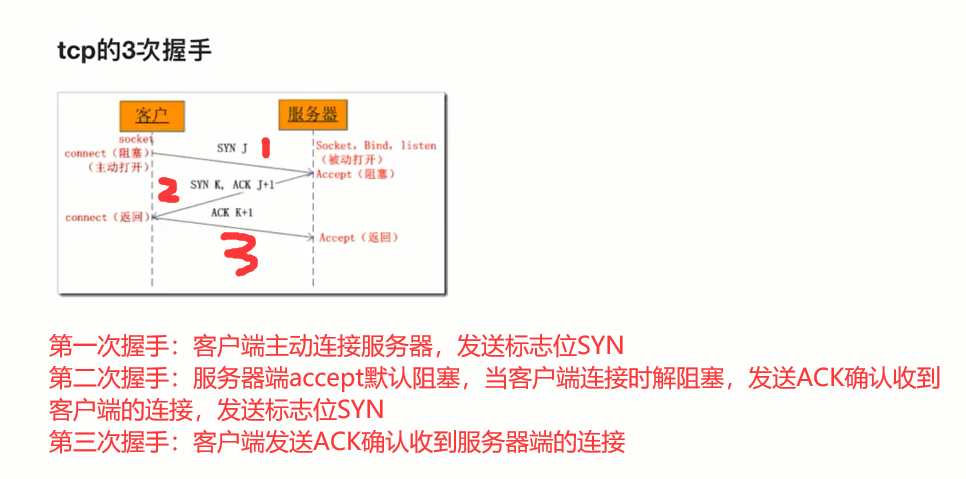
**http协议**

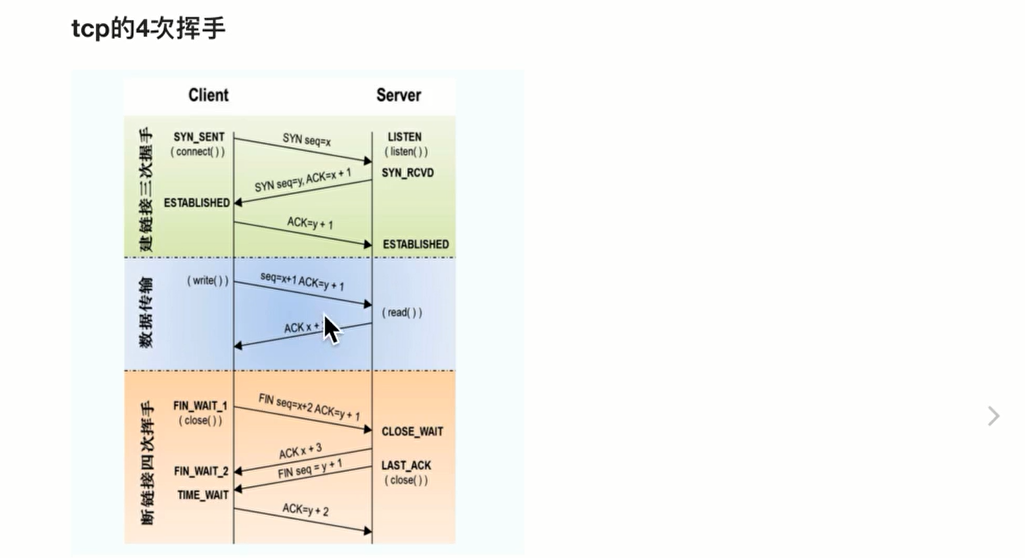
http协议是基于tcp协议的，所以http服务器的实现可以套用tcp服务器实现方式

