计算机网络复习提纲

Chapter 1：计算机网络和因特网

1.1 什么是因特网

1.1.1 具体构成描述（bolts and nuts view）

- ✧ 主机或端系统
- ✧ 通信链路
- ✧ 路由器
- ✧ 协议
- ✧ 网络的网络

1.1.2 服务描述

- ✧ 因特网是一种基础设施，新应用程序在其上不断地被发明和设置；
- ✧ 因特网为应用程序提供多种服务。

1.1.3 什么是协议

- ✧ 网络协议：类似于人类协议，交换报文和采取动作的实体是某些设备的硬件或软件组件。
- ✧ 协议的定义：一个协议定义了两个或多个通信实体之间交换的报文格式和次序，以及在报文传输和/或接受其他时间方面所采取的动作。

1.2 网络边缘

- ✧ 端系统也称为主机，主机被划分为客户机和服务器。
- ✧ 客户机非正式的等同于桌面 PC、移动 PC 和 PDA，服务器非正式的等同于更为强大的及其，用于存储和发布 web 页面、流视频以及转发电子邮件等。

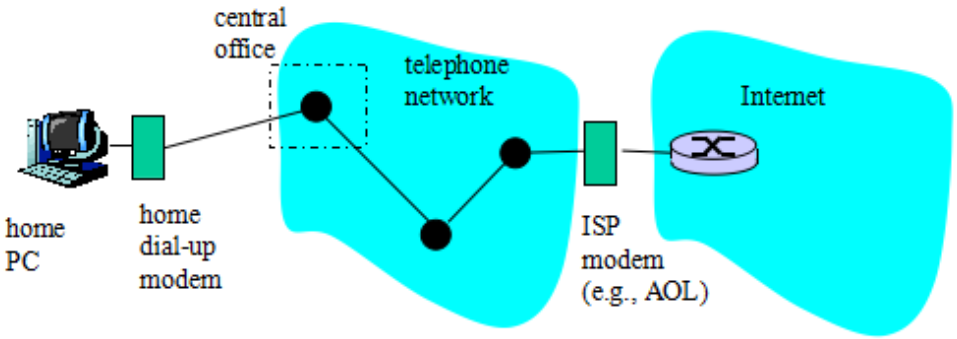
1.2.1 客户机和服务器程序

- ✧ 客户机程序是运行在一个端系统上的程序，它发出请求，并从运行在另一个端系统上的服务器程序接收服务。

1.2.2 接入网

- ✧ 边缘路由器：由端系统到任何其他远程端系统的路径上的第一台路由器。
- ✧ 接入网（access network）：将端系统连接到边缘路由器的物理链路。
- ✧ 接入网大致不严格的分为以下三种类型：
- ❖ 住宅接入（residential access networks）：

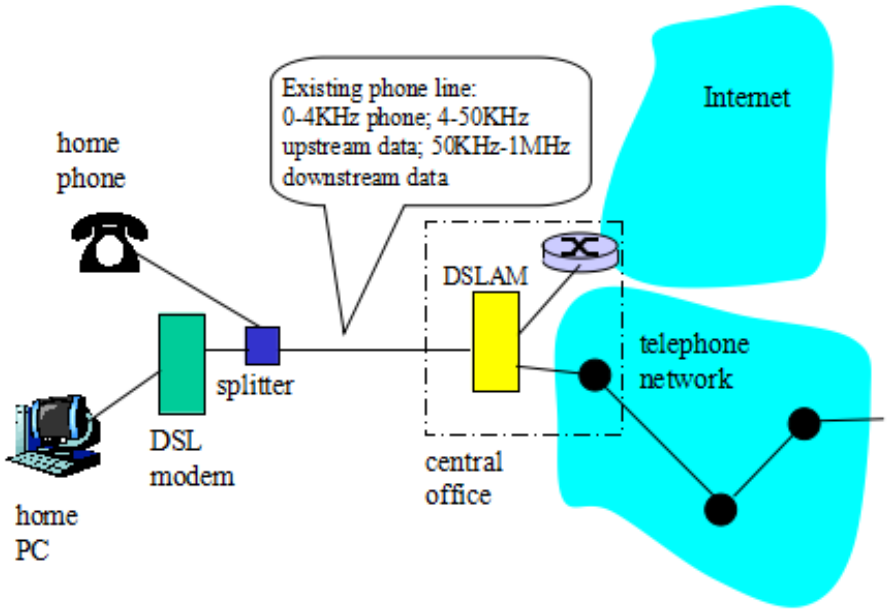
1) 拨号调制解调器



家用调制解调器将 PC 输出的数字信号转换为模拟形式，在模拟电话线上传输，ISP 的调制解调器接收到模拟信号转换成数字形式。

速率：最高至 56kbps。（缓慢地令人难以接受）

2) 数字用户线（digital subscriber line， DSL）：



高速下行信道，位于 50kHz 到 1MHz 频段；

中速上行信道，位于 4kHz 到 50kHz 频段；

普通的双向电话信道，位于 0 到 4kHz 频段。

带宽专用。

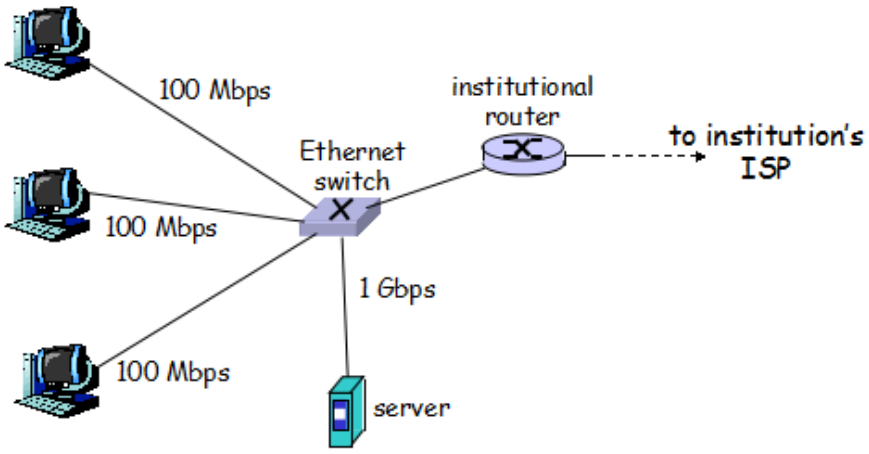
3) 混合光纤同轴电缆（hybrid fiber-coaxial cable， HFC）：

共享广播媒体。

带宽共享。

4) 卫星链路：速率超过 1Mbps。

❖ 公司接入（company access networks）：以太网技术（Ethernet）



❖ 无线接入（wireless access networks）：

1) 无线局域网：无线用户与位于几十米半径内的基站（无线接入点）之间传输/接收分组。

2) 广域无线接入网（wide-area wireless access network）：分组经用于蜂窝电话的想听无线基础设施发送，基站由电信提供商管理，为数万米半径内用户提供无线接入服务。

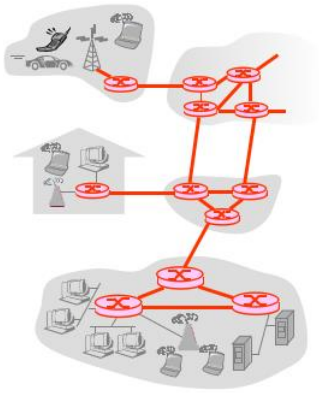
1.2.3 物理媒体

✧ 导引型媒体：电波沿着固体媒体被导引。

✧ 非导引型媒体：电波在空气或外层空间（如无线局域网或数字卫星频道）中传播。

1.3 网络核心

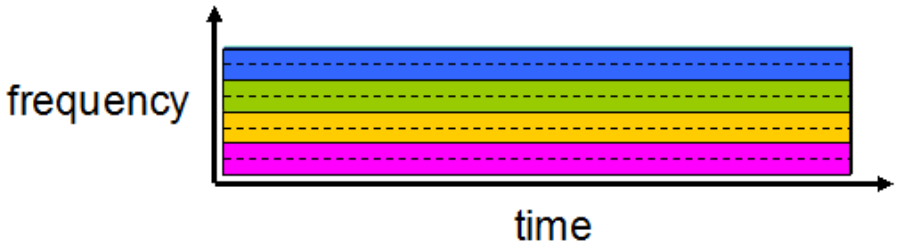
✧ 网络核心：互联因特网端系统的分组交换机和链路的网状网络。



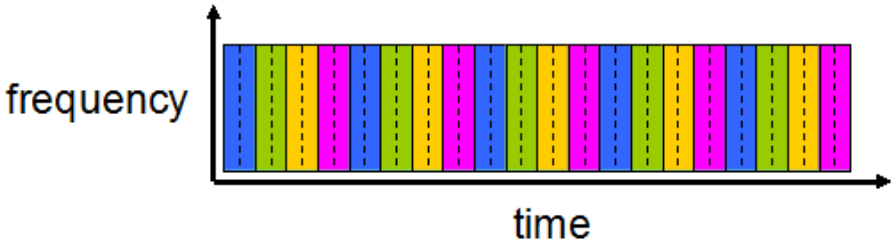
1.3.1 电路交换和分组交换

✧ 电路交换：两台主机要通信时，网络在两台主机之间创建一条专用的端到端连接（预留一条电路）。

1) 每条链路具有 n 条电路，每条链路由端到端连接使用，该连接在连接期间获得该链路带宽的 1/n。
2) 电路交换网络中的多路复用：
   a. 频分多路复用（FDM）：链路的频谱由跨越链路创建的所有链路共享，每条连接专用一个频段。



   b. 时分多路复用（TDM）：时间被划分为固定区间的帧，并且每帧被划分为固定数量的时隙，创建连接时，网络在每帧中为该连接指定一个时隙，这些时隙专门为该连接单独使用。



3) 电路交换的问题：静默期专用电路空闲——效率低。

✧ 分组交换：源主机将分组划分成为较小的数据块，并称之为分组。在源和目的地之间，分组以链路的最大传输速率在通信链路上传输。

1) 存储转发机制：交换机能够开始向输出链路传输该分组的第一个比特之前，必须接收到整个分组。（引入存储转发时延）
2) 排队时延：分组在输出缓存中等待。
3) 分组丢失（丢包）：分组到达时输出缓存被充满。

✧ 电路交换与分组交换的对比：

1) 分组交换的端到端时延是变动和不可预测的，故不适合实时服务；
2) 分组交换提供了比电路交换更好的带宽共享；

3) 分组交换比电路交换更简单、有效，实现成本更低；

4) 总的来说，分组交换的性能能够优于电路交换。

✧ 统计多路复用（statistical multiplexing）：按需（而不是预分配）共享资源被称为资源的统计多路复用。

1.3.2 分组是怎样通过分组交换网形成其通路的

每个通过网络传输的分组在它的首部包含了其目的地址，当分组到达一台路由器时，该路由器检查分组目的地址的一部分，并向相邻路由器转发该分组。每个路由器具有一个转发表用于将目的地址（或目的地址的一部分）映射到输出链路。

1.3.3 ISP 和因特网主干

✧ 因特网是网络的网络。

✧ 因特网边缘的接入网络通过分层的 ISP 层次结构与因特网的其他部分相连：第一层 ISP、第二层 ISP、接入 ISP。

✧ 第一层 ISP：也被称为因特网主干（Internet backbone）网络。

1) 直接与其他**每一个**第一层 ISP 相连；

2) 与大量的第二层和其他客户网络相连；

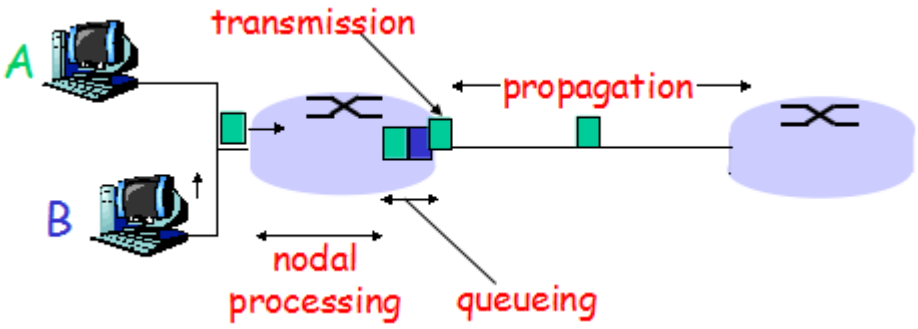
3) 覆盖国际区域。

✧ 当两个 ISP 直接相连时，它们被称为彼此时对等的。

✧ 汇集点（Point of Presence， POP）：在一个 ISP 中，某 ISP 和其他 ISP 的连接点被称为汇集点。

1.4 分组交换网中的时延、丢包和吞吐量

1.4.1 分组交换网中的时延概述



✧ 处理时延（nodal processing）：检查分组首部和决定该分组导向何处，检查比特差错所需的时间。

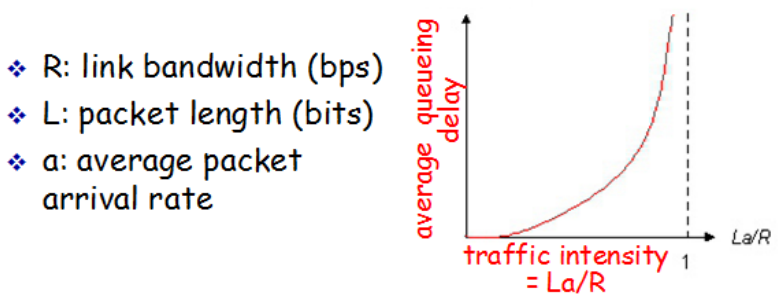
✧ 排队时延（queuing）：分组在链路上等待传输的时延，取决于先期到达的、正在排队等待向链路传输的分组的数量。

✧ 传输时延（transmission）：将所有分组的比特推（传输）向链路所需要的时间。

✧ 传播时延（propagation）：从链路的起点到下一个路由器传播所需要的时间。

1.4.2 排队时延和丢包

✧ 排队时延对不同的分组不同的，取决于流量到达该队列的速率、链路的传输速率和到达流量的性质（周期性到达或突发形式到达）。



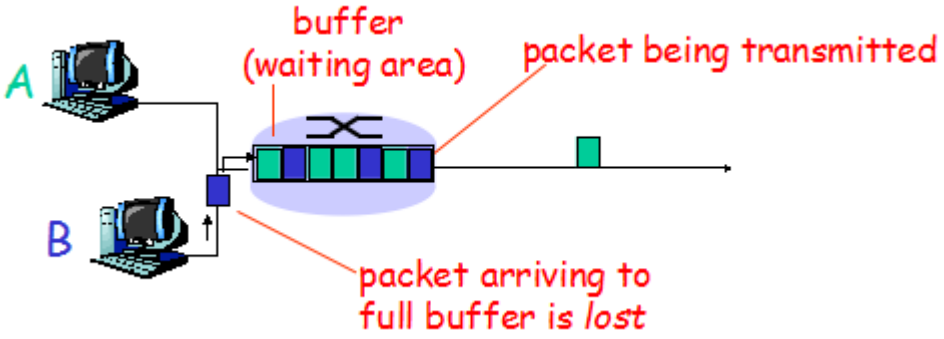
✧ 流量强度（traffic intensity）：La/R

La/R→0：几乎没有分组到达，并且到达间隔很大，平均排队时延接近于 0。

La/R→1：可能有很大的平均排队时延。

La/R > 1：比特到达队列的平均速率超过从该队列传输出去的速率，队列的增加趋于**无界。**

✧ 丢包：到达分组发现满队列时将被丢弃。（丢失分组的数量随着流量强度的增加而增加）



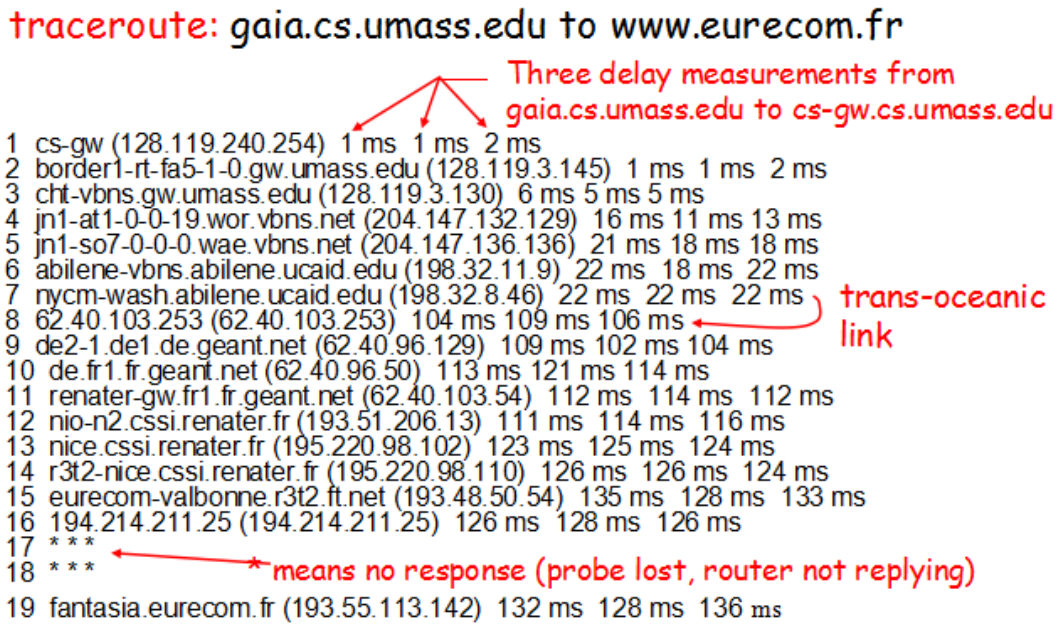
1.4.3 端到端时延

✧ Traceroute：

假设源和目的地之间有 N-1 台路由器，则源向网络发送 N 个特殊的分组，其中每个分组地址指向最终目的地。这 N 个特殊分组标识为 1 到 N。当第 n 台路由器接受到第 n 个标识为 n 的分组时，该路由器不是向它的目的地转发该分组，而是向源回送一个报文。该源记录从它发送一个分组到它接收到对应返回报文所经受的时间以及返回该报文的路由器（或目的主机）的名字和地址

源能够重建从源到目的地所采用的路由，并且该源决定到所有中间路由器的往返时延。

Traceroute 在上述描述的实验重复了 3 次。（向目的地发送 3*N 个分组）



1.4.4 计算机网络的吞吐量

✧ 瞬时吞吐量：任何瞬间主机 B 接收文件的速率。

✧ 平均吞吐量：在一段时间内接收文件的平均速率。

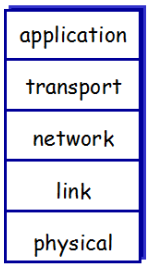
✧ 服务器到客户机传输文件的吞吐量是沿着它们之间路径的瓶颈链路的速率。

✧ 因特网中对吞吐量的限制因素通常是接入网。

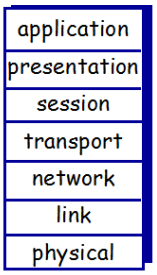
1.5 协议层次和它们的服务模型

1.5.1 分层的体系结构

- 分层的原因：
  1) 可以讨论一个定义良好且大而复杂的系统的特定部分；
  2) 模块化使得维护和提升更加容易。
- 分层的潜在缺点：
  1) 某层可能重复较低层的功能；
  2) 某层的功能可能需要仅在其他某层才出现的信息。
- 各层的所有协议称为协议栈。
- 因特网的协议栈组成（自顶向下方法）：
  1) 应用层；
  2) 运输层：提供在应用程序端点之间传送应用层报文的服务；
  3) 网络层：将网络层分组从一台主机移动到另一台主机；
  4) 链路层：为了将分组从一个节点（主机或路由器）移动到路径上的下一个节点，网络层必须依靠链路层的服务；
  5) 物理层：将帧中的每个比特从一个节点移动到下一个节点。

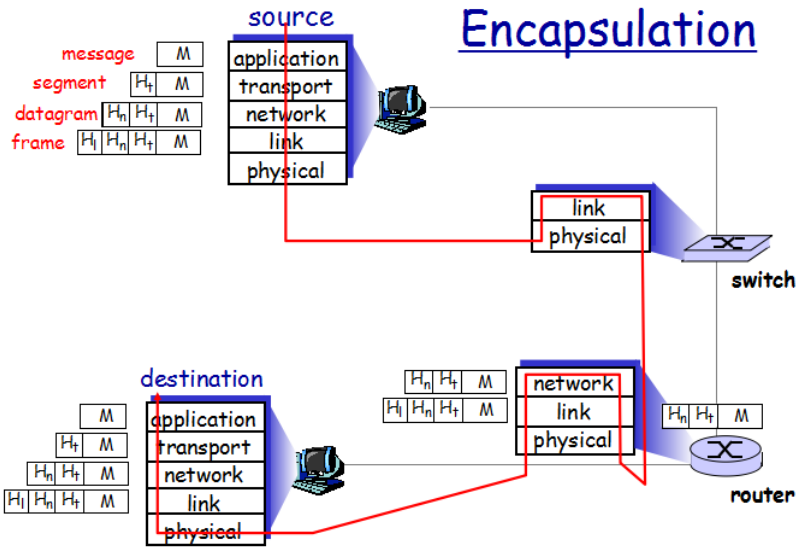


- ISO 模型：国际标准化组织（ISO）提出开放系统互联模型（OSI），将计算机网络组织分为 7 层。
  1) 表示层（presentation）：使通信的应用程序能够解释交换数据的含义，提供的服务包括数据压缩、数据解密、数据描述；
  2) 会话层（session）：提供数据交换的定界和同步功能，包括建立检查点和恢复方案的方法。



