

计算机网络:实验室1

- Student1 的姓名 (电子邮件)

- Student2 的姓名 (电子邮件)

提交材料清单:

- 实验报告

- 代码

检查点截止时间:2023 年 10 月 29 日下午 23:59

第 1 部分:简单的 Socket 程序 (Python、Java、C) (26%)

请选择以下主题之一来完成。

分数点:

- 程序正确性 (20%)

- 代码运行截图&代码说明 (6%)

- TCP 套接字程序

问题要求:

客户端程序创建一个TCP套接字,然后向指定的服务器地址和端口发起连接,等待服务器连接,然后通过套接字发送用户输入的字符串,然后显示服务器返回的消息。

服务器程序始终维护一个TCP欢迎套接字,可以接收任何客户端的连接请求。收到客户端的连接请求后,创建一个新的TCP连接套接字,用于单独与客户端通信,同时显示客户端地址和端口。接收到客户端发送的字符串后,将其改为大写,然后将修改后的字符串返回给客户端。最后,关闭TCP连接套接字。

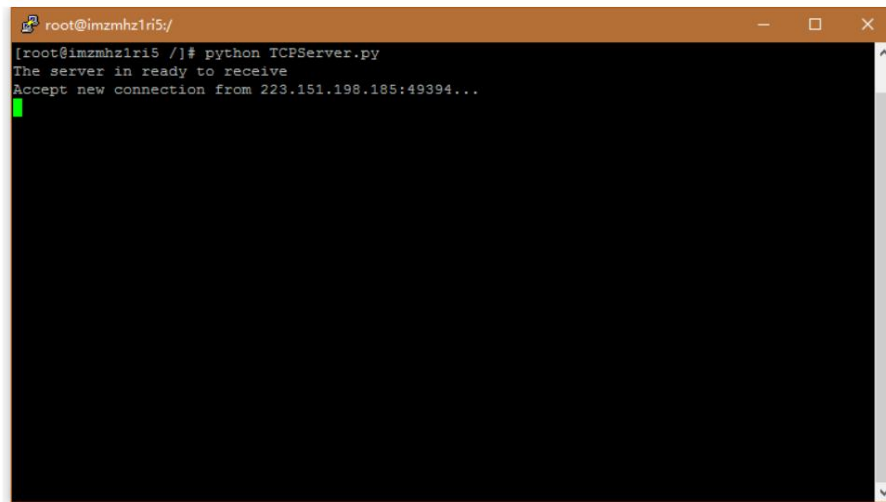
运行结果:

客户



```
Windows PowerShell
FS C:\> python D:\TCPClient.py
Input lowercase sentence:hello
HELLO
FS C:\>
```

服务器



```
root@imzmhz1r15:/
[root@imzmhz1r15 /]# python TCPServer.py
The server is ready to receive
Accept new connection from 223.151.198.185:49394...
```