http三次握手：

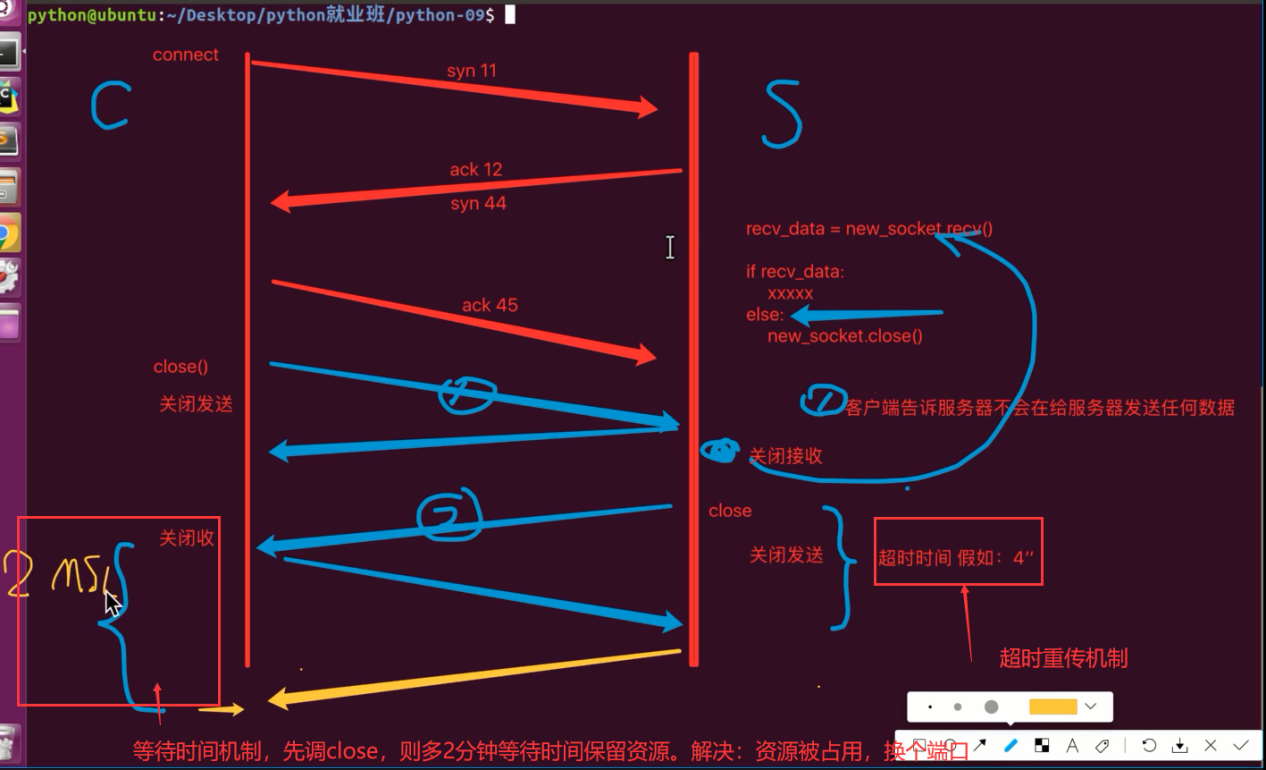


只有三次握手的原因：为了提高tcp传输效率，节省时间。第二步发送了确认位和标志位

http四次挥手：



http超时重传和等待时间概念：



浏览器返回需要的页面步骤：

1、将客户端请求解码成字符串，然后对字符串splitlines，re模块提取/api地址

2、然后打开相应的文件，在文件中查找，若匹配到则返回页面，匹配不到，返回file not found

def server\_response(new\_socket):

'''

为客户端服务提供数据

1、先收到客户端的http请求， 格式如： GET / HTTP/1.1 ......

2、构造返回给客户端的应答数据， 格式如： HTTP/1.1 200 OK ......

'''

# recv()方法1024，表示最大接收1024个字节，返回的是普通数据

client\_request = new\_socket.recv(1024).decode('utf-8')

# print(client\_request) # 调试看客户端请求,数据是字节型数据，需要decode，解码后为字符串

# 对客户端请求解码后的字符串进行切割

client\_request\_line = client\_request.splitlines()

print(client\_request\_line)

# re匹配请求的url,格式如：GET /06table.html HTTP/1.1

ret = re.match(r'[^/]+/([^ ]\*)', client\_request\_line[0])

file\_name = '' # 如果下面if语句不成立，则file\_name变量不存在，需要先定义file\_name变量

if ret:

file\_name = ret.group(1)

print(file\_name)

if ret == '/':

file\_name = '06table.html'

try:

# 发送用户请求的页面

f = open(r"C:\Users\yolo\Desktop\web\web01" + "\\" + file\_name, 'rb')

except:

response = "HTTP/1.1 404 NOT FOUND\r\n"

response += "\r\n"

response += "----file not found ----"

new\_socket.send(response.encode('utf-8'))

else:

response = "HTTP/1.1 200 OK\r\n" # python中用\r\n表示换行

response += "\r\n"

# 发送固定页面数据

# response += "<h1>YOLo</h1>"

html\_content = f.read() # 为字符型数据

f.close()

# 发送response header

new\_socket.send(response.encode('utf-8'))

# 发送response body

new\_socket.send(html\_content)