1.5.2 报文、报文段、数据报和帧

- 封装（encapsulation）



- 报文：应用层信息分组。
- 报文段：运输层分组。
- 数据报：网络层分组。
- 帧：链路层分组。
- 物理层传输比特。

1.6 攻击威胁下的网络

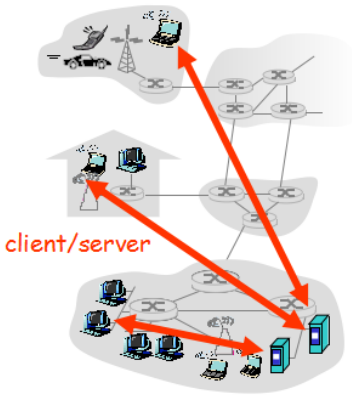
- ✧ 病毒（virus）：需要某种形式的用户交互来感染用户设备的恶意软件。
- ✧ 蠕虫（worm）：无需任何明显用户交互就能进入用户设备的恶意软件。
- ✧ 特洛伊木马（Trojan horse）：隐藏在有用软件中的恶意软件。
- ✧ 坏家伙如何攻击计算机网络：
  1) 经因特网将恶意软件放入你的计算机；
     - ◆ 病毒（virus）：需要某种形式的用户交互来感染用户设备的恶意软件。
     - ◆ 蠕虫（worm）：无需任何明显用户交互就能进入用户设备的恶意软件。
     - ◆ 特洛伊木马（Trojan horse）：隐藏在有用软件中的恶意软件。
  2) 攻击服务器和网络基础设施；
     - ◆ 弱点攻击：向目标主机易受攻击的应用程序或操作系统发送制作精细的报文；
     - ◆ 带宽洪泛：向目的主机发送大量的分组，导致目标的接入链路变得拥塞，使合法分组无法到达服务器；
     - ◆ 连接洪泛：在目标主机中创建大量的半开或全开 TCP 连接，目标主机因伪造的连接而陷入困境，停止合法的连接。
  3) 嗅探分组；
     - ◆ 记录每个流经的分组拷贝的被动接收机：分组嗅探器。
     - ◆ 分组嗅探器是被动的，不向信道中注入分组，从而哪一检测他们的存在
  4) 伪装成你信任的人；
     - ◆ IP 哄骗（IP spoofing）：将具有虚假原地址的分组诸如因特网的能力。
     - ◆ 采用端点鉴别机制，确保报文源自正确的地方。
  5) 修改或删除报文。
     - ◆ 中间人攻击：坏家伙插入到两个通信实体之间的通信路径中，可以嗅探到主机之间传递的分组，甚而危及到发送数据的完整性。

1.7 计算机网络和因特网的历史

Chapter 2：应用层

2.1 应用层协议原理

2.1.1 网络应用程序体系结构



- ✧ 客户机/服务器体系结构：
  1) 服务器：
     - ◆ 总是打开；
     - ◆ 固定、周知的 IP 地址；
     - ◆ 主机群集（服务器场，server farm）：创建强大的虚拟服务器，避免一台服务器处理所有请求不堪重负。
  2) 客户机：
     - ◆ 不总是打开；
     - ◆ 向服务器发出请求；
     - ◆ IP 地址不固定；
     - ◆ 客户机与客户机之间不直接联系。
- ✧ P2P 体系结构：对总是打开的基础设施服务器有最小的（或没有）依赖，任意间断连接的主机对（对等方）直接相互通信。
     - ◆ 最突出特性：自扩展性（self-scalability）——尽管每个对等方请求文件产生负载，但每个

对等方向其他对等方分发文件也为系统增加了服务能力。

- 不需要庞大的服务器基础设施和服务器带宽。

✧ CS 和 P2P 混合体系结构。

2.1.2 进程通信

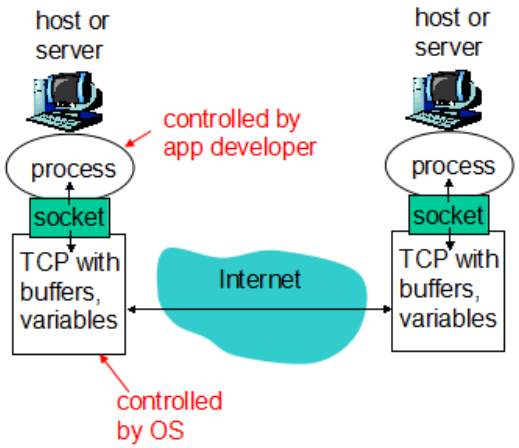
✧ 进程运行在同一端系统：进程间通信机制。

✧ 进程运行在不同端系统：交换报文。

✧ 客户机和服务器进程：在给定的一对进程之间的通信会话中，发起通信（即在会话开始时与其他进程联系）的进程被标识为客户机，在会话开始时等待联系的进程是服务器。

✧ 对于 P2P 文件共享，下载文件的对等方被标识为客户机，上载文件的对等方被标识为服务器。

✧ 套接字（sockets）：同一台主机内应用层与运输层之间的接口，是在网络上建立网络应用程序的可编程接口（因此套接字也成为应用程序和网络之间的应用程序编程接口）。



2.1.3 可供应用程序使用的运输服务

✧ 可靠数据传输

✧ 吞吐量：运输层协议能够以某种特定的速率提供确保的可用吞吐量。

1) 带宽敏感的应用：具有吞吐量要求；
2) 弹性应用：根据需要充分利用可够使用的吞吐量。

✧ 定时：在实时应用中，为了有效性而对数据交付有严格的时间限制。

✧ 安全性：对发送进程和接收进程保密，以防发送进程和接收进程以某种方式观察导数据。

2.1.4 因特网提供的运输服务

✧ TCP：

1) 面向连接服务：连接是全双工的——连接双方的进程可以在此连接上同时进行报文收发。
2) 可靠数据传输服务：无差错、按适当顺序交付发送的数据。
3) 拥塞控制机制：发送方与接收方之间的网络出现拥塞时，TCP 协议抑制发送进程。TCP 协议试图限制每个 TCP 连接，以达到公平共享带宽的目的。

✧ UDP：不提供不必要服务的轻量级运输层协议，仅提供最小服务。

✧ 因特网运输层协议：不提供吞吐量和定时服务。

✧ 进程寻址：

1) 主机的名称或地址——IP 地址；
2) 指定目的主机上接收进程的标识——目的地端口号。

2.1.5 应用层协议

✧ 应用层协议定义：

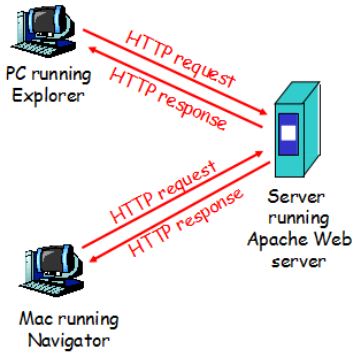
1) 交换报文的类型：请求报文、响应报文；

2) 各种报文类型的语法；
3) 字段的语义；
4) 进程何时、如何发送报文及对报文进行响应的规则。

2.1.6 本书涉及的网络应用

2.2 Web 应用和 HTTP 协议

2.2.1 HTTP 概况



- Web 页面由对象组成。
- 对象：文件，例如 HTML 文件、JPEG 图形文件、Java 小程序或视频片段文件，可通过 URL 地址寻址。
- URL 地址：由存放对象的服务器主机名和对象的路径名共同组成。
- HTTP（超文本传输协议）：
  1) 由一个客户机程序和一个服务器程序组成，通过交换报文进行会话；
  2) HTTP 协议定义报文的格式以及交换报文的方式；
  3) 使用 TCP 作为运输层协议；
  4) 无状态协议：HTTP 协议不保存客户机的任何信息。

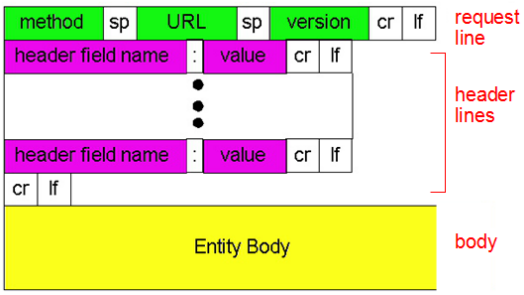
2.2.2 非持久连接和持久连接

- RTT（Round-trip TIme）：一个小分组从客户机到服务器再回到客户机所花费的时间。
- 非持久连接：每个请求及相应的响应对经一个单独的 TCP 连接发送。

  缺陷：
  1) 为每一个请求的对象建立和维护一个全新的连接，对于每个这样的连接，在客户机和服务器都需要分配 TCP 的缓冲区和变量，给服务器带来严重负担；
  2) 每一个对象的传输时延很大（2 个 RTT）。
- 持久连接：所有请求及相应的响应经相同 TCP 连接发送。
  1) 服务器再发送响应后保持该 TCP 连接打开；
  2) 在相同的客户机与服务器之间的后续请求和相应报文可通过相同的连接进行传送；
  3) 一个连接经过一定时间间隔仍未被使用，HTTP 服务器将关闭该连接。

2.2.3 HTTP 报文格式

- HTTP 报文有两种：请求报文、响应报文。
- 请求报文：



1) POST：用户提交表单，输入在实体主体；
2) GET：输入数据在 URL 地址中；
3) HEAD：不返回请求对象，用于故障跟踪；
4) PUT：将对象上传到指定的 Web 服务器上指定的路径；
5) DELETE：删除指定 Web 服务器上的对象。

- 响应报文：
  1) 状态行：协议版本、状态码、响应状态信息。
     - 状态码：
       - 200 OK：请求成功，信息包含在返回的响应报文中；
       - 301 Moved Permanently：请求的对象被永久转移，新的 URL 定义在响应报文的 Location：首部行；

- 400 Bad Request：请求不能被服务器理解；
- 404 Not Found：被请求的文档不再服务器上；
- 505 HTTP Version Not Supported：服务器不支持请求报文使用的 HTTP 协议版本。

2.2.4 用户与服务器的交互：cookie

✧ cookie：允许站点跟踪用户。

✧ cookie 的四个组成部分：

1) HTTP 响应报文中有一个 cookie 首部行；
2) HTTP 请求报文中有一个 cookie 首部行；
3) 用户端系统中保留一个 cookie 文件，，由用户浏览器管理；
4) Web 站点有一个后端数据库。

✧ 如何 cookie 跟踪用户：

1) 用户首次访问某站点，第一次发出请求报文；
2) 请求报文到达 Web 服务器后，服务器产生唯一识别码，并以此作为索引在后端数据库中产生一个表项；
3) 服务器用包含 Set-cookie：首部行的 HTTP 相应报文对 Susan 的浏览器进行响应；
4) 之后每次用户向服务器发送的请求报文都包括以下首部行 cookie:；
5) 这样就实现了服务器对用户的跟踪。

2.2.5 Web 缓存

✧ Web 缓存器也叫代理服务器（proxy server），是能代表初始 Web 服务器来满足 HTTP 请求的网络实体。

✧ Web 缓存器有自己的磁盘存储空间，在该存储空间中保存最近请求过的对象的拷贝。

✧ Web 缓存器的工作流程：

1) 浏览器建立一个到 Web 缓存器的 TCP 连接，并向其发送一个 HTTP 请求以请求对象；
2) Web 缓存器检查本地是否存储了该对象的拷贝，若有，则用 HTTP 响应报文向客户机浏览器返回该对象；
3) 若 Web 缓存器没有该对象，就向初始服务器打开一个 TCP 连接，请求该对象；
4) Web 缓存器接收到该对象后，在本地存储空间存储一份拷贝，并向浏览器发送该拷贝对象。

✧ Web 缓存器既是服务器又是客户机。

✧ 部署 Web 缓存器的原因：

1) Web 缓存器可以大大地减少对客户机请求的响应时间；
2) Web 缓存器可以大大减少一个机构内部网与因特网接入链路上的通信量；
3) Web 缓存器能从整体上大大降低网上的 Web 流量，从而改善了所有应用的性能。

2.2.6 条件 GET 方法

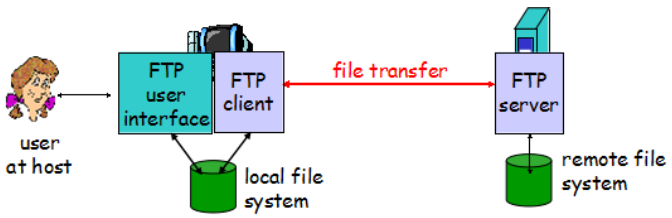
✧ 条件请求报文：包含 If-modified-since：首部行，保证缓存器中存储的对象拷贝是最新的。

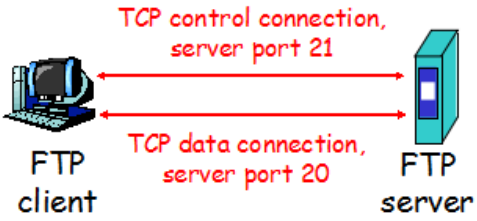
✧ 服务器响应条件 GET 时，并不包含所请求对象。原因：不浪费带宽，不增加用户感受到的响应时间。

2.3 文件传输协议：FTP

✧ FTP 会话中，用户通过一台主机，向一台远程主机上传文件或从远程主机下载文件。

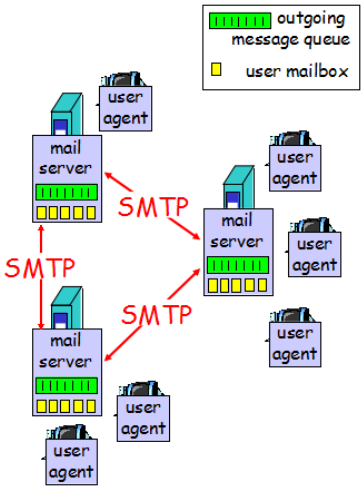
✧ 用户通过一个 FTP 用户代理与 FTP 交互。





- FTP 用两个并行的 TCP 连接传输文件：
  1) 控制连接：在两个主机之间传输控制信息；
  2) 数据连接：实际传输一个文件。
     - 控制连接贯穿了整个用户会话期间，针对每次文件传输都需新建一个数据连接。
- FTP 的控制信息是带外（out-of-band）传输：使用一个分离的控制连接。
  注：HTTP 为带内传输。
- FTP 服务器在整个会话期间保留用户的状态信息。（对每个活动着的用户会话的状态进行追踪，可以对 FTP 会话总述进行限制）
- FTP 常见命令：
  1) USER username：向服务器发送用户标识
  2) PASS password：向服务器发送口令
  3) LIST：请求服务器返回远程主机上当前目录的所有文件列表
  4) RETR filename：从远程主机的当前目录检索文件。触发远程主机建立一个数据连接，并在该数据连接上发送所请求的文件
  5) STOR filename：用于向远程主机的当前目录存放文件
- FTP 典型回答：
  1) 331 Username OK，Password required
  2) 125 Data connection already open；transfer starting
  3) 425 Can’t open data connection
  4) 452 Error writing file

2.4 因特网中的电子邮件



- 电子邮件系统的组成部分：
  1) 用户代理：允许用户阅读、回复、转发、保存和撰写报文；
  2) 邮件服务器：是电子邮件系统的核心。每个接收方在某服务器上有一个邮箱，邮箱管理和维护发送给这个用户的报文；
  3) 简单邮件传输协议（SMTP）：使用 TCP 可靠数据传输服务，从发送方的邮件服务器向接收方的邮件服务器发送邮件。

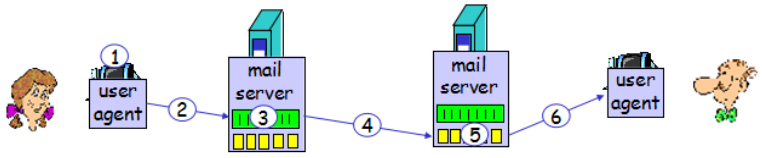
2.4.1 SMTP

- SMTP 不使用中间邮件服务器发送邮件（direct transfer），即使这两个服务器位于地球两端。
- SMTP 在 25 号端口创建 TCP 连接。
- 传输阶段：
  1) 简单握手：SMTP 客户机指明发送方和接收方邮件地址；
  2) 发送报文：利用 TCP 连接提供的可靠数据传输将报文从发送邮件服务器传输到接收服务器；
  3) 关闭连接。
- 限制邮件报文的主体部分采用简单的 7 为 ASCII 码表示。
- SMTP 基本操作：
  1) Alice 调用她的邮件代理程序并提供 Bob 的邮件地址，撰写邮件，通过用户代理发送该邮件；
  2) Alice 的用户代理将报文发送给 Alice 的邮件服务器，在那里报文被放在报文发送队列中；
  3) 运行在 Alice 邮件服务器上的 SMTP 客户机端发现报文队列中的报文，创建一个到运行在 Bob 邮件服务器上的 SMTP 服务器的 TCP 连接；

4) 握手后，SMTP 客户机通过该 TCP 连接发送 Alice 的报文；

5) Bob 的邮件服务器上的服务器端接收该报文，Bob 的邮件服务器将该报文放入 Bob 的邮箱中；

6) Bob 方便的时候，调用用户代理阅读该报文。



✧ SMTP 客户机与服务器之间交换报文脚本的例子：

1) HELO 是 HELLO 的缩写；

2) HELO、MAIL FROM、REPT TO、DATA、QUIT 等命令均是自解释的。

3) 客户机通过发送一个只含一个句点“.”的行，告诉服务器报文结束。

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

✧ 与 SMTP 服务器直接对话：telnet ServerName 25

2.4.2 与 HTTP 的对比

✧ 两者的重要区别：

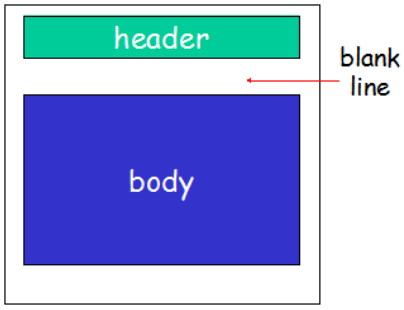
1) HTTP 是一个拉协议，用户使用 HTTP 从服务器拉取信息，TCP 连接由想获取文件的机器发起；SMTP 是一个推协议，TCP 连接由要发送文件的机器发起；

2) SMTP 要求每个报文（包括他们的主体）都使用 7 位 ASCII 格式，HTTP 没有这个限制；

3) HTTP 把每个对象封装到它自己的 HTTP 响应报文中，而因特网电子邮件把所有报文对象放在一个报文之中；

2.4.2 邮件报文格式和 MIME

✧ 邮件报文格式：



✧ 发送非 ASCII 文本的内容，发送方的用户代理需在报文中使用附加的首部行，定义在多用途因特网邮件扩展（Multipurpose Internet Mail Extension，MIME）。

1) Content-Type： 允许接收用户代理对报文采取适当的动作

2) Content-Transfer-Encoding： 提示接收用户代理该报文主题已经使用 ASCII 编码，并指出所用的编码类型。

2.4.4 邮件访问协议

✧ 邮件访问使用了客户机/服务器体系结构。

✧ 邮件访问协议：

1) 第三版的邮局协议（Post Office Protocol-Version 3）

◆ 特许（authorization）：用户代理发送（明文形式）用户名和口令以鉴别用户。

◆ 事务处理：用户代理取回报文，可以对报文做删除标记、取消报文删除标记以及获取邮件的统计信息。

◆ 更新：出现在客户机发出 quit 命令后，结束 POP3 会话，这是邮件服务器删除那些被标记

为删除的报文。

- ✓ 用户代理为用户配置“下载并删除”或“下载并保留”方式。
- ✓ “下载并删除”方式的缺陷：在一个端系统读取邮件后，将不能在另一端系统重新收取该邮件。
- ✓ POP3 会话中不携带状态信息。
- ✓ POP3 没有给用户提供任何创建远程文件夹以及为报文指派文件夹的方法。

2) 因特网邮件访问协议（Internet Mail Access Protocol，IMAP）

- ✧ IMAP 协议为用户提供创建文件夹以及在文件夹之间移动邮件的命令。
- ✧ IMAP 允许用户代理获取报文组件的命令，例如用户代理可以只读取一个报文的报文首部。
- ✧ IMAP 服务器维护了 IMAP 会话的用户状态信息。

✧ 基于 Web 的电子邮件：用户代理是普通的浏览器，用户和其远程邮箱之间的通信通过 HTTP 进行。

2.5 DNS：因特网的目录服务

2.5.1 DNS 提供的服务

✧ 域名系统（Domain Name System，DNS）：进行主机名到 IP 地址的转换。

✧ DNS：

1) 分层 DNS 服务器实现的分布式数据库；
2) 允许主机查询分布式数据库的应用层协议。

✧ DNS 协议运行在 UDP 之上，使用 53 号端口。

✧ DNS 提供的其他服务：

1) 主机别名：有着复杂主机名的主机可以拥有一个或者多个别名；
2) 邮件服务器别名；
3) 负载分配：在冗余的服务器之间进行负载分配。

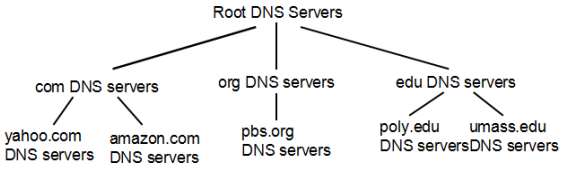
2.5.2 DNS 工作机理概述

✧ 集中式 DNS 服务器的问题：

1) 单点故障：如果该 DNS 服务器崩溃，整个因特网将随之瘫痪；
2) 通信容量：单个 DNS 服务器不得不处理所有的 DNS 查询；
3) 远距离的集中式数据库：单个 DNS 服务器不可能临近多有查询客户机，远距离查询可能导致严重时延；
4) 维护：DNS 中央数据库十分庞大，且需为不断添加的主机而频繁更新。

