School of Economics ang Management, Beihang University

现代程序设计技术

赵吉昌
jichang@buaa.edu.cn

本周内容

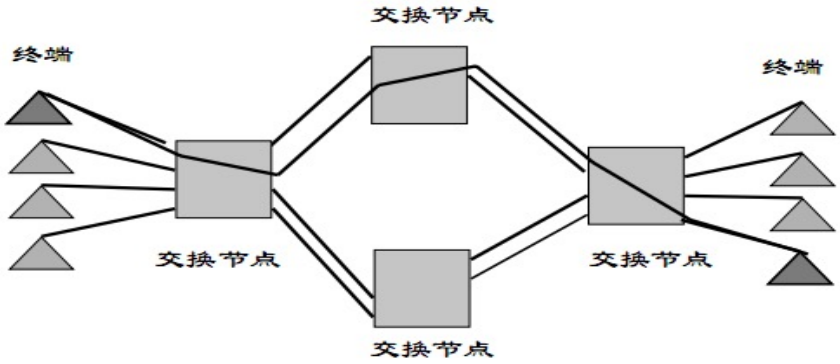
- 面向对象编程
  - 网络编程
    - TCP
    - UDP
    - SMTP
    - POP3

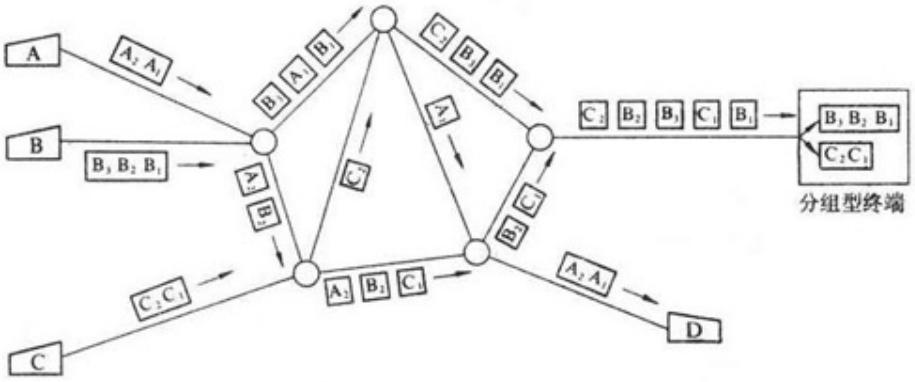
计算机网络

- 基本概念
  - 因特网
    - Internet
    - 世界范围的计算机网络，即互联网
    - 网络的网络
  - 万维网
    - WWW
    - 超文本相互链接而成，是网互联所能提供的服务之一
  - 内联网
    - Intranet
    - 专用网络，防火墙

计算机网络

- 数据交换
  - 电路交换
    - 预分配
    - 链路独享
  - 分组交换
    - 端到端时延变动，不可预测，不适合实时服务，但按需分配，带来更好的带宽共享，实现成本更低
    - 报文(package)





