计算机网络期末考前整理

颜色示例：标题 考纲 考纲各个小点 整理笔记 各类公式/表格/图文 笔者随便写写

一、计算机网络基础

1. 网络的构成：网络边缘、接入网、网络核心

(1)网络边缘：端系统是位于互联网边缘与互联网相连的计算机和其他设备，由各类主机构成

①主机的命名：每个网卡设备有唯一的设备 ID（48 位 MAC 地址）；IP 地址由一组数据构成，根据需要自行配置；主机名是一个字符串。

(2)接入网：各种接入方式在物理介质上有区别

①物理介质：包括引导型介质（固体介质，如铜，光纤，电缆）、非引导型介质（自由传播，如无线电）。

(3)网络核心：分组交换、电路交换、转发与路由、存储转发、排队与丢包、Internet 架构

①分组交换（包交换）：将大消息拆分成多个小组，通信双方以分组为单位（以分组作为数据传输单元）使用存储-转发机制实现数据交互的通信方式。每个分组的首部都含有地址（目标地址、源地址），每个分组在互联网中独立选择传输路径，支持灵活的统计复用（报文分组按需共享带宽）。

• 大部分分组交换设备采用存储&转发传输，收到完整数据包后才能向下一条发送。因此会带来额外的报文的传输延迟。L 位的数据包以 R bps 速率发送，需要 L/R 秒。

• 分组独立传输：主机之间端到端通信可以由多条路径构成。

• 排队延迟与丢包：某段电路上分组到达速率超过链路传输速率时，分组需要在缓冲区内排队，队列填满时后续分组会被丢弃（带来丢包）。

• 缺点：容易引起阻塞，造成排队延迟甚至丢包；无法提供带宽保障。

②电路交换：先呼叫建立连接，实现端到端资源（链路交换资源、交换机的交换能力）的预留，连接建立后物理通道被通信双方独占，资源专用，传输性能好，但传输过程中发生设备故障时连接中断。

③Internet 架构：端系统通过本地网络提供商接入 Internet，本地网络提供商也需要互相连接。

• 如何连接：每个本地网络提供商接入一个全局 ISP，全局 ISP 之间互相连接，可能出现区域 ISP 连接 Access ISP 与 Global ISP。

2. 网络连接：面向连接，无连接，性能指标

(1)面向连接：传输数据之前先建立连接

(2)无连接：数据传输前不需事先建立连接

(3)性能指标：比特率、带宽、包转发率、时延、往返时间 RTT、时延带宽积、吞吐量、丢包率、利用率、时延抖动

①比特率（数据率）：主机在数字信道上传送数据的速率，单位是 bps。

②带宽：单位时间内网络中的某信道所能通过的“最高数据率”，单位是 bps。

③包转发率：交换机或路由器等网络设备以包为单位的转发速率。

④吞吐量：单位时间内通过某个网络（或信道、接口）的数据量，单位是 bps。

⑤有效吞吐量：单位时间内，目的地正确接收到的有用信息的数目。

⑥利用率：信道利用率指出某信道有百分之几的时间是被利用的，网络利用率则是全网络的信道利用率的加权平均值。

⑦丢包率：所丢失数据包的数量占所发送数据包的比率。

⑧时延：是指数据（一个报文或分组）从网络（或链路）的一端传送到另一端所需的时间，也称为延迟。

• 传输时延：数据从结点进入到传输媒体所需要的时间，又称为发送时延。

• 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：主机或路由器在收到分组时，为处理分组（例如分析首部、提取数据、差错检验或查找路由）所花费的时间。
• 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。

$$总时延 = 传输时延 + 传播时延 + 处理时延 + 排队时延$$

⑨往返时延 RTT：从发送方发送数据开始，到发送方收到来自接收方的确认，经历的总时间。
⑩时延带宽积：按比特计数的链路长度。

$$时延带宽积 = 传播时延 \times 带宽$$

⑪其他指标：可靠性、完整性、隐私性、可审计性。

3. 网络协议基本概念、分层模型、各层实现位置

(1)基本概念：为进行网络中的数据交换而建立的规则、标准或约定，其三要素为语法、语义和时序
(2)分层模型：将网络分为多个层次。每层提供特定功能与服务保障，层与层之间有接口调用，同层实体通过该层协议进行对话，屏蔽各层实现细节
①OSI 参考模型：从下到上依次为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。支持无连接和面向连接。
• 物理层：定义如何在信道上传输 0/1。
• 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（进程到进程）。
• 会话层：利用传输层提供的服务，在应用程序之间建立和维持会话，并使会话获得同步。
• 表示层：关注所传递信息的语法和语义，管理数据的表示方法，传输的数据结构。
• 应用层：通过应用层协议，提供应用程序便捷的网络服务调用。
②TCP/IP 参考模型：从下到上依次为网络接口层、互联网层、传输层、应用层。只支持无连接。
• 网络接口层：描述了为满足无连接的互联网络层需求，单跳链路必须具备的功能。
• 互联网层：通过多跳连接，将数据包独立传输至目的地，并定义了数据包格式和协议。
• 传输层：允许源主机与目标主机上的对等实体，进行端到端的数据传输（TCP，UDP）。
• 应用层：传输层之上的所有高层协议（DNS、HTTP、FTP、SMTP…）。
③本课程：从下到上物理层、数据链路层、网络层、传输层、应用层。

二、应用层

1. 基本概念：客户端-服务器模式、P2P 模式

(1)基本概念：应用层运行在端系统的各类软件中，负责与网络进行通信，提供各类网络服务并屏蔽底层网络细节
①应用程序以两种方式组织：客户/服务器模式和对等（P2P）模式。
(2)客户端-服务器模式
①客户和服务器是指通信中所涉及的 2 个应用进程，客户是服务请求方，服务器是服务提供方。
②C/S 方式可以是面向连接的，也可以是无连接的。面向连接时，连接一旦建立就是双向的。
③服务器进程工作方式分为循环式（TCP,UDP）和并发式（TCP）。
• 循环式：一个服务进程在同一时间只能向一个客户进程提供服务。
• 并发式：面向连接的 TCP 服务进程通常都工作在并发服务方式，服务进程在同一时间可

同时向多个客户进程提供服务。具体流程为：服务器监听一个（熟知）服务端口，如 HTTP 的 80 端口，主服务进程在熟知端口等待客户进程发出的请求。一旦收到客户的请求，就创建一个从属服务进程，并指明从属服务进程使用临时套接字与该客户建立 TCP 连接，然后主服务进程继续在原来的熟知端口等待向其他客户提供服务。

• 由于 UDP 只有一个套接字，无法被多个从属进程同时访问，所以 UDP 只能采用循环方式，不能采用并发方式。

(3)P2P 模式

①对等方式是指两个进程在通信时并不区分服务的请求方和服务的提供方。

• 只要两个主机都运行 P2P 软件，它们就可以进行平等、对等的通信。
• 双方都可以下载对方存储在硬盘中的共享文档，如果权限允许的话。
• P2P 方式从本质上看仍然是使用了 C/S 方式，但强调的是通信过程中的对等，这时每一个 P2P 进程既是客户同时也是服务器。

