一、用 python 完成一个 tcp 小实验：

1. 向 client 输入一个电话号码并向 server 发送，server 经过调用 api 查询后，向 client 返回该电话号码的信息。

Server 的代码

```
def query_phone_info(p_number):
    url = "https://tcc.taobao.com/cc/json/mobile_tel_segment.htm?tel="
    req = urllib.request.Request(url+p_number)
    data = urllib.request.urlopen(req).read()
    js = bytes.decode(data, 'GBK')
    start_index = js.find('{')
    return js[start_index:]


if __name__ == '__main__':
    server_port = 8888
    server_socket = socket(AF_INET, SOCK_STREAM)
    server_socket.bind(('', server_port))
    # 最大连接数为1
    server_socket.listen(1)
    print('等待连接中...')
    while True:
        connect_socket, addr = server_socket.accept()
        print(datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'), end=': ')
        print('与%s建立连接成功' % addr[0])
        phone_number = connect_socket.recv(1024).decode()
        json_data = query_phone_info(phone_number)
        connect_socket.send(json_data.encode())
```

Client 代码

```
from socket import *

if __name__ == '__main__':
    server_name, server_port = 'localhost', 8888
    client_socket = socket(AF_INET, SOCK_STREAM)
    client_socket.connect((server_name, server_port))
    print('连接成功！')
    phone_number = ''
    while phone_number != "q":
        phone_number = input('请输入手机号码：(输入q退出)')
        client_socket.send(phone_number.encode())
        query_result = client_socket.recv(1024).decode()
        print('你的号码信息是:', query_result)
    # 退出循环时关闭连接
    client_socket.close()
```

运行：

```
"C:\Program Files (x86)\python3.6\python.exe" E:/desktop/tcp/server.py
等待连接中...
2019-03-15 17:38:00: 与127.0.0.1建立连接成功
```

```
"C:\Program Files (x86)\python3.6\python.exe" E:/desktop/tcp/client.py
连接成功!
请输入手机号码: (输入q退出)13886050564
你的号码信息是: {
    mts:'1388605',
    province:'湖北',
    catName:'中国移动',
    telString:'13886050564',
    areaVid:'30513',
    ispVid:'3236139',
    carrier:'湖北移动'
}

请输入手机号码: (输入q退出)
```

