ELEC3506/9506通信网络
-实验室03

教学

本实验课程分为两个阶段，每个阶段包含多个问题，学生需在实验过程中完成作答以理解该阶段的核心目标。虽然无需将答案纳入最终实验报告，但需提交至Canvas平台的最终报告不得超过7页，内容应涵盖实验过程中的整体学习体验及主要研究成果。第一阶段中，学生将探究传输控制协议（TCP）的工作原理。具体操作包括：向远程服务器传输150KB文件时分析TCP数据包传输轨迹，研究序列号与确认号在保障数据可靠传输中的作用，同时掌握拥塞控制算法（慢启动与拥塞避免机制），最后通过分析吞吐量曲线和往返时间图，评估系统与服务器间TCP连接的实际性能表现。在第二阶段，学生将探索超文本传输协议（HTTP）的几个方面，包括基本的GET/响应交互、HTTP消息格式、检索大型HTML文件、检索带有嵌入对象的HTML文件以及HTTP身份验证和安全。

每份实验报告都提供了一个模板，您需要按照该模板来准备最终的实验报告。您可以在Canvas平台找到该模板。在开始实验前，学生应按照实验课提供的说明进行操作。具体说明如下：

-您需要遵循Canvas中提供的关于组建小组和提交最终实验报告的实验室说明。
-按照以下页面中的说明完成实验。
-你应该遵守实验室规则，并在实验室里表现得体。
-相关资料：TCP -第715 - 735页；三向抖动-第723 - 724页；推送数据（PSH）-第725 - 726页；
控制与容限——第769 - 773页；HTTP——第861 - 868页。
-此外，还有关于TCP的补充说明如下：
- *o在Wireshark中查看TCP流的更简单方法：统计->流图->选择流类型：TCP流->确定*
- o过滤您想要的特定TCP流：右键单击一条消息->跟踪TCP流
- o第1节TCP Q4-15：强烈建议使用zip文件。
- o第1节TCP Q3和Q14是可选的。
- o Q7的有用公式：
  估计的RTT=0.875×前估计的RTT+0.125×样本RTT，其中样本RTT是您可以在Wireshark中找到的实际RTT。
- o了解TCP序列和确认号的好文章：
  http://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-

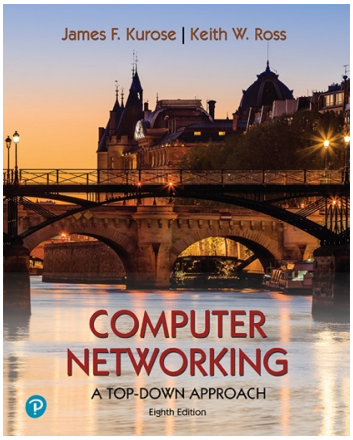
致谢-编号/？pdf

注：学生可以在本课程提供的主要参考文献中找到相应的概念：

阶段1：
行车调度台

计算机网络：自上而下的方法，第8th 版，J.F. Kurose和K.W. Ross

“告诉我，我忘记。给我看，我记住。让我参与，我理解。”中国谚语



在本次实验中，我们将深入探究经典TCP协议的工作原理。通过分析从本地计算机向远程服务器传输150KB文件（包含刘易斯·卡罗尔《爱丽丝梦游仙境》文本）时发送和接收的TCP数据包轨迹，我们将系统研究该协议如何利用序列号与确认号实现可靠传输；观察拥塞控制算法——慢启动机制与拥塞避免机制的实际运作；解析接收端主动通告的流量控制机制。此外，我们还将简要探讨TCP连接建立过程，并测试您计算机与服务器之间TCP连接的性能表现（包括吞吐量和往返时间）。

