网络

应用层

HTTP协议

特点：基于TCP

简要过程，浏览网页

1.浏览器输入URL（统一资源定位符）

2.2.DNS解析成IP

3.请求构建

HTTPS协议

- 加密分对称加密和非对称加密。对称加密效率高，但是解决不了密钥传输问题;
- 非对称加密可以解决这个问题，但是效率不高。非对称加密需要通过证书和权威机构来验证公钥的合法性。
- HTTPS 是综合了对称加密和非对称加密算法的 HTTP 协议。既保证传输安全，也保证传输效 率。

加解密

1.对称加密

2.非对称加密

客户端给外卖网站发送的时候，用外卖网站的公钥加密。而外卖网站给客户端发送消息的时候，使用客户端的公钥。这样就算有黑客企图模拟客户端获取一些信息，或者半路截获回复信息，但是由于它没有私钥，这些信息它还是打不开。

3.数字证书

Https的简要过程

流媒体协议

视频图片特点

时间/空间/视觉/编码冗余

视频编码两大流派

1.国际电联下的 VCEG

2.ISO 旗下的 MPEG

直播简要流程

1. 网络协议将编码好的视频流，从主播端推送到服务器，在服务器上有个运行了同样协议的服务端来接收这些网络包，从而得到里面的视频流，这个过程称为接流。
2. 服务端接到视频流之后，可以对视频流进行一定的处理，例如转码，也即从一个编码格式，转成另一种格式。因为观众使用的客户端千差万别，要保证他们都能看到直播。
3. 流处理完毕之后，就可以等待观众的客户端来请求这些视频流。观众的客户端请求的过程称为拉流。
4. 如果有非常多的观众，同时看一个视频直播，那都从一个服务器上拉流，压力太大了，因而需要一个视频的分发网络，将视频预先加载到就近的边缘节点，这样大部分观众看的视频，是从边缘节点拉取的，就能降低服务器的压力。
5. 当观众的客户端将视频流拉下来之后，就需要进行解码，也即通过上述过程的逆过程，将一串串看不懂的二进制，再转变成一帧帧生动的图片，在客户端播放出来，这样你就能看到美女帅哥啦。

编码：如果将图片变成二进制流？

一个视频，可以拆分成一系列的帧，每一帧拆分成一系列的片，每一 片都放在一个 NALU 里面，NALU 之间都是通过特殊的起始标识符分隔，在每一个 I 帧的第一 片前面，要插入单独保存 SPS 和 PPS 的 NALU，最终形成一个长长的 NALU 序列。

视频序列分三种帧

- I 帧，也称关键帧。里面是完整的图片，只需要本帧数据，就可以完成解码。
- P 帧，前向预测编码帧。P 帧表示的是这一帧跟之前的一个关键帧(或 P 帧)的差别，解码时需要用之前缓存的画面，叠加上和本帧定义的差别，生成最终画面。
- B 帧，双向预测内插编码帧。B 帧记录的是本帧与前后帧的差别。要解码 B 帧，不仅要取得 之前的缓存画面，还要解码之后的画面，通过前后画面的数据与本帧数据的叠加，取得最终 的画面。

可以看出，I 帧最完整，B 帧压缩率最高，而压缩后帧的序列，应该是在 IBBP 的间隔出现的。 这就是通过时序进行编码。

推流：如何把数据流打包传输到对端？

RTMP 协议

基于TCP

首先建立TCP连接

在TCP的基础上建立RTMP连接

为什么还要一个连接

主要就是两个事情，一个是版本号，如果客户端、服务器的版本号不一致，则不能工作。另一个就是时间戳，视频播放中，时间是很
重要的，后面的数据流互通的时候，经常要带上时间戳的差值，因而一开始双方就要知道对方的
时间戳。

P2P协议（peer-to-peer）

- 下载一个文件可以使用 HTTP 或 FTP，这两种都是集中下载的方式，而 P2P 则换了一种思 路，采取非中心化下载的方式;
- P2P 也是有两种，一种是依赖于 tracker 的，也即元数据集中，文件数据分散;另一种是基 于分布式的哈希算法，元数据和文件数据全部分散。