②P2P 实体的特征：

• 不需要总是在线。
• 实体可以随时进入与退出。
• 任意两个实体之间可以直接通信。
• 易于扩展。

2. WWW：Web 对象、URL、静态 Web 对象、动态 Web 对象

(1)WWW：world wide web 的简称，指万维网

①构成：

• Web 对象（网页，多媒体资源，动态对象与服务），通过 URLs 定位。
• HTTP 服务器和客户端。
• 服务器与客户端之间执行的 HTTP 协议。

②在 WWW 体系结构与协议中，服务器包括 Web 页面和 Web 对象，对象用 URL 编址。客户端发出请求、接收响应、解释 HTML 文档并显示。

(2)Web 对象：包括静态对象与静态网页、动态对象与动态网页、超链接

(3)URL：统一资源定位器，格式为：协议类型://主机名:端口//路径和文件名

(4)静态 Web 对象

①文本，表格，图片，图像和视频等多媒体类型的信息（实现语言：标记语言，如：HTML，XML，PHP 等）。

②字体、颜色和布局等风格类型的信息（实现语言：层叠样式表 CSS）。

(5)动态 Web 对象

①交互信息，比如，用户注册信息、登录信息等（实现：PHP/JSP 等语言，MySQL 等数据库）。

3. HTTP：服务过程、报文格式、缓存、Cookie

(1)HTTP：超文本传输协议

①在传输层通常使用 TCP 协议，缺省使用 TCP 的 80 端口。

②HTTP 为无状态协议，服务器端不保留之前请求的状态信息（效率低但简单）。

(2)服务过程

①HTTP1.0：缺省为非持久连接，每次获取对象都需要三次握手建立连接，完成后都要关闭连接。每次连接都需要经历慢启动过程。

②HTTP1.1：缺省为持久连接，在相同的 TCP 连接上，服务器响应后保持连接；支持流水线机制；经历较少的慢启动过程，减少往返时间（降低响应时间）。

③HTTP2：允许多路复用；压缩；预测资源请求；流量控制。

(3)报文格式

①请求报文：由三个部分组成，即开始行（请求行）、首部行和实体主体。

• 开始行包括：方法、URL、版本、CRLF（回车换行）。

• GET：请求读取由 URL 所标志的信息。

• POST：向 URL 提交数据（例如，参数、注释）。

②响应报文：由三个部分组成，即开始行（状态行）、首部行和实体主体。

• 开始行包括：版本、状态码、短语、CRLF。

• 状态码都是三位数字，1xx 表示通知信息，2xx 表示成功，3xx 表示重定向，4xx 表示客户的差错，5xx 表示服务器的差错。

• 典型状态码：

200：*OK*

301：*Moved Permanently*

400：*Bad Request*

404：*Not Found*

505：*HTTP Version Not Supported*

(4)缓存

①浏览器缓存：在浏览器主机保存用户访问过的服务器 Web 页副本；再次访问该页，不必从服务器再次传输，提高访问效率。

• 目标：再次访问缓存在浏览器主机中的 Web 页副本，不必从原始服务器读取。

• 确保缓存页与原始服务器一致：请求、检查过期、条件 GET、服务器返回未改变/相应、浏览器缓存并响应。

②Web 代理服务器缓存：设置用户浏览器，通过代理服务器进行 Web 访问。

• 目标：代理服务器缓存已访问过的 Web 页副本，满足用户浏览器从代理服务器提取 Web 页，尽量减少原始服务器参与。

• 浏览器将所有的 HTTP 请求发送到代理服务器，如果缓存中有被请求的对象，则直接返回对象；否则，代理服务器向原始服务器请求对象，再将对象返回给客户端。

• 询问式策略：通过特殊的关键字头询问原始服务器，Web 副本对应的原始 Web 页是否已更新。

• 原始服务器明确指令限制缓存某些 Web 页，服务器返回 Web 页时，带一个 no-cache 禁止缓存，需要授权访问的 Web 页也限制缓存。

(5)Cookie：HTTP 无状态协议，服务器用 cookies 保持用户状态

①HTTP 在响应的首部行里使用一个关键字头 set-cookie：选择的 cookie 号具有唯一性；后继的 HTTP 请求中使用服务器响应分配的 cookie。

②Cookie 文件保存在用户的主机中，内容是服务器返回的一些附加信息，由用户主机中的浏览器管理；Web 服务器建立后端数据库，记录用户信息，cookie 作为关键字。

③Cookie 包含五个字段：域、路径、内容、过期、安全。

④Cookie 是双刃剑，能分析用户喜好，向用户进行个性化推荐，也能跟踪用户网络浏览痕迹，泄露用户隐私。

4. DNS：域名服务器、域名解析过程、DNS 安全

(1)域名系统服务（DNS）

①向所有需要域名解析的应用提供服务，负责将可读性好的域名映射成 IP 地址。

②可以基于域名查询 IP 地址，也可以基于 IP 地址反向查询域名。

③提供的是网络层的功能，但以应用层的技术实现。

(2)域名服务器（名字服务器）

①保存关于域树的结构和设置信息，负责域名解析服务。
②每个名字服务器保存相邻域名服务器信息，域名解析过程对用户透明。
③根据对应域的层次，域名服务器可以分为根/顶级域/二级域/三级域名字服务器，三级域及以下的域名服务器称为本地域名服务器。

(3)域名解析过程

①某一应用进程需要进行域名解析时，该应用进程将域名放入 DNS 请求报文（UDP，目标端口号为 53）发送给本地 DNS 服务器，本地 DNS 服务器得到查询结果后将对应 IP 地址放在应答报文中返回给应用程序。
②分为递归查询和迭代查询两种方式。本地 DNS 先尝试递归查询，随后尝试迭代查询。
• 递归查询：查询报文在本地域名服务器中逐级向上递归查询。查询对象是本地 DNS 服务器。
• 迭代查询：查询请求到达本地域最上一层名字服务器后仍无法解析，先求助于根服务器，逐步迭代查询。查询对象是根服务器、顶级域名字服务器、二级域名字服务器。
③DNS 报文分为三部分：基础结构（报文首部）、问题、资源记录，报文类型分为查询请求和查询相应。

(4)DNS 安全

①DNS 协议的脆弱性：
• 几乎所有 DNS 流量都是基于 UDP 明文传输的。
• DNS 的资源记录未加上任何的认证和加密措施，DNS 的用户隐私容易被泄露。
②DNSSEC：依靠数字签名保证 DNS 应答报文的真实性和完整性。
• 域名服务器用私有密钥签名，解析服务器使用域名服务器的公钥对应答信息进行验证。
③域名系统隐私：
• 源 IP 地址的保密性：在递归服务器上源 IP 地址是用户主机的 IP 地址，在域名服务器上源 IP 地址是发出查询请求的服务器的 IP 地址。
• 解析过程的保密性：QNAME 字段提供了用户的操作信息，可能包含敏感信息。
④DNS 敏感数据泄露的主要途径：
• 通信链路窃听。
• 服务器收集。

5. 电子邮件：电子邮件系统的组成、各个协议的功能

(1)组成：包括用户代理、传输代理和简单邮件传输协议 SMTP

①用户代理（邮件客户端）：
• 用户通过用户代理和电子邮件系统交互。
• 实现功能：显示入境邮件信息，邮件处置，自动处理邮件，发送邮件，邮件列表……
②传输代理（邮件服务器）：
• 将邮件从发件人中继给收件人。
• 采用 SMTP 协议。

(2)各个协议的功能：SMTP、POP3、IMAP、Webmail

①SMTP 协议
• 利用 TCP 可靠传递邮件，使用端口 25。
• 直接投递，发送端直接到接收端。
• 三个阶段：连接建立，邮件传送，连接关闭。
• 命令：ASCII 码字符串，响应：状态码 + 短语。
• 邮件格式：包括 RFC 5322 和 MIME。
• 最终交付（邮件访问）协议：需要 POP3、IMAP、Webmail（HTTP）等协议。（SMTP 是

一个推协议）
• 不足：不包括认证；传输 ASCII 而不是二进制数据；邮件以明文形式出现。
②POP3 协议：一个非常简单的最终交付协议。
• 三个阶段：认证、事务处理、更新。
• 使用客户/服务器工作方式。
③IMAP 协议：用于最终交付的主要协议，是 POP3 的改进版。
• 邮件服务器运行侦听端口 143 的 IMAP 服务器，用户代理运行 IMAP 客户端。
• 允许用户在不同的地方使用不同的计算机随时上网阅读处理邮件。
• 将每个邮件与一个文件夹联系起来，提供了在远程文件夹中查询邮件的命令。
• 维护了用户状态信息。
④Webmail：基于 Web 的电子邮件。
• 使用 Web 作为界面，用户与远程邮箱的通信通过 HTTP 进行。

6. P2P：基本概念、BitTorrent、分布式哈希表

(1)基本概念：每个实体都是一个对等结点

①服务：采用去中心化的连接与传输，不需要长期在线服务器，任意对等结点可以直接建立连接、随时加入或退出、变换 IP 地址。
②优势：避免单一中心化结点造成的性能瓶颈。
③核心问题：Peer 索引。
• 中心化索引：由中心化服务器帮助检索。问题：单点故障、性能瓶颈。
• 解决方案：洪泛技术。每个 peer 独立建立索引，peer 之间形成一个图，查找资源时递归向邻居查询，查询到结果后沿查询路径返回查询发起者。
• 混合索引：介于中心化索引和洪泛索引之间，存在超级结点，普通结点和超级结点间使用中心化索引，超级结点间使用去中心化的洪泛索引。

(2)BitTorrent：基于 P2P 思想的文件分发的协议

①所有正在交换某个文件的 peer 组成一个 torrent，中心化的跟踪器维护一个正在主动上传和下载该内容的所有其他对等用户列表，对等方可以通过跟踪器找到其他对等方。
②工作流程：
• 文件被划分成 256Kb 大小的块，torrent 中的结点发送或接收文件。
• 当 peer 加入 torrent 时向追踪器注册，随后从其他 peers 逐渐获取文件块。
• peers 彼此交换各自拥有的块清单，也可与其他 peers 互相交换各自知道的 peers。
• 结点动态加入退出。
③peer 结点优化策略：选择罕见块下载，匹配同级别结点……

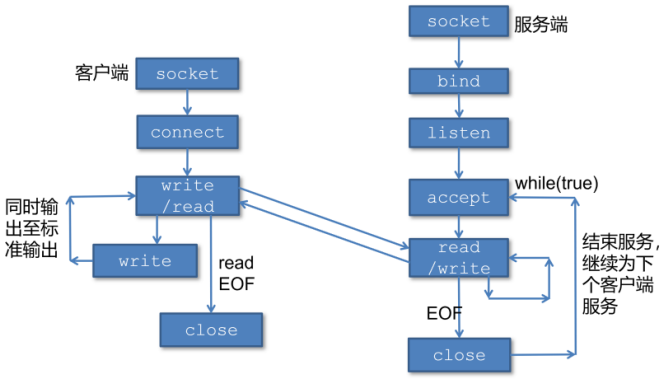
(3)分布式哈希表：不需要中心化追踪器，就能查询每个 key 在哪个 peers 上