二、完成 5 道课后习题：

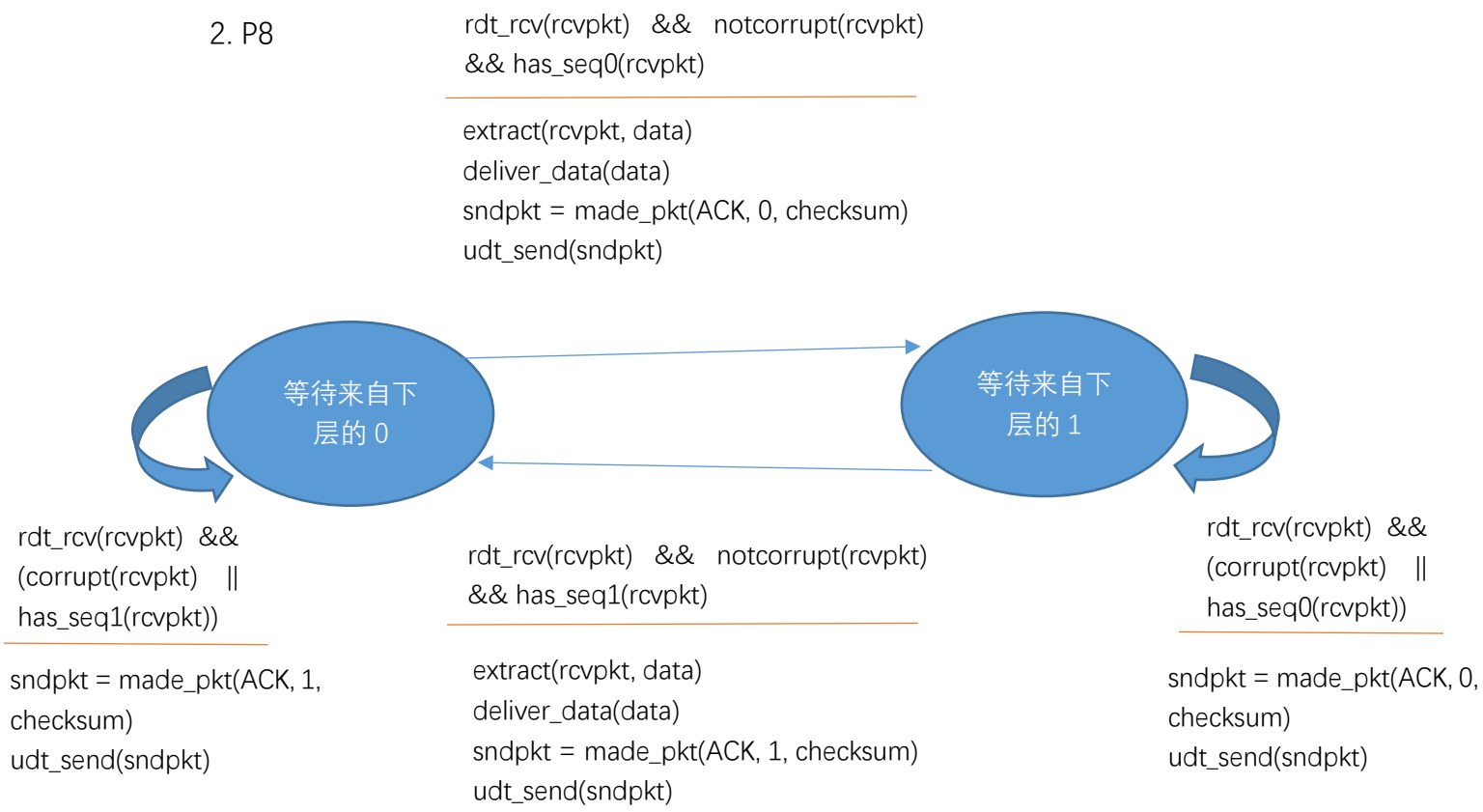
1. P4

a. 01011100 的 01100101 的和为 11000001，其反码为 00111110。

b. 11011010 和 01100101 的和为 01000000,，其反码为 10111111。

c. 只需要两个字节同一位不同即可。比如第一个字节为 01011101，第二个字节为 01100100；或者第一个字节为 01010100，第二个字节为 01101101。

2. P8



3. P15

发送一个分组计入链路的时间 $T_{trans} = \frac{L}{R} = \frac{1500*8}{10^9} = 12\mu s/pkt$

利用率$U_{sender} = \frac{T_{trans}*N}{T_{trans}+RTT} = \frac{0.012*N}{30.012} \geq 0.90$

