1. 名词解释
2. DNS：域名系统，用于主机名到IP地址的转换
3. URL：统一资源定位符，标识了万维网上的各种文档，全网范围唯一。
4. 流量控制：控制发送节点发送数据的速率，使接收方来得及接收。
5. 拥塞控制：当网络发生拥塞时，通过相关算法调节发送方数据发送速率而使网络恢复到正常状态，是网络来得及转发。
6. 协议栈 (protocol stack)：各层所有协议的集合
7. COOKIE：为了辨别用户，进行SESSION跟踪而储存在用户本地终端的数据。
8. OSI：开放系统互连，是计算机广域网的体系结构的国际标准，共分为7层。
9. 套接字：应用程序进程与传输层协议间的门户和接口。
10. 基带信号：信源发出的没有经过调制的的原始电信号。
11. 基带传输：将基带信号直接通过有线传输。
12. 宽带传输：将数字信号调制成特定频段的模拟信号传输。
13. MIME：多用途因特网邮件扩展，是SMTP的一个扩展，用于非ASCII码传输
14. 吞吐量（throughput）：在发送端和接收端之间传送比特的速率
15. WEB缓存：代替起始服务器来满足HTTP请求的网络实体，保存最近请求过的对象副本。
16. 起始服务器：对象最初存放并始终持有其副本的服务器。
17. CRC：循环冗余检测，实现双方约定好生成多项式，发送节点在发送数据后附加冗余码，使得整个数据可以整除生成多项式，接收节点接到后，若仍能整除，则认为数据正确，否则认为数据出错。
18. 多路复用：在一条传输链路上建立多个连接，分别传输数据。
19. FDM：频分多路复用，将链路传输频段划分为若干小的频段，每个连接专用不同频段进行传输。
20. TDM：时分多路复用，将链路的传输时间划分为若干时隙，每个连接轮流专用不同的时隙进行传输。
21. 电路交换（circuit switching）:通信节点间采用面向连接方式，，预留带宽等资源，建立专用电路进行传输。
22. 分组交换（packet switching）：通信节点间无连接，分组使用目的IP地址经中间若干路由器转发，相同源和目的地的分组可能采用不同的路径。
23. 全双工通信：通信节点可在同一时刻进行数据的收发。
24. CSMA：载波侦听多路访问，一种随机访问技术，在发送帧前先监听信道，若空闲则立即发送，否则等待一段随机时间后发送。
25. 多播：组播，一对多通信。
26. 调制：将数字信号转换为模拟信号。
27. 解调：将模拟信号转换为数字信号。
28. ADSL：非对称数字用户线，采用不对称的上行与下行传输速率，常用于用户宽带接入。
29. CDMA：码分多路复用，各站点采用不同的编码码片，混合发送，接受方使用对应码片等正确提取所需信息。
30. 计算机网络：使用通信设备和线路将不同地点的具有独立功能的计算机系统连接起来，并按照网络协议进行数据通信，实现资源共享的计算机集合。
31. ARQ：自动重传请求协议，基于重传机制的可靠数据传输协议。
32. LAN：局域网，是一个地理范围较小的计算机网络。
33. ATM：异步传输模式，建立在电路交换和分组交换的基础上，是一种面向连接的快速分组交换技术。
34. Torrent：洪流，参与一个特定文件转发的所有对等方的集合。
35. 网络协议（Internet protocol）：互相通信的对等方实体间遵守的约定和规则，包括语法，语义，定时。
36. Modem：调制解调器，在发送方可进行调制，将数字信号转换为模拟信号，在接受方可进行解调，将模拟信号转换为数字信号。
37. 默认路由器：与主机直接相连的第一个路由器。
38. SAP：服务访问点，是OSI相邻两层的逻辑接口。
39. PDU：协议数据单元，对等层间传输的数据单位
40. PPP：点对点传输协议
41. Proxy server:代理服务器
42. GBN:回退N步流水线协议，允许发送方连续发送多个分组而无需等待确认，若出错则从出错分组开始重发，接收方按序接受分组，若正确接收则发确认ACK，若出错则将出错分组及后面分组丢弃，不做任何应答。
43. Hierarchical Routing:层次化路由技术，按AS形式组织路由器。同一个AS内的路由器运行相同的选路算法。
44. RIP选路信息协议：基于距离向量的路由协议
45. OSPF开放最短路径优先：采用Dijkstra最短费用路径算法，是一种链路状态协议。
46. 转发表：路由设备内，从入端口到出端口间建立起来的对应表，主要用来转发数据帧或IP分组。
47. 毒性逆转技术：DV算法中，解决计数到无穷问题的技术，即告知从相邻路由获得最短路径信息的相邻路由器到目的网络的距离为无穷大。
48. 加权公平排队（WFQ）：排队策略根据权值大小不同，将超出队列的数据包丢弃。
49. 自治系统AS：由一组通常在相同管理者控制下的路由器组成，在相同AS中，路由器运行相同的选路算法。
50. Tunnel:隧道，在链路层或网络层通过对等协议建立起来的逻辑通信通道。
51. MAC地址：网卡或网络设备端口的物理地址
52. Ad Hoc网络：自主网络，无基站
53. 停等协议：发送方发送完一个分组后停止，等待对方回答
54. 分组阻塞：来自其他输入端口的分组当前正在使用交换结构。

54.拒绝服务攻击 Denial of Service (DoS): 通过构造大量

假的通信占用资源（如服务器和网络带宽），使正常用户得不到服务

55.AIMD：处理拥塞控制的一个算法，加法增，乘法减。

56.CIDR：无类别域间路由，基于可变长子网掩码来进行任意长度的前缀分配。

57.DHCP：动态主机配置协议，基于UDP的应用层协议，可使主机自动获取IP地址。

58.openflow：一种网络通信协议，属于数据链路层，运行于SDN控制器与交换机之间。

59.归属网络：移动节点的永久居所

60.归属代理：归属网络中执行移动管理的实体

61.永久地址：移动节点在归属网络中的地址

62.外部地址: 移动节点在被访网络的地址

63.被访网络: 移动节点当前访问的网络

64外部代理: 被访网络移动管理的实体

65.间接路由: 通信者使用移动节点的永久地址通信，需归属代理转发

66.直接路由: 通信者先获取移动节点的外部地址，然后直接通信

填空

1. 以太网MAC地址的长度为6个字节
2. 对载波波形进行调制，可以使用调频，调幅，调相
3. 三种主要差错检测技术：奇偶校验，检验和，CRC
4. 四种时延：节点处理时延，传输时延，传播时延，排队时延
5. 识别主机进程：通过IP地址和端口号
6. 各层常用协议：