开始本实验前，您可能需要复习教材第3.5和3.7节[1]。

1. 捕获从计算机到远程服务器的大量TCP传输

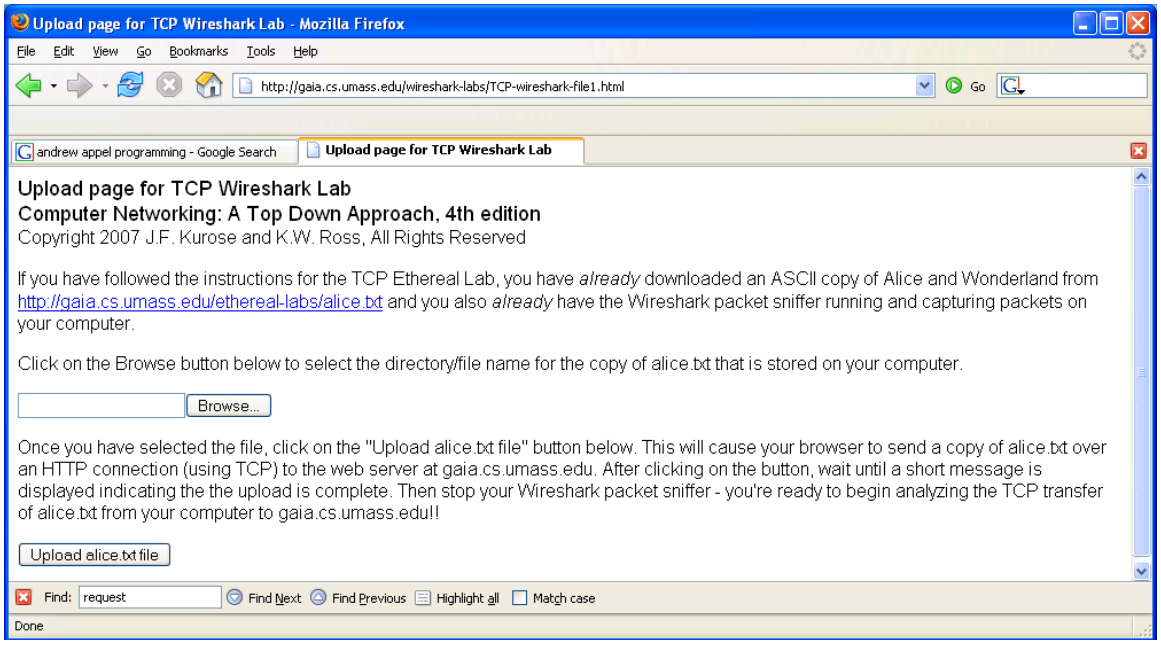
在开始TCP协议的探索之前，我们需要使用Wireshark获取文件传输的TCP数据包跟踪。具体操作是：访问一个网页输入本地存储的文件名（该文件包含《爱丽丝梦游仙境》的ASCII文本），然后通过HTTP POST方法将文件上传至Web服务器（详见教材第2.2.3节）。选择POST方式而非GET方式，是因为需要从本地计算机向另一台计算机传输大量数据。整个过程中我们将持续运行Wireshark，记录并追踪从本地计算机发送和接收的TCP数据段。

