

计算机网络原理

IA

协议

协议设计目的：为进行网络中的数据交换而建立的规则、标准或约定，即网络协议

协议三要素：**语法**（规定传输数据的格式）、**语义**（规定要完成的功能）、**时序**（规定各种操作的顺序）

协议分层结构的必要性：明晰简化，便于分析学习、各层独立，加速技术演进、统一接口，确保技术互通。
（分层结构，统一标准，模块独立）；缺点：（效率低，难以协同优化，开发复杂，抽象）

分层结构

基础概念：接口定义**服务原语**，**层和协议的集合**构成网络体系结构，每层协议称为**协议栈**

服务：两种类型无连接和面向连接；六个核心原语：连接请求、接受响应、请求数据、应答、请求断开、断开连接。

网络要求：1.不为特殊应用设计 2.可以运行在任何通信技术上 3.允许在网络边缘创新（不为增加新应用而改变网络） 4.足够可拓展 5.为新协议、新技术和新应用开放

OSI七层模型：物理层（定义如何在信道上传输比特）、数据链路层（实现相邻网络实体间的数据传输）、网络层（将数据包跨越网络从源设备发送到目的设备）、传输层（将数据从源端口发送到目的端口）、会话层（在应用程序之间建立和维持会话）、表示层（关注信息的语法语义，管理数据的表示方法）、应用层（提供应用程序便捷的网络服务调用）；缺点：1.从未被真正实现 2.技术实现糟糕（欠缺技术考虑，过于复杂，功能重复） 3.非技术因素（TCP/IP免费，OSI强加）

TCP/IP四层模型：网络接口层、网络层、传输层、应用层；优点：1.未体现核心概念 2.不具备通用性 3.混用接口与分层的设计（网络接口层） 4.模型不够完整（缺少物理层，数据链路层）

网络度量单位

带宽：在数字信道上传送数据的速率，单位为b/s

时延：指数据从网络一端传送到另一端所需的时间。**传输时延**：数据从结点进入传输媒体所需的时间（发送时延）；**传播时延**：电磁波在信道中需要传播一定距离而花费的时间；**处理时延**：主机或路由器处理分组花费的时间；**排队时延**：分组在路由器输入输出队列中排队等待处理所经历的时延

RTT：指从发送方发送数据到发送方接收到接收方的确认所经历的时间

吞吐量：单位时间通过某个网络的数据量，单位为b/s

利用率：单位时间内网络被利用的时间比例

时延带宽积：传播时延与带宽的乘积，按比特计数的链路长度

补充

端到端：指数据传输前，在两个端系统间建立连接，发送完毕后断开连接；例子：TCP协议，SMTP，HTTP

点到点：指发送端把数据发送给直接相连的设备，由设备进行转发；例子：UDP

三种交换：电路交换（抗毁性差，无法应对突发流量），报文交换（存储转发导致传输时延高），分组交换（灵活性高，传输时延低）（后两者在传送突发数据时可提升网络利用率）【这里可能考计算，要点：电路交换算上建立连接时间，分组转发加上单个分组转发时间乘上转发次数】

物理层

基本概念

功能：在连接各计算机的传输媒介上传输比特流。

四个特性：机械特性（接口的物理结构），电气特性（信号线的电气连接及有关电路特性），功能特性（描述接口执行的功能），过程特性（指明对于不同功能的各种可能事件出现顺序）

傅里叶分析

已知 $g(t)$ ，求 c, a_n, b_n

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$

- 将等式两边从 0 到 T 积分可得 c

$$c = \frac{2}{T} \int_0^T g(t) dt$$

- 用 $\sin(2\pi kft)$ 乘等式两边，并从 0 到 T 积分，可得 a_n

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi nft) dt$$

- 用 $\cos(2\pi kft)$ 乘等式两边，并从 0 到 T 积分，可得 b_n

$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi nft) dt$$

波特率：单位时间信号变化的次数，也称调制速率，频谱带宽

截止频率：低于截止频率的振幅衰减较弱，截止频率越高带宽越高，信号越逼真。

信道最大数据传输速率

无噪声有限带宽信道： $2H \log_2 V$ bps; V 为信号电平数，H 为信道带宽；推论：任何信号通过一个带宽为 H 的低通滤波器，则每秒采样 $2H$ 次即可完整重现该信号。

有噪声信道： $H \log_2(1 + S/N)$ bps; S/N 为信号功率与噪声功率之比，H 为信道带宽(Hz)，

信噪比： $10 \log_{10}(S/N)$ dB 【这里可能考计算，带入公式即可】

传输方式

串行传输：逐位传输，1.所需线路少，利用率高 2.端必须进行串并转换 3.必须同步，适用于远距离传输

并行传输：1.不需要对传输代码作时序变换 2.需要n条信道，成本高，但速率快

单工：只能单向传输

半双工：双向传输，但不能同时

全双工：双向传输，可同时

传输介质

导引型介质：指电磁波被导向沿着某一媒介传播；双绞线、同轴电缆、光纤；期望：低成本，低损耗，抗干扰，可弯曲

非导引型介质：指电磁波在空间中传播；无线电波、微波、红外线、激光；期望：低损耗，抗干扰，防遮挡，传播远

数字调制和多路复用

基带传输：不搬移基带信号频谱；

制码：**不归零制码：**低电平表示0，高电平表示1，但难以分辨一位的开始和结束，必须有时钟同步。**曼彻斯特编码：**从低电平到高电平表示0，从高电平到低电平表示1，每位中间的跳变可作为时钟自同步。**差分曼彻斯特编码：**每一位中间都有跳变（作为时钟），每位开始有跳变表示0，无跳变表示1。

频带传输：利用调制解调器搬移信号频谱，三种调制：调幅、调频、调相

复用：**频分复用(FDM)：**将频谱分成多个频段，每个频段分配给一个用户；**波分复用(WDM)：**将频谱分成多个波长，每个波长分配给一个用户；**时分复用 (TDM)：**将时间分成多个时隙，每个时隙分配给一个用户（信道利用率不高）；**统计时分复用 (STDM)：**根据用户需求动态分配时隙 **码分复用 (CDMA)：**多个站一起发送，信号叠加，靠地址码区分，发送比特1则发送码片，发送比特0则发送反码片，不同站点码片正交【这里可能考计算，要点：将基站的码片和站点码片内积，值为1则发送了1，为-1则发送了0，为0则没有发送（记得要除以码片长度）】

