1. Multiple Choice

(1).D

CSMA/CD：链路层协议

ICMP:网络层协议

ICMP是（Internet Control Message Protocol）Internet控制报文协议。它是TCP/IP协议族的一个子协议，用于在IP主机、路由器之间传递控制消息。

OSPF: 链路层协议

(Open Shortest Path First开放式最短路径优先）是一个内部网关协议(Interior Gateway Protocol，简称IGP），用于在单一自治系统（autonomous system,AS）内决策路由。是对链路状态路由协议的一种实现，隶属内部网关协议（IGP），故运作于自治系统内部。与RIP相比，OSPF是链路状态协议，而RIP是距离矢量协议。

IMAP:应用层协议

（Internet Mail Access Protocol）它与POP3协议的主要区别是用户可以不用把所有的邮件全部下载，可以通过客户端直接对服务器上的邮件进行操作。

（2）A

5-Application ： message

4-Transport ： segment

3-Network ： datagram

2-Link ： frame

（3）A

（4）D

（5）B

（6）B

（7）D

（8）B

解析：求广播地址，需要将非网络位置为1，202.115.32.0写成2进制为0010 0000 0000 0000 前23位不可变，将后9位置为1，0010 0001 1111 1111.转换为十进制即得带广播地址。

（9）D Internet是数据报网络

（10）C

DHCP发现报文源IP地址为默认地址0.0.0.0

（11）A

（12）A

HOL（Head-Of-the-Line）Blocking，线路前部阻塞，即在一个输入队列中排队的分组必须等待通过交换结构发送（即使输出端口是空闲的），因为它由位于线路前部的另一个分组阻塞。