✧ DNS 服务器分为三种类型：

1) 根 DNS 服务器：
2) 顶级域（Top Level Domain，TLD）DNS 服务器：
3) 权威 DNS 服务器：在因特网上具有公共可访问主机的每个组织机构必须提供公共可访问的 DNS 记录，这些记录将这些主机的名字映射为 IP 地址，由权威 DNS 服务器负责保存这些 DNS 记录。



✧ 递归查询：被联系的服务器不断递归查询下一可能知道的服务器直至得到答案。

✧ 迭代查询：服务器返回下一个可查询的服务器名。

✧ DNS 将能将信息缓存在本地存储器，在一段时间后丢弃缓存的信息。

2.5.3 DNS 记录和报文

✧ DNS 分布式数据库的所有 DNS 服务器共同存储着资源记录（Resource Record）。

✧ 资源记录：（Name, Value, Type, TTL）

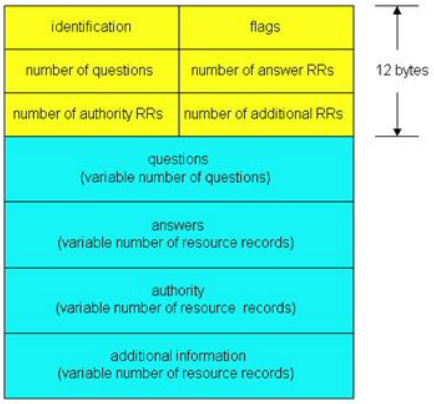
✧ TTL 是该记录的生存时间，决定了资源记录应该从缓存中删除的时间。

✧ 资源记录的类型：

1) Type=A，则 Name 是主机名，Value 是主机名的 IP 地址，类型为 A 的资源记录提供了标准主机名到 IP 地址的映射。
2) Type=NS，则 Name 是域，而 Value 是知道如何获得该域中主机 IP 地址的权威 DNS 服务器的主机名。这个记录用于沿着查询链路进一步路由 DNS 查询。
3) Type=CNAME，则 Value 是别名为 Name 的**主机**对应的规范主机名。
4) Type=MX，则 Value 是别名为 Name 的**邮件服务器**的规范主机名。

✧ DNS 报文：

1) 前 12 个字节是首部区域；
2) 问题区域：正在进行的查询信息。
   - ◆ 名字字段：指出正在被查询的主机名字；
   - ◆ 类型字段：住处正在询问的问题类型。
3) 回答区域：对最初请求的名字的资源记录。
4) 权威区域：其他权威 DNS 服务器的记录；
5) 附加区域。



✧ 在 DNS 数据库中插入记录的过程：

1) 向注册登记机构提供权威 DNS 服务器和辅助权威 DNS 服务器的名字和 IP 地址；
2) 注册登记机构将资源记录输入 TLD 服务器。

2.6 P2P 应用

2.6.1 P2P 文件分发

✧ 在 P2P 文件分发中，每个对等方都能够重新分发其所有的该文件的任何部分，从而协助服务器进行分发。

✧ P2P 体系结构的扩展性：

1) 变量名简述：
   - ◆ u_s：服务器接入链路的上载速率；
   - ◆ d_i：第 i 个对等方接入链路的上载速率；
   - ◆ d_i：第 i 个对等方接入链路的下载速率；
   - ◆ F：被分发的文件长度；
   - ◆ N：要获得文件拷贝的对等方的数量；
   - ◆ D_{cs}：服务器/客户机体系结构分发时间。
2) 服务器/客户机模型：

   服务器需向每个对等方传输一个文件拷贝，因此服务器必须传输 NF 比特：NF/u_s

   d_{min} 表示下载速率最小的对等方的下载速率，最小分发时间至少为：F/d_{min}

   最小分发时间：$D_{cs} \geq \max(NF/u_s, F/d_{min})$
3) P2P 体系结构：

   服务器发过一次的比特无需再次发送（对等方可互相重新分发）：F/u_s

   下载速率最小的对等方：F/d_{min}

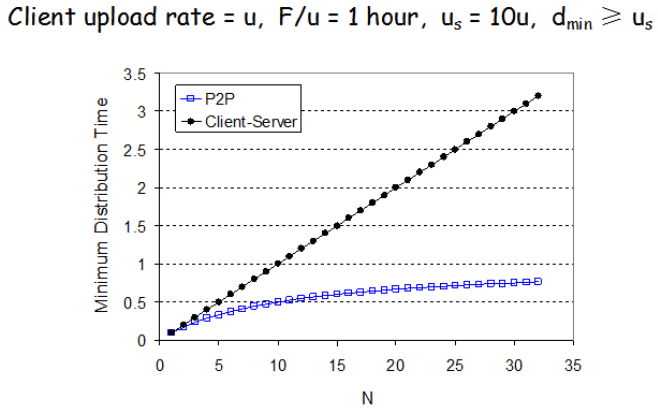
   系统总的上载能力等于服务器的上载能力加上每个对等方的上载速率，$u_{total}=u_s+u_1+...+u_n$，系统需向 N 个对等方每个上载 F 比特，这不可能快于 utotal 的速率完成：$NF/(us + \Sigma ui)$

   最小分发时间：$D_{P2P} \geq \max\{ F/us, F/min(di) , NF/(us + Sui) \}$

4) 两者比较：

客户机/服务器体系结构：随着对等方数量的增加，分发时间线性增长且无界；

P2P 体系结构：分发时间总是小于客户机/服务器体系结构，对任何多的对等方，分发时间总小于 1 小时。



✧ BitTorrent

1) 洪流（torrent）：参与**一个特定文件**分发的所有对等方的集合。
2) 基础设施节点，追踪器（tracker）：一个对等方加入洪流时，首先向追踪器注册，并周期性通知追踪器它仍在洪流中。
3) 邻近对等方：与某对等方建立 TCP 连接的其他所有对等方成为该对等方的邻近对等方。
4) 最稀罕优先（rarest first）：根据她没有的块从邻居中确定最稀罕的块（在她的邻居中拷贝数量最少的文件块），并优先请求那些块。——均衡每个块在洪流中的拷贝数量。
5) 对换算法：确定邻居的优先权（按当前提供数据的最高熟虑为标准），并持续测量接收比特的速率，确定以最高速率流入的四个邻居最为邻近对等方，然后将文件块发给这四个邻居。每过 10 秒，她重新计算该速率并可能修改这四个对等方。每过 30 秒，将随机选择另一个对等方（optimistically unchoke）并向它发送数据块。

2.6.2 在 P2P 区域中搜索信息（ppt 上没有，不知道考不考）

✧ 集中式索引

1) 具集中式索引的 P2P 文件共享系统是 P2P 和客户机/服务器混合体系结构。
2) 缺点：
   - ◆ 单点故障；
   - ◆ 性能瓶颈和基础设施费用：集中式服务器必须维护一个庞大的索引，每秒钟对数千次查询做出响应，通信量巨大；
   - ◆ 侵犯版权：P2P 文件共享系统允许用户免费获取受版权保护的内容，可能有法律问题。

✧ 查询洪泛

1) 完全分布式方法：索引全面地分布在对等方的区域中，每个对等方索引可供共享的文件二步索引其他文件。
2) 覆盖网络：对等方形成的抽象网络。
   - ◆ 如果对等方 X 和 Y 之间维护了一个 TCP 连接，则 X 和 Y 之间有一条边；
   - ◆ 覆盖网络由活跃对等方和他们之间的边构成；
   - ◆ 一条边不是一条物理链路，而是一条抽象链路，可能由许多物理链路组成。
3) 扩展性差：查询会传播到覆盖网络中的每个其他对等方，在支撑网络中产生大量流量。
4) 改进——受限查询洪泛：初始查询报文中设置对等方计数字并赋初值，将查询限制在一定范围内（可能无法定位查找的内容）。

5) 如何处理对等方的加入和离开：

假设新对等方 X 要加入覆盖网络：

- 发现覆盖网络中的其他对等方，让 X 维护一张对等方的列表（IP 地址）或能够联系维护这样列表的跟踪站点；
- X 试图与列表上的对等方建立 TCP 连接，直到与 Y 建立连接为止；
- X 向 Y 发送 ping 报文，包括对等方记数字段；接收到该报文后，Y 向其覆盖网络中的所有邻居转发，这些邻居继续转发直至计数字段为 0；
- 若对等 Z 接收到 ping 报文，将通过覆盖网络向 X 回发一个 pong 报文，包括 Z 的 IP 地址；
- X 接收到该 pong 报文后，将会知道覆盖网络中许多其他对等方的 IP 地址，因此与之创建 TCP 连接，进而创建多条边。

✧ 层次覆盖

1) 并非所有对等方都是平等的：超级对等方、子对等方。
2) 超级对等方维护一个索引，其中包括其子对等方正在共享的所有文件的标示符、有关文件的元数据和保持这些文件的子对等方的 IP 地址。
3) 超级对等方之间彼此建立 TCP 连接，形成覆盖网络。
4) 层次覆盖设计利用了对等方的异质性。

✧ 分布式散列表（Distributed Hash Table，DHT）

1) 产生一个全分布式索引，索引将文件标示符映射到文件位置；
2) 允许用户（原则上）确认文件的所有位置，而不产生过量的搜索流量；

2.6.3 案例学习：Skype 的 P2P 因特网电话

2.7 TCP 套接字编程

✧ 网络应用程序的核心：客户机程序和服务器程序。

✧ 网络应用程序的分类：

1) 网络应用程序，满足 RFC 所定义的标准协议；
2) 专用的网络应用程序，不必符合任何现有 RFC。

2.7.1 TCP 套接字编程

✧ 服务器为了能对客户机程序发起的连接做出响应：

1) 不能处于睡眠状态；
2) 需要有某种类型的门（welcome socket）。

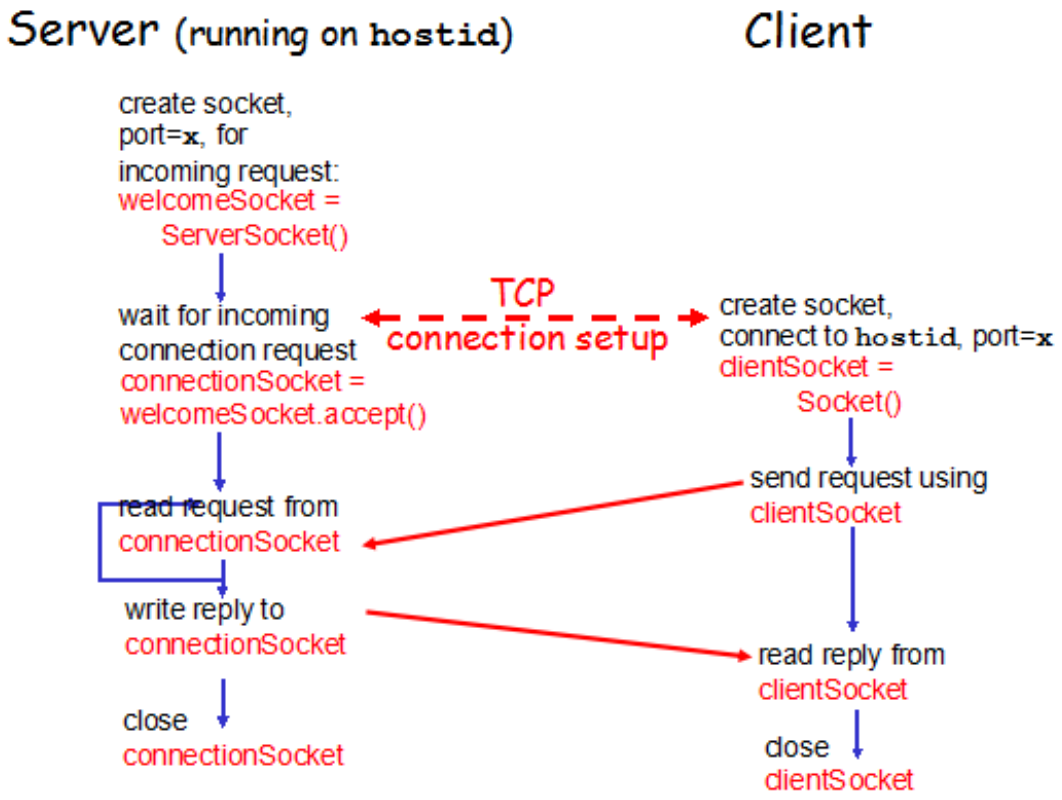
✧ 在创建 TCP 连接的三次握手期间，客户机进程敲服务器进程的欢迎之门。当服务器“听”到敲门时，将创建一个新套接字（连接套接字），为某个特定的客户机程序服务。

✧ 当客户机程序“敲门”时，服务器程序调用 WelcomeSocket 中的 accept（）方法创建新套接字。

✧ TCP 在客户机进程和服务器进程之间提供可靠**字节流**服务。

2.7.2 一个 Java 客户机/服务器应用程序例子

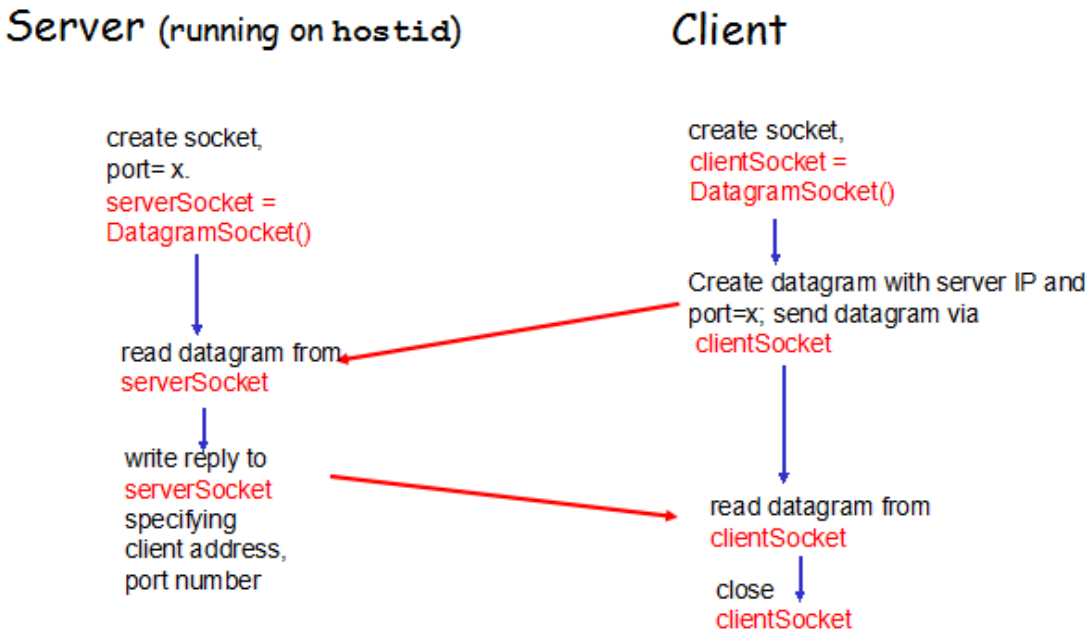
代码见 ppt



2.8 UDP 套接字编程

✧ UDP 是面向报文的（不面向字节）。

代码见 ppt



Chapter 3：运输层

3.1 概述和运输层服务

3.1 概述和运输层服务

- ✧ 运输层协议提供逻辑通信。
- ✧ 运输层协议在端系统而不是网络路由器实现。

3.1.1 运输层和网络层的关系

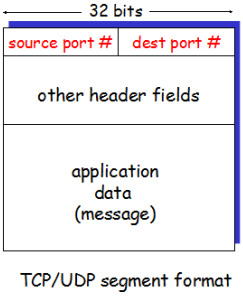
- ✧ 运输层为运行在不同主机上的**进程**提供逻辑通信，网络层为**主机**提供逻辑通信。
- ✧ 运输层协议只工作在端系统，将进程的报文移动到网络边缘（即网络层）。
- ✧ 运输层协议能提供的服务受到了底层网络协议的限制。（若网络层协议不能为两主机间报文段提供时延和带宽保证，那么运输层协议也不能）

3.1.2 因特网运输层概述

- ✧ 因特网是以个 TCP/IP 协议。
- ✧ 因特网网络层协议 IP：网际协议。
  1) 服务模型：“尽力而为”交付服务；
  2) 不确保报文段的交付与报文段的按序交付——不可靠服务。

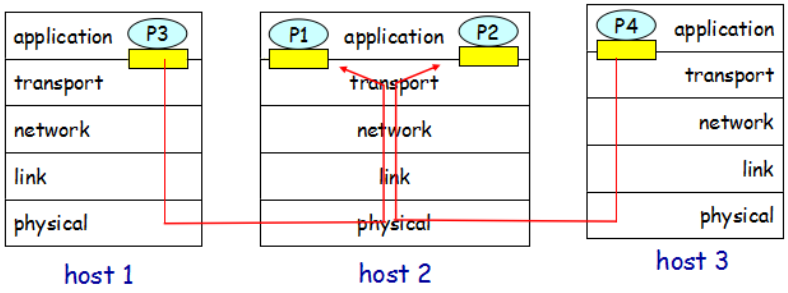
3.2 多路复用和多路分解

- ✧ 多路分解（Multiplexing）：将运输层报文段的数据交付到正确套接字。
- ✧ 多路复用（Demultiplexing）：从源主机不同套接字中收集数据块，并为每个数据块封装上首部信息从而生成报文段，然后将报文段传递到网络层。



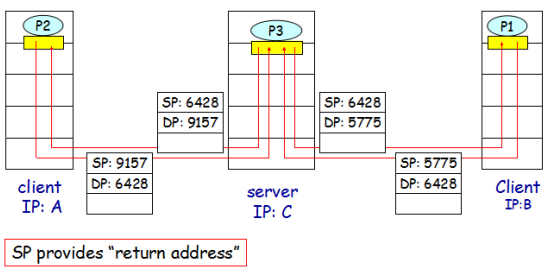
TCP/UDP segment format

- ✧ 运输层多路复用的要求：
  1) 套接字有唯一标识符；
  2) 每个报文段有特殊字段指示该报文段所要交付的套接字。
- ✧ 周知端口号：0~1023，保留给周知应用层协议。
- ✧ 实现多路分解的简单思路：
  1) 主机上的每个套接字被分配一个端口号；
  2) 报文段到达主机时，运输层检查报文段的目的端口号，定向到相应的套接字；
  3) 报文段中的数据经过套接字进入所连接的进程。



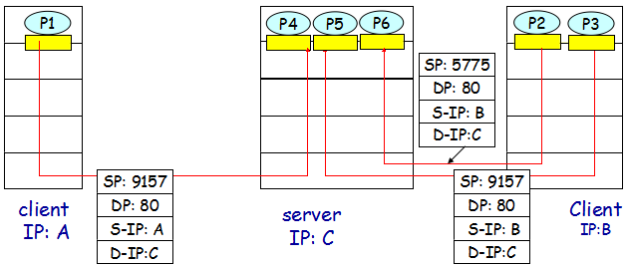
- ✧ 无连接的多路复用和多路分解
  1) UDP 套接字由包含目的 IP 地址和目的端口号的二元组全面标识。
  2) 若两个 UDP 报文段源报文段和/或源端口号不同，但目的 IP 地址和目的端口号相同，那么它们将通过相同的目的套接字定向到相同的目的进程。

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```



✧ 面向连接的多路复用和多路分解

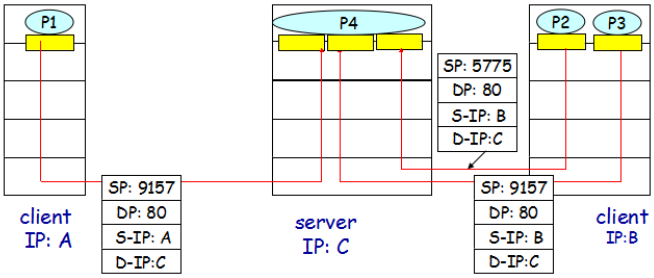
1) TCP 套接字由四元组标识：源 IP 地址、源端口号、目的 IP 地址、目的端口号。
2) 若两个 UDP 报文段源报文段和/或源端口号不同，但目的 IP 地址和目的端口号相同，那么它们将通过相同的目的套接字定向到两个不同套接字。



✧ Web 服务器与 TCP

1) 连接套接字与进程间并非总是一一对应的关系。
2) 当今的高性能 Web 服务器通常只使用一个进程，但是为每个新的客户机连接创建一个具有新连接套接字的新线程。

Threaded Web Server



3.3 无连接运输：UDP

✧ UDP 被称为无连接的原因：发送报文段之前，发送方和接收方的运输层实体之间没有进行握手。

✧ 选择在 UDP 上构建应用的原因：

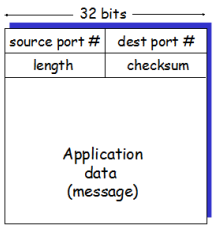
1) 应用层能更好的控制要发送的数据和发送时间：只要应用进程将数据传递给 UDP，UDP 就会将数据打包成 UDP 报文段并立即传递给网络层；
2) 无需连接建立：UDP 不会引入建立连接的时延；
3) 无连接状态：不维护连接状态，也不跟踪这些参数，以便能支持更多的活动客户机；
4) 分组首部开小小。