①基本思想：
• 对所有 peers 地址、key 计算哈希值。
• 哈希值取模后排列在一个圆环上。
• 每个 key 由圆环上顺时针方向的下一个 peer 存储，查询失败时继续沿顺时针方向查询。
②peer 加入：通知前驱、后继 peer 更改邻居关系，从后续 peer 迁移数据。
③peer 退出：
• 正常退出：通知前驱、后继 peer 更改邻居关系，数据迁移至后续 peer。
• 异常退出：发送数据丢失，解决方案为每份数据在多个 peer 上备份，异常退出时恢复。

7. socket 编程

①进程标识：包括主机地址 + 该进程关联的端口号。
②有关函数：bind 通过绑定监听描述符声明对某个端口的占用，listen 监听某个端口上是否

有连接请求，accept 从队列取出连接请求，若接受则分配新的描述符。
③基于 UDP 的套接字通信流程：无连接，直接用文件描述符收发数据。
④基于 TCP 的套接字通信流程：面向连接，在收发数据前还需要建立连接。



8. 流媒体：三种服务模式，RTSP、RTP、RTCP 功能

(1)三种服务模式：媒体点播、媒体直播、实时交互

①媒体点播：浏览器从服务器下载并播放流媒体文件，发送端以恒定速率产生分组。
- 网络传输的后果：由于抖动特性，分组到达接收端时变成非恒定速率，会出现卡顿。
- 如何应对：缓存后以恒定速率播放，能够一定程度上消除时延抖动，但增加了时延。
- 客户端缓冲区：播放本地缓冲区内容，基于两个阈值控制（低阈值标记、高阈值标记）。

②直播与实时音视频
- 使用协议：RTSP、RTP/RTCP。
- 传输方式：实时类应用主要使用 UDP（效率高），失败则转化为 TCP。

(2)RTSP、RTP、RTCP 功能

①实时流式协议 RTSP
- 不传送数据，是一个多媒体播放控制协议。
- 对用户下载的实时数据的播放情况进行控制。

②实时传输协议 RTP
- 为实时应用提供端到端的数据传输，但不提供任何服务质量的保证。
- 是一个协议框架，只包含实时应用的一些共同功能。
- 不对多媒体数据块做任何处理。

③实时传输控制协议 RTCP
- 与 RTP 配合使用的控制协议。
- 主要功能：服务质量的监视与反馈、媒体间的同步、播组中成员的标识。

三、传输层

1. 传输层基本概念：套接字、端口号、复用与分用

(1)基本概念：位于应用层和网络层之间，基于网络层提供的服务，向不同主机上的应用程序提供通信服务

①从网络的角度：屏蔽底层网络的复杂性，应用层只允许在终端上，不需要为网络设备编写程序。
②从应用程序的角度：屏蔽应用的复杂性，提供进程之间本地通信的抽象。
③传输层可以通过差错恢复、重排序等手段提供可靠、按序的交付服务，但无法提供延迟保证、带宽保证等服务。

(2)套接字：是应用层和传输层的接口，也是应用程序和网络之间的 API

(3)端口号：是套接字标识的一部分。每个套接字在本地关联一个 16 比特的端口号

①分类

• 熟知端口：0~1023，公共域协议使用。

• 注册端口：1024~49151，需要向 IANA 注册才能使用。

• 动态和/或私有端口：49152~65535，一般程序使用。

②报文段中，携带源端口号和目的端口号。

③分配：自动分配（客户端常用，49152~65535），指定端口号（服务器常用，0~1023）。

(4)复用与分用

①复用：发送方传输层将套接字标识置于报文段中交给网络层，传输端从多个套接字收集数据交给网络层发送。

②分用：接收方传输层根据报文段中的套接字标识，将报文段交付到正确的套接字。

③UDP 分用：使用 IP 地址和端口号的二元组进行标识。

④TCP 分用：同时使用一个监听套接字和多个连接套接字。服务器在监听套接字上等待客户的连接请求，收到客户的连接请求后创建连接套接字与客户通信。使用源 IP 地址、目的 IP 地址、源端口号、目的端口号四元组进行标识。

2. UDP

①提供的服务：对网络层接口进行最简单的封装（网络层 + 多路复用与分解），进行报文完整性检查（可选）。

②需要实现的功能：复用和分用，报文检错。

③报文段结构：源端口，目的端口，报文长度，校验和，数据段。

• 校验和的计算：将数据划分为一系列 16bit 整数，所有整数相加（溢出的 1 加到低位 16bit 部分），最终结果取反。计算时需要包括伪头、UDP 头和数据三部分，校验和段计算时视为 0。

• 接收方对 UDP 报文（包括校验和）以及伪头求和，若结果为 0xffff，则认为没有错误。

④通常实现：无发送缓冲区，有接收缓冲区。

⑤为什么需要 UDP

• 可以尽可能快地发送报文，无建立连接的延迟，不限制发送速率。

• 报头开销小，协议处理简单。

• 适合：容忍丢包但对延迟敏感的应用（流媒体）、以单次请求/响应为主的应用（DNS）。

• 若要求 UDP 提供可靠传输，由应用层实现可靠性。

3. 一般性可靠传输及其性能优化

(1)一般性可靠传输

①可靠性机制要求：发送方、接收方各自都应执行相应操作，协同保证可靠性。

②完美信道（rdt1.0）：帧不丢失或受损，发送方和接收方始终处于就绪状态，能生成/处理无穷多数据。

• 不处理流量控制或纠错工作，接近于无确认的无连接服务。

• 发送方：获得数据，封装，发送。

• 接收方：等待数据，解封装，完成接收。

③有错但不丢包信道（rdt2.0，rdt2.1，rdt2.2）：某些比特会 0-1 翻转，但能被校验和检测。

• 解决方案：重传出错数据包，统称自动重传请求。

• rdt2.0：若出错返回 NAK，否则返回 ACK，收到 NAK 后重传。缺陷在于 ACK 或 NAK 也可能出错。

• rdt2.1：发送方直接重传。发送方为每个数据包加入序列号 seq（取值 0/1），检查 ACK 或 NAK 是否正确；接收方在收到数据包 seq 与等待的不一致（重复数据包）时忽略。

• rdt2.2：在 ACK 中加上最近成功接收的 seq，发送方就可以判断发送是否成功，NAK 就不需要了。当 ACK 出错或 seq 不等时重传。

④有错且丢包信道（rdt3.0）：不仅会出错还会丢失数据包，永远不可能到达接收方。

• 解决方案：新增计时器，经过一段时间没有收到确认时重传。

• ACK 丢失或延时过长会导致接收方收到重复数据，需要判重。

⑤rdt2.0，rdt2.1，rdt2.2，rdt3.0 均采用停等式协议，信道利用率低且只能有一个没有被确认的帧在发送中。

(2)一般可靠性方案的性能优化

①流水线传输：允许发送方在没收到确认前连续发送多个帧。

• 要求：增大序号 seq 的范围，发送方要保存所有未确认的数据包，处理多个数据包的丢失、损坏、超长延迟。

• 机制：回退 *N*（GBN），选择重传（SR）。

• 滑动窗口机制：限制最多有 *N* 个未确认的数据包，从而降低开销，可以循环重复使用有限的帧序号。

②回退 N 算法

• 设计思想：接收方收到出错帧或乱序帧时丢弃所有后续帧，不为这些帧发送确认。发送端超时后重传所有未确认的帧。

• 发送方保存所有未确认数据包（构成队列，维护 seq 上下界），接收方无需保存数据包，只需记录下一个期望收到的 seq。

• 优点：减轻接收端负担。缺点：重传包数量大，增加发送端与信道负担。

③选择重传算法

• 设计思想：接收方对每个数据包独立确认，某一包的 ACK 错误或超时时只重传该数据包。

• 要点：发送端为每个包维护计数器，接收端需要缓存已收到的数据包。

• 优点：减少重传数量。缺点：接收端需要缓存，发送端需要逐包维护计时器。

• 可以证明，窗口大小不能超过 seq 取值空间的一半。

4. TCP 可靠传输

①基本机制：流水线传输。

• TCP 为字节建立序号而非报文。

• 接收方与发送方收到字节后，将 seq 更新作为新的 ACK，将 ACK 作为新的 seq 发送给对方。

②TCP 发送端事件处理

• 收到应用数据，创建并发送 TCP 报文段，若没有定时器在运行则启动定时器。

• 超时，重传包含最小序号的、未确认的报文段并重启定时器。

• 收到 ACK，若确认序号大于基序号则推进发送窗口，若还有未确认的报文段则启动计时器，否则中止定时器。

• 只使用一个定时器，只重发第一个未确认报文，避免了大量重发；利用流水式发送和累计确认，避免重发某些丢失了 ACK 的报文段。

• TCP 快速重传：当发送方收到对同一序号的三次重复确认时立即重发。

③TCP 接收端事件处理

• 收到期待的报文段时发送更新的确认序号，否则重复当前确认序号。

• 允许接收端推迟确认以减小通信量。

• TCP 协议规定：推迟确认的时间至多为 500ms，且至少每隔一个报文段使用正常方式进行确认。

• 只有在收到期待的报文段，且之前的所有报文段均已发送过确认时可以推迟发送，其余情况都必须立刻发送确认。

④差错恢复机制：GBN 与 SR 的混合体，定时器与 GBN 类似，超时重传与 SR 类似。

⑤总结

• TCP 可靠传输的设计要点：流水式发送报文段，缓存失序的报文段，采用累计确认，支队最早未确认的报文段使用一个重传定时器，超时后只重传包含最小序号的、未确认的报文段。