数据链路层

成帧

字节计数法：在帧头加上字节计数，但一旦出错，后续字节计数也会出错（长度包括本身）

带字节填充的定界符法：定界符（FLAG）用于区分前后两个不同的帧；如果数据中出现定界符，则在定界符前加转义字符（ESC），如果转义字符出现在数据中，则在前面加转义字符；缺点:不够灵活

带比特填充的定界符法：在数据中出现连续5个1时，后面加一个0；接收端检测到连续5个1，且后面为0，则删除0，若下一比特为1，则连同后面的0构成定界符，一帧结束。

差错控制

冗余信息：增加冗余信息，用于检错和纠错（比如每个比特传三份）

基础概念：**码字**： $n=m+r$, m 为数据位， r 为校验位， (n,m) 码。**码率**： m/n ，**海明距离**：两个码字不同位数的个数，海明距离为 d 可以检测 $d-1$ 位错误，纠正 $(d-1)/2$ 位错误。

奇偶校验：偶数个1校验位为0，奇数个1校验位为1，检测单比特错误，不能纠正

校验和：发送方：求和后取反，接收方：求和，若结果不是全1，则出错

循环冗余校验CRC：生成 n 位CRC，先选定 $n+1$ 位生成多项式 G ，将原始数据乘以 2^n ，再除以 G ，余数即为CRC码；接收方若能整除则无错，否则有错。可以检测少于 $n+1$ 位的错误。【这里可能靠计算，记住：2进制除法实际上是异或运算】

海明码：设置2的幂次方位为校验位，将数据位 k 写成2的幂次位的和，即影响这些位，奇数个1则校验位为1，偶数个1则校验位为0；接收方：检测校验位，若出错则找到出错位，取反即可。【这里可能考计算，画图解即可】



可】

数据链路层协议

乌托邦单工协议P1：假设1，单工，数据单向传输；2.完美信道，数据不会丢失或受损；3.始终就绪：发送方和接收方的网络层始终处于就绪状态 4.瞬间完成：发送方和接收方能够生成处理无穷多的协议

发送方不断发送，接收方不断接收

无错信道上的停等协议P2：考虑到接收方处理速率有限，可能被发送方淹没，构建基于反馈的流量控制；仍然假设信道无措，数据单向（但是用半双工）；增加**确认机制**。

有错信道上的停等协议P3：考虑到信道可能出错，且帧可能丢失；增加**计时器机制**，如果经过一段时间未收到确认，则重传。导致问题：1.ACK丢失，B收到副本。2.延时过长，同样产生副本。——为了解决接收端重复接收问题，引入**序号机制**：发送方在帧头加入序号，接收方收到后检查序号，若与上一帧相同则丢弃（序号最短1bit即可）。

滑动窗口协议：为了解决停等协议的信道利用率低下的问题，允许发送方一次发送多个帧。但产生了新的问题，发送方要知道发了哪些帧，需要重传什么帧；接收方要排序帧，并且避免被发送方淹没。解决方案：发送方维护**发送窗口**，已发送但未确认的帧；接收方维护**接收窗口**，允许接受的帧，在外则丢弃。

一比特滑动窗口协议P4：窗口大小为1bit，采用**捎带确认**，接收方在发送确认时，将下一个帧一起发送，以减少延时。

问题：双方同时发送会有一半重复帧（本质上仍然是停等）

回退N协议P5：目标1：向上层按顺序提交；目标2：实现流水线机制，提高信道利用率。简化：接收窗口为1

发送方一次发送N个帧，接收方接收回发ACK，如果发送方收到ACK（累积确认），则窗口右移，如果有一个帧超时，就重传已发送但未确认的所有分组。（发送窗口小于 $2^n - 1$ ，因为 2^n 发送方无法判断ACK是那一次的，接收方无法判断数据是否是重传）**三个功能**：1.上层发送数据2.收到ack3.超时事件

采取**累计确认**，确认代表前面的帧都已收到。累积确认不代表接收到不返回ack，而是全部返回目前已接收的最大序号ack（所以如果返回过程中丢包也没有关系，不会重传）

缺点在于：重传所有未确认的帧，可能导致浪费。

选择重传协议P6 目标：能否仅重传出错的帧，不重传后续可能正确的帧。

由于乱序接收，按序交付，所以接收方要暂存接收到的帧——接收窗口大于1；所以缺点在于接收方需要占用一定的存储空间。

过程：发送方一次发送N个帧，接收方接收后窗口右移并返回ACK，发送方收到ACK后窗口右移。发送方对每一个帧设置一个计时器，如果超时则重传。

选择重传不采用累积确认，因为发送方要知道哪个帧没有收到ack；所以返回ack丢失时，选择重传协议的效率是不如回退N协议的。

窗口最大为 $2^n - 1$ ，

信道利用率 停等协议为 $F / (F + R \cdot RTT)$; F为帧大小，R为带宽，总式子 $t_D(\text{有效发送时延}) / (t_F(\text{发送时延}) + t_A(\text{返回发送时延}) + 2t_P(\text{传播时延}) + 2t_{\text{proc}}(\text{处理时延接收方和发送方}))$ ；回退N帧和选择重传协议的信道利用率为 $WT(\text{发送窗口大小}) \cdot t_D / (t_F(\text{这里是一帧的发送时延}) + t_A + 2t_P + 2t_{\text{proc}})$

吞吐量： $WT \cdot F / (RTT + t_F)$ （这里是一帧的发送时延）

【这里也可能考计算，要点：信道利用率的关键，就是有效时间（实际发送数据的时间）除以整个发送周期（从发送数据开始到接收完ACK）易错：如果是捎带确认，需要包含接收数据的时间】

MAC子层

多路访问协议

为了解决共享信道访问的冲突问题，协同多个设备。

随机访问 纯ALOHA协议：随意发送，冲突重传；冲突危险期为 $2D$ ，没有帧到达的概率为 e^{-G} ; (G 为平均负载，即所有站点发送的平均帧数)；一个帧时内到达的平均帧数： $S = G \cdot e^{-G}$ ；信道最高利用率为**18.4%**

