浮动主题

- NAT
  - NAT服务器本身就是一个路由器，只是NAT比路由器多了一个「IP 转换」的功能
    - 内网访问外网 SNAT
      - 首先将请求数据包（源：ipA，目标：ipC）发送到防火墙所在主机B，B收到后将数据包源地址改为本机公网网卡的ip（也就是能在公共互联网上路由的IP地址），并将私有IP与公有IP保存到内存中（源：ipB，目标：ipC），然后经互联网发送给C
      - C收到后将回应包（源：ipC，目标：ipB）转发给C的路由器，经互联网将回应包转发给B，B收到回应包后修改其目的地址，即回应包改为（源：ipC，目标：ipA）然后将数据包转发给A。
    - 外网访问内网 DNAT
      - C可以访问防火墙的公网地址，C的请求数据包（源：ipC，目标：ipB）到达防火墙B后，B会对比内存中已录的私有IP数据，然后在prerouting上将请求数据包的目标地址进行修改，并将数据包（源：ipC，目标：ipA）发送给A
      - A收到后进行回复发送响应包（源：ipA，目的ipC）到防火墙
      - 防火墙收到后对数据包源地址进行修改，并将响应包（源：ipB，目标：ipC）给C
  - 两个接口一边是公共 IP (public IP) 但一边是私有 IP (private IP) 呢？ 由于私有 IP 不能直接与公共 IP 沟通其路由信息，此时就得要额外的「IP 转译」功能了
  - SNAT与DNAT的关键在于nat表的PREROUTING与POSTROUTING链
- 防火墙
  - iptables只是一个命令行工具，位于用户空间，它依赖的是netfilter，位于内核空间　通过iptables来实现防火墙
    - 链
      - prerouting
      - input
      - forward
      - output
      - postrouting
    - 表
      - filter：负责过封包滤功能
      - nat表：网络地址转换：内核模块
      - mangle表：拆解报文，做出修改，并重新封装
      - raw表：关闭nat表启用的连接追踪机制
    - 1、不同的表只能存在特定的链上。2、一个链上可以存在多个表　链与表的关系
- lan
  - 一个网路区域
    - vlan
      - 可能会发出很多广播信息，比如arp、dhco、rip等。为了减少受影响的主机，通过vlan将域进行划分
      - 在二层划分广播域，每个广播域称为一个vlan
      - 每台主机只能属于一个 VLAN ，同属一个 VLAN 的主机通过二层直接通信，不同 VLAN 的主机只能通过 IP 路由功能才能实现通信。
      - 实现
        - 将交换器的端口以不同的vlan ID进行划分为不同的vlan
        - access模式
        - trunk模式
    - vxlan
      - 依托UDP层构建的overlay的逻辑网络，使逻辑网络与物理网络解耦
      - 对原有的网络架构影响小，不需要对原网络做任何改动，就可在原网络的基础上架设一层新的网络。
      - 依托于虚拟化隧道通信技术，通过三层的网络搭建虚拟的二层网络
      - VXLAN设备能通过像网桥一样的学习方式学习到其他对端的IP地址，也可以直接配置静态转发表
  - 域
    - 广播域
      - 目标MAC地址全部为1能够直接通信的范围。
      - 二层交换机只能构建单一的广播域，不过使用VLAN功能后，它能够将网络分割成多个广播域
- 协议
  - 将IP地址转换为MAC地址　arp
  - 为客户端动态分配ip　dhcp
  - 域名解析　dns
- 路由器
  - 根据目的IP地址的网络号进行路由
    - 广播
      - 主机号全为1就是该网段的广播地址
    - 组播(多播)
      - 用于将包发送给特定组内的所有主机。组播的地址范围是 224.0.0.0 ~ 239.255.255.255 ，其中 224.0.0.0 ~ 224.0.0.255 既可以在同一个网段内实现组播，又可以跨网段给全网所有组员发送组播包
- 交换机
  - 根据目的MAC地址的进行转发
    - 单播
      - 主机一对一的发送数据。发送地址是对端主机的 MAC 地址
    - 广播
      - 向局域网内所有设备发送数据。发送地址是 FF-FF-FF-FF-FF-FF
    - 泛洪
      - 将某个端口收到的数据从除该端口之外的所有端口发送出去。这个进行泛洪的必须是普通数据帧而不能是广播帧
- 负载均衡
  - 负载平衡器通常用于三层体系结构中
- 网络命名空间
  - 不同network namespace之间的路由表和iptables规则等也是隔离的
  - 用户可以随意将虚拟网络设备分配到自定义的network namespace里，而连接真实硬件的物理设备则只能放在系统的根network namespace中
  - 进程与网络命名空间
    - 将某个进程关联到这个网络命名空间后，这个网络命名空间是其进程组私有的，和其他进程组不冲突。
- Linux bridge
  - 子主题　Linux bridge的行为更像是一台虚拟的网络交换机，任意的真实物理设备（例如eth0）和虚拟设备（例如，前面讲到的veth pair和后面即将介绍的tun设备）都可以连接到Linux bridge上
  - 有多个端口，数据可以从任何端口进来，进来之后从哪个口出去取决于目的MAC地址
  - 特点
    - 1、Bridge 是二层设备，在数据链路层实现的，仅用来处理二层的通讯。2、Bridge 使用 MAC 地址表来决定怎么转发帧。3、Bridge 会从 host 之间的通讯数据包中学习 MAC 地址。如果它遇到一个自己从未学习到的地址，会将此报文广播到所有的网络接口。
- linux隧道
  - linux支持5种L3隧道　工作在三层的虚拟设备：为在操作系统内核和用户应用程序之间传递IP包
    - ipip：即IPv4 in IPv4，在IPv4报文的基础上封装一个IPv4报文
    - ...
  - 优点：简单，大部分工作都是由Linux内核的模块实现。缺点：需要额外的解包与封包，性能低
  - 实现
    - 在隧道ip头外面包一层原ip头，从当前主机的tun设备发送到物理网卡，由物理网卡根据新添加的ip头发送到另一端主机，另一端主机接收后解开这个头，发现内层的IP报文是自己的tun设备，于是将报文转给自己的tun设备。
- overlay网络
  - 优点：不需要额外的解包与封包，性能高。缺点：需要想办法维护路由表　在L3网络的基础上构建大二层的虚拟网络
- 应用层
  - 协议有：HTTP、FTP、STMP等　应用层
  - 负责数据格式转换，加密解密等编码工作　协议有：ASCL、JPEG　表现层
  - 负责建立、管理和终止表示层实体之间的通信会话　协议有：RPC、NFS等　会话层
- 传输层
  - 负责端到端的数据传输　协议有：tcp/udp
- 网络层
  - 负责数据的路由、转发、分片　协议有：ip、ICMP等
- 网络接口层
  - mac寻址，协议有802.2、arp等　数据链路层
  - 负责在物理网络中传输数据帧　物理层