# 关闭套接字

new\_socket.close() # 因为tcp的机制，先关闭的服务器端的close，则有2分钟等待时间保留资源。要么更换端口，或者设定套接字

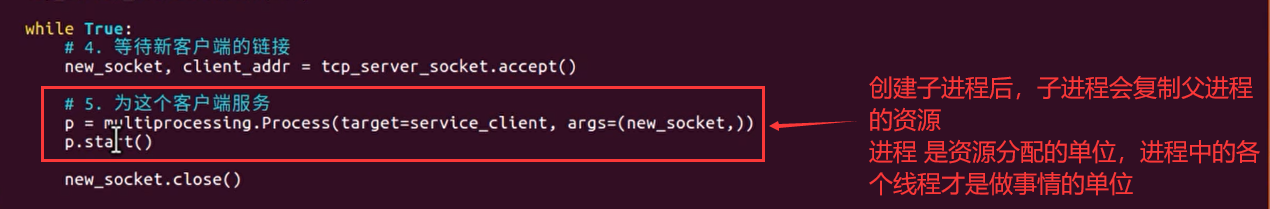
Request Headers：告诉服务器，浏览器的一些信息

Request Body：向服务器提交的数据

Response Headers：告诉浏览器，服务器的一些信息

Response Body：浏览器要展示的内容

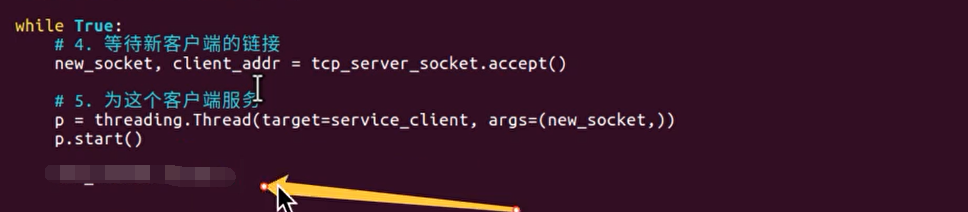
使用多进程实现http服务器：



主要代码如上。

使用多线程实现http服务器：

多线程共享全局变量，若while循环中主线程中new\_socket对象调用close，子线程中再使用new\_socket则会直接报错。错误为：bad file descriptor