分隙ALOHA协议：将时间分为多个时隙，每个时隙发送一个帧；信道最高利用率为**36.8%**（冲突只发生在时隙开始，时隙的长度为一帧的传输时间，冲突危险期为 D ）

CSMA协议：**（持续式）**监听信道，空闲则发送，冲突则等待随机时间再侦听，忙碌则持续侦听；优点：等待延迟小，缺点：介质空闲容易发生冲突；**（非持续式）**侦听，空闲则发送，忙碌则等待随机时间再侦听；优

点：冲突减少，缺点：信道利用率下降，传输时延增加；**(p-持续式)** 侦听，空闲则p概率发送，1-p概率等待一个时间单元发送，忙碌则持续侦听，如果发送已推迟，则重复步骤1。

受控访问 位图协议：每个周期开始时设置竞争期，每个站点发送竞争比特；传输器按序发送。**信道利用率**： $d/(d+N)$ N为站点数，d为每帧比特数；高负荷状态下信道利用率高，缺点是时延较大，没有考虑优先级

二进制倒数计数协议：站点被分配序号，如果站点要使用，则广播序号，序号低者监听到序号高者，则放弃，序号高者获得发送权。**信道利用率**： $d/(d+\log_2 N)$ 缺点：高序号优先，低序号可能无法发送

令牌环协议：令牌环传递，发送站抓取令牌后发送，监控站保证令牌不丢失，清除坏帧。优点：重负载效率高；缺点：需要维护令牌。

有限竞争协议 自适应树搜索协议：将站点组织成二叉树，从根节点开始竞争。如果有冲突，向下探索（优先左子树），获得信道后，探索右子树。重负载时为提高效率，可以考虑从中间节点开始竞争

以太网

CSMA/CD协议：为解决发送后产生冲突的问题。侦听，空闲则发送，忙碌则持续侦听，发送冲突，停止并发送Jam强化序号，等待随机时间重复步骤1。为了能够检测到冲突，冲突窗口（发送站发出帧后能检测到冲突图的最长时间）**必须为RTT**，因此最小帧长即为**RTT*带宽**。（以太网最短帧长为64字节）【这里可能考计算，要点：最小帧长即为RTT时间内传输的数据大小】

等待随机时间：**二进制指数后退**，重传k次，取 $k=\min[\text{重传次数}, 10]$ ；再从 $[0, 2^k-1]$ 中随机选取一个数，乘以512bit（时间槽2t），作为等待时间。

帧格式：目的地址（6字节）+源地址（6字节）+类型（2字节）+IP数据报（46-1500字节）+校验和（4字节）；

MAC地址：物理地址，长度6字节，区分单播广播和组播地址；广播为全FF。厂商分配

链路层交换

交换：将帧从一个链路传输到另一个链路，实现扩展。**Hub集线器**：物理层设备，使用同意更总线，同一个冲突域使用集线器。使用CSMA/CD

交换机：链路层交换设备，每个端口代表一个冲突域，全双工，并行传输。通过检测MAC帧的**目的地址**进行转发。

理想的网桥：1.即插即用无需任何配置 2.提升网络效率避免无谓转发 3.站点无需感知网桥的存在 4.网桥本身应避免复杂的配置

交换原理：1.Forwarding（转发）：当目的地址在MAC地址表中存在且端口不同时，直接发送。2.Flitering（过滤）：如果目的地址在MAC地址表中找到匹配项并且端口相同，丢弃。3, Flooding（泛洪）：在MAC地址中没有找到匹配项（或者广播帧），则向所有端口发送（除了入境口）；4.**逆向学习**：逆向学习MAC的源地址和端口，记录帧到达时间和老化时间，老化时间到期时表项删除；如果相同地址的帧重新到达，则重置帧到达时间和老化时间。

生成树协议

问题：可靠网络需要冗余拓扑，但会导致物理环路；物理环路导致1.MAC地址表不稳定 2，重复帧 3.广播风暴：交换机无限循环转发广播帧

BPDU：根桥ID（8字节）：被选为根的桥ID；根路径开销：到根桥的最小路径开销；指定桥ID：生成和转发BPDU的桥ID；指定端口ID：生成和转发BPDU的端口ID；

生成树协议：目的得到无环的生成树。**1.选举根桥**，选取优先级最小的交换机，优先级相同选取MAC地址最小。根桥的所有端口处于转发状态。**2.为非根桥选出根端口**：非根桥比较每个端口到根桥的路径开销，选取最小开销的端口为根端口，相同取端口ID最小的；根端口处于转发状态。（根路径开销为到根桥上所有端口链路开销之和，开销由速率定义）；**3.为每个网段确定一个指定端口**：对于每一个网段，选取所有连接到他的交换机端口的最小根路径开销作为指定端口，指定端口处于转发状态，负责转发该网段的数据 **缺点**在于两网段间的路径并不是最短的。

源路由网桥：1.需求：定位目的地，构造路由序列； 2.通过广播发现帧获得最佳路由，经过网桥加上桥标识，目的站收到后发送应答； 3. 优点在于：对拓扑带宽进行最优使用； 缺点：网桥的插入对于网络不透明

交换机交换模式：1.存储转发：转发前接收整个帧执行CRC校验，优点不转发出错帧，支持非对称转换 缺点转发延迟大 2.直通交换：接收到目的地址（前14字节）后直接转发，优点转发快 缺点可能转发错误帧，不支持非对称交换 3.无碎片交换：接收到帧的前64字节开始转发 优点：过滤了冲突碎片，延迟和转发错帧在中间 缺点：不支持非对称交换，仍然可能转发错误帧

虚拟局域网：避免安全泄露，将同一个域划分为不同广播域，可以基于端口、MAC、协议、子网构建；数据帧中需要携带VLAN标识，对终端站点透明。


以太网的扩展：1.希望隔离互通且无配置：使用交换机 2.避免物理环路：沈城树 3.大规模以太网时延：转发模式 4，虚拟局域网

CSMA/CA：因为CSMA/CD在无线信道中遇到1.冲突检测困难 2.载波侦听失效 问题。所以采用CSMA/CA——当信道空闲时间大于IFS（帧间隙）时发送，当信道忙时：延迟当前传输结束+IFS时间，随机退后，从（0，CWINDOW）中选取一个随机数作为计数器；侦听时间槽，如果空闲则继续减少退后时间，如果忙则挂起暂停退后过程。 **并且会确认接收到的帧。**

