

数据通信



数据通信综述

殷亚凤

yafeng@nju.edu.cn

<http://cs.nju.edu.cn/yafeng/>
Room 901, Building of CS





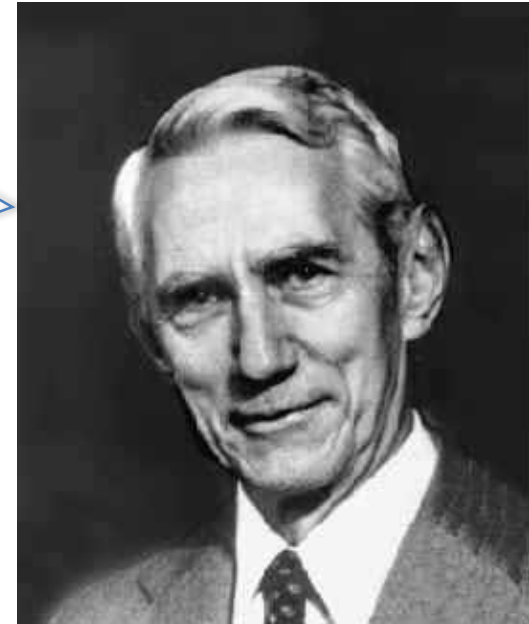
一、数据通信简介

通信的定义



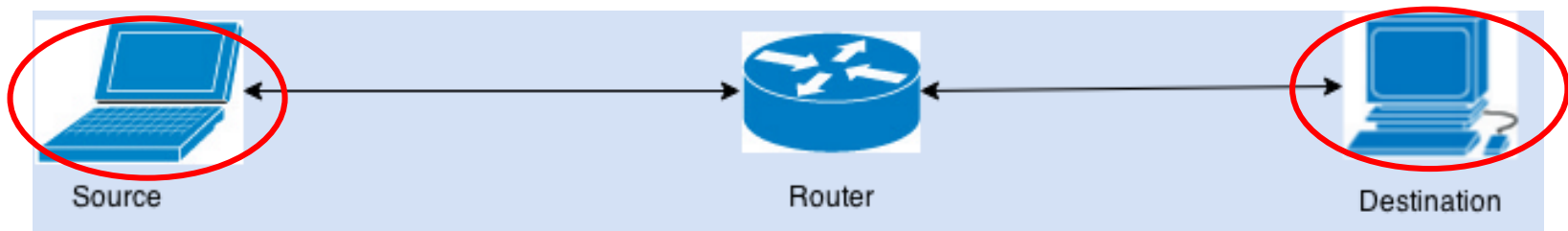
The fundamental problem of communication is that of **reproducing** at one point either exactly or approximately a **message** selected at another point.