• 超时值的确定：基于 RTT 估计超时值 + 定时器补偿策略。

• 测量 RTT：不对重传的报文段测量 RTT，不连续使用推迟确认。

• 快速重传：收到三次重复确认后重发报文段。

• 延迟确认优化：考虑对 RTT 的估计。

5. TCP 报文结构：包括源/目的端口号、序列号、ACK 号、TCP 头长度、接收端还可以接收的字节数、校验和、选项等。

6. TCP 连接建立与关闭

(1)连接建立

①需要确定的两件事：双方都同意建立连接，初始化连接参数。

②三次握手建立连接

• 客户 TCP 发送 SYN 报文段，SYN = 1，ACK = 0（给出客户选择的起始序号，不包含数据）。

• 服务器 TCP 发送 SYNACK 报文段，SYN = ACK = 1（服务器端分配缓存和变量，给出服务器选择的起始序号，确认用户的起始序号，不包含数据）。

• 客户 TCP 发送 ACK 报文段，SYN = 0，ACK = 1（客户端分配缓存和变量，确认服务器的起始序号，可能包含数据）。

③如何选择 TCP 起始序号：使用时钟，每隔 Δt 时间计数器加 1，以最低 32 位作为起始序号（避免新旧连接序号重叠，确保序号回绕时间远大于分组在网络中的最长寿命）。

(2)连接关闭

①服务器、客户端都可以主动关闭连接（设置 FIN = 1），发送 FIN 后不能再发送数据。

②四次握手过程，两端各自发送 FIN，各自确认对方的 FIN。

③关闭连接时的异常情况

• 丢包：四次握手均可能丢包，解决方式为重传。

• 客户端或服务器端下线，另一端仍不断重试，解决方案为失败若干次后放弃连接或等待重新建立连接。

7. TCP 流量控制：非零窗口、糊涂窗口

(1)流量控制：发送端通过调节发送速率，不使接收端缓存溢出

①UDP 不保证交付，不需要流量控制。

②TCP 流量控制：接收方将接收缓存中的可用空间（RcvWindow）放在报头中，向发送方通告接收缓存的可用空间。发送方限制未确认字节数不超过接收窗口大小。接收方通告接收窗口为 0 时，发送方必须停止发送。

(2)非零窗口：接收方窗口由 0 变为非 0 时，接收方应通告增大的窗口

①发送方收到零窗口通告后，可以发送零窗口探测报文段，接收方可以发送包含接收窗口的响应报文段。

②实现

• 发送方收到零窗口通告时启动坚持定时器。

• 定时器超时后发送端发送一个零窗口探测报文段，序号为上一个段中最后一个字节的序

号。
• 接收端在响应报文段中通告当前接收窗口的大小。
• 若发送端仍收到零窗口通告，重新启动坚持定时器。
(3)糊涂窗口：发送数据很快、消费速度很慢时，接收方不断发送微小窗口通告，发送方不断发送很小的数据分组，导致大量带宽的浪费
①接收方启发式策略
• 解决方案：通告零窗口后，仅当窗口大小显著增加（达到缓存空间的一半或一个 MSS）时才发送更新的窗口通告。
• TCP 做法：窗口大小不满足时推迟发送确认（最多 500ms，且至少每隔一个报文段使用正常方式进行确认），仅当窗口大小满足时通告新的窗口大小。
②发送方启发式策略
• 解决方案：积聚足够多的数据再发送，避免发送太短的报文段。
• Nagle 算法：当发送方数据量达到一个 MSS 或上次传输的确认到来时用一个 TCP 段将缓存的字节全部发走。
8. TCP 拥塞控制：慢启动、拥塞避免、快速恢复、Tahoe 算法与 Reno 算法区别、AIMD 吞吐量与公平性
(1)拥塞控制：发送端通告调节发送速率，使之不超过网络的处理能力
①需要解决的问题
• 发送方如何感知网络拥塞。
• 发送方采用什么机制限制发送速率。
• 发送方感知到网络拥塞后，采取什么策略调节发送速率。
②感知拥塞：利用丢包事件（包括：重传定时器超时、收到三个重复 ACK）。
③限制发送速率：使用拥塞窗口 cwnd 限制。
• 始终要求 LastByteSent – LastByteACKed ≤ cwnd。
• 发送速率 rate = cwnd / RTT。
④感知到拥塞后进行调节：总体思路为 AIMD（乘性减，加性增）。
• 乘性减：感知到丢包后 cwnd 大小减半（不能小于一个 MSS）。目的：迅速减小发送速率，缓解拥塞。
• 加性增：若无丢包，每经过一个 RTT 将 cwnd 增大一个 MSS，直至检测到丢包。目的：缓慢增大发送速率，避免震荡。
⑤实际拥塞策略由慢启动、拥塞避免、快速恢复组成，近似实现了 AIMD。
(2)慢启动：低 cwnd 时的策略
①基本思想：在新建连接上指数增大 cwnd 至检测到丢包，随后终止慢启动。
②策略：每经过一个 RTT，cwnd 加倍。
③具体实施：每收到一个 ACK 段，cwnd 增加一个 MSS。
④特点：以一个很低的速率开始，按指数增大发送速率。
(3)拥塞避免：高 cwnd 时的策略
①基本思想：当 cwnd 增大到一定程度时距离拥塞并不遥远，继续指数增长易导致拥塞。
②解决方案：将指数增长改为线性增长。
③区分慢启动与拥塞避免：维护 ssthresh 阈值变量，cwnd < ssthresh 时为慢启动，cwnd > ssthresh 时为拥塞避免。
(4)Tahoe 算法与 Reno 算法
①Tahoe 算法：不区分收到三个 ACK 与超时，两种情况都重新开始慢启动。
②Reno 算法：

• 收到三个重复 ACK：进入快速恢复阶段，ssthresh 降为 cwnd / 2，cwnd 降为当前 cwnd / 2 + 3，采用新机制调节 cwnd 直至再次进入慢启动或拥塞避免阶段。
• 超时：重新开始慢启动，ssthresh 降为 cwnd / 2，cwnd 降为 1，使用慢启动增大 cwnd 至门限。

(5)快速恢复阶段：收到三个重复 ACK 时进入
①继续收到该重复 ACK：每次将 cwnd 增加一个 MSS。
②收到新 ACK：降低 cwnd 至 ssthresh，进入拥塞避免阶段。
③超时；重新开始慢启动。

(6)AIMD 吞吐量与公平性
①吞吐量：忽略慢启动阶段，只考虑拥塞避免、快速恢复阶段，近似 AIMD 过程。
• 若发生丢包时拥塞窗口大小为 W，则平均吞吐量约为 $0.75W$/RTT。
②公平性：TCP 是公平的。
• 两个连接按照相同的速率增大拥塞窗口，相同的速率将拥塞窗口减半，得到斜率为 1 的直线。
• 允许应用建立多条并行 TCP 连接时，不能保证公平分配。

9. 新型传输层技术：

(1)BIC 与 CUBIC
①BIC 算法：利用二分查找搜索合适的 cwnd。
• 思想：初始化 $W_{max} = W_1$（丢包），$W_{min} = W_2$（未丢包），进行 ACK 驱动查找。每次收到 ACK 时将 W_{min} 设置为 W_{max} 和 W_{min} 的中点。
• 若 cwnd 度过 W_{max} 仍未丢包，说明 W_{max} 还没有达到，BIC 按照逼近 W_{max} 的路径倒回去。
②CUBIC 算法
• 思想：窗口增长函数仅仅取决于当前距离上次丢包经过的时间 t，完全独立于网络的时延 RTT。
• 用三次函数拟合：

$$W_{cubic} = C(t - K)^3 + W_{max}，\quad K = (W_{max}\beta / C)^{1/3}.$$

其中 C，β 都是常数。

(2)BBR
①算法优化目标（与传统 TCP 差异）：传统 TCP 认为 BDP（瓶颈链路装满时整个网络管道里的数据量）+ BtlneckBufSize（缓冲区大小）是最优窗口大小，而 BBR 的目标是认为 BDP 是最优窗口大小。

(3)DCTCP
①发送端任务：每个 RTT 更新一次发送窗口，cwnd 修改为原来的$(1 - \alpha / 2)$倍。
②接收端任务：仅当 ECN 报文出现或消失时才立即发送 ACK，否则采取 Delay ACK 策略，
③交换机任务：当队列长度超过 K 时，为随后到来的包标记 ECN（队列长度超过阈值）。