IFS类别：DIFS：优先级最低，最长，用于异步数据服务；PIFS：优先级中等，长度居中，轮询服务；SIFS：优先级最高，最短，用于ACK,CTS,轮询响应等。信道空闲后等待DIFS发送RTS。

RTS-CTS机制：目的，通过信道预约，避免数据帧冲突。无线问题：1.隐藏终端问题：由于距离太远导致站点无法检测到竞争对手存在 2.暴露终端问题：由于侦听到其他站点的发送误以为信道忙不能发送 解决：发送方发送RTS，接收端回送CTS，RTS和CTS持续时间指明传输所需时间，其他站点收到RTS或CTS维护NAV。

帧格式：



说明	去往DS Distribution System	来自DS Distribution System	地址1 (物理接收者)	地址2 (物理发送者)	地址3 (逻辑发送者)	地址4 (逻辑接收者)
自组织模式	0	0	DA	SA	IBSSID	—
接收自AP	0	1	DA	BSSID	SA	—
发送至AP	1	0	BSSID	SA	—	
AP到AP	1	1	接收AP	发送AP	SA	DA

其中DA表示目标地址，SA为源地址，BSSID：AP的MAC地址；AP为基站。

WLAN IEEE 802.11：以无线信道作为传输介质，使用CSMA/CA协议，采用RTS-CTS机制，帧格式如上图，采用32位CRC校验，采用停等机制

网络层

IPv4编码和分片

IPv4地址：32位，A类：0.0.0.0~127.255.255.255 B类：128.0.0.0~191.255.255.255; C类：192.0.0.0~223.255.255.255(ABC用于单播，分配给主机) D类：224.0.0.0~239.255.255.255 (用于组播) E类：240.0.0.0 ~ 255.255.255.254(保留位今后使用)。

CIDR：斜线后为网络前缀长度，A类第一个字节为网络号，后三个为主机号；B类前两个为网络号，后两个为主机号；C类前三个为网络号，后一个为主机号。主机号全为1为子网广播地址，全为0表示子网地址，其余为主机。

子网：前缀全为1，主机全为0为子网掩码；前缀相同为一个子网，子网内可分配ip地址为去掉主机号全1和全0的地址，主机数量为 2^n-2 个。【这里可能考计算，要点：根据子网掩码和ip地址计算子网地址和广播地址】

特殊地址：127.0.0.0指本地，全0主机地址：网络本身；全1主机地址：广播，全0，任意地址；全1，本地广播地址。

最长前缀匹配：路由表中的前缀最长的匹配为最佳匹配。

分片：当数据包超过MTU时，需要分片分片IP头，DF标志置为0，MF标志除最后一片外全置为1，最后一片置为0；偏移为前面分组总数据长度除以8（单位为8字节），去除首部后能装1400字节（不同不一样）（头基础为20B，后面如果有选项（严格源路由、时间戳等）和填充则更长）

网络层基础协议

DHCP协议：动态获取IP地址，工作模式为C/S。工作过程：

阶段	源MAC	目标MAC	源IP	目标IP	链路层
Discover	PC机的MAC	全FF	0.0.0.0	255.255.255.255	广播
Offer	DHCP服务器 (如路由器)的 MAC	DHCP客户 机的MAC	DHCP服务器 (如路由器)的 IP地址	255.255.255.255	单播
Request	PC机的MAC	全FF	0.0.0.0	255.255.255.255	广播
Ack	DHCP服务器 (如路由器)的 MAC	DHCP客户 机的MAC	DHCP服务器 (如路由器)的 IP地址	255.255.255.255	单播

ARP协议：通过IP地址获取MAC地址，国博ARP请求分组，目标ip主机单播发送ARP响应分组。ARP表中缓存IP地址和MAC地址的对应关系。

IP包转发：在一个ip子网内，直接利用MAC地址通过交换机转发；不在一个IP子网内，先mac转发给路由器，路由器转发给目的主机

ICMP协议：用于差错报告和询问（ping），4字节头，包含类型、代码（前两字节指定功能），检验和

路由协议

距离向量算法RIP：采用分布式Bellman-Ford算法，各节点维护到达每个节点的最优<距离，向量>，各节点与邻居节点交换信息，更新；RIP：1.路由器更具配置地址初始化路由表，把直连路由距离记为0；2.周期性向相邻路由表广播路由信息，根据收到的信息更新路由表；3.路由表最后收敛。**简化**：RIP只使用跳数代表距离，周期性更新

RIP更新规则：1，如果新的目的地址，增加相应表项 2.如果目的地址相同，且下一跳路由器相同，直接更新 3.如果目标地址相同，下一跳路由器不同，且距离更短，更新

RIP无穷计算问题：当链路断开时，路由器无法知道链路断开，会导致路由表不收敛，本质是路由表不加选择的洪泛和使用。解决：1.水平分裂算法：从邻居学到的路由不向邻居节点报告 2.毒性逆转：从邻居学到的路由，将距离设置为无穷大，向邻居节点报告

RIP问题：1.无穷计算 2.收敛慢 3.传输开销 4.计算开销 5.网络动态性 ——只适用中小型网络

链路状态算法OSPF：Dijkstra算法。**OSPF**：1.发现邻居，学习网络地址 2.设置到邻居的成本度量（比如延迟和链路带宽反比） 3.构造链路状态信息 4.将LSP分组发送给其他路由器，判断——如果是新分组，洪泛路由信息，重复分组丢弃，过时分组拒绝。5.计算最短路径，使用Dijkstra算法，更新路由表。

OSPF与RIP比较：1.网络状态信息交换范围：LS全网扩散；2.网络状态信息可靠性：LS使用原始状态信息 3.健壮性：LS各自计算健壮性好 4.收敛速度：LS快。