· UDP 套接字程序

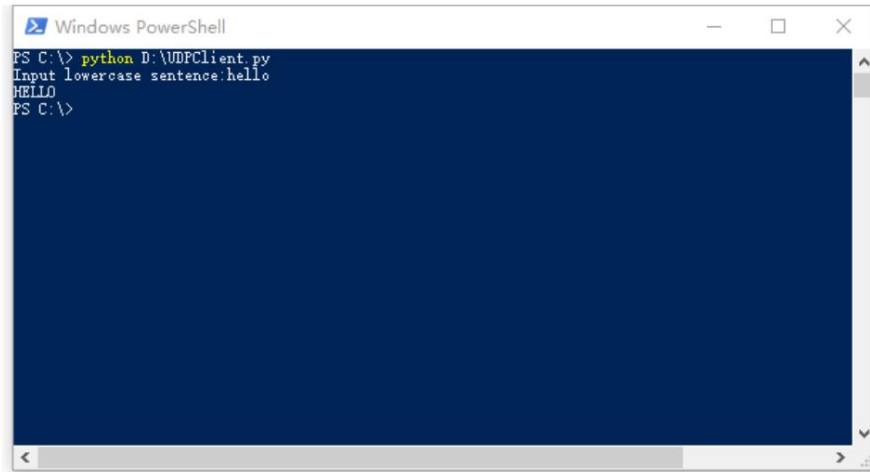
问题要求：

客户端程序在用户输入一串小写字母后创建一个UDP套接字并将其发送到指定的服务器地址和相应的端口。它等待服务器返回消息,然后显示该消息。

服务器程序始终维护一个可连接的UDP套接字。接收到字符串后,将其更改为大写,然后将修改后的字符串返回给客户端。

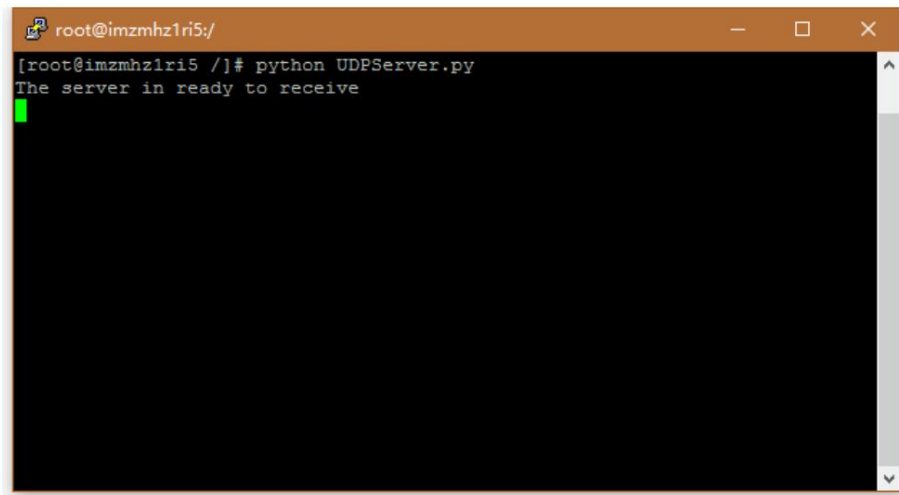
运行结果：

客户



```
Windows PowerShell
PS C:\> python D:\UDPCliet.py
Input lowercase sentence:hello
HELLO
PS C:\>
```

服务器



```
root@imzmhz1ri5:/
[root@imzmhz1ri5 /]# python UDPServer.py
The server is ready to receive
```

第 2 部分:HTTP

在介绍性实验室中熟悉了 Wireshark 数据包嗅探器后,我们现在准备使用 Wireshark 来研究运行中的协议。在本实验中,我们将探讨 HTTP 协议的几个方面:基本的 GET/响应交互、HTTP 消息格式、检索大型 HTML 文件、检索包含嵌入对象的 HTML 文件以及 HTTP 身份验证和安全性。在开始这些实验之前,您可能需要查看本文的第 2.2 节。

分数点：

- 小问题 (1.5% * 19)
- 实验截图&文字描述的连贯性、逻辑性和简洁性 (10%)

· 基本HTTP GET/响应交互

让我们通过下载一个非常简单的 HTML 文件开始对 HTTP 的探索 该文件非常短,并且不包含任何嵌入对象。请执行下列操作:

- 1. 启动您的网络浏览器。
- 2. 启动 Wireshark 数据包嗅探器,如介绍性实验中所述 (但尚未开始数据包捕获)。在显示过滤器规范窗口中输入 “http” (仅字母,而不是引号,并且是小写),以便稍后在数据包列表窗口中仅显示捕获的 HTTP 消息。

(我们在这里只对 HTTP 协议感兴趣,并且不想看到所有捕获的数据包的混乱)。

- 3. 等待一分钟多一点 (我们很快就會知道原因),然后开始 Wireshark 数据包捕获。
- 4. 在浏览器中输入以下内容
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>
您的浏览器应该显示非常简单的一行 HTML 文件。
- 5. 停止Wireshark抓包。