* 应用层：SMTP,HTTP,FTP,POP3,IMAP(因特网邮件访问协议),DNS,Telnet（远程终端协议）
* 传输层：TCP,UDP
* 网络层：IP，ICMP（网络控制报文协议）,ARP（dizhijieyi协议）,RARP（反向地址转换协议）,IGMP（网组管理协议）
* 链路层：以太网，802.3，PPP，帧中继，ATM

1. 常见面向连接协议：HTTP，FTP，TCP
2. 自治系统内选路协议：RIP,OSPF
3. 自治系统间选路协议：BGP
4. IEEE802.3以太网采用的多路访问协议是CSMA/CD

无线局域网采用多路访问协议是CSMA/CA

1. 协议三要素：语法，语义，定时
2. 网络应用程序体系结构主要有客户机/服务器模式(e.g. WEB，FTP,Telnet)，纯P2P模式(e.g.因特网电话，文件共享，迅雷)及混合模式三种。
3. 路由器组成：输入端口，输出端口，交换结构，选路处理器
4. 分组交换网络包括：虚电路网络和数据报网络
5. 采用1000BaseT的以太网，其传输速率是1000Mbps
6. 双绞线网线连接设备有两种方法，其中计算机与计算机见直接连接采用的是交叉线。双绞线绞合的目的是减少干扰
7. 无线网络的组成主要包括：无线主机，基站，无线链路，网络基础设施
8. 交换机交换的两种方式：存储转发方式，直通交换方式
9. P2P内容组织有三种：集中式目录，查询洪泛，层次覆盖
10. 宽带接入方式中，用户可以同时上网和拨打电话采用的是ADSL技术
11. CSMA/CD有冲突时传输一个48比特的拥塞信号，作用是：强化冲突信号，确保其他出传输适配器均检测到冲突
12. GBN协议中，若序号字段为K位，则发送和接受窗口的最大值分别为-1和1
13. SR协议中，若序号字段为K位，则发送和接受窗口的最大值分别为和
14. 通信子网包括物理层，链路层，网络层；资源子网包括应用层，表示层，会话层；传输层处于两者之间
15. IEEE802.11无线LAN属于单跳基于基础（看是否需要基站）设施网络

