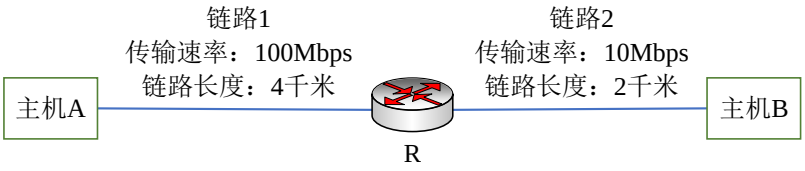
第一章问题

网络结构如下图所示，主机 A 与主机 B 之间通过两段链路和一台转发设备 R 进行连接，每条链路的长度和传输速率已经在图中标出，R 采用存储转发机制。主机 A 向主机 B 发送一个长度为 10000 字节的报文，请回答以下问题（设电磁波传播速度为 $2*10^8$ 米/秒）

(1) 如果采用报文交换，请计算端到端的最小时延，即从主机 A 传输报文的第一位开始，到主机 B 接收到报文的最后一位为止所用的时间。

(2) 如果将报文分成 5 个报文分组传输，请计算完成报文传输的最小端到端时延（忽略报文分组的封装开销）。

在统计多路复用机制中，端到端的时延具有不确定性，请简要分析影响端到端时延的主要因素。



解：

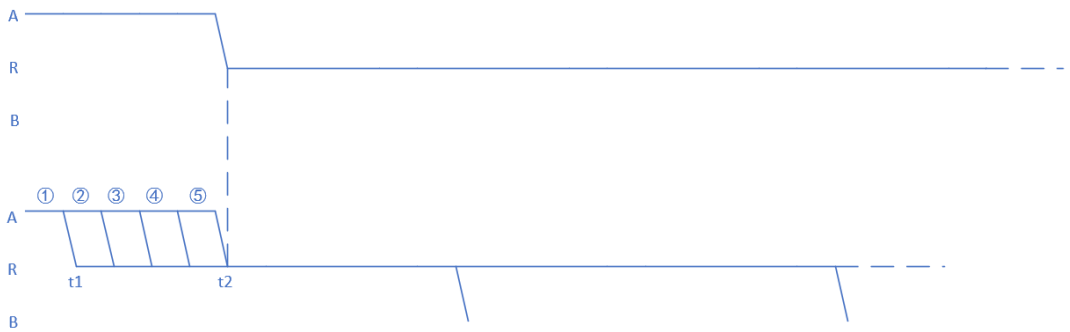
（1）

从 A 发送，到 R 完全接收：$d_1 = \frac{10000*8}{100*10^6} + \frac{4000}{2*10^8} = 0.00082s$

从 R 发送，到 B 完全接收：$d_2 = \frac{10000*8}{10*10^6} + \frac{2000}{2*10^8} = 0.00801s$

总时延为：$d = d_1 + d_2 = 8.83ms$

（2）



如上图所示，图片上侧为没有将报文分成 5 个报文分组传输时的时间图，可以看到，当整个报文完全发送至 R 之后，在 t2 时刻，R 才开始将报文转发给 B。而图片下侧为将报文分成 5 个报文分组后的时间图，可以看到，在 t1 时刻，R 接收到分组 1 的所有内容，并开始将分组 1 转发给 B。这里由于链路 2 的传输速率为链路 1 的十分之一，当分组 1 还没有传输完时，其他四个分组均会到达 R。我们可以很容易的得出两种方式总耗时相差了(t2-t1)，即四个报文分组在链路 1 中的总传输时间。

因此，将报文分成 5 个报文分组后，传输的最小端端延时为：$d' = d - (t2 - t1) = 8.83ms - \frac{4}{5} * \frac{10000*8}{100*10^6} s = 8.19ms$

（3）

影响端到端时延的主要因素：

1. 网络本身的带宽、传输速度、端到端的相隔距离；

2. 对于一个报文来说，其传输时延、传播时延、处理时延等等基本是确定的，对端到端时延影响比较大的是报文在路由器中的排队时延。如果数据传入路由器的速率高于传出路由器的速率，将会导致较为严重的排队现象，会造成较大的排队时延。

第二章问题

（1）通过使用 Windows 命令行模式提供的 nslookup 命令查询 www.baidu.com 的 IP 地址，给出结果截图，并对返回的结果进行解释。同时，利用 Wireshark 捕获查询的交互过程，给出结果截图，并进行简要说明。

（2）以反复解析为例，说明域名解析的基本工作过程（可以结合图例）。给出内容分发网络（CDN）中 DNS 重定向的基本方法，说明原始资源记录应该如何修改，并描述重定向过程。

