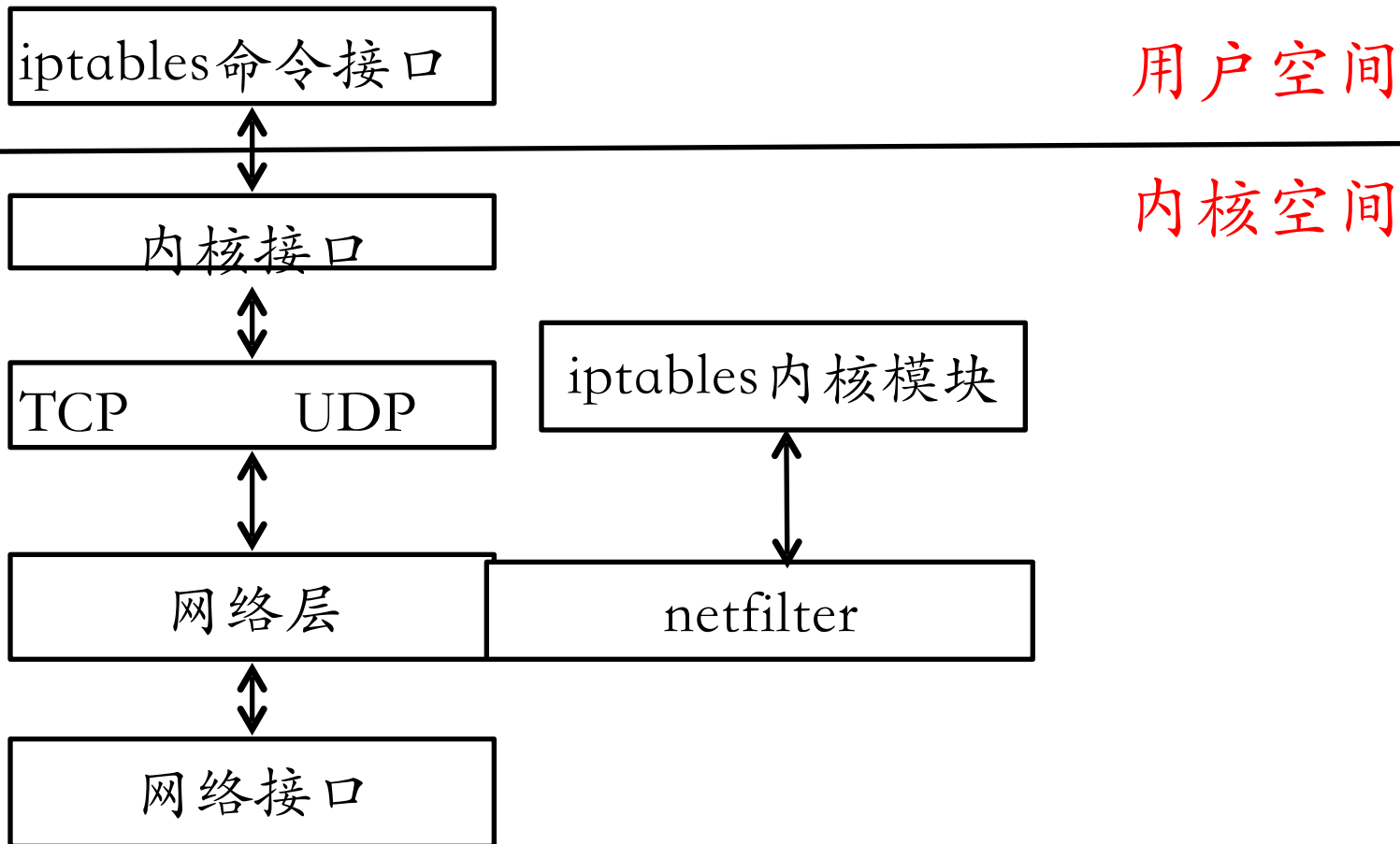
Netfilter/iptables框架简介

- Netfilter/iptables从Linux内核版本2.4开始，默认被包含在内核源代码树中
- 可以对操作系统的流入和流出数据报文进行控制
 - 防火墙
 - NAT
 - 数据报文自定义修改
- Netfilter工作在系统内核层
- iptables工作在用户层



Netfilter架构(1/2)

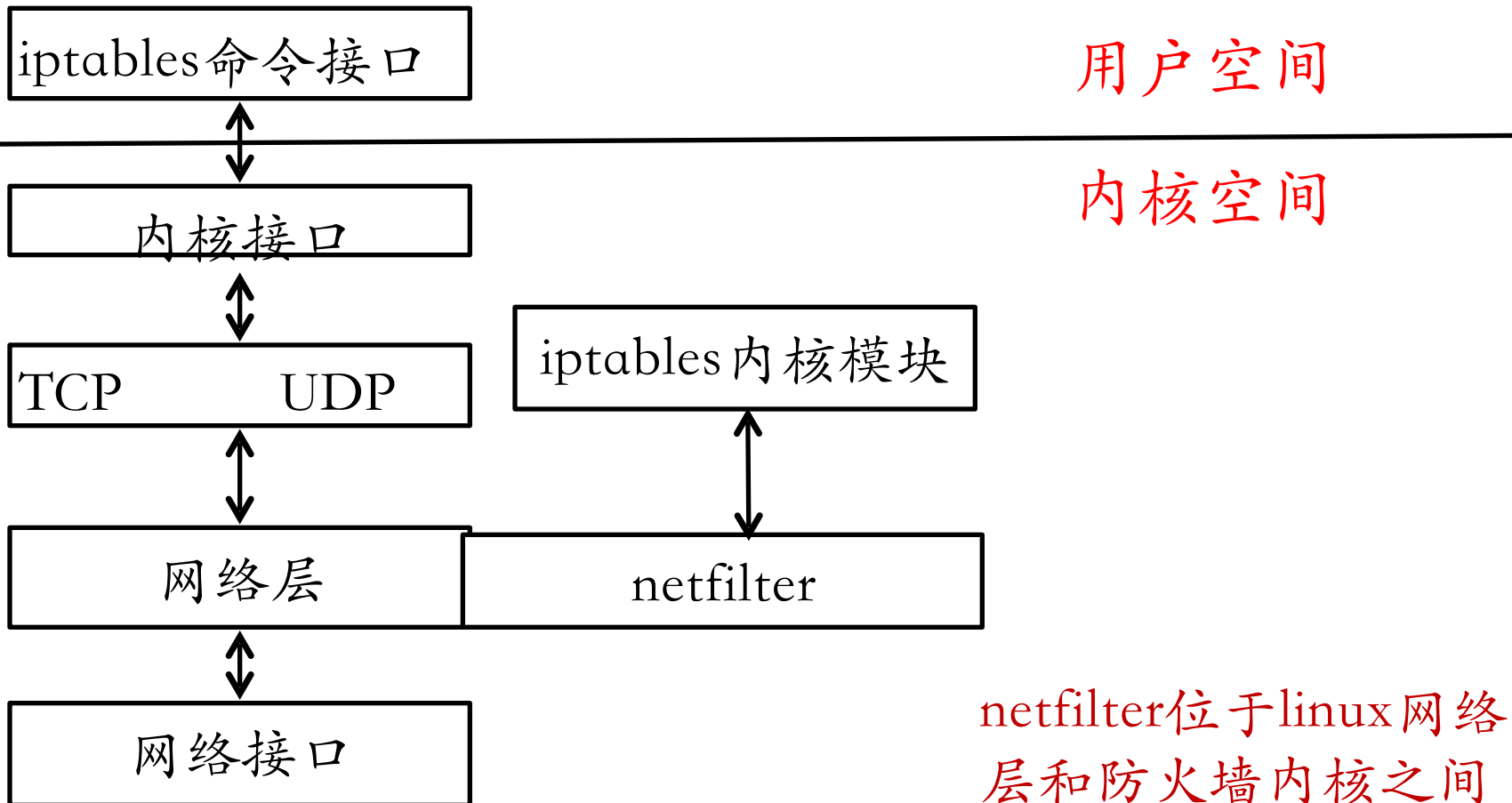
- netfilter是 linux内核中一个强大的网络子系统





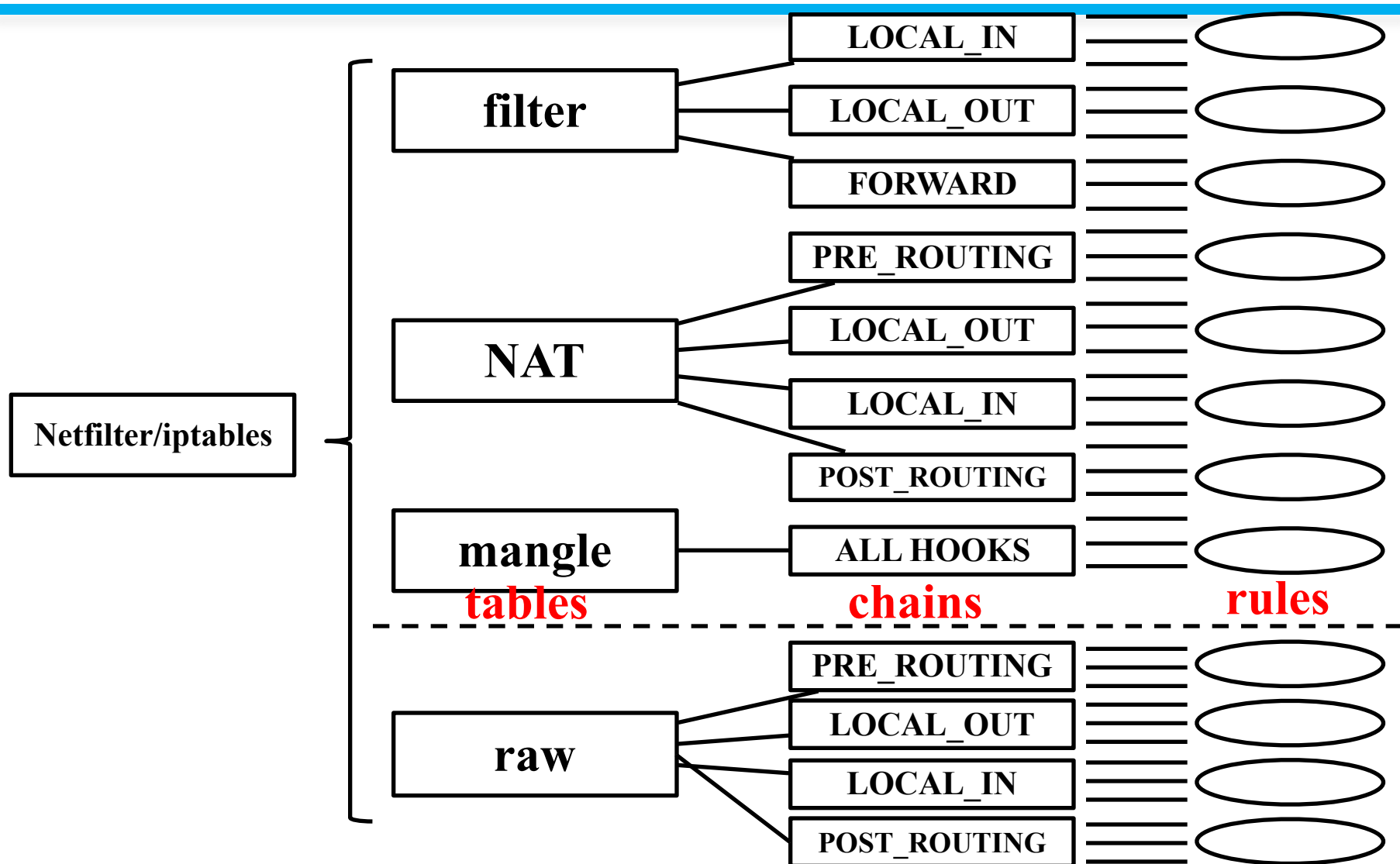
Netfilter架构(1/2)