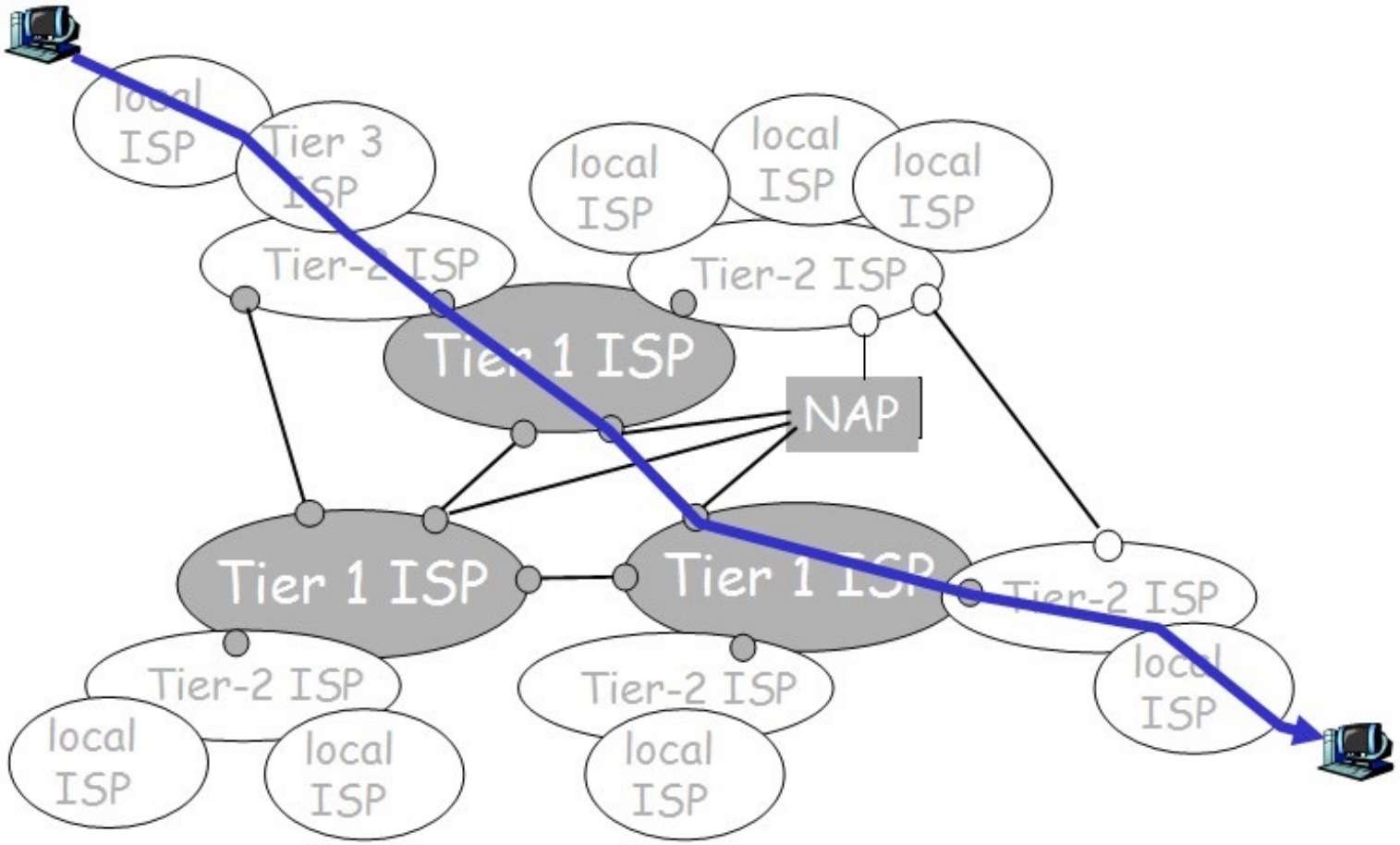
计算机网络

- 网络结构
  - ISP
    - Internet Service Provider
    - 一个或多个分组交换机在多段通信链路组成的网络
    - 提供不同类型的网络接入
    - 独立运营
  - ISP分层
    - 第一层：因特网主干
      - Sprint, AT&T, Level3
    - 第二层：具有区域性或国家性覆盖规模，引导流量通过第一层，也可以跟其他第二层ISP交换流量
      - Provider-customer
      - Peer-peer
    - 较低层：通过一个或多个第二层ISP与更大的因特网连接

