

- 网络中进程如何通信？
  - 首要解决的问题是如何标识一个进程，否则通信无从谈起！
  - 其实TCP、IP协议族已经帮我们解决了这个问题
  - 网络层的ip地址可以唯一标识网络中的主机
  - 传输层的“协议+端口”可以唯一标识主机中的应用程序（进程）
- 什么是套接字
  - socket（简称套接字）是进程间通信的一种方式，它与其他进程间通信的一个主要不同是：  
他能实现不同主机间的进程通信，我们网络上各种各样的服务大多是基于socket来完成通信的  
例如我们每天浏览网页、qq聊天、收发email等等
- 创建socket

```
1 import socket
2
3 # socket方法有两个参数
4 # 1、address family: 可以选择AF_INET（用于internet进程间通信）或者AF_UNIX(用于同一台机器进程间通信)
5 # 实际工作中常用AF_INET
6 # 2、Type: 套接字类型，可以是SOCK_STREAM（流式套接字，主要用于TCP协议）或者SOCK_DGRAM（数据报套接字，主要用于UDP协议）
7
8
9 # 创建udp套接字
10 s1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
11
12 # 创建tcp套接字
13 s2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- udp介绍
  - UDP --- 用户数据报协议，是一个无连接的简单的面向数据报的传输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快。
  - UDP是一种面向无连接的协议，每个数据报都是一个独立的信息，包括完整的源地地址或目的地址，它在网络上以任何可能的路径传往目的地，因此能否到达目的地，

到达目的地的时间以及内容的正确性都是不能被保证的。

特点：

UDP是面向无连接的通讯协议，UDP数据包括目的端口号和源端口号信息，由于通讯不需要连接，所以可以实现广播发送。UDP传输数据时有大小限制，每个被传输的数据报必须限定在64KB之内。UDP是一个不可靠的协议，发送方所发送的数据报并不一定以相同的次序到达接收方。

适用情况：

UDP是面向消息的协议，通信时不需要建立连接，数据的传输自然是不可靠的，UDP一般用于多点通信和实时的数据业务，比如

- 语音广播
- 视频
- qq
- TFTP（简单文件传送）
- SNMP（简单网络管理协议）
- RIP（路由信息协议，如报告股票市场，航空信息）
- DNS（域名解析）

注重速度流畅

udp操作简单，而且仅需要较少的监护，因此通常用于局域网高可靠性的分散系统中client/server应用程序。例如视频会议系统，并不要求音频视频数据绝对的正确，只要保证连贯性就可以了，这种情况下显然使用UDP会更合理一些。

- udp网络程序-发送数据
  - 创建套接字
  - 发送、接收数据
  - 关闭套接字

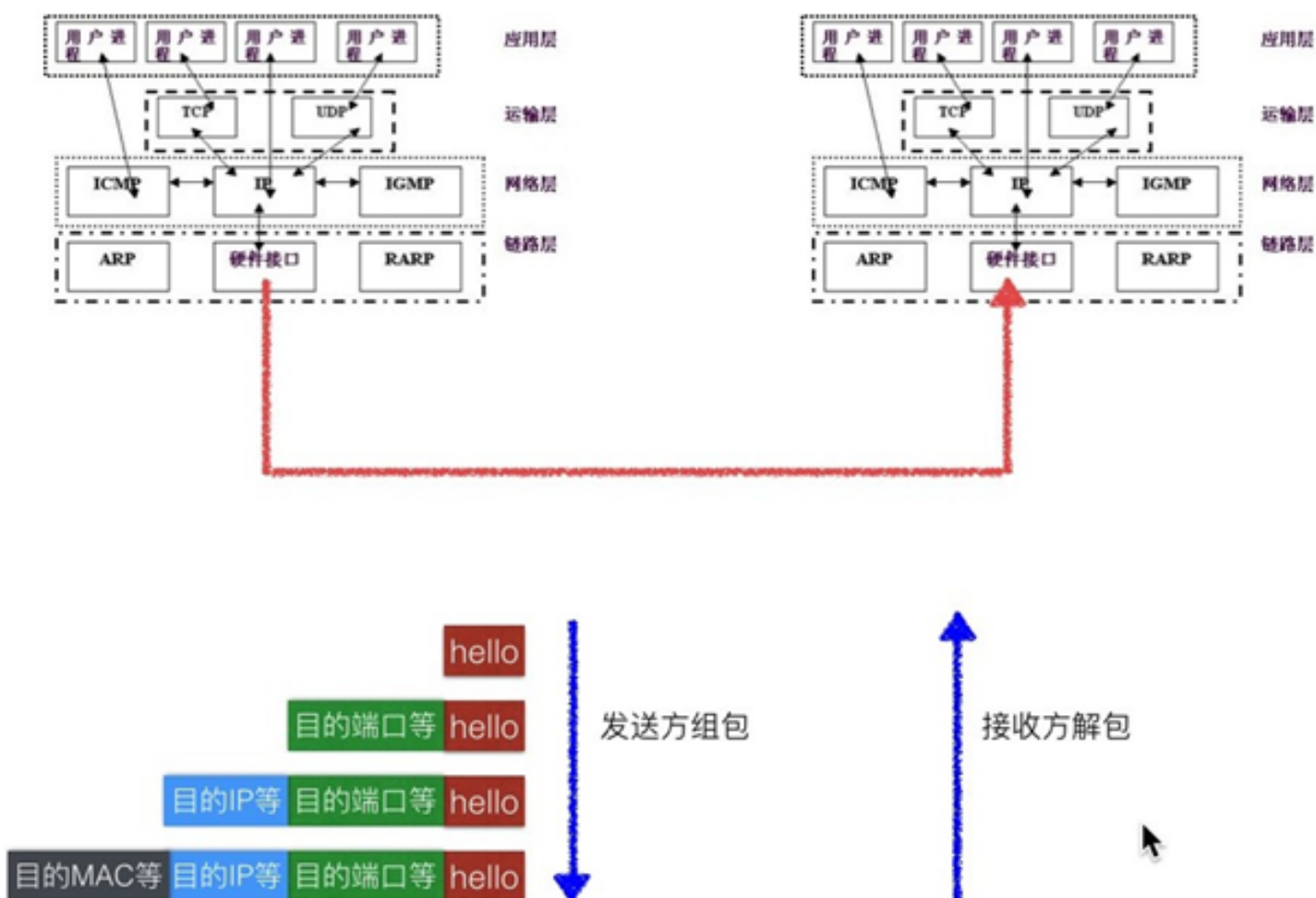
```
1 from socket import *
2
3 # 1、创建套接字udp
4 udp_socket = socket(AF_INET, SOCK_DGRAM)
5
6 # 2、准备接收方的地址信息
7 sendaddr = ("192.168.61.104", 9999)
8
9 # 3、输入信息
```