- Claude Shannon

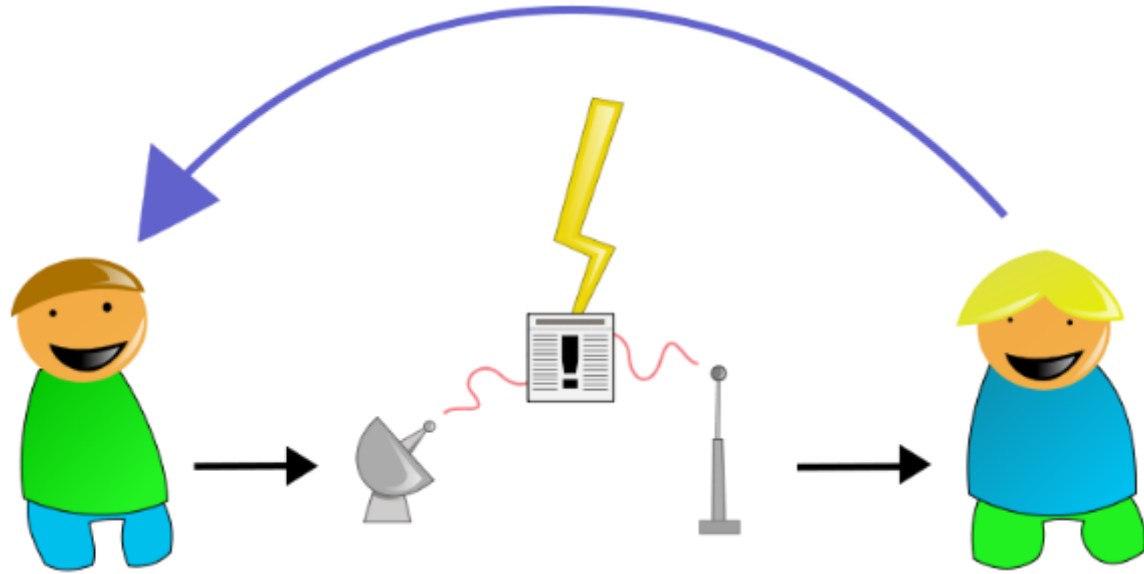


克劳德·香农

通信的基本问题：在**一点**精确地或近似地**恢复另一点**所选择的**消息**。



通信模型

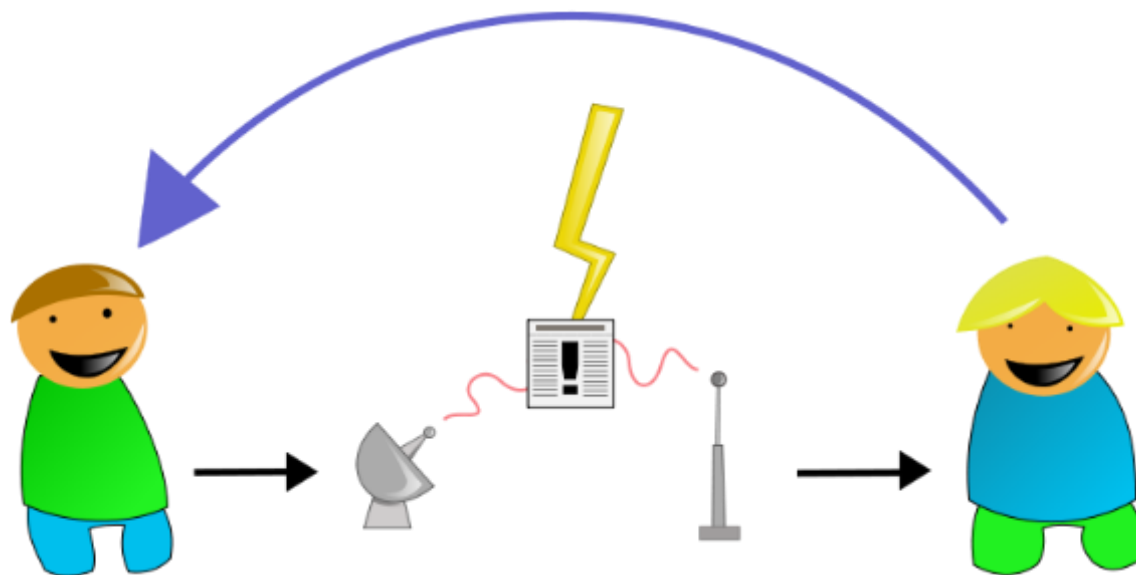


源点

传输
系统

终点

通信模型

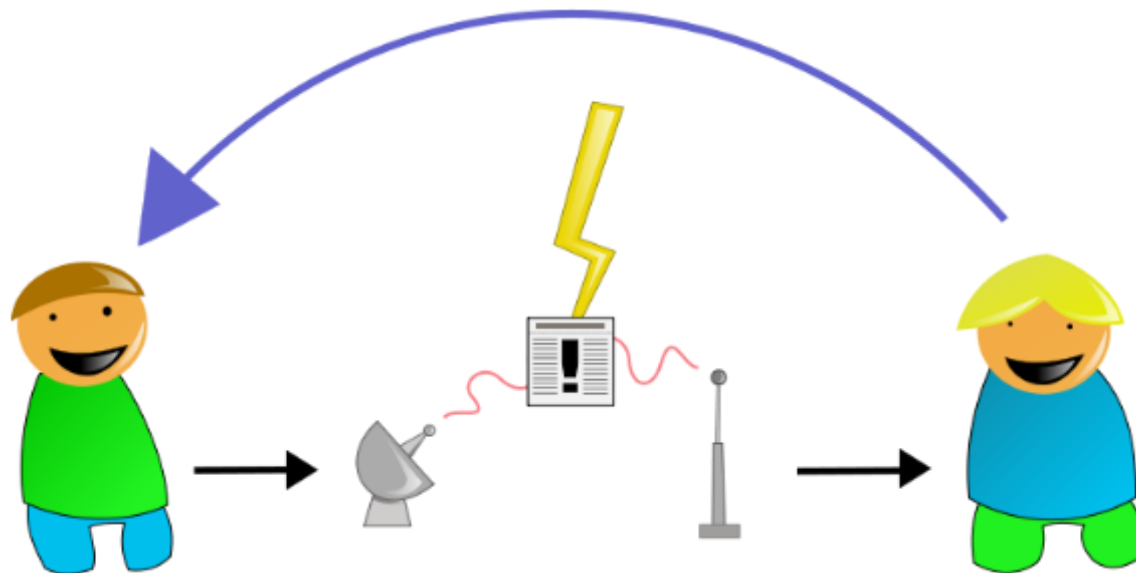


源点

传输
系统

终点

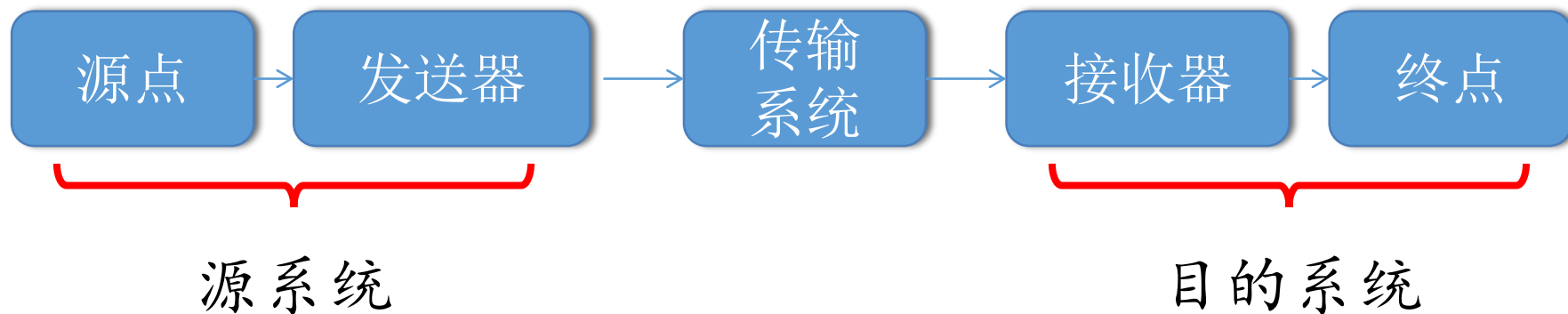
通信模型



源系统

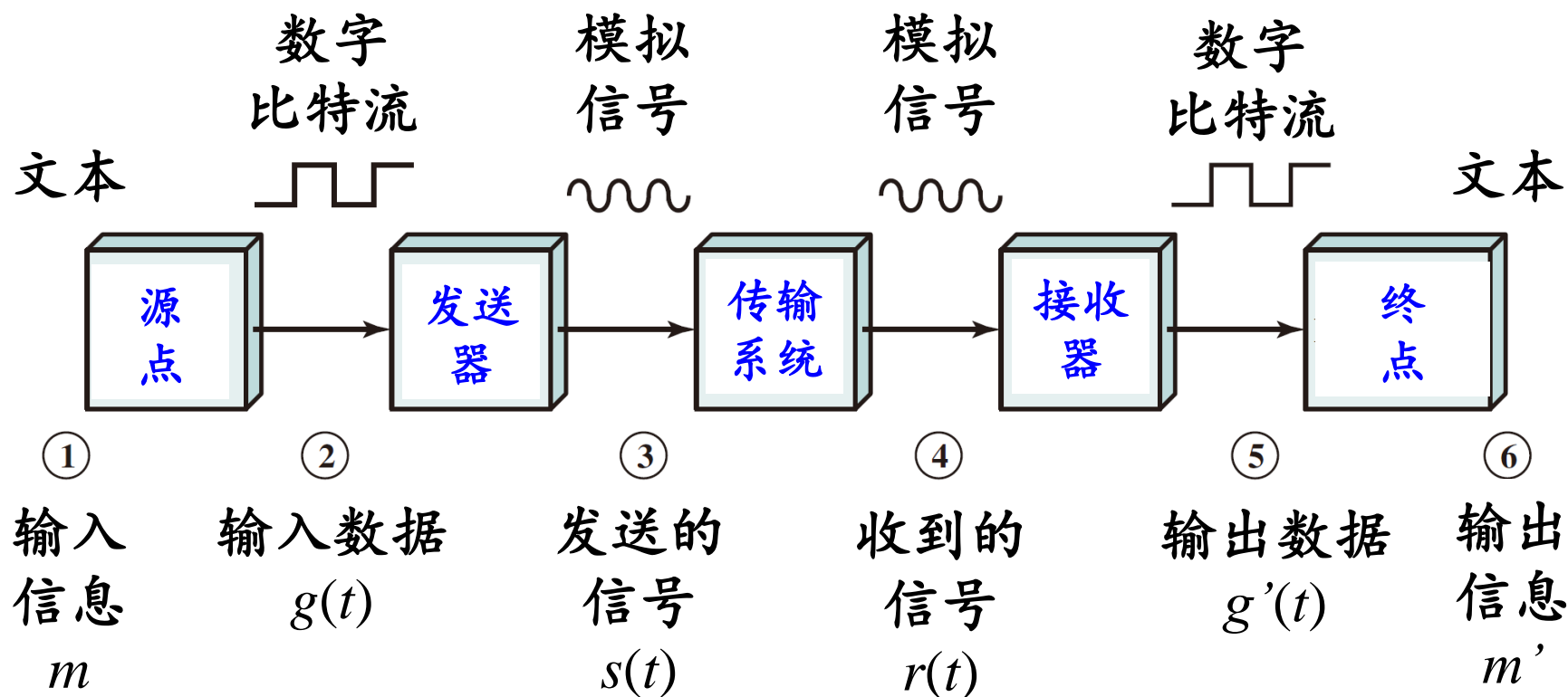
目的系统

通信模型



具体实例：工作站和服务端之间通过公用电话网进行通信

数字通信模型



简化的数据通信模型

通信模型与实际系统的差别



■ 通信模型示意图

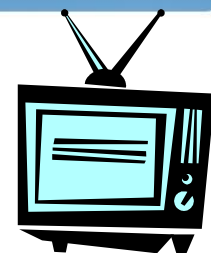
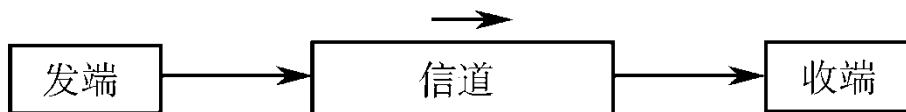
- 表面上描述的是一个一对一的**单向**通信系统
- 图中的**同步**也只是一种时间上协调的会意性描述
- 适于各种通信系统的一个**抽象**模型
- 概括地反映了各种通信系统实际应用中的**共性**。

■ 实际应用中的通信系统

- 往往是**双向**的，而且可能是**多个信源与多个信宿间**的通信。
- 传输系统一般不是简单的点对点的关系，而是涉及比较复杂的**网络结构**，包含交换技术、路由选择等内容

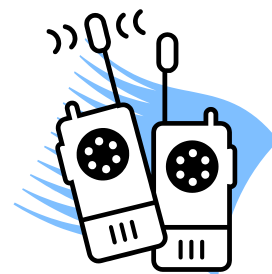
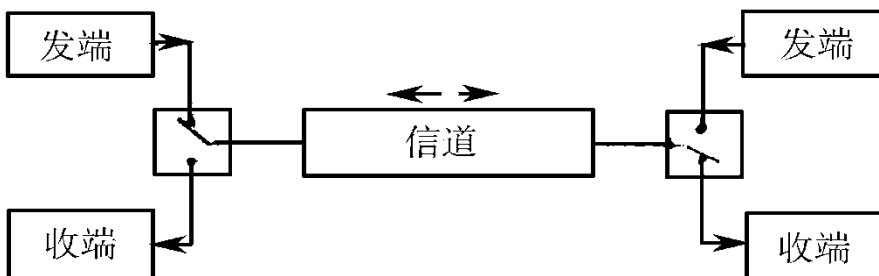
通信传输