- I/O类型
  - Linux为了改进性能，有时候会选择不直接操作硬盘，而是将读写都在内存中，然后批量读取或者写入硬盘。一旦能够命中内存，读写效率就会大幅度提高。根据是否使用内存做缓存，我们可以把文件的I/O操作分为两种类型：
  - 缓存I/O
    - 对于读操作来讲，操作系统会先检查，内核的缓冲区有没有需要的数据
    - 对于写操作来讲，操作系统会先将数据从用户空间复制到内核空间的缓存中。至于什么时候写到磁盘，由操作系统决定
  - 直接IO
- 硬连接与软连接
  - 硬链接与原始文件共用一个inode的，但是inode是不跨文件系统的，每个文件系统都有自己的inode列表，因而硬链接是没有办法跨文件系统的
  - 软链接相当于重新创建了一个文件。这个文件有独立的inode。可以跨文件系统
  - 创建软、硬链接时，要用绝对路径
- 文件存储
  - 文件在存储时，是将属性信息与具体内存分开存储的
  - 当磁盘读取时，以block为单位进行读取，需要读取的block越多，磁盘IO就越高，性能也就越低
  - block的默认大小是越大越好吗？不是
    - 调大点：视频公司、图片公司 因为他们的文件都比较大，可以节省IO消耗
    - 调小点：互联网公司 大部分都是小文件，如果block越大，则磁盘利用率越低
- 存储类型
  - 块存储　硬盘
  - 对象存储　专门创建一个数据库，保存文件保存路径，文件的md5值等信息，即把每个文件当成一个对象保存在数据库，而不是把每个文件都放在用户目录下
    - s3
    - 各种网盘
  - 文件存储
- LVM　将多块磁盘整合为一块进行分区。LVM可以实现分区弹性伸缩

Linux网络与存储

硬盘　ext4文件系统　驱动　VFS　文件子系统　内核态：系统调用　用户态：进程

二层到四层都是在linux内核里面处理的

- 请求方式
  - get、post、put、delete、option、trace、cnnect、head
    - head：和GET方法一样，只是不返回报文的主体部分。通常用于确认URI的有效性及资源更新的日期时间等
    - option：用来查询针对请求URI指定资源支持的方法
    - trace：客户端可以对请求消息的传输路径进行追踪，TRACE方法是让Web服务器端将之前的请求通信还给客户端的方法
    - connect：与代理服务器通信时建立隧道，实现用隧道协议进行TCP通信
  - get与POST的区别
    - GET请求会被浏览器主动cache，而POST不会，除非手动设置
    - GET请求在URL中传送的参数是有长度限制的，而POST么有
- 请求报文　请求
  - 请求行
  - 请求头
  - 空行　\r\n
  - 请求体
  - Accept：接收什么样的类型。Accept-Encoding：希望服务端返回的内容编码，这通常是一种压缩算法。Content-Length：指服务器发数据的大小。
- 响应报文　响应
  - 状态行
  - 响应头
  - 响应体
- 特点
  - 一个TCP连接可以被多个HTTP请求复用　长连接
  - 子主题　无状态