（3）在 DNS 域名系统中，域名解析时使用 UDP 协议提供的传输层服务（DNS 服务器使用 UDP 的 53 端口），而 UDP 提供的是不可靠的传输层服务，请你解释 DNS 协议应如何保证可靠机制。

解：

（1）

①.nslookup 查询结果分析

返回结果解释：

服务器： UnKnown

Address：222.30.45.41

返回本机的 DNS 服务器的名称（由于没有找到本地 DNS 服务器 IP 地址对应的 PTR 资源记录，这里返回 UnKnown）以及本机的 DNS 服务器 IP 地址。

非权威应答

表示下面几行的数据不是从权威 DNS 服务器上返回的，而是从非权威服务器的缓存中查找到的数据。

名称: www.a.shifen.com

百度服务器的域名，是百度原来的网址。

Addresses: 182.61.200.6

182.61.200.7

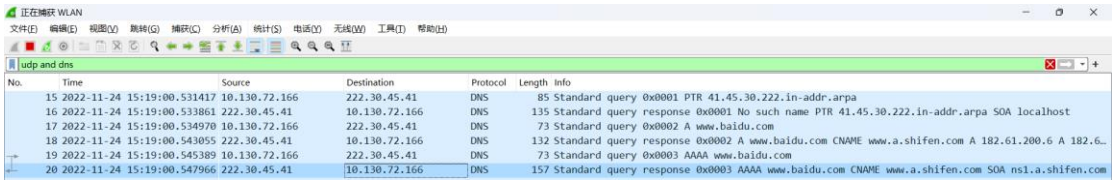
百度服务器的 IP 地址。

Aliases: www.baidu.com

DNS 资源记录中的一个别名。

②.Wireshark 抓包分析

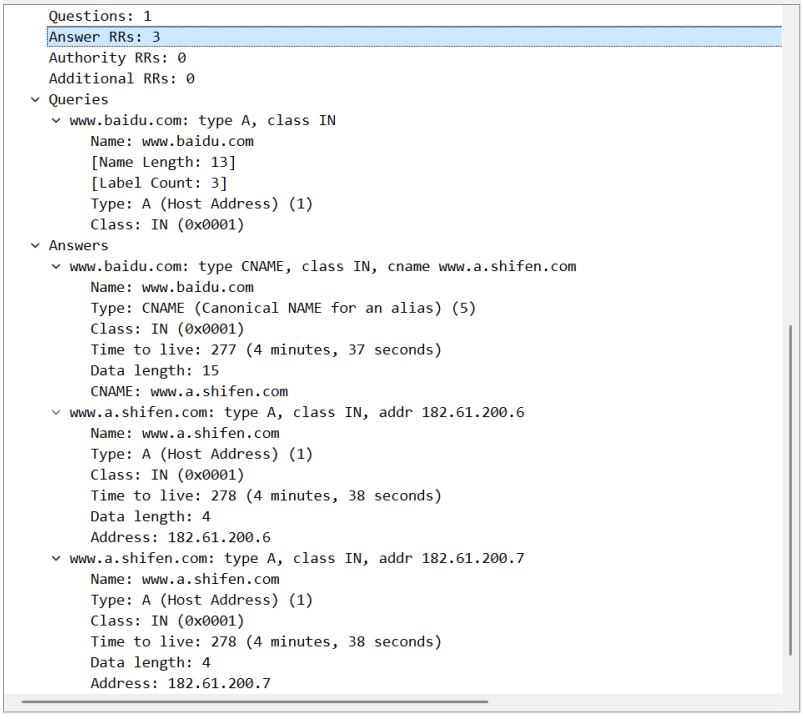
将过滤器设置为 udp and dns。得到结果如下所示：



从上图我们可以看到，nslookup 的整个查找过程一共分为三个阶段：通过 IP 反向解析获取本地 DNS 服务器的域名；通过本地 DNS 服务器查询 www.baidu.com 的 IPV4 地址；通过本地 DNS 服务器查询 www.baidu.com 的 IPV6 地址。

a) 通过 IP 反向解析获取本地 DNS 服务器的域名：对应标号 15 和 16 的两个数据包。首先本机向本地 DNS 服务器发送一个标准查询报文，查询类型为 PTR 类型，查询域名为 41.45.30.222.in-addr.arpa，即反向解析 222.30.45.41IP 地址（本地 DNS 服务器的 IP 地址）的合法域名。然后本地 DNS 服务器经过一系列处理，向本机发回一个响应报文，提示没有对应的 PTR 记录，这也是 nslookup 查询结果中服务器为 UnKnown 的原因。