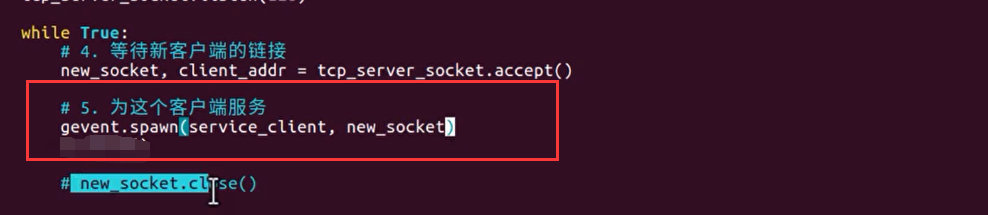
使用协程实现http服务器：协程之间共享全局变量，占用资源最少

Gevent遇到延时操作才切换任务

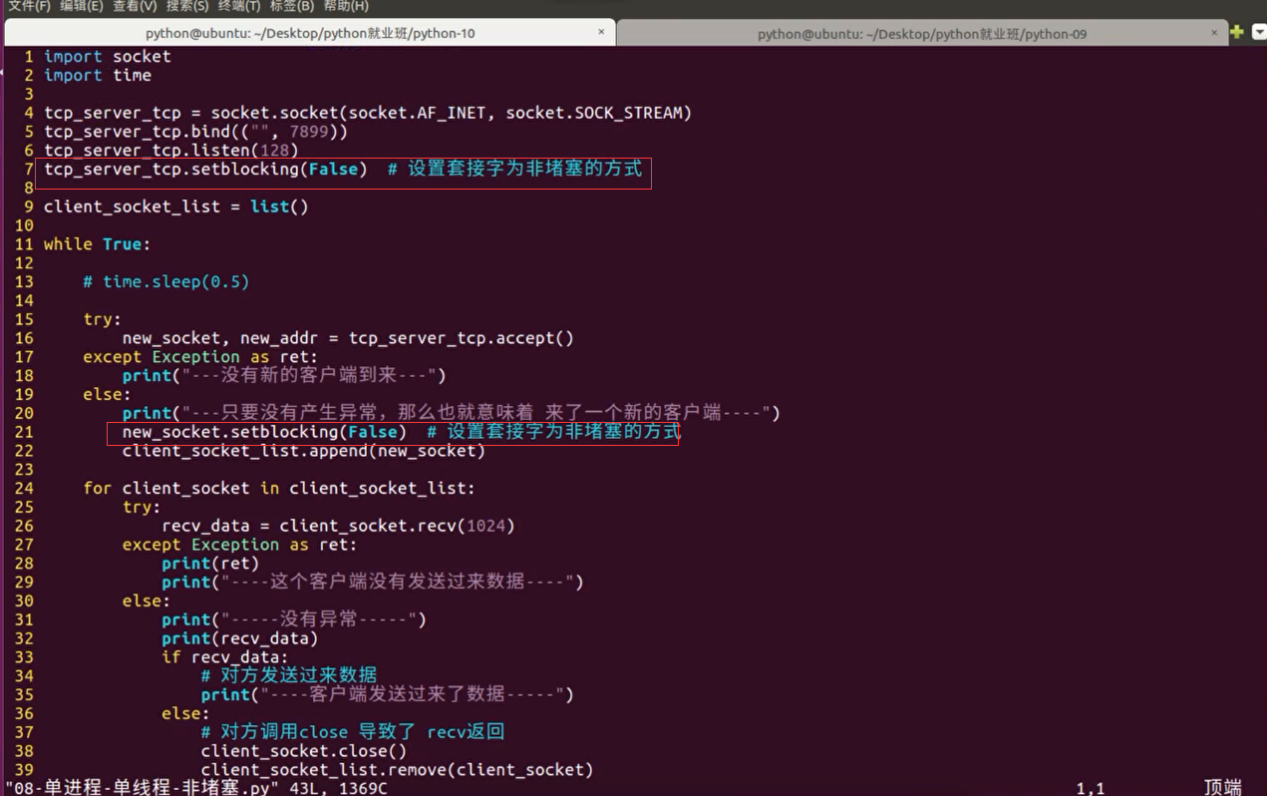
Import gevent

From gevent import monkey

Monkey.patch\_all()