✧ 75%的按需和实况流式媒体使用了 TCP 的原因：分组丢包率低、某些机构为了安全阻塞 UDP 流量。

✧ UDP 中缺乏拥塞控制机制，可能导致 UDP 发送方和接收方之间的高丢包率，击垮 TCP 会话。

✧ 使用 UDP 的应用是可以实现可靠数据传输的：通过在应用程序自身中建立可靠性机制来完成（如增加确认与重传机制）。但可能需要较长时间调试。

3.3.1 UDP 报文结构



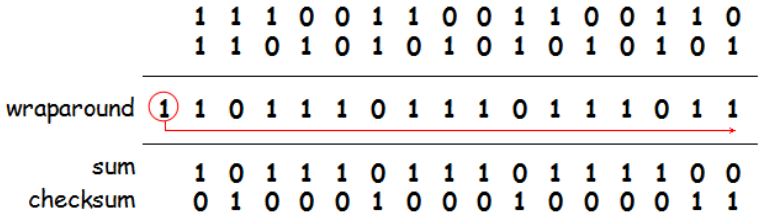
✧ 接收主机使用检验和（checksum）检查报文段是否有差错。

✧ 长度字段指明包括首部在内的 UDP 报文段长度。

3.3.2 UDP 检验和

✧ 发送方对 UDP 对报文段中的所有 16 比特字的和进行反码运算，求和时遇到的任何溢出均被回卷，得到的结果就是 UDP 检验和。

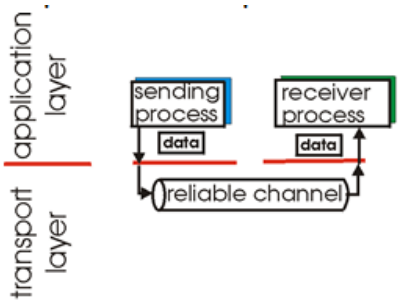
✧ 简单例子：



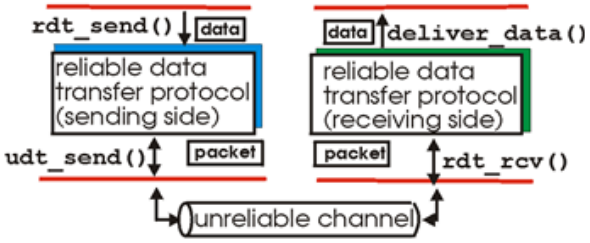
✧ UDP 提供差错检测，但不提供差错恢复。

3.4 可靠数据传输的原理

✧ 服务抽象：数据可以通过一条可靠信道传输。（实际上可靠信道的下层协议也许是不可靠的）



（数据通过可靠信道传输）

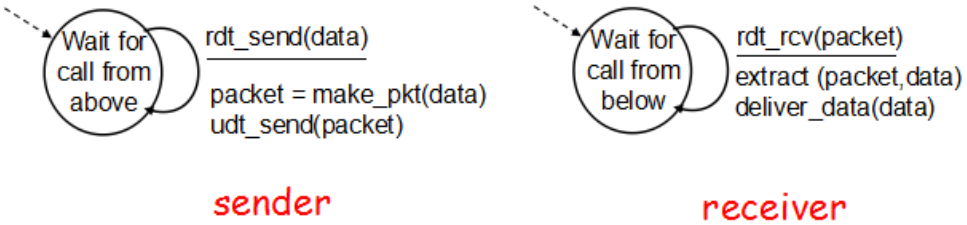


（可靠信道的下层协议也许不可靠）

3.4.1 构造可靠数据传输协议

✧ 完全可靠信道上的可靠数据传输：rdt1.0

1) 考虑底层信道是完全可靠的。
2) FSM：

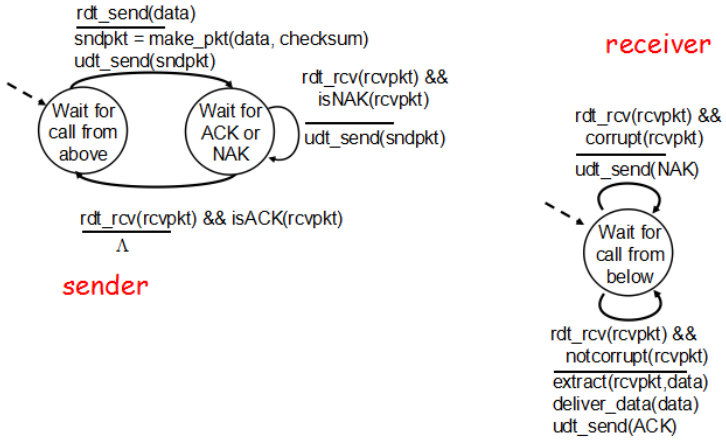


✧ 具有比特差错信道上的可靠数据传输：rdt2.0

1) 假定所有分组按序接收，但可能有比特差错。
2) **使用肯定确认和否定确认。**
3) 自动重传请求（Automatic Repeat reQuest）：接收到否定确认时自动重传。
4) ARQ 协议处理比特差错：

- ◆ 差错检测：需要一种机制检测何时出现了比特差错。
- ◆ 接收方反馈：ACK、NAK。
- ◆ 重传：接收方接收到有差错的分组时，发送方重传。

5) FSM：



6) rdt2.0 被称为停等（stop-and-wait）协议：发送方不会发送一块新数据，直到发送方确信接收方已接收当前分组。

7) rdt2.0 的问题：ACK 和 NAK 分组可能受损。处理受损的 ACK 或 NAK：

- ◆ 发送方接到受损的确认分组时，回复“你说什么”，说话者复述该回复。但可能“你说什么”再次受损，接受者将不明白这个受损的分组是口述内容的一部分还是要求重复上次回答。困难重重。
- ◆ 增加足够的检验和比特，使发送方不仅可以检测差错，还可以恢复。
- ◆ 重发当前分组——引入**冗余分组(duplicate packet)**。冗余分组的困难：接收方不知道上次传输的 ACK 或 NAK 是否被正确收到，因此无法知道当前接收到的分组是新的还是一次重传。

8) rdt2.1：引入对数据分组的编号（对于停等协议，1 比特的序号足够）。FSM 见 ppt。

- ◆ 收到失序分组：对所接受分组发送肯定确认；
- ◆ 收到受损分组：对所接受分组发送 NAK；
- ◆ 不发送 NAK，而是发送对上次正确接收分组的 ACK：与收到 NAK 效果一样（冗余 ACK）。

9) rdt2.2：无 NAK 的可靠数据传输协议。

- ◆ 接收方：包括一个由 ACK 报文确认的分组序号；
- ◆ 发送方：检查接收到的 ACK 报文中所确认的分组序号。

✧ 具有比特差错的丢包信道上的可靠数据传输：rdt3.0（比特交替协议）

1) 倒计数定时器：在一个给定的时间过期后，可中断发送方。

2) 发送方需要能做到：

- ◆ 每发送一个分组（即第一次分组和重传分组）时，启动一个定时器；
- ◆ 响应定时器中断；
- ◆ 终止定时器。

3.4.2 流水线可靠数据传输协议

✧ 停等协议的有效吞吐量十分低。

✧ 采用流水线可靠数据传输协议提高有效吞吐量：

1) 增加序号范围；

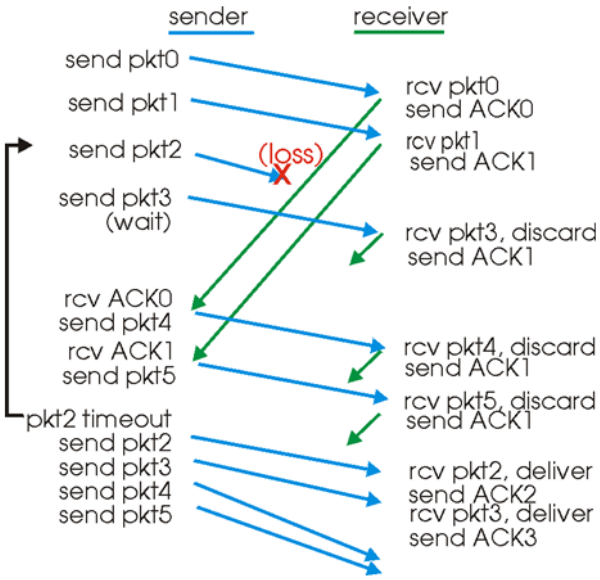
2) 发送方和接收方需要可以缓存多个分组。

3.4.3 回退 N 步

✧ 回退 N 步（Go-Back-N）协议（又被称为滑动窗口协议）中，允许发送方发送多个分组（当有多个分组可用时）而不需等待确认。在流水线中未确认分组不能超过某个最大允许数 N。

✧ GBN **发送方**必须响应三种类型的事件：

1) 上层的调用：上层调用 rdt_send 时，发送方首先检查发送窗口是否已满，即是否有 N 个已发送但未被确认的分组，如果未满，则发送分组，否则发送方返回数据给上层，隐式的通知上层。实际操作中，发送方更可能做的是缓存分组，或使用同步机制——仅当发送窗口未满时允许上层调用。
2) 收到 ACK：对序号为 n 的分组采取累积确认的方式，表明接收方已正确接收 n 以前的所有分组。
3) 超时事件：如果出现超时，则发送方重传所有已发送但未经确认的分组。



✧ GBN **接收方**：如果一个序号为 n 的分组被正确接收，且按序，则接收方为分组 n 发送一个 ACK，并将分组中的数据交付到上层。在所有其他情况下，接收方都将丢弃该分组。（注意：如果分组 k 已接收并正确交付，则所有序号比 k 小的分组都已经交付）

✧ GBN 协议中接收方不需要缓存任何失序分组（发送方察觉后会全部重传），唯一需要维护的信息就是下一个按序接收的分组序号。

✧ GBN 协议的问题：一个单个分组的差错就可能引起 GBN 重传大量分组。

3.4.4 选择重传

✧ 选择重传（Selective Repeat）协议通过让发送方仅重传那些它怀疑在接收方出错（即丢失或受损）的分组而避免了不必要的重传。

✧ SR **发送方**：

1) 从上层收到数据：收到数据后，检查下一个可用于该分组的序号。如果序号在发送方窗口内，则将数据打包并发送；否则将数据缓存或返回上层。
2) 超时：每个分组拥有自己的逻辑定时器。
3) 收到 ACK：收到 ACK 且分组序号在窗口内时，SR 发送方将这个被确认的分组标记为已接收。若该分组序号等于 send_base，则窗口序号向前移动到具有最小未确认分组处。如果窗口移动了并且有序号落在窗口内的未发送分组，则发送这些分组。

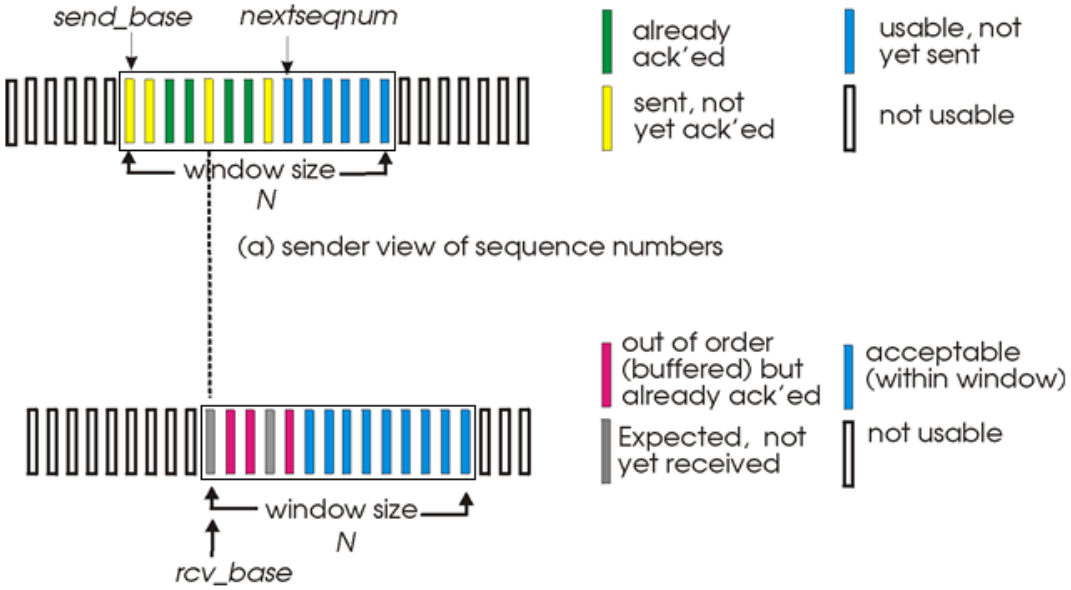
✧ SR **接收方**：

1) 序号在[rcv_base, rcv_base+N-1]内的分组被正确接收：收到的分组落在接收方的窗口内，一个 ACK 分组被发回给发送方。

- ◆ 若分组在之前未被接收过，则被缓存；
- ◆ 若该分组的序号等于接收窗口的继续好，则该分组以及以前缓存的序号连续的分组（交付给上层）。

2) 序号在[rcv_base-N, rcv_base-1]内的分组被正确接受到，产生 ACK，即使该分组是接收方以前确认过的分组。

3) 其他情况，忽略该分组。



✧ SR 协议中，窗口长度必须小于等于序号空间大小的一半。

3.5 面向连接的运输：TCP

3.5.1 TCP 连接

✧ TCP 概览：

1) 点到点（point-to-point）：TCP 连接是单个发送方和单个接收方的连接（多播：一次发送操作中，一个发送方将数据传送给多个接收方）；

2) 可靠、有序的字节流（byte stream）；

3) TCP 连接的每一端都有发送缓存和接受缓存；

4) TCP 连接提供全双工服务（full duplex data）：应用数据可以从进程 B 流向进程 A 的同时，从进程 A 流向进程 B；

5) 面向连接：应用进程彼此发送数据前，必须相互“握手”，建立连接；

6) 流量控制：发送方与接收方的数据传输速度处于平衡状态（sender will not overwhelm receiver）。

✧ 最大报文段长（maximum segment size， MSS）。

3.5.2 TCP 报文段结构

✧ TCP 通常是将文件划分成长度为 MSS 的若干文件块。

✧ TCP 报文段结构：

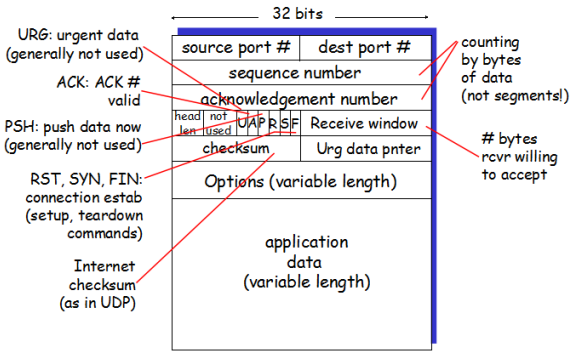
1) 源端口号、目的端口号：用于多路复用/多路分解；

2) 序号字段（32bits）、确认号字段（32bits）；

3) 接收窗口字段：用于流量控制；

4) 首部长度字段：指示首部长度；

5) 选项字段：可选且长度可变，用于发送方和接收方

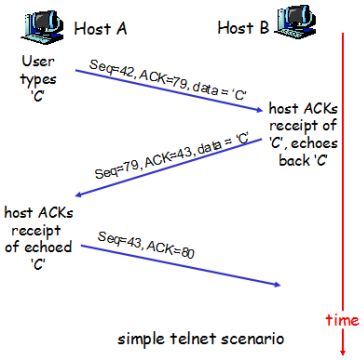


协商最大报文段长度，或在高速网络环境下用作窗口调节因子；

6) 标志字段：
   - ◆ ACK：指示确认字段中的值是有效的；
   - ◆ RST、SYN、FIN：用于连接建立和拆除；
   - ◆ PSH：指示接收方应立即将数据交给上层；
   - ◆ URG：指示报文段里存在着被发送方的上层实体置为“紧急”的数据。
7) 紧急数据指针字段：当紧急数据存在并给出紧急数据尾的指针时，TCP 必须通知接收方的上层实体。

✧ 序号和确认号：

1) TCP 把数据看成一个无结构但是有序的字节流。
2) TCP 的序号建立在传送的字节流之上而不是传送的报文段序列之上。
3) 一个报文段的序号是报文段首字节的字节流编号。
4) 主机 A 填充进报文段的确认号是主机 A 期望从主机 B 收到的下一字节的序号。
5) TCP 提供累计确认。



3.5.3 往返时延的估计和超时

这部分看书就好。

3.5.4 可靠数据传输

✧ TCP 发送方必须处理的三个主要事件：

1) 从上层应用程序接收数据；
2) 定时器超时：通过重传引起超时的报文段来响应超时；
3) 收到 ACK 报文：将 ACK 的值 y 与 SendBase（最早未确认的字节序号）进行比较
   - ◆ y > SendBase：确认一个或多个之前未经确认的报文段（累计确认）；
   - ◆ 如果当前还有未经确认的报文段，重启定时器。

✧ 产生 TCP ACK 的建议：

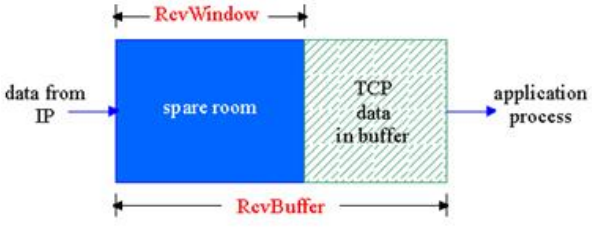
事件	TCP 接收方动作
所期望序号的报文段按序到达，所在期望序号及其以前的数据都已经被确认。	延迟的 ACK。对另一个按序报文段的到达最多等待 500ms，如果在此时间间隔中下一报文段没有到达，则发送一个 ACK。
有期望序号的报文段按序到达。另一个按序报文段等待 ACK 传输。	立即发送单个累计 ACK，以确认两个按序报文段。
比期望序号大的失序报文段到达，检测出数据流中的间隔。	立即发送冗余 ACK，指明下一个期待字节的序号（该间隔的低端字节序号）
能部分或完全填充接收数据间隔的报文段到达	倘若该报文段起始于间隔的低端，则立即发送 ACK。

✧ 一旦收到 3 个冗余 ACK，TCP 就执行快速重传，即在该报文段的定时器过期之前重传丢失的报文段。

3.5.5 流量控制

✧ 流量控制是一个速度匹配服务，即发送方的发送速率与接收方应用程序的毒素率相匹配。

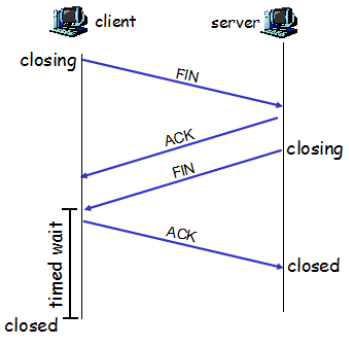
✧ 接收窗口：用于告诉速度发送方接收方还有多少可用的缓存空间。



3.5.6 TCP 连接管理

✧ 客户机向服务器请求建立 TCP 连接（三次握手）：

1) 客户机→服务器：发送 SYN=1，含初始序号的报文段；
2) 服务器→客户机：为该 TCP 连接分配 TCP 缓存和变量，并向客户机发送允许连接的报文段；SYN 被置为 1，服务器选择初始序号；
3) 客户机→服务器：客户机为该 TCP 连接分配缓存和变量，然后向服务器发送另外一个报文段，对服务器允许连接的报文段进行确认，SYN 被置为 0。



✧ TCP 连接终止的步骤：

1) 客户机→服务器：FIN=1；
2) 服务器→客户机：发送确认报文段；
3) 服务器→客户机：发送终止报文段，FIN=1；
4) 客户机→服务器：对服务器的终止报文段进行确认。

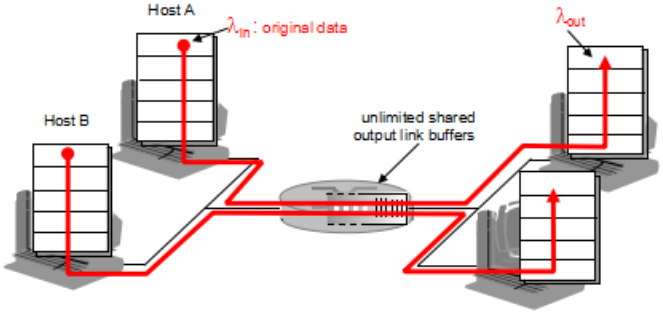
3.6 拥塞控制原理

✧ 分组重传是网络拥塞的征兆。

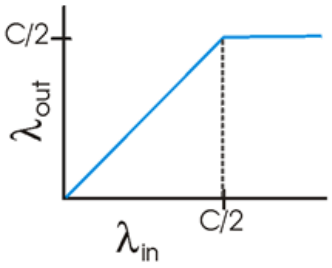
3.6.1 拥塞原因与开销

✧ 两个发送方和一个具有无穷大缓存的路由器：

◆ 主机 A 和主机 B 向路由器提供流量的速率均为 λ_{in}。

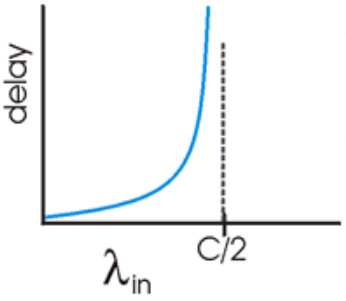


◆ 吞吐量与发送速率的关系：



（看似是好事，因为分组在被交付到目的地的过程中链路被充分利用了）

◆ 时延与发送速率的关系：



（网络拥塞的开销：当分组到达速率接近链路容量时，分组将经历巨大排队时延）

✧ 两个发送方和一个具有有限缓存的路由器：

◆ 有限缓存的路由器：分组到达已满的缓存时会被丢弃。

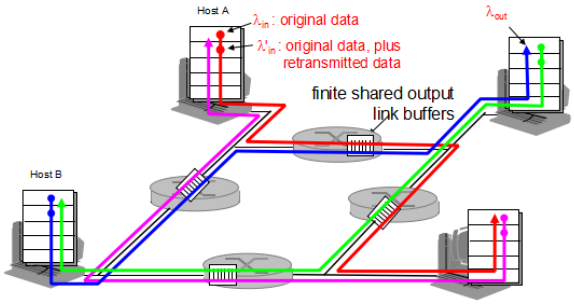
◆ 假定：

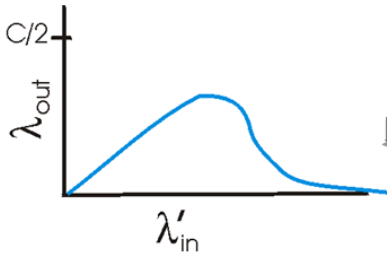
1) λ_{in}：应用程序将初始数据发送到套接字的速率；
2) λ_{in}'：运输层向网络层中发送报文段的速率。