- **单工**: 数据信息在通信线上始终向一个方向传输



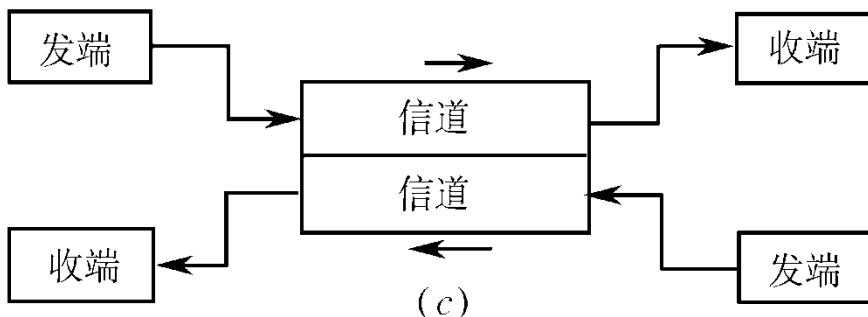
电视

- **半双工**: 可以双向传输, 但必须交替进行, 同一时刻一个信道只允许单向传送



对讲机

- **双工**: 可同时进行双向的数据传输

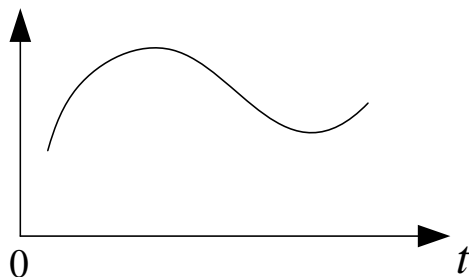


电话

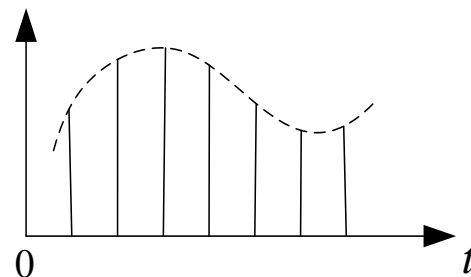
数字通信与模拟通信



- **模拟信号**：代表消息的信号参量取值连续，例如麦克风输出电压

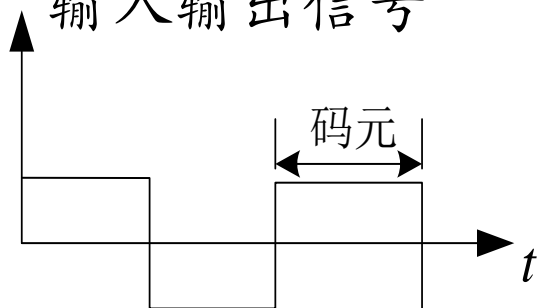


(a) 语音信号

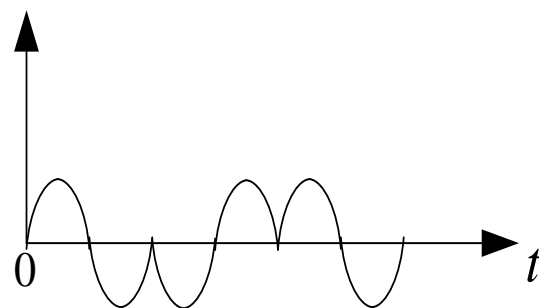


(b) 抽样信号

- **数字信号**：代表消息的信号参量取值为有限个，例如计算机输入输出信号



(a) 二进制信号



(b) 2PSK信号

数字通信与模拟通信



◆ 通常，按照信道中传输的是模拟信号还是数字信号，相应地把通信系统分为模拟通信系统和数字通信系统。

➤ **数字通信系统**: 利用数字信号来传递信息的通信系统

- **数字**设备产生的数据，数字基带传输（数字-数字编码）
- **模拟**设备产生的数据，数字基带传输（模拟-数字编码）

➤ **模拟通信系统**: 利用模拟信号来传递信息的通信系统

- 模拟信号以固有频率基带传输
- 模拟信号频谱搬移后频带传输（模拟-模拟编码）

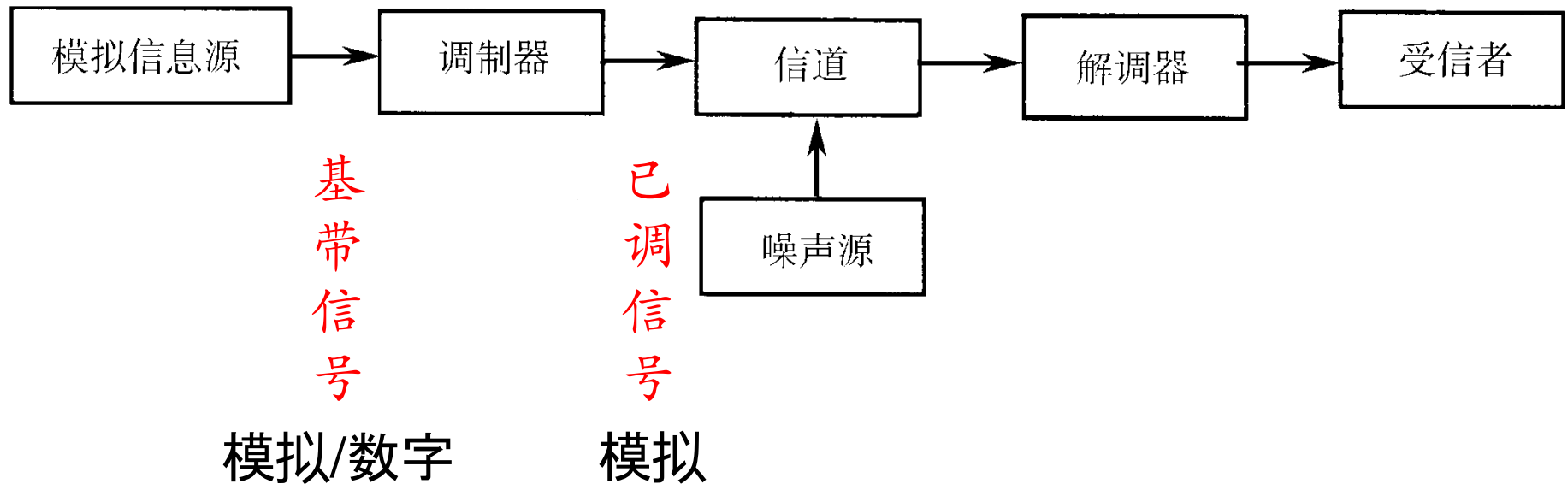
➤ **混成系统**

- 同一系统中包含模拟通信和数字通信

模拟与数字通信



- 模拟通信系统模型

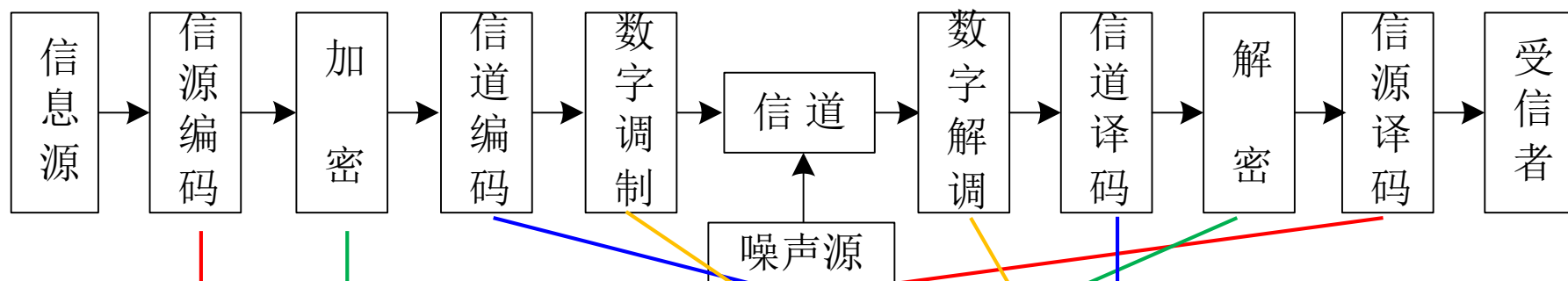


模拟消息 → 原始电信号 (基带信号) → 已调信号 → 基带信号 → 模拟消息

模拟与数字通信



• 数字通信系统模型



- 信源编码与译码目的：
 - 提高信息传输的有效性
 - 完成模/数转换

- 加密与解密目的：
保证所传信息的安全

- 信道编码与译码目的：
增强抗干扰能力

- 数字调制与解调目的：
形成适合在信道中传输的带通信号

数字与模拟通信



- 数字通信的特点

- 优点

- 抗干扰能力强，且噪声不积累
 - 传输差错可控
 - 便于处理、变换、存储
 - 便于将来自不同信源的信号综合到一起传输
 - 易于集成，使通信设备微型化，重量轻
 - 易于加密处理，且保密性好

- 缺点：

- 需要较大的传输带宽
 - 对同步要求高

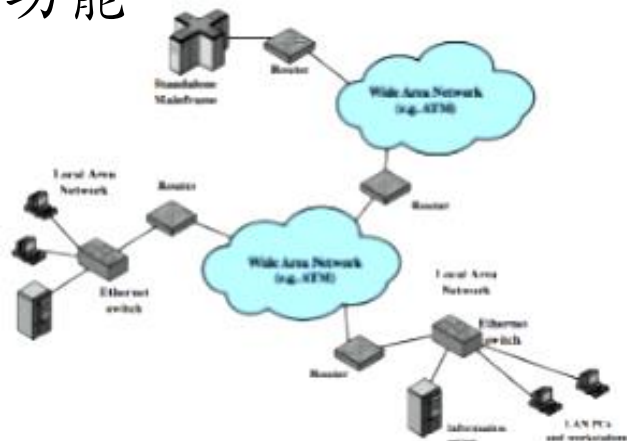


二、网络和因特网

网络术语



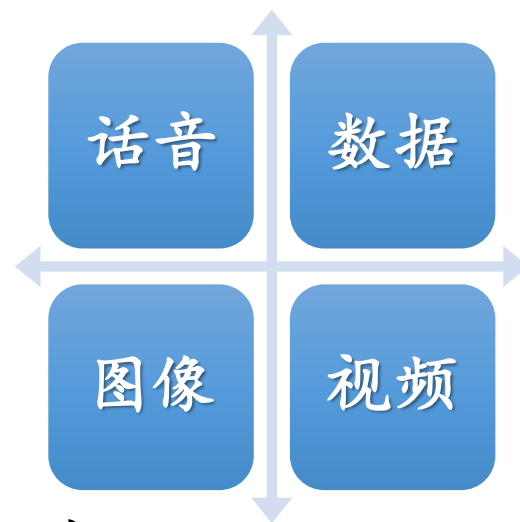
- **节点**：一个网络端口
 - 访问节点：信息交换的源点和目标点，如网卡接口
 - 转节点：网络中间设备的网络端口
- **结点**：一台网络设备，连接多个节点
- **线路**：节点间通信的物理连接
- **链路**：建立在两个节点上的信道，分为物理链路和逻辑链路
- **交换**：交换节点建立、保持和改变数据传输通路的过程
- **路由选择**：交换技术中选择传输路径的功能