b) 通过本地 DNS 服务器查询 www.baidu.com 的 IPV4 地址：对应标号 17 和 18 的两个数据包。相似的，本机向本地 DNS 服务器发出查询有关 www.baidu.com 的 IPV4 地址信息的请求，DNS 服务器收到请求后，进行一系列的查询，之后发回给本机一个响应报文。本机接收到服务器发回的 DNS 响应数据包，其中包含了 www.baidu.com 的 IPV4 地址信息。如下图所示，一共返回了三条资源记录。分别记录了 www.baidu.com 是 www.a.shifen.com 的别名，www.a.shifen.com 域名对应的 IP 地址为 182.61.200.6 和 182.61.200.7。这与上图中 nslookup 的查询结果相对应。



c) 通过本地 DNS 服务器查询 www.baidu.com 的 IPV6 地址：对应标号为 18 和 19 的两个数据包。该交互过程与上一步是相似的，本机向本地 DNS 服务器发送标准请求报文，请求类型为 AAAA，表示请求 www.baidu.com 域名的 IPV6 地址的记录。然后本地 DNS 服务器返回给本机响应报文。但是通过查看 DNS 发回的响应报文，我并没有找到有关 www.baidu.com 的 IPV6 地址的信息。报文内容如下所示。通过使用 nslookup -qt=aaaa www.baidu.com 命令，也没有查询到相关的 IPV6 地址。网上说百度的 IPV6 地址为

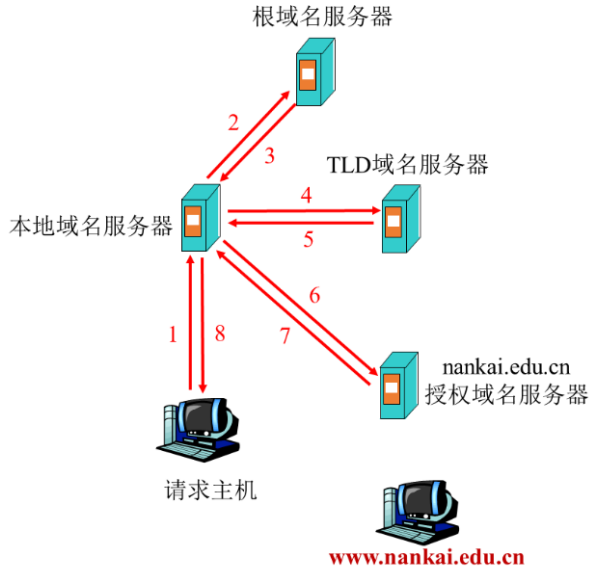
[2400:da00::dbf:0:100]，但是使用浏览器也无法访问该地址。

```
    .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authen
    .... .... ...0 .... = Non-authenticated data: Unacceptable
    .... .... .... 0000 = Reply code: No error (0)
  Questions: 1
  Answer RRs: 1
  Authority RRs: 1
  Additional RRs: 0
▾ Queries
  › www.baidu.com: type AAAA, class IN
▾ Answers
  ▾ www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
      Name: www.baidu.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 277 (4 minutes, 37 seconds)
      Data length: 15
      CNAME: www.a.shifen.com
▾ Authoritative nameservers
  ▾ a.shifen.com: type SOA, class IN, mname ns1.a.shifen.com
      Name: a.shifen.com
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 278 (4 minutes, 38 seconds)
      Data length: 45
      Primary name server: ns1.a.shifen.com
      Responsible authority's mailbox: baidu_dns_master.baidu.com
      Serial Number: 2211240018
      Refresh Interval: 5 (5 seconds)
      Retry Interval: 5 (5 seconds)
      Expire limit: 2592000 (30 days)
      Minimum TTL: 3600 (1 hour)
  [Request In: 19]
  [Time: 0.002577000 seconds]
```

```
C:\Users\Lenovo>nslookup -qt=aaaa www.baidu.com
服务器:  UnKnown
Address:  222.30.45.41

非权威应答:
非权威应答:
名称:    www.baidu.com
```

（2）



域名解析的基本工作过程：

1. 主机期望获得某个域名的 IP 地址，先查询缓存内是否有相应的记录。若存在直接返回，若不存在向本地 DNS 服务器发送一个 DNS 查询报文，该报文包含被查询的域名。
2. 若本地 DNS 缓存中存在该域名到 IP 地址的映射，则直接跳到第 8 步，向请求主机返回响应报文，否则本地 DNS 服务器将该报文转发到根 DNS 服务器，执行第三步。
3. 根 DNS 服务器检查其所属的 TLD 服务器，并向本地 DNS 服务器返回负责该域名的 TLD 的 IP 地址列表。
4. 本地 DNS 服务器向得到的 TLD 服务器 IP 地址发送查询报文。
5. TLD 服务器查看其所属的权威 DNS 服务器，并用该权威 DNS 服务器的 IP 地址作出响应，发回给本地 DNS 服务器。
6. 本地 DNS 再向对应的权威 DNS 服务发送查询报文。
7. 权威 DNS 服务器根据接收到的请求包中的域名进行查询，用该域名对应的 IP 地址进行响应，发回给本地 DNS 服务器。本地 DNS 服务器缓存中存入该域名到 IP 地址的映射。
8. 本地 DNS 服务器将该 IP 地址返回给请求主机，请求主机缓存该域名到 IP 地址的映射。至此，查询结束。

