

## 端系统、接入网

端系统 (End Systems): 也称为主机 (Hosts), 包括所有联网设备。端系统是终端设备, 负责生成和接收数据。

接入网 (Access Networks): 连接端系统到最近的路由器的网络部分。

接入网技术包括: DSL (数字用户线路电话线传输数据。光纤接入: 如 FTTH。无线接入: 包括 Wi-Fi、蜂窝网络等。

## 分组交换, 电路交换, 网络结构

**分组交换 (Packet Switching)**: 数据被分割成独立的封包, 通过网络独立传输, 最终在目的地重组。主要优点是高效

$$\text{分组传输时延} = \frac{\text{传输 } L\text{-bit 的分组到链路所需时间}}{L(\text{bit})} = \frac{L(\text{bit})}{R(\text{bit/s})}$$

利用网络资源, 适合突发数据传输。**电路交换 (Circuit Switching)**: 为通信双方建立一个专用的通信通道, 在通话期间独占这个通道。传统电话网络就是基于电路交换的。时延 (Delay): 数据包从源到目的地所需的时间, 包括: 传输时延: 数据包在链路上所花的时间。传播时延: 信号在介质中传播所需的时间。

处理时延: 路由器处理数据包所花的时间。排队时延: 数据包在路由器队列中等待的时间。

**丢包 (Packet Loss)**: 数据包在传输过程中因网络拥塞或其他原因未能到达目的地。高丢包率会显著影响网络性能和用户体验。吞吐量 (Throughput): 单位时间内成功传输的数据量。吞吐量受到带宽、网络拥塞、时延等因素的影响。高吞吐量意味着网络可以更快地传输大量数据。协议层次及其服务模型 OSI 七层模型: 物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

**客户-服务器 (Client-Server)** 客户端: 发起请求并等待响应。处理用户输入、显示结果。

服务器: 接收客户端请求, 处理并返回响应。通常负责数据存储、业务逻辑。

单点故障: 服务器故障会导致整个服务不可用。性能瓶颈: 服务器可能成为性能瓶颈, 限制系统的扩展性。示例: Web 应用 电子邮件: 客户端通过 SMTP、IMAP 或 POP3 协议与邮件服务器通信。

**点对点 (P2P)** 所有节点 (Peer) 都是对等的, 没有固定的客户端和服务器角色, 每个节点既可以作为客户端, 也可以作为服务器。

**Web 和 HTTP** HTTP (超文本传输协议) 工作原理: 基于请求-响应模型, 客户端发送请求, 服务器返回响应。请求方法: 如 GET、POST、PUT、DELETE。状态码: 如 200 (成功)、404 (未找到)、500 (服务器错误)。持久连接: HTTP/1.1 引入持久连接, 允许在同一个 TCP 连接上传输多个请求和响应。HTTPS: HTTP 的安全版本, 利用 TLS/SSL 加密数据传输, 确保数据安全性 and 完整性。**Cookies**: 是由 Web 服务器发送并存储在用户浏览器中的小数据片段, 用于跟踪和保存用户的会话状态和其他信息。Cookies 由 HTTP 协议管理

**Web 缓存器(代理服务器)** 用户设置浏览器:通过缓存器访问 § 浏览器将所有的 HTTP 请求发送到缓存器 • 请求的是缓存器中的对象:缓存器直接返回对象 • 否则: 缓存器向初始服务器请求对象, 然后返回给客户条件 **GET 方法** 目标: 保证缓存器发送的对象副本是最新的

## 2.3 因特网中的电子邮件

SMTP (简单邮件传输协议) 使用 TCP 在客户和服务器之间可靠地传输邮件, 端口为 25

从发送方邮件服务器直接传输到接收方邮件服务器 **传输的三个阶段 • 握手(问候) • 传输报文 • 关闭连接**

工作流程: 客户端通过 SMTP 服务器发送邮件, 服务器之间通过 SMTP 传输邮件。

POP 也使用客户/服务器的工作方式, 在传输层使用 TCP, 端口号为 110。接收方的用户代理上必须运行 POP 客户程序, 而接收方的邮件服务器上则运行 POP 服务器程序。POP 有两种工作方式: “下载并保留”和“下载并删除”。在“下载并保留”方式下, 用户从邮件服务器上读取邮件后, 邮件依然会保存在邮件服务器上, 用户可再次从服务器上读取该邮件; 而使用“下载并删除”方式时, 邮件一旦被读取, 就被从邮件服务器上删除, 用户不能再次从服务器上读取。