种子.torrent

1. 下载时，BT 客户端首先解析.torrent 文件，得到 tracker 地址，然后连接 tracker 服务器。 tracker 服务器回应下载者的请求，将其他下载者(包括发布者)的 IP 提供给下载者。下载者 再连接其他下载者，根据.torrent 文件，两者分别对方告知自己已经有的块，然后交换对方没有 的数据。此时不需要其他服务器参与，并分散了单个线路上的数据流量，因此减轻了服务器的负担。
2. 下载者每得到一个块，需要算出下载块的 Hash 验证码，并与.torrent 文件中的对比。如果一 样，则说明块正确，不一样则需要重新下载这个块。这种规定是为了解决下载内容的准确性问 题。

- 从这个过程也可以看出，这种方式特别依赖 tracker。tracker 需要收集下载者信息的服务器， 并将此信息提供给其他下载者，使下载者们相互连接起来，传输数据。虽然下载的过程是非中

心 化的，但是加入这个 P2P 网络的时候，都需要借助 tracker 中心服务器，这个服务器是用来登 记有哪些用户在请求哪些资源。

- 所以，这种工作方式有一个弊端，一旦 tracker 服务器出现故障或者线路遭到屏蔽，BT 工具就 无法正常工作了。

announce(tracker URL)

文件信息

- info 区:这里指定的是该种子有几个文件、文件有多长、目录结构，以及目录和文件的名 字。
- Name 字段:指定顶层目录名字。
- 每个段的大小:BitTorrent(简称 BT)协议把一个文件分成很多个小段，然后分段下载。
- 段哈希值:将整个种子中，每个段的 SHA-1 哈希值拼在一起。

去中心化网络（DHT-Distributed Hash Table）

DHT协议（Kademlia 协议）

DNS协议

- DNS 是网络世界的地址簿，可以通过域名查地址，因为域名服务器是按照树状结构组织的， 因而域名查找是使用递归的方法，并通过缓存的方式增强性能;
- 在域名和 IP 的映射过程中，给了应用基于域名做负载均衡的机会，可以是简单的负载均衡， 也可以根据地址和运营商做全局的负载均衡。

域名解析

负载均衡

HTTPDNS

- HTTPNDS 其实就是，不走传统的 DNS 解析，而是自己搭建基于 HTTP 协议的 DNS 服务器集群，分布在多个地点和多个运营商。当客户端需要 DNS 解析的时候，直接通过 HTTP 协议进行请求这个服务器集群，得到就近的地址。
- 使用 HTTPDNS 的，往往是手机应用，需要在手机端嵌 入支持 HTTPDNS 的客户端 SDK。

传统DNS的问题

域名缓存问题

域名转发问题

出口NAT(网络地址转换)问题

域名更新问题

解析延迟

主要应用于移动端

HTTPDNS缓存设计

缓存设计模式

- 对于应用架构来讲，就是应用、缓存、数据库。常见的是 Tomcat、Redis、MySQL。
- 对于 HTTPDNS 来讲，就是手机客户端、DNS 缓存、HTTPDNS 服务器。

客户端

缓存

数据源

HTTPDNS调度设计

传输层

TCP

特点：面向连接；提供可靠交付，无差错，不丢失，不重复，按序到达

如何可靠

窗口数据结构

发送端

发送已确认

发送未确认

未发送，可发送

未发送，不可发送

接收端

接收已确认

等待接收，未确认（最大工作量范围内）

不能接收（超过工作量）

顺序问题和丢包问题

1.超时重试

2.快速重传

接收方收到大于下一个期望报文时发三个冗余ACK，要求下一个包是指定包

3.SACK

在TCP头里加一个SACK，类似缓存地图告知丢哪一个包

流量控制：控制窗口大小，怕将接收方缓存塞满，控制发送包的速度

拥塞控制

控制窗口大小，怕把网络塞满

避免包丢失和超时重传

慢启动

TCP BBR 拥塞算法

它企图找到一个平衡点，就是通过不断的 加快发送速度，将管道填满，但是不要填满中间设备的缓存，因为这样时延会增加，在这个平衡 点可以很好的达到高带宽和低时延的平衡。

基于数据流

拥塞控制

三次握手

建立连接【Q】

了解建立连接时候的状态机

1. 客户端和服务端都处于 CLOSED 状态；
2. 先是服务端主动监听某个端口，处于 LISTEN 状态；
3. 客户端主动发起连接 SYN，之后处于 SYN-SENT 状态；