(4)QUIC
①网络体系架构：在传统的传输架构中，TCP 提供数据传输服务，TLS 为数据进行加密，HTTP 定义如何发起请求-响应请求。QUIC 替代 TCP、TLS 和部分 HTTP 功能，实现在用户态中，底层基于 UDP 实现。
②优势
• 解决了重传歧义：ACK 没有歧义。
• IP 地址/端口切换无需重新建立连接：支持 IP/端口切换。
• 解决了队头阻塞问题：减少了不必要的等待。
• 易于部署和更新：QUIC 包加密传输保护了用户的隐私；在用户态实现，可以快速迭代。

四、网络层

1. 网络层基本概念

①基本功能：使主机-主机间数据传输可达

• 发送端：将传输层数据单元封装在数据包中。

• 接收端：解析接收的数据包中，取出传输层数据单元，交付给传输层。

• 数据传输采用多跳传输，网络层功能存在每台主机和路由器中。

②Internet：采用无连接的数据报服务

• 无连接服务：不需要提前建立连接。

• 数据报服务：向上只提供简单灵活无连接的、尽最大努力交付的数据报服务，不提供服务质量的承诺。

• 尽力而为交付：不提供端到端的可靠传输服务，可能出现丢包、乱序、错误。

③关键功能：转发和路由

• 转发：将数据报从路由器的输入接口转送到正确的输出接口，核心是转发函数。

• 路由：选择数据报从源端到目的端的路径，核心是路由算法与协议。

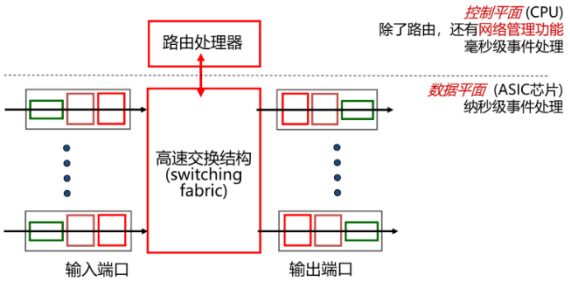
•转发与路由功能将网络层进一步划分为数据平面和控制平面。数据平面实现转发功能（单个交换机上局部功能），控制平面实现路由功能（全网计算，通过传统路由算法和软件定义网络 SDN 实现）。

• 传统的单个交换机上的数据平面：每个交换机执行转发函数，根据输入端口和分组的目的地址转发到输出端口。

• 传统的单个交换机上的控制平面：每个交换机运行一个路由算法模块。

• SDN 中心化控制平面：一个远程控制器与各个路由器交互。

2. 路由器架构：包括控制平面和数据平面



(1)控制平面：包括路由和网络管理功能

①路由器可以同时运行多个路由协议，也可以只使用静态路由和直连路由。

②路由管理根据路由优先级选择最佳路由，形成核心路由表。

③控制层将核心路由表下发到数据层形成转发表（FIB）。

④若存在多个去往同一目的 IP 的不同类型路由，路由器根据优先级选择，优先选择优先级数值小的。

(2)数据平面

①路由器中，链路层解封装，IP 头部校验，获取报文目的 IP 地址。

②查询转发表成功时，链路层封装更新，IP 头部 TTL 减 1 并重新计算校验和。

③数据报在不同硬件单元的处理：

• 输入端接口卡进行物理层处理和链路层解封装，查询转发表并通过交换结果将报文排队发往目的接口卡。

• 交换结构将数据报从输入接口卡发往输出接口卡。

• 输出端接口卡从交换结构接收报文，进行链路层封装和物理层处理，并从输出端口发送

报文。
④输入端口：采用去中心化的数据交换。
• 每个数据端口执行独立计算任务。输入端口基于报文头部字段，在转发表中查找对应的输出端口。
• 优势：每个端口独立工作，达到线速。
• 转发策略：基于目的地址的转发或通用转发。
• 基于目的地址的转发：使用最长前缀匹配，匹配最长的地址前缀。在 TCAM 中，每个 bit 支持三个值：0，1，don't care。
• 缓冲队列：报文到达速度超过交换结构速度时，报文在输入端口队列中缓存。会出现排头阻塞现象，后续报文对应的输出端口空闲时也需要等待。
⑤交换结构：将报文从输入端口的缓冲队列传输到正确的输出端口。
• 三种典型的交换结构：共享内存、共享总线、纵横式。
• 共享内存：交换结构中只有一块内存，路由处理器负责路由与转发功能。性能瓶颈在于内存拷贝。
• 共享总线：数据报从输入端口直接到达输出端口，无需处理器干预。性能瓶颈在于总线每次只能广播一个报文。
• 纵横式 Crossbar：使用 $2N$ 条总线连接 N 个输入端口和 N 个输出端口，控制点可以开启或闭合。
⑥输出端口：关注缓冲队列与队列调度。
• 缓冲队列：交换结构数据超过发送数据时。
• 调度机制：包括 FIFO、优先级调度、轮转调度、加权公平队列等。

调度算法请参见操作系统

3. IPv4 报文格式：分片

(1)报文格式：由首部和数据两部分组成

①首部包括版本、首部长度、区分长度、总长度、标识、标志、片偏移、生存时间、协议、首部校验和、源地址、目的地址、选项、填充等。

(2)数据报分片：包括分片策略与重组策略

①分片策略：
• 允许途中分片：根据下一跳链路的 MTU（最大传输单元）实施分片。
• 不允许途中分片：发出的数据报长度小于路径 MTU（路径 MTU 发现机制）。
②重组策略：
• 途中重组实施难度太大，互联网一般采用目的端重组。
• 重组所需信息：原始数据报编号、分片偏移量、是否收集所有分片。
③IPv4 分组在传输途中可以多次分片，只在目的 IP 对应的目的端系统进行重组，分片、重组字段在基本 IP 头部（标识、标志、片偏移）。

4. IP 地址、路由器转发

(1)IP 地址：网络上每一台主机（路由器）的每一个接口都会分配一个全球唯一的 32 位标识符

①IP 地址划分为固定的类，每一类由两个字段组成。网络号相同的连续 IP 地址空间称为地址的前缀，或网络前缀。
②网络接口：连接主机/交换机与物理链路之间的模块。
• 路由器有多个接口，每个主机通常由 1~2 个接口。
• IP 地址按照接口分配，接口之间通过链路层技术互相连接。
③子网划分：IP 地址按照点分十进制记法书写，每一段取值范围为 0~255。

• IP 地址 = 网络地址 + 主机号。
• 子网掩码与 IP 地址一一对应，是 32bit 二进制数，置 1 为网络位，置 0 为主机位。
• 子网：由相同网络地址的网络接口。子网内的接口不需要网络层路由就可以通过链路层技术进行数据传输。
• 子网由接口组成，与主机/路由器等设备无关。
• 主机位全 0 标识子网地址，主机位全 1 表示广播地址。若主机位有 n 位，则子网拥有主机数量为 $2^n - 2$。

④无类域间路由（CIDR）：网络地址可以是任意长度
• 表示：将 32 位 IP 地址划分为前后两部分，采用斜线记法，在 IP 地址后加上斜线“/”。
• 地址聚合：CIDR 子网内地址可以进一步划分为多个子网，对外只暴露一个 CIDR 网络地址。
• 最长前缀匹配：IP 地址与 IP 前缀匹配时，总是选取子网掩码最长的匹配项。

(2)路由器转发（IP 包转发）：分为直接交付和间接交付。

①直接交付：与目的主机在同一个 IP 子网内，使用目的 MAC 地址。通过 ARP 协议获取子网内 IP-MAC 映射。

②间接交付：与目的主机不在同一个 IP 子网内，使用目的 IP 地址。

5. ARP、DHCP、NAT、ICMP

(1)ARP

①功能：A 已知 B 的 IP 地址，需要获得 B 的 MAC 地址（物理地址）。

②交互流程：
• 若 A 的 ARP 表中缓存有 B 的 IP 地址、MAC 地址的映射关系则直接获取。
• 若 A 的 ARP 表中未缓存 B 的 IP 地址、MAC 地址的映射关系，则 A 广播包含 B 的 IP 地址的 ARP query 分组；B 接收到 ARP 分组后将自己的 MAC 地址发送给 A，A 在 ARP 表中缓存映射关系。

(2)DHCP

①功能：主机加入 IP 网络时，允许主机从 DHCP 服务器动态获取 IP 地址。

②交互流程：
• DHCP 客户从 UDP 端口 68 以广播形式向服务器发送发现报文（DHCP DISCOVER）。
• DHCP 服务器广播或单播发出提供报文（DHCP OFFER）。
• DHCP 客户从多个 DHCP 服务器中选择一个，并向其以广播形式发送 DHCP 请求报文（DHCP REQUEST）。
• 被选择的 DHCP 服务器广播或单播发送确认报文（DHCP ACK）。

(3)NAT

①功能：将私有（保留）地址转化为公有 IP 地址的转换技术，用于解决 IPv4 地址不足的问题。
• 私有 IP 地址：包括 A、B、C 类地址。

②工作机制：
• 所有从本地网络发出的报文具有相同的单一源 IP 地址：138.76.29.7。
• 报文的源地址（发出时）或目的地址（接收时）都属于 10.0.0.0/24 的地址。