执行以下操作：

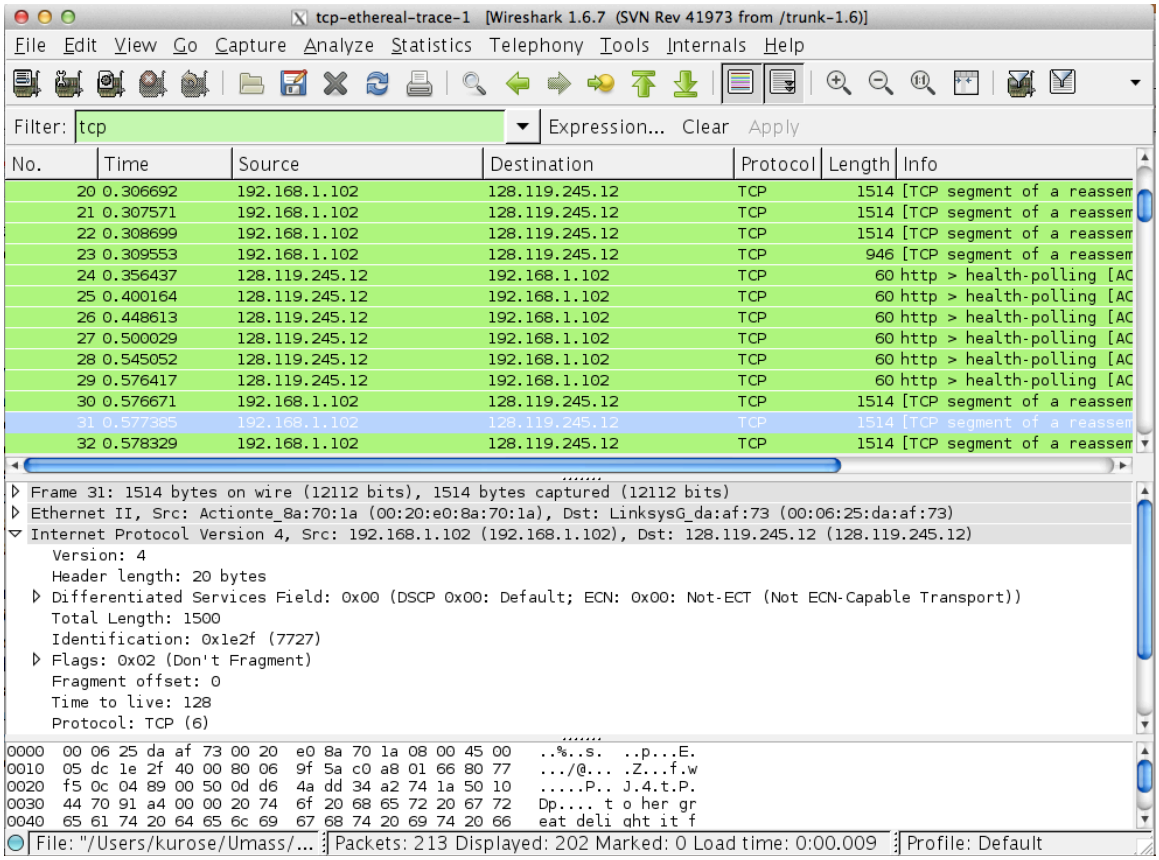
---

[1] *文中引用的图表和章节均以第八版（2020年）《计算机网络：自上而下方法》（J.F.Kurose与K.W.Ross合著，Addison-Wesley/Pearson出版社出版）为依据。*

- 启动您的web浏览器。转到http://gaia.cs.umass.edu/wireshark labs/alice.txt并检索出《爱丽丝梦游仙境》的ASCII副本。将该文件存储在计算机的某个位置。
- 接下来转到http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
- 您应该会看到如下所示的屏幕：



- 使用此表单中的浏览按钮输入计算机上包含Alice in Wonderland的文件名（完整路径名）（或手动输入）。不要立即点击“上传alice.txt文件”按钮。
- 现在启动Wireshark并开始数据包捕获（Capture->Start），然后在Wireshark数据包捕获选项屏幕按OK（此处不需要选择任何选项）。
- 返回浏览器后，点击“上传alice.txt文件”按钮将文件上传至gaia.cs.umass.edu服务器。文件上传完成后，浏览器窗口会显示简短的祝贺信息。
- 停止Wireshark数据包捕获。此时，您的Wireshark窗口应与下面显示的窗口类似。



如果您无法在实时网络连接上运行Wireshark，可以下载一个数据包跟踪文件，该文件是在您按照上述步骤在作者的计算机上进行操作时捕获的[2]。即使您自己也捕获了数据包跟踪并在下面的问题中使用了您自己的跟踪记录，下载这个跟踪文件仍然可能非常有价值。

2. 对捕获的轨迹进行初步观察

在详细分析TCP连接的行为之前，让我们从一个较高的角度查看跟踪。

- 首先，通过输入“tcp”来过滤Wireshark窗口中显示的数据包（小写字母，不带引号，并且输入后别忘了按回车键！）进入Wireshark窗口顶部的显示过滤器规格窗口。

您应该看到计算机与gaia.cs.umass.edu之间的一系列TCP和HTTP消息。您应该看到包含SYN消息的初始三次握手。您应该看到一个HTTP POST消息。这取决于版本。