4. 服务端收到发起的连接，返 回 SYN，并且 ACK 客户端的 SYN，之后处于 SYN-RCVD 状态；
5. 客户端收到服务端发送的 SYN 和 ACK 之后，发送 ACK 的 ACK，之后处于 ESTABLISHED 状态，因为它一发一收成功 了；
6. 服务端收到 ACK 的 ACK 之后，处于 ESTABLISHED 状态，因为它也一发一收了。

解决TCP包的序号问题

四次挥手

正常四次挥手（和平分手）：

1. A:B 啊，我不想玩了。
2. B:哦，你不想玩了啊，我知道了。
3. 此时A不玩了，B还会有数据要处理；处于半关闭状态
4. 此时A可以不接收数据，也可以等B把数据发完，等待B主动关闭
5. B:A 啊，好吧，我也不玩了，拜拜。
6. A:好的，拜拜。
7. 整个连接关闭

不和平分手：

1. A说不玩了，直接跑路，此时是有问题的；

UDP

可以这样比喻，如果 MAC 层定义了本地局域网的传输行为，IP 层定义了整个网络端到端 的传输行为，这两层基本定义了这样的基因:网络传输是以包为单位的，二层叫帧，网络层叫 包，传输层叫段。我们笼统地称为包。包单独传输，自行选路，在不同的设备封装解封装，不保 证到达。基于这个基因，生下来的孩子 UDP 完全继承了这些特性，几乎没有自己的思想。

特点：面向无连接，继承IP包特性，不保证不丢失及按序到达

基于数据报

区别

- 面向连接和面向无连接
- 面向连接会先建立连接，所谓的建立连接，是为了在客户端和服务端维护连接，而建立一定的数据结构来维护双方交互的状态，用这样的数据结构来保证所谓的面向连接的特性。

网络层

协议

ICMP

ping 是基于 ICMP 协议工作的。ICMP全称Internet Control Message Protocol，就是互联网 控制报文协议。

查询报文类型

常用ping就是查询报文；

- 主动请求，并且获得主动应答
- 相当于侦查兵，发了多少包，回复多少个，时间花多少

差错报文类型

由于异常情况发起的

终点不可达

- 网络/协议/主机/端口不可达
- 需要设置分片但是设置了不分片位

源站抑制：让源站放慢发送速度

时间超时

路由重定向

三层设备：路由器

- 就是把 MAC 头和 IP 头都取下来，然后根据里面的内容，看看接下来把包往哪里转发的设备。
- 路由器是一 台设备，它有五个网口或者网卡，相当于有五只手，分别连着五个局域网。每只手的 IP 地址都 和局域网的 IP 地址相同的网段，每只手都是它握住的那个局域网的网关。

链路层（MAC层）

媒体访问控制（Medium Access Control）

MAC层解决问题

1.信息发送规则

信道划分

轮流协议

随机接入协议

2.发给谁，谁接收

链路层地址（MAC地址）

了解第二层网络包格式

3.差错校验

循环冗余检测

协议

ARP协议

地址解析协议：已知IP地址求mac地址

广播ARP请求，对应IP回复自己的mac地址，然后缓存到ARP表中

二层设备：交换机

交换机会记录每个机器插口的MAC地址，称之为转发表

物理层

每讲小结

1-3讲

第一讲：为什么学习网络协议
在浏览器应用输入url（应用层）后，dns解析，或者httpsdns，解析成IP，然后应用层打包封装给传输层；
传输层协议（tcp，udp），封装应用跟服务端端口，传输层封装给网络层；
网络层协议（IP协议），封装源与目的IP，给mac层；
mac层封装了mac地址（ARP协议——本地网络or外地，外地给网关，网关往往是路由器，通过路由表知道数据怎么走）