◆ 发送方的重传策略：仅当确定了一个分组丢失时才重传。

◆ 网络拥塞的开销：发送方必须执行重传以补偿因为缓存溢出而丢弃（丢失）的分组。

✧ 四个发送方、具有有限缓存的多台路由器和多跳路径：





◆ 网络拥塞的开销：一个分组沿一条路径被丢弃时，每个上游路由器用于转发该分组到丢弃该分组而使用的传输流量最终被浪费掉了。

3.6.2 拥塞控制方法

✧ 端到端拥塞控制：

◆ 网络层没有为运输层拥塞控制提供显式支持；

◆ 端系统需要对网络行为进行观察以察觉拥塞的存在，例如检测到 TCP 报文段的丢失。

✧ 网络辅助的拥塞控制：

◆ 网络层组件（即路由器）向发送方提供关于网络中拥塞状态的显式反馈信息。

1) 网络路由器发送给发送方直接反馈信息：阻塞分组。

2) 路由器标记或更改从发送方流向接收方的分组中的某个字段来标识拥塞的产生。（至少要经过一整个往返时间）

3.6.3 网络辅助的拥塞控制例子：ATM ABR 拥塞控制

✧ 资源管理信元：用于在主机和交换机之间传递和拥塞相关的信息。

✧ 当 RM 信元到达目的地时，它将被调转方向回送给发送方（内容可能已经更改）。交换机也可直接产生信元，并将其**直接**发送给源。（也即分为直接网络反馈和经由接收方的网络反馈）

✧ ATM ABR 是一种基于速率的办法：发送方明确计算出它所能发送的最大速率，并据此对自己进行调整。

✧ ABR 提供的三种机制：

◆ EFCI 比特：数据信元包含 1 比特的显式转发拥塞指示比特，如果多数到达的数据信元中数据信元中此比特都为 1，则目的地将资源管理信元中的拥塞指示（CI）比特置为 1，并发送给发送方，使发送发得知网络拥塞。

◆ CI 和 NI 比特：拥塞指示（CI）和无增长（NI）比特可被一台拥塞的交换机设置。特别的，交换机在轻微拥塞时将经过的 RM 信元中的 CI 比特置为 1，在严重拥塞的情况下把 NI 比特置为 1。

◆ ER：显式速率字段。拥塞的交换机也许会降低经过的 RM 信元总 ER 的值，在这种方式中，ER 字段被设置为在源至目的地的路径上的所有交换机中的最小可支持速率。

3.7 TCP 拥塞控制

✧ TCP 拥塞控制机制让每一端都记录一个额外的变量：拥塞窗口（congestion window），对 TCP 发送方能向网络中发送流量的速率进行了限制。**特别是，一个发送方中未被确认的数据量不会超过 CongWin 和 RcvWindow 中的最小值。**

✧ TCP 使用确认来触发（或计时）它的拥塞窗口长度的增大，所以 TCP 被称为是自计时的。

✧ TCP 拥塞控制算法：

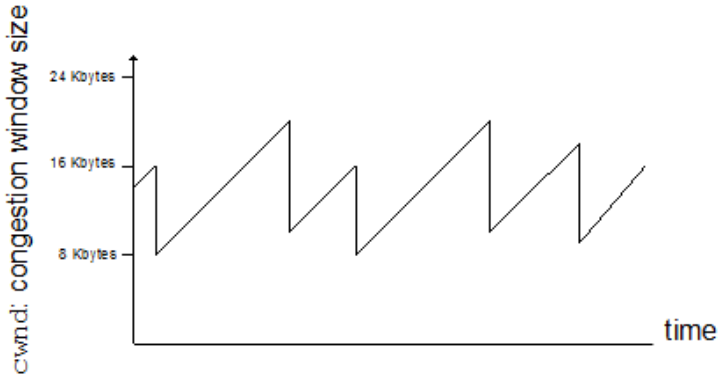
◆ 加性增，乘性减

1) 乘性减：每发生一次丢包时间就将当前的 CongWin 减半。（但是不能低于一个 MSS）

2) 加性增：每次收到一个确认后就把 CongWin 增大一个 MSS——每个往返时延内 CongWin

增大一个 MSS。

3) 避免拥塞（congestion avoidance）：TCP 拥塞控制协议的线性增长阶段。



（加性增、乘性减使得长寿命的 TCP 连接的 CongWin 变化呈锯齿形状）

◆ 慢启动

1) 慢启动：TCP 发送方以慢速率发送，但是以指数的速率增加其发送速率。

2) 慢启动的原因：TCP 发送发初始速率很慢，但可用带宽可能比这个速率大得多，若仅仅线性增加发送速率，在发送速率增加到某个可观的级别可能会经历很长时间。

◆ 对超时时间做出反应

1) 收到三个冗余 ACK：拥塞窗口长度减小一半。

2) 超时：TCP 发送方进入慢启动阶段，将拥塞窗口设置为 1MSS，然后窗口长度以指数速度增长，直到拥塞窗口达到超时前窗口值的一半为止。

3) 阈值（Threshold）：确定慢启动将结束并且拥塞避免开始的窗口长度。

❖ 每发生一个丢包事件，Threshold 被设置为**当前 CongWin 值**的一半。

✧ 对 TCP 拥塞控制算法的讨论：

状态	事件	TCP 发送方拥塞控制动作	注释
慢启动（SS）	收到前面未确认数据的 ACK	CongWin=CongWin+MSS If(CongWin>Threshold)设置状态“拥塞避免”	每过 RTT， CongWin 翻倍
拥塞避免（CA）	收到前面未确认数据的 ACK	CongWin=CongWin +MSS*MSS/CongWin	加性增，每过 RTT 使 CongWin 增加 1 个 MSS
SS 或 CA	由 3 个冗余 ACK 检测到的丢包事件	Threshold=CongWin/2 CongWin=Threshold 设置状态拥塞避免	快速恢复，实现乘性减。CongWin 将不低于 1 个 MSS。
SS 或 CA	超时	Threshold=CongWin/2 CongWin=Threshold 设置状态慢启动	进入慢启动
SS 或 CA	冗余 ACK	对确认的报文段增加冗余 ACK 计数	CongWin 和 Threshold 不变

✧ 快速恢复：收到三个冗余 ACK 后取消慢启动阶段的行为。

✧ TCP 吞吐量的宏观描述：

一个连接的平均吞吐量： 0.75*W / RTT （W：丢包时的拥塞窗口长度）

✧ UDP 源压制 TCP 流量的现象可能发生——开发研究因特网拥塞控制机制，阻止 UDP 流量压制直至中断因特网吞吐量。

第四章 网络层

4.1 概述

✧ 路由器的主要作用：将数据包从入链路转发到出链路。

4.1.1 转发和选路

✧ 网络层功能：

1) 转发：分组从一个输入链路接口转移到适当的输出链路接口的路由器本地动作。
2) 选路：分组从源到目的地时，决定端到端路径的网络范围的进程。
3) 连接建立：某些网络层体系结构而非因特网，要求从源到目的地沿着所选择的路径彼此握手，以便在网络层数据分组开始流动之前，给定的源到目的地之间建立起连接状态。

4.1.2 网络服务模型

✧ 网络层提供的特定服务：

1) 确保交付；
2) 具有时延上界的确保交付；
3) 有序分组交付；
4) 确保最小带宽；
5) 确保最大时延抖动：确保发送方发送的两个相继分组之间的时间量等于在目的地接收到它们之间的时间量；
6) 安全性服务。

4.2 虚电路和数据包网络

✧ 网络层连接和无连接服务与运输层连接和无连接服务的区别：

1) 在网络层中，这些服务是主机到主机之间的服务；在运输层中，这些服务是进程到进程之间的服务。
2) 在网络层中，不同时提供连接和无连接服务；而在运输层中，可同时提供这两种服务。
3) 运输层面向连接服务仅在网络边缘的端系统中实现；网络层的连接服务除了在端系统中实现，还在位于网络核心的路由器中实现。

4.2.1 虚电路网路

✧ 虚电路的组成：

1) 源和目的主机之间的路径（一系列链路和路由器）；
2) VC 号，沿着该路径的每段链路一个号码；
3) 沿着该路径的每台路由器的转发表表项。

✧ 为什么一条分组沿着其路由在每条链路上不能保持相同的 VC 号：

1) 逐链路代替该号码减少了分组首部中的 VC 字段的长度；
2) 简化虚电路的建立。

若沿着某路径的所有链路需要一个共同的 VC 号的话，路由器将不得不交换并处理大量的报文，以约定一个共同的 VC 号。

✧ 在虚电路网络中，路由器必须为进行中的连接维持连接状态信息。

1) 每当建立一个新连接，将一个新的连接项加到该路由器转发表中；
2) 每当释放一个连接，从该表中删除该表项。

✧ 虚电路建立的三个阶段：

1) 虚电路建立：
   - ◆ 发送运输层与网络层联系，指定接收方地址；
   - ◆ 网络层决定发送方与接收方之间的路径
   - ◆ 网络层为每条链路确定一个 VC 号；

- ◆ 网络层在沿着路径的每条路由器的转发表中增加一项；
- ◆ 虚电路连接期间，网络层预留虚电路路径上的资源（如带宽）。

2) 数据传送；

3) 虚电路拆除：

- ◆ 网络层通知网络另一侧的端系统结束呼叫；
- ◆ 更新路径上每台路由器中的转发表。

✧ 运输层的连接建立仅涉及两个端系统，网络中的路由器对这个连接其实毫不知情。

✧ 网络层中的虚电路建立时，不仅端系统参与虚电路建立，且每台路由器都知道经过它的所有虚电路。

✧ 信令报文：端系统向网络发送指示虚电路启动与终止的报文，以及路由器之间传递的用于建立虚电路（即修改路由器表中的连接状态）的报文。

✧ 信令协议：用来交换信令报文的协议。

4.2.2 数据报网络

✧ 数据报网络无需建立任何虚电路，路由器不维护任何有关虚电路的有关信息。

✧ 分组从源到目的地的传输过程中，每台路由器通过转发表将目的地址映射到输出链路——通过将目的地址的**前缀**与转发表中的表项相匹配。

✧ 最长前缀匹配规则：在转发表中寻找最长的匹配项。

✧ 数据报网络中的路由器不维持**连接状态信息**，但在其转发表中维持了**转发状态信息**。

✧ 数据报网络中的转发表周期性更新，因此从一个端系统到另一个端系统的一系列分组可能在通过网络时走不通的路径，并可能无序到达。

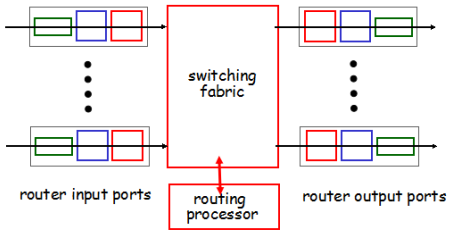
4.2.3 虚电路和数据报网络的由来

4.3 路由器工作原理

✧ 路由器的组成部分：

1) 输入端口：

- ◆ 将一条输入的物理链路端接到路由器的物理层功能；
- ◆ 执行需要与位于入链路远端的数据链路层功能交互的数据链路层功能；
- ◆ 完成查找与转发功能。



2) 交换结构：将输入端口连接到输出端口，是一台路由器中的网络。

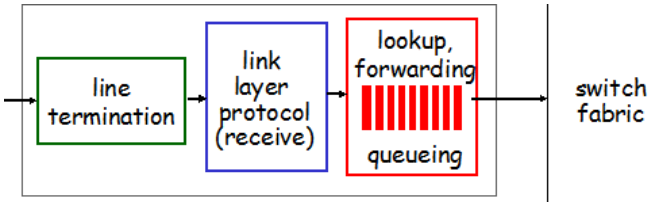
3) 输出端口：执行与输入端口相反的物理层功能。

4) 选路处理器：

- ◆ 选路处理器执行选路协议；
- ◆ 维护选路信息于转发表；
- ◆ 执行路由器中的网络管理功能。

4.3.1 输入端口

✧ 通常一份转发表的影子拷贝将被存放在每个输入端口，且更新——可在输入端口做出转发决策，而不调用中央选路处理器。



✧ 人们希望输入端口的处理速度能够达到线路速度，即执行一次查找的时间应少于从输入端口接收一个分组所需的时间。

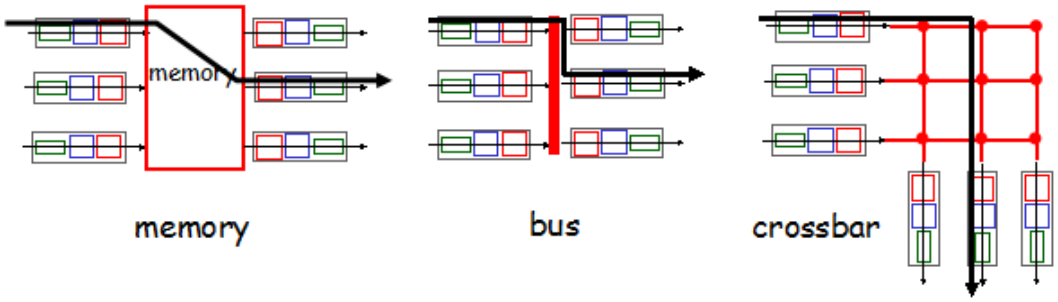
✧ 分组可能会在进入交换结构时暂时阻塞——由于来自其他输入端口的分组当前正在使用该交换结构。

4.3.2 交换结构

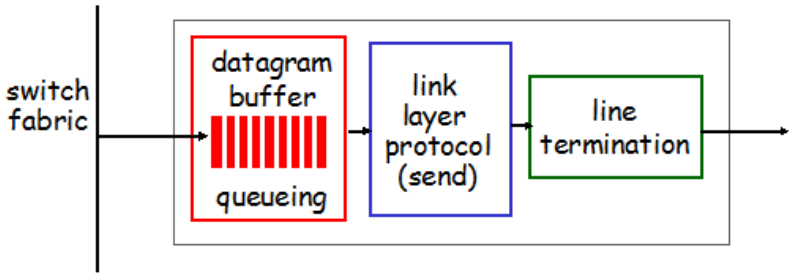
✧ 通过交换结构，分组从输入端口转发哒输出端口。

✧ 交换技术：

1) 经内存交换：输入端口到输出端口之间的交换是在 CPU（选路处理器）的直接控制下完成。输入端口和输出端口类似于传统操作系统中的 I/O 设备。
2) 经一根总线交换：输入端口经一根共享总线将分组直接传送到输出端口，不需选路处理器的干预。——一次只能有一个分组通过总线传送，路由器的交换带宽受总线速率的限制。
3) 经一个互联网络交换：一个到达某个输入端口的分组沿着输入端口的水平总线串行，直至该水平总线与连到所希望的输出端口的垂直总线的交叉点。



4.3.3 输出端口



4.3.4 何时出现排队

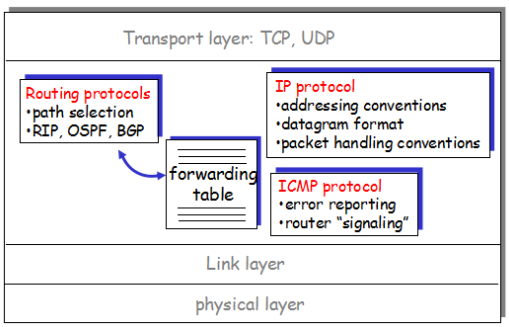
✧ 输入端口和输出端口处都能形成分组队列。

✧ 分组丢失的实际位置取决于流量负载、交换结构的相对速率和线路速率。

✧ 在输出端口产生丢包：当交换结构到达输出端口的分组速度大于输出端口的线性速度，分组缓存，持续缓存直至超过输出端口缓存时，产生丢包。

✧ 线路前部阻塞（Head-Of-the-Line，HOL）：在一个输入队列中排队的分组必须等待通过交换结构发送（即使输出端口是空闲的）。

4.4 网络协议：因特网中的转发和编址



✧ 因特网网络层组件：

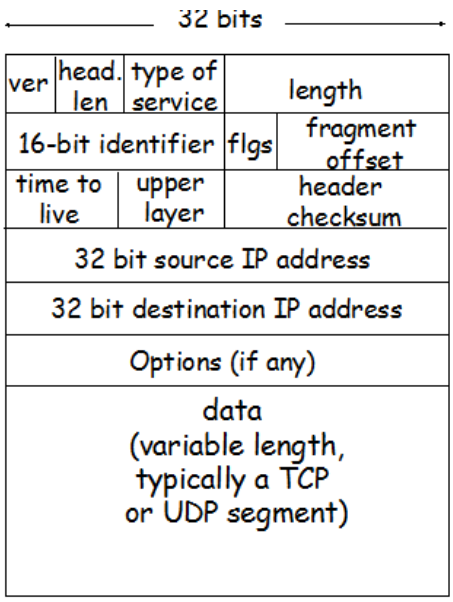
1) IP 协议；
2) 选路组件：决定数据报从源到目的地所流经的路径；
3) 报告数据报中的差错和对某些网络层信息请求进行响应的设施。

4.4.1 数据报格式

✧ IPv4 数据报中的关键字段：

1) 版本号：规定数据报 IP 协议的版本，不同的 IP 版本使用不同的数据报格式。
2) 首部长度：由于 IPv4 包含一些可选项，故需用首部 4 个比特确定 IP 数据报中的数据部分实际的开始位置。

3) 服务类型：区别不同类型的 IP 数据报，例如将实时数据报与非实时流量区分开。
4) 数据报长度：首部长度加上数据长度，IP 数据报理论上最大的长度为 65535.
5) 标识、标志、片偏移：与 IP 分片有关。
6) 寿命（Time-To-Live，TTL）：确保数据报不会永远在网络中循环。每经过一台路由器，TTL 值减 1，直到该值为 0 时，该数据报必须丢弃。
7) 协议：协议号的作用类似于运输层报文段中端口号字段的作用，是将网络层与运输层绑定到一起的粘合剂——指明了 IP 数据报的数据部分应交给哪个运输层协议。协议号为 6——TCP，协议号为 17——UDP。
8) 首部检验和：帮助路由器检测收到的 IP 数据报中的比特错误。**计算方法：**每两个字节当作一个数，用反码运算对这些数求和。**路由器一般会丢弃出错的数据报。**



9) 源和目的 IP 地址：源 IP 字段插入源主机的 IP 地址，目的 IP 地址字段插入最终目的地址。
10) 选项。
11) 数据（有效载荷）：包含要交付给目的地的运输层报文段，也可承载其他类型的数据，如 ICMP 报文段等。

✧ IP 数据报分片：

1) 并不是所有链路层协议都能承载长度相同的网络层分组（有些协议可以承载大分组，而有些协议只能承载小分组）。
2) 最大传输单元（Maximum Transmission Unit，MTU）：链路层帧能承载的最大数据量。
3) 数据报的重新组装工作在端系统而非路由器中完成——确保路由器的性能，降低工作复杂性。
4) 组装数据报的具体做法：
   - ◆ 标识（identification）：在数据报设置源和目的地址的同时加上标识号，发送主机发送的每个数据报中的标识号加一；
   - ◆ 标志（flag）：数据报需要分片时，形成的数据报附加上初始数据报的源地址、目的地址和标识号。为确保目的主机绝对相信接收到初始数据报的最后一个片，将一个片的标志号设置为 0，其他数据报的标志号设置为 1。
   - ◆ 偏移字段（offset）：指定片应放在初始 IP 数据报的哪个位置——确定一个片是否丢失。
5) 在目的地，数据报的有效载荷仅在 IP 层已完全重构为初始 IP 数据报时，才被传递给目的地运输层。
6) 分片的问题：
   - ◆ 使路由器和端系统更为复杂，即必须将数据包分成适当大小的片并且需要重新组装；
   - ◆ 可能引发致命的 DoS 攻击，例如 Jolt2 攻击、发送交迭的 IP 分片等等方式。

4.4.2 IPv4 编址

✧ 接口（interface）：主机与物理链路之间的边界、路由器与它的任意一条链路之间的边界。

1) 路由器有多个接口；
2) 主机只有唯一接口；
3) 每个接口都拥有自己的 IP 地址。

✧ 点分十进制记法（dotted-decimal notation）：地址中的每个字节用十进制形式书写，各字节间以句号（点）隔开。

✧ 子网：

1) 具有相同子网掩码（IP 地址的高字节）的设备接口；
2) 在同一子网中的设备可通过一个不包含路由器的网络互连起来。

✧ 为了确定子网，分开主机和路由器的每个接口，从而产生了几个分离的网络岛，接口端接了这些独立的网络的端点。这些独立的网络中的每一个都叫做一个子网。

✧ 因特网的地址分配策略——无类别域间选路（Classes Interdomain Routing，CIDR），IP 地址被划分为两部分：a.b.c.d/x，其中 x 指示了在地址的第一部分中的比特数。

✧ IP 广播地址：255.255.255.255——主机发出一个目的地址为 255.255.255.255 的数据报时，该报文会被交付到同一个子网中的所有主机。