（13）B

（14）D

（15）A

子网掩码用1表示网络地址位，中间不能有0，而A答案中32写成2进制为0010 0000，因此不合法。

（16）A 前26位不能改变 64写成2进制为0100 0000，将末位置为1记得到A答案。

（17）C 邮件服务器之间通信采用的是SMTP（simple mail transport protocol）协议

（18）B 链路层提供的服务有：成帧（framing） 链路接入（link access）

可靠交付（reliable delivery） 流量控制（flow control） 差错检测（error detection）

差错纠正（error correction）半双工和全双工（half-duplex and full-duplex）

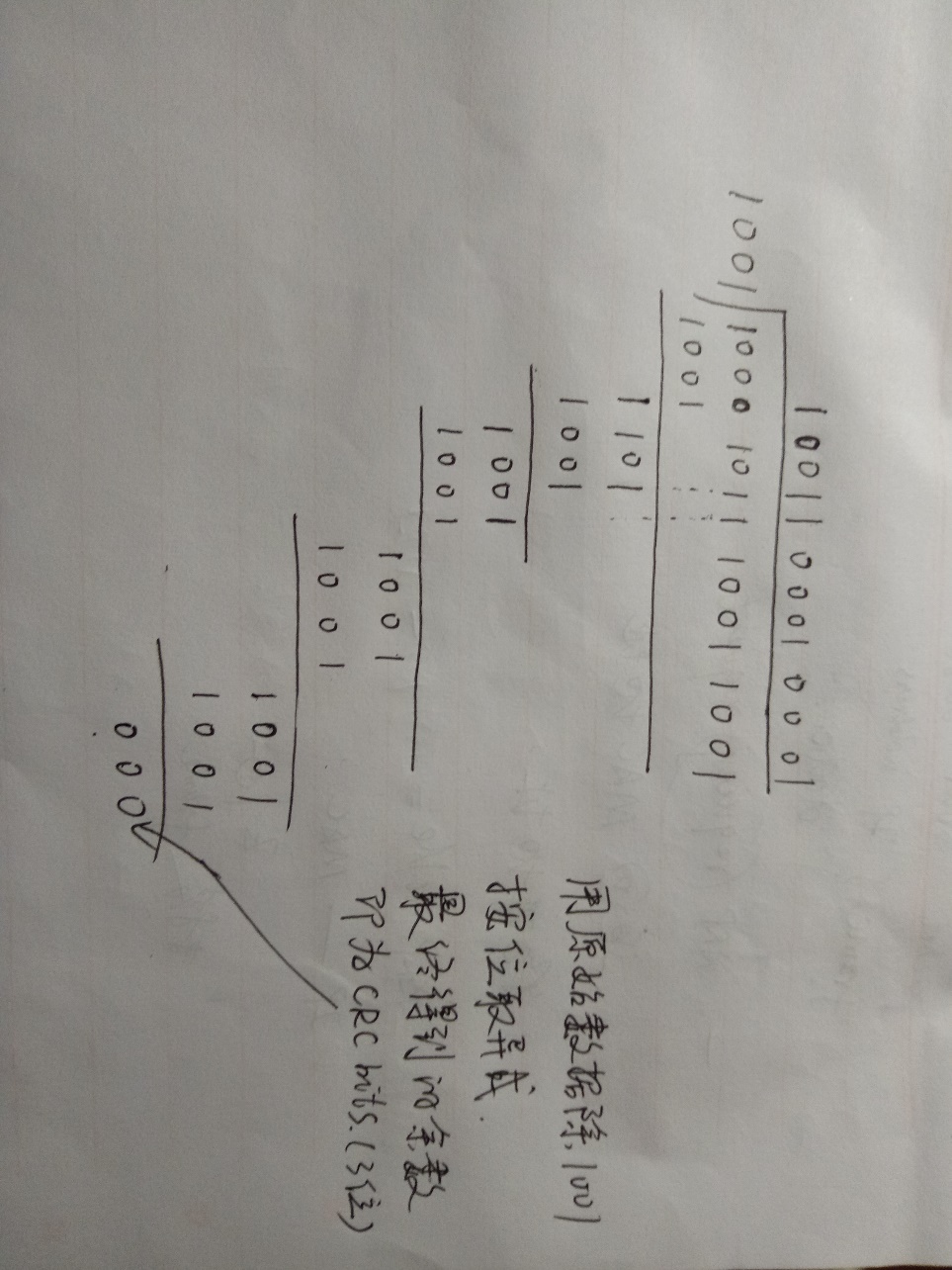
（19）C

检验和计算 ，将溢出位加到最后即为0010 0100 + 1 = 0010 0101

最后将计算结果取反记得到检验和1101 1010

（20）B

CRC计算步骤



1. True or False
2. T
3. F

IP报文的分片和重组都由端系统完成

1. F

ADSL属于DSL技术的一种，全称Asymmetric Digital Subscriber Line（ 非对称数字用户线路）。 采用频分多路复用（Frequency-Division Multiplexing，FDM）技术把普通的电话线分成了电话、上行和下行三个相对独立的信道，从而避免了相互之间的干扰。

1. F 非持久连接 传输每一个对象都要建立一个TCP连接
2. F

开放最短路径优先（Open Shortest Path First）。OSPF的核心就是一个使用洪泛链路状态信息的链路状态协议和一个Dijkstra最低费用路径算法。

1. F

跨越两个AS的BGP会话被称为外部BGP（eBGP）会话，同一个AS中的两台路由器之间的BGP会话被称为内部BGP（iBGP）会话。而网关路由器要获取其相邻网关路由器的信息需要跨越一个AS，因此为外部BGP会话。