您的 Wireshark 窗口应类似于图 1 中所示的窗口。如果您无法在实时网络连接上运行 Wireshark,您可以下载执行上述步骤时创建的数据包跟踪。

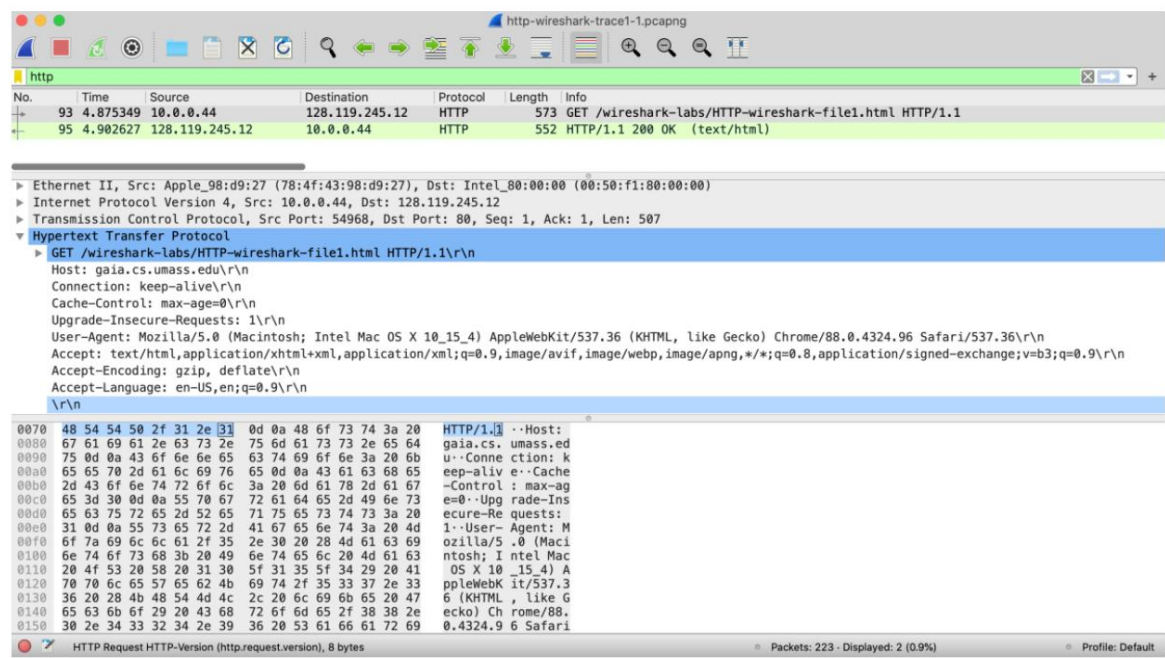


图 1:浏览器检索到 http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html 后 Wireshark 显示

图 1 中的示例显示数据包列表窗口中捕获了两条 HTTP 消息:GET 消息 (从浏览器到 gaia.cs.umass.edu Web 服务器)和从服务器到浏览器的响应消息。数据包内容窗口

显示所选消息的详细信息（在本例中为 HTTP OK 消息,在数据包列表窗口中突出显示）。回想一下,由于 HTTP 消息是在 TCP 段内承载的,而 TCP 段是在 IP 数据报内承载的,而 IP 数据报是在以太网帧内承载的,因此 Wireshark 还会显示帧、以太网、IP 和 TCP 数据包信息。我们希望最大限度地减少显示的非 HTTP 数据量（我们在这里对 HTTP 感兴趣,并将在稍后的实验室中研究这些其他协议）,因此请确保帧最左侧的框、以太网、IP 和 TCP 信息有一个加号或一个向右的三角形（这意味着有隐藏的、未显示的信息）,HTTP 线路有一个减号或一个向下的三角形（这意味着显示有关 HTTP 消息的所有信息）。

（注意:您应该忽略 favicon.ico 的任何 HTTP GET 和响应。如果您看到对此文件的引用,那么您的浏览器会自动询问服务器它（服务器）是否有一个小图标文件,该文件应显示在浏览器中显示的 URL。在本实验中我们将忽略对此讨厌文件的引用。）。

通过查看 HTTP GET 和响应消息中的信息,回答以下问题。

1. 您的浏览器运行的是 HTTP 版本 1.0、1.1 还是 2? HTTP 是什么版本
服务器正在运行?
2. 您的浏览器表明服务器可以接受哪些语言（如果有）?
3. 您电脑的 IP 地址是多少? IP 地址是什么
gaia.cs.umass.edu 服务器?
4. 服务器返回给浏览器的状态码是什么?
5. 您正在检索的 HTML 文件最后一次在服务器上修改是什么时候?
6. 有多少字节的内容返回到您的浏览器?
7. 通过检查数据包内容窗口中的原始数据,您是否看到数据中未显示在数据包列表窗口中的任何标头?
如果是这样,请命名
一。