- netfilter是 linux内核中一个强大的网络子系统





Netfilter架构(2/2)





Netfilter/iptables 基本概念

- 表(tables)
 - filter表、nat表、mangle表、raw表
- 链(chains)
 - 数据包的传输路径，每条链其实就是众多规则中的一个检查清单
 - Input、Forward、PreRouting、PostRouting、Output
- 规则(rules)
 - 网络管理员预定义的网络访问控制策略



iptables 中表的概念(1/2)

- filter 表
 - 报文过滤
 - 只读过滤报文
- nat 表
 - 实现NAT服务
- mangle 表
 - 报文处理
 - 修改报文
 - 附加额外数据到报文



iptables 中表的概念(2/2)

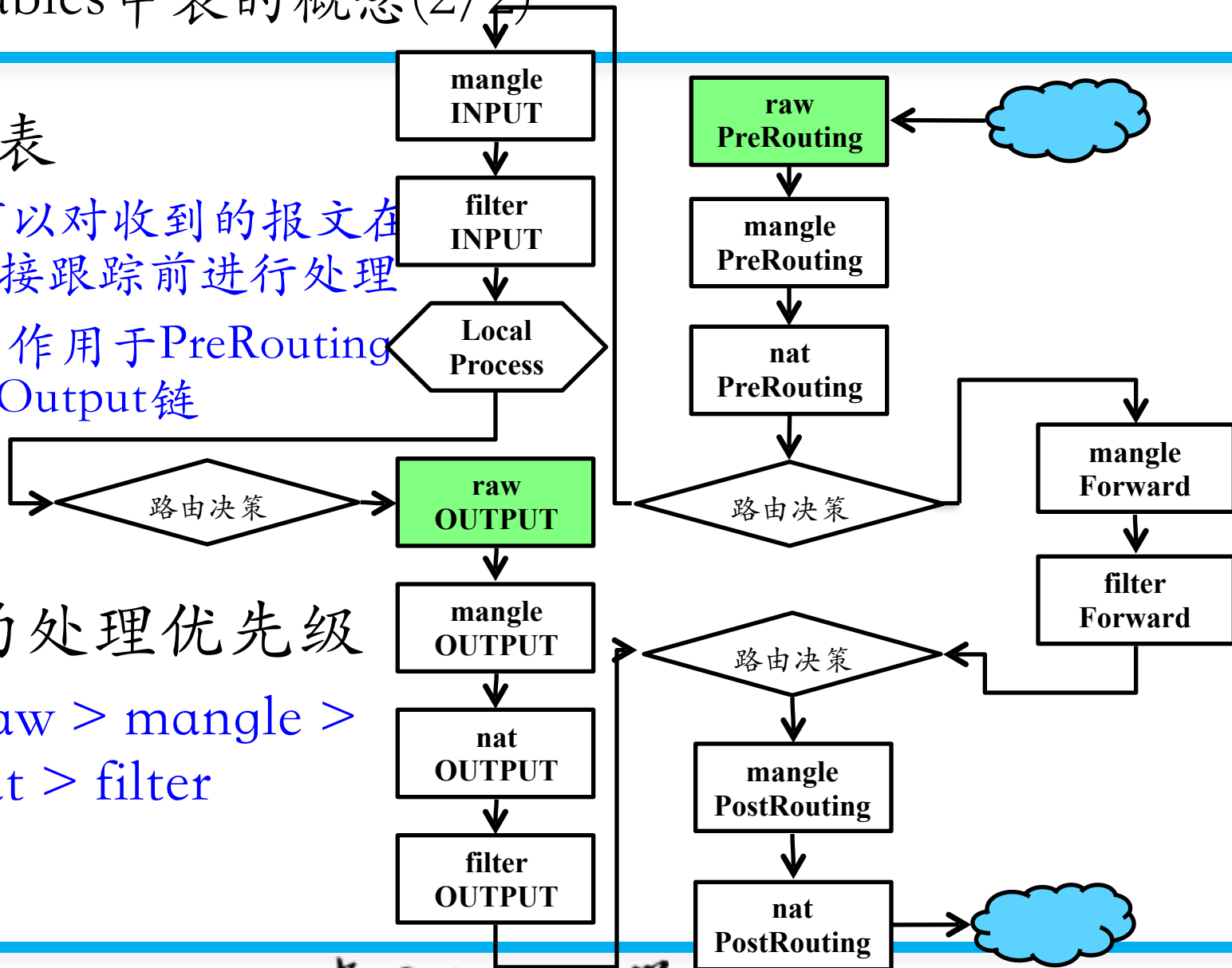
- raw 表

—可以对收到的报文在连接跟踪前进行处理

—只作用于PreRouting和Output链

- 表的处理优先级

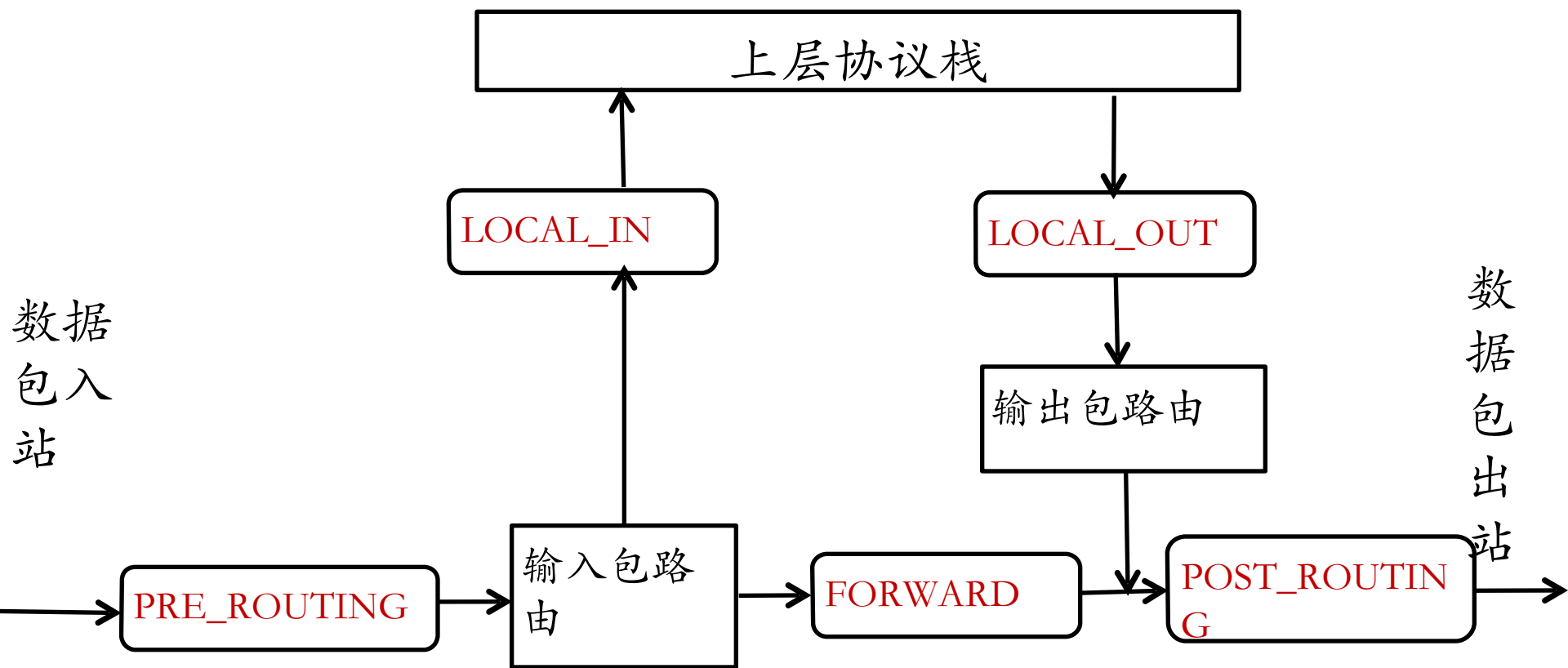
—raw > mangle > nat > filter





Netfilter架构的数据流图(1/2)