✧ 设备如何从本组织地址块中分配到一个地址：

1) 获取一块地址：
   - ◆ 与 ISP 联系获取一组地址；
   - ◆ 因特网名字与号码分配机构分配地址——分配 IP 地址，管理 DNS 根服务器，分配域名与解决域名纷争。
2) 获取主机地址：动态主机配置协议：
   - ◆ 路由器中的 IP 地址：系统管理员手工配置；
   - ◆ 主机 IP 地址：动态主机配置协议（Dynamic Host Configuration Protocol， DHCP），又称为即插即用协议（plug-and-play protocol）。
     - ❖ 主机可自动获取 IP 地址，且通过配置 DHCP，主机可在每次连接网络时获取相同或临时 IP 地址。
     - ❖ DHCP 允许一台主机获取其他信息，例如子网掩码、第一条路由器地址（默认网关）、本地 DNS 服务器地址等。
     - ❖ DHCP 服务器需更新其可用 IP 地址表——主机加入时，服务器从当前可用地址池分配一个任意地址给它；主机离开时，其地址被回收到池中。
     - ❖ DHCP 是客户机/服务器协议——客户机通常是新到达的主机，服务器时 DHCP 服务器或 DHCP 中继代理（通常为路由器）。
     - ❖ DHCP 协议工作的四个步骤：
       a. **DHCP 服务器发现**——新到达主机寻找一个与其交互的 DHCP 服务器。**方法：**客户机生成包含 DHCP 发现报文的 IP 数据报，使用广播目的笛子 255.255.255.255 并且使用源地址 0.0.0.0，将其广播到所有与该子网连接的子网。
       b. **DHCP 服务器提供**——DHCP 服务器接收到 DHCP 发现报文时，使用 IP 广播地址发送 DHCP 提供报文。提供报文中含有收到的发现报文的事务 ID、向客户机推荐的 IP 地址、网络掩码以及 IP 地址租用期（IP address lease time，即 IP 地址有效的时间量）。
       c. **DHCP 请求**——新到达主机从一个或多个服务器中选择一个，并用一个 DHCP 请求报文对选中的服务器进行响应，回显配置参数。
       d. **DHCP ACK**——服务器响应 DHCP 请求报文，证实所要求的参数。
3) 网络地址转换（NAT）：
   - ◆ 具有专用地址的地域：地址仅对网络中的设备有意义的网络。
   - ◆ NAT 路由器对外界的行为如同一个具有单一 IP 地址的单一设备。
   - ◆ 使用 NAT 的原因：
     - ❖ 不再需要 ISP 分配一块连续地址，对所有设备仅使用一个 IP 地址；
     - ❖ 可以再不通知外界的情况下改变内网中设备的 IP 地址；
     - ❖ 可以再不改变内部设备地址的情况下更改 ISP；
     - ❖ 内网中的设备将不被外界显示的定位查找（安全性考虑）。

- NAT 转换表：包含端口号和 IP 地址。
- NAT 路由器的工作过程：
  - 用户向 LAN 中发送数据报：NAT 路由器收到该数据报，为该数据报生成一个新的端口号，将其源 IP 地址修改为广域网一侧接口的 UP 地址，且将源端口号换为生成的新端口号；
  - NAT 路由器在其转发表中新增这一项转换记录；
  - 当数据报被发回切到达 NAT 路由器时，路由器使用当前数据报中的 IP 地址与端口号在转发表中进行检索，将其更换为内网中该设备的正确 IP 地址与端口号，向该设备发送数据报。
- NAT 备受争议：
  - 端口号适用于编址进程而非主机；
  - 路由器应当处理最多达第三层的分组；
  - NAT 协议违反了所谓端到端原则，主机间应相互直接对话，而不应介入修改 IP 地址与端口号；
  - 应使用 IPv6 来解决 IP 地址短缺的问题，而不是不计后果的使用 NAT 之类的权宜之计。

4) UPnP

- UPnP：允许主机发现并配置邻近 NAT 的协议。
- 使用 UPnP，一台主机上的应用程序能够为某些请求的公共端口号请求一个 NAT 映射，该映射位于（专用 IP 地址，专用端口号）和（公用 IP 地址，公用端口号）之间。如果 NAT 接受该请求并生成该映射，则来自外部的节点能够发起到（公用 IP 地址，公用端口号）的 TCP 连接。

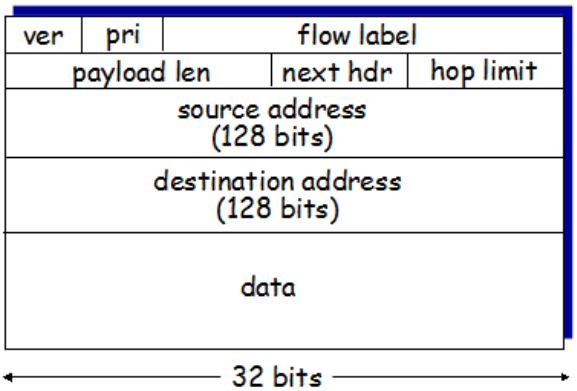
4.4.3 ICMP：互联网控制报文协议

- ICMP：用于主机和路由器彼此交互网络层信息。
- ICMP 最典型的用途是差错报告。
- ICMP 报文：
  1) 类型字段。
  2) 编码字段。
  3) 引起该 ICMP 报文首次生成的 IP 数据报。
- Traceroute 通过 ICMP 报文实现，具体见书 P230。

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

4.4.4 IPv6

- IPv6 数据报格式：



- IPv6 引入的变化在其数据报格式中：
  1) 扩大的地址容量：IP 地址长度从 32 比特增长到 128 比特。引入**任播地址**——使一个数据报能交付到一组主机中的任意一个。

2) 简单高效的 40 字节首部。
3) 流标签与优先级：流标签——给属于特殊流的分组加上标签，这些特殊流是发送方要求进行特殊处理的流，如一种非默认服务质量或需要实时服务的流。

✧ IPv6 中定义的字段：
1) 版本号；
2) 流量类型；
3) 流标签；
4) 有效载荷长度；
5) 下一个首部：标识数据报中的内容需要交付给哪个协议；
6) 跳限制：转发数据报的每台路由器对该字段的内容减 1；
7) 源和目的地址；
8) 数据。

✧ IPv6 不允许在上对路由器进行分片和组装。若数据报因太大而无法转发到输出链路上，该数据报将被丢弃，且路由器向发送发发回一个“分组太大”的 ICMP 差错报文即可。

✧ 从 IPv4 向 IPv6 的迁移：
1) 宣布一个标志日，关闭所有机器进行升级；
2) 双栈（dual-stack）：IPv6 节点也具有完整的 IPv4 实现，这样的节点具有发送和接收两种数据报的能力。但当发送方和接收方其中一个仅是 IPv4 使能时，只能使用 IPv4 数据报。
3) 建隧道（tunneling）：隧道——假定两个 IPv6 节点要使用 IPv6 数据报进行交互，但它们之间是经由中间 IPv4 路由器而互连的，则中间 IPv4 路由器集合被称为隧道。**具体做法：**在隧道发送端的 IPv6 节点，将整个 IPv6 数据报放到一个 IPv4 数据报中，到达隧道接收端时，IPv6 节点再从中取出 IPv6 数据报。

4.4.5 IP 安全性概述

4.5 选路算法

✧ 选路算法在网络路由器中运行、交换和计算，以配置路由器中转发表的信息。

✧ 默认路由器（default router），或称第一条路由器（first-hop router）：与主机直接相连接的路由器。

✧ 源/目的路由器（source/destination router）：源/目标主机的默认路由器。

✧ 广义上的一种选路算法分类：
1) 全局选路算法：以所有节点之间的连通性及所有链路的费用为输入。通常又可成为链路状态（Link-State）算法。
2) 分布式选路算法：以迭代的、分布式的方式计算出最低费用路径。每个节点仅有与其直接相连链路的费用知识即可开始工作，通过迭代计算过程并与相邻节点交换信息，一个节点逐渐计算出到达目的节点或一组目的节点的最低费用路径。

✧ 广义上的一种选路算法分类：
1) 静态选路算法：随时间推移，路由的变化非常缓慢，通常由于人工干预调整。
2) 动态选路算法：能够在网络流量负载或拓扑发生变化时改变选路路径，可周期性地运行或直接地响应拓扑或链路费用的变化而运行。

✧ 广义上的一种选路算法分类：
1) 负载敏感算法：链路费用动态变化以反映出底层链路的当前拥塞水平。
2) 负载迟钝算法：某条链路的费用不明显地反映当前（或最近）拥塞水平。

4.5.1 链路状态选路算法

✧ 获知网络拓扑和所有链路费用的方法：每个节点向网络中的所有其他路由器广播链路状态分组来完成的。

✧ Dijkstra 算法：

1) 定义如下记号：
   - ◆ D（v）：随着算法进行本次迭代，从源节点到目的节点 v 的最低费用路径的费用。
   - ◆ p（v）：从源节点到目的节点 v 沿着当前最低费用路径的前一节点。
   - ◆ N'：节点子集；如果从源节点到目的节点 v 的最低费用路径已确知，v 在 N'中。
2) 通过为每个目的节点存放从 u 到它的最低费用路径上的下一跳节点，可以构建一个节点（如节点 u）的转发表。
3) 算法复杂性：最坏情况下 O（n^2）。
4) 算法可能出现的问题：震荡。
5) 震荡的解决方法：
   - ◆ 强制链路费用不依赖于所承载的流量。（不可接受）
   - ◆ 确保并非所有的路由器都同时运行 LS 算法。

```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v adjacent to u
5        then D(v) = c(u,v)
6     else D(v) = ∞
7
8   Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N' :
12     D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    shortest path cost to w plus cost from w to v */
15  until all nodes in N'
```

4.5.2 距离向量选路算法

✧ 距离向量选路算法：

1) 分布式的：每个节点都要从一个或多个直接相连的邻居接收某些信息，执行计算，然后将计算结果发回给邻居。
2) 迭代的：（1）中的过程持续到邻居之间没有更多的信息要交换为止。
3) 异步的：不要求所有节点相互之间步伐一致地操作。

✧ Bellman-Ford 方程：

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

1) d_x（y）：节点 x 到节点 y 的最低费用路径的费用；
2) $\min_v$：取遍 x 所有的邻居。
3) 方程的意义：从 x 到 y 的最低费用是对所有邻居 v 的 c（c，v）+d_v（y）的最小值。

✧ 距离向量选路算法基本思想：

1) 每个节点不时向每个邻居发送它的距离向量拷贝；
2) 当某节点收到其任意一个邻居的新距离向量时，使用 Bellman-Ford 方程更新其距离向量；
3) 所有节点持续以异步方式交换他们的距离向量时，每个费用估计 D_x（y）将会收敛到 d_x（y）——从节点 x 到节点 y 的实际最低费用路径的费用。

✧ 距离向量算法：链路费用变化与链路故障

- ◆ Good news travels fast. Bad news travels slow——选路环路（routing loop）和计数到无穷问题（count-to-infinity）。

✧ 距离向量算法：增加毒性逆转（poisoned reverse）——解决选路环路及计数到无穷问题。

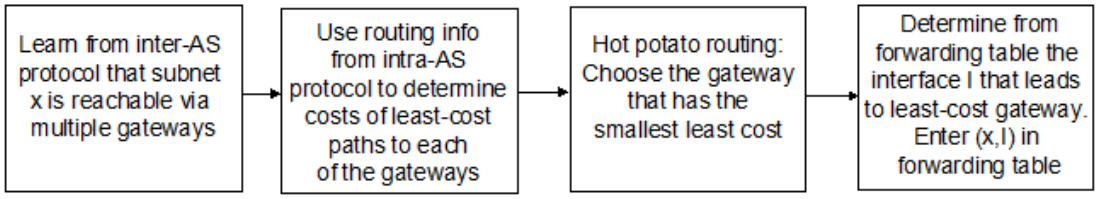
✧ LS 与 DV 选路算法的比较

	LS	DV
报文复杂性	发送 O（NE）个报文，且一条链路费用改变时，必须向所有节点发送新的链路费用。	在邻居间交换报文，收敛所需时间依赖诸多因素。
收敛速度	发送 O（NE）个报文的复杂度为 O（N^2）的算法。	收敛较慢，会遇到选路环路、计数到无穷等问题。

健壮性（一台路由器发生故障时）	路由计算在某种程度上是分离的，提供了一定的健壮性。	一个节点可向任意或所有目的节点通告不正确的最低费用路径，引起大量流量流向故障路由器的现象。

4.5.3 层次选路

- LS 和 DV 算法略微理想化：
  1) **规模**：随着路由器数目增多，选路信息的计算、存储及通信的开销将高的惊人。广播 LS 更新的开销将导致没有剩余的带宽供发送数据分组使用，距离向量算法在大规模的路由器中的迭代将永远不会收敛。
  2) **管理自治**：一个组织应当能够按自己的愿望运行和管理其网络，且还能将其网络与其他外部网络连接。
- 解决办法：自治系统（Autonomous System，AS）
  1) 每个 AS 由一组在相同管理控制下的路由器组成；
  2) 相同 AS 内的路由器执行同样的选路算法——自治系统内部选路协议；
  3) AS 彼此互连——AS 内一台或多台路由器（称为网关路由器，gateway router）负责向本 AS 外的目的地转发分组。
- AS 仅有一个网关路由器连接一个其他 AS 时：网关路由器一旦接收到分组，就将其向通往外部 AS 的一条链路转发，该链路另一端的 AS 为该分组向最终目的地选路。
- AS 有两条或更多条链路通往外部 AS 时：
  1) 从相邻 AS 获取可达性信息；
  2) 向本 AS 中所有路由器传播可达性信息。
- 热土豆选路（hot potato routing）：AS 尽可能快（尽可能便宜）的摆脱分组。



4.6 因特网中的选路

- 选路协议的任务：确定数据报在源与目的地之间采用的路径。

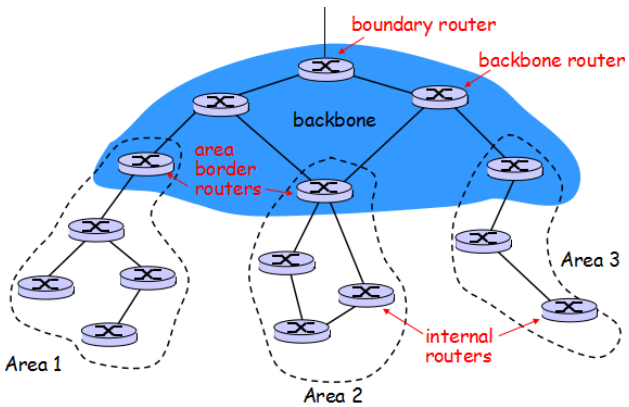
4.6.1 因特网中自治系统内部选路：RIP

- AS 内部选路协议又称为内部网关协议（interior gateway protocol），历史上广泛应用于因特网上的选路协议：
  1) 选路信息协议（Routing Information Protocol，RIP）
  2) 开放最短路径优先（Open Shortest Path First，OSPF）
- RIP 是一种距离向量协议。
- 跳：沿着从源路由器到目的子网（包括目的子网）的最短路径所经过的子网数量。
- RIP 被限制在网络直径不超过 15 跳的自制系统内，也即一条路径的最大费用被限制为 15。
- 在 RIP 中，选路更新信息在邻居之间通过使用 **RIP 响应报文**（RIP response message）交换，大约 30 秒交换一次。
- 响应报文又被称作 RIP 通告（RIP advertisement）。
- 如果一台路由器超过 180 秒没有监听到其邻居，则该邻居不再被认为是可达的，此时 RIP 修改本地选路表，然后通过向相邻路由器发送通告来传播该信息。
- 路由器在 **UDP** 使用端口 520 发送 RIP 请求和响应。

- RIP 是一个运行在 UDP 上的应用层协议。

4.6.2 因特网中 AS 内部选路：OSPF

- RIP 被设置在较低层 ISP 和企业网中，OSPF 与 IS-IS 被设置在较顶层的 ISP 中。
- OSPF 中的开放（OPEN）是指选路协议规约是公众可用的。
- OSPF 算法的核心：使用洪泛链路状态信息的链路状态协议和一个 Dijkstra 最低费用路径算法。
- OSPF 算法中，各条链路的费用是由网络管理员配置的：
  1) 设置每条链路费用为 1，实现最少跳数选路。
  2) 将链路权值与链路容量成反比来设置，从而不鼓励流量使用低带宽链路。
- OSPF 算法中，路由器向自制系统内所有其他路由器广播选路信息：
  1) 一条链路的状态发生变化时，路由器广播状态信息；
  2) 链路状态未发生变化时，周期性广播链路状态信息。——增强了链路状态算法的健壮性。
- OSPF 通告包含在 OSPF 报文中，该报文直接承载在 IP 分组中。——OSPF 协议必须自己实现可靠报文传输、链路状态广播等功能。
- OSPF 的优点：
  1) 安全：路由器之间的交换都是经过鉴别的，意味着仅有受新任的路由器能参与一个 AS 内的 OSPF 协议。两类鉴别方式——简单鉴别与 MD5 鉴别。
  2) 多条相同费用的路径：当存在多条路径具有相同费用时，允许使用多条路径，而不是选择单一的路径来承载所有的流量。
  3) 对单播选路与多播选路的综合支持。
  4) 支持在单个选路域内的层次结构：OSPF 具有按层次结构构造一个自治系统的能力。
- **层次选路的实现：**一个 OSPF 自治系统可以配置成多个区域，每个区域内都运行自己的 OSPF 链路状态选路算法——
  1) 一个区域内的每台路由器都向该区域内的所有其他路由器广播器广播其链路状态；
  2) 一个区域的内部细节对于该区域外的所有路由器都是不可见的；
  3) 在一个区域内，一台或多台区域边界路由器（area border router）负责为发送到该区域外的分组选路；
  4) 主干区域：一个 OSPF 区域配置成主干区域，为 AS 内其他区域之间的流量选路，该主干区域包含所有区域边界路由器，还可能包括无边界路由器。



- OSPF 路由器的分类：
  1) 内部路由器：位于非主干区域，只执行 AS 内部选路；
  2) 区域边界路由器：同时属于区域与主干两个部分；
  3) 主干路由器（非边界路由器）：执行主干中的选路，但自身不是边界路由器；
  4) 边界路由器：与属于其他自治系统的路由器交换选路信息。

4.6.3 自治系统间的选路：BGP

- 边界网关协议（Border Gateway Protocol，BGP）：跨越 AS 的源和目的对之间确定路径。
- BGP 为每个 AS 提供：
  1) 从相邻 AS 处获得子网可达性信息；
  2) 向该 AS 内部的所有路由器传播这些可达性信息；
  3) 基于可达性信息和 AS 策略，决定到达子网的“好”路由。

4) 允许每个子网向因特网的其余部分通告它的存在，确保 AS 知道该子网及如何到达——避免每个子网隔离。

✧ BGP 基础：

1) 路由器对使用 179 端口的半永久 TCP 连接来交换选路信息，这样的连接位于：
   - ◆ 两个不同的 AS 中的路由器链路——eBGP（external BGP session，外部 BGP 会话）会话；
   - ◆ 一个 AS 中的路由器之间——iBGP 会话（internal BGP session，内部 BGP 会话）。
2) 在 BGP 中，目的地不是主机而是 CDIR 化的前缀，每个前缀表示一个子网或一个子网的集合。
3) 任何 AS 中的网关路由器接收到 eBGP 学习到的前缀后，该网关路由器使用它的 iBGP 会话来向该 AS 中的其他路由器发布这些前缀。

✧ 路径属性和 BGP 路由

1) 在 BGP 中，一个自治系统有其全局唯一的自治系统号（Autonomous System Number，ASN）。
2) 带有属性的前缀被称为一条路由。
3) BGP 对等方彼此通告路由。
4) 两个较为重要的属性：
   - ◆ AS-PATH：包含前缀的通告已经通过的 AS。当一个前缀传送到一个 AS 时，该 AS 将它的 ASN 增加到 AS-PATH 属性中。——用于检测和防止循环通告，特别是，如果一台路由器看到它的 AS 包含在该路径列表中，它将拒绝该通告。
   - ◆ NEXT-HOP：开始某 AS-PATH 的路由器接口。

✧ BGP 路由选择——顺序调用下列消除规则选择：

1) 路由被指派一个本地偏好值作为其属性之一，具有更高偏好值的路由将被选择；
2) 在余下的路由中，选择具有最短 AS-PATH 的路由；
3) 在余下的路有种，选择最靠近 NEXT-HOP 路由器的路由——由 AS 内部算法决定，通常称为热土豆选路；
4) 仍余下许多路由时，使用 BGP 标示符来选择路由。

✧ 选路策略

1) 所有进入桩网络（stub network）的流量必定是以该网络为目的地的，所有离开桩网络的流量必定是源于该网络的。
2) 多宿桩网络：经由两个或两个以上的不同提供商练到其余网络。

4.7 广播和多播选路

✧ 广播选路（broadcast routing）：网络层提供从一个源节点到网络中的所有其他节点的交付分组的服务。

✧ 多播选路（multicast routing）：使单个源节点能够向其他网络节点一个子集发送分组的拷贝。

4.7.1 广播选路算法

✧ N 次单播（N-way-unicast）

1) 源节点产生分组的 N 次拷贝，对不同目的地的每个拷贝进行编址，并用单播选路向 N 个目的地传输这 N 份拷贝。
2) 缺点：
   - ◆ 低效率；
   - ◆ N 次单播的隐含假设是广播的接收方及其地址均为发送方所知，这将增加更多的开销；
   - ◆ 使用单播选路基础设施来实现广播是不明智的。

