IP

- 私有ip
  - 10.0.0.0～10.255.255.255
  - 172.16.0.0～172.31.255.255 IP地址127．0．0．1~127．255．255．255用于回路测试
  - 192.168.0.0～192.168.255.255
- 常用C类
  192.168.1.1-254 每一个网络段可以有254台机器

端口

```
目标IP：
源IP：
目标端口：
源端口：
信息内容：
```

- 知名端口（不要轻易使用）
  范围：0-1023

```
80：HTTP
21:FTP
```

- 动态端口
  1024-65535
- 查看端口
  netstat -an
- 端口绑定问题
  - 一个udp网络程序，可以不绑定，此时操作系统会随机进行分配一个端口，如果重新运行此程序端口可能会发生变化
  - 一个udp网络程序，也可以绑定信息（ip地址，端口号），如果绑定成功，那么操作系统用这个端口号来进行区别收到的网络数据是否是此进程的

TCP UDP

- UDP
- TCP
  - 介绍： TCP协议，传输控制协议（英语：Transmission Control Protocol，缩写为 TCP）是一种面向连接的、可靠的、基于字节流的传输层通信协议
  - TCP注意事项点：
  - tcp服务器一般情况下都需要绑定，否则客户端找不到这个服务器
  - tcp客户端一般不绑定，因为是主动链接服务器，所以只要确定好服务器的ip、port等信息就好，本地客户端可以随机
  - tcp服务器中通过listen可以将socket创建出来的主动套接字变为被动的，这是做tcp服务器时必须要做的
  - 当客户端需要链接服务器时，就需要使用connect进行链接，udp是不需要链接的而是直接发送，但是tcp必须先链接，只有链接成功才能通信
  - 当一个tcp客户端连接服务器时，服务器端会有1个新的套接字，这个套接字用来标记这个客户端，单独为这个客户端服务
  - listen后的套接字是被动套接字，用来接收新的客户端的链接请求的，而accept返回的新套接字是标记这个新客户端的
  - 关闭listen后的套接字意味着被动套接字关闭了，会导致新的客户端不能够链接服务器，但是之前已经链接成功的客户端正常通信。
  - 关闭accept返回的套接字意味着这个客户端已经服务完毕
  - 当客户端的套接字调用close后，服务器端会recv解堵塞，并且返回的长度为0，因此服务器可以通过返回数据的长度来区别客户端是否已经下线
- 三次握手、四次挥手
  - TCP为什么是三次握手，为什么不是两次或者四次握手？
  - 为什么不是两次握手：
    - 关键点是TCP是全双工通信，即客户端在给服务器端发送信息的同时，服务器端也可以给客户端发送信息，简化模型为：
    - 第一次握手： A给B打电话说，你可以听到我说话吗？(把SYN比作询问)
    - 第二次握手： B收到了A的信息，然后对A说： 我可以听得到你说话啊(把ACK比作回答)，你能听得到我说话吗？
    - 第三次握手： A收到了B的信息，然后说可以的，我要给你发信息啦
    - 补充说明：而半双工的意思是A可以给B发，B也可以给A发，但是A在给B发的时候，B不能给A发，即不同时，为半双工。单工为只能A给B发，B不能给A发； 或者是只能B给A发，不能A给B发。
  - 为什么不是四次握手：
    - 三次可以完成的事情，四次浪费网络资源
  - TCP为什么是四次挥手，而不是三次挥手或者五次挥手？
  - 关键点还是在于TCP是全双工通信，简化模型为：
    - 第一次挥手：A:“喂，我不说了 (FIN)。”，此时A->FIN_WAIT1，即A处于等待结束状态，需要B的回复
    - 第二次挥手：B:“我知道了(ACK)。等下，上一句还没说完。Balabala…..（传输数据）”，B->CLOSE_WAIT | A->FIN_WAIT2，此时B在传输数据等待结束，同时回复A已收信息，
    - 第三次挥手： B:”好了，说完了，我也不说了（FIN）。”B->LAST_ACK
    - 第四次挥手：A:”我知道了（ACK）。”A->TIME_WAIT | B->CLOSED
    - A等待2MSL,保证B收到了消息,否则重说一次”我知道了”,A->CLOSED
  - 补充说明：MSL 是Maximum Segment Lifetime英文的缩写，中文可以译为“报文最大生存时间”，他是任何报文在网络上存在的最长时间，超过这个时间报文将被丢弃。RFC 793中规定了MSL为2分钟，实际应用中常用的是30秒，1分钟和2分钟等

- 四次挥手释放连接时，等待2MSL的意义？

- 第一，为了保证A发送的最有一个ACK报文段能够到达B。这个ACK报文段有可能丢失，因而使处在LAST-ACK状态的B收不到对已发送的FIN和ACK 报文段的确认。B会超时重传这个FIN和ACK报文段，而A就能在2MSL时间内收到这个重传的ACK+FIN报文段。接着A重传一次确认。

- 第二，就是防止上面提到的已失效的连接请求报文段出现在本连接中，A在发送完最有一个ACK报文段后，再经过2MSL，就可以使本连接持续的时间内所产生的所有报文段都从网络中消失

- 为什么在四次挥手的过程中一般都是客户端先发起呢?