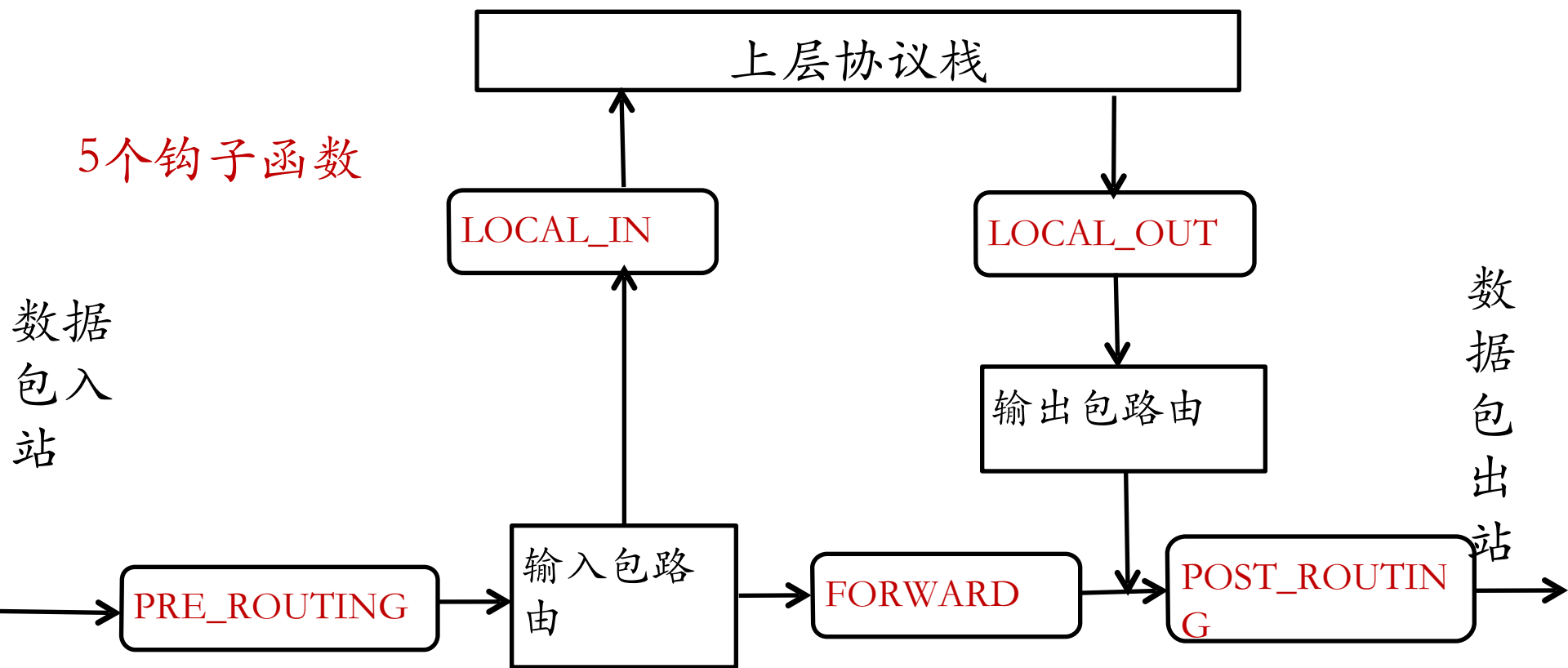
- netfilter模块将防火墙功能引入IP层，实现防火墙代码与IP协议栈代码完全分离





Netfilter架构的数据流图(1/2)

- netfilter模块将防火墙功能引入IP层，实现防火墙代码与IP协议栈代码完全分离





Netfilter架构的数据流图(2/2)

- 对于ipv4协议来说，netfilter在IP数据包处理流程中的5个关键位置定义了5个钩子函数
 - 若数据包是送给本机的，则要经过钩子函数LOCAL_IN处理后传给本机上层协议
 - 若数据包应该被转发，则它将被钩子函数FORWARD处理，然后还要经过钩子函数POST_ROUTING处理后才能传输到网络
 - 本机进程产生的数据包要先经过钩子函数LOCAL_OUT处理后，再进行路由选择处理，然后经过钩子函数POST_ROUTING处理后再发送到网络

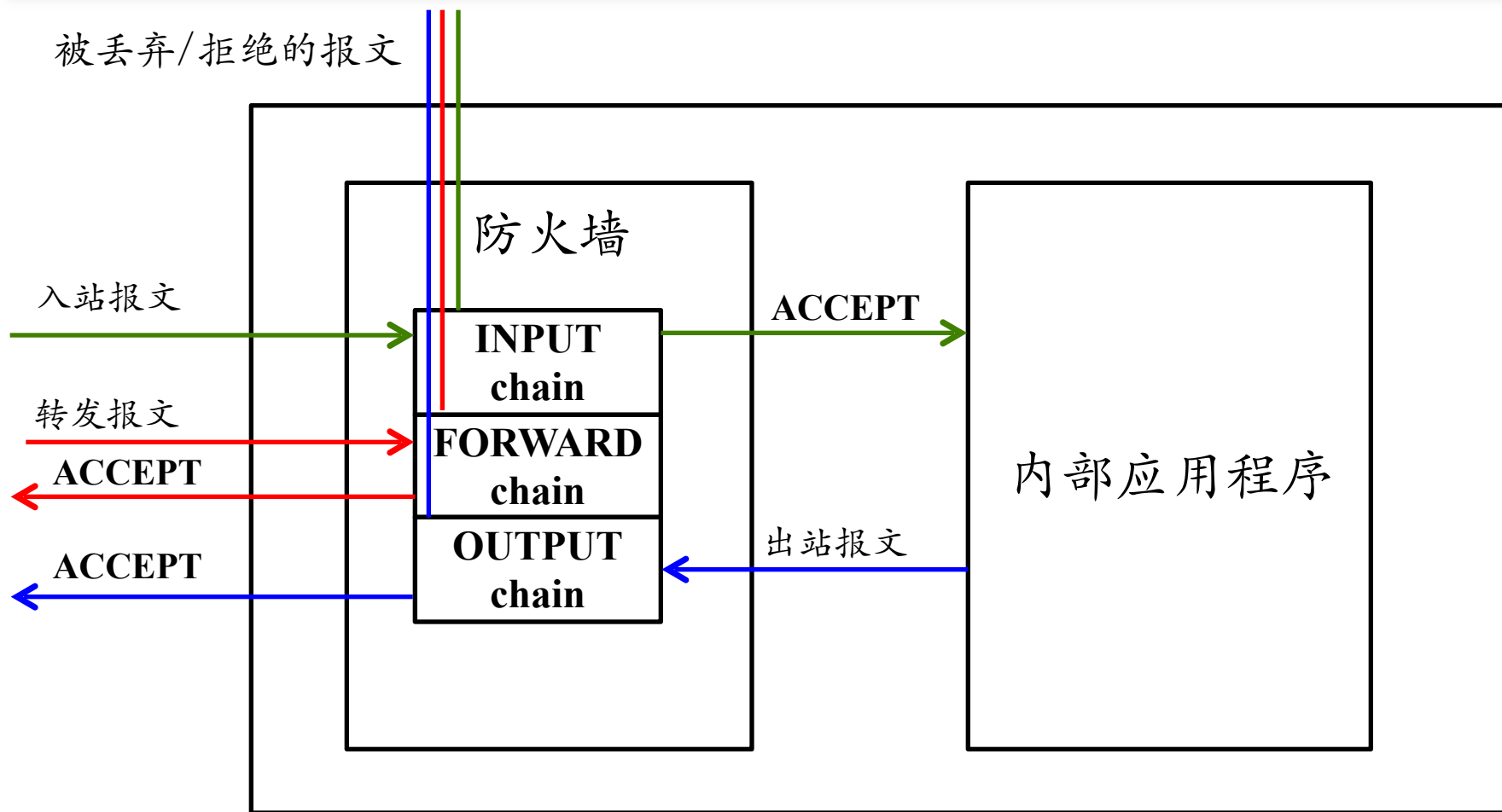


iptables 防火墙内核模块

- 内核的防火墙模块正是通过把自己的函数注册到netfilter的钩子函数这种方式介入了对数据包的处理
- 函数按功能分为4种
 - 连接跟踪
 - 数据包过滤
 - 网络地址转换
 - SNAT
 - DNAT
 - 对数据包进行修改



Netfilter/iptables 防火墙工作原理



部署了Netfilter/iptables的操作系统



- 通用CPU架构
- ASIC架构
- 网络处理器架构



通用CPU架构

- 又被称为x86架构
 - 采用通用CPU和PCI总线接口
- 可编程性高
 - 更灵活
 - 更易扩展
- 产品功能主要由软件实现
- 代表产品
 - 大部分的开源/商业软件防火墙（基于*nix系统）



ASIC架构

- Application Specific Integrated Circuit
 - 专用集成电路
 - 一种带有逻辑处理的加速处理器
- 把一些原先由CPU完成的经常性和重复工作交给ASIC芯片来负责完成
 - 交换机、路由器、智能IC卡
- 通常配合通用CPU单元来完成复杂运算
- 代表产品
 - 大部分国外的商业硬件防火墙



NP架构

- Network Processor
 - 网络处理器
- 通用CPU架构和ASIC架构的折衷
 - 开发难度
 - 性能
 - 灵活性/可扩展性
- 代表产品
 - 大部分国内的商业硬件防火墙