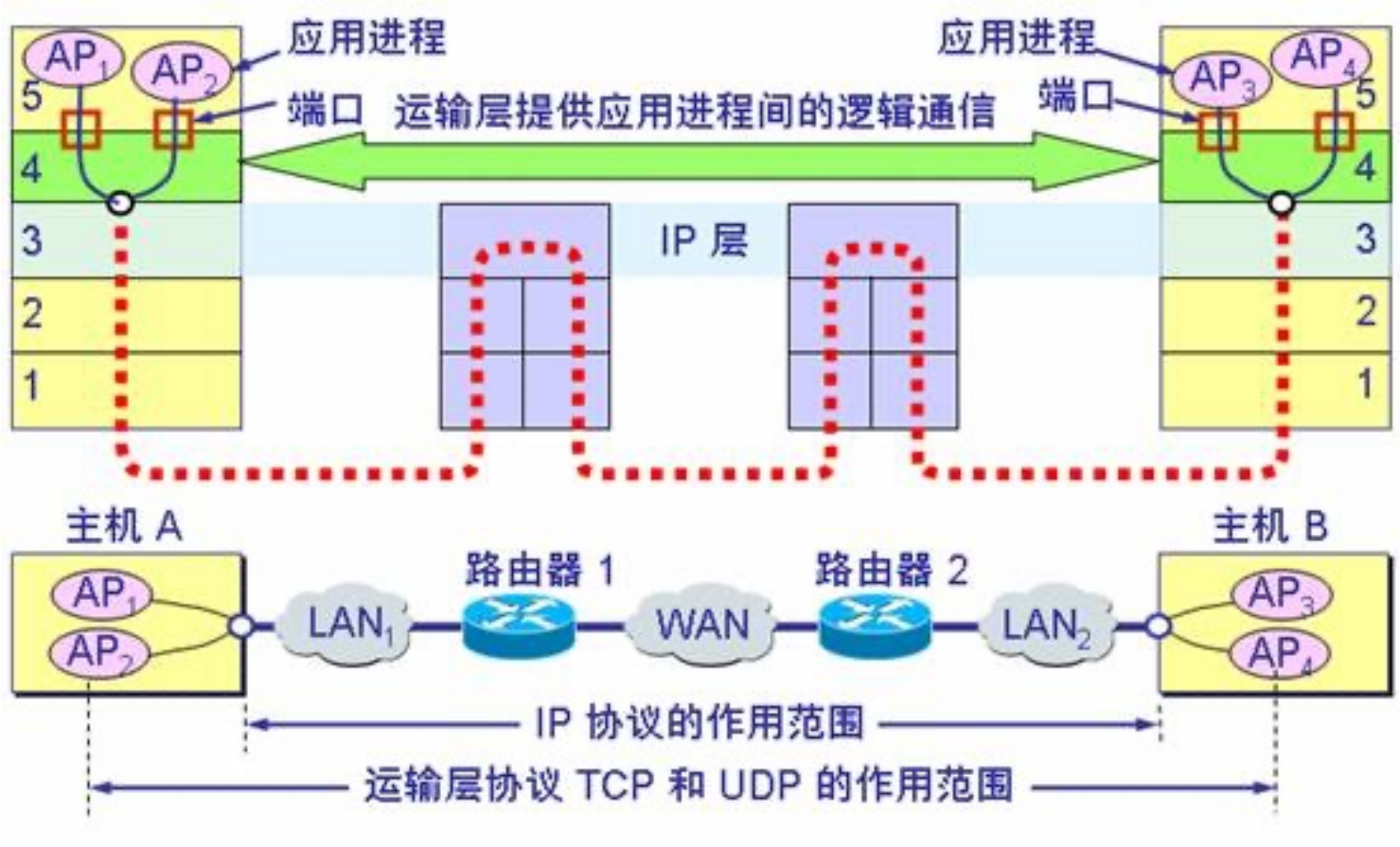
- 网络结构
  - 接入点(POP)
    - ISP与ISP的连接点，由一个或多个路由器组成
  - 网络接入点(NAP)
    - 由第三方通信公司或因特网主干提供
  - 接入方式
    - 住宅接入
      - ADSL，光纤
    - 公司接入
      - 以太网
    - 无线接入
      - 无线局域网，广域无线接入网（数万米半径）