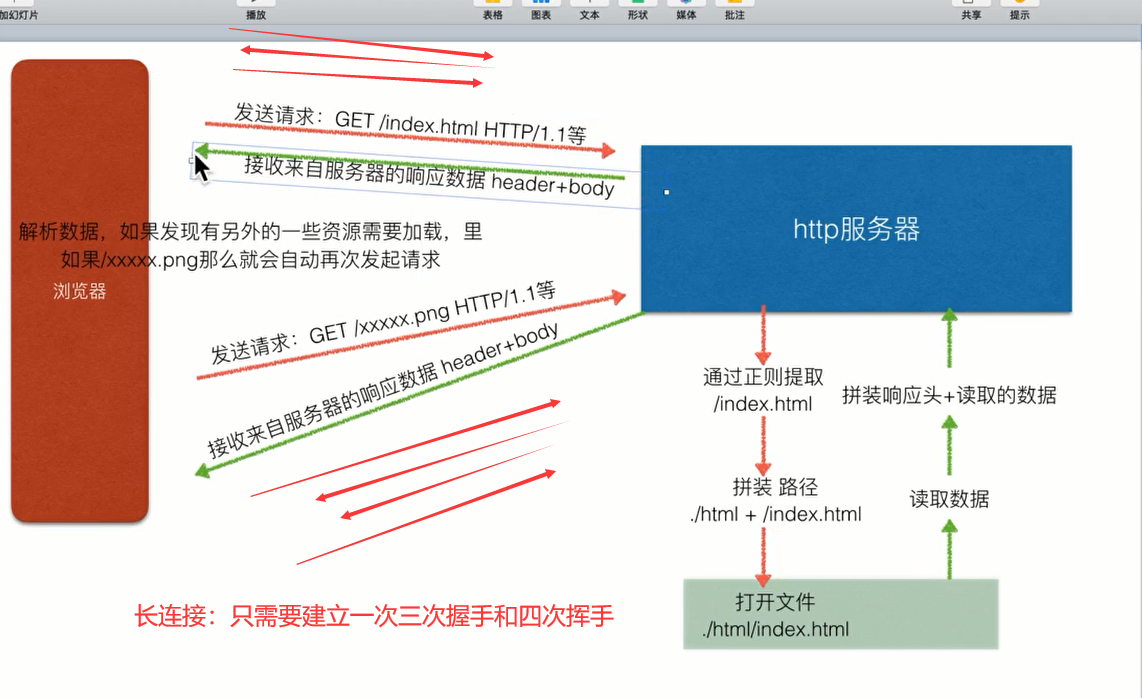
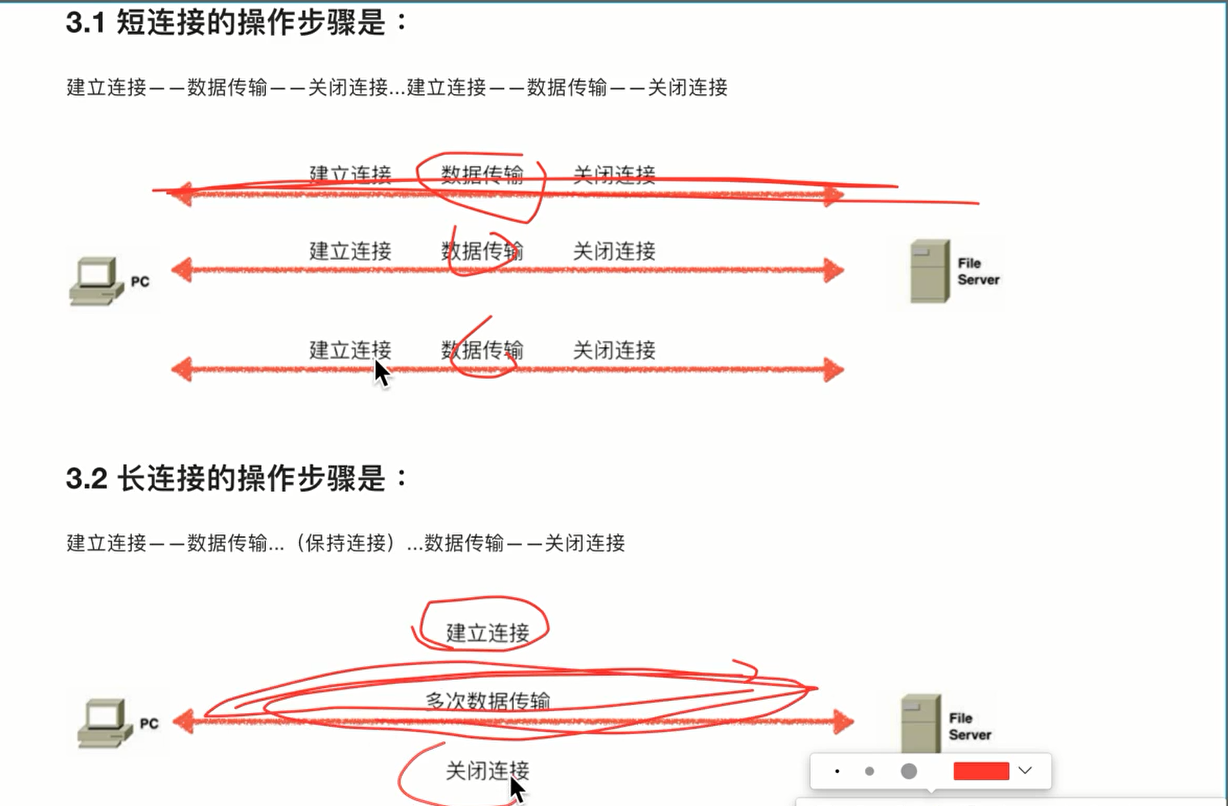
单进程-单线程-非阻塞的方式实现并发的验证：



原理：

1. 监听套接字accept，若有新的客户端连接，则把new\_socket存储在列表中。若没有新的客户端连接则抛出异常
2. 循环列表中的客户端
3. new\_socket的recv，若未收到数据则抛出异常，若收到数据，则判断数据是否为空
   1. 数据为空，则表示客户端调用了close。此时则可以关闭套接字，并从列表中移出该客户端
   2. 数据不为空，则表示客户端正常发送数据