1. F 电路交换网络应对突发数据传输时，性能要比分组交换弱很多。
2. F 在IP层只对IP首部进行了检验，而TCP/UDP检验和才是对整个报文段进行检验的。
3. T

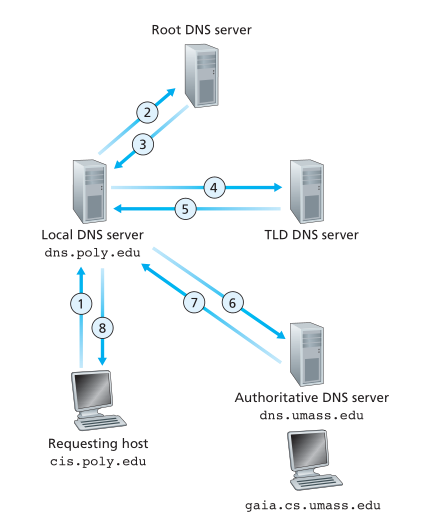
因特网、ATM CBR和ATM ABR服务模型比较

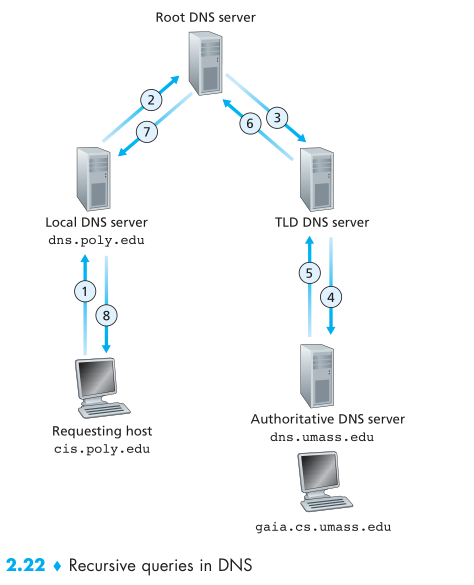
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 网络体系结构 | 服务模型 | 带宽保证 | 无丢失保证 | 排序 | 定时 | 拥塞指示 |
| Internet | Best-effort | 无 | 无 | 任何可能的顺序 | 不维持 | 无 |
| ATM | CBR | 保证恒定速率 | 是 | 有序 | 维持 | 不出现拥塞 |
| ATM | ABR | 保证最小速率 | 无 | 有序 | 不维持 | 提供拥塞提示 |

1. F

迭代（iteration）查询和递归（recursive）查询比较

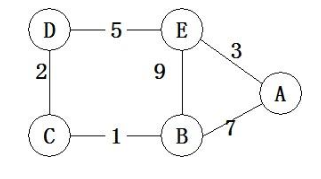
**迭代查询**：DNS 服务器另外一种查询方式为迭代查询，DNS 服务器会向客户机提供其他能够解析查询请求的DNS 服务器地址，当客户机发送查询请求时，DNS 服务器并不直接回复查询结果，而是告诉客户机另一台DNS 服务器地址，客户机再向这台DNS 服务器提交请求，依次循环直到返回查询的结果为止。



递归查询：递归查询是一种DNS 服务器的查询模式，在该模式下DNS 服务器接收到客户机请求，必须使用一个准确的查询结果回复客户机。如果DNS 服务器本地没有存储查询DNS 信息，那么该服务器会询问其他服务器，并将返回的查询结果提交给客户机。

3.(这两道题不怎么会，会的宝宝帮我完善一下吧)

（1）



1. 结点E的距离向量表（不是很清楚distance entries是啥意思，暂且认为是E的距离向量表吧，但是6分似乎也太容易了一点=\_=!）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| E | 3 | 9 | ∞ | 5 | 0 |

1. 收敛后的距离向量表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 7 | 8 | 8 | 3 |
| B | 7 | 0 | 1 | 3 | 8 |
| C | 8 | 1 | 0 | 2 | 7 |
| D | 8 | 3 | 2 | 0 | 5 |
| E | 3 | 8 | 7 | 5 | 0 |

**C到D的距离变为10后**

5

10

1

9

7

3

其距离向量表为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| C | 8 | 1 | 0 | 2 | 7 |

这里使用了毒性逆转算法，因此C将通告B，它到D的距离为无穷大，即

所以C发送给B的距离向量表为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| C | 8 | 1 | 0 | ∞ | 7 |