计算机网络

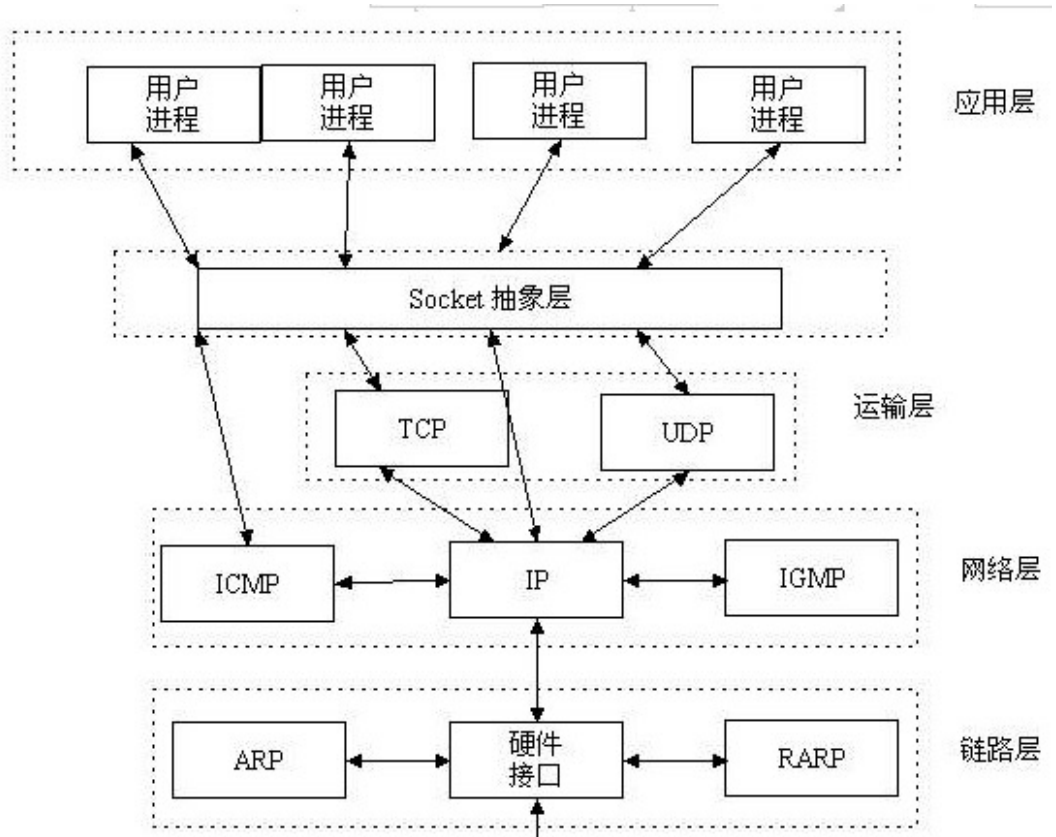
- 网络结构


local ISP
Tier 3 ISP
local ISP
local ISP
local ISP
Tier-2 ISP
Tier-2 ISP
Tier 1 ISP
NAP
Tier 1 ISP
Tier 1 ISP
Tier-2 ISP
Tier-2 ISP
Tier-2 ISP
local ISP
local ISP
local ISP
local ISP
local ISP

- 数据传输
  - 传输层（TCP, UDP)