自组织802.11无线LAN和蓝牙网：单跳，无基础设施

无线传感网络：多跳，基于基础设施

Ad Hoc网络：多跳无基础设施

26.GBN协议中用到的技术：序号、累积确认、检验和以及超时/重传

1. 重要简答题
2. 浏览器邮件投递过程

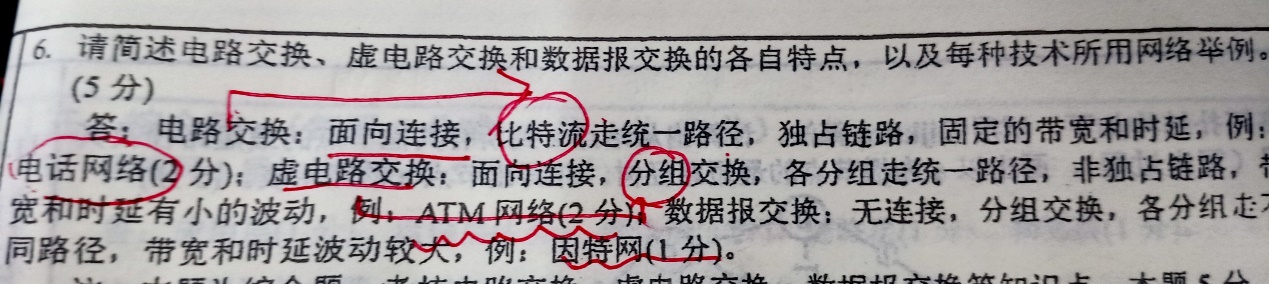
第一阶段：邮件从发送方代理传送到发送方邮件服务器，使用HTTP

第二阶段：邮件从发送方邮件服务器传送到接收方邮件服务器，使用SMTP

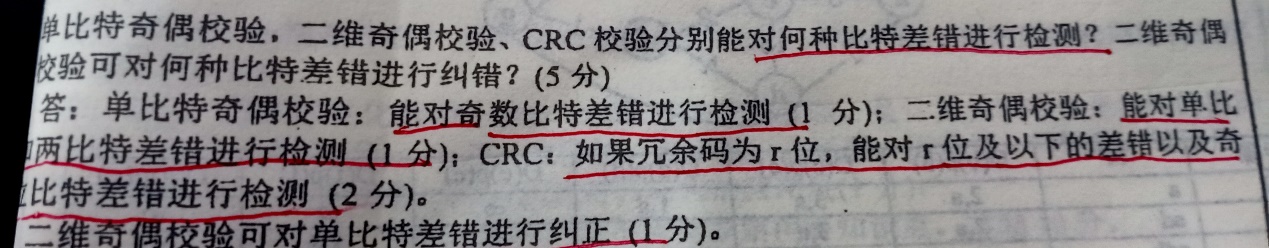
第三阶段：从接收方邮件服务器取回并读取邮件，使用SHTTP

各阶段均使用DNS协议

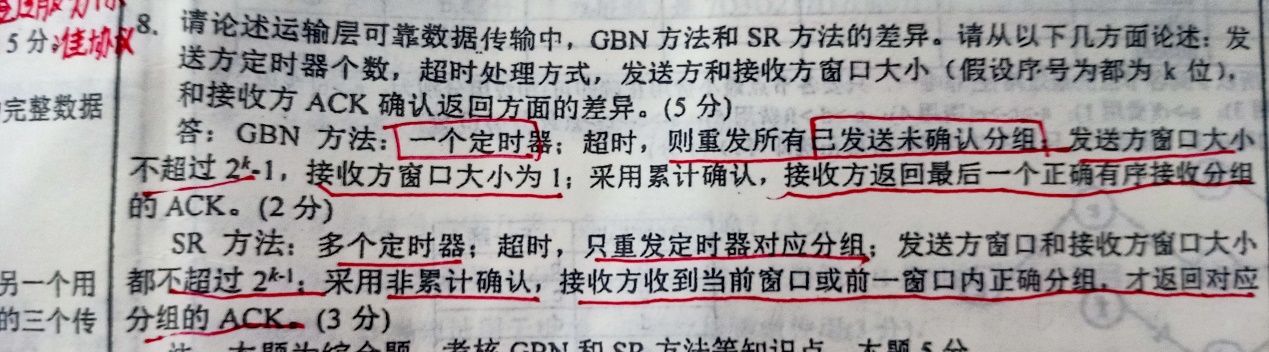
1. 简述电路交换，虚电路交换，数据报交换各自特点



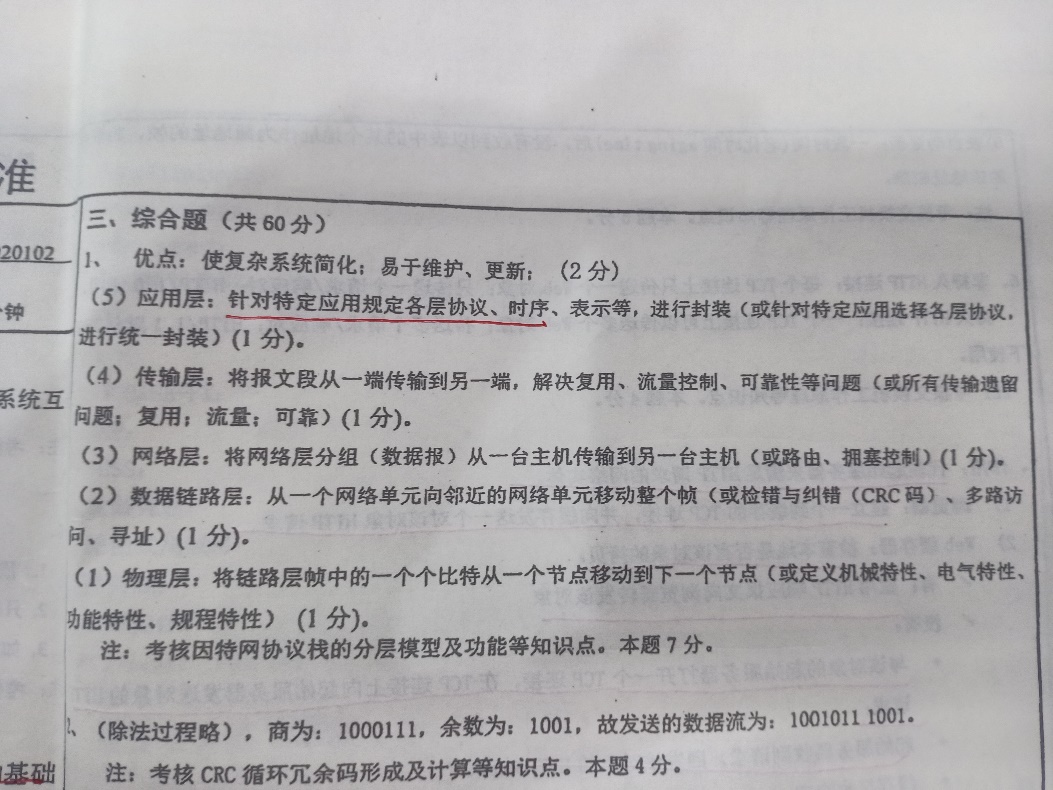
1. 单比特，二维奇偶校验，CRC各自特点



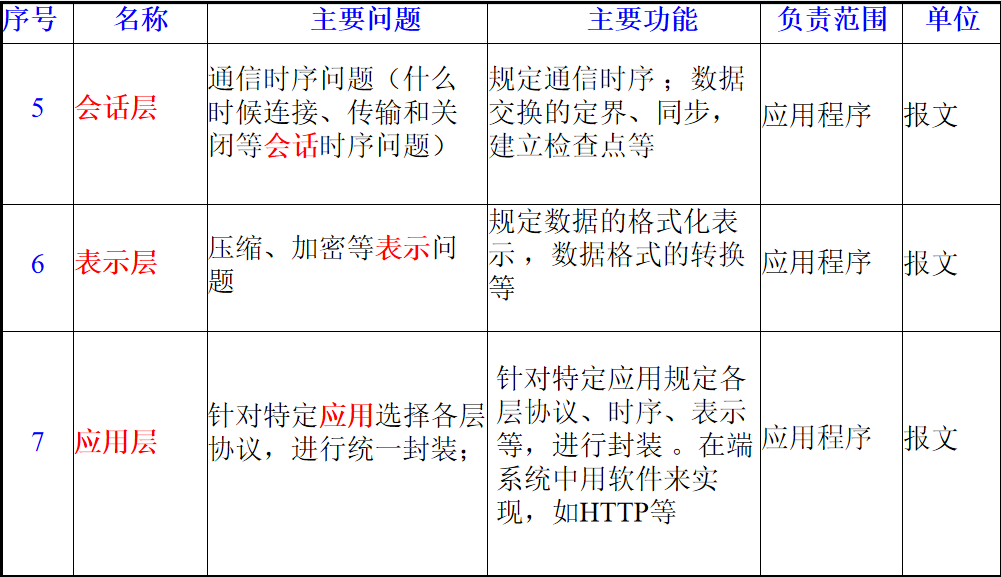
1. 简述GBN,SR异同



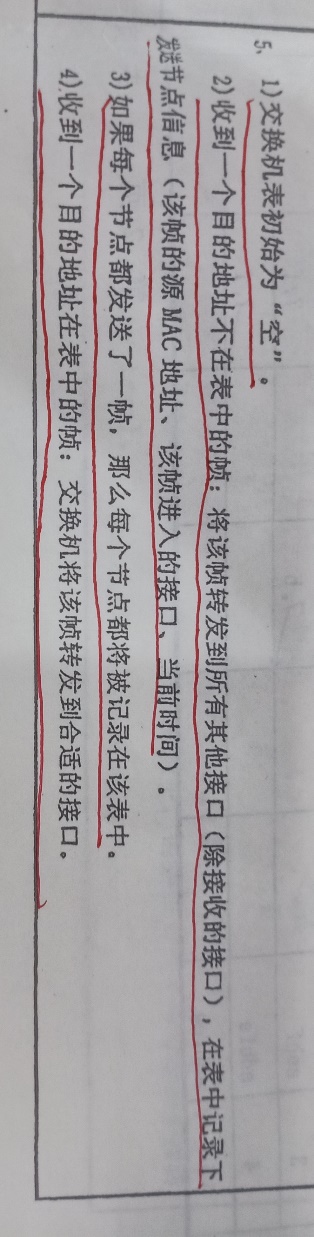
1. IP地址分类及详解
2. 简述计算机网络分层优点及因特网协议栈分层模型中各层功能。