网络



- **广域网**：覆盖了很大的地理范围，需要穿越公众设施，并且至少有一部分依靠的是由公共电信运营公司建立的电路进行传输的网络。



- **局域网**：
 - 范围较小，通常是一栋楼或一片楼群。
 - 局域网和局域网相连接的设备都属于同一个组织。
 - 局域网内的数据率要比广域网高很多。

广域网



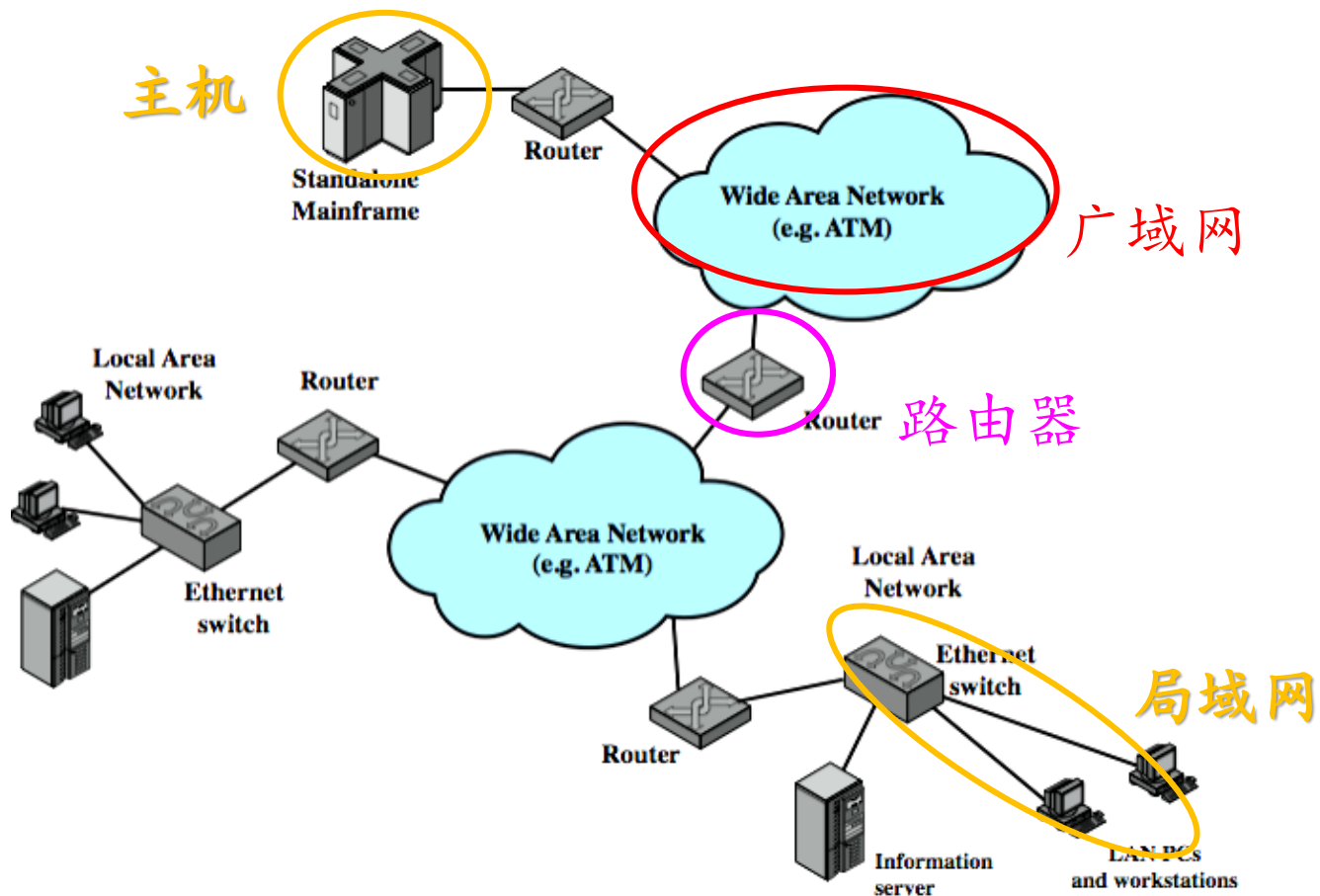
- 在广域网中，一个设备出发的传输过程要通过路由选择，途径这些网络结点，最后到达某个终点设备。
 - **电路交换**：在两个结点之间，建立一条专用的途径多个网络结点的通信路径。（最常见的例子就是电话网）
 - **分组交换**：
 - 数据以一连串小块的形式发送出去，这些小数据块被称为分组。
 - 每个分组沿着从源点到终点的路径，途中经过一个个的结点，穿越整个网络。
 - 对每个结点来说，它接收完整的分组，先将其暂存，然后再传送给下一个结点。
- **帧中继**
- **异步传输方式 (ATM)**

局域网



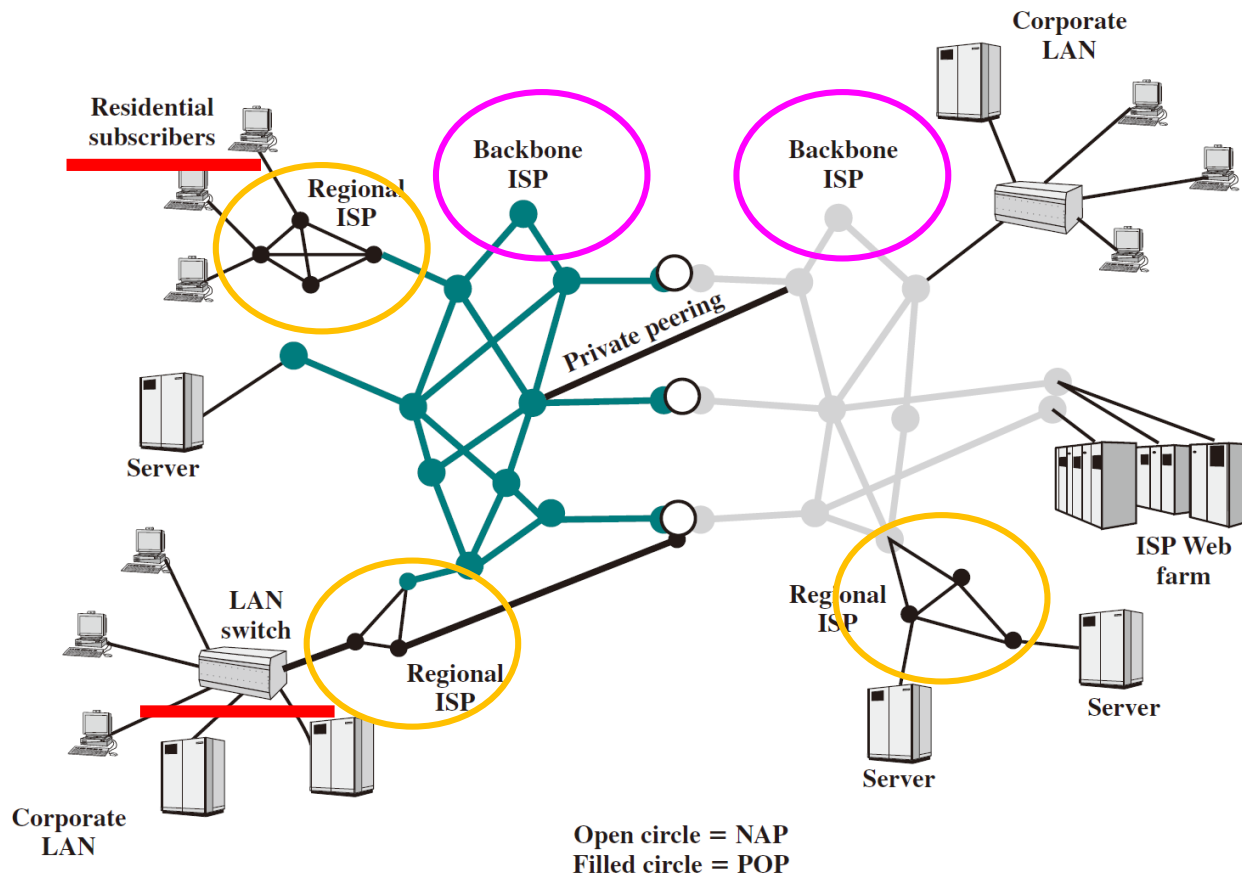
- 局域网也是一种连接各式各样的设备并向这些设备提供交换信息手段的通信网络。
- **交换局域网**：最常见的是交换以太网局域网，它可能只含有一个交换机，其上挂接一些设备，也可能含有多个互连的交换机。
- **无线局域网**：最常见的是WiFi局域网。无线局域网在商务环境下的应用非常广泛。（优势：移动性和安装配置的简易性）

因特网



目标: 终端系统的相互连接。 (终端系统统称为主机)