- 关键点在于：TCP挥手时要等2MSL

    - 现象：在调试客户端和服务器端（使用TCP套接字）的代码时我发现，如果先结束服务器端后结束客户端，紧接着再重启服务器端就会出现绑定失败的错误 OSError: [Errno 98] Address already in use 等待一段时间后大概一分钟左右就能正常重启服务器端
    - 分析：
    - 因为TCP是全双工的，所以客户端和服务端都可以先进行挥手。在socket编程中哪一方先执行close()操作，哪一方则先进行挥手（发送FIN包）。
        - 以客户端先挥手为例，在TCP处于TIME_WAIT状态时，客户端从TIME_WAIT状态到CLOSED状态需要2MSL，因为客户端要确定服务端收到了ACK，如果服务端没收到ACK，客户端则一定会在2MSL时间内再收到一次FIN。而在socket编程中客户端一般不需要绑定，而服务器端一般都要绑定，如果先结束服务器端则是服务器端先进行挥手操作，那么服务器端从TIME_WAIT到CLOSED状态则需要2MSL。这段时间服务器端绑定的端口号被占用了，套接字不会释放。所以这段时间重启了服务器之后，会出现绑定失败的错误OSError: [Errno 98] Address already in use

- tcp长连接和短连接

    - TCP长/短连接的优点和缺点:
    - 长连接可以省去较多的TCP建立和关闭的操作，减少浪费，节约时间。对于频繁请求资源的客户来说，较适用长连接。
    - client与server之间的连接如果一直不关闭的话，会存在一个问题，随着客户端连接越来越多，server早晚有扛不住的时候，这时候server端需要采取一些策略，如关闭一些长时间没有读写事件发生的连接，这样可以避免一些恶意连接导致server端服务受损,如果条件再允许就可以以客户端机器为颗粒度，限制每个客户端的最大长连接数，这样可以完全避免某个蛋疼的客户端连累后端服务。
    - 短连接对于服务器来说管理较为简单，存在的连接都是有用的连接，不需要额外的控制手段。
    - 但如果客户请求频繁，将在TCP的建立和关闭操作上浪费时间和带宽。

- TCP长/短连接的应用场景：

    - 数据库的连接用长连接，如果用短连接频繁的通信会造成socket错误，而且频繁的socket 创建也是对资源的浪费
    - 而像WEB网站的http服务一般都用短链接，因为长连接对于服务端来说会耗费一定的资源，而像WEB网站这么频繁的成千上万甚至上亿客户端的连接用短连接会更省一些资源
    - 如何区分：请求头中有：Connection: keep-alive

TCP/IP协议

- 互联网协议包含了上百种协议标准，但是最重要的两个协议是TCP和IP协议，所以，大家把互联网的协议简称TCP/IP协议(族)

应用层协议

- 域名系统(Domain Name System，DNS)：用于实现网络设备名字到IP地址映射的网络服务

    - DNS解析的过程：

    i. 浏览器缓存:当用户通过浏览器访问某域名时，浏览器首先会在自己的缓存中查找是否有该域名对应的IP地址（若曾经访问过该域名且没有清空缓存便存在）；

    2. 系统缓存:当浏览器缓存中无域名对应IP则会自动检查用户计算机系统Hosts文件DNS缓存是否有该域名对应IP;

    3. 路由器缓存:当浏览器及系统缓存中均无域名对应IP则进入路由器缓存中检查，以上三步均为客服端的DNS缓存;

    4. ISP（互联网服务提供商）DNS缓存:当在用户客服端查找不到域名对应IP地址，则将进入ISP DNS缓存中进行查询。比如你用的是电信的网络，则会进入电信的DNS缓存服务器中进行查找;

    5. 根域名服务器:当以上均未完成，则进入根服务器进行查询。全球仅有13台根域名服务器，1个主根域名服务器，其余12为辅根域名服务器。根域名收到请求后会查看区域文件记录，若无则将其管辖范围内顶级域名（如.com）服务器IP告诉本地DNS服务器;

    6. 顶级域名服务器:顶级域名服务器收到请求后查看区域文件记录，若无则将其管辖范围内主域名服务器的IP地址告诉本地DNS服务器;

    7. 主域名服务器:主域名服务器接受到请求后查询自己的缓存，如果没有则进入下一级域名服务器进行查找，并重复该步骤直至找到正确纪录;

    8. 保存结果至缓存:本地域名服务器把返回的结果保存到缓存，以备下一次使用，同时将该结果反馈给客户端，客户端通过这个IP地址与w eb服务器建立链接。

- DNS主要基于UDP运输层协议，这里解释下为什么使用UDP（User Datagram Protocol）这样的无连接的?

    - 尽最大能力交付的不可靠数据连接，而不是使用TCP(Transmission Control Protocol 传输控制协议)这样的面向连接的可靠数据连接。 一次UDP名字服务器交换可以短到两个包：一个查询包、一个响应包。一次TCP交换则至少包含9个包：三次握手初始化TCP会话、一个查询包、一个响应包以及四次分手的包交换。 考虑到效率原因，TCP连接的开销大得，故采用UDP作为DNS的运输层协议

- HTTP协议

    - 超文本传输协议(HyperText Transfer Protocol，HTTP)，基于 TCP(Transfer Control Protocol, 传输控制协议)的可靠传输

并发编程IO多路复用