三种硬件架构的横向比较

架构类型	X86	NP	ASIC
灵活性	★★★	★★	★
扩展性	★★★	★★	★
性能	★	★★	★★★
安全性	★	★★	★★★
价格	低	中等	较高



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用



本章内容提要

- 防火墙发展简史
- 防火墙关键技术原理
- 防火墙的实现技术
- 防火墙的配置和应用

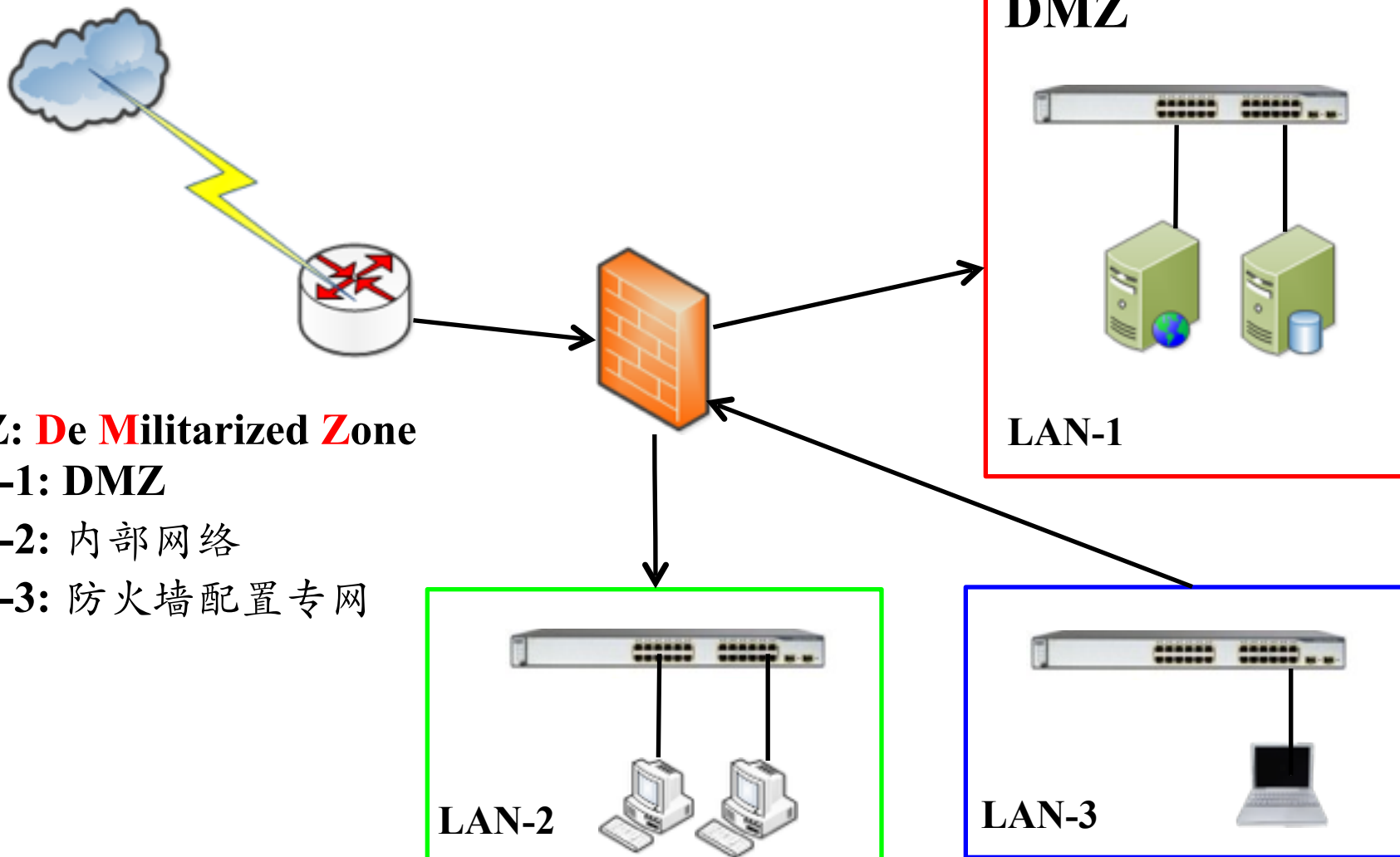


典型网络部署模型

- 路由模式
- 透明模式
- 混合模式



路由模式



DMZ: **D**e **M**ilitarized **Z**one

LAN-1: DMZ

LAN-2: 内部网络

LAN-3: 防火墙配置专网



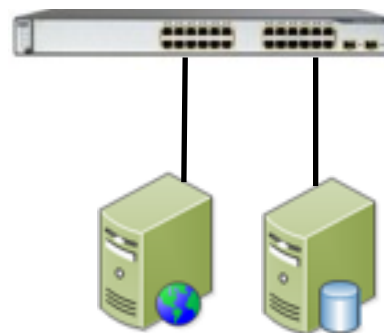
路由模式



子网之间相互访问控制被**隔离**



DMZ



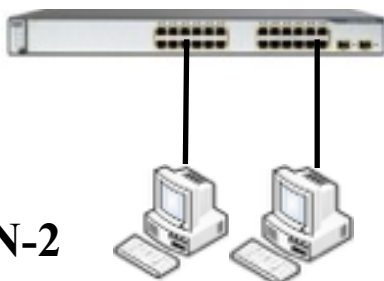
LAN-1

DMZ: De Militarized Zone

LAN-1: DMZ

LAN-2: 内部网络

LAN-3: 防火墙配置专网



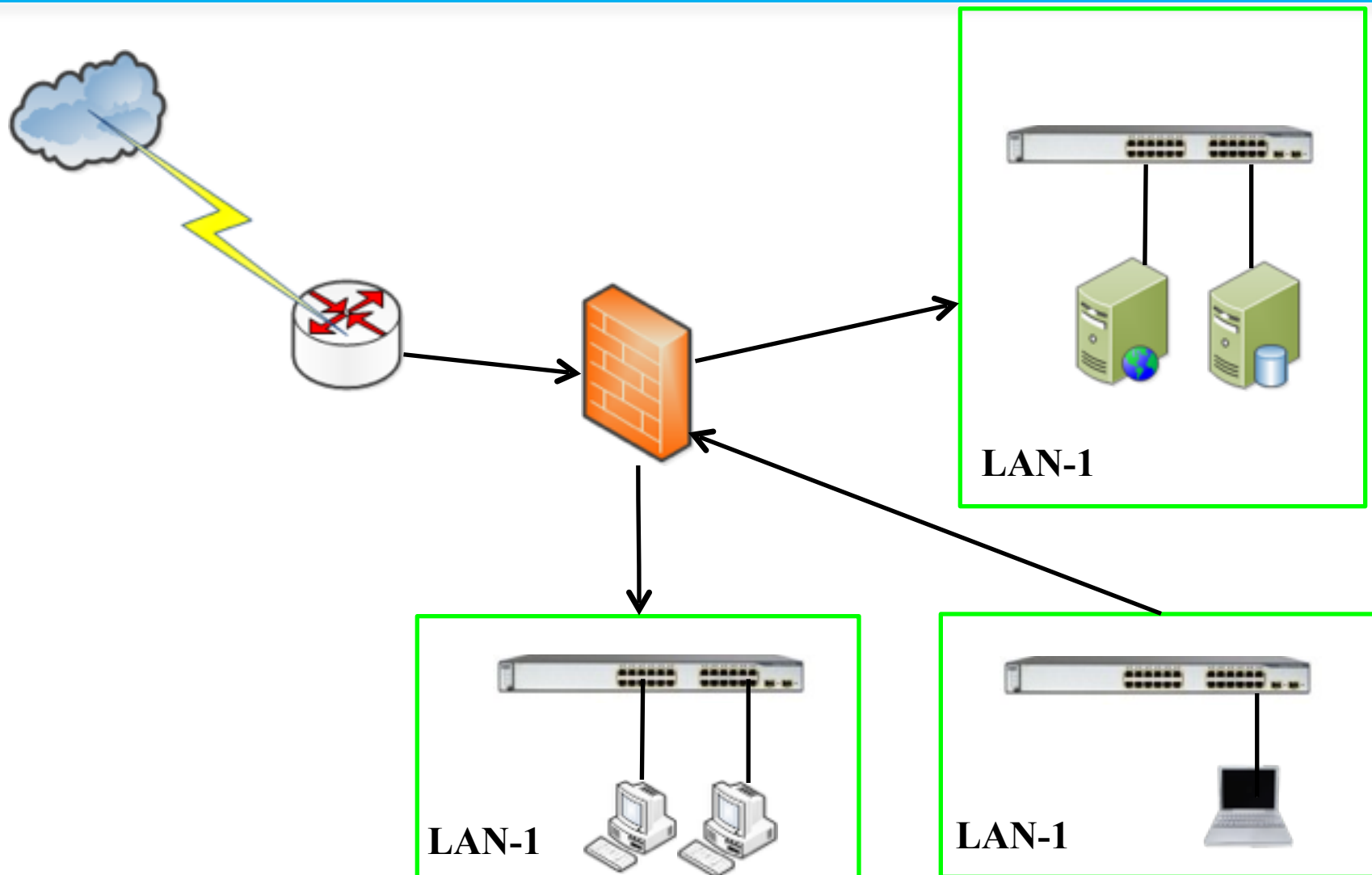
LAN-2



LAN-3

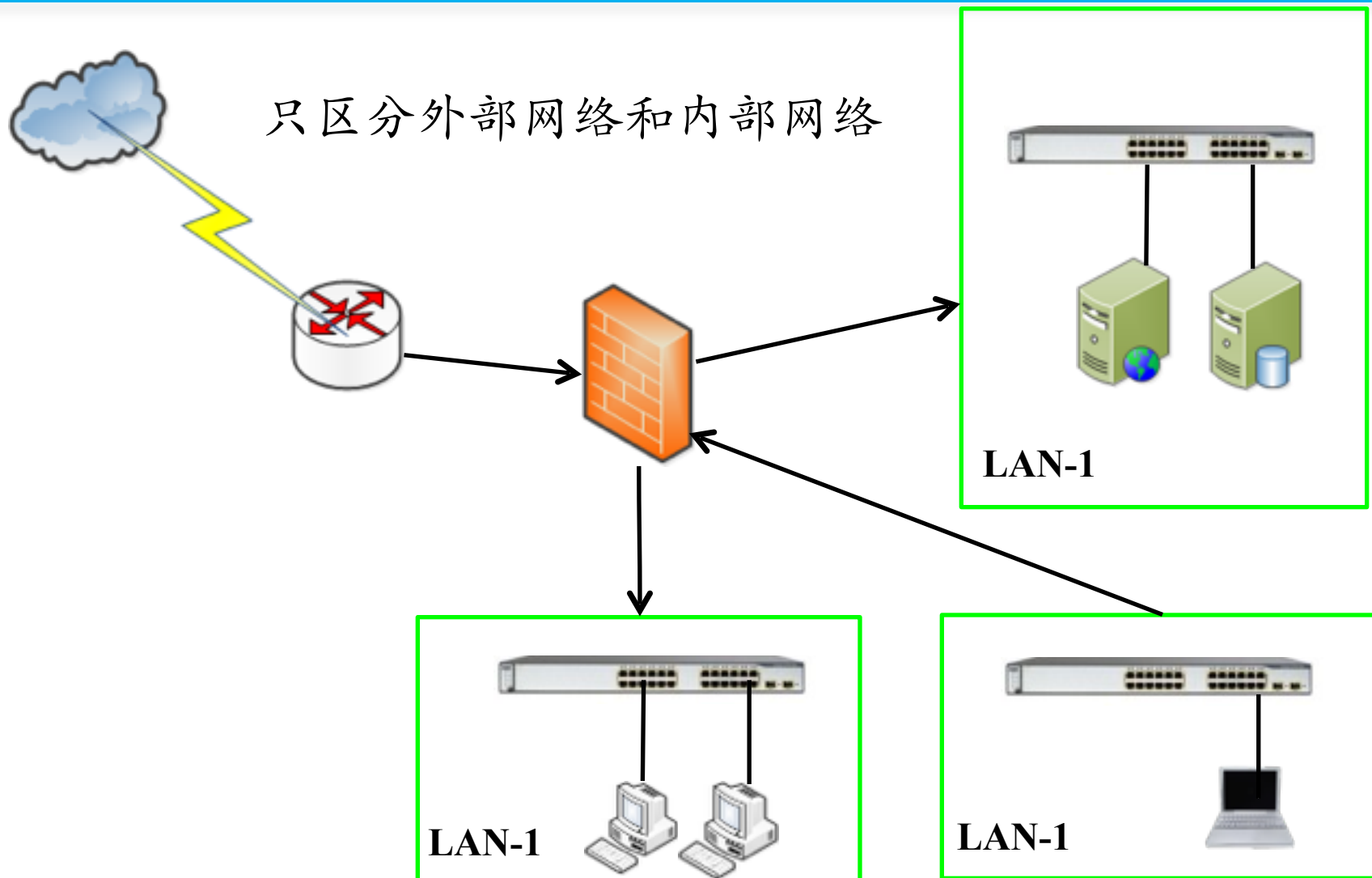


透明模式



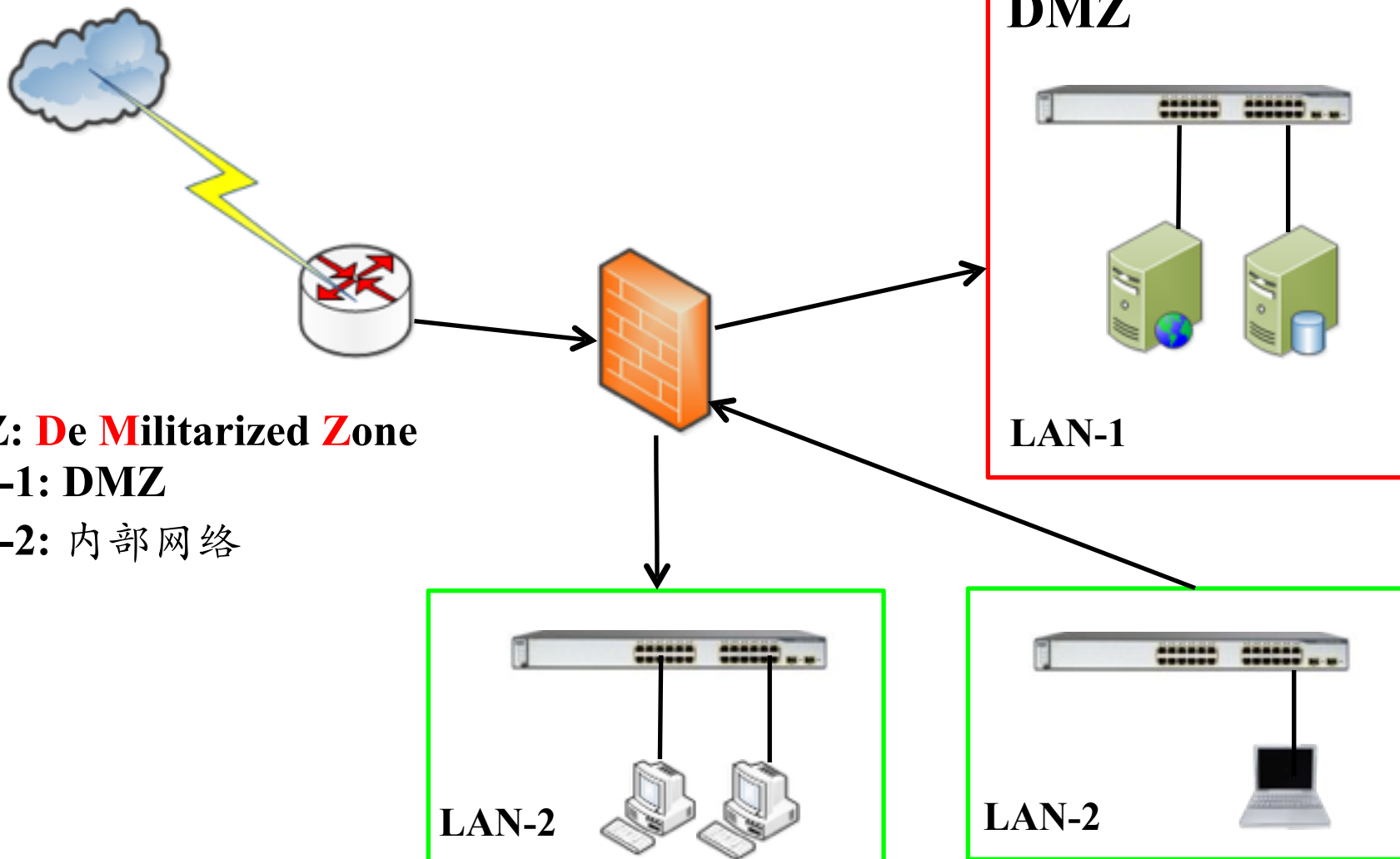


透明模式





混合模式



DMZ: **D**e **M**ilitarized **Z**one

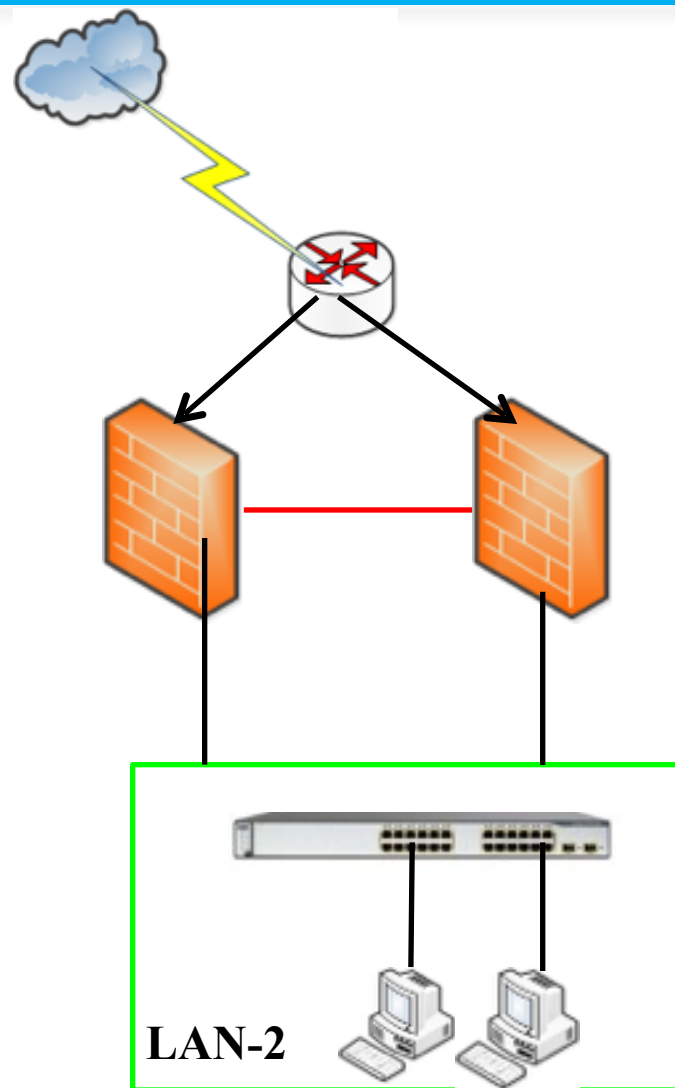
LAN-1: DMZ

LAN-2: 内部网络



防火墙部署的其他细节(1/2)

- 双机热备模式
 - 避免单点故障
- 负载均衡模式
 - 性能扩展
 - 避免单点故障





防火墙部署的其他细节(2/2)

- 防火墙在网络中的实际部署位置

- 串行接入在网络设备之前

- 骨干网路由器防火墙

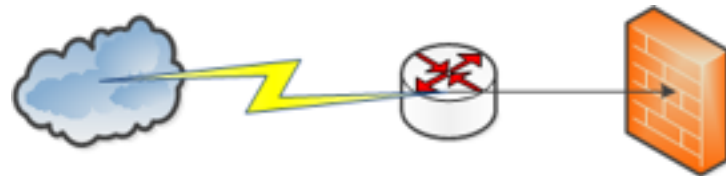
- 接入网核心交换机之后的防火墙



- 串行接入在网络设备之后

- 小型网络的接入防火墙

- 内网的子网防火墙



- 直接部署于应用服务器之上

- 单机网络防火墙

- 应用防火墙





单机防火墙配置

- ufw
 - ufw简介
 - ufw使用
- iptables
 - iptables简介
 - iptables使用



ufw简介

- ufw是为了使linux防火墙更易于使用和管理
- ufw和其他linux类防火墙一样，使用iptables作为后台
- 安装ufw