另一个邮件接收协议是因特网报文存取协议 (IMAP), 它比 POP 复杂得多, IMAP 为用户提供了创建文件夹、在不同文件夹之间移动邮件及在远程文件夹中查询邮件的命令, 为此 IMAP 服务器维护了会话用户的状态信息。IMAP 的另一特性是允许用户代理只获取报文的某些部分, 例如可以只读取一个报文的首部, 或一个多部分 MIME 报文的一部分。这非常适用于低带宽的情况,

邮件发送协议和读取协议: 邮件发送协议用于用户代理向邮件服务器发送邮件或在邮件服务器之间发送邮件, 通常使用的是 SMTP; 邮件读取协议用于用户代理从邮件服务器读取邮件, 如 POP3。注意, SMTP 采用的是“推”(Push)的通信方式, 即在用户代理向邮件服务器发送邮件及在邮件服务器之间发送邮件时, SMTP 客户端主动将邮件“推”送到 SMTP 服务器端。而 POP3 采用的是“拉”(Pull)的通信方式, 即用户读取邮件时, 用户代理向邮件服务器发出请求, “拉”取用户邮箱中的邮件。

## 2.4 DNS: 因特网的目录服务

DNS: 将人类易读的域名转换为 IP 地址。

结构: 分布式数据库, 包括根域名服务器、顶级域名服务器、权威 DNS 服务器、递归解析器。

DNS 缓存 任何 DNS 服务器一旦学习到某个映射, 就缓存这个映射到本地存储器 一段时间(TTL)后缓存条目超时消失

## 2.5 P2P 应用

特性：节点既是客户端也是服务器，直接交换资源。

对等方加入洪流（Swarm）

初始状态：新加入的对等方（peer）开始时没有任何文件块。它将逐渐从其他对等方那里获取文件块。

追踪器注册：对等方使用种子文件中的追踪器信息注册自己，获取当前洪流中其他对等方的列表。通过追踪器，获取到的对等方列表中会有一部分被选为邻居，建立连接。

文件块获取：对等方从邻居中下载文件块。文件被分割成小块

文件块发送：一旦下载了部分文件块，对等方会开始将这些块上传给其他对等方。对等方会定期评估与当前邻居的连接，基于下载和上传速度，动态调整连接，选择更有效的邻居进行块交换。对等方的加入和离开是动态的，

对等方可能随时上线或下线，一旦对等方下载了整个文件，它有两种选择：**自私地离开**：停止上传并离开洪流。

这会减少网络中的有效上传资源。**利他地留下**：继续作为种子节点，上传文件块给其他对等方，帮助他们完成下载

**视频流**：指通过互联网以流媒体形式传输视频内容，允许用户在数据传输过程中即时观看视频，而无需等待整个视频文件下载完毕

**内容分发网络（CDN）**是由分布在全球各地的服务器网络组成的系统。

**内容缓存**：CDN 服务器存储网站的静态内容的副本。就近访问：CDN 从最近的 CDN 服务器传输给用户。负载均衡：

### 运输层服务

多路复用（Multiplexing）：指在发送端，将不同来源的数据流合并成一个数据流进行传输，以便在接收端进行分解。

多路分解（Demultiplexing）：指在接收端，将合并的数据流分解成各个单独的数据流，交付给相应的应用程序。

无连接运输：UDP 是一种无连接的、不可靠的运输层协议，它提供简单的数据传输服务，但不保证数据的可靠性和顺序传输。丢失、无序地到达应用程序。UDP 发送端和接收端之间不需要握手

发送方：将整个报文段（包括首部）划分成一个个 16 比的字

**校验和**：对报文段中所有 16 比特字的和进行反码运算，求和时遇到的任何溢出都被回卷将检验和的值放入首部的检验和字段

接收方：计算接收到的报文段校验和并检查计算出的检验和是否等于首部检验和字段的值：

**可靠数据传输原理**

rdt1.0: 经完全可靠信道的可靠数据传输

rdt2.0: 具有比特差错信道，检验和用来检测比特错误 -> 发送反馈 -> 重传

rdt2.1: 加数据分组的序号（1 比特）

rdt2.2: 发送方收到重复 ACK 之后，采取与收到 NAK 消息相同的动作：重传当前分组

RD 3.0: 具有比特差错的丢包信道

**序列号和确认**：发送方送序列号，并等待确认消息。接收方收到数据包后，检查序列号，发送确认消息。**超时重传**：如果发送方在一定时间内未收到确认消息，会将未确认的数据包进行重传。

**比特校验**

停-等协议：发送方每次只发送一个数据包，并等待确认后才发送下一个数据包。

流水线协议：**回退 N 步 GBN**：发送方可以发送多个分组（但流水线中未确认的分组数不能超出 N）接收方只发送累积确认。发送方维护最早的已发送但未被确认的分组所使用的定时器 当定时器超时，重新发送所有未确认的分组