OSPF优缺点：优点：1.支持CIDR 2.无自环 3.收敛速度快 4.支持多条等值路由 5.支持协议报文认证 6.可使用区域概念优化协议交互 缺点：1.计算量大 2.协议复杂，交互传输压力大 3.基于IP，人工实现“可靠传输” ——适用大中型网络

BGP协议：问题：路由表过于庞大，构成自治系统，内部使用RIP、OSPF;自治系统之间使用BGP。eBGP：从相邻AS获得网络可达信息；iBGP：将网络可达信息传播给AS内路由器。通过TCP连接交换报文，使用路径向量。

BGP路由通告：AS邻居之间通告路由信息，内部使用iBGP通告。

BGP路由选择：路由器可能学到多个同一目的ip的路由，通过不同规则选择路由。**热土豆问题**：选择next-hop开销最小的路由，尽快把分组发出去

BGP路由策略：基于策略决定接受或拒绝路由通告，决定是否向相邻AS通告路由

路由器体系架构

路由器路由控制平面：允许多个路由协议；根据路由优先级，形成核心路由表；将核心路由交给接口版，形成转发表；

路由器数据平面：接受ip报文，更新链路层头，IP头部TTL减1，IP头部校验和更新

路由器数据层：接口卡和交换结构，包括共享内存、共享总线、纵横式三种典型交换结构

NAT技术

NAT：网络地址转换，将内部私有地址转换为外部公有地址，解决IP地址不足问题，提高网络安全性，减少网络管理成本。

优势：1.节省合法地址，减少地址冲突2.灵活连接Internet 3.保护局域网私密性

缺点：1.违反IP结构，路由器需要处理传输层协议 2.违反端到端原则 3.违反最基础协议分层规则 4.不能处理IP报头加密 5.新的网络应用设计者必须考虑

IPv6技术

地址：使用128位地址，分为8个字段，每个字段用冒号分隔，0省略，连续0用::表示，地址分为单播、组播、任播地址

IPv6头部：40字节，包含1.版本号2.流量类型 3.流标签 4.有效负荷长度 5.下一个首部 6.跳数限制 7.源地址 8.目的地址。**与ipv4比较：**1.固定长度，去除首部长度部分 2.去除首部校验和，提升转发速度 3.不能在传输途中分片，只在源端进行分片 4.增加下一个首部字段，可承载多个扩展头，但需要按规定顺序，5.引入流标签，区分服务类型和优先级

特殊地址：1::/128，未指定地址 2::1/128,回环地址，表示节点自己 3.FF00::/8 组播地址 4.FE80::/10 链路本地地址

ND协议：1，邻居请求NS，请求链路层地址 2.邻居通告NA，响应NS 3.路由器请求RS，向路由器请求地址前缀等信息 4.路由器通告RA，路由器向端系统发送地址前缀等信息 5.重定向：重新选择正确的下一跳地址

IPv4到IPv6过渡技术；难点：1.控制域：两个复杂状态机映射，2，地址域：两个海量地址和路由映射 3.纵向涉及结构各层，横向涉及网络结构各单元。**翻译技术：**翻译ipv4报文和ipv6报文，但难以处理内嵌ip地址的应用层，导致分片问题，破坏端到端原则，异构地址难以选址 **隧道技术：**将A协议包装在B协议中传输：可以广泛应用，无需对所有设备升级。缺点是增加网络的复杂性

流量和拥塞控制

流量整形：限制流量速率，平滑流量，减少突发流量对网络的影响

漏桶算法：漏桶排队b个字节，已满则丢弃，每秒流出r个字节，平滑流量。

令牌桶算法：周期性以速率r向令牌桶中增加令牌；输入数据包则消耗令牌，当令牌数量足够，则输出，否则丢弃。

拥塞控制与流量调节：1.抑制包：向源主机发送抑制包，源主机减少流量 2.逐跳抑制包：抑制包对经过的每个路由器都起作用 3.显式拥塞通告ECN：在IP包头中记录数据包是否经历拥塞，接收方回显拥塞信号，发送端降速

RED算法：将路由器队列划分为三个区域：排队区，依概率p丢弃区，丢弃区。

其他

SDN：软件定义网络，将控制平面和数据平面分离，通过控制器集中管理网络，提高网络灵活性和可编程性。

虚电路交换：建立虚电路，每个数据包携带虚电路标识（分组标签），路由器根据标识转发数据包，减少路由表大小，提高转发速度。

MPLS：多协议标签交换，将数据包标记为标签，路由器根据标签转发数据包，并进行标签交换，提高转发速度。（运行在ip协议之下）

传输层

复用和分用：复用：传输层从多个应用收集数据，交给网络层发送；分用：传输层从网络层接收数据，交给正确应用——目的将主机交付扩展到进程交付

UDP

UDP：无连接不可靠，想发就发；功能：增加**复用和分用**的功能，增加**差错检测**的功能；**面向报文**：应用程序必须选择合适大小报文，报文太长IP层需要分片，报文太短相对ip首部太小，效率低下。**使用例子**：