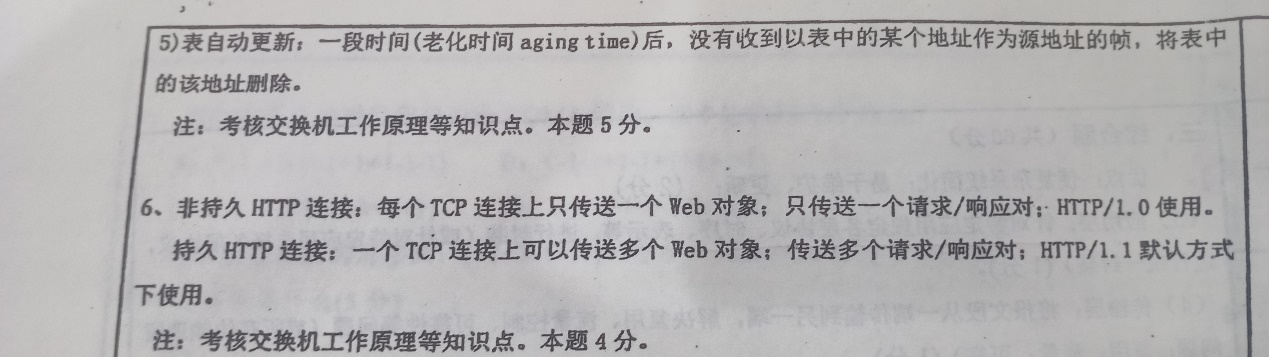




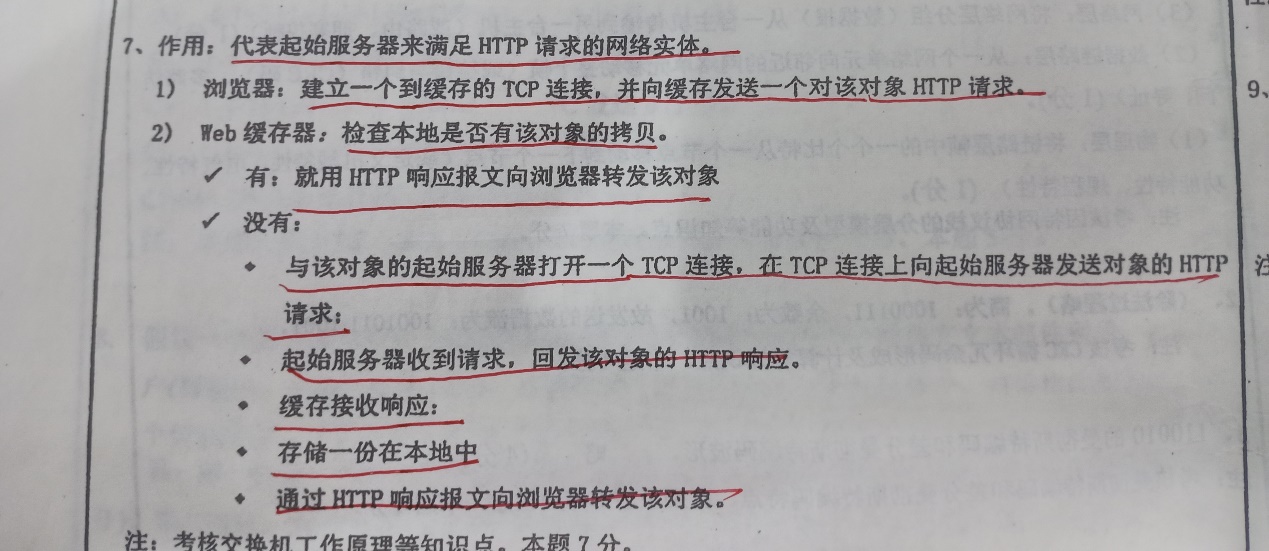


1. 简述交换机逆向扩散式路径学习法基本原理