✧ 无控制洪泛：

1) 最显而易见的方法。
2) 当某节点接收了一个广播分组时，它复制该分组并向它的所有邻居转发。
3) 致命缺点：

- ◆ 若网络节点形成圈，则分组将在期中无休止地循环。
- ◆ 广播风暴（broadcast storm）。

✧ 受控洪泛：

1) 序号控制洪泛（sequence-number-controlled flooding）：
   - ◆ 源节点将其地址（或其他的唯一标示符）以及广播序号放入广播分组，然后向其所有邻居转发；
   - ◆ 一个节点接收到一个广播分组时，首先检查该分组是否在其列表中，在则丢弃该分组，不在则复制该分组并向其邻居转发。
2) 反向路径转发（reverse path forwarding，RPF）：当路由器接收到具有给定源地址的广播分组时，仅当该分组到达的链路正好是位于它自己到源的最短单播路径上，它才向所有出链路转发分组。

✧ 生成树广播：

1) 序号控制洪泛和 RPF 不能完全避免冗余广播分组的传输。
2) 最小生成树：所有生 成树中费用最小的生成树。
3) 生成树广播的做法：
   - ◆ 首先对网络节点构造出一棵生成树；
   - ◆ 源节点发送广播分组时，向生成树的特定链路发送分组；
   - ◆ 接收广播分组的节点向生成树中的所有邻居转发该分组。

   一个节点不需要知道整棵树，只需知道哪些邻居是生成树邻居。

4.7.2 多播

✧ 实现多播选路要面对的问题：

1) 如何标识多播分组的接收方；
2) 如何为发送到接收方的分组编址。

✧ 多播数据报使用间接地址（address indirection）编址——用一个标识来表示一组接收方（D 累多播地址）。

✧ 与一个 D 类地址相关联的接收方组称为一个多播组。

✧ 互联网组管理协议——通知与其相连的路由器在本主机上运行的一个应用程序想一个特定的多播组。

✧ 多播选路算法：

1) 使用一棵组共享树进行多播选路；
2) 使用一棵基于源的树进行多播选路——为每个源构建一棵多播选路树。

✧ 距离向量多播选路协议（Distance Vector Multicast Routing Protocol，DVMRP）：实现了具有反向路径转发与剪枝算法的基于源的树。

✧ 协议无关的多播协议（Protocol-Independent Multicast，PIM）：

1) 稠密模式中，多播组成员的位置分布稠密，即该区域内的许多或大多数路由器都需要参与到多播数据报选路之中。——PIM 稠密模式时一种洪泛与剪枝反向路径转发技术。
2) 稀疏模式中，具有相连组成员的路由器数量相对于路由器总数来说很少。——使用汇合点来建立多播分布树。

Chapter 5：链路层和局域网

✧ 链路层信道的两种类型：

1) 广播信道：许多主机被连接到相同的通信信道，需要媒体访问协议来协调传输和避免“碰撞”。
2) 点对点通信链路：如两台路由器之间的通信链路或一台住宅的拨号调制解调器与一台 ISP 路由器之间的通信链路。

5.1 链路层：概述和服务

- ✧ 节点（node）：主机和路由器。
- ✧ 链路（link）：沿着通信路径连接相邻节点的通信信道。

5.1.1 链路层提供的服务

- ✧ 链路层协议定义：
  1) 链路两端的节点之间交互的分组格式；
  2) 发送和接收分组时节点采取的动作。——差错检测、重传、流量控制和随即接入。
- ✧ 链路层协议的任务：将网络层的数据报通过路径中的**单段链路**节点到节点的传送。
- ✧ 链路层协议能够提供的服务：
  1) 成帧（framing）
  2) 链路接入（link access）
  3) 可靠交付（reliable delivery）——许多有线的链路层协议不提供可靠交付服务。
  4) 流量控制（flow control）：链路每一端的节点都具有有限容量的帧缓存能力。
  5) 差错检测（error detection）：十分复杂，且用硬件实现。
  6) 差错纠正（error correction）
  7) 半双工和全双工（half-duplex and full-duplex）：全双工——链路两端可以同时传输分组；半双工——节点不能同时进行传输和接收。

5.1.2 链路层在何处实现

- ✧ 链路层的主题部分在网络适配器（network adapter）中实现，网络适配器也称为网络接口卡（network interface card，NIC）。
- ✧ 网络适配器的内核是链路层控制器，实现了许多链路层服务的单个特定目的芯片。许多功能是用硬件实现的。

5.2 差错检测和纠错技术

- ✧ 发送方接受的挑战：为避免比特差错，使用差错检测和纠错比特（error-detection and-correction，EDC）来增强数据 D。由于传输中比特翻转，到达目的地的 EDC’和 D’可能不同。
- ✧ 接收方接受的挑战：只接收到 D’和 EDC’判断 D’和初始 D 是否相同，接收方可能无法知道接收的信息中包含比特差错。

5.2.1 奇偶校验

- ✧ 单个比特校验：不够健壮。
- ✧ 二位奇偶校验（two dimensional parity）：包含比特值改变的列和行的校验值都会出现差错。——检测到出错并纠正。
- ✧ 前向纠错（Forward Error Correction，FEC）：接收方检测和纠正差错的能力。

5.2.2 检验和方法

- ✧ 互联网检验和（Internet checksum）：数据的两个字节作为 16 比特的整数对待并求和，这个和的反码形成了携带在报文首部的互联网检验和。
- ✧ 对检验和方法的评价：分组开销小，差错保护相对较弱。

5.2.3 循环冗余检测

- ✧ 循环冗余检测（Cyclic Redundancy Check，CRC），也称为多项式编码（polynomial code），将发送的比特传看做是系数为 0 和 1 的一个多项式。
- ✧ CRC 编码操作：发送方和接收方首先协商一个 r+1 比特模式，称为生成多项式（generator），表示为 G。要求 G 的最高有效位为 1。对于一个给定的数据 D，发送方选择 r 个附加比特 R，并附加到 D 上，用模 2 运算正好能被 G 整除。
- ✧ CRC 差错检测过程：接收方用 G 除接收到的 d+r 比特，余数非 0 则出现差错。

- ✧ 所有 CRC 计算采用模二算术操作，在加法中不进位，减法中不借位。
- ✧ 每个 CRC 标准都可以检测小于 r+1 比特的突发差错，即所有连续的 r 比特或者更少的差错都可以被检测到。
- ✧ 适当的假设下，长度大于 r+1 比特的突发差错能以概率 $1-0.5^r$ 被检测到。
- ✧ 每个 CRC 标准可都能检测到任何奇数个比特差错。

5.3 多路访问协议

- ✧ 点对点链路：由链路一端的单个发送方和链路另一端的单个接收方组成。
- ✧ 广播链路：能够让多个发送和接收节点连接到相同的、单一的、共享的广播信道。（以太网、无线 LAN）
- ✧ 多路访问协议（multiple access protocol）：规范节点在共享的广播信道上的传输行为。
  1) 信道划分协议（channel partitioning protocol）
  2) 随机接入协议（random access protocol）
  3) 轮流协议（taking-turns protocol）
- ✧ 对于速率为每秒 R 比特的广播信道，多路访问协议应该有以下所希望的特性：
  1) 当只有一个节点有数据发送时，该节点具有 Rbps 的吞吐量；
  2) 当有 M 个节点要发送数据时，每个节点吞吐量为 R/Mbps（平均传输速率）；
  3) 协议是分散的，不会因某主节点故障而使整个系统崩溃；
  4) 协议是简单的，实现代价不会过高。

5.3.1 信道划分协议

- ✧ 在共享信道节点之间用于划分广播信道带宽的技术：时分多路复用（TDM）、频分多用复用（FDM）、码分多址（Code Division Multiple Access，CDMA）。
- ✧ TDM：
  1) 将时间划分为时间帧（time frame），并进一步划分每个时间帧为 N 个时隙（slot）；
  2) 将每个时隙分配给 N 个节点中的一个；
  3) 某个节点在有分组要发送时，在循环的 TDM 帧中在指定的时隙中传输其分组比特。
- ✧ 对 TDM 的评价：
  1) 优点：
     - ◆ 消除了碰撞；
     - ◆ 公平——每个节点在每个帧时间内获得专用传输速率 R/Nbps。
  2) 缺陷：
     - ◆ 节点被限制于 R/Nbps 的平均速率，即使它是唯一有分组要发送的节点；
     - ◆ 节点必须总是等待在传输序列中的轮次，即使它是唯一有帧要发送的节点。
- ✧ FDM：
  1) 将 Rbps 信道划分为不同频段（每个频段具有 R/N 带宽），并把每个频率分配给 N 个节点中的一个。
  2) 对其评价与对 TDM 评价一致。
- ✧ 码分多址（Code Division Multiple Access，CDMA）：
  1) 为每个节点分配一种不同的编码，每个节点用它唯一的编码来对它发送的数据进行编码。
  2) CDMA 的特性——不同节点能够同时传输，且各自相应的接收方仍然能正确接收发送方编码后的数据比特，而不在乎其他节点的干扰传输。

5.3.2 随机接入协议

- ✧ 随机接入协议中，一个传输节点总是以信道的全部速率（即 R bps）进行发送。当有碰撞时，设计碰撞的每个节点反复重发它的帧，直到无碰撞的通过为止。（当经历碰撞时，不立刻重发，而是等待一个随机时延）

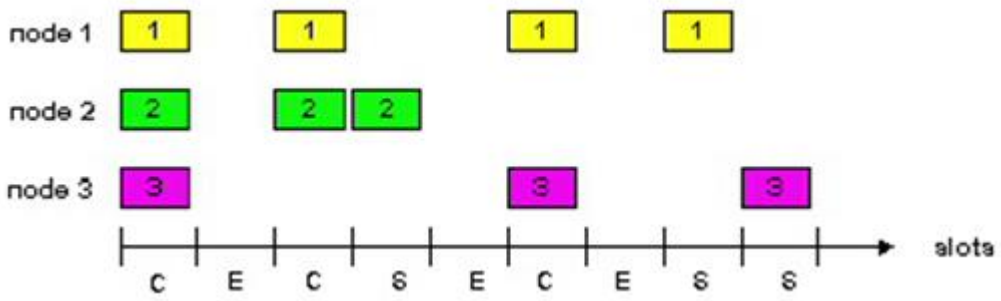
✧ 时隙 ALOHA 协议：

1) 假设：

- ◆ 所有帧均为 L 比特；
- ◆ 时间被划分为长度为 L/R s 的时隙（也即一个时隙等于传输一帧的时间）；
- ◆ 节点只在时隙起点传输帧；
- ◆ 节点是同步的，每个节点都知道时隙何时开始；
- ◆ 如果一个时隙中有碰撞，则所有节点在该时隙结束前得知。

2) 时隙 ALOHA 协议的操作：

- ◆ 节点由新帧发送时，等到下一个时隙开始并在该时隙传输整个帧；
- ◆ 没有碰撞，则无需考虑重发；
- ◆ 有碰撞，则在时隙结束前检测，然后节点以概率 p 在后续的每个时隙中重新传输该帧，直至其被无碰撞的传输出去。



3) 评价：

- ◆ 优点：
  - ➢ 当某节点唯一活跃时，可以全速连续传输；
  - ➢ 高度分散：每个节点决定何时重传；
  - ➢ 十分简单。
- ◆ 缺点：
  - ➢ 多个活跃节点存在时，一部分时隙将因碰撞而浪费；
  - ➢ 时隙的另一部分将是空闲的：所有活跃节点由于概率传输策略会节制传输。

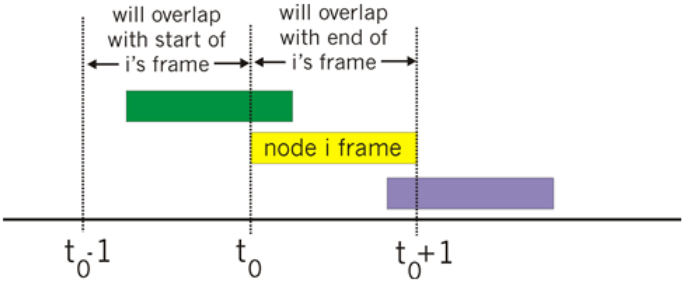
4) 成功时隙：刚好有一个节点传输的时隙。

5) 时隙多路访问协议的效率：有大量的活跃节点且每个节点总有大量帧要发送时，长期运行中成功时隙的份额。

6) 对时隙 ALOHA 最大效率的推导可得知：37%为协议最大效率，也即最大传输速率仅为 0.37R bps；此外，37%的时隙将是空闲的，36%的时隙将有碰撞产生。

✧ ALOHA：

1) 纯 ALOHA 中，第一帧首次到达，节点立刻将该帧完整的传播进广播信道。若有碰撞产生，则立即（传输完当前碰撞帧后）以概率 p 重传该帧，否则该节点等待一个帧的传输时间后以概率 p 重传。



2) ALOHA 的最大效率：2e/1。

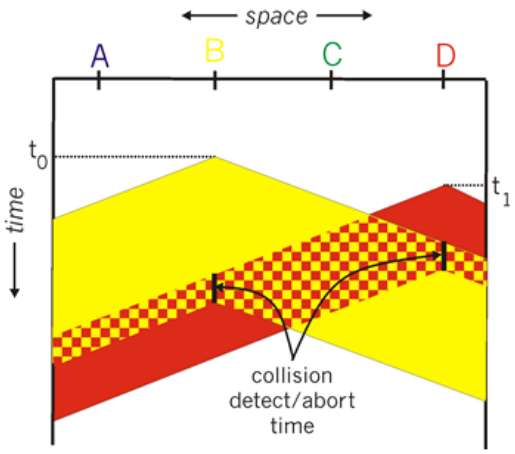
✧ 载波侦听多路访问（CSMA）：

1) 载波侦听（carrier sensing）：即一个节点在传输前先听信道。

2) 碰撞检测（collision detection）：即一个传输节点在传输时一直在侦听信道。

3) 碰撞还是会存在：信道传播时延。

4) 节点执行碰撞检测时，一旦检测到碰撞立即停止传输。



5.3.3 轮流协议

✧ 轮询协议（polling protocol）：节点之一被指定为主节点，主节点以循环的方式轮询每个节点。

✧ 轮询协议消除了碰撞和空时隙，使得效率更高。

✧ 轮询的缺点：

1) 引入了轮询时延，即通知一个节点它可以传输所需的时间；

2) 主节点有故障时，整个信道都将变得不可操作。

✧ 令牌传递协议（token-passing protocol）：一个称为令牌（token）的特殊目的帧在节点之间以某种固定的次序进行交换。当一个节点接到令牌时，仅当它有一些帧要发送时，它才持有这个令牌。若一个节点接收到令牌且需要发送帧，则发送最大数目的帧，然后转交令牌给下一节点。

✧ 对令牌传递协议的评价：

1) 分散，且效率十分高；

2) 若节点忘记了释放令牌，必须使用某些恢复步骤使令牌返回到循环中来。

5.3.4 局域网

✧ 流行的两类 LAN 技术：

1) 以太网 LAN，基于随机接入；

2) 由令牌传递技术组成的 LAN 技术，包括令牌环、光纤式分布数据接口（Fiber Distributed Data Interface，FDDI）。

✧ 令牌环 LAN：LAN 的 N 个节点（主机和路由器）通过直接链路连接成一个环。令牌环的拓扑定义了令牌传递的次序。当某节点获得了令牌并发送一个帧时，该帧绕着整个换传播，从而创建一个虚拟广播信道。

5.4 链路层编址

5.4.1 MAC 地址

✧ LAN 地址的不同称呼：LAN 地址、物理地址、MAC 地址。

✧ 链路层地址也称为 MAC 地址（因其流行）。

✧ MAC 地址被设计为永久的，但用软件改变一块适配器的 MAC 地址是可能的。

✧ 没有两台适配器具有相同的地址。

✧ 适配器的 MAC 地址具有扁平结构（与层次结构相反），无论适配器到哪里都不会发生变化。

✧ 当某适配器要向某些适配器发送一个帧时，发送适配器将目的适配器的 MAC 地址插入到该帧中，并将该帧发送到 LAN 上。

1) 若该 LAN 是广播 LAN，这个帧会被该 LAN 上的所有其他适配器接收和处理；

2) 接收到该帧的每个适配器将检查该帧的目的地址与自己的 MAC 地址是否匹配：

◆ 匹配：取出封装的数据包，并沿协议栈向上传递给它的父节点。

◆ 不匹配：该适配器丢弃该帧，不向上传递。

◆ 当目的节点收到该帧时，仅有它将会中断父节点。

✧ MAC 广播地址——让 LAN 上所有的其他适配器来接收并处理打算发送的帧：48 个连续的 1。

5.4.2 地址解析协议

✧ 地址解析协议（Address Resolution Protocol，ARP）：网络层地址和链路层地址之间的转换。

✧ ARP 只为在同一个子网上的节点解析 IP 地址。

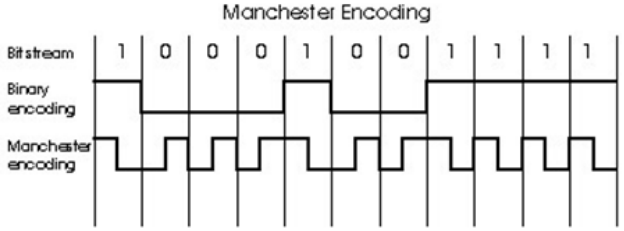
- ARP 的工作流程：
  1) 每个节点的 ARP 模块都在它的 RAM 中有一个 ARP 表，包含 IP 地址到 MAC 地址的映射和生存周期。
  2) 当所查询的节点在发送方的 ARP 表中有相应表项时，直接查询即可。
  3) 没有相应表项时，广播 ARP 查询分组，匹配节点给查询节点发送回一个带有所希望映射的响应 ARP 分组。
- 关于 ARP 协议需要注意：
  1) 查询 ARP 报文是在广播帧中发送的，而响应 ARP 报文在一个标准帧中发送；
  2) ARP 是即插即用的，无需配置，节点断开连接时，表项最终会被删掉。
- 发送数据包到子往外的节点：P302-303

5.5 以太网

- 集线器（hub）：物理层设备，作用于比特。当一个比特到达一个接口时，集线器重新生成这个比特，将其能量强度放大，并向其他所有接口传输出去。
- 基于集线器星型拓扑的以太网是一个广播 LAN。
- 若某集线器同时从两个不同的接口接收到帧，出现碰撞，则生成该帧的节点必须重新传输该帧。

5.5.1 以太网帧结构

- 以太网帧的字段：
  1) 数据字段（data field）：承载 IP 数据报（也能承载其他网络层分组，此处以 IP 数据报为例）。以太网最大传输单元（MTU）是 1500 字节，大于则需给数据报分段；最小长度为 46 字节，小于则需填充数据报字段。
  2) 目的地址（destination address）：包含目的适配器的 MAC 地址。
  3) 源地址（source address）：包含传输该帧到 LAN 上的适配器的 MAC 地址。
  4) 类型字段（type field）：允许以太网复用多种网络层协议。
  5) 循环冗余检测（Cyclic Redundancy Check，CRC）：接收适配器检测帧中是否引入了差错。
  6) 前同步码（preamble）：以太网帧以一个 8 字节的前同步码开始。前同步码的前七个字节都是 10101010，最后一个字节是 10101011。前七个字节用于“唤醒”接收适配器，并且将它们的时钟与发送方同步。第八个字节的最后两个比特“11”警告接收适配器：“重要的内容”即将到来。
- 曼彻斯特编码：每个比特包含一次跳变——1 从高电平跳变为低电平，0 从低电平跳变为高电平。
- 使用曼彻斯特编码的原因：发送适配器和接收适配器没有精确同步，通过每一次跳变，接受主机能够将时钟同步，时钟同步后，接收方就能对每个比特定界，确定它是 1 还是 0。
- 曼彻斯特编码是一种物理层操作。
- 以太网技术向网络层提供无连接服务。
- 以太网技术向网络层提供不可靠服务。例如，适配器接收到帧后，对该帧执行 CRC 校验，但无论通过校验与否，此适配器都不会发送确认帧或否定帧。



5.5.2 CSMA/CD：以太网的多路访问协议

- CSMA/CD 使用的机制：
  1) 适配器可以在任何时刻开始传输，没有时隙的概念；
  2) 使用载波侦听；
  3) 使用碰撞检测；
  4) 尝试重传之前，适配器等待一个随机时间（通常小于传输一帧的时间）。
- CSMA/CD 的工作方式：
  1) 适配器从网络层得到一个数据报，准备一个以太网帧，并将该帧放到适配器缓存区；

2) 若适配器侦听到信道空闲，则传输该帧；若侦听到信道忙，则等待到不再听到信号能量后传输；
3) 传输过程中，若未侦听到其他信号能量，则持续传输直至结束；
4) 传输过程中，若侦听到其他信号能量，则停止传输，代之以传输 48 比特的阻塞信号；
5) 在中止传输后，适配器进入一个指数后退（exponential backoff）阶段。

注：

1) 阻塞信号的目的：确保所有其他的传输适配器都意识到此次碰撞。
2) 指数后退算法：在帧经受了一连串的 n 此碰撞后，适配器随机的从{0,1,2,⋯2^m-1}等概率的为 K 选择一个值，其中 m=min（0,10），然后适配器等待 K*512 比特时间后重新开始传输该帧。由于选择 K 的集合长度随着碰撞次数的增加呈指数增长（直到 n=10），该算法称为指数后退。

✧ 以太网效率：当有大量的活跃节点，并且每个节点由大量的帧要发送时，帧在信道中无碰撞地传输的那部分时间占长期运行时的份额。

5.5.3 以太网技术

✧ 吉比特以太网标准：IEEE802.3z
1) 使用标准以太网帧格式，后向兼容 10BASE-T 与 100BASE-T 技术，很容易在已有以太网设备基础上集成；
2) 允许点对点链路及共享广播信道；
3) 使用 CSMA/CD 来共享广播信道；
4) 对于点对点信道，允许在两个方向上以 1000Mbps 全双工操作。