（2）

a) NAT 因为192.168.1.0是私有地址

b) 因为E,F具有相同的网络号

c)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Seq | Source MAC | Destination MAC | Source IP | Destination IP |
| 1 | E’s MAC | FF-FF-FF-FF-FF-FF |  |  |
| 2 | R2’s Interface3 | E’s MAC |  |  |
| 3 | E’s MAC | R2’s Interface3 | E’s IP | A’s IP |
| 4 | R2’s Interface1 | FF-FF-FF-FF-FF-FF |  |  |
| 5 | A’s MAC | R2’s Interface1 |  |  |
| 6 | R2’s Interface1 | A’s MAC | E’s IP | A’s IP |
| 7 |  |  |  |  |

d)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Host | Source MAC | Destination MAC | Source IP | Destination IP |
| C | C’s MAC | R2’s interface2 | C’s IP | A’s IP |
| A | R2’s interface1 | A’s MAC | R2’s interface1 | A’s IP |

4.Analytic

初始状态： RTT=60ms MSS=1KB Timeout = 100ms threshold=4KB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event | Time(ms) |  | Time(ms) | Event |
| 发送报文0 | 0 | 0 | 0 | **接受窗口为4KB** |
| 初始发送窗口大小为1KB | 10 |  | 10 |
| 20 |  | 20 |  |
| 30 |  | 30 | 接收0，确认0 |
| 40 |  | 40 |  |
| 50 |  | 50 |  |
| 确认0，发送1  1RTT窗口变为2KB | 60 | 1 | 60 |  |
| 发送2 | 70 |  | 70 |  |
| 窗口已满，等待确认1 | 80 | 2 | 80 |  |
| 90 |  | 90 | 接收1，确认1 |
| 100 |  | 100 | 接收2，确认2 |
| 110 |  | 110 |  |
| 确认1（窗口=4）,发送3 | 120 |  | 120 |  |
| 确认2 ,发送3 | 130 | 3 | 130 |  |
| 确认1，发送4 | 140 | 4 | 140 |  |
| 发送5 | 150 | 5 | 150 |  |
| 发送6 | 160 |  | 160 | 报文3（4th）丢失 |
| 窗口满，等待报文3 | 170 | 6 | 170 | 接收4，确认4 |
| 发送窗口\*+1=5KB | 180 | 7 | 180 | 接收5，确认5 |
| 发送7 | 190 |  | 190 | 接收6，确认6 |
|  | 200 |  | 200 |  |
|  | 210 |  | 210 |  |
|  | 220 |  | 220 | 接收7，确认7  接收窗口设为2KB |
| 3超时重传，窗口=1KB | 230 | 3 | 230 |
| 窗口满，等待确认3 | 240 |  | 240 |
| 250 |  | 250 |  |
| 260 |  | 260 | 接收3 |
| 270 |  | 270 |  |
| 280 |  | 280 |  |
| 发送8（窗口=2） | 290 | 8 | 290 |  |
| 发送9 | 300 | 9 | 300 |  |
| 窗口满，等待确认8 |  |  | 310 |  |
| 320 |  | 320 | 接收8，确认8 |
| 330 |  | 330 | 接收9，确认9 |
|  | 340 |  |  |  |
| 确认8，发送10 | 350 |  |  |  |
| 确认9，发送11 | 360 |  |  |  |

**最终结果为**

|  |  |  |
| --- | --- | --- |
| **Sending time** | **Seq** | **Ack** |
| **0** | **0** | **100** |
| **60** | **1** | **100** |
| **70** | **2** | **100** |
| **130** | **3** | **100** |
| **140** | **4** | **100** |
| **150** | **5** | **100** |
| **160** | **6** | **100** |
| **190** | **7** | **100** |
| **230** | **3** | **100** |
| **290** | **8** | **100** |
| **300** | **9** | **100** |
| **350** | **10** | **100** |
| **360** | **11** | **100** |