```

10 senddata = input("请输入你要发送的内容:").encode("gbk")
11
12 # 4、发送
13 # sendto有两个参数 1.发送的信息（经过编码）2.对方的地址（元组的类型）
14 udp_socket.sendto(senddata, sendaddr)
15
16 # 5、接收对方的数据
17 recv_data = udp_socket.recvfrom(1024) # 1024表示本次接收的最大字节数
18 # recv_data会接收到两个信息
19 # 1、对方发送的数据
20 # 2、对方的地址信息
21 print(recv_data)
22 print(recv_data[0].decode("gbk"))
23
24
25 # 关闭套接字
26 udp_socket.close()

```

## • udp网络通信的过程



## 发送手机方



## 购买手机方



- udp聊天器（能够循环接收数据）

```
1 from socket import *
2
3 # 1、创建套接字udp
4 udp_socket = socket(AF_INET, SOCK_DGRAM)
5
6 # 绑定本地相关信息
7 udp_socket.bind(("192.168.61.104", 9999))
8
9 while True:
10     # 等待对方发送数据
11     recv_data = udp_socket.recvfrom(1024) # 1024表示本次接收的最大字节数
12
13     # 打印接收到的信息
14     print(recv_data[0].decode("gbk"))
```

- 模拟qq
  - 任务要求
    - qq聊天程序

## ■ 和网络调试助手进行聊天

```
1 from socket import *
2
3 # 发送数据函数
4 def send_msg(socket):
5     # 输入发送的内容
6     dest_ip = input("请输入对方的ip: ")
7     dest_port = int(input("请输入对方的port: "))
8     dest_data = input("请输入发送的信息: ")
9     socket.sendto(dest_data.encode('gbk'), (dest_ip, dest_port))
10
11 # 接收数据函数
12 def recv_msg(socket):
13     # 接收
14     recv_data = socket.recvfrom(1024)
15     # 打印
16     print(recv_data[0].decode("gbk"))
17
18
19 # 主函数
20 def main():
21     #1、创建套接字
22     udp_socket = socket(AF_INET, SOCK_DGRAM)
23     # 2、绑定ip和port
24     udp_socket.bind(("", 8899))
25     # 循环收发数据
26     while True:
27         print("===udp聊天器===")
28         print("0: 退出")
29         print("1: 发送")
30         print("2: 接收")
31         op = input("请输入你要的功能: ")
32         if op == "0":
33             break
34         elif op == "1":
35             send_msg(udp_socket)
36         elif op == "2":
37             recv_msg(udp_socket)
```

```
38         else:
39             print("输入有误")
40
41
42 if __name__ == '__main__':
43     main()
```