在您对上述问题 5 的回答中（假设您“实时”运行 Wireshark,而不是使用之前记录的跟踪文件）,您可能会惊讶地发现您刚刚检索的文档的最后修改时间是在之前的一分钟内。您下载了该文档。这是因为（对于这个特定文件）,gaia.cs.umass.edu 服务器将文件的上次修改时间设置为当前时间,并且每分钟执行一次。因此,如果您在两次访问之间等待一分钟,该文件将显示为最近被修改过,因此您的浏览器将下载该文档的“新”副本。

· HTTP CONDITIONAL GET/响应交互

回想一下本文的 2.2.5 节,大多数 Web 浏览器都会执行对象缓存,因此在检索 HTTP 对象时通常会执行条件 GET。在执行以下步骤之前,请确保浏览器的缓存为空。现在执行以下操作:

- 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。

- 启动Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>
您的浏览器应该显示一个非常简单的五行 HTML 文件。 · 再次快速在浏览器中输入相同的 URL（或只需选择浏览器上的刷新按钮）
- 停止Wireshark 数据包捕获,并在显示过滤器规范窗口中输入“http”（同样是小写字母,不带引号）,以便稍后在数据包列表窗口中仅显示捕获的HTTP 消息。

如果您无法在实时网络连接上运行 Wireshark（或者无法让浏览器在第二个 HTTP GET 请求上发出 If-Modified-Since 字段）,您可以下载执行上述步骤时创建的数据包跟踪被跟踪。回答下列问题：

8. 检查从浏览器到服务器的第一个 HTTP GET 请求的内容。您是否在 HTTP GET 中看到“IF-MODIFIED-SINCE”行？
9. 检查服务器响应的内容。服务器是否明确返回文件的内容？你怎么知道？
10. 现在检查从浏览器到服务器的第二个 HTTP GET 请求的内容。您是否在 HTTP GET 中看到“IF-MODIFIED-SINCE:”行？如果是这样,“IF-MODIFIED-SINCE:”标头后面有哪些信息？
11. 服务器响应第二次 HTTP GET 返回的 HTTP 状态代码和短语是什么？服务器是否显式返回了文件的内容？
解释。

· 检索长文档

到目前为止,在我们的示例中,检索到的文档都是简单且简短的 HTML 文件。接下来让我们看看当我们下载一个长 HTML 文件时会发生什么。请执行下列操作：

- 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。
- 启动Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>
您的浏览器应该显示相当冗长的美国权利法案。
- 停止Wireshark 数据包捕获,并在显示过滤器规范窗口中输入“http”,以便仅显示捕获的HTTP 消息。

在数据包列表窗口中,您应该看到 HTTP GET 消息,后面是对 HTTP GET 请求的多数据包 TCP 响应。确保清除 Wireshark 显示过滤器,以便多数据包 TCP 响应将显示在数据包列表中。

这个多数据包响应值得一些解释。回想一下 2.2 节（参见正文中的图 2.9）,HTTP 响应消息由状态行、标题行、空行和实体主体组成。就我们而言

HTTP GET,响应中的实体主体是整个请求的 HTML 文件。在我们的例子中,HTML 文件相当长,4500 字节太大,无法容纳在一个 TCP 数据包中。因此,单个 HTTP 响应消息被 TCP 分成几个部分,每个部分都包含在一个单独的 TCP 段中(参见正文中的图 1.24)。在 Wireshark 的最新版本中,Wireshark 将每个 TCP 段指示为单独的数据包,并且单个 HTTP 响应跨多个 TCP 数据包分段的事实由 Wireshark 显示的信息列中的“重组 PDU 的 TCP 段”指示。

回答下列问题:

12. 您的浏览器发送了多少条 HTTP GET 请求消息?跟踪中的哪个数据包编号包含法案或权利的 GET 消息?
13. 跟踪中的哪个数据包编号包含关联的状态代码和短语与 HTTP GET 请求的响应?
14. 响应中的状态代码和短语是什么?
15. 承载单个 HTTP 需要多少个包含数据的 TCP 段
回应和权利法案的文本?