---

2 下载zip文件http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip并解压缩文件tcpethereal-trace-1。该zip文件中的数据包记录，是作者在计算机上运行Wireshark进行实验时采集的。下载完成后，您可将数据包导入Wireshark，通过文件菜单选择打开选项，随后选取tcp-ethereal-trace-1数据包文件即可查看。

使用Wireshark时，您可能会看到从计算机发送到gaia.cs.umass.edu的一系列“HTTP续传”消息。回想我们之前在HTTP Wireshark实验中的讨论，实际上并不存在所谓的HTTP续传消息——这是Wireshark用来表示需要通过多个TCP数据段来传输单个HTTP消息的特殊标识。在较新版本的Wireshark中，您会在显示信息栏看到“[重组PDU的TCP数据段]”，这表明该
TCP段包含属于上层协议消息（在我们的案例中是HTTP)的数据。您还应该看到从gaia.cs.umass.edu返回到您的计算机的TCP确认段。

请通过打开http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip中的Wireshark捕获数据包文件tcpethereal-trace-1来回答以下问题（即下载该跟踪记录并在Wireshark中打开；参见脚注2）。在回答问题时，应尽可能提交用于解答问题的跟踪记录中的数据包打印件，并在打印件上标注注释说明答案。要打印数据包，请使用文件->打印，选择“仅打印选定的数据包”，选择“数据包摘要行”，并选择回答问题所需的最少数据包详细信息。

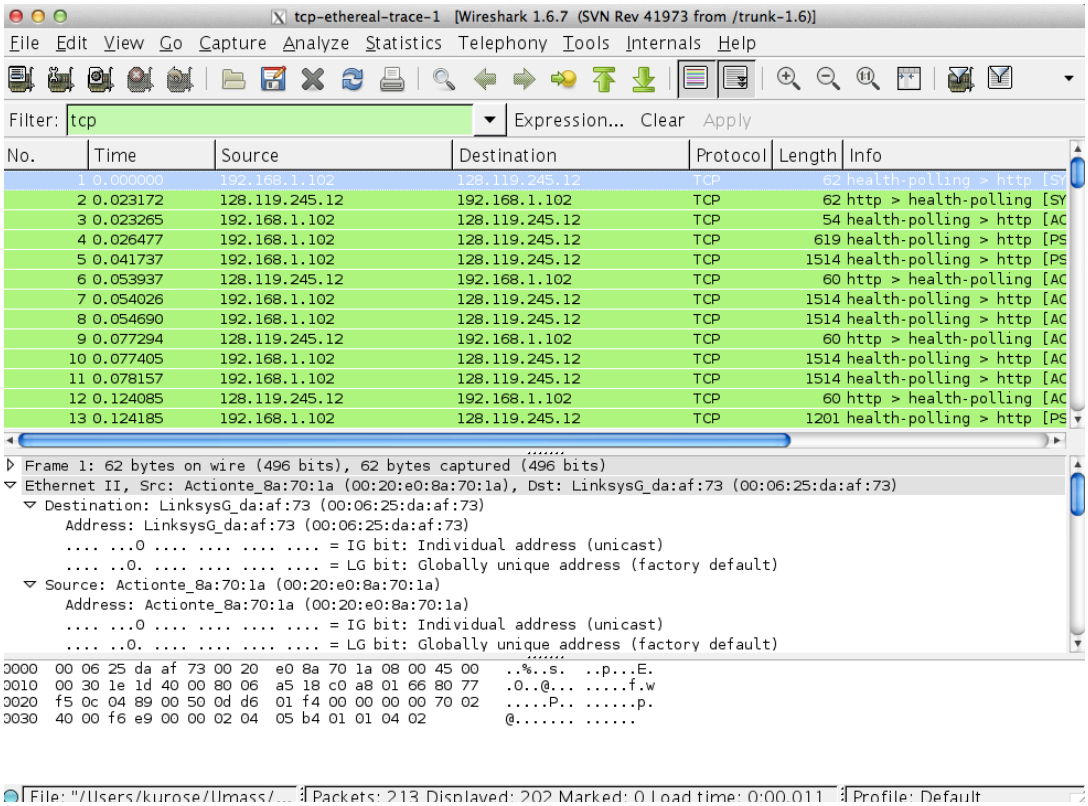
1. 正在向gaia.cs.umass.edu传输文件的客户端计算机（源）使用的是什么IP地址和TCP端口号？要回答这个问题，最简单的方法可能是选择一个HTTP数据包，并查看用于承载该HTTP数据包的TCP数据包细节——具体操作是通过“显示所选数据包头窗口的详细信息”功能（如果对Wireshark窗口设置有疑问，可以参考《Wireshark入门实验》中的图2）。
2. gaia.cs.umass.edu的IP地址是什么？它在哪个端口号上发送和接收该连接的TCP分段？