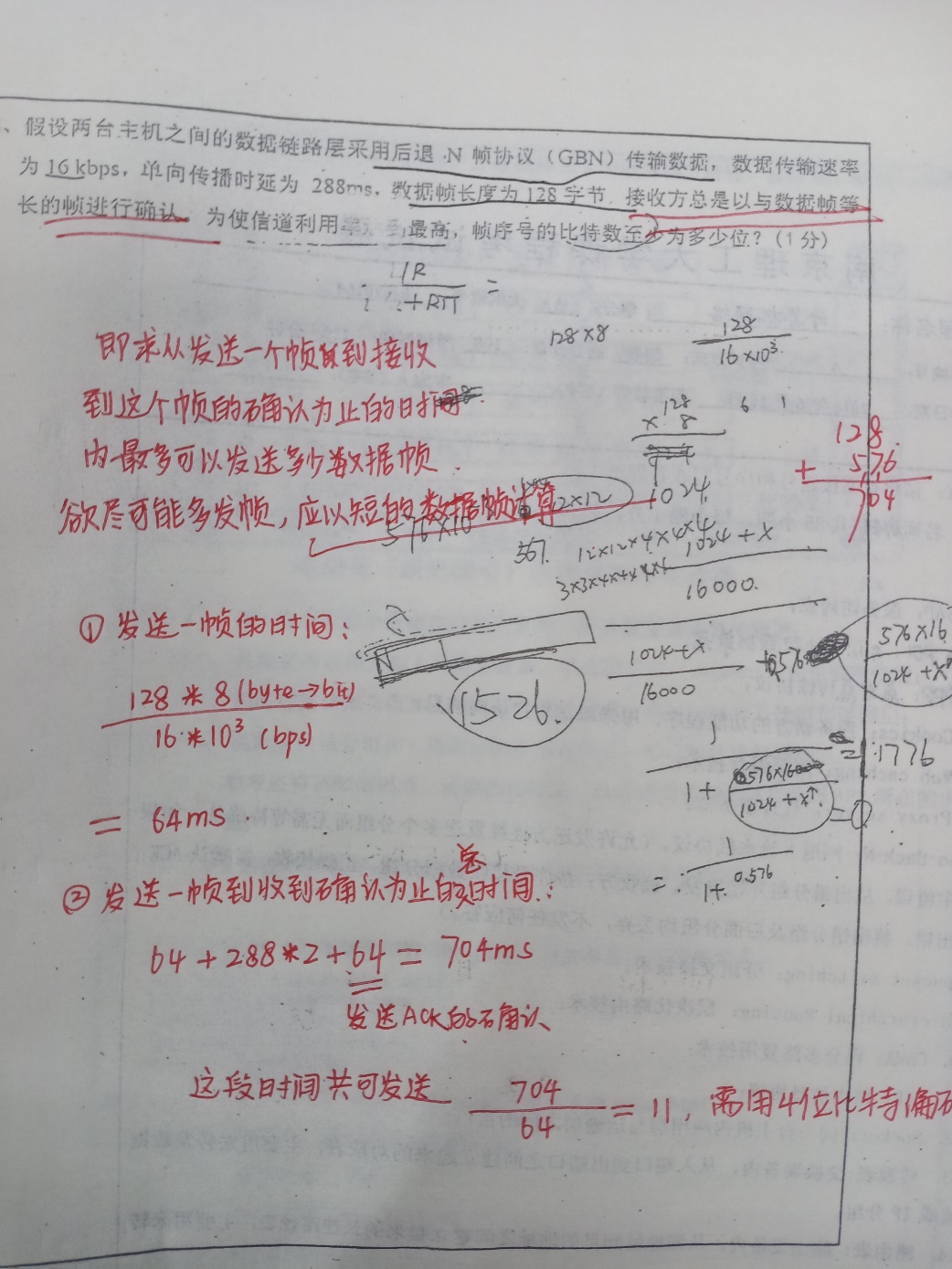




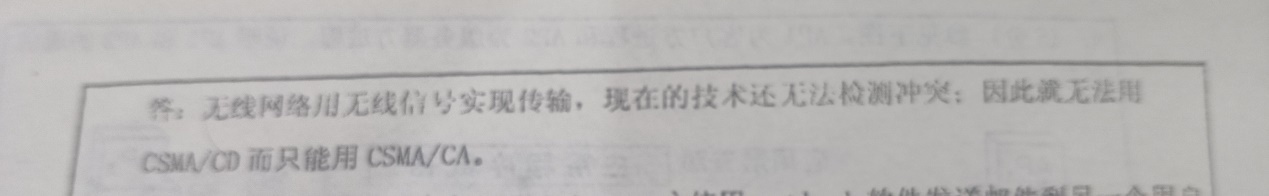
1. 简述HTTP非持久连接和持久连接的不同
2. 简述WEB缓存的作用和功能