因特网体系结构



部分因特网简化视图

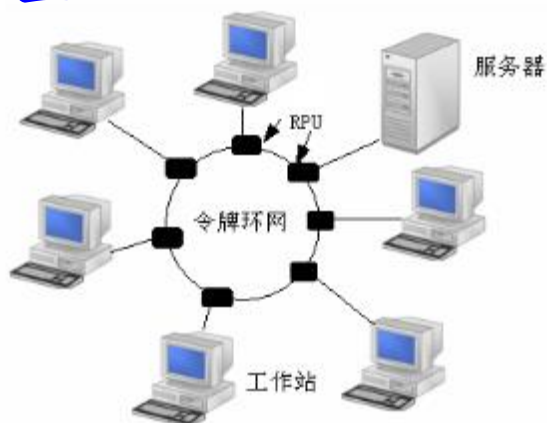
网络拓扑

• 星型



优点：结构简单，方便管理，传输效率高等
缺点：集中控制，主机负载过重，可靠性低等

• 环型



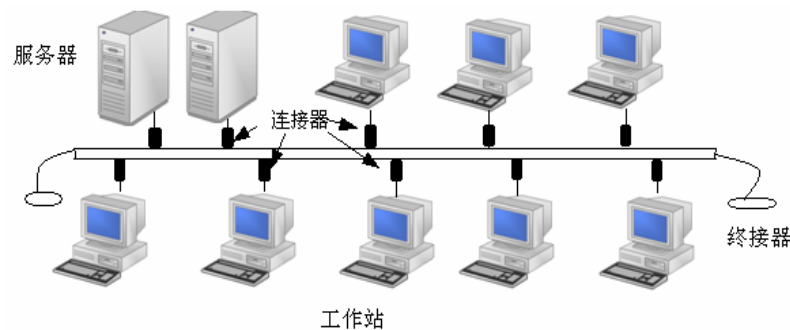
优点：令牌控制，传输控制容易
缺点：维护困难，可靠性不高

• 树型



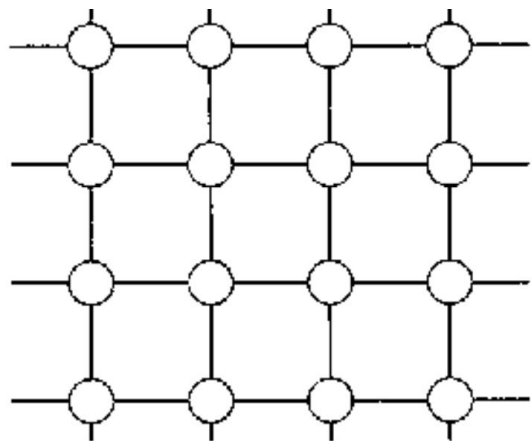
优点：易于扩充，管理较方便，故障隔离较容易
缺点：根节点如发生故障，则全网不能正常工作

• 总线型

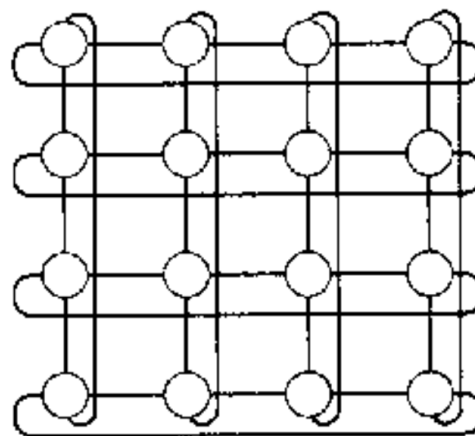


优点：费用较低，易于扩展，线路的利用率高
缺点：可靠性不高，维护困难，传输效率低

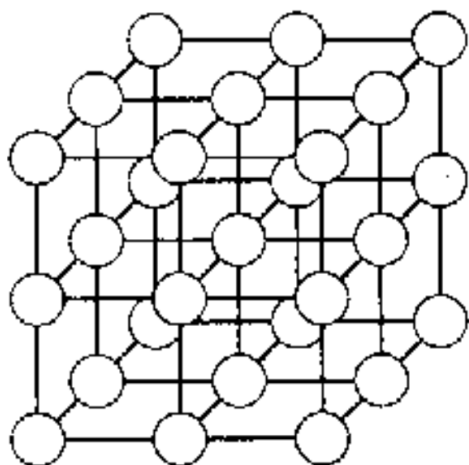
网络拓扑



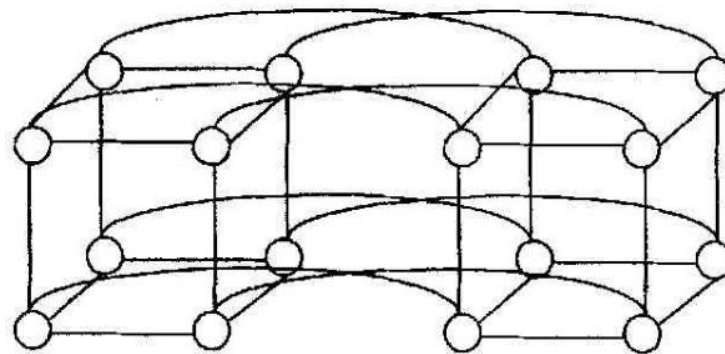
二维网格



二维环



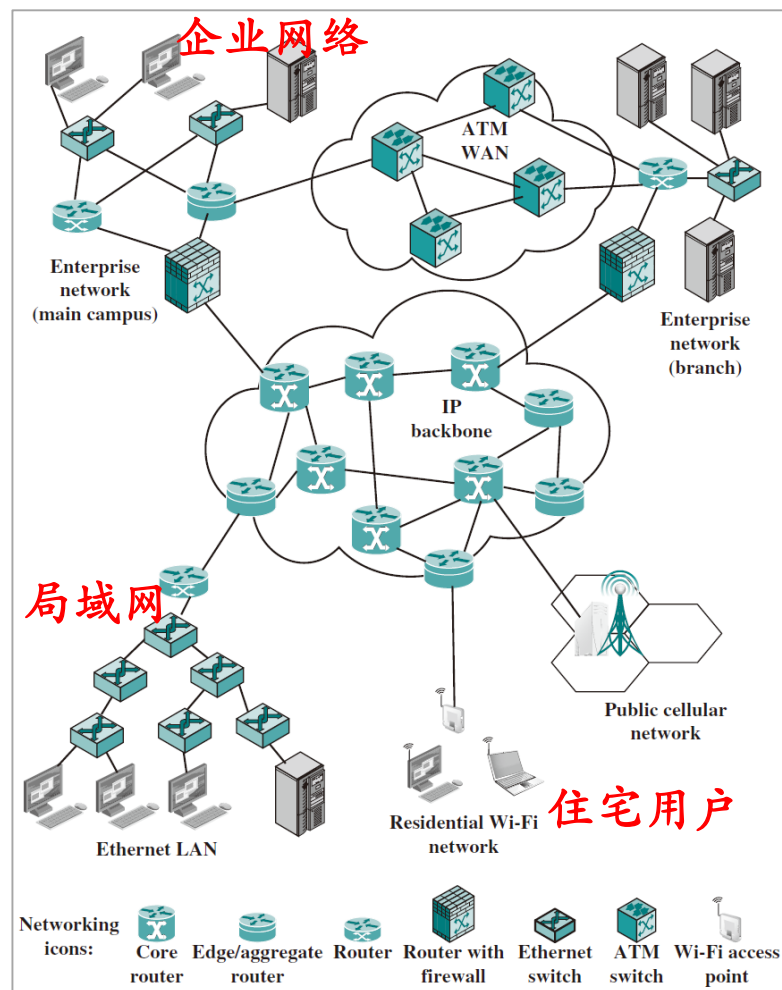
三维立方体



四维超立方体

网络配置：校园网络配置

- 需求分析
 - 用户需求分析，网络需求分析
 - 信息服务，管理需求分析
- 系统设计
 - 实用性，安全性，灵活性，可扩展性
- 总体方案
 - 网络技术，外网连接，信息点，拓扑图，防火墙
 - 网络管理方案，安全措施，病毒防范
- 产品配置与选型建议
- 施工布线要求
- 后期服务与维护



一种网络配置

校园网络配置



- 需求分析

- 用户需求分析

- 教学，科研，办公管理
 - 建筑分布，用户分布：10000用户需求

- 网络需求分析

- 千兆局域网，多网段ip端口分配，多媒体服务，
 - 多媒体教室 60人；图书馆2000人；

- 信息服务

- 数据库查询服务，www-email服务器，账号管理，

- 管理需求分析

- 支持IP组播
 - 支持虚拟网络VLAN
 - IP端口过滤

校园网络配置



- 系统设计

- 实用性：长时间服务，支持汉语，图形功能
- 安全性：国际互联保密，内部安全检查
- 灵活性：模块化组织
- 可扩展性：scale-up, scale-out

校园网络配置



- 总体方案

- 网络技术：千兆接入，万兆互联
- 外网连接：复用，教育科研网，电信网络
- 信息点：720房间/A楼… 50房间/图书馆
- 拓扑图
- 防火墙
- 网络管理方案：故障管理，配置管理，性能管理
- 安全措施：黑名单，白名单
- 病毒防范

校园网络配置



- 产品配置与选型建议
 - 10G多模光纤，100M/1G电缆接入
 - 华为3500核心网交换机，思科64口接入网交换机
 - 双机备份
- 施工布线要求
 - 环境要求，供电要求，接地要求，防火要求
 - 水平子系统，垂直干线子系统
 - 设备间子系统
- 后期服务与维护