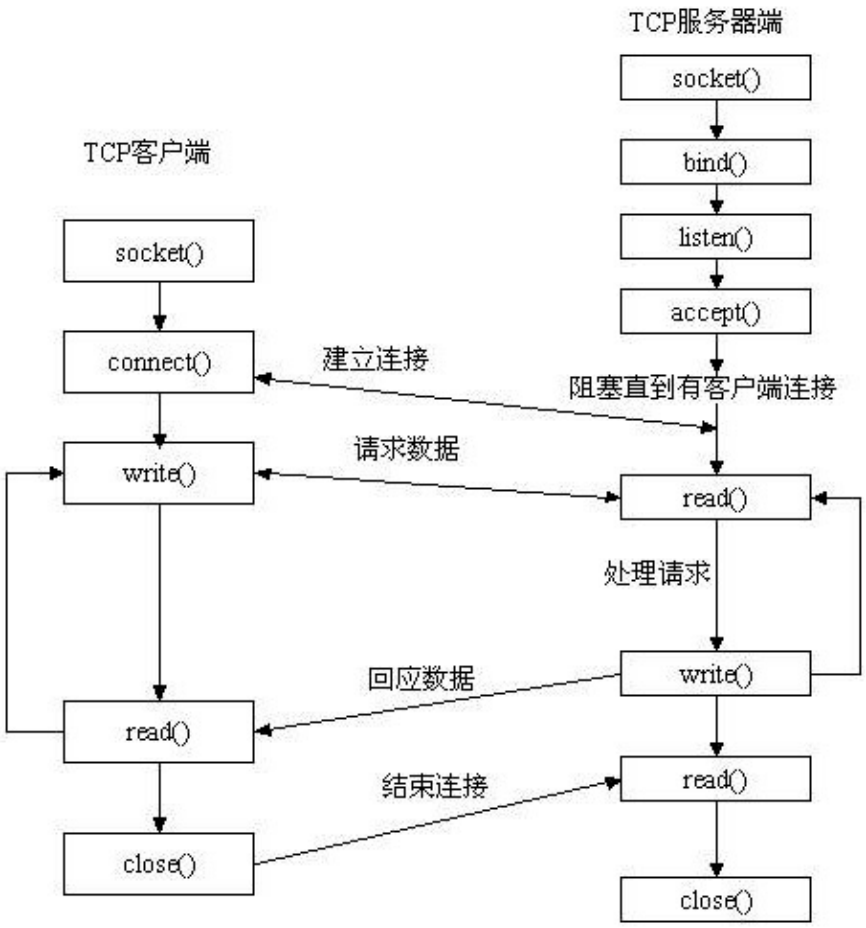


- Socket
  - 应用层与TCP/IP协议族通信的中间软件抽象层
  - `import socket`



网络编程

- TCP socket编程
  - c/s结构



网络编程

- TCP socket编程
  - sock=socket(AF_INET,SOCK_STREAM)
  - AF_INET使用IPv4协议，AF_INET6为IPv6协议
  - SOCK_STREAM指定使用面向流的TCP协议
  - bind()绑定端口，端口号应不小于1024
  - listen()监听端口，并指定等待连接的最大数量
  - accept()等待并返回一个客户端连接
  - connect()主动连接服务端
  - recv()接收TCP数据
  - send()发送TCP数据

网络编程

- Demo:
  - `tcpserver.py, tcpclient.py`
  - `mtserver.py, mtclient.py`
  - `schat.py`

Http

- http.server
  - Defines classes for implementing HTTP servers (Web servers)
    - `HttpServer/ThreadingHttpServer`: creates and listens at the HTTP socket, dispatching the requests to a handler
    - Handler: `BaseHTTPRequestHandler`, `SimpleHTTPRequestHandler`
- http.client
  - Defines classes which implement the client side of the HTTP and HTTPS protocols
  - It is normally not used directly
  - `urllib.request` uses it to handle URLs that use HTTP and HTTPS
- Demo
  - `httpsdemo.py httpcdemo.py`

- UDP的优势
  - 应用层能更好地控制要发送的数据和发送时间
  - 无需连接建立
  - 不需要任何准备即可进行数据传输
  - 不引入建立连接的时延
  - 无连接状态
  - 无需维护一些状态参数
  - 分组头部开销小
    - TCP头至少占20个字节，UDP头占8个字节