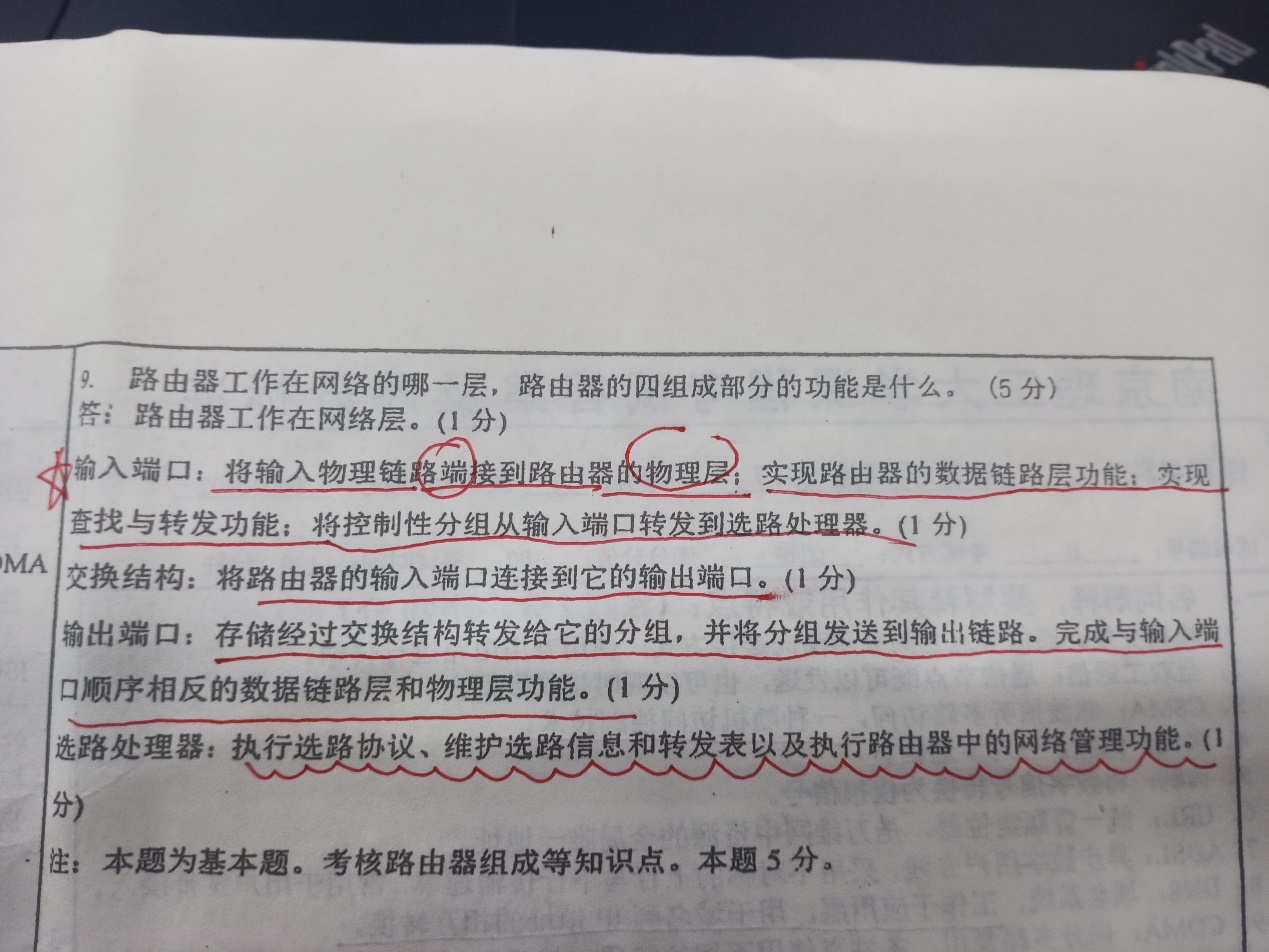
1. 计算题



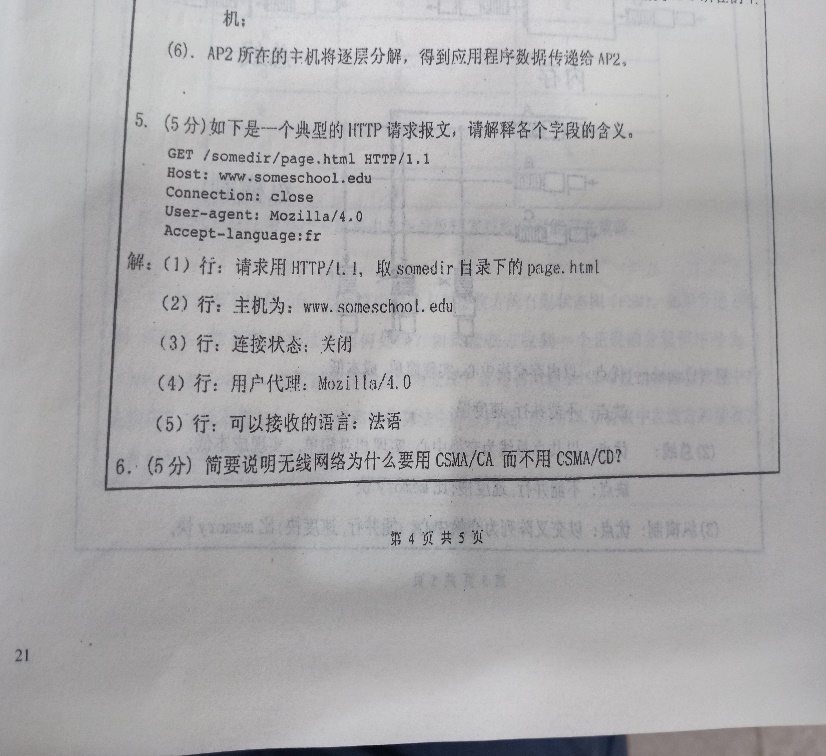
1. 为什么无线网不使用CSMA/CD？



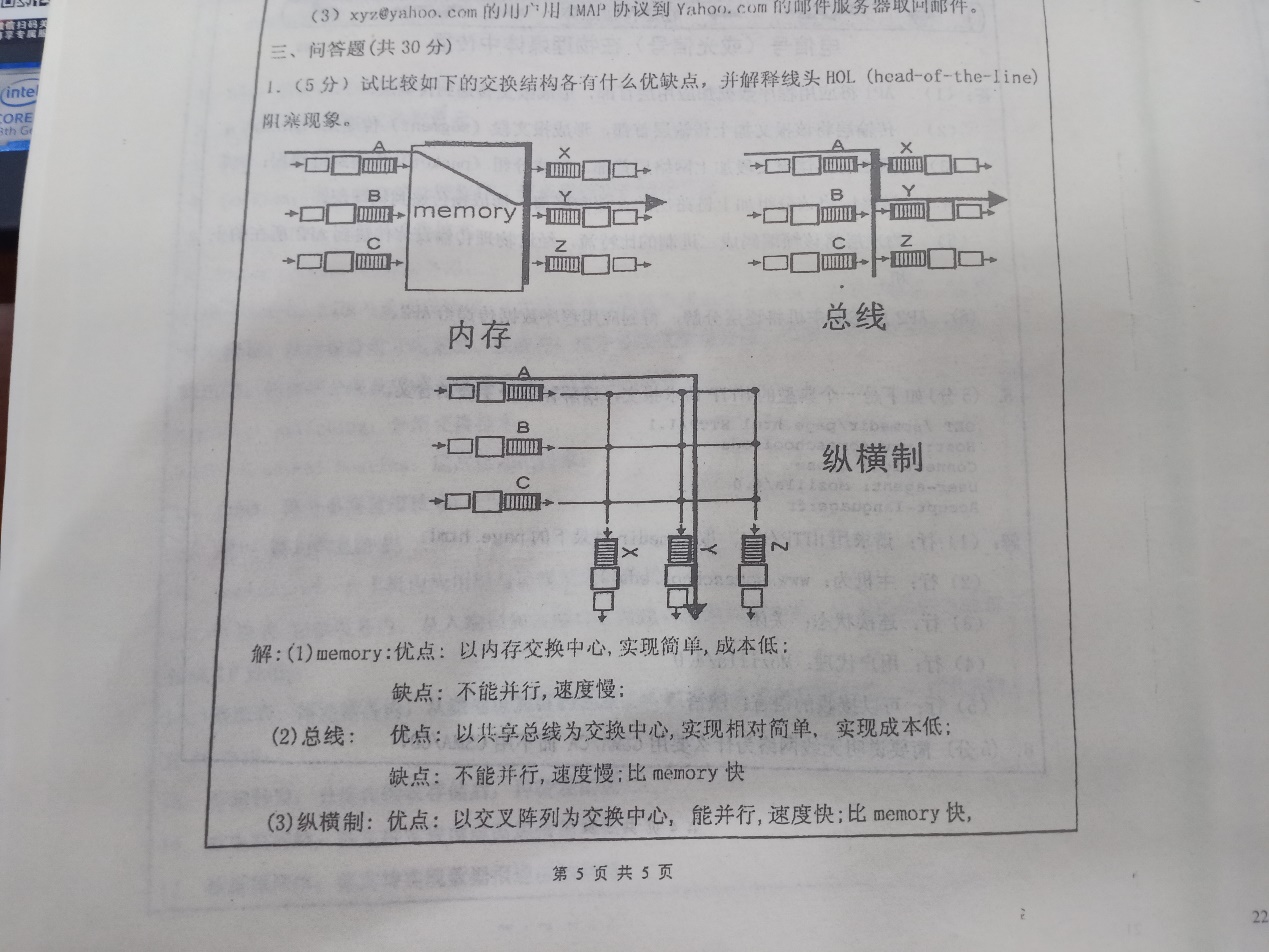