 - sudo apt-get install ufw
 - 通常ufw默认安装



ufw使用(1/3)

- 启用ufw
 - sudo ufw enable
 - sudo ufw default deny
- 作用
 - 开启了防火墙并随系统启动同时关闭外部对主机的所有访问，本机访问外部正常
- 关闭ufw
 - sudo ufw disable



- 查看防火墙状态
 - sudo ufw status
- 开启/禁用相应端口和服务举例
 - 允许外部访问80端口
 - sudo ufw allow 80
 - 禁止外部访问80端口
 - sudo ufw delete 80
 - 允许此IP访问所有的本机端口
 - sudo ufw allow from 192.168.1.54



ufw使用(3/3)

- 开启/禁用相应端口和服务举例(续)
 - 禁止外部访问smtp服务
 - `sudo ufw deny smtp`
 - 拒绝所有的流量从TCP的10.0.0.0/8到端口22的地址192.168.0.1
 - `ufw deny proto tcp from 10.0.0.0/8 to 192.168.0.1 port`
 - 可以允许指定网段访问这个主机
 - `sudo ufw allow from 10.0.0.0/8`



iptables 命令格式(1/3)

• iptables [-t 表] 命令 匹配 操作

—表选项，指定命令应用于哪个内置表(filter表、nat表、mangle表)

—命令选项

命令	说明
-P 或 --policy <链名>	定义默认策略
-L 或 --list <链名>	查看iptables规则列表
-A 或 --append <链名>	在规则列表的最后增加1条规则
-I 或 --insert <链名>	在指定的位置插入1条规则
-D 或 --delete <链名>	从规则列表中删除1条规则
-R 或 --replace <链名>	替换规则列表中的某条规则
-F 或 --flush <链名>	删除表中所有规则
-Z 或 --zero <链名>	将表中数据包计数器和流量计数器归零



iptables 命令格式(2/3)

—匹配选项

匹配	说明
-i<网络接口名>	指定数据包从哪个网络接口进入，如ppp0、eth0和eth1等
-o<网络接口名>	指定数据包从哪块网络接口输出，如ppp0、eth0和eth1等
-p<协议类型>	指定数据包匹配的协议，如TCP、UDP和ICMP等
-s<源地址或子网>	指定数据包匹配的源地址
--sport <源端口号>	指定数据包匹配的源端口号，可以使用“起始端口号:结束端口号”的格式指定一个范围的端口
-d<目标地址或子网>	指定数据包匹配的目标地址
--dport 目标端口号	指定数据包匹配的目标端口号，可以使用“起始端口号:结束



iptables 命令格式(3/3)