选择重传：当定时器超时，**只重传未确认的数据包**

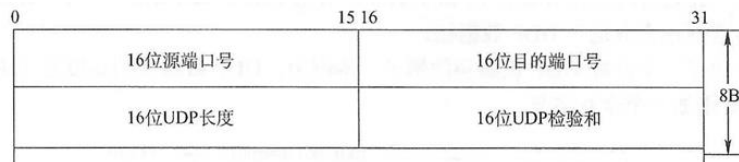
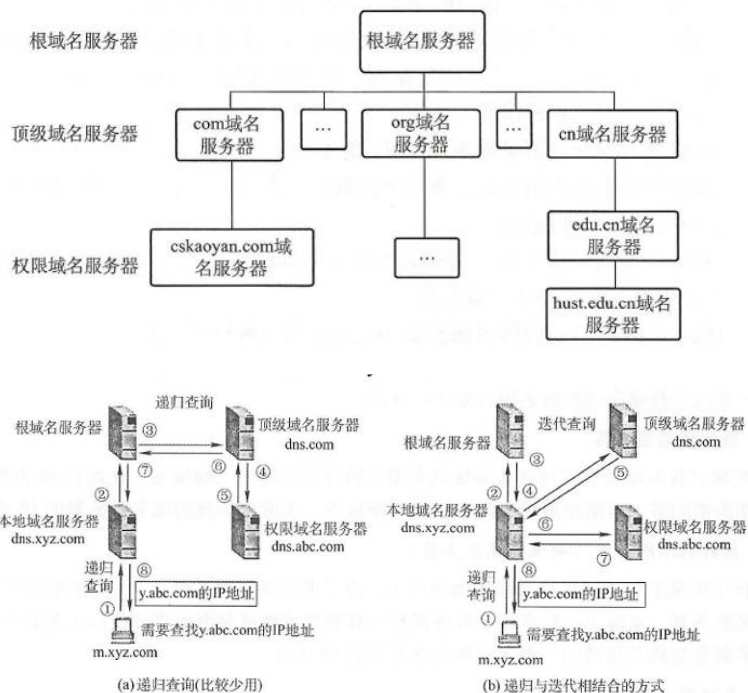
### 面向连接的运输：TCP

TCP 报文段结构 TCP 报文段（Segment）：是 TCP 传输的基本单位，包含头部和数据两部分。序列号和确认号：用于确认数据的传输顺序和确认已接收的数据。

超时重传机制：发送方在定时内未收到确认消息时，会重新发送未确认的数据包。

**流量控制 rwnd：滑动窗口**：发送方可以连续发送多个数据包，接收方发送确认消息，发送方根据收到的确认消息调整窗口大小和重传数据包。

**流水线发送**：TCP 发送方可以连续发送多个报文段而不必等待之前发送的报文段得到确认，这种方式称为流水线发送。发送方会维护一个发送窗口，表示可以发送的未确认的报文段数量。



源端口号	目的端口号
序号	
确认号	
	接收窗口
检验和	紧急数据指针



**累积确认：**TCP 接收方不会为每个收到的报文段发送一个单独的确认，而是只发送一个确认，确认已经成功接收到了一系列报文段中的最后一个报文段。例如，如果接收方收到了连续的报文段 1、2、3、4，则只发送一个确认，表示成功接收到了 4 号报文段。

**单个重传计时器：**TCP 发送方维护一个用于重传的计时器，用来监控报文段的发送情况。当发送一个报文段后，会启动计时器，如果在计时器超时之前没有收到对应的确认，就会认为报文段丢失，并进行重传

**重传的触发事件 超时事件：**当发送方启动计时器后，如果在超时时间内未收到确认，则会触发超时事件，发送方会将未确认的报文段进行重传。**重复 ACK：**接收方收到的序号为 N 的及之后的，但是接收方之前已经收到了序号为 N 的报文段，则接收方会发送一个重复 ACK，通知 N 的报文段已经收到。如果发送方连续收到多个相同的重复 ACK，就表示有报文段丢失，发送方会根据这些重复 ACK 触发快速重传，立即重传丢失的报文段

**TCP 连接管理** **三次握手：**建立 TCP 连接的过程，包括客户端发送 SYN（同步）报文，服务器返回 SYN+ACK（同步和确认）报文，最后客户端发送 ACK（确认）报文。**四次挥手：**关闭 TCP 连接的过程，包括发送方发送 FIN（结束）报文，接收方返回 ACK（确认）报文，接收方发送 FIN 报文，发送方返回 ACK 报文。