如果您能够创建自己的跟踪，请回答以下问题：

3. 您的客户端计算机（源）用于将文件传输到gaia.cs.umass.edu的IP地址和TCP端口号是什么？

由于本实验是关于TCP而非HTTP，因此，让我们更改Wireshark的“捕获数据包列表”窗口，使其显示包含HTTP消息的TCP段的信息，而不是显示HTTP消息。要让Wireshark执行该操作，请选择分析->已启用协议。然后取消选中HTTP复选框并选择确定。此时您应该会看到一个Wireshark窗口，其外观如下所示：

[3] 我们所说的“标注”具体指什么？如果是纸质版作业，请在答案所在位置用彩色笔圈出并添加文字说明，注明每个答案的具体出处。如果是电子版作业，同样建议圈出重点内容并添加批注。



这就是我们要找的——在你的计算机和gaia.cs.umass.edu之间发送的一系列TCP数据包。我们将使用你捕获的数据包跟踪（和/或http://gaia.cs.umass.edu/wireshark-labs/wiresharktraces.zip中的tcp-ethereal-trace-1数据包跟踪；参见前面的脚注）来研究本实验室剩余部分中的TCP行为。

3. TCP基础知识

回答有关TCP段的下列问题：

4. 用于在客户端计算机和gaia.cs.umass.edu之间启动TCP连接的TCP SYN段的序列号是什么？该段中标识该段为SYN段的是什么？
5. gaia.cs.umass.edu发送给客户端计算机的SYNACK段的序列号是多少？SYNACK段中的确认字段值是多少？gaia.cs.umass.edu是如何确定该值的？该段中标识为SYNACK段的字段是什么？
6. 包含HTTP POST命令的TCP段的序列号是什么？请注意，为了找到POST命令，您需要深入到Wireshark窗口底部的包内容字段中，寻找一个在DATA字段中有“POST”的段。
7. 将包含HTTP POST的TCP数据段视为TCP连接中的第一个数据段。前六个数据段的序列号是什么？

TCP连接（包含HTTP POST请求的报文段）？每个报文段分别在什么时间发送？每个报文段的确认信息又是在何时收到的？根据各报文段发送时间与确认接收时间的差异，计算六个报文段的往返时间（RTT）值。在收到每个确认信息后，根据第3.5.3节（教材第242页）的估算RTT值，计算出后续所有报文段的往返时间。假设第一个报文段的估算RTT值等于其实际测量的往返时间（RTT），之后所有后续报文段的往返时间均通过教材第242页的估算RTT公式进行计算。

注意：Wireshark有一个很好的功能，允许您绘制每个发送的TCP分段的往返时间。在“捕获的数据包列表”窗口中选择一个从客户端发送到gaia.cs.umass.edu服务器的TCP分段。然后选择：统计->TCP流图->往返时间图。