—动作选项

动作	说明
ACCEPT	接受数据包
DROP	丢弃数据包
REDIRECT	将数据包重新转向到本机或另一台主机的某个端口，通常用功能实现透明代理或对外开放内网某些服务
SNAT	源地址转换，即改变数据包的源地址
DNAT	目标地址转换，即改变数据包的目的地址
MASQUERADE	IP伪装，即是常说的NAT技术，MASQUERADE只能用于ADSL等拨号上网的IP伪装，也就是主机的IP是由ISP分配动态的；如果主机的IP地址是静态固定的，就要使用SNAT
LOG	日志功能，将符合规则的数据包的相关信息记录在日志中，以便管理员的分析和排错



iptables的使用(1/3)

- 定义默认策略

—当数据包不属于链中任一规则时，iptables将根据该链预先定义的默认策略处理数据包

- 默认策略定义格式

—iptables [-t表名] <-P> <链名> <动作>

—参数说明

- 表名,默认策略将应用于哪个表
- -P,定义默认策略
- 链名,默认策略应用于哪条链
- 动作, 处理数据包的动作



iptables的使用(2/3)

- 查看iptables规则

—iptables [-t表名] <-L> [链名]

- [-t表名],查看哪个表的规则列表
- -L,查看指定表指定链的规则列表
- 链名,查看指定表中哪个链的规则链表

- 增加、插入、删除、替换规则

—iptables [-t表名] <A|I|D|R>链名 [规则编号]

[i|o 网卡名称] [-s 源IP地址] [-d 目标IP地址]

<-j 动作>



iptables的使用(3/3)

- 清除规则和计数器

—iptables [-t 表名] <-F|-Z>

- [-t 表名], 指定默认策略应用于哪个表
- -F, 删除表中所有规则
- -Z, 将指定表中的数据包计数器和流量计数器归零



iptables 配置实例

- 传输层防护实例
 - 禁止其他主机ssh连接自己
 - 防止各种端口扫描
 - 禁止自己主机使用FTP协议下载
 - 禁用DNS接口
- 网络层防护实例
 - 防止ping洪水攻击
 - 屏蔽一个IP
- 数据链路层防护实例



传输层防护实例(1/4)

- 禁止其它机器通过ssh连接自己
—iptables -t filter -A INPUT -p tcp --dport 22 -j DROP
- 查看主机防火墙规则
—iptables -t filter -L

```
root@wzy-desktop: ~  
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)  
root@wzy-desktop:~# iptables -t filter -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination          tcp dpt:ssh  
DROP        tcp  --  anywhere              anywhere             tcp dpt:ssh  
  
Chain FORWARD (policy ACCEPT)  
target      prot opt source                destination  
  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination
```



传输层防护实例(2/4)

- 防止各种端口扫描

—iptables -A FORWARD -p tcp --tcp-flags
SYN,ACK,FIN,RST RST -m limit --limit 1/s -j
ACCEPT

- 参数解释

— -limit 1/s 表示每秒一次; 1/m 则为每分钟一次



传输层防护实例(3/4)

- 禁止自己主机使用FTP协议下载（即封闭TCP协议的21端口）
—iptables -I OUTPUT -p tcp --dport 21 -j DROP





传输层防护实例(4/4)

- 禁用主机的DNS端口(DNS为UDP协议, 使用53端口)
—iptables -I OUTPUT -p udp --dport 53 -j DROP





- 防止ping洪水攻击

—iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT

—说明

— 限制ping的并发数，每秒一次

- 限制一个ip访问自己主机

—iptables -A INPUT -s 192.168.1.102 -j DROP

—说明

— 限制了ip地址为192.168.1.102主机对自己的访问



数据链路层防护实例

- 阻断来自某个mac地址的数据包

—iptables -A INPUT -m mac --mac-source 00:1e:ec:f0:ae:77 -j DROP

—说明：

– 阻断了mac地址为00:1e:ec:f0:ae:77 对本机的连接

- 查看本机iptables表

```
root@wzy-desktop: ~  
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)  
root@wzy-desktop:~# iptables -t filter -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination  
DROP        all  --  anywhere               MAC 00:1E:EC:F0:AE:  
77  
  
Chain FORWARD (policy ACCEPT)  
target      prot opt source                destination  
  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination
```



基于防火墙实现NAT

- 什么是私有地址
- 什么是NAT
- NAT的工作原理



私有地址

- 私有地址(private address)属于非注册地址，是专门为组织机构内部使用而划定的

私有IP地址范围	子网掩码
10.0.0.0~10.255.255.255	255.0.0.0
169.254.0.0~169.254.255.255	255.255.0.0
172.16.0.0~172.31.255.255	255.255.0.0
192.168.0.0~192.168.255.255	255.255.255.0

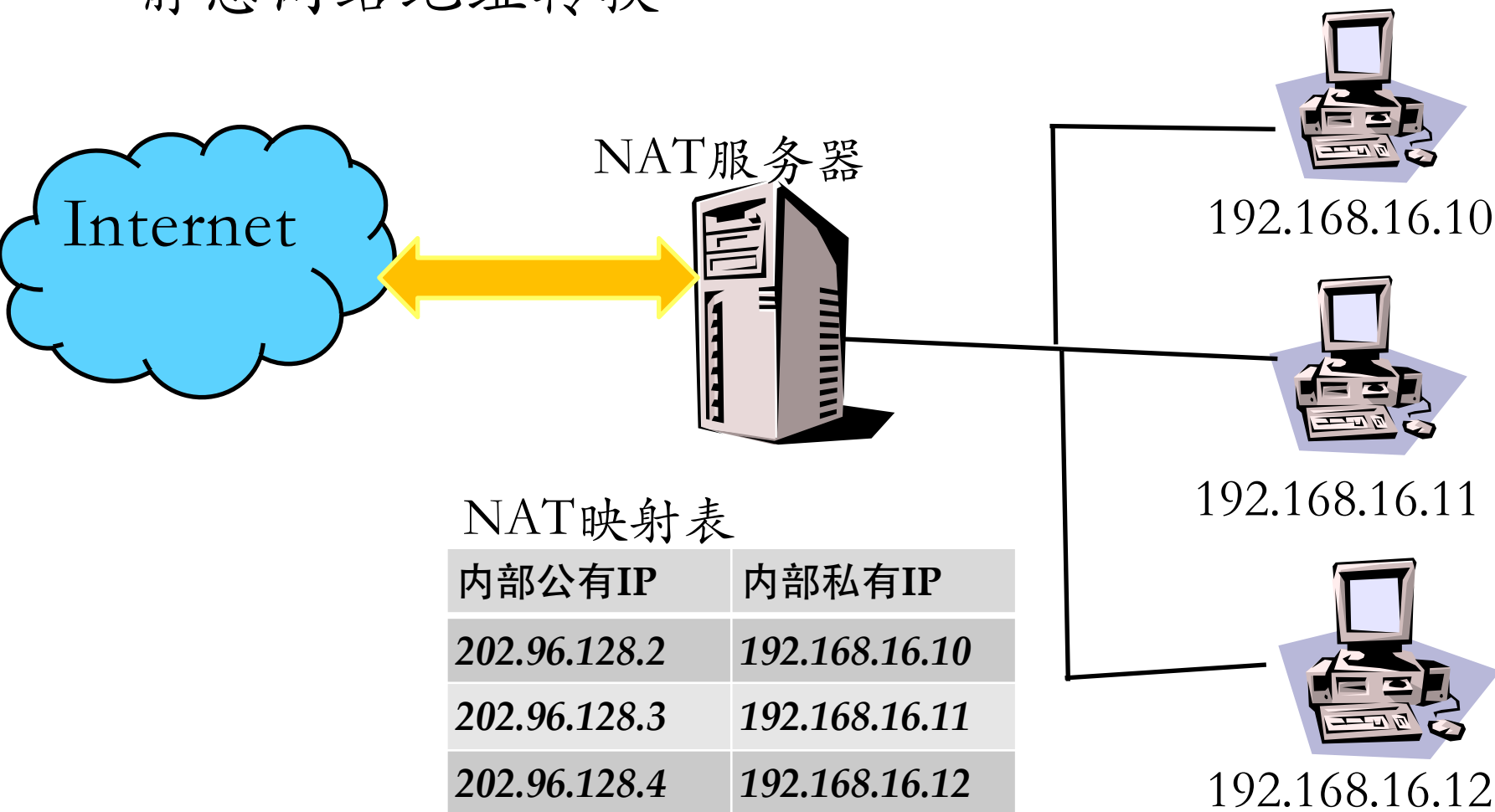


- NAT是将一个地址域(如专用Intranet)映射到另一个地址域(如internet)的标准方法
 - NAT可以将内部网络中的所有节点的地址转换成一个IP地址，反之亦然
 - 可以应用到防火墙技术里，把个别IP地址隐藏起来不被外部发现，使外部无法直接访问内部网络设备



NAT工作原理(1/3)