③交互流程：
• 主机发送数据到 NAT 路由器。
• NAT 路由器改变源 IP 与源端口，更新 NAT 转换表。
• 收到返回数据。
• NAT 路由器修改返回地址与端口。

(4)ICMP

①实现 PING：测试两个主机的连通性。

• 使用 ICMP 回送请求与回送回答报文。

②实现 Traceroute：知道整个路径上路由器的地址。

• 源向目的地发送一系列 UDP 段（不可能的端口号），第 n 个 TTL = n。

• 第 n 个数据报到达第 n 个路由器时，路由器丢弃数据报，并向源发送一个 ICMP 报文，报文的源 IP 地址就是路由器的 IP 地址。

• 路由器根据 ICMP 报文计算 RTT。

• 停止条件：UDP 段到达目的地主机，目的地返回 ICMP“端口不可达分组”，源得到该 ICMP。

6. 网络路由：距离向量、链路状态、层次路由、BGP 路由通告与路由策略。

(1)网络路由：为每对发送主机、接收主机找到好的网络传输路径。

①核心问题：给定两个结点，最小代价的路径是什么。

②路由算法分类：全局/区中心，静态/动态。

(2)距离向量算法（DV，Bellman-Ford 算法）

①距离向量：$d_x(y)$定义为从结点 x 到结点 y 的最短路径的代价，其计算公式为

$$d_x(y) = \min\{c(x, v) + d_v(y): v \text{ 为 } x \text{ 的邻居}\} \quad \cdots\cdots\text{Bellman-Ford 方程}$$

②分布式更新：包括数据的分布式与计算的分布式。

• 数据的分布式：将单机算法维护的信息分散到各个结点。每个结点 x 维护：到达每个邻居结点 v 的开销 $c(x, v)$；距离向量 $D_x = \{d_x(y): y \in V\}$，它表示该结点 x 到网络中所有其他结点 y 的最小代价的估计值；每个邻居结点 v 的距离向量 $D_v = \{d_v(y): y \in V\}$。

• 计算的分布式：每个结点 x 重复以下计算：

每个结点向邻居发送它自己到某些节点的距离向量；

节点 x 收到来自邻居 y 的新 DV 估计时，更新所保存的 y 的 DV 信息；

使用 B-F 方程更新自己的 DV。

③特点：异步、迭代（包括：本地链路代价改变，邻居 DV 更新）、分布式（仅当 DV 信息变化时通知其他结点）。

④链路状态代价变小：“好消息迅速传播”。

⑤链路状态代价变大：“坏消息传播慢”（又称“无穷计数”问题）。

• 毒性逆转：若某结点 a 经过 b 到达 c，则 a 通知 b：$d_a(c) = +\infty$（无法解决一般的无穷计数问题）。

(3)链路状态算法（LS，Dijkstra 算法）

①步骤：

• 发现邻居，了解他们的网络地址。

• 设置到每个邻居的成本度量。

• 构造链路状态分组（LSP），分组中包含最新的链路信息。

• 将分组发送给其他路由器。

• 计算到其他路由器的最短路径（Dijkstra 算法）。

Dijkstra 算法请参见数据结构与算法

②与距离向量算法的比较：

• 消息数量：对于 n 个结点，$|E|$条链路，LS 需要发送 $O(n|E|)$条消息，DV 取决于收敛速度，最多 $O(n|E|)$条消息。

• 收敛速度：LS 在消息传播完毕后每个节点需要 $O(n^2)$或者 $O(n\log n)$时间完成计算，DV 不确定，可能存在无穷计数问题。

• 可靠性：对于路由器或链路故障，LS 影响小（传播链路开销，每个结点独立计算），DV 影响大（传播计算后的结果，取决于邻居的计算结果）。

(4)层次路由

①产生原因：

• 地址分配随机，难以进行高效的地址聚合。

• 每个网络的管理员不希望每个路由器都干涉本网络内部的地址分配问题。

②基本思路：

• 互联网由大量不同的网络互连，每个管理机构控制的网络是自治的。

• 自治系统内部使用内部网关路由协议。

• 自治系统之间使用外部网关路由协议。

(5)BGP 路径通告与路由策略

①路径通告：路径包括两个重要属性：

• AS 路径：IP 前缀经过的所有 AS 号。

• 下一跳：说明路由信息对应的下一跳 IP 地址。

②路由策略：可以基于策略决定接收/拒绝接收到的路由通告，也可以基于策略决定是否向其他相邻 AS 通告路径信息。

7. 广播、组播、选播

(1)广播：源主机同时给全部目标地址发送同一个数据包

①实现：泛洪，每个进入数据包发送到除了进入线路外的每条出去线路。

②泛洪爆炸的解决方案：受控制的泛洪。

• 序号控制泛洪：收到数据包时，若序号表中曾经有该数据包的记录则直接丢弃，否则在序号表中记录并转发。

• 逆向路径转发：收到数据包时，若是从最佳路径来的则转发，否则丢弃。

(2)组播：源主机给网络中的一部分目标用户发送数据包

①基本步骤：

• 确定组成员。路由器通过 TGMP 获得该网段的组播组成员。

• 生成树。根据组的分布建立最佳生成树。

(3)选播：将数据包传送给一个组中最近的一个组成员

①典型应用：DNS

• 未指定 DNS 服务器情况下，用户始终连接到最接近的服务器。

• 易于配置管理，在每个位置配置一个 IP 地址。

• 可以提供一定程度的高可用性。服务器故障时用户请求无缝转发至下一个最接近的 DNS 实例。

• 可以水平缩放，服务器负载过重时可以在另一个位置部署另一台服务器承担一部分重载服务器请求。

8. 软件定义网络：流量工程、SDN 思想与基本概念、数据平面、控制平面

(1)流量工程：根据对传输流量的预测，规划流量的传输路径

①目的：提高带宽利用率，避免拥塞。

②通常需要线性规划、网络流等算法。

③传统网络控制平面的缺陷：

• 无法控制流量的路径。

• 无法对流量进行划分。

• 无法让路由器区别对待两组流量。

(2)SDN 思想与基本概念

①基本思想：
• 数据平面与控制平面分离。
• 中心化控制平面。
②技术要点：
•“基于流表”的通用转发（如 OpenFlow 接口）；
• 控制平面与数据平面分离。
• 中心化控制平面。
• 应用程序可编程开发。
③优势：
• 数据平面：与控制平面分离，提供开放接口，允许对网络设备进行“编程”。
• 控制平面：中心化控制器，全局网络视角，更好的网络管理（加快链路状态传播/路由收敛速度，支持流量工程：全局更优的数据选择，避免路由器故障导致网络配置错误，让网络编程更加容易）。

(3)数据平面

①组成：包括 SDN 交换机、SDN 控制器（网络操作系统）、SDN 应用程序。
②OpenFlow：一种流表结构
• 每个路由器维护一张流表，流表由控制器计算后写入每个路由器。
• 流表项由四部分组成：模式、动作、优先级、计数器。
•“匹配-动作”可以统一实现各类网络设备。

(4)控制平面

①架构：包括接口模块、分布式数据库、通信模块。
• 接口模块提供网络抽象。
• 分布式数据库管理全网分布式状态。
• 通信模块与交换机交互。
②协议：OpenFlow
• 用于控制器和支持 OpenFlow 的交换机交互。
• 使用 TCP 传输消息，可选择加密传输。
• OpenFlow 定义了两类消息：控制器到交换机，交换机到控制器。

9. IPv6：协议、与 IPv4 兼容

(1)协议：用于替代 IPv4 的下一代协议。

①初始动机是应付 32 位地址耗尽的问题，增加地址空间，后续动机是简化头部，加快处理与转发，提升服务质量。
②地址：长度为 128 位。
• 地址表示方法：冒分十六进制 *x*:*x*:*x*:*x*:*x*:*x*:*x*:*x*。
• 简化：每个 *x* 前面的 0 可以省略，连续的值为 0 的 *x* 表示为::，且::只能出现一次。
③数据报文：头部固定 40 字节长度，数据传输不允许在途中分片。

(2)与 IPv4 兼容：隧道技术

①隧道技术：两个相同类型网络的设备跨越中间异构类型网络进行通信。其将一种网络的数据包作为另一种网络的数据载荷进行封装。
②隧道类型：
• 应用层隧道（SSH 隧道；HTTPS 隧道）。
• 传输层隧道（TCP 隧道；UDP 隧道）。
• 网络层隧道（4 in 6；6 in 4；通用路由封装隧道）。
• 链路层隧道（L2TP 协议，链路层隧道；PPTP 协议，点对点隧道）。