三、协议体系结构

协议体系结构的必要性



1) 源点系统必须激活直连的数据通信通道，或者告诉通信网络它所期望的终点系统的标识。

2) 源点系统必须确定终点系统已经准备好接收数据。

数据传输涉及一系列任务

3) 源点系统上的文件发送应用程序必须确定：终点系统上的文件管理程序已经准备好为它这个特定的用户接收并存储文件。

4) 如果两个系统上使用的文件格式不一格式转换功能致，那么其中的一个系统必须执行。

协议体系结构的功能



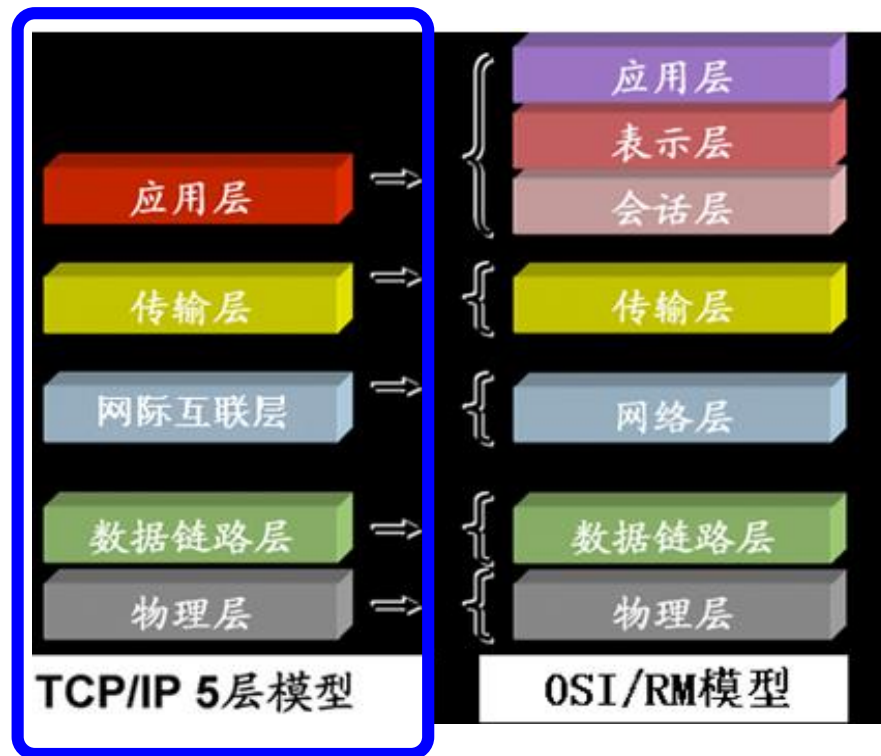
- 为了完成数据传输的任务，将整个**大任务分解为很多子任务**，然后分别独立实现每个子任务。在协议体系结构中，这些**模块竖直排列成栈**：
 - 每一层只完成与其他系统通信时所需要的的相关功能子集；
 - 依赖下一层执行的更原始的功能；
 - 向上一层提供服务；
 - 某一层发生变化，其它层不需要相应改动。

协议的关键要素



- 对等层之间的通信由受限于一组规则或规约的格式化的数据块完成，这些规则或规约称为**协议**。协议的几个关键要素如下：
 - **语法**：考虑有关数据块的格式。
 - **语义**：包含用户相互协调及差错处理的控制信息。
 - **定时关系**：包含速率匹配和数据排序。

协议体系结构



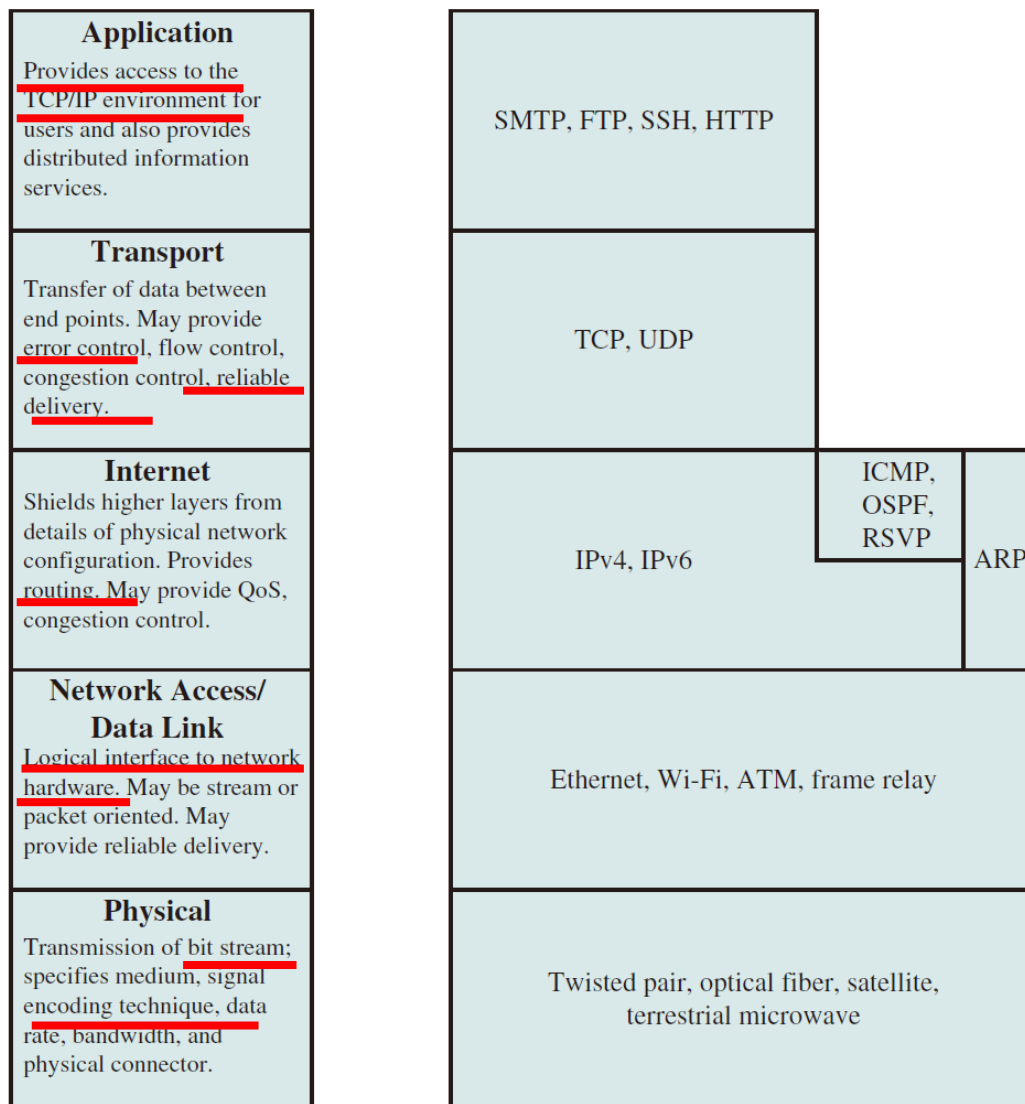
协议体系结构分层的好处:

- 为应用提供一个抽象, 对应用设计者隐藏网络的复杂性
- 促进标准化
- 各层相互独立, 技术升级和扩展灵活性好
- 便于方案设计和维护

TCP/IP协议体系结构



- **应用层：**用于支持各种不同应用程序的逻辑
- **运输层：**提供端到端的传输服务
- **网际层：**提供多个网络的路由选择功能，能够让数据跨越多个互联的网络
- **数据链路层：**为与同一个网络相连的两个系统提供网络接入
- **物理层：**负责数据传输设备与传输媒体的物理接口