CDN 中，重定向的方法分为两种：

1. HTTP 重定向：

在客户端访问 DNS 服务器后，返回的 IP 地址固定是一个原始服务器的 IP 地址，客户端具体被分配给哪一个 CDN 服务器完全由原始服务器决定，原始服务器会根据当前各个 CDN 服务器的负载状况和位置，向客户端发送一个 HTTP 响应，其中的 Location 指明了分配给该客户端的 CDN 服务器位置。

2. DNS 辅助重定向：

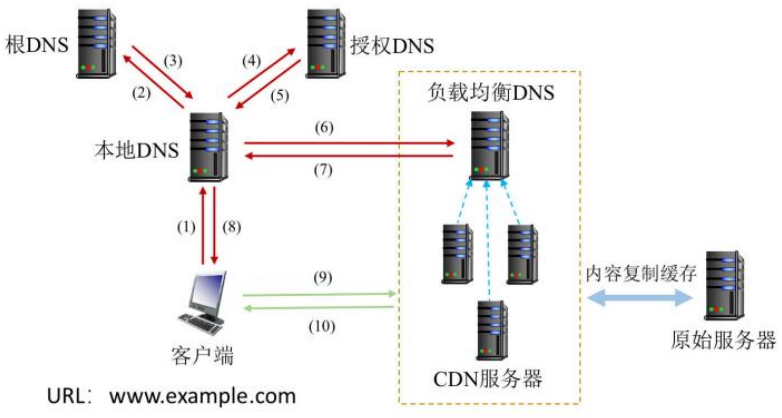
在本地 DNS 服务器请求完权威 DNS 服务器后，会进一步去访问负载均衡 DNS 服务器，负载均衡 DNS 服务器基于 CDN 服务器的负载状况和位置来决定 CDN 服务器的选择，并将选择的 CDN 服务器的 IP 告知本地的 DNS 服务器，再通过本地 DNS 服务器将 IP 地址传递给客户端。

原始资源记录应该如何修改：

对于 HTTP 重定向的方法，不需要对原始资源记录进行修改。

对于 DNS 辅助重定向的方法，目前最常用的是使用 CNAME 的方式。在第一次原始域名解析的时候，由于 CDN 对域名解析过程进行了调整，所以解析得到的是该域名对应的 CNAME 记录，为了得到实际 IP 地址，浏览器需要再次对获得的 CNAME 域名进行解析以得到实际的 IP 地址；在此过程中，使用的全局负载均衡 DNS 解析，如根据地理位置信息解析对应的 IP 地址，使得用户能就近访问。

DNS 辅助重定向过程：



前五步与域名解析的过程是相似的。在第五步中，授权 DNS 服务器对此域名的解析设置了 CNAME 记录，最终请求将会被指向 CDN 网终中的 DNS 负载均衡系统。智能 DNS 负载均衡系统对域名进行智能解析，将响应速度最快的 CDN 服务器节点 IP 地址返回给本地 DNS 服务器，然后本地 DNS 服务器会将得到的 IP 地址返回给用户。

（3）DNS 协议如何保证可靠性

UDP 协议是传输层协议，DNS 协议是应用层协议，应用层协议运行在传输层协议之上。既然传输层不是可靠的，就要在应用层上实现一定的可靠性。可以使用课上学到的可靠数据传输协议 RDT 的思想，如自动重传请求、设定 ACK 消息用于通知是否正确接收消息、通过校验和来进行差错检测、为数据包添加序号、添加定时器超时重传等方式保证可靠性。

通过我们在 nslookup 实验中的观察，发现在 DNS 应用层，其报文格式中包括了问题的数量、回答的数量，可以用来进行一定的校验。返回的报文中，也包含着发送过去的问题，也可以在一定程度上保证可靠性。此外，DNS 服务器中的资源记录还有以秒为单位的生存周期，当生存周期过期后，会重新请求更新，以保证数据的正确性。DNS 协议还会通过冗余设置，避免单点失效。具体来说，为了保证高可用性，会有多台权威服务器冗余支持每个区域。某个区域的资源记录通过手动或自动方式更新到单个主权威服务器上，其它冗余服务器用作同一区域中主服务器的备份服务器，以防主服务器无法访问或宕机。