1. 路由器四个结构的功能：

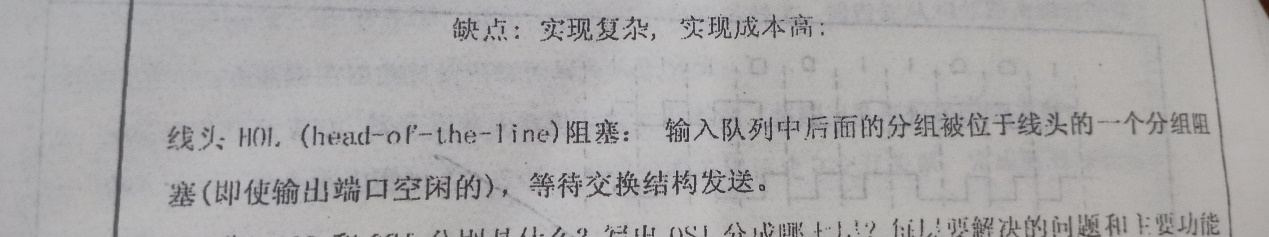


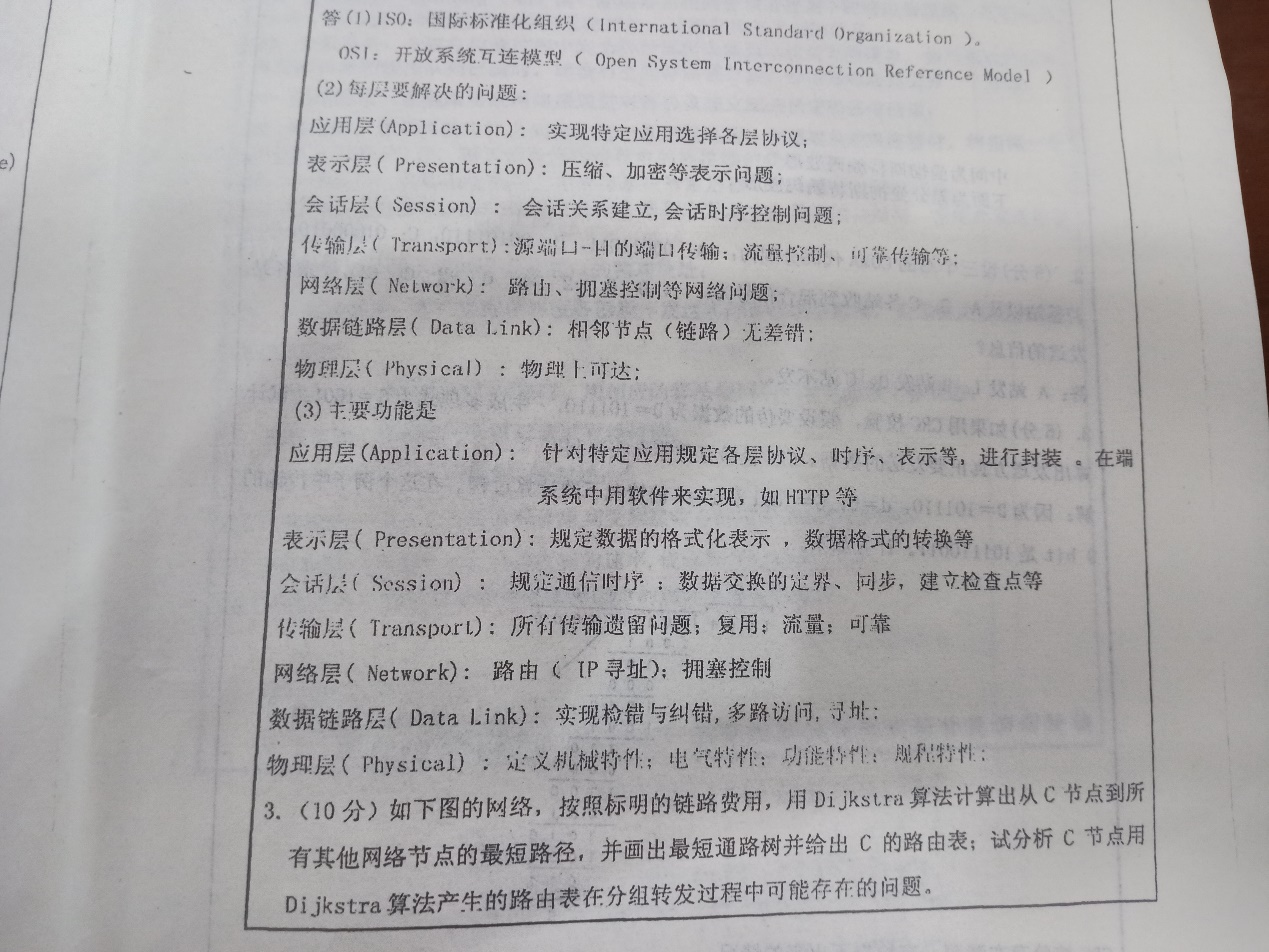
13.