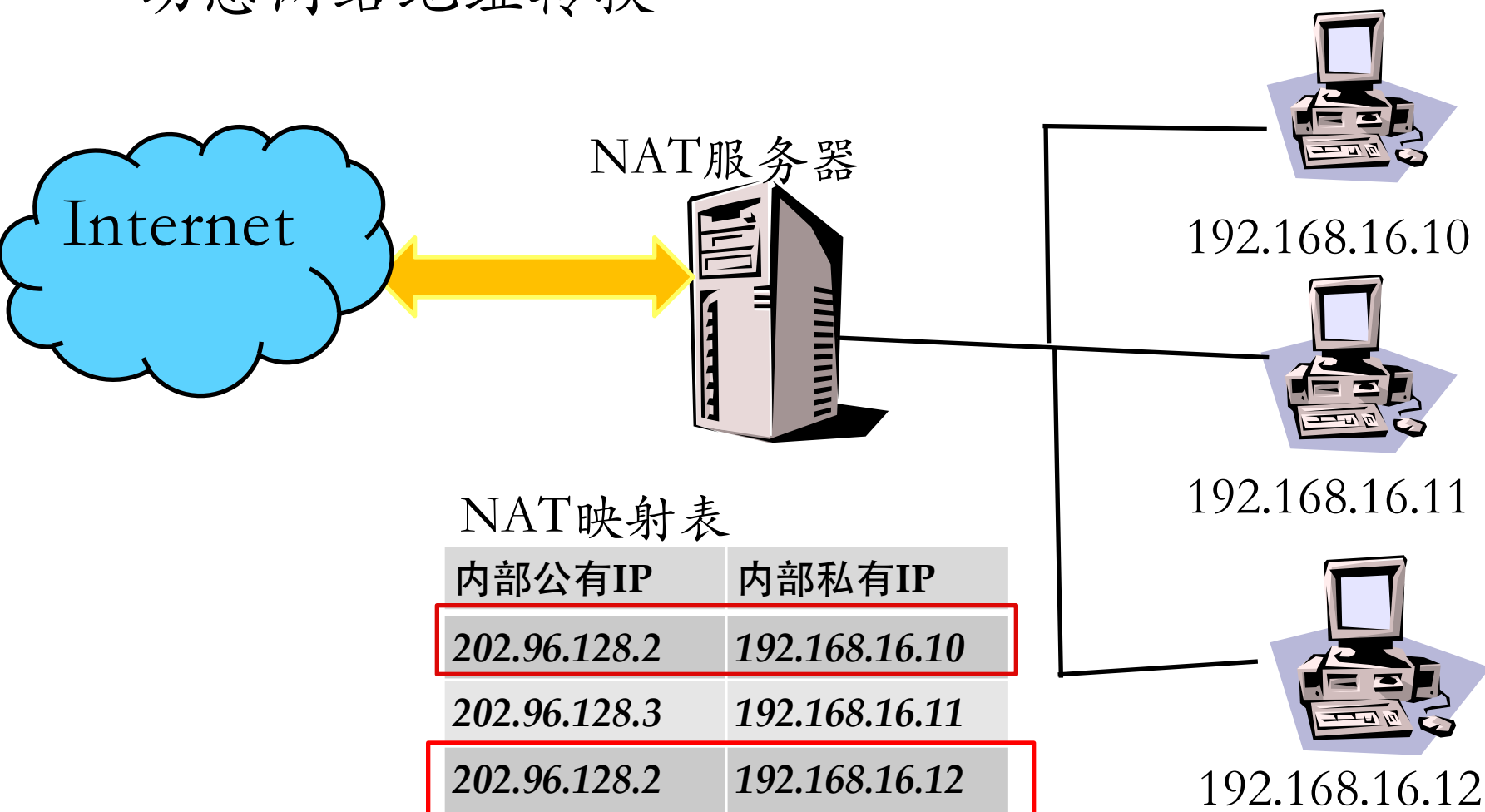
- 静态网络地址转换





NAT工作原理(2/3)

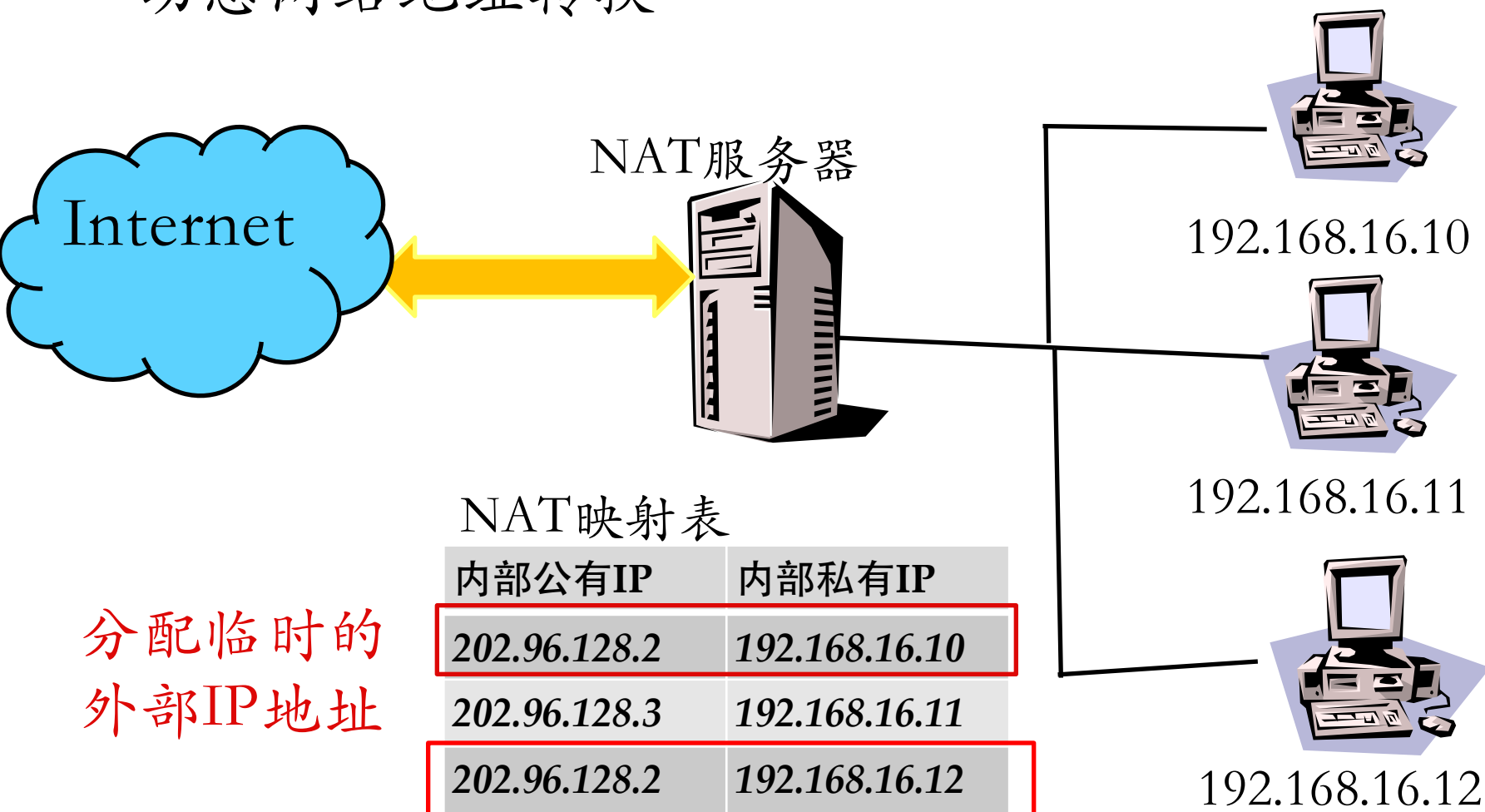
- 动态网络地址转换





NAT工作原理(2/3)

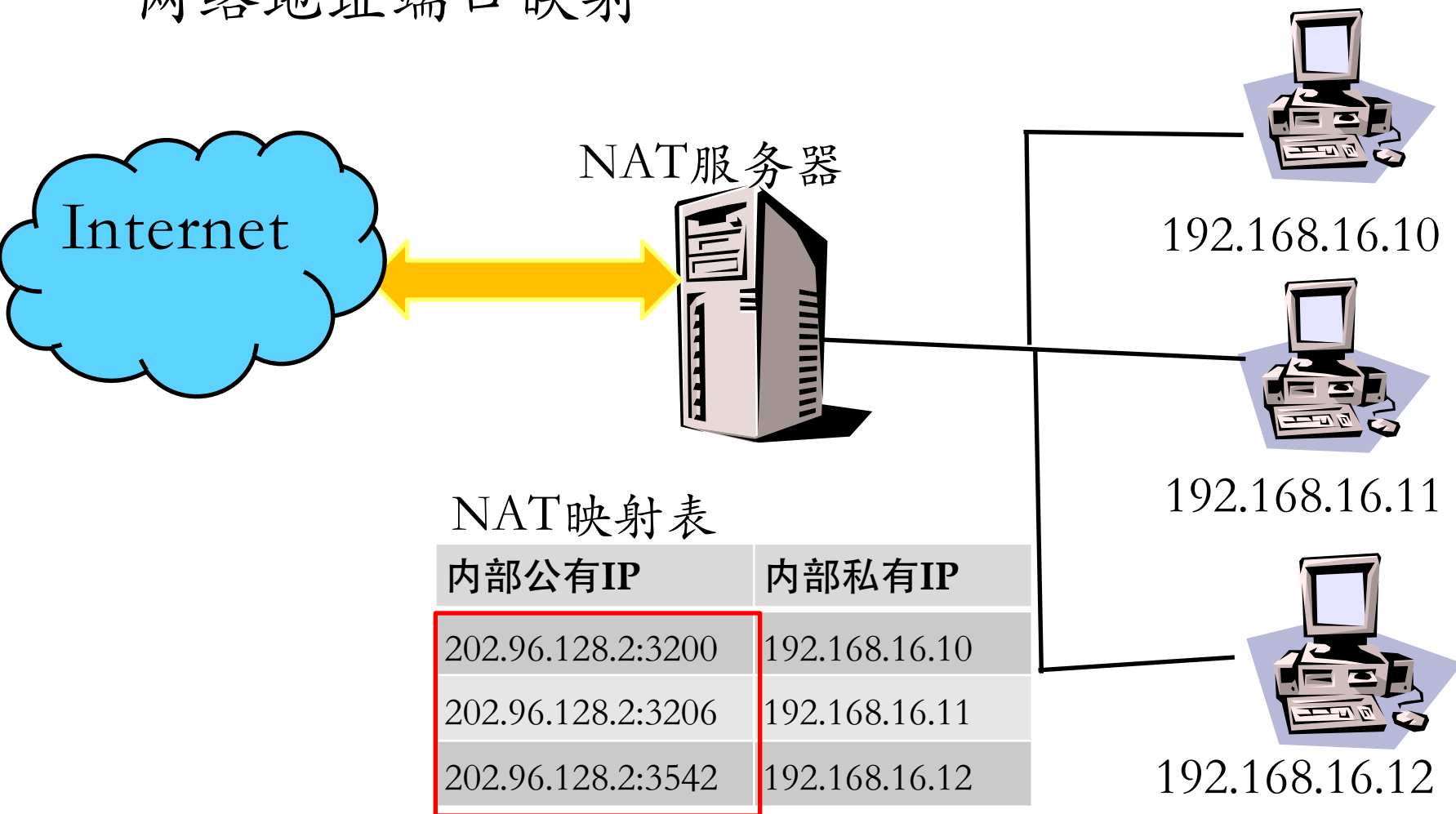
- 动态网络地址转换





NAT工作原理(3/3)

