**软件学院本科生2022——2023学年第1学期计算机网络课程期末考试试卷答案（B卷）**

1. **选择（共10题，每题2分）**

1 - 5 D D D B C

6 - 10 B D C B B

**二、填空（共14题，每空1分）**

1. 广域网、城域网、局域网、个人区域网

2. 网络接口层、互联网层、传输层、应用层  
3. 路由、转发

4. 寻址和分片

5. ICMP差错报告报文、ICMP询问报文

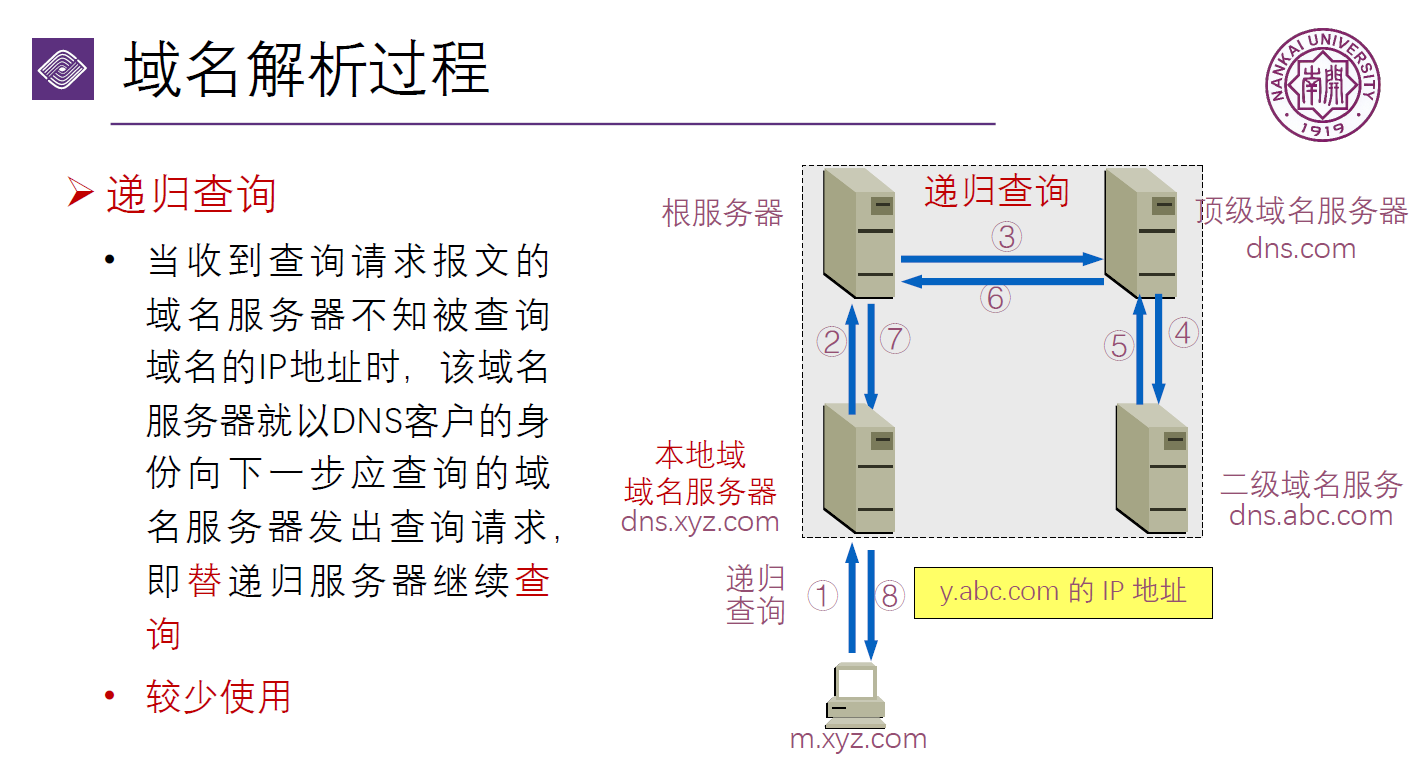
6. 64、1518

7. NAT

8. 动态主机配置协议

1. DNS、ARP
2. TCP协议、UDP协议
3. IP地址、端口号
4. C/S方式、B/S方式、P2P方式
5. org域名服务器、com域名服务器、edu域名服务器

**三、简答题（共10分）**

****

**四、计算题（共15分）**

1.数据包分片（7分）

（1）固定首部长度是20字节；片偏移指的是当数据长度MTU>最大传输数据长度时，无法一次传输完成，三个为划分片长(1500-20)\*2/8。