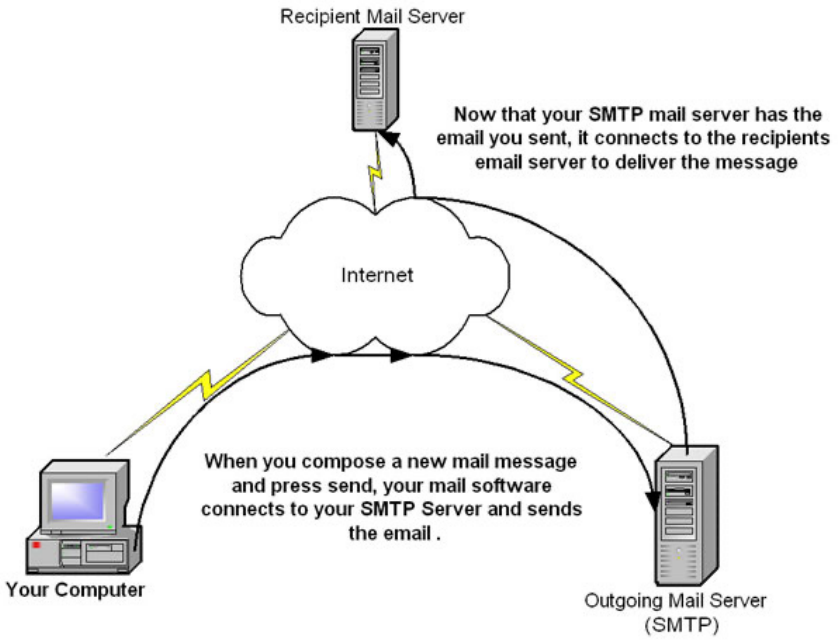
网络编程

- UDP socket编程
  - socket(AF_INET, SOCK_DGRAM)
  - bind()绑定端口，端口号应不小于1024
  - recvfrom()接收UDP数据
  - sendto()发送UDP数据
- Demo
  - udpserver.py
  - udpclient.py
  - campost.py(client)
  - camrecieve.py(server)

- SMTP
  - Simple Mail Transfer Protocol
  - 从发送方的邮件服务器向接收方的邮件服务器发送邮件
  - 运行在发送方邮件服务器的客户机端和接收方邮件服务器的服务器端
    - 邮件服务器同时有SMTP客户端和服务端
  - 在25端口建立与服务器的TCP连接
    - HELO, MAIL FROM, RCPT TO, DATA, QUIT
  - MIME: 多用途互联网邮件扩展

网络编程

- SMTP编程
  - email模块负责邮件构建
  - smtplib模块邮件发送
  - Demo
    - smtp.py
    - smtp_att.py



网络编程

- POP3
  - 与服务器建立连接（默认是110端口）
  - 认证→事务处理→更新
    - list, retr, dele, quit
  - 未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
- IMAP
  - 互联网邮件访问协议
  - 创建文件夹并在文件夹之间移动邮件
  - 允许只读取报文头部

网络编程

- POP3编程
  - 用poplib下载原始邮件
  - 用email解析并还原为邮件对象
  - Demo
    - `pop.py`

网络编程

- IMAP编程
  - imaplib模块
  - `M = imaplib.IMAP4()`
  - `M.login(user,pass)`
  - `M.select()#选择文件夹，默认是INBOX`
  - `typ, data = M.search(None, 'ALL')`
  - `for num in data[0].split():`
  - `    typ, data = M.fetch(num)`
  - `M.close()`
  - `M.logout()`

网络编程

- 补充：其他的一些库
  - Twisted
    - 支持多种网络协议
  - Tornado
    - Web开发
    - 异步
    - 协程及并发
  - 建议了解
    - 可能在前后端连接的时候使用
      - 如后端做机器学习等服务，前端获取输入并显示结果

本周作业

- 利用socket和多线程，实现支持多人对话的聊天室。具体地，实现Manager和Chatter 两个类，Chatter只需和Manager之间建立一对一联系，而Manager则负责广播或转发所有用户的消息。相关要求如下：
  - 1. 实现Manager类, 服务器，管理成员进入和离开聊天室，接收成员消息并广播
  - 2. 实现Chatter类, 用户, 向管理员发送加入和退出请求，发送和接收消息
  - 3. Manager类使用多线程服务多个用户
  - 4. Chatter用户发送和接收消息需要依赖不同线程进行
  - 5. Manager类具备定向转发功能，比如Chatter可以在消息中通过@指定特定用户，这样Manager将仅转发给被指定用户。
  - 6. Chatter在离开时，自动保存聊天记录到硬盘（包括时间、发信人，信息）。
  - 7. Manager也应保存所有聊天室记录到硬盘。