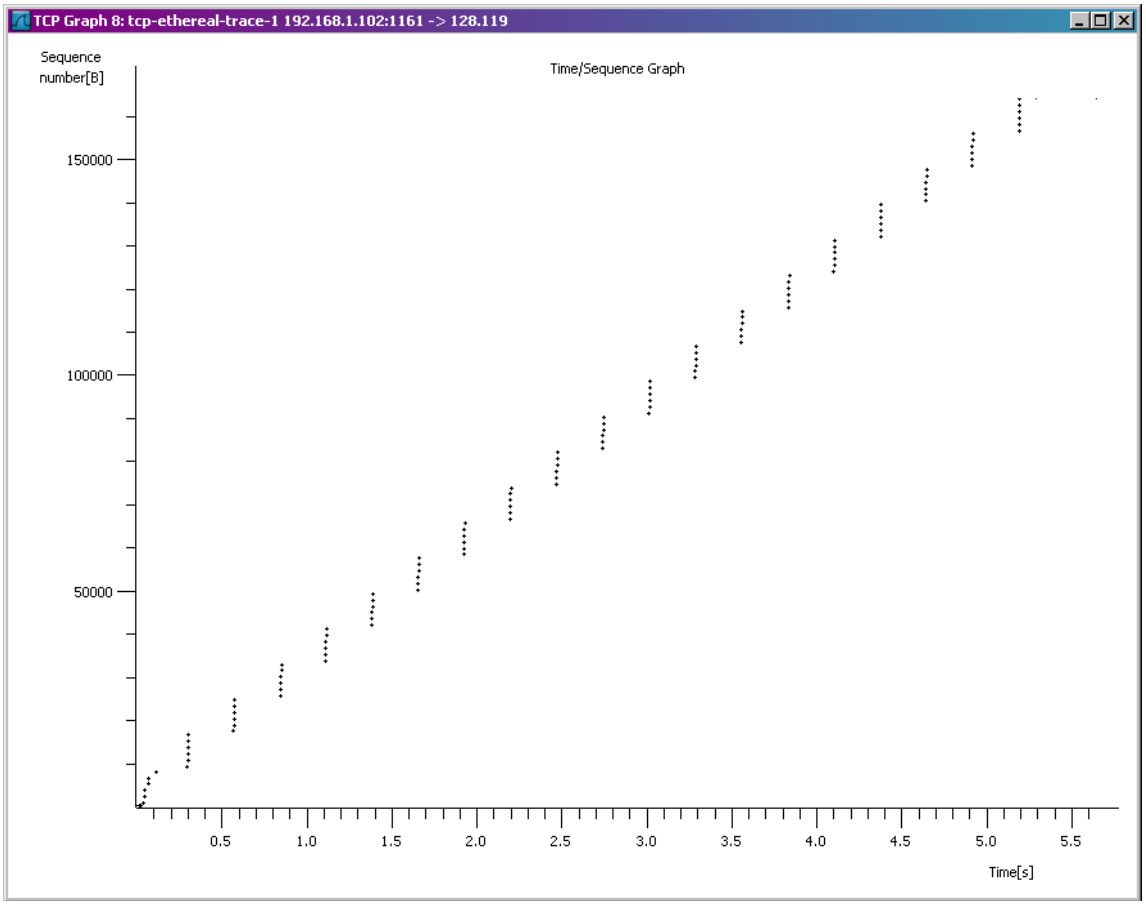
8. 前六个TCP数据段的长度各是多少？[4]
9. 对于整个跟踪，接收端通告的可用缓冲区空间最小量是多少？接收端缓冲区空间不足是否会导致发送端速度变慢？
10. 跟踪文件中是否有任何重新传输的片段？为了回答这个问题，您在跟踪中检查了什么？
11. 接收器在ACK中通常确认多少数据？您是否能确定接收器正在确认每两个接收到的数据段的情况（参见文本第250页的表3.2）。
12. 该TCP连接的吞吐量（单位时间内传输的字节数）是多少？请解释如何计算此值。

[4] 在tcp-ethereal-trace-1跟踪文件中，所有TCP数据段的长度都小于1460字节。这是因为采集该数据的计算机使用以太网卡，其将最大IP数据包长度限制为1500字节（包含40字节的TCP/IP头部数据和1460字节的TCP负载）。这个1500字节是Ethernet协议允许的最大标准长度。如果您的跟踪数据显示TCP数据段长度超过1500字节，且计算机使用以太网连接，说明Wireshark报告了错误的数据段长度——它很可能只会显示一个大块数据段，而非多个小段。实际上，根据您收到的确认应答（ACK），您的计算机很可能正在发送多个较小的数据段。报告的链路段长度不一致是由于以太网驱动程序和Wireshark软件之间的交互作用。如果出现此不一致，我们建议您使用提供的跟踪文件执行本实验。