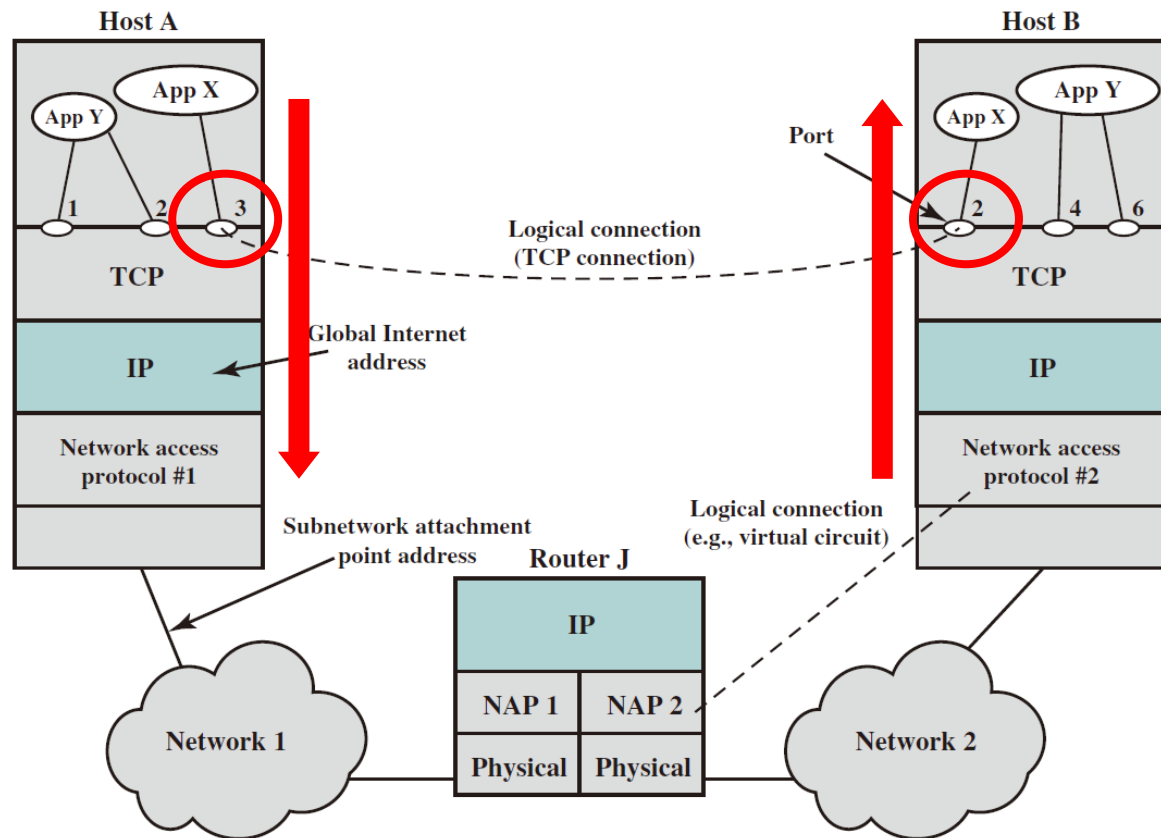


TCP/IP各层及其协议举例

TCP/IP的操作



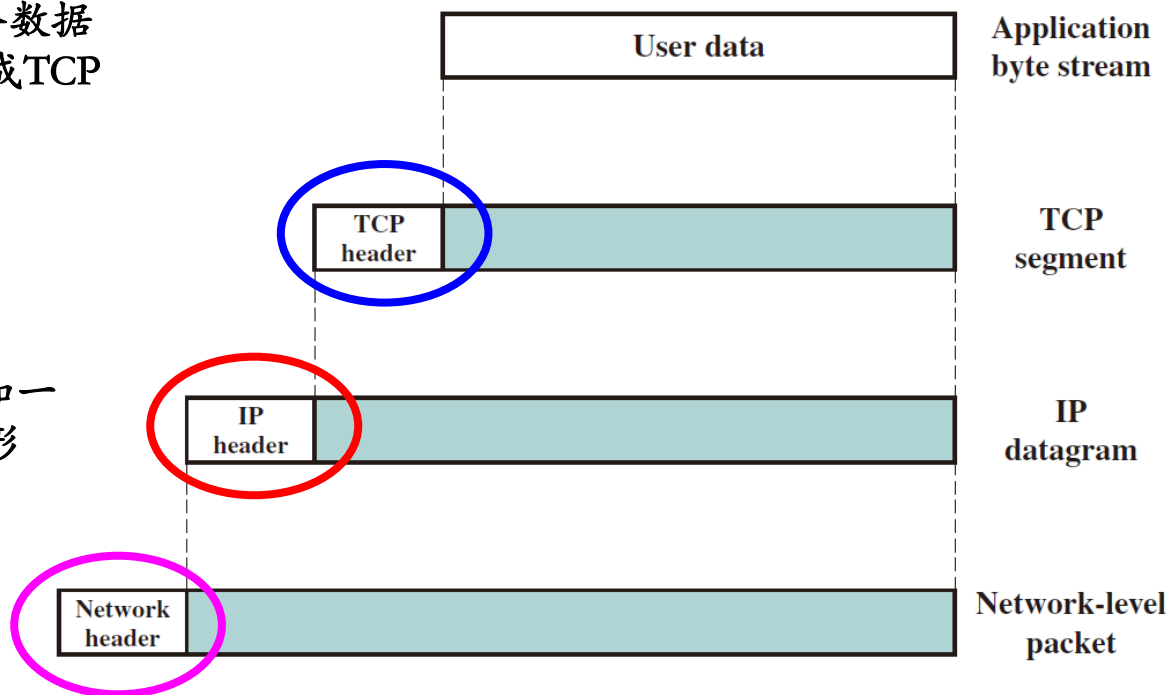
- 一个**进程**与主机A的端口3相关联，它希望向主机B上的端口2相关联的进程发送一个报文
- 主机A上的进程将报文向下递交给TCP，命令**TCP**发送给主机B的端口2
- TCP将报文向下递给**IP**，命令IP将其发送给主机B
- IP将报文向下递交给**网络接入层**，命令其将这个报文发送给路由器J



TCP/IP中的协议数据单元



- **TCP首部**：TCP为分割后的每个数据块添加一些控制信息（从而形成TCP报文段）
 - 目的端口
 - 序号
 - 检验和
- **IP首部**：IP在每个报文段上附加一个包含控制信息的首部（从而形成IP数据报）
- **网络层首部**：网络接入层附上自己的首部（从而生成分组或帧）



当数据被主机B接收后，发生的操作过程**正好相反**，每一层相应的首部被剥离，并将剩余的数据传递给上一层，知道原始的用户数据交付到目的进程。



四、套接字编程

套接字编程



套接字和套接字编程是1980s在UNIX环境下发展起来的，称为Berkeley套接字。

- **套接字**可以被认为是通信中的一个端点，是为了客户端与服务器进程之间能够相互通信。
- 可能是**面向连接**的，也可能是**无连接**的。
- 一台计算机上的**客户端套接字**通过地址调用另一台计算机上的**服务器套接字**；一旦两个合适的套接字相互约定好，便可以**交换数据**了。

套接字编程



- 具有服务器套接字的计算机要使TCP或UDP端口保持打开，随时准备接收未经预约的传入调用。
- 客户端一般通过查找域名系统（DNS）数据库来确定它所需要的服务器套接字的标识符。
- 一旦连接建立，服务器就要将对话过程交换到另一个不同的端口上，以便为后续传入的调用释放主端口号。

套接字



套接字：端口值和IP地址组合，在整个互联网中是唯一的
(在编写程序时，当套接字被用来定义API时，用三个要素标识：协议、本地地址、本地进程)

- **流套接字** (stream socket)
 - TCP，面向连接的可靠数据传输，并保证按发送时的顺序到达
- **数据报套接字** (datagram socket)
 - UDP，快速，交付没有保证，也不一定会保留初始顺序
- **原始套接字** (raw socket)
 - 直接访问底层协议，如IP

套接字接口调用



Format	Function	Parameters
socket ()	初始化套接字	domain Protocol family of the socket to be created (AF_UNIX, AF_INET, AF_INET6) type Type of socket to be opened (stream, datagram, raw) protocol Protocol to be used on socket (UDP, TCP, ICMP)
bind ()	绑定套接字到一个端口地址	sockfd Socket to be bound to the port address localaddress Socket address to which the socket is bound addresslength Length of the socket address structure
listen ()	在套接字上侦听是否有传入的连接	sockfd Socket on which the application is to listen queuesize Number of inbound requests that can be queued at any time
accept ()	接受传入的连接	sockfd Socket on which the connection is to be accepted remoteaddress Remote socket address from which the connection was initiated addresslength Length of the socket address structure
connect ()	去往服务器的外出连接	sockfd Socket on which the connection is to be opened remoteaddress Remote socket address to which the connection is to be opened addresslength Length of the socket address structure
send () recv () read () write ()	在流套接字上发送和接收数据	sockfd Socket across which the data will be sent or read data Data to be sent, or buffer into which the read data will be placed datalength Length of the data to be written, or amount of data to be read
sendto () recvfrom ()	在数据报套接字上发送和接收数据	sockfd Socket across which the data will be sent or read data Data to be sent, or buffer into which the read data will be placed datalength Length of the data to be written, or amount of data to be read
close ()	关闭套接字	sockfd Socket which is to be closed

套接字接口调用



- **套接字的建立**：通过`socket()`命令创建一个新的套接字，该命令包含`domain`，`type`，`protocol`三个参数
- `socket()`命令返回值：一个整数值，用于标识套接字，类似UNIX的文件描述符
- 侦听地址：`bind()`函数将套接字绑定到一个套接字地址

套接字接口调用



对于流套接字，一旦套接字被创建，就必须与一个远程套接字建立连接。
(一方作为客户端，并向服务器的另一方请求连接)

• 服务器端的连接：

- 服务器应用程序发出**listen()调用**，并设置传入队列种允许的连接数；
- 每个传入的连接请求被放置在队列中，直至服务器发出一个匹配的**accept()**；
- accept()调用从队列中**移出一个请求**：
 - 如果队列为空，accept()阻塞进程，知道有连接请求到达
 - 如果队列中有正在等待的调用，则返回一个新的文件描述符，一创建新的套接字；

• 客户端的连接：

- 客户端应用程序**发出connect()**，指定本地套接字以及远程套接字地址
 - 如果连接尝试失败，则connect()返回-1；
 - 如果尝试成功，则connect返回0，并在文件描述符参数中填写本地和远程的IP地址和端口号
- 一旦**连接建立**，可以通过getpeername()找出流套接字连接的另一端