第二讲：网络分层的真实含义
想像自己是一个处理网络包的程序，怎么处理网络包
原则：所有在网络上跑的包，可以有下层没上层，不能有上层没下层

第三讲：ifconfig查看IP地址
- IP是用来寻址的，有定位功能
- mac是用来身份确认的，无定位功能

- ip addr / ifconfig

4-7讲

第四讲：DHCP协议

- 动态主机配置协议（DHCP）

第七讲：ping跟ICMP

了解ping的工作过程

8讲

网关：

- Gateway 的地址一定是和源 IP 地址是一个网段的。
- 网关往往是一个路由器，是一个三层转发的设备。
- 经过网关的包MAC地址都会变

路由：

- 静态路由

1. 玄奘西游
2. nat，包中IP地址会变
3. 欧洲十国游
4. 包中IP地址不变

- 动态路由

11-12讲（TCP）

第11讲：

- TCP 包头很复杂，但是主要关注五个问题，顺序问题，丢包问题，连接维护，流量控制，拥 塞控制;
- 连接的建立是经过三次握手，断开的时候四次挥手，**一定要掌握的状态图。**

第12讲：TCP协议下

笔记

- 如何保证可靠：对客户端每一个发送的包，服务端都会有回复；服务端没有回复，客户端会重发包

16讲

- 视频名词比较多，编码两大流派达成了一致，都是通过时间、空间的各种算法来压缩数据;
- 压缩好的数据，为了传输组成一系列 NALU，按照帧和片依次排列;

- 排列好的 NALU，在网络传输的时候，要按照 RTMP 包的格式进行包装，RTMP 的包会拆分 成 Chunk 进行传输;
- 推送到流媒体集群的视频流经过转码和分发，可以被客户端通过 RTMP 协议拉取，然后组合 为 NALU，解码成视频格式进行播放。

问题记录

网络包有了IP地址，为什么还要mac地址

1. 局域网内的IP是动态分配的，下线之后IP发生变化
2. 历史遗留问题，早期用mac地址，后来才用IP + mac
3. mac地址唯一

TCP建立连接为什么就三次握手，不是两次，或者四次？

建立连接过程：
1. A想跟B建立连接，A首先跟B发送请求；（一次）
2. B收到请求，回复A；（两次）
3. A收到回复，给B确认收到回复；（三次）

至此，开始发送数据；

如果两次，可能A没收到回复，建立连接就失败；
如果四次，就是B收到回复之后再回复A，没必要；

相关网络知识

CDN

- CDN 和电商系统的分布式仓储系统一样，分为中心节点、区域节点、边缘节点，而数据缓存 在离用户最近的位置。
- CDN 最擅长的是缓存静态数据，除此之外还可以缓存流媒体数据，这时候要注意使用防盗链。它也支持动态数据的缓存，一种是边缘计算的生鲜超市模式，另一种是链路优化的冷链 运输模式。

数据中心

VPN（虚拟专用网）

VPN 通过隧道技术在公众网络上仿真一条点到点的专线，是通过利用一种协议来传输另外一种协议的技术，这里面涉及三种协议:乘客协议、隧道协议和承载协议。

移动网络

2G

- 手机通过无线信号连接基站;

- 基站一面朝前接无线，一面朝后接核心网;
- 核心网一面朝前接到基站请求，一是判断你是否合法，二是判断你是不是本地号，还有没有钱，一面通过网关连接电话网络。

云中网络

SDN（软件定义网络）

- 用 SDN 控制整个云里面的网络，就像小区保安从总控室管理整个物业是一样的，将控制面和 数据面进行了分离;
- 一种开源的虚拟交换机的实现 OpenvSwitch，它能对经过自己的包做任意修改，从而使得云 对网络的控制十分灵活;
- 将 OpenvSwitch 引入了云之后，可以使得配置简单而灵活，并且可以解耦物理网络和虚拟 网络。

云中网络安全

- 云中的安全策略的常用方式是，使用 iptables 的规则，请记住它的五个阶段， PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING。
- iptables 分为四种表，raw、mangle、nat、filter。其中安全策略主要在 filter 表中实现， 而虚拟网络和物理网络地址的转换主要在 nat 表中实现。

流量控制技术QoS(Quality of Service)

- 云中的流量控制主要通过队列进行的，队列分为两大类:无类别队列规则和基于类别的队列规则。
- 在云中网络 Openvswitch 中，主要使用的是分层令牌桶规则(HTB)，将总的带宽在一棵树 上按照配置的比例进行分配，并且在一个分支不用的时候，可以借给另外的分支，从而增强 带宽利用率。

容器网络