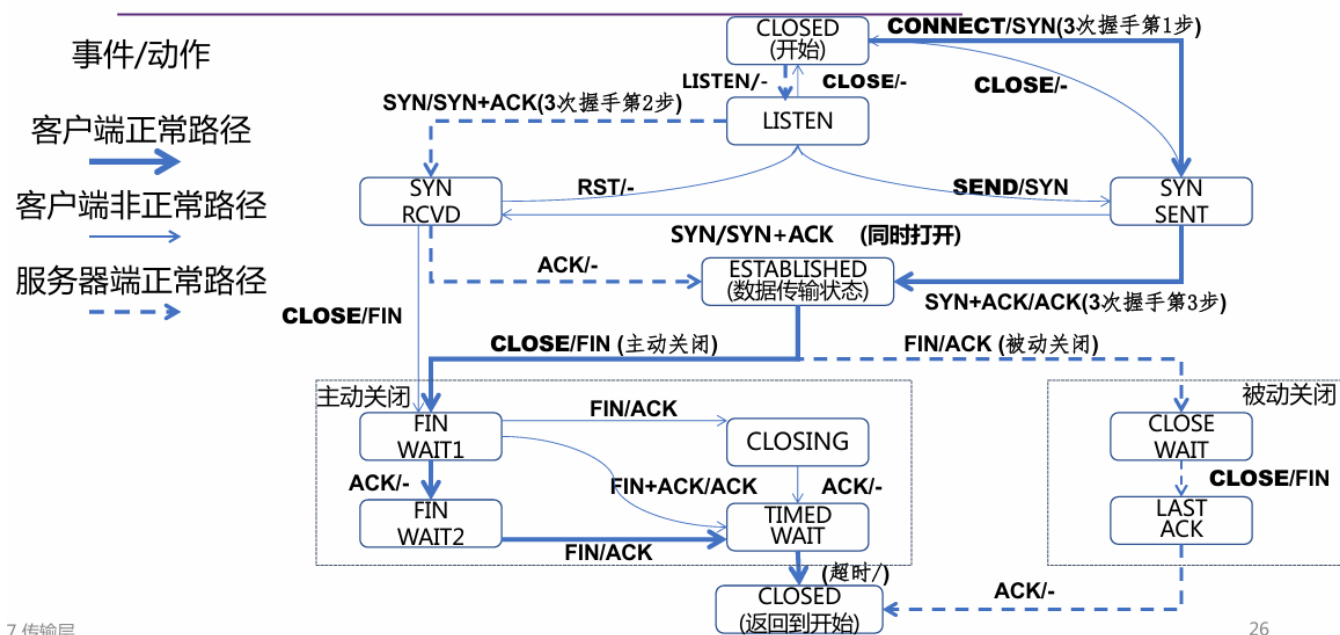
RPC\SNMP\TFTP

UDP头部：取ip头的后12字节作为伪首部，加上2字节的源端口，2字节的目的端口，2字节的长度，2字节的检验和

TCP

TCP：面向连接、可靠的、端到端的、基于字节流的传输协议；**面向连接**：连接只能有两个端点，只能端到端，提供可靠交付和全双工通信；**面向字节流**：TCP把应用程序的数据看成一连串字节流，保证接收方收到的字节流和发送方发出的顺序完全相同。**使用例子**：SMTP\FTP\SSH\HTTP

TCP连接和释放连接（重要！）



7 传输层

26

三次握手：**第一次**：客户端发送SYN=1, seq=x到服务端，自身进入SYN_SENT状态；**第二次**：服务端收到SYN后，发送SYN+ACK回去，其中SYN=ACK=1, seq=y,ack=x+1,自身进入SYN_RCVD状态；**第三次**：客户端收到SYN+ACK后，发送ACK回去，ACK=1,seq=x+1,ack=y+1.自身进入ESTABLISHED状态，服务端收到ACK后也进入ESTABLISHED状态。**非正常情况**：客户端会维护自身想要收到的ack，如果旧的SYN阻塞，客户端发送新的SYN，旧的SYN比新的SYN先到达，客户端收到不正确的ack后，会发送RST，转入LISTEN状态，再次发送SYN进入SYN_SENT状态。如果在SYN_SENT状态下，客户端等待超时，会进入CLOSED状态，如果收到SYN会进入SYN_RCVD状态。

四次挥手：第一次：客户端发送FIN=1, seq=u (等于数据的最后一个字节序号加1) 到服务端, 自身转入FIN_WAIT_1状态; **第二次：**服务端收到后, 发送ACK=1, ack=u+1, seq=v, 自身进入CLOSE_WAIT状态, 此时客户端不能向服务端发送数据, 但是服务端还可以发送数据; **第三次：**客户端收到服务端的ACK后, 转入FIN_WAIT_2状态, 服务器将最后的数据发送完后, 发送FIN=1, ack=u+1, seq=w, 自身进入LAST_ACK状态; **第四次：**客户端收到FIN后, 发送ACK=1, ack=w+1, seq=u+1, 自身进入TIMED_WAIT状态, 等待2MSL后进入CLOSED状态, 服务端收到ACK后进入CLOSED状态。(因此服务端结束的时间比客户端早)

非正常情况：客户端可以通过发送FIN, 直接从SYN_RCVD状态进入FIN_WAIT_1状态; 在FIN_WAIT_1状态下, 如果客户端直接收到FIN+ACK(一起), 可以直接转入TIMED_WAIT状态, 如果只收到FIN, 可以直接发送ACK, 进入CLOSING状态, 等待服务端的ACK后进入TIMED_WAIT状态;

TCP头：20字节, 包含源端口、目的端口、序号、确认号、头部长, ECN, 接收窗口大小, 检验和, TLV等。TCP端最大静载荷为65535-20-20=65495字节。

TCP拥塞控制

拥塞：随着负载的增加, 互联网的有效吞吐量反而下降, 延迟、丢包、抖动等现象增加。拥塞控制的关键在于发送方在合适的时间降低速率, 并且兼顾效率和公平

最大最小公平分配：1.所有流量从速率0开始 2.增加流量, 直到网络中出现新的瓶颈 3.固定瓶颈流量的速率 4.转到步骤2, 查看是否有剩余流量

TCP窗口：发送窗口=min (拥塞窗口, 流控窗口), 其中发送窗口是滑动窗口协议的发送方窗口, 拥塞窗口是根据网络拥塞情况动态调整的窗口, 用于保证发送窗口大小不超过拥塞; 流控窗口是根据接收方的接收能力动态调整的窗口, 用于保证发送窗口大小不超过接收方的接收能力。

发送方案设计思路：1.快速探测, 以便迅速达到最大可用带宽 2.随时间变化: 线性、二次、log等 3.鼓励后来者: 鼓励低速, 惩罚高速

TCP Tache：1.慢启动：发送方按照初始值发送段, 每收到一个ack, cwnd++; cwnd对RTT呈指数增加 (是因为每个RTT会发送多个包, 每确认1字节, cwnd会加1) **2.线性增加：**慢启动超过阈值后即转换到线性增加状态, 每过一个RTT, cwnd++; 发生超时后, 阈值降为当前拥塞窗口的一半, 重新启动。

AIMD原理：加性增加, 乘性减少, 可以快速规避拥塞, 收敛于高效且公平的分配 【这里可能考画图, 注意: 加性增加, 线按照y=x方向, 乘性减少, 方向朝原点】