套接字的通信

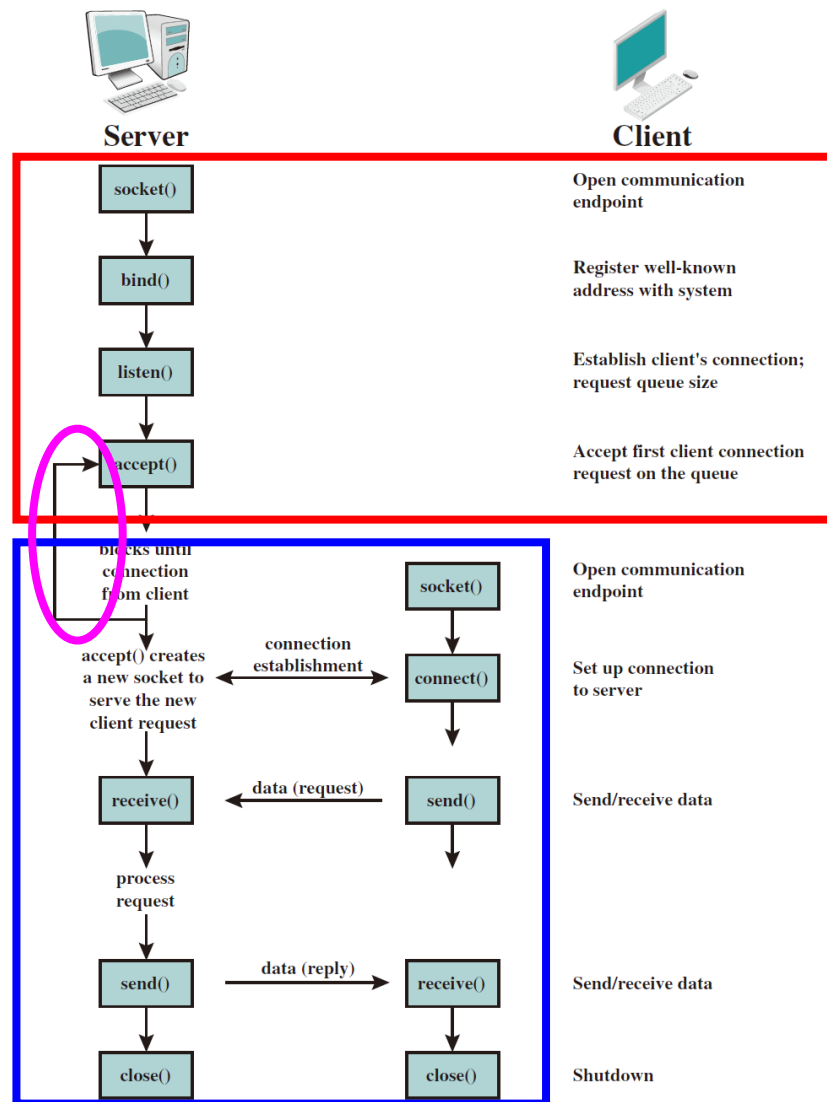


• 服务器端的连接:

- 发出listen()调用, 表明指定的套接字已做好准备接受传入的连接;
- 传入的连接请求被放置在队列中, 直至服务器发出一个匹配的accept();
- accept()为请求的连接返回一个新的文件描述符, 创建新的套接字, 以便本地应用程序可以继续侦听其他请求

• 客户端的连接:

- 发出connect(), 连接后在本地描述符中填写新的远程IP和端口号;
- 连接后, 发送接收数据



任何时候任何一方, 都可以用close()关闭连接

面向连接的协议的套接字系统调用

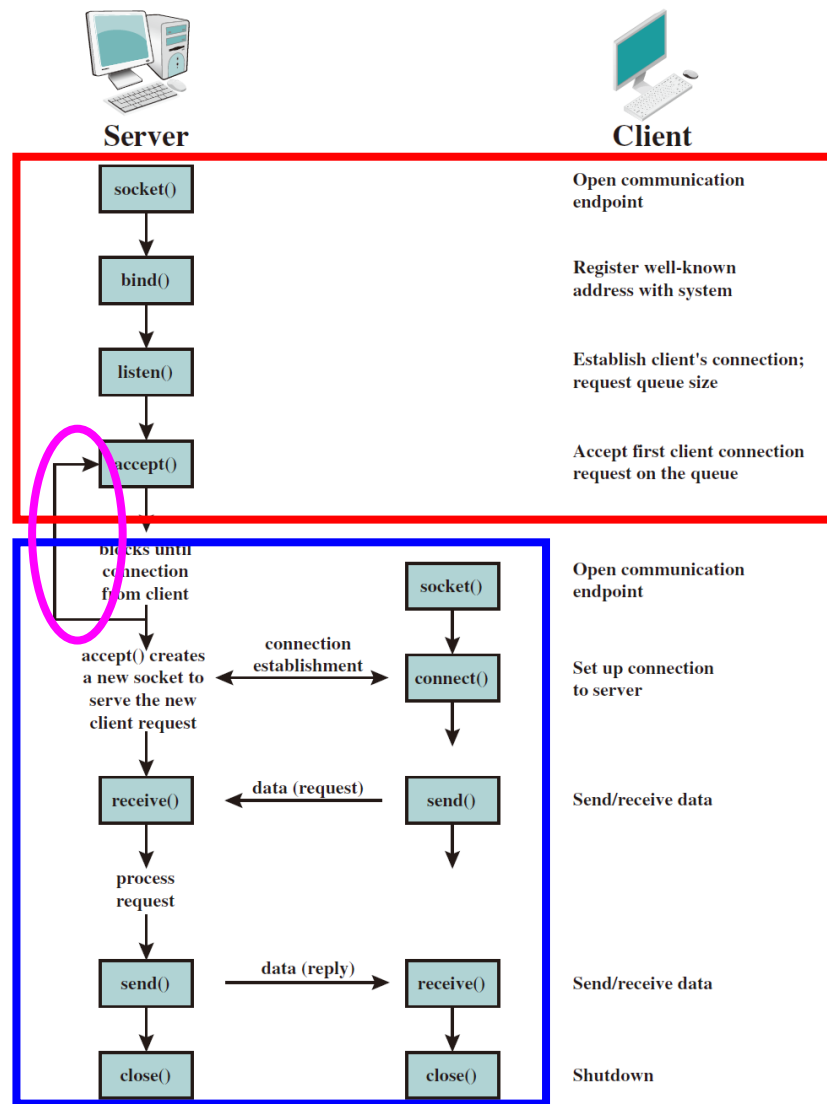
套接字的通信



• 流通信:

- 通过函数send()和recv()就可以在sockfd参数指定的连接上发送或接收数据;
- send()调用指定发送的数据块和字节数;
- recv()调用指明存储数据的缓冲区和字节数

任何时候任何一方，都可以用close()关闭连接



面向连接的协议的套接字系统调用

套接字：C语言实现的实例



• Server

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <sys/socket.h>
4 #include <netinet/in.h>

5 void error(char *msg)
6 {
7     perror(msg);
8     exit(1);
9 }

10 int main(int argc, char *argv[])
11 {
12     int sockfd, newsockfd, portno, clien;
13     char buffer[256];
14     struct sockaddr_in serv_addr, cli_addr;
15     int n;
16     if (argc < 2) {
17         fprintf(stderr, "ERROR, no port provided\n");
18         exit(1);
19     }
20     sockfd = socket(AF_INET, SOCK_STREAM, 0);
21     if (sockfd < 0)
22         error("ERROR opening socket");
23     bzero((char *) &serv_addr, sizeof(serv_addr));
24     portno = atoi(argv[1]);
25     serv_addr.sin_family = AF_INET;
26     serv_addr.sin_port = htons(portno);
27     serv_addr.sin_addr.s_addr = INADDR_ANY;
28     if (bind(sockfd, (struct sockaddr *) &serv_addr,
29         sizeof(serv_addr)) < 0)
30         error("ERROR on binding");
31     listen(sockfd, 5);
32     clien = sizeof(cli_addr);
33     newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clien);
34     if (newsockfd < 0)
35         error("ERROR on accept");
36     bzero(buffer, 256);
37     n = read(newsockfd, buffer, 255);
38     if (n < 0) error("ERROR reading from socket");
39     printf("Here is the message: %s\n", buffer);
40     n = write(newsockfd, "I got your message", 18);
41     if (n < 0) error("ERROR writing to socket");
42     return 0;
43 }
```

新套接字的创建

将套接字绑定到一个套接字地址

accept()

读取数据

• Client

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <sys/socket.h>
4 #include <netinet/in.h>
5 #include <netdb.h>

6 void error(char *msg)
7 {
8     perror(msg);
9     exit(0);
10 }

11 int main(int argc, char *argv[])
12 {
13     int sockfd, portno, n;
14     struct sockaddr_in serv_addr;
15     struct hostent *server;
16     char buffer[256];
17     if (argc < 3) {
18         fprintf(stderr, "usage %s hostname port\n", argv[0]);
19         exit(0);
20     }
21     portno = atoi(argv[2]);
22     sockfd = socket(AF_INET, SOCK_STREAM, 0);
23     if (sockfd < 0)
24         error("ERROR opening socket");
25     server = gethostbyname(argv[1]);
26     if (server == NULL) {
27         fprintf(stderr, "ERROR, no such host\n");
28         exit(0);
29     }
30     bzero((char *) &serv_addr, sizeof(serv_addr));
31     serv_addr.sin_family = AF_INET;
32     bcopy((char *) server->h_addr,
33         (char *) &serv_addr.sin_addr.s_addr,
34         server->h_length);
35     serv_addr.sin_port = htons(portno);
36     if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
37         error("ERROR connecting");
38     printf("Please enter the message: ");
39     bzero(buffer, 256);
40     fgets(buffer, 255, stdin);
41     n = write(sockfd, buffer, strlen(buffer));
42     if (n < 0)
43         error("ERROR writing to socket");
44     bzero(buffer, 256);
45     n = read(sockfd, buffer, 255);
46     if (n < 0)
47         error("ERROR reading from socket");
48     printf("%s\n", buffer);
49     return 0;
50 }
```

套接字的创建

请求连接

发送数据

套接字编程



<https://cs.nju.edu.cn/yafeng/DataComThu.htm>

Download

Code

[Socket通信代码](#)

[Socket通信代码\(Book\)](#)

总结



问题？

殷亚凤

yafeng@nju.edu.cn

<http://cs.nju.edu.cn/yafeng/>

Room 901, Building of CS