MF标志指的是此数据报后是否还有数据报，没有为0，有就1； 所以 1400-20=1380字节1380/64=1344字节 4000/1344=3片（5分）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据字段长度（首部和数据部分） | MF位 | DF位 | 片偏移 |
| 原始数据报 | 4000 | 0 | 0 | 0（0/8） |
| 数据报片1 | 1500 | 1 | 0 | 0（0/8） |
| 数据报片2 | 1500 | 1 | 0 | 185（1480/8） |
| 数据报片3 | 1040 | 0 | 0 | 370（1480\*2/8） |

（2）各数据报片的数据字段长度1500 1500 1040（5分）  
（3）各数据报片的片偏移字段值0 185 370（5分）

2.时延（8分）

（1）传播时延dprop = 信道长度(m) / 电磁波在信道上的传播速率(m/s) = M / V（2分）

（2）传输时延dtrans = 数据帧长度(b) / 信道带宽(b/s) = L / R（2分）

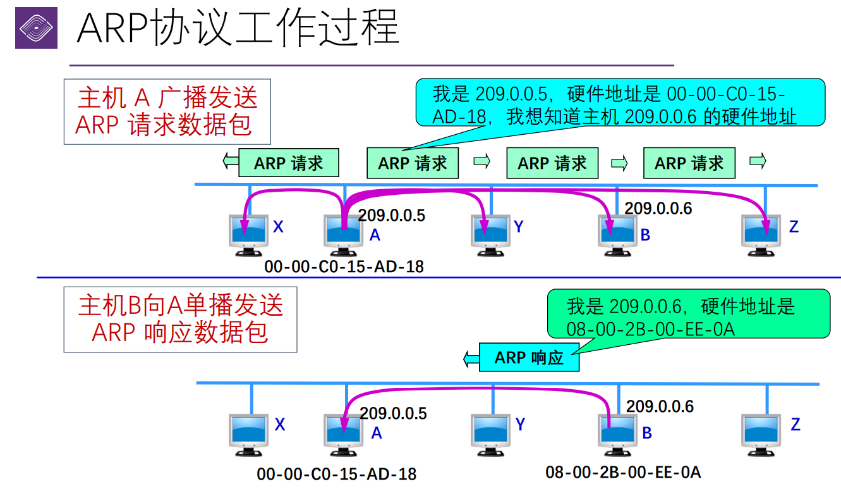
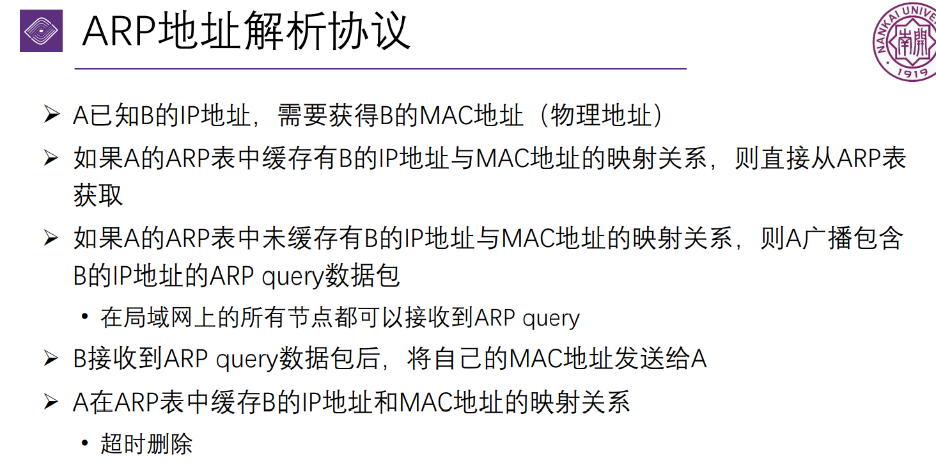
（3）总时延dtotal = 传播时延 + 传输延迟 = M / V + L / R（2分）