4. TCP拥塞控制的实际应用

现在，让我们检查从客户端到服务器的每个单位时间发送的数据量。
而不是（乏味地！）从Wireshark窗口的原始数据中计算，我们将使用Wireshark的TCP图形工具之一——时间序列图（Stevens）来绘制数据。

- 在Wireshark的“捕获数据包列表”窗口中选择一个TCP段。
  然后选择菜单：Statistics->TCP Stream Graph-> Time-SequenceGraph（Stevens）。您应该会看到一个与以下图形类似的图表，该图表是根据http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip中捕获的数据包trace tcp-etherealtrace-1（参见前面的脚注）生成的：



此处，每个点代表发送的TCP分段，绘制该分段的序列号与发送时间的关系。请注意，一组堆叠在上方的点表示发送者连续发送的一系列数据包。

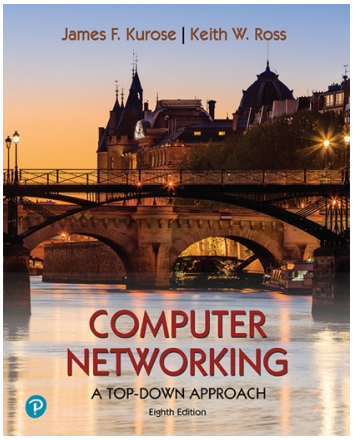
回答以下问题，针对http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip中的数据包跟踪tcp-etherealtrace-1的TCP段

13. 使用时间序列图（Stevens）绘图工具查看从客户端发送到的分段的序列号与时间关系曲线。
    gaia.cs.umass.edu服务器。你能确定TCP的慢启动阶段从哪里开始，到哪里结束，以及拥塞避免机制从哪里接管吗？请谈谈实际测量的数据与我们在课本中研究的TCP理想化行为之间的差异。
14. 回答上述两个问题中的每一个，以说明将文件从您的计算机传输到gaia.cs.umass.edu时所收集的跟踪信息

第二阶段：
服务程序所用的协议

计算机网络：自上而下的方法，第8th 版，J.F. Kurose和K.W. Ross

“告诉我，我忘记。给我看，我记住。让我参与，我理解。”中国谚语



在入门实验中，我们已经通过Wireshark数据包嗅探器初步掌握了网络技术。现在，我们将运用这个强大的工具深入探究网络协议的实际运作机制。本次实验将重点解析HTTP协议的多个核心方面：基础的GET/响应交互、HTTP报文格式解读、大容量HTML文件的下载、含嵌入式对象的HTML文件获取，以及HTTP认证与安全机制。在开始实验前，建议大家先复习教材第2.2节的相关内容。[1]

1. 基本HTTP GET/response交互

让我们通过下载一个非常简单的HTML文件开始HTTP的探索——这个文件非常短，而且不包含任何嵌入对象。请执行以下操作：

1. 启动web浏览器。
2. 按照入门实验中的说明启动Wireshark数据包嗅探器（但尚未开始数据包捕获）。在display-filter-specification窗口中输入“HTTP”（仅字母，不带引号），以便以后在packet-listing窗口中只显示捕获的HTTP消息。（此处我们只对HTTP协议感兴趣，不想看到所有捕获数据包的混乱信息）。
3. 稍等一分钟以上（我们稍后将看到原因），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数据包捕获。

---

[1] *文中引用的图表和章节均以第八版（2020年）《计算机网络：自上而下方法》（J.F.Kurose与K.W.Ross合著，Addison-Wesley/Pearson出版社出版）为依据。*

您的Wireshark窗口应该看起来与图1中所示的窗口类似。如果无法在实时网络连接上运行Wireshark，您可以下载按照上述步骤创建的包跟踪记录。[2]