计算得 N≥ 2251

4. P22

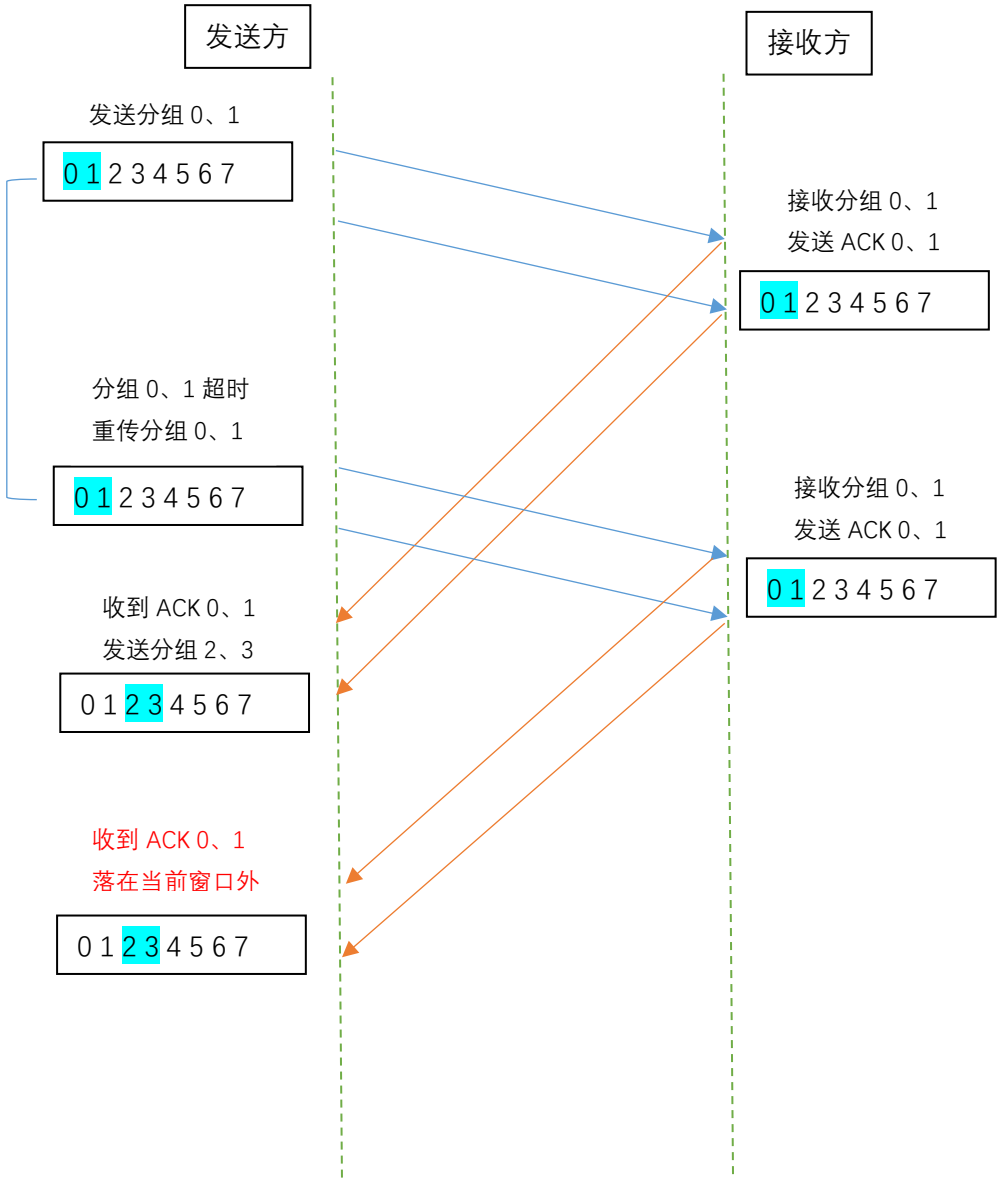
a. 如果发送方收到了之前所有的 ACK，则窗口内的报文序号是[k, k+3]；

如果发送方未收到之前的 ACK，则窗口内的序号为[k-4, k-1]。

b. 如果接收方在等待分组 k，则它已经接受到了序号 k-1 的分组；则序号区间[k-4, k-1]的 ACK 要么被发送者接收到了，或者正在传输中。

5. P24

a. 对于 SR 协议，发送方有可能会收到落在其当前窗口之外的分组的 ACK。 假设这里窗口大小为 2。



b. 对于 GBN 协议，发送方也可能会收到落在其当前窗口之外的分组的 ACK。原理同上。

c. 当发送方、接收方窗口大小都为 1 时，发送方在等待确认之前只能发送一个报文，和停等协议是一样的。故此时比特交替协议与 SR 协议相同。

d. 当发送方、接收方窗口大小都为 1 时，比特交替协议与 GBN 协议相同。原理同上。