10. 网络服务质量：数据包调度、流量工程、漏桶算法、令牌桶算法、综合服务、区分服务

(1)数据包调度：路由器输出端口决定把缓冲区中的哪些数据包发送到输出链路上

①算法：FCFS、公平队列算法、加权公平队列算法、优先级调度。

(2)流量工程：根据对传输流量的预测规划流量的传输路径

①目的：提高带宽利用率，避免拥塞。

②算法：通常需要线性规划、网络流等。

(3)流量整形：限制流出某一网络的某一链接的流量与突发，使这类报文以均匀的速度向外发送

①漏桶算法：平衡网络上的突发流量，整形突发流量以为网络提供稳定流量。

• 原理：数据包被放在缓冲区（漏桶）中，漏桶最多可以容纳 b 个字节（多余丢弃）。数据包从漏桶中以常量速率 r 注入网络。

②令牌桶算法：允许突发数据的发送，但控制流速。

• 原理：周期性以速率 r 向令牌桶中增加令牌，令牌不断增多（超过上限丢弃），输入数据包会消耗桶中令牌。输入数据包经过令牌桶时，若桶中令牌数量满足数据包需求则输出，否则丢弃。

(4)综合服务

①特点：

•（面向连接）基于资源预留协议 RSVP，在主机间建立传输流的连接。

•（预留资源）逐节点建立或拆除流的状态和资源预留状态，根据 QoS 进行路由。

②要求：需要所有的路由器支持综合服务，在控制路径上处理每个流的消息，维护每个流的路径状态和资源预留状态，在路径上执行基于流的分类、调度、管理。

③现实中难以实现。

(5)区分服务

①要求：

• 在 IP 报头的 8 位区分服务字段（DS 字段）中使用 6 位区分服务码点（DSCP）进行分组分类，指明分组的类型。

• 路由节点在转发这种包时，只需根据不同的 DSCP 选择相应的调度和转发服务即可。

②问题：DS 字段的使用无法控制。

11. 虚电路与 MPLS：转发过程

(1)虚电路

①表示逻辑连接，建立在 Internet 分组交换之上，不是真正建立物理连接。

②转发过程：转发决策基于分组标签，即虚电路号。

(2)MPLS

①全称是多协议标签交换，设计初衷是为了提升查找速度。

②基本操作：加标签、标签交换、去标签。

③转发过程：为每个转发等价类（一组同样方式处理的报文）分配唯一的标签。

④应用：VPN、流量工程。

12. VPN：背景、原理

(1)背景：许多机构希望建立专有网络

①连接该机构各部分网络。

②与 Internet 隔离的路由器、链路以及提供 DNS、DHCP 等基础服务的代价昂贵。

(2)原理：利用公用网络架设专用网络的远程访问计数。通过隧道技术在公用网络上模拟出一条点到点的逻辑专线，从而达到安全数据传输的目的

五、数据链路层

1. 基本概念

①在协议栈中的位置：向下利用物理层提供的位流服务，向上向网络层提供明确的服务接口。

②作用：在物理相连的两个结点间进行数据传输。

- 主机与路由器统称为结点或站点。
- 连接两个结点间的物理链路称为链路或信道。
- 两层的数据包称为帧。

③特点：差异性

- 不同链路上采用不同协议。
- 不同的链路协议提供不同的服务。

④实现位置：通常包括硬件、固件、软件部分。每一台主机与网络内部设备都需要实现链路层。

⑤提供的服务：成帧、差错控制、流量控制、无确认无连接服务、有确认无连接服务、有确认有连接服务。

2. 成帧：字节计数、带字节填充的定界符、带比特填充的定界符、物理层编码违例

(1)字节计数法：用一个字节标识该帧的大小，后面紧跟着该帧的所有字节。

(2)带字节填充的定界符：用一个特殊的字节（FLAG）区分前后两个不同的帧。

①字节填充：在有效载荷中出现定界符，此时在出现的定界符前插入转义字节 ESC。因此，对于有效载荷中的所有 ESC、FLAG，在前面插入 ESC。

②接收方的处理：逐个检查收到的每一个字节。

- 收到 ESC：后一字节无条件成为有效载荷，不检查。
- 收到 FLAG：帧的边界，当前帧结束。

③问题：效率不高。

(3)带比特填充的定界符：两个 0 比特之间连续 6 个 1 比特作为定界符。

①字节填充：在有效载荷中出现连续 5 个 1 比特，则直接插入一个 0 比特。

②接收方的处理：若出现连续 5 个 1 比特：

- 下一比特为 0，则为有效载荷，直接丢弃 0 比特。
- 下一比特为 1，则连同后一比特的 0 构成定界符，当前帧结束。

(4)物理编码为例

①核心思想：选择的定界符不会在数据部分出现。

②4B/5B 编码方案：将 4 比特数据映射成 5 比特，剩余的一半码字未使用，可以用作帧定界符。

3. 差错控制：基本概念、奇偶校验、校验和、CRC、海明码、海明距离、纠正单比特错误所需校验位下界

(1)基本概念

①链路层存在的问题：信道的噪声导致数据传输问题（差错、丢失、乱序、重复）。

②解决方案：差错检测与纠正、确认重传。

- 确认：接收方校验数据，并给发送方应答，放之差错。
- 定时器：发送方启动定时器，防止丢失。
- 顺序号：接收方检查序号，防止乱序、重复。

③目标：保证一定差错检测和纠错能力的前提下减少冗余信息量。

- 考虑的问题：信道的特征和传输需求、冗余信息的计算方法、携带的冗余信息量、计算复杂度……

• 主要策略：检错码、纠错码。
• 典型检错码：奇偶校验、校验和、CRC、海明码。

(2)奇偶校验

①1 位奇偶检验：增加 1 位校验位，可以检测奇数位错误。
②二维奇偶检验：将比特组织成二维数组，可以检测并纠错单个比特错误。

(3)校验和：见 UDP 部分

(4)循环冗余校验（CRC）

①计算方法：
• 原始数据 D 是 k 位二进制串。
• 为产生 n 位 CRC 校验码，事先选定 $n+1$ 位二进制串 G（生成多项式，提前商定），其首位为 1.
• 将 D 乘 2^n 后模 G，余数 R（n 位）即为 CRC 校验码，填充在 D 的后 n 位。
• 在模 G 的过程中，加法不进位，减法不借位，只根据当前首位是否为 1 决定是否有商。
②接收端校验：收到 D 和 R 后除以 G，若没有余数则通过校验。

关于有限域的更多知识，欢迎选修抽象代数

(5)海明码

①以奇偶校验为基础，找到出错位置并提供一位纠错能力。
②缺省为偶校验。
③发送方：对于 2^n-n-1 个比特，取 n 个位作为校验位（即一共 2^n-1 个比特）放在 2 的幂次位，每个校验位取值等于所有该位为 1 的比特位的值求和（在 $GF(2)$意义下）。
④接收方收到码字后，顺次检查校验位是否正确。若正确则传输没有差错。若出现了一位差错，将所有出错的校验位求和即可得到出错位。

关于海明码的更多知识，欢迎选修信息论

(6)纠正单比特错误所需校验位下界

对于 m 个信息位，r 个校验位，需要纠正单比特错误。
对 2^m 个有效信息中的任何一个，有 $m+r$ 个与之距离为 1 的无效码字。
这些码字一共有 $(m+r+1)2^m$ 个，要求不能超过 2^{m+r}，所以

$$m+r+1\le 2^r.$$

解以上不等式即可。

4. 多路访问控制

(1)产生原因

①广播信道面临的问题：多个站点同时请求占用信道，产生冲突。
②解决方法：介质的多路访问控制，在多路访问信道商确定下一个使用者（信道分配）。
③目标：性能、公平、去中心化、简单易实现。
④主要方法：信道划分、随机接入、轮流协议。

(2)信道划分

①TDMA：划分出等长时间片依次分给各站点，未使用的时间片处于空闲。
②FDMA：将信道分为多个频段分配给各个站点，未使用的频段处于空闲。

关于排队系统的更多知识，欢迎选修应用随机过程

(3)随机访问

①ALOHO 类：
• 纯 ALOHA 协议：原理：想发就发。特点：随时可能冲突，冲突导致破坏的帧需要重传。
• 分隙 ALOHA 协议：假设所有帧大小一样，时间划分为等长时间槽（刚好传输一个帧），站点只能在时间槽开始发起传输。站点需要发送帧时，在下一个时间槽开始进行传输。若

发生冲突则以概率 p 在下一时间槽重传。

②CSMA 类：若信道忙则推迟发送。

• 非持续式 CSMA：经侦听介质空闲立刻发送；介质忙时等待一个随机分布的时间继续侦听。

• 1-持续式 CSMA：经侦听介质空闲立刻发送；介质忙则持续侦听；发送推迟一个时间单元后继续侦听。

• p-持续式 CSMA：经侦听介质空闲以 p 的概率立刻发送，以 $1-p$ 的概率推迟一个时间单元再处理；介质忙则持续侦听；发送推迟一个时间单元后继续侦听。

• CSMA/CD：1-持续式 + 冲突检测。经侦听介质空闲立刻发送；介质忙则持续侦听，一旦空闲立刻发送；传输过程中进行冲突检测，发生冲突时立刻终止传输，等待一个随机分布的时间再进行侦听。