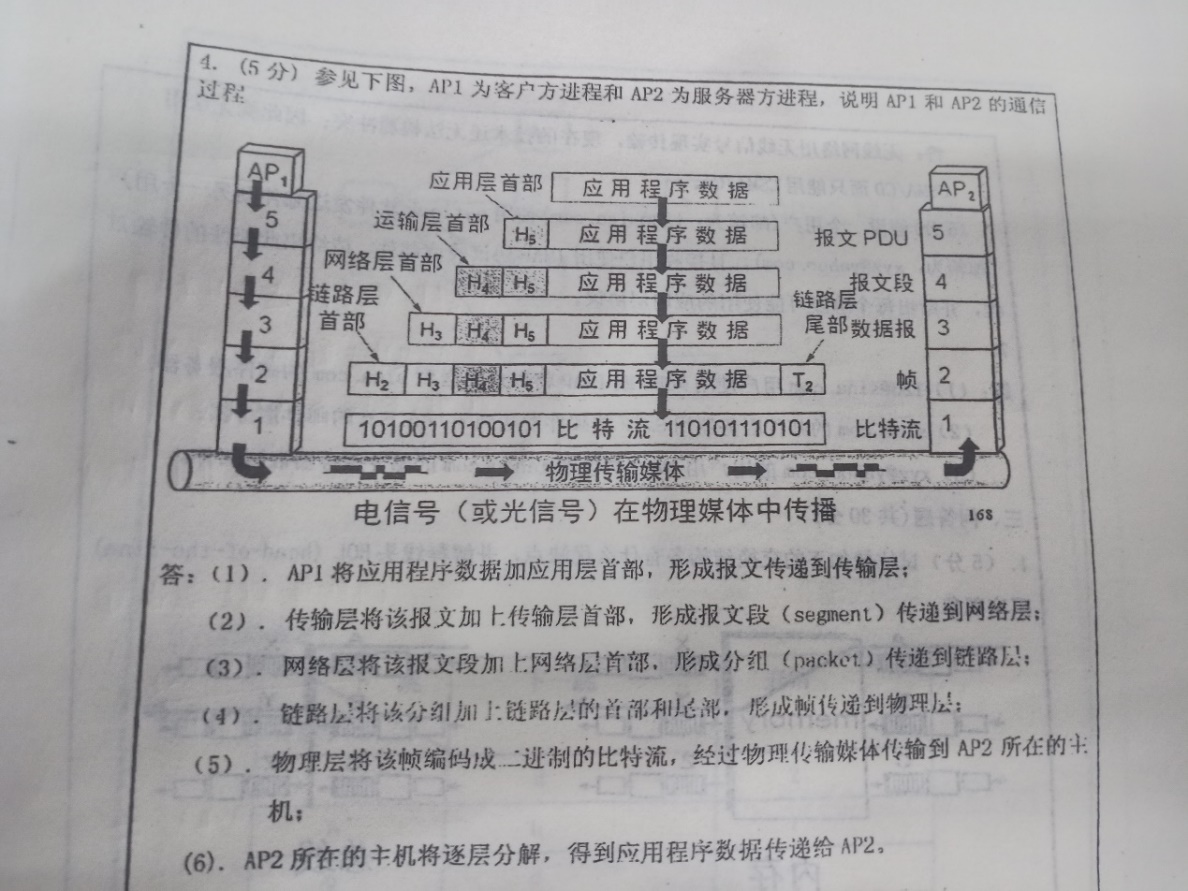
14.





15. 

16.



17.

SOCKET sock,sock1;

sockaddr\_in ServerAddr;

sockaddr\_in ClientAddr;

int Addrlen;

DWORD StartSock()

{

WSADATA WSAData;

if (WSAStartup(MAKEWORD(2,2),&WSAData)!=0) //初始化套接字

{

printf("sock init fail !\n");

return(-1);

}

return(1);

}

DWORD CreateSocket()

{

sock=socket(AF\_INET,SOCK\_STREAM,0); //创建套接字

if (sock==SOCKET\_ERROR)

{

printf("sock create fail !\n");

WSACleanup();

return(-1);

}

ServerAddr.sin\_family=AF\_INET; //填充服务器地址

ServerAddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

ServerAddr.sin\_port=htons(RECV\_PORT);

if (bind( sock, (struct sockaddr FAR \*) &ServerAddr, sizeof(ServerAddr)) == SOCKET\_ERROR) {//绑定套接字为创建的sock指定通信对象

printf("bind is the error");

return(-1);

}

return (1);

}

DWORD ConnectProcess()

{

char buff[80];

int i;

Addrlen=sizeof(sockaddr\_in);

if (listen( sock, 5 ) < 0)

{

printf("Listen error");

return(-1);

}

printf("Listening...\n");

for (;;)

{

sock1 = accept( sock,(struct sockaddr FAR \*) &ClientAddr,&Addrlen);

printf("connect ok \n");

printf("wait for receive... \n");

for ( ;; )

{

memset(buff,0,80);

if (recv(sock1,buff,80,0)<=0)

{

break;

}

printf(buff);

printf("\n");

// Sleep(100);

}

}

}

SOCKET sock;

sockaddr\_in ServerAddr;

DWORD CreateSocket()

{

sock=socket(AF\_INET,SOCK\_STREAM,0); //创建套接字

if (sock==SOCKET\_ERROR)

{

printf("sock create fail !\n");

WSACleanup();

return(-1);

}

return (1);

}

DWORD CallServer()

{

CreateSocket();

if (connect( sock, (struct sockaddr\*) &ServerAddr,

sizeof( ServerAddr))==SOCKET\_ERROR)

{

printf("Connect fail \n");

closesocket( sock );

return(-1);

}

return(1);

}

DWORD TCPSend(char data[])

{

int length;

length=send(sock,data,strlen(data),0);

if (length<=0)

{

printf("send data error !\n");

closesocket(sock);

WSACleanup();

return(-1);

}

return(1);

}

DWORD StartSock()

{

WSADATA WSAData;

if (WSAStartup(MAKEWORD(2,2),&WSAData)!=0) //初始化套接字

{

printf("sock init fail !\n");

return(-1);

}

ServerAddr.sin\_family=AF\_INET; //填充服务器地址

ServerAddr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

ServerAddr.sin\_port=htons(RECV\_PORT);

return(1);

}

int main()

{

char buff[80];

int num,i;

StartSock();

while (CallServer()==-1);

printf("connect ok!\n");

printf("press Enter to send !");

getchar();

for (;;)

{

printf("Input the number of message to send:(0-exit)");

memset(buff,0,80);

scanf("%d",&num) ;

if (num<=0)

break;

for (i=0;i<num;i++)

{

sprintf(buff,"data%d",i);

printf(buff);

printf("\n");

TCPSend(buff);

// Sleep(100);

}

}

return(0);

}