5.6 链路层交换机

✧ 现代以太网 LAN 使用星型拓扑——每个节点与中心交换机互连。
✧ 交换机的任务：接收入链路层帧并转发到出链路。
✧ 交换机自身对于节点来讲是**透明的**——某节点向另一个节点寻址一个帧（而不是通过交换机），顺利将其发送进 LAN，而不知道交换机将会接收该帧并将其转发至另一节点。

5.6.1 交换机转发和过滤

✧ 过滤（filtering）：交换机决定一个帧是应该转发到某个接口还是应当将其丢弃的功能。
✧ 转发（forwarding）：决定一个帧应该被导向哪个接口，并把该帧接口移动到这些接口的交换机功能。
✧ 过滤与转发都借助交换机表完成：
1) 包含其 LAN 上的某些节点但不是全部节点；
2) 交换机表中的一个表项包含：
   - ◆ 节点的 MAC 地址；
   - ◆ 到达该节点的交换机接口；
   - ◆ 用于节点的表项放置的表中的时间。

✧ 交换机转发分组基于 MAC 地址而非 IP 地址。
✧ 交换机的工作过程：

假设具有目的地址 DD-DD-DD-DD-DD-DD 的帧从交换机接口 x 到达，交换机用 MAC 地址 DD-DD-DD-DD-DD-DD 索引它的表：

1) 表中没有针对该 MAC 地址的表项：向除了接口 x 外的所有接口前面的缓存转发该帧的拷贝（也即广播该帧）。
2) 表中有表项将该 MAC 地址与接口 x 联系起来：说明该帧从包括适配器 DD-DD-DD-DD-DD-DD 的 LAN 网段来，此时只需丢弃该帧，执行过滤功能。
3) 表中有表项将该 MAC 地址与接口 y 联系起来，且 y≠x：转发该帧到与接口 y 相连的网段。

5.6.2 自学习

✧ 交换机的自学习（self-learning）：交换机表示自动的、动态的、自治的建立起来的。

✧ 自学习的实现方式：

1) 交换机表初始为空；
2) 对于某接口接收到的每个入帧，交换机在其表中存储：
   - ◆ 该帧源地址地段中的 MAC 地址；
   - ◆ 该帧到达的接口；
   - ◆ 当前时间。

   注：这样如果在 LAN 节点上每个节点最终都发送了一个帧，则每个节点将在这张表中记录下来。
3) 在一段时间（老化期）后，若交换机没有接受到以改地址作为源地址的帧，就在表中删除这个地址。

   注：以这种方式，若一台 PC 被另一台 PC 替换，则原 PC 的 MAC 地址最终将被交换机清除。

✧ 交换机是即插即用设备（plug-and-play device）：无需网络管理员或用户的干预。

✧ 交换机是双工的：任何节点与交换机连接的链路，节点好交换机能够同时传输而无碰撞。

5.6.3 链路层交换机的性质

✧ 交换机的优点：

1) 消除碰撞：交换机缓存帧，且决不会在网段上同时传输多余一个帧，最大聚合带宽是该交换机所有接口速率之和，比广播链路的 LAN 具有更高的性能；
2) 异质的链路：交换机将链路彼此隔离，LAN 中的不同链路能够以不同速率运行，并且能够在不同媒体上运行；
3) 管理：
   - ◆ 若适配器工作异常并持续发送以太帧（快而含糊的，jabbering），交换机可检测到该问题并在内部断开异常适配器；
   - ◆ 交换机可手收集带宽使用的统计数据、碰撞率和流量类型，并把这些信息提供给网络管理者使用。

5.6.4 交换机和路由器的比较

✧ 交换机是第二层的分组交换机，路由器是第三层的分组交换机。

✧ 交换机：

1) 优点：
   - ◆ 即插即用；
   - ◆ 具有相对高的分组过滤和转发速率。
2) 缺点：
   - ◆ 为防止广播帧的循环，交换网络的活跃拓扑限制为一棵生成树；
   - ◆ 交换网络要求在该节点有大的 ARP 表，将生成可观的 ARP 流量和处理量；
   - ◆ 交换机对于广播风暴不提供任何保护措施，若某主机出现故障并传输大量以太网广播帧流，交换机将转发所有帧，致使以太网崩溃。

✧ 路由器：

1) 优点：
   - ◆ 网络寻址是分层次的，即使网络中存在冗余路径，分组通常也不会通过路由器循环；且分组不会限制在一棵生成树上，可以使用源到目的地的最佳路径，可以用各种拓扑结构来构建因特网；
   - ◆ 对第二层的广播风暴提供防火墙保护。
2) 缺点：
   - ◆ 不是即插即用；
   - ◆ 对每个分组的处理时间相较于交换机更长；
   - ◆ 浪费了许多时间争论路由器的正确发音。

✧ 怎样选择是使用路由器还是交换机？

1) 几百台主机组成的小网络——交换机：不要求 IP 地址的任何配置就能使流量局部化并增加总计吞吐量；
2) 几千台主机组成的大网络——路由器：提供更健壮的流量隔离方式和广播风暴控制。

✧ 流行的互联网设备的典型特色比较：

	集线器	路由器	交换机
流量隔离	无	有	有
即插即用	有	无	有
优化选路	无	有	无
直通交换	有	无	有

5.7 PPP：点对点协议

✧ 点对点协议：运行于点对点链路之上的链路层协议，也即一条直线连接两个节点的链路，链路的每一端有一个节点。

✧ 对 PPP 的初始设计要求：

1) 分组成帧：PPP 协议链路层的发送方必须能封装网络层分组在其链路层帧中；
2) 透明性：不能对网络层分组中的数据做任何限制；
3) 多种网络层协议：必须能够支持同时运行在物理链路上的多种网络层协议，要求 PPP 至少需要一个协议类型字段或其他相似的机制；
4) 多种类型链路：必须能运行在不同类型的链路上，如串行的、并行的等；
5) 差错检测：必须能够检测到接收帧中的比特差错；
6) 连接的活性：必须能够检测到链路层次的故障，并且通知网络层；
7) 网络层地址协商：必须向网络层提供机制，获知或者配置相互的网络层地址；
8) 简单性。

✧ 不要求 PPP 实现的功能：差错纠正、流量控制、有序、多点链路。

✧ PPP 帧包含的字段：

1) 标识字段：每个 PPP 帧都是用值 01111110 的 1 字节标识字段作为开始和结束。
2) 地址字段：唯一可能值为 11111111。
3) 控制字段：唯一可能值为 00000011.

PPP 规范指出地址字段和控制字段“可能在以后定义”，协议允许发送方不发送这两个字段的值，为帧省下了两个字节的开销。

4) 协议：该字段指明所接收的封装数据所属的上层协议。
5) 信息：包含上层协议在 PPP 链路上发送的被封装分组（数据），最大的默认长度是 1500 字节（首次配置时能改变这个值）。
6) 检验和：检测已传输帧中的比特差错。

✧ 字节填充：解决标识字段（01111110）的值出现在信息字段中，导致接收方错误检测为帧结束的问题。

✧ 字节填充的实现：在非标识字段的 01111110 前加入控制转义字节，指示随后的 01111110 不是标识字段。类似的，如果控制转义字节的比特模式自身作为实际数据出现，它前面也必须填充一个控制转义字节。——接收方在数据流中看到单个控制转义字节时，知道该字节是被填充到数据流中的。相继出现的一堆控制转义字节意味着要在发送的初始数据中出现了一个控制转义字节的实例。

5.8 链路虚拟化：网络作为链路层

✧ 互联网媒体是一种复杂的交换基础设施。

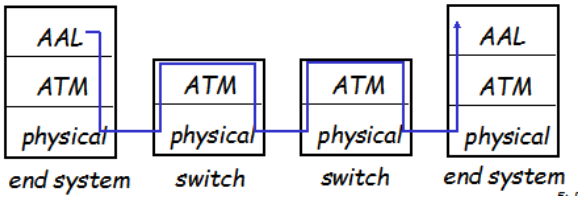
✧ 主机维持的观点：互联媒体是连接两台或多台主机的链路层信道。——主机可以不知道它是通过一个

单一短 LAN 网段还是一个地理上分布的交换 LAN 与其他 LAN 主机进行连接。

5.8.1 异步传输方式

- ✧ 异步传输方式（Asynchronous Transfer Mode，ATM）的目标：设计一种单一的网络技术，传输实时音频、视频以及文本、电子邮件和图像文件。
- ✧ ATM 支持的服务模式：恒定比特率服务、可变比特率服务、可用比特率服务和不指明比特率服务。
- ✧ ATM 适配层（ATM Adaption Layer，AAL），类比于因特网的运输层，仅出现在位于因特网边缘的 ATM 设备上。
- ✧ AAL 被定义用于以下服务：
  1) 恒定比特率服务和电路仿真；
  2) 可变比特率服务（可变比特率视频）；
  3) 数据服务（如 IP 数据报传输）。

  注：AAL 执行的服务中有差错检测和分段/重装。



- ✧ AAL 处理的数据单元称为 AAL 协议数据单元（Protocol Data Unit，PDU）。
- ✧ ATM 层：定义了 ATM 信元的结构和该信元中各个字段的含义。
- ✧ ATM 信元中各字段：
  1) 虚通道表示（VCI）：指示信元属于哪个虚通道。
  2) 负载类型（PT）：指示包含在信元中的有效载荷的类型。包含一个比特用于指示在一个 AAL PDU 中的最后一个信元。
  3) 信元丢失优先级（CLP）比特：发生拥塞并且 ATM 交换机必须丢弃信元时，交换机使用该比特丢弃优先级低的信元。
  4) 首部差错控制（HEC）字节：保护信元首部差错检测比特。
- ✧ 源能够向目的地开始发送信元之前，ATM 网络首先创建一条从源到目的地的虚通道，虚通道标识与每条链路相联系。无论何时创建或拆除一个 VC 必须更新 VC 转换表（使用永久 VC 的情况除外）。
- ✧ ATM 物理层：处理物理媒体上的电压、比特定时和帧。
- ✧ 在 ATM 上传输 IP
  1) 每个入口/出口点是一台路由器，当数量相对少时，可在入口/出口点采用永久的 VC，避免动态的建立和拆除 VC。
  2) 连接到 ATM 的路由器需要两个地址：IP 地址、MAC 地址。
  3) 入口路由器的任务：
     - ◆ 检查数据报的目的地址；
     - ◆ 索引路由器的选路标，决定出口路由器 IP 地址；
     - ◆ 在 ARP 表中，使用出口路由器的 IP 地址，索引其物理地址；
     - ◆ 将该数据报连同其出口路由器的 ATM 地址，向下传递到链路层。
  4) 数据报被入口路由器向下传递到链路层后：
     - ◆ ATM 确定通向 ATM 目的地址的虚通道的 VCI；
     - ◆ 在虚通道的接收端，ATM 将数据报分段为信元，在该虚通道的接收端，将这些信元重装为初始数据报。
       - ➢ IP 分段：将初始数据报分成不超过 48 字节的报文段——存在问题，IP 报文段首部为 20 字节，因此 ATM 信元携带的报文段仅有 28 字节有效信息。
       - ➢ AAL5 分段/重装。
  5) 在 ATM 源和 ATM 目的地之间的每台交换机，ATM 信元由 ATM 物理层和 ATM 层处理，而不涉及 AAL。当到达 ATM 目的地址后，信元被分配给特殊虚通道的 AAL 缓存，然后重新构造 AAL5 PDU，提取出数据报，并向上传递给 IP 层协议栈。

5.8.2 多协议标签交换

- ✧ 多协议标签交换（Multiprotocol Label Switching，MPLS）：对于基于固定长度标签和虚电路的技术，通过选择性的标识数据报并允许路由器基于固定长度的标签转发数据报（而不是目的地址）。
- ✧ MPLS 首部：
  1) 标签：虚电路标示符；
  2) 3 比特保留用于实验；
  3) 单个 S 比特用于指示一系列的“成栈”的 MPLS 首部的结束（不讨论）；
  4) 寿命字段。
- ✧ MLPS 加强的帧仅能在两个均为 MLPS 使能的路由器之间发送，这样的路由器被称为标签交换路由器。
- ✧ 标签交换路由器不需要提取目的 IP 地址和在转发表中执行最长前缀匹配的查找。

Chapter 6：无线网络和移动网络

6.1 概述

- ✧ 无线网络中的元素：
  1) 无线主机：主机本身可能移动，也可能不移动。
  2) 无线链路：主机通过无线链路连接到一个基站或者另一个主机。
  3) 基站（base station）：向与之关联的无线主机发送数据和从主机那里接收数据（例如分组），例如蜂窝塔（cell tower）和 802.11 无线 LAN 中的接入点。——起链路层中继的作用

  注：一台无线主机与基站“相关联”是指
  - ◆ **主机位于该无基站的无线通信覆盖范围内；**
  - ◆ **主机使用该基站中继它和更大网络之间的数据。**

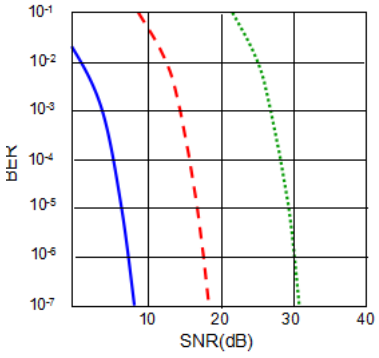
  4) 网络基础设施：无线主机希望与之进行通信的更大网络。
- ✧ 自组织网络（ad hoc network）：
  1) 没有基站；
  2) 节点只能与在自身连接范围内的节点联系；
  3) 节点之间搭建局域网（不能连接互联网）。
- ✧ 切换（handoff）：当一台移动主机移动范围超出一个基站的覆盖范围而到达另一个基站的覆盖范围后，它将改变其接入更大网络的连接点。
- ✧ 对无线网络的分类：

	基于基础设施	无基础设施
单跳	具有与较大的有线网络连接的基站，该基站与无线主机之间的所有通信都经过一个无线跳。	存在与无线网络相连的基站，例如蓝牙网络、自组织模式的 802.11 网络。
多跳	一个表现为以有线网络与较大网络相连，节点与该基站通信，可能通过其他无线节点中继通信。	没有基站，且节点为到达目的地可能在其他几个无线节点之间中计报文。节点可能是移动的，可在多个节点中改变连接关系。

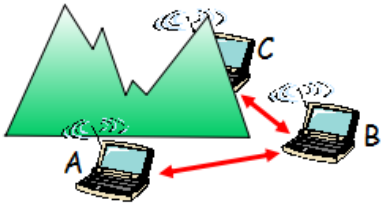
6.2 无线链路和网络特征

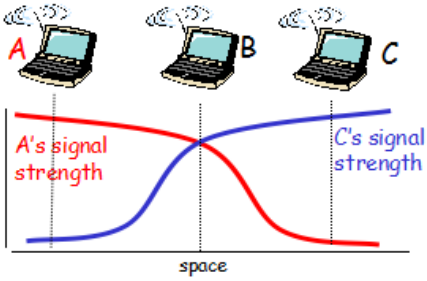
- ✧ 无线链路的特征：
  1) 递减的信号强度（路径损耗，Pass loss）：电磁波在穿过物体时强度会减弱，在自由空间中，信号会扩散；
  2) 来自其他源的干扰：在同一个频段发送信号的电波源将相互干扰（波峰加波峰信号增强，波谷加波谷信号减弱）；
  3) 多路径传播：电磁波的一部分受物体和地面反射，在发送方和接收方之间通过长度不同的路径，称为多径传播（multiple propagation），这使得接收方收到的信号变得模糊。

- ✧ 信噪比（Signal-to-Noise Ratio，SNR）：所收到的信号与噪声强度的相对测量。
- ✧ 较大的 SNR 使得接收方更容易从背景噪声中提取传输的信号。
- ✧ 比特率差错（Bit Error Rate）：接收方收到的有错的传输比特的概率。
- ✧ 为在理想信道上传输信息，需要使用调制技术对信息进行编码：
  1) 对于给定的调制方案，SNR 越高，BER 越低。
  2) 对于给定的 SNR，具有较高传输率的调制技术（无论差错与否）将具有较高的 BER。
  3) 物理层调制技术的动态选择能用于适配对信道条件的调制技术。



- ✧ 隐藏终端问题（hidden terminal problem）

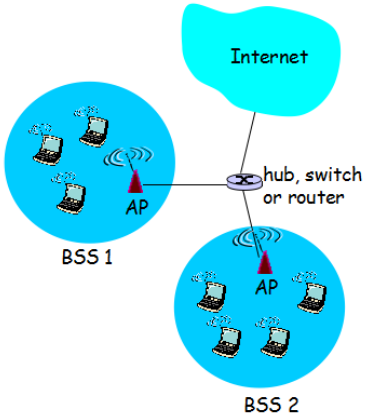




  1) 环境的物理阻挡，例如一座大山或者一座建筑；
  2) 无线媒体传输时信号强度的衰减。
- ✧ CSMA（码分多址）
  1) 要发送的每一比特都乘以一个信号（编码）的比特来进行编码，这个信号的变化速率（通常称为码片速率）比初始比特序列的速率快得多。
  2) *encoded signal* = (original data) X (chipping sequence)
     decoding: inner-product of encoded signal and chipping sequence

6.3 WiFi：802.11 无线 LAN

6.3.1 820.11 体系结构



- ✧ 基本结构：
  1) 基本服务集（Base Service Set，BSS）：包含一个或多个无线站点和中央基站（接入点）；
  2) BSS 连接到一个互联设备上，可能为交换机或路由器；
  3) 互联设备连接到因特网中。
- ✧ 配置 AP 的无线 LAN 被称作基础设施无线 LAN，其中基础设施是指 AP 以及互联 AP 和路由器的有限以太网。
- ✧ 802.11 定义了 11 个部分重叠的信道。
- ✧ 被动扫描：扫描信道和监听信标帧的过程。
- ✧ 主动扫描：向位于无线主机范围内的所有 AP 广播探测帧。

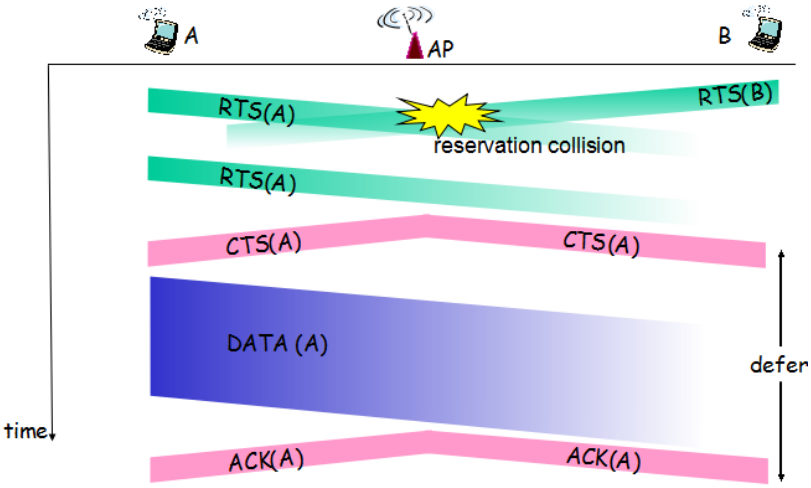
6.3.2 802.11 MAC 协议

- ✧ 一个站点有一个帧要发送时：
  1) 若站点初始时监听到信道空闲，则在分布式帧间间隔（Distributed Inter-Frame Space，DIFS）后发送该帧；
  2) 否则，选取一个随机回退值并且在侦听到信道空闲时递减该值，侦听到信道忙时，保持该值不变；

3) 计数值减为 0 时，站点发送整个数据帧并等待确认；
4) 若收到确认，则说明该帧被正确接收。

✧ 预约机制（在隐藏终端的情况下避免碰撞）：

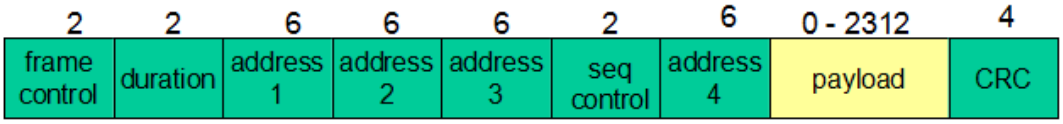
1) 发送方要发送 DATA 帧之前，首先向 AP 发送一个短请求发送帧（Request to Send，RTS），指出传输 DATA 帧和确认帧需要的总时间；
2) AP 接到 RTS 帧后，广播一个 CTS（Clear To Send）帧作为响应：
   - ◆ 给发送方明确的发送允许；
   - ◆ 指示其他站点在预约期内发送。



✧ 使用 RTS 和 CTS 的评价：

1) 缓解隐藏终端问题，长 DATA 帧只有在信道预约后才能传输；
2) RTS 和 CTS 帧本身很短，就算碰撞，也仅持续很短时间，一旦它们被正确传输，后续的 DATA 帧和 ACK 帧将被无碰撞的发送；
3) 引入了时延，且消耗了信道资源。

6.3.3 IEEE 802.11 帧



✧ 有效载荷与 CRC 字段。

✧ 地址字段：

1) 地址 1：接收帧的无线站点的 MAC 地址；
2) 地址 2：传输帧的站点的 MAC 地址；
3) 地址 3：BSS 是一个无线子网的一部分，经一些路由器接口与其他子网相连，地址 3 包括这个路由器接口的 MAC 地址；
4) 地址 4：AP 在自组织模式中互相转发时使用。

✧ 序号、持续期和帧控制字段：

1) 序号：使接收方重新区分新传输的帧和以前帧的重传；
2) 持续期：传输数据帧的时间和传输确认的时间。