长连接和短连接：

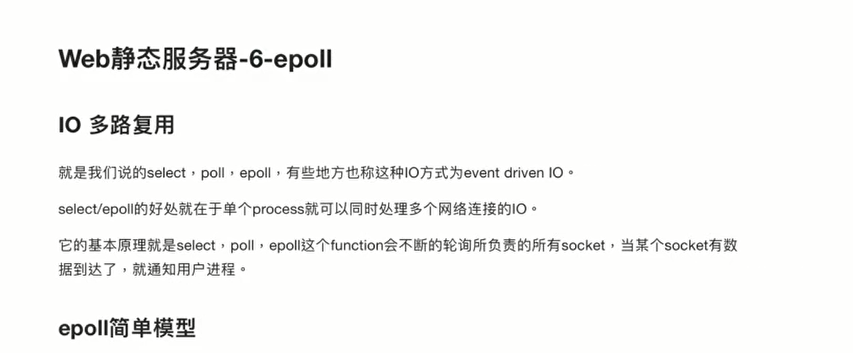


短连接：获取每一个数据，就要进行tcp 的三次握手和四次挥手

长连接：进行三次握手，然后获取想要的所有数据，然后再四次挥手

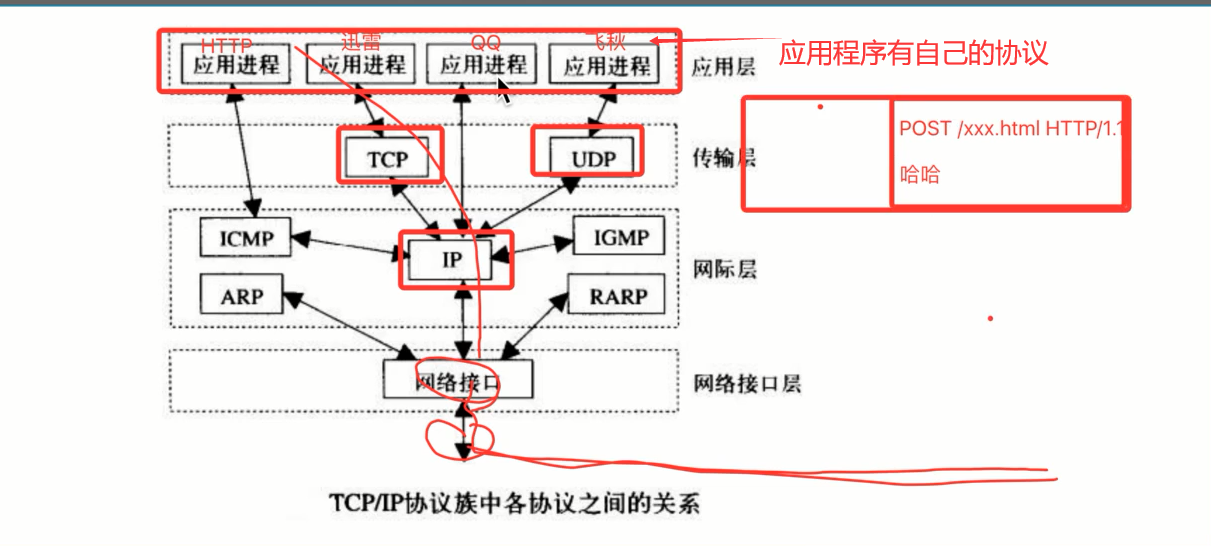