http

- 加密过程
  - 1、服务器将自己的公钥key1发给CA，CA用自己的私钥加密key1，并通过服务端网址等信息生成一个证书签名，并将证书返回服务端。
  - 2、通信时，服务端直接将证书发给客户端。客户端此时先验证证书真伪(本地)，得到服务器公钥。
  - 3、客户端检查证书合法性，颁发机构、过期时间。验证通过后就可以再次利用这个机构公钥，解出服务端的公钥key1。
  - 4、客户端生成自己的对称密钥key2，并用key1加密，发生给服务端。
  - 5、最后，服务端用自己的私钥解开加密，得到key2.于是双方用key2进行通信。

https

- 主要优化了传输压缩

http2

- 一种面向连接的、可靠的、基于字节流的传输层通信协议
  - 安全，稳定
  - 适合传输大量数据
  - 流量控制
  - 传输效率较低
  - 占用系统资源高
  - 三次握手，四次挥手
    - 为什么3?
      - 服务端接收到客户端的建立连接的消息，这时服务端只能确定它可以接收客户端的小心，需要客户端再回一个确认可以接收服务端的消息，这时才能说明连接是可靠的，此时才会正式建立连接
    - 为什么4?
      - server在接收到FIN时知道不会再有数据发来了，但是可能还有数据没发送完，所以先告诉客户端它的断开连接请求已经收到了，等服务端将剩它发送FIN时，说明服务端该全部发送完了，已经准备好断开了，此时客户端向服务端发送确认信号就可以断开了。
  - 可靠性机制
    - 序列号、确认应答
    - 超时重传
    - 流量控制：
    - 滑动窗口
    - 拥塞控制

tcp

- 由TLS 记录协议（TLS Record）和 TLS 握手协议组成　TLS
- 子主题　粘包　常见问题

- 主机号+网络号　子网划分

ip

- 四层负载均衡服务器在接受到客户端请求后，以后通过修改数据包的地址信息（IP+端口号）将流量转发到应用服务器。　四层
- 可根据七层的URL、主机名、浏览器类别、语言来决定是否要进行负载均衡。　七层

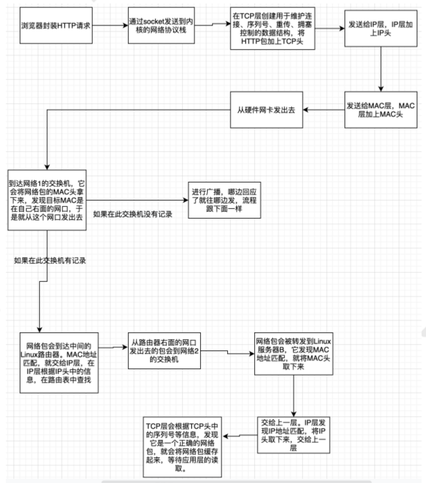
负载均衡

- 每个域的cookie数量是有限的（20个）
- cookie由服务器生成，发送给浏览器，浏览器把cookie以kv形式保存到某个目录下的文本文件内　cookie
- 分别作为key和value保存到缓存中，也可以持久化到数据库中，然后服务器再把sessionid，以cookie的形式发送给客户端。这样浏览器下次再访问时，会直接带着cookie中的sessionid。然后服务器根据sessionid找到对应的session进行匹配；　session
- 服务器第一次访问服务器，根据传过来的唯一标识userId。服务端会通过一些算法，如常用的HMAC-SHA256算法，然后加一个密钥，生成一个token。然后通过BASE64编码一下之后将这个token发送给客户端，客户端将token保存起来（保存到哪由前端决定，后端返回就行了）
- 下次请求时，带着token，服务器收到请求后，然后会用相同的算法和密钥去验证token，如果通过，执行业务操作，不通过，返回不通过信息；　token
- 为每个表单添加token并验证，果请求中没有token或者token内容不正确，则认为可能是CSRF攻击而拒绝该请求。　csrf

浏览器

浏览器

普通网络请求流程



图中漏了DNS解析　如果请求的是域名，那么会先向DNS服务器询问ip地址

ARP请求、应答帧格式

ARP请求包

目的地址，一般第一次都是ff:ff:ff:ff:ff:ff	源地址：源主机mac地址00:3b:4f:7f:43:1b	其他	源地址：源主机mac地址00:3b:4f:7f:43:1b	源主机ip地址:172.16.194.2	目的地址，一般第一次都是ff:ff:ff:ff:ff:ff	目标主机ip地址:172.30.25.1/165

ARP应答包

目的地址 00:3b:4f:7f:43:1b	源地址：源主机mac地址00:3b:4f:42:69:3c	其他	源地址：源主机mac地址00:3b:4f:42:69:3c	源主机ip地址:172.30.251.165	目的地址 00:3b:4f:7f:43:1b	目标主机ip地址:172.16.194.2

不管是隧道模式也好，还是overlay的大二层模式也好。本质都是在L3做文章，一个是再封装报文，一个是配置下一条主机的路由