· 带有嵌入对象的 HTML 文档

现在我们已经了解了 Wireshark 如何显示捕获的大型 HTML 文件的数据包流量,我们可以看看当您的浏览器下载带有嵌入对象的文件(即包含其他对象的文件)时会发生什么情况(在下面的示例中为图像文件)存储在另一台服务器上。

请执行下列操作:

- 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。
- 启动Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>
您的浏览器应显示一个包含两个图像的简短 HTML 文件。这两个图像在基本 HTML 文件中引用。也就是说,图像本身不包含在 HTML 中;相反,图像的 URL 包含在下载的 HTML 文件中。
- 停止Wireshark 数据包捕获,并在显示过滤器规范窗口中输入“http”,以便仅显示捕获的 HTTP 消息。

回答下列问题:

16. 您的浏览器发送了多少条 HTTP GET 请求消息?这些 GET 请求发送到哪些 Internet 地址?
17. 您能否判断您的浏览器是串行下载这两个图像,还是从两个网站并行下载它们?解释。

- HTTP 身份验证

最后,让我们尝试访问受密码保护的网站,并检查与此类网站交换的 HTTP 消息的序列。URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html 受密码保护。用户名是“wireshark-students”(不带引号),密码是“network”(同样不带引号)。那么让我们访问这个“安全”的受密码保护的网站。请执行下列操作:

- 如上所述,确保浏览器的缓存已清除,然后关闭您的浏览器。然后,启动浏览器
- 启动Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
在弹出框中输入请求的用户名和密码。
- 停止Wireshark 数据包捕获,并在显示过滤器规范窗口中输入“http”,以便稍后在数据包列表窗口中仅显示捕获的HTTP 消息。

回答下列问题:

18. 服务器对初始请求的响应(状态代码和短语)是什么?
来自浏览器的 HTTP GET 消息?
19. 当您的浏览器第二次发送HTTP GET消息时,HTTP GET消息中包含哪些新字段?

您输入的用户名(wireshark-students)和密码(network)编码在客户端 HTTP GET 消息中“Authorization: Basic”标头后面的字符串(d2lyZXNoYXJrLXN0dWRLbnRzOm5ldHdvcm5l)中。虽然您的用户名和密码可能看起来已加密,但它们只是以 Base64 格式进行编码。用户名和密码未加密!要查看此内容,请访问<http://www.motobit.com/util/base64-decoder-encoder.asp>并输入base64编码的字符串d2lyZXNoYXJrLXN0dWRLbnRz并解码。瞧!您已从 Base64 编码转换为 ASCII 编码,因此应该可以看到您的用户名!

要查看密码,请输入字符串 Om5ldHdvcm5l 的其余部分并按“解码”。由于任何人都可以下载像 Wireshark 这样的工具并嗅探通过其网络适配器传递的数据包(不仅仅是他们自己的数据包),并且任何人都可以将 Base64 转换为 ASCII(您刚刚做到了!),因此您应该清楚 WWW 上的简单密码除非采取额外措施,否则站点并不安全。

不要害怕!正如我们将在第 8 章中看到的,有一些方法可以使 WWW 访问更加安全。然而,我们显然需要一些超出基本 HTTP 身份验证框架的东西!

第三部分 :DNS

正如本文第 2.4 节所述,域名系统 (DNS) 将主机名转换为 IP 地址,在互联网基础设施中发挥着关键作用。在本实验中,我们将仔细研究 DNS 的客户端。回想一下,客户端在 DNS 中的角色相对简单 – 客户端向其本地 DNS 服务器发送查询,并接收返回的响应。正如教科书中的图 2.19 和 2.20 所示,当分层 DNS 服务器相互通信以递归或迭代地解析客户端的 DNS 查询时,许多事情都可以在“幕后”进行,而 DNS 客户端是看不到的。然而,从 DNS 客户端的角度来看,该协议非常简单 – 向本地 DNS 服务器制定查询并从该服务器接收响应。