• 总结：持续式关注发送前的操作，冲突检测关注发送后的操作。

(4)轮流协议：结合信道划分和随机访问的优势

①轮询协议：在站点间选择一个主结点，为其他站点分配信道使用权。传输完成后通知下一个站点。

• 缺点：轮询需要占用带宽，通知引入延迟，单点故障。

②令牌传递：令牌代表发送权限。

• 只有获得令牌的站点可以发送数据，令牌通过特殊的令牌消息进行传递。

• 站点组织成一定的结构（如环）。可以安排顺序。

• 缺点：令牌的维护代价，令牌的可靠性。

③位图协议：分为竞争期和传输器。

• 竞争期：在自己的时槽内发送竞争比特，举手示意资源预留。

• 传输期：按需发送，明确了使用权以避免冲突。

• 缺点：无法考虑优先级。

④二进制倒计数协议：站点编为相同长度的序号。

• 竞争期有数据发送的站点从高序号到地序号排队，高者得到发送权。

• 特点：高序号站点优先。

⑤有限竞争协议：低负荷时使用竞争法减少延迟时间，高负荷时使用无冲突法获得高信道效率。

(5)应用

①信道划分：电缆等。

②随机访问：冲突检测（有线信道），CSMA/CD（以太网、802.11 WiFi）。

③轮流协议：蓝牙、光纤。

5. 局域网内数据传输（链路层交换、逆向学习、MAC 地址表转发）、局域网间数据传输（联系网络层内容）

(1)局域网：不需要网络层技术就可以传输数据的网络（就是子网）

①MAC 地址（局域网地址/物理地址）：物理上相连的网络接口之间收发帧。

(2)局域网内数据传输

①链路层交换：核心是交换机。

• 交换机工作在数据链路层，检查 MAC 帧的目的地址对收到的帧进行转发。

• 理想的交换机是透明的，即插即用，无需任何配置，网络中的站点无需感知交换机的存在。

②逆向学习：交换机通过逆向学习帧的源地址获取主机所在的位置，构建 MAC 地址表。

• 交换机会逆向学习数据帧的源 MAC 地址以及对应端口。

• 交换机记录帧到达时间，设定老化时间。老化时间到期时该表项被清除。
• MAC 地址表的构建：帧的源地址对应的项不在表中时增加表项；老化时间到期时删除表项；帧的源地址在表中时更新表项中的时间戳。
③MAC 地址表转发：对于收到的一个帧：
•（转发）若在 MAC 地址表中找到匹配项，则从对应端口转发出去。
•（过滤）若输出端口与输入端口相同，丢弃。
•（泛洪）若找不到匹配表项，则从除了入境口外的所有端口发送出去（广播帧和未知单播帧需要泛洪）。

(3)局域网间数据传输

①从 A 传输数据到 B，A 需要知道的信息：B 的 IP 地址，A 的最近路由器 R 的 IP 地址（通过静态配置或 DHCP 得到），R 的 MAC 地址（通过 ARP 得到）。
②传输过程：
• A 创建网络层数据报、链路层帧。
• 帧从 A 发往 R，R 收到帧后去除链路层帧头交往网络层。
•R 根据目的 IP：B 查询转发表决定下一条，R 封装链路层帧，目的 MAC 地址为 B 的 MAC 地址。
• R 将数据发往 B。
• B 提取 IP 报文。
③关于路由器：
• 有两个接口，每个接口都有自己的 IP 地址和 MAC 地址，它们属于不同的子网。
• 两个接口的 ARP 功能独立，分别记录两个不同子网的 IP、MAC 映射。

6. 以太网

①以太网规定最短有效帧长为 64 字节，长度小于 64 字节的帧都是无效帧（直接丢弃）。
②提供的服务：无连接、不可靠、多路访问控制。

7. 虚拟局域网

(1)动机：为每一组用户建立各自局域网、广播域，但不添置新交换机

(2)类型：基于端口/MAC/协议/子网

①基于端口的 VLAN：将端口分为不同组，每一组如同一个独立的交换机（最常见）。
• 过程：创建 VLAN，指定成员端口。
• 好处：流量隔离，动态配置。
②基于 MAC 地址的 VLAN：MAC 地址决定了成员身份。
③基于协议的 VLAN 地址：通常需要服务器参与。
④基于子网的 VLAN：一个子网就是一个 VLAN。

(3)优点：

①有效控制广播域范围（广播流量被限制在一个 VLAN 内）。
②增强网络的安全性（VLAN 间相互隔离，无法进行二层通信，不同 VLAN 需通过三层设备通信）。
③灵活构建虚拟工作组（同一工作组的用户不必局限于同一物理范围）。
④提高网络的可管理性（将不同的业务规划到不同 VLAN 便于管理）。

8. 无线网络

(1)组成元素：无线主机、基站、无线链路

(2)分类（见下一页）

(3)核心问题

①无线：如何通过无线链路进行数据传输（数据链路层技术）。

	单跳	多跳
基于基础设施	主机通过基站 连向更大的网络 (蜂窝网、WiFi)	主机通过多个无线节点 进行中继，连接到更大网络 (无线网状网络 wireless mesh networks)
无基础设施 (自组织)	无基站，源-目的直连 (蓝牙)	无基站， 通过多个无线节点中继 (无线自组织网络MANET， 车载自组织网络VANET)

②移动：处理主机所连基站发生的变动。

(4)无线链路特征与多路访问

①特征：递减的信号强度，其他信号源的干扰，多路径传播。

②多路访问问题：隐藏终端，信号衰减。

• 码分复用（CDMA）：将所有可能的编码集合划分给用户，允许用户同时传输数据。

编码过程：　　编码结果 = 数据 × 码片序列

解码过程：　　解码结果 =(收到的编码数据 · 发送者码片序列) / M

(5)802.11 无线局域网

①所有的 802.11 无线局域网都使用 CSMA/CA 进行多路访问控制，都支持基础设施模式与自组网模式。

②架构：网络由基本服务集组成。

• 基本服务集包含无线主机、接入点、自组织模式。

③每个无线主机在能够发送或接收网络层数据前必须与一个 AP 关联。

• 被动扫描：AP 周期发送信标帧，主机扫描信道监听信标帧，H1 向选择的 AP 发送关联请求帧，被选择的 AP 向 H1 回复关联响应帧。

• 主机也可以主动扫描，H1 广播探测请求帧，AP 返回探测响应帧，H1 向选择的 AP 发送关联请求帧，被选择的 AP 向 H1 回复关联响应帧。

④多路访问控制：发送前侦听信道，发送时不再进行冲突检测。

⑤CSMA/CA：

• 发送方：发送前信道空闲时间达到 DIFS 时发送整个帧；发送前检测到信号忙时选择随机值作为计时器，计时器在信道空闲时递减；计时器减为 0 时发送并等待 ACK。收到 ACK 后若还需要发送则继续检测信道，否则使用更大的随机值作为计时器。

• 接收方：收到正确的帧后经过 SIFS 时间发送 ACK。

⑥预约机制：允许发送者“预约保留”信道。

• 发送者先使用 CSMA/CA 发送小报文 RTS 到基站（RTS 预约可能冲突）。

• 基站广播 CTS 消息作为对 RTS 回复。

• 所有站点收到 CTS，发送者开始传输，其他站点推迟传输。

• 其他站点在传输完成后也能收到 ACK，开始传输。

(6)蜂窝网

①核心思想：通过移动电话网络提供移动数据传输。

②架构：包括小区、移动交换中心、有线网络。

③第一跳：空中接口，将无线设备连接到基站。

• 2G 组合使用 FDMA 与 TDMA。

• 3G 使用 CDMA。

(7)管理移动性的一般方法

①基本架构：

• 对移动性分类，无移动性~高移动性。

• 移动性术语：关注归属网络、归属代理、永久地址。

• 设备注册：外部代理通知归属代理，该设备属于自己的访问网络以及拥有的转交地址。外部代理从而知道设备的存在，归属代理知道设备的位置。

②间接路由：通信者-归属代理-外部代理-移动设备。

• 在新访问网络进行注册，新访问网络中的外部代理通知归属代理，归属代理更新该设备的转交地址记录，后续报文使用新转交地址发往该设备。

• 对于通信者而言，设备移动、访问网络改变、转交地址改变是透明的。

• 问题：三角路由问题，导致效率低下。

③直接路由：通信者-外部代理-移动设备。

• 解决了三角路由问题。

• 对通信者不透明，通信者必须从归属代理获取转交地址。

• 使用锚外部代理，通信者数据发往锚外部代理，由锚外部代理转发到当前网络。

• 设备到达新的访问网络时向新的外部代理注册，新的外部代理向锚外部代理提供新的转交地址。

(8)移动 IP、蜂窝网的移动性管理

①移动 IP：包括报文间接路由、代理发现、代理注册。

• 间接路由：归属代理到外部代理：报文封装；外部代理到设备：解封装后转发。

• 代理发现：代理通告、代理请求。

• 代理注册：ICMP 代理通告、注册请求、注册回复。

②蜂窝网：包括归属网络和被访网络。

• 归属网络：设备订购服务的网络，归属位置服务器（HLR）和归属 MSC 构成归属代理。

• 被访网络：用户设备当前所处网络，被访者位置注册（VLR）和被访网络 MSC 构成外部代理（被访网络和归属网络往往是同一个网络）。