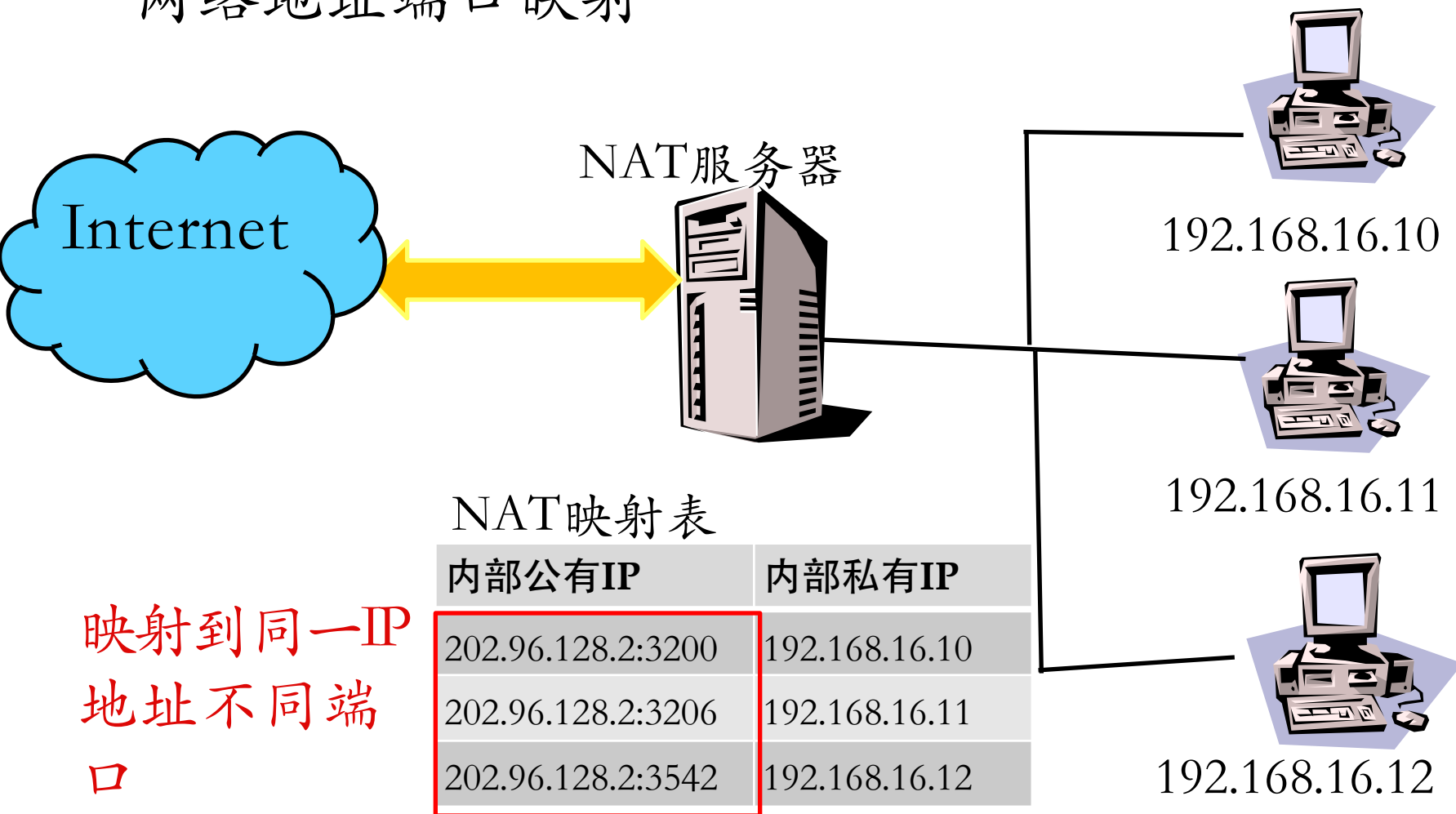
- 网络地址端口映射





NAT工作原理(3/3)

- 网络地址端口映射





使用实例(1/2)

- 源NAT

- 更改所有来自192.168.1.0/24的数据包的源ip地址为1.2.3.4

- iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -i eth0 -jSNAT to 1.2.3.4

- 注意，系统在路由及过虑等处理直到数据包要被送出时才进行SNAT



使用实例(2/2)

- 目的SNAT(DNAT)

- 更改所有来自192.168.1.0/24的数据包的目的ip地址为1.2.3.4

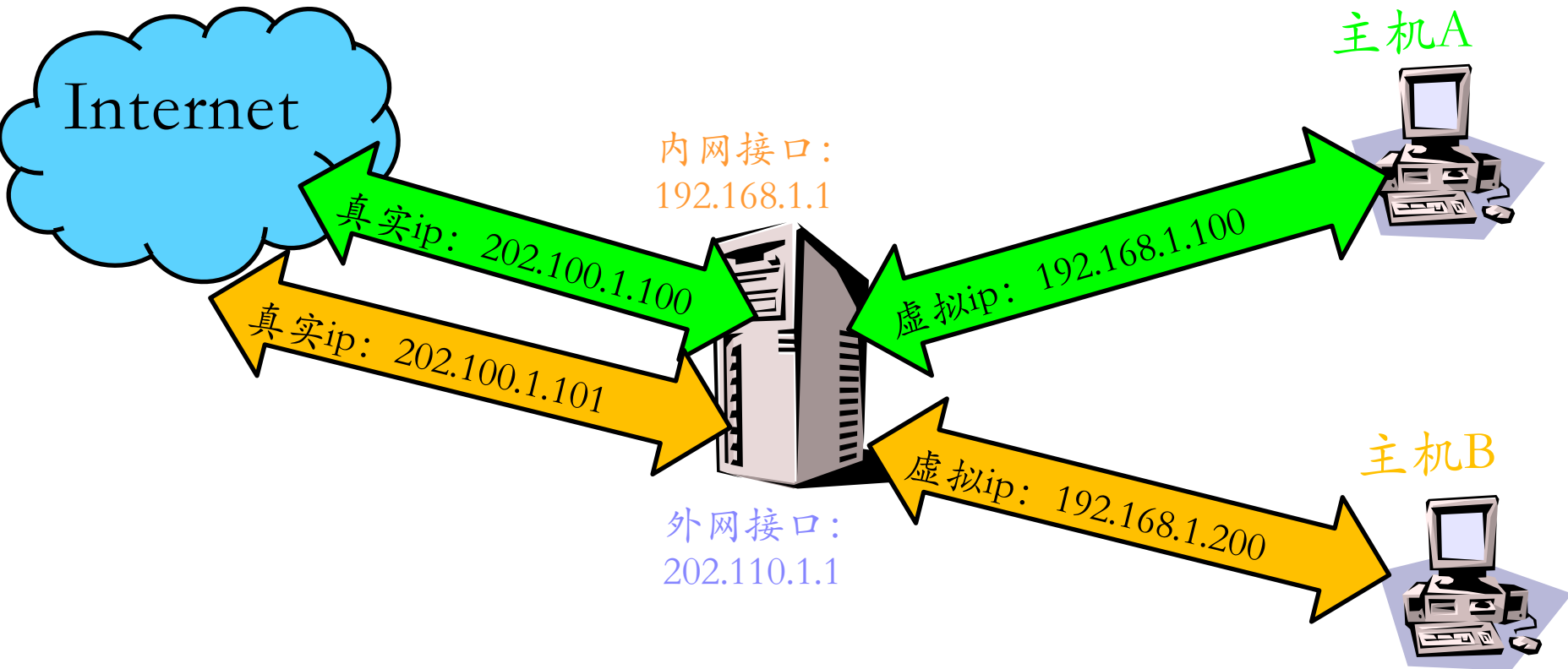
- iptables -t nat -A PREROUTING -s 192.168.1.0/24 -i eth1 -jDNAT--to 1.2.3.4

- 注意，系统是先进行DNAT，然后才进行路由及过虑等操作



iptables 实现NAT综合实例

- IP映射情景





IP映射原理

- 将分配给主机A、B的真实IP绑定到防火墙的外网接口
 - `ifconfig eth0 add 202.110.1.100 netmask 255.255.255.0`
 - `ifconfig eth0 add 202.110.1.101 netmask 255.255.255.0`
- 对防火墙接收到的目的ip为202.110.1.100和202.110.1.101的所有数据包进行DNAT
 - `iptables -A PREROUTING -i eth0 -d 202.110.1.100 -j DNAT--to192.168.1.100`
 - `iptables -A PREROUTING -i eth0 -d 202.110.1.101 -j DNAT--to192.168.1.200`



IP映射原理

- 对防火墙接收到的源ip地址为192.168.1.100和192.168.1.200的数据包进行SNAT
 - iptables -A POSTROUTING -o eth0 -s 192.168.1.100 -j SNAT--to202.110.1.100
 - iptables -A POSTROUTING -o eth0 -s 192.168.1.200 -j SNAT--to202.110.123.101



防火墙规则调试

- 查看概要统计数据

- iptables -L -v

```
Chain INPUT (policy DROP 15986 packets, 931K bytes)
  pkts bytes target    prot opt in     out     source    destination
 30642  33M ACCEPT    all  --  lo      any     anywhere  anywhere
   157 79153 DROP      tcp  --  any     any     anywhere  anywhere      tcp flags:!FIN,SYN,RST,ACK/SYN state NEW
2287K 1227M ACCEPT    all  --  any     any     anywhere  anywhere      state RELATED,ESTABLISHED
   49  3008 ACCEPT    tcp  --  any     any     anywhere  anywhere      tcp dpt:8322 state NEW
27246 1563K ACCEPT    tcp  --  any     any     anywhere  anywhere      tcp dpt:www state NEW
 1117 68884 ACCEPT    tcp  --  any     any     anywhere  anywhere      tcp dpt:https state NEW
   72  4472 ACCEPT    icmp --  any     any     anywhere  anywhere      icmp echo-request

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 1702K packets, 1600M bytes)
  pkts bytes target    prot opt in     out     source    destination
 30642  33M ACCEPT    _all  --  any     lo      anywhere  anywhere
```

- 日志法

- j LOG --log-prefix "DEBUG_IPT"

- t raw -A PREROUTING -j TRACE

- t raw -A OUTPUT -j TRACE

syslog

} /var/log/kern.log

- 黑盒测试

- 利用扫描器



防火墙规则安全审查

- 静态分析工具
 - 防火墙规则的语义理解
 - 数据流图分析
 - 自动化规则树生成
 - 举例
 - ITVal (见参考文献)
- 黑盒测试工具
 - 基于网络扫描器的fuzz测试



参考文献

- Netfilter/iptables官方文档 <http://www.netfilter.org/>
- iptables的相关概念和数据包的流程 <http://selboo.com.cn/post/721/>
- Robert Marmorstein , Phil Kearns, A Tool for Automated iptables Firewall Analysis, 2005 USENIX Annual Technical Conference



课后思考题

- 防火墙的典型网络部署方式有哪些
- 防火墙能实现的和不能实现的防护各有哪些