在开始本实验之前,您可能需要阅读本文的第 2.4 节来回顾 DNS。特别是,您可能需要查看有关本地 DNS 服务器、DNS 缓存、DNS 记录和消息以及 DNS 记录中的 TYPE 字段的材料。

分数点:

- 小问题 (1.5% * 17)
- 实验截图&文字描述的连贯性、逻辑性和简洁性 (10%)

· nslookup

让我们通过检查 nslookup 命令开始对 DNS 的研究,该命令将调用底层 DNS 服务来实现其功能。

nslookup _

该命令在大多数 Microsoft、Apple iOS 和 Linux 操作系统中可用。要运行 nslookup,只需在 DOS 窗口、Mac iOS 终端窗口或 Linux shell 的命令行中键入 nslookup 命令即可。

在最基本的操作中, nslookup 允许运行 nslookup 的主机查询任何指定的 DNS 服务器以获取 DNS 记录。查询的 DNS 服务器可以是根 DNS 服务器、顶级域 (TLD) DNS 服务器、权威 DNS 服务器或中间 DNS 服务器 (有关这些术语的定义,请参阅教科书)。例如, nslookup 可用于检索将主机名映射到其 IP 地址的“Type=A”DNS 记录。为了完成此任务, nslookup 向指定的 DNS 服务器 (或者运行 nslookup 的主机的默认本地 DNS 服务器,如果未指定特定的 DNS 服务器) 发送 DNS 查询,从该 DNS 服务器接收 DNS 响应,然后显示结果。

让我们来试试 nslookup ! 我们首先在位于复旦大学的 www.fudan.edu.cn 主机上的 Linux 命令行上运行 nslookup。

命令: nslookup www.fudan.edu.cn

除了使用 nslookup 查询 DNS “Type=A”记录外,我们还可以使用 nslookup to nslookup 查询 “TYPE=NS”记录,该记录返回主机名

权威 DNS 服务器的（及其 IP 地址），该服务器知道如何获取权威服务器域中主机的 IP 地址。

使用选项 “-type=NS”和域 “fudan.edu.cn”调用nslookup会导致nslookup向默认本地 DNS 服务器发送对 type=NS 记录的查询。换句话说,该查询是在说:“请将 fudan.edu.cn 的权威 DNS 的主机名发送给我”。
(当不使用-type选项时, nslookup使用默认值,即查询类型A记录。)

除了 “-type=NS”之外, nslookup还有许多您可能想要探索的附加选项。这是一个网站,其中包含 10 个流行的nslookup用途的屏幕截图: <https://www.cloudns.net/blog/10-most-used-nslookup-commands/>以下是 nslookup 的 “手册页”: <https://linux.die.net/man/1/nslookup>。

最后,我们有时可能有兴趣发现与给定 IP 地址关联的主机名称。nslookup还可用于执行所谓的 “反向 DNS 查找”。例如,我们可以指定一个 IP 地址作为 nslookup 参数,然后nslookup返回具有该地址的主机名。

现在我们已经提供了 nslookup 的概述,现在您可以亲自测试一下它了。执行以下操作（并写下结果）。

1. 运行nslookup获取复旦大学Web服务器IP地址:
https://www.fudan.edu.cn.fudan.edu.cn的IP地址是多少?
2. 为您提供答案的 DNS 服务器的 IP 地址是什么
上面问题 1 中的nslookup命令?
3. 上述问题 1 中nslookup命令的答案是否来自
权威服务器还是非权威服务器?
4. 使用nslookup命令确定fudan.edu.cn的权威名称服务器名称。那名字是什么? (如果有多个权威服务器,则nslookup 返回的第一个权威服务器的名称是什么)?如果您必须找到该权威名称服务器的 IP 地址,您会怎么做?

· 您计算机上的 DNS 缓存

从我们教科书中对迭代和递归 DNS 查询解析的描述 (图 2.19 和 2.20)来看,您可能会认为每次都必须联系本地 DNS 服务器。

应用程序需要将主机名转换为 IP 地址的时间。但实际情况并非总是如此!