TCP Reno：快速恢复: 三次重复确认后, 窗口减半

TCP 确认时钟：利用返回的ack的速率作为时钟, 调整发送速率, 可以避免在瓶颈链路产生队列, 使得流量平滑

TCP 自适应超时： $SRTT_1 = aSRTT_0 + (1-a)R$, 其中R为本次测量确认所花时间 $RTTVAR_1 = bRTTVAR_0 + (1-b)|SRTT_1 - R|$; RTTVAR为往返时间变化, 超时RTO=SRTT+4*RTTVAR, 动态调整超时时间

TCP CUBIC：传统的TCP拥塞控制, 拥塞窗口增长过慢, 导致大量的空闲信道被浪费。CUBIC算法用三次函数模拟拥塞窗口, 使用折半查找Wmax。

BBR 思路：传统的拥塞控制以丢包事件为驱动, 实际上此时缓冲区已经被填满, 并不是最优点, 最优点应当是**带宽不再增长, 时延又最低的点**。BtlBw (瓶颈链路带宽, 不会引起排队的最大发送速率), RTprop (最小往返事件) BDP (带宽时延积) = BtlBw * RTprop。BBR试图测量最佳点, 尽量将cwnd收敛到BtlBw, 避免拥塞丢包和排队。

BBR 算法：1.启动阶段：类似于慢启动，指数增加发送速率 2.发现最大带宽：若经过三次窗口增长，发现投递率不再增长，说明已达到最大带宽（已开始排队） 3.排空阶段：指数降低发送速率，将buffer排空 4.瓶颈带宽探测：进入稳定状态后，现在一个RTT内增加发送速率，探测最大带宽，减少发送速率，排空前一个RTT多发的包，后面六个周期重复； 5.时延探测：探测RTTprob，cwnd固定为4个包。

其他

TCP的问题：1.多流复用会导致对头阻塞：TCP不区分多流，一流被阻塞导致所有流被阻塞 2.实现在操作系统内核中：应用和公司难以对TCP进行优化和调整 3.握手时延大：需要多个RTT握手 4.移动互联网支持力度差：网络切换要重新建立连接

QUIC:基于UDP，替代TCP,TLS和部分HTTP的功能，拥塞控制模块化（可以使用TCP拥塞控制算法**） 1.时延优化，数据和密钥同时发送，建立连接只需1RTT（原本TCP加TLS需要3RTT） 2.解决重传歧义：TCP重传包使用和原包相同序号，导致ack判断错误，导致RTO；QUIC的packet number单调递增（包括重传包），避免RTP，ack没有歧义，方便测量RTT 3.IP地址或端口切换无需重新建立连接：TCP连接基于ip地址/端口，而QUIC使用connection ID表示每个连接，IP地址和端口变化不影响连接 4.无队头阻塞的多流复用：QUIC会建立独立的子流，不会导致阻塞 5.QUIC包被加密传输 6.QUIC与操作系统解耦，能快速迭代

MPTCP：传统TCP只能利用一个端口传输数据，MPTCP能将单一数据流切分为子流，同时利用多条路径；设计目标：1.多径带宽聚合，可以聚合不同路径的可用带宽 2.提高传输可靠性：避免单条路径中断导致连接中断 3.支持平滑切换：在不同接入网络间快速平滑切换——基于TCP实现

数据中心网络：问题：1.Cubic队列累积：缓存空间受限，长流占用缓冲区，短流时延大 2.短时突发：多个发送端同时向一个接收端发送数据——设计要求（原则）：1.容忍高突发流量，避免突发流量丢包 2.低时延，需要将排队时延降到最低 3，高吞吐：需要维持高带宽利用率

DCTCP：数据中心TCP 核心思想：1.根据交换机队列长度标记ECN，控制突发流量 2.根据拥塞程度精细调整发送窗口 与TCP相比，DCTCP能够将队列长度长期稳定在低水平。

应用层

应用进程通信方式：1.**C/S方式**，指两个应用进程，客户是服务请求方，服务器是服务提供方；可以面向连接也可以无连接。2.**B/S方式**，将客户软件改为浏览器，采取浏览器请求、服务器响应的工作模式，用户界面由浏览器实现，主要事务逻辑在服务器端实现 特点在于：1/界面同一使用简单 2.易于维护，只需更新服务器端 3.可扩展性好 4.信息共享度高，大部分软件均支持HTML 5浏览器兼容问题：可能一个浏览器开发的界面不完全适配另一种浏览器 3.**p2p**：对等通信，不区分服务请求方和提供方。

服务器进程工作方式：1.循环方式，一次只运行一个服务进程，多个客户请求则按照先后顺序依次作出响应 2.并发方式：同时运行多个服务进程

域名系统DNS

域名结构：采用层次树状结构命名方法，点"."个数至少为一个；分为顶级域名（com等），二级域名（edu，net，等），三级域名，四级域名等。

域名服务器：保存域树的结构和设置信息的服务器程序，称为名字服务器或域名服务器，负责域名解析工作；域名与ip地址可以是一对一、一对多、多对一的关系。分为两大类：1.权威名字服务器：根据本地知识知道本DNS区内容的服务器，可以直接回答询问无需查询其他服务器 2.递归解析器：以递归方式运行的、是用户程序联系域名字服务器的程序

域名解析过程：1.递归查询：域名服务器向下一步域名服务器发出请求，替递归服务器继续查询 2.迭代查询：域名服务器把下一步应查的域名服务器IP地址高速本地域名字服务器，由本地域名字服务器继续查询

DNS报文格式：域名：DNS请求的域名 类型：资源记录的类型 类：地址类型 生产时间：以秒为单位，表示资源记录的生命周期 资源长度 资源数据

DNSSEC：DNS未考虑安全问题，DNSSEC依靠数字签名保证报文的真实性和完整性，域名服务器用自己的私有密钥对资源记录进行签名，解析服务器用公开密钥对应答信息进行验证

WWW和HTTP

万维网：web页面使用HTML文档（解决显示问题），web对象包含各类图像、视频、声音等（解决获取什么信息），对象编制为URL（解决如何获取信息）；

URL：包括协议类型，主机名（服务器），端口，路径和文件名

web对象：静态对象和网页采用HTML等表示，动态对象采用脚本语言（php等）+数据库技术（mysql）以及CGI标准（定义动态文档如何创建、输入数据如何提供、输出结果如何使用）CGI在服务器端执行，脚本程序在浏览器执行