长连接优点：浏览器尽可能用一个套接字传输数据，减少服务器资源

Epoll原理讲解：

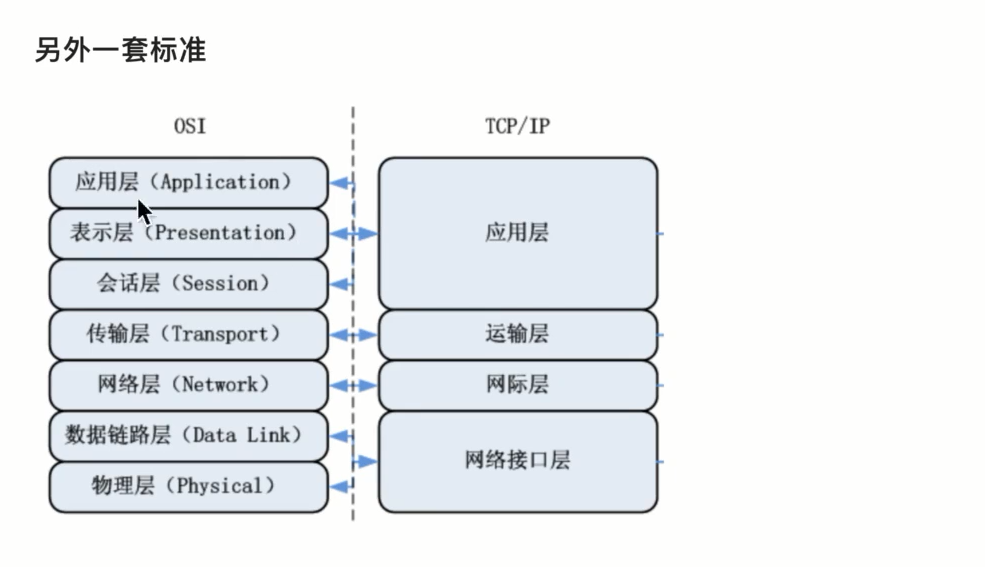
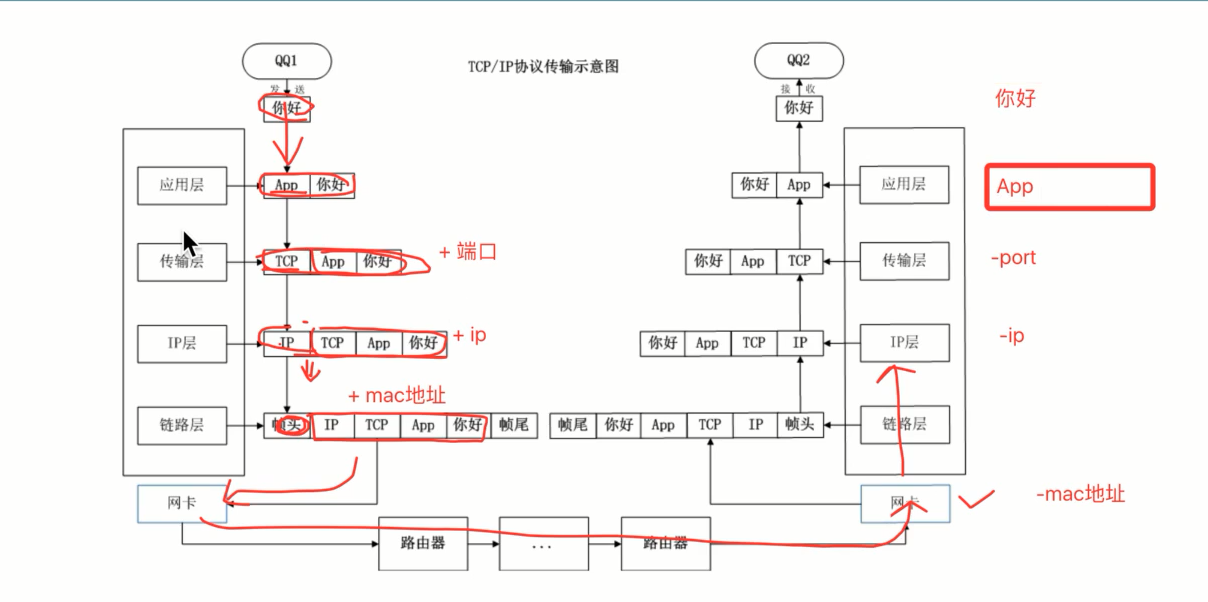