**慢启动：**发送方初始化窗口大小为一个报文段，并随着收到确认消息指数倍增，直到达到拥塞窗口的大小。

**拥塞避免：**在达到拥塞窗口大小后，发送方每次只增加一个报文段大小，以便更谨慎地调整窗口大小。

**快速重传：**发送方在连续收到三个重复确认消息时，会立即重传对应的丢失数据包，而不是等待超时重传。

**TCP 拥塞控制：**加性增、乘性减（AIMD）

网络层概述

**数据平面、 控制平面：**负责管理网络中的路由器和交换机，路由表的构建、更新和维护，路由选择算法的执行。

**路由器内部**

**输入端口：**使用首部字段值，使用输入端口内存中的转发表来查找输出端口。**查找给定目标地址的转发表条目时，使用与目标地址匹配的最长地址前缀** 使用三态内容可寻址存储器（TCAM）

**三种类型的交换结构，内存，总线，网络互联**

**输出端口：**缓存溢出、尾丢弃：丢弃到达的数据包优先权：基于优先级进行丢弃或删除随机：随机丢弃或删除。

**IP：互联网协议** **分片：**当数据报长度超过网络的最大传输单元（MTU）时，IP 协议会将数据报进行分片，使其能够在网络中正确传输。

**IPv4 编址** 是 32 位二进制数，通常用点分十进制表示 IPv4 地址由网络部分和主机部分组成

**网络地址转换（NAT）：**使用 NAT 时需要在专用网连接到因特网的路由器上安装 NAT 软件，NAT 路由器至少有一个有效的外部全球地址。使用本地地址的主机和外界通信时，NAT 路由器使用 NAT 转换表将本地地址转换成全球地址，或将全球地址转换成本地地址。NAT 转换表中存放着{本地 P 地址：端口} 到 {全球 P 地址：端口} 的映射。可让多个私有 P 地址映射到同一个全球 IP

**子网：CIDR：**无类别域间路由选择 **地址格式：**a.b.c.d/x，其中 x 是地址子网部分的 # 位

**动态主机配置协议(Dynamic Host Configuration Protocol, DHCP)**用于给主机动态地分配 P 地址，它提供了即插即用的机制，它是基于 UDP 的。

**DHCP 的工作原理如下：**使用客户/服务器方式。需要 P 地址的主机在启动时就向 DHCP 服务器广播发送发现报文，这时该主机就成为 DHCP 客户。本地网络上所有主机都能收到此广播报文，但只有 DHCP 服务器才回答此广播报文。DHCP 服务器先在其数据库中查找该计算机的配置信息。若找到，则返回找到的信息。若找不到，则从服务器的 P 地址池中取一个地址分配给该计算机。DHCP 服务器的回答报文称为提供报文。

**IPv6 地址：**是 128 位二进制数，IPv6 地址被设计为 IPv4 地址空间的扩展，提供了更大的地址空间

通用转发和 SDN

**匹配：**在网络设备中，匹配是指根据数据包的各种属性（如源 IP 地址、目标 IP 地址、端口号等）与预设的规则进行比较，以确定数据包的处理方式。**动作：**匹配成功后，根据预设的动作规则，网络设备会执行相应的动作，如转发、丢弃、修改报头等。

**OpenFlow：**是一种用于 SDN（软件定义网络）的开放式协议，它将网络设备的数据平面和控制平面分离，允许网络管理员通过控制器对网络进行集中管理。当交换机收到数据包时，根据 OpenFlow 规则进行匹配，并执行对应的动作。例如，根据源 IP 地址匹配规则，将数据包转发到指定的输出端口；根据目标 IP 地址匹配规则，将数据包丢弃；或者根据特定协议类型匹配规则，将数据包进行报头修改。



## 路由协议

### 链路状态

### 距离向量

## RIP 使用

### 距离向量

### 算法。

## OSPF 是

### 一种链路

### 状态路由

协议，用于自治系统（AS）内部的路由选择。OSPF 基于 Dijkstra 算法计算最短路径，并通过洪泛算法交换链路状态信息，构建网络拓扑图，并选择最佳路径。两级层次结构：本地区域，骨干网。链路状态仅在域内进行广播每个节点都有详细的区域拓扑；只知道到其他区域的网络的最短路径方向。

§ 区域边界路由器：“汇总”到自己区域内网络的距离，向其他区域边界路由器广播。§ 骨干路由器：运行仅限于骨干网的 OSPF 路由。§ 边界路由器：连接到其他 AS