HTTP：超文本传输协议，基于TCP，无状态（不保留之前请求的状态信息）。工作流程：1.建立TCP连接 2.request（GET/POST） 3.responses服务器响应 4.关闭TCP连接

连接：HTTP1.0为非持久连接，每次获取对象都要重新建立TCP连接；HTTP1.1为持久连接，一个TCP连接可以传输多个对象，采用流水线机制。

HTTP后续版本：HTTPS增加SSL/TLS层，提供安全机制；HTTP2.0：增加二进制格式、TCP多路复用等提高带宽利用率，降低延迟

HTTP报文格式：请求报文中：包含请求行（包含方法、URL、协议版本），首部行（如Host，Connection等）

响应报文：状态行（包含协议版本、状态码、状态描述），首部行（如Server，Content-Type等）状态码为三位数字，常见的有（200 OK，404 Not Found，301 Moved Permanently等）

HTTP代理与缓存：共享HTTP使用同一条链路，容易造成拥塞——解决：代理服务器，代表浏览器发出HTTP请求，把请求和响应暂存在本地磁盘中，当相同请求到达时，直接响应，无需请求服务器。

web缓存与代理：目标再次访问缓存在浏览器主机的web页副本，不必访问原始服务器；但需要保证web页副本和原始服务器一致——询问式策略：询问原始服务器是否有更新，客户端在请求中指定缓存时间，包含If-Modified-Since:<date> 如果缓存对象时最新的，无需返回对象，返回304 Not Modified；否则返回新对象，返回200 OK

web安全和隐私：访问安全：在HTTP请求中包含用户名和密码；**Cookie：**用于跟踪用户状态，服务器在响应的首部行里包含set-cookie字段，浏览器在请求的首部行里包含cookie字段,服务器使用cookies保持用户状态，建立数据库；cookie保存在用户主机中。

应用层应用

电子邮件：问题：对方不可能一直在线，需要邮件服务器代理收发；

SMTP协议：设计目标：简单易用，可读性可维护性好，基于TCP实现，是一个ASCII协议；不断重复命令-响应的过程，命令包括HELO，MAIL FROM，RCPT TO，DATA，QUIT等——SMTP是一个推协议，接收方不能使用

POP3协议：基于TCP，分三个阶段：1.认证阶段：用户登录 2.事务处理阶段：用户收取电子邮件并标记为删除 3.更新：将标记为删除的邮件删除

IMAP协议：改进POP3，1.允许用户使用不同计算机同步和处理邮件，IMAP将每个邮件和一个文件夹关联起来，用户可以移动邮件、阅读和删除 2.允许用户代理获取邮件某些部分：可以只读取首部等

webmail：基于web和HTTP

流媒体：连续媒体 特点：1.端到端时延约束 2.时序性约束：需按照一定顺序 3.具有一定容错性：可以丢失部分数据包

MPEG视频压缩：包含3类帧，帧内编码帧（I帧）：包含压缩的静止图片。预测帧（P帧）：与前一帧的逐块差值。双向帧（B帧）：与前后帧的逐块差值。I帧必须周期性的出现在媒体流中

流式存储媒体：由于网络传输的抖动，分组到达时间不确定，如果直接播放可能出现卡顿，需要缓冲区。（如果缓冲区小于低阈值，说明数据即将播完需要加速传输；如果大于高阈值，增大播放时延，可以减慢传输）过程：浏览器先使用HTTP的GET报文接入万维网服务器，服务器返回一个包含URL的元文件；浏览器调用媒体播放器，媒体播放器使用URL向媒体服务器请求音视频文件。

RTSP：RTSP不传送数据，对用户的播放情况进行控制（如暂停、后退等），计入用户的状态。

RTP和RTCP：RTP传输数据，RTCP与RTP配合使用，主要监控和反馈服务质量等

流媒体动态自适应传输：**DASH**：将完整视频拆分为不同片段，每段缓存多个码率，客户端自适应选择合适的码率下载；**ABR**：自适应码率，基于吞吐量或缓冲，决定下一个视频块的码率；**SVC**：使用增强层，增强层可以基于基础层提高码率，优先传输基础层避免卡顿，带宽充足则传输增强层。

CDN：问题提出：需要将大量内容同时分发给大量用户，单个大型服务器会遇到单点故障、拥塞、部分用户远等问题——CDN，内容分发网络，CDN节点缓存数据，服务提供者返回CDN清单给订阅者。DNS重定向将请求调度到附近或负载较轻的CDN服务器。

Telnet：远程登陆协议，使用TCP

FTP：文件传输协议，通过TCP传输文件

SNMP：简单网络管理协议，使用UDP。轮询进行读写，陷阱进行通知

课上补充

5.8:热土豆问题：优先采取域间协议（即尽快把分组转发出自身的自治域）

5.22:TCP可靠性：接收端向上层提交的一定是发送端按顺序发送的

BBR：拥塞丢失：避免进入拥塞；随机丢失：不影响

mptcp：慢速路径造成快速路径突发（因为快速提前到，等待慢速，慢速到后导致窗口巨大前移，导致快速路径突发发送大量），导致快联路带宽上不去，网络效率更低

有了mac为什么还要ip：因为不能用mac在全局进行路由，因为Mac地址随物理地址随时变化，无法形成层次化结构

dns虽然希望可靠，但使用udp因为tcp好几次握手开销太大

http地址是否要4字节对齐：无需，因为tcp面向字节流，所以无需对齐；IP地址在路由硬件处理，字节对齐更方便处理

客户端邮件比webmail好的地方：可以把邮件收在本地

客户端在流媒体缓存：缓存小延迟小，浪费带宽小但抗抖动能力差

解决客户端卡顿：DASH解决客户端根据网络情况要不同清晰度的视频；ABR客户端判断何时要什么码率的视频；SVC解决的是本地已经拿到，低清转为高清

FTP对ipv6影响:ftp在应用层里明确写了ip地址，而ipv6和ipv4的转换网关只能处理网络层的地址，需要应用层网关

反射放大攻击：利用dns服务器不断发送请求，可以通过看源地址是否假来消除