Tcp-IP协议：是一类协议的总称，不具体指某些协议

有众多协议的目的是：避免数据包丢失，保证数据包的正常传输



1. 应用层可以有自己的协议。应用层的数据格式为：应用层自身协议+数据
2. 应用程序想要能够进行网络通信，则必须实现tcp/udp协议
3. 然后加上网络层的协议
4. 通过网络接口，然后经过路由器协议，传输数据



OSI标准属于理论标准。实际使用的仍然是Tcp-Ip协议

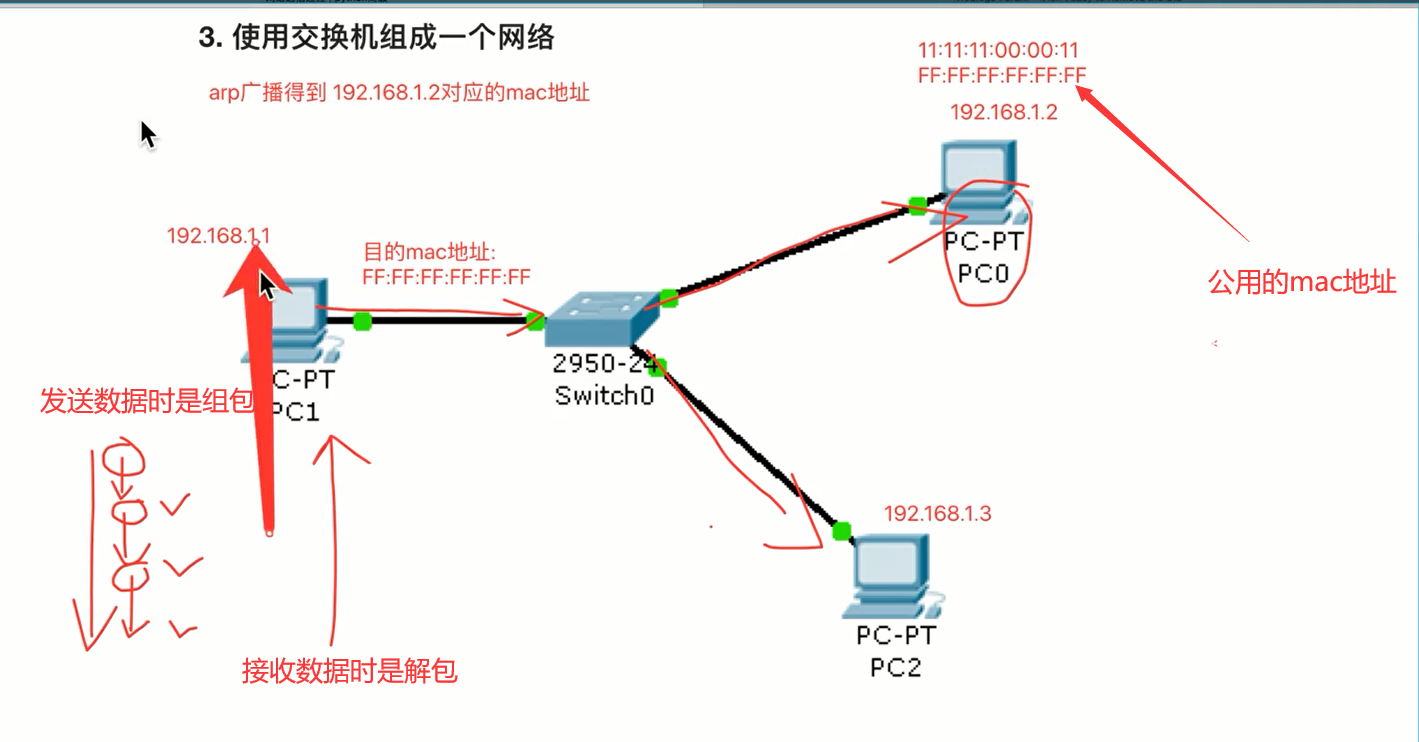
**通信原理**

两台电脑网络通信过程：需要知道ip 和子网掩码， 子网掩码用来确定是在哪个网段，同一个网段中的两个电脑才能通信

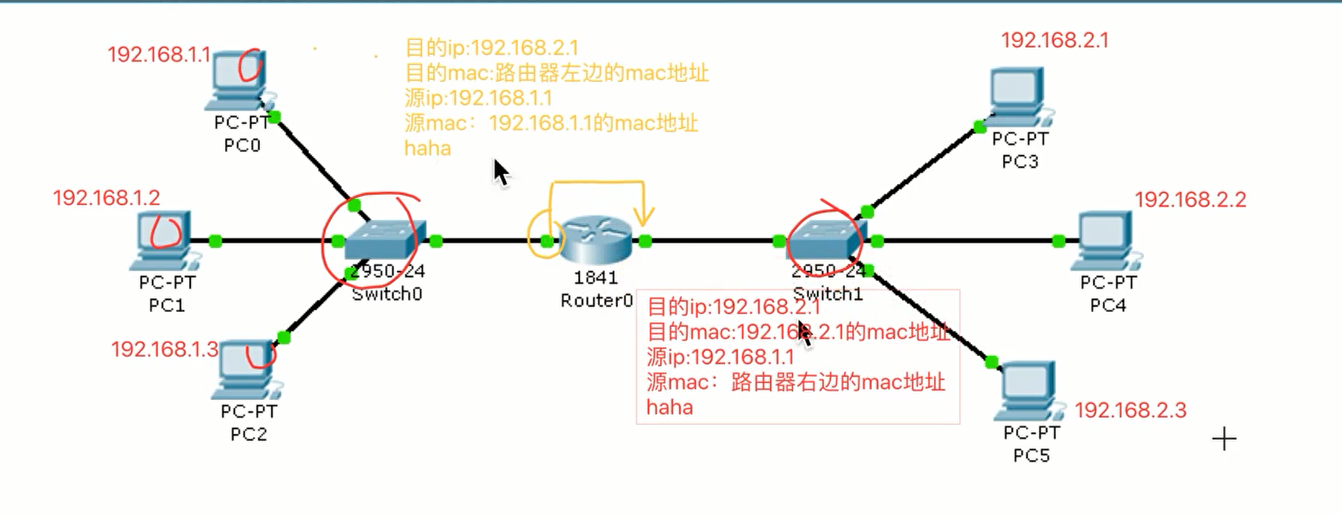
多台电脑通信过程：

第一代：通过集线器hub， 集线器的缺点是全部以广播的形式发出

第二代：交换机，可以广播也可以单播



第三代：路由器的作用，使得不同网络间的设备进行通信。具有转发数据到不同网络的能力的设备，叫做网关，一般是路由器



使用路由器发送数据时：ip不变才能确定数据最终到达的地方，mac地址一直在变，是因为mac地址只是确定收发双方的mac地址，发送数据时，可能需要通过多个网关，才能把数据发送到地方。

访问www.baidu.com 的过程：