（4）时延带宽积 = 传播时延 \* 带宽 = M / V \* R = 512，解得M = 1280米（2分）

**五、综合题（共10分）**

（1）ARP协议的功能是将主机的IP地址解析为相应的MAC地址。（2分）

（2）当主机A要向主机B发送数据之前，必须解析出主机B的MAC地址，解析过程如下：（共5分）

****

①主机A先查看自己的ARP缓存表，如果有主机B的IP地址与MAC地址的映射关系，则直接从ARP表获取。（1分）

②如果主机A的ARP表中未缓存有B的IP地址与MAC地址的映射关系，则广播发送ARP请求数据包到以太网，该ARP报文大致意思是：“我的IP地址是209.0.0.5，我想知道主机209.0.0.6的MAC地址。” （1分）

③这个广播帧会传到以太网上的所有机器，每个机器在收到该广播帧后，都会去查看自己的IP地址。（1分）

④但是只有IP地址为209.0.0.6的主机B会向主机A单播发送ARP响应数据包，该ARP报文**大**致意思是：“我的IP地址是209.0.0.6，我的MAC地址是08-00-2B-00-EE-OA。” （1分）

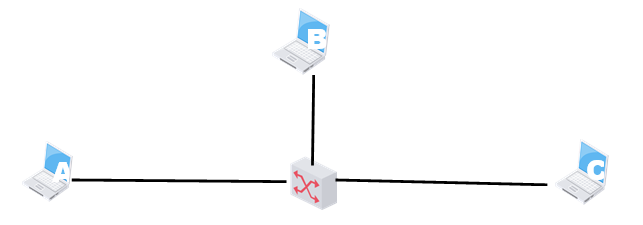
⑤这样主机A就知道了IP地址为209.0.0.6的主机所对应的MAC地址为08-00-2B-00-EE-OA，随后就可以向主机B发送数据。（1分）

（3）（共3分）

B回应ARP响应报文（单播）：目的MAC地址为主机A的MAC地址00-00-C0-15-AD-18

我的IP地址是209.0.0.6，

我的MAC地址是08-00-2B-00-EE-OA



A发送ARP请求报文（广播）：目的MAC地址为FF-FF-FF-FF-FF-FF

我的IP地址是209.0.0.5

我的MAC地址为00-00-C0-15-AD-18

我想知道IP地址209.0.0.6对应的MAC地址？

**六、Socket编程题（共15分）**

服务器端：（8分）

public class Server

{ // 服务器端类

private ServerSocket ss; // 声明ServerSocket对象

private Socket socket; // 声明Socket对象

private BufferedReader in; // 声明输入流对象BufferedReader

private PrintWriter out; // 声明输出流对象

public Server()

{

try

{

// 创建一个ServerSocket，指定等待端口

ss = new ServerSocket(10000);

while (true)

{

// 使用ServerSocket接收用户请求（处于监听状态）

socket = ss.accept(); // 获得客户端的socket对象

// 创建输入流并读取信息

in = new BufferedReader(new InputStreamReader(socket

.getInputStream()));

String line = in.readLine();

System.out.println("客户端说:" + line);

// 创建输出流并输出信息

out = new PrintWriter(socket.getOutputStream(), true);

out.print("您的信息已接收!");

// 关闭流和socket

out.close();

in.close();

socket.close();

}

} catch (IOException e)

{

e.printStackTrace();

out.println("出错了!");

} // 创建ServerSocket对象

}

public static void main(String[] args)

{// 程序主入口

// 创建对象

new Server();

}

}

客户端：（7分）

public class Client

{ // 声明客户端类

Socket socket; // 声明Socket对象

BufferedReader in; // 声明输入流对象

PrintWriter out; // 声明输出流对象

public Client()

{ // 构造方法

try

{

// 创建Socket对象，用来发送请求

socket = new Socket(InetAddress.getLocalHost(), 10000);

// 创建字符缓冲区对象

BufferedReader line = new BufferedReader(new InputStreamReader(

System.in));

// 创建输出流对象并输出

out = new PrintWriter(socket.getOutputStream(), true);

out.println(line.readLine());

// 创建输入流对象

in = new BufferedReader(new InputStreamReader(socket

.getInputStream()));

System.out.println(in.readLine());

// 关闭流对象

out.close();

in.close();

} catch (Exception e)

{

e.printStackTrace();

System.out.println("出错了!");

}

}

public static void main(String[] args)

{

// 创建对象

new Client();

}

}