大多数主机 (例如,您的个人计算机)都会保留最近检索的 DNS 记录的缓存 (有时称为 DNS解析器缓存),就像许多 Web 浏览器保留最近通过 HTTP 检索的对象的缓存一样。当主机需要调用DNS服务时,该主机首先会检查所需的DNS记录是否驻留在该主机的DNS缓存中;如果找到该记录,主机甚至不会费心联系本地 DNS 服务器,而是使用此缓存的 DNS 记录。解析器缓存中的 DNS 记录最终将

超时并从解析器缓存中删除,就像本地 DNS 服务器中缓存的记录 (参见图 2.19.2.20)将超时一样。

您还可以明确清除 DNS 缓存中的记录。这样做并没有什么坏处,只是意味着您的计算机下次需要使用 DNS 名称解析服务时需要调用分布式 DNS 服务,因为它将在缓存中找不到任何记录。在 Mac 计算机上,您可以在终端窗口中输入以下命令来清除 DNS 解析器缓存:

```
sudo killall -HUP mDNSResponder
```

在 Windows 计算机上,您可以在命令提示符下输入以下命令:

```
ipconfig /flushdns
```

Linux 计算机上输入:

```
sudo systemd-resolve --flush-caches
```

· 使用 Wireshark 跟踪 DNS

现在我们已经熟悉了 nslookup 和清除 DNS 解析器缓存,我们准备开始处理一些重要的事情了。让我们首先捕获普通网上冲浪活动生成的 DNS 消息。

- 清除主机中的 DNS 缓存,如上所述。
- 打开 Web 浏览器并清除浏览器缓存。
- 打开 Wireshark 并在显示过滤器中输入 `ip.addr == <your_IP_address>`, 其中 `<your_IP_address>` 是您计算机的 IPv4 地址。使用此过滤器, Wireshark 将仅显示源自或发往您的主机的数据包。
- 在 Wireshark 中启动数据包捕获。
- 使用浏览器访问网址: `https://www.fudan.edu.cn/`
- 停止数据包捕获。

回答下列问题。

5. 找到第一个解析名称 `fudan.edu.cn` 的 DNS 查询消息。DNS 查询消息的跟踪中的数据包编号是多少?该查询消息是通过 UDP 还是 TCP 发送的?
6. 现在找到对初始 DNS 查询的相应 DNS 响应。DNS 响应消息跟踪中的数据包编号是多少?该响应消息是通过 UDP 还是 TCP 接收的?
7. DNS 查询报文的目的端口是什么?源端口是什么
DNS 响应消息?
8. DNS 查询报文发送到什么 IP 地址?
9. 检查 DNS 查询消息。这个 DNS 消息有多少“问题”
包含?它包含多少个“答案”答案?
10. 检查对初始查询消息的 DNS 响应消息。多少
此 DNS 消息包含“问题”吗?它包含多少个“答案”答案?

现在让我们来玩一下nslookup。 ·
开始抓包。 · 登录
www.fudan.edu.cn 进行nslookup
· 停止抓包。

您应该在 Wireshark 窗口中看到类似于以下内容的跟踪。
让我们看一下第一个类型A查询（由该数据包的信息列中的“A”表示）。

11. DNS查询报文的目的端口是什么?源端口是什么
DNS 响应消息?
12. DNS查询报文发送到什么IP地址?这是您的 IP 地址吗
默认本地 DNS 服务器?
13. 检查 DNS 查询消息。DNS 查询是什么“类型”?查询消息中是否包含任何“答案”?
- 14.检查对查询消息的DNS响应消息。多少
此 DNS 响应消息包含“问题”吗?有多少个“答案”?

最后,让我们使用nslookup发出一个命令,该命令将返回 NS 类型的 DNS 记录,输入以下命令:

```
nslookup -type=NS fudan.edu.cn
```

然后回答以下问题:

15. DNS查询报文发送到什么IP地址?这是您的 IP 地址吗
默认本地 DNS 服务器?
16. 检查 DNS 查询消息。该查询有多少个问题?
查询消息中是否包含任何“答案”?
17. 检查 DNS 响应消息。回应有多少个答案?
答案中包含哪些信息?返回了多少额外的资源记录?这些附加资源记录中包含哪些附加信息?