## ISP 之间的路由：BGP

AS-PATH:前缀通告通过其的 AS 列表、NEXT-HOP:指示特定的内部 AS 路由器到下一跳 AS、

hot potato routing:选择域内成本最低的本地网关。域内成本计算：路由器计算从源节点到本自治系统内所有可能出口点的成本。跳数、带宽、延迟。选择域内成本最低的路径（即最低开销路径）作为转发路径。出口点选择：数据包将通过成本最低的本地网关出口转发出自治系统。该出口点是离源节点最近的网关。

SDN 控制平面是一种网络架构，将网络的控制平面和数据平面分离，通过集中式的控制器对网络进行管理和配置。SDN 通过 OpenFlow 等协议与数据平面交互，实现灵活的网络管理和编程。

ICMP：因特网控制消息协议 是一种用于在 IP 网络中发送控制消息的协议。ICMP 消息用于检测网络中的错误、故障和异常情况，常见的 ICMP 消息包括 ping 请求、traceroute 请求、目的地不可达消息等。

网络管理和 SNMP 是一种用于网络管理的协议，通过 SNMP 代理程序收集和管理网络设备的信息，并向网络管理系统提供状态报告和警报信息，以便对网络进行监控和管理。

错误检测 奇偶校验单比特奇偶校验：检测单个位错误二 维位奇偶校验、循环冗余校验（CRC）

信道划分 MAC 协议：TDMA 时分多址 FDMA：频分多址 码分多址 CDMA 为每个节点分配不同的编码使用其唯一编码对其发送的数据位进行编码 不同节点可以同时传输的奇妙特性

随机接入协议 时隙 ALOHA 时间划分为大小相等的时隙（时间足够传输 1 帧）节点在时隙开始时进行传输

碰撞：节点在每个后续时隙中用概率  $p$  重新传输帧

纯（非时隙）ALOHA 当帧首次到达时立即传输

CSMA：传输前侦听：如果检测到信道繁忙，延迟传输 碰撞：浪费整个数据包传输时间

CSMA/CD（碰撞检测）另一个传输，将中止并发送干扰信号中止后，网卡进入二进制（指数）退避：在第  $m$  次碰撞后，网卡从  $\{0, 1, 2, \dots, 2^m-1\}$  中随机选择  $K$ 。网卡等待  $K \cdot 512$  位时间

轮流 MAC 协议，令牌传递：控制令牌从一个节点按顺序传递到下一个节点

48 位 MAC 地址。使用“本地”将帧从一个接口获取到另一个物理连接的接口（

ARP：地址解析协议 LAN 上的每个 IP 节点（主机、路由器）都有表。某些 LAN 节点 IP/MAC 地址映射

以太网：物理拓扑，总线。星，每个“分支”运行一个（单独的）以太网协议。以太网的 MAC 协议：无时隙 CSMA/CD，带二进制退避 交换机：存储，转发以太网帧。检查传入帧的 MAC 地址，当帧要在段上转发时，有选择地将帧转发到一个或多个传出链路，使用 CSMA/CD 访问段

基于端口的虚拟局域网：流量隔离动态成员资格在虚拟局域网之间进行转发

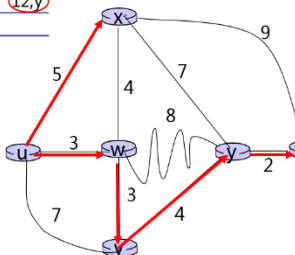
中继端口：在通过多个物理交换机定义的虚拟局域网之间传输帧

多协议标签交换（MPLS）MPLS 路由：目标路径可以基于源地址和目标地址快速重新路由：在链路发生故障时预先计算备份路由由入口路由器（R4）可以使用不同的 MPLS 路由，

Step	N'	D(v)	D(w)	D(x)	D(y)	D(z)
0	u	7,u	3,u	5,u	$\infty$	$\infty$
1	uw	6,w	5,u	11,w	$\infty$	$\infty$
2	uwx	6,w	11,w	14,x	$\infty$	$\infty$
3	uwxv	10,v	14,x	$\infty$	$\infty$	$\infty$
4	uwxvy	12,y	$\infty$	$\infty$	$\infty$	$\infty$
5	uwxvyz	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

### notes:

- 通过跟踪前置节点构造最短路径树
- 关系可以存在（可以任意断开）



node x table	cost to	node y table	cost to	node z table	cost to
	x y z		x y z		x y z
from x	0 2 7	from x	0 2 7	from x	0 2 7
from y	$\infty$ $\infty$ $\infty$	from y	2 0 1	from y	2 0 1
from z	$\infty$ $\infty$ $\infty$	from z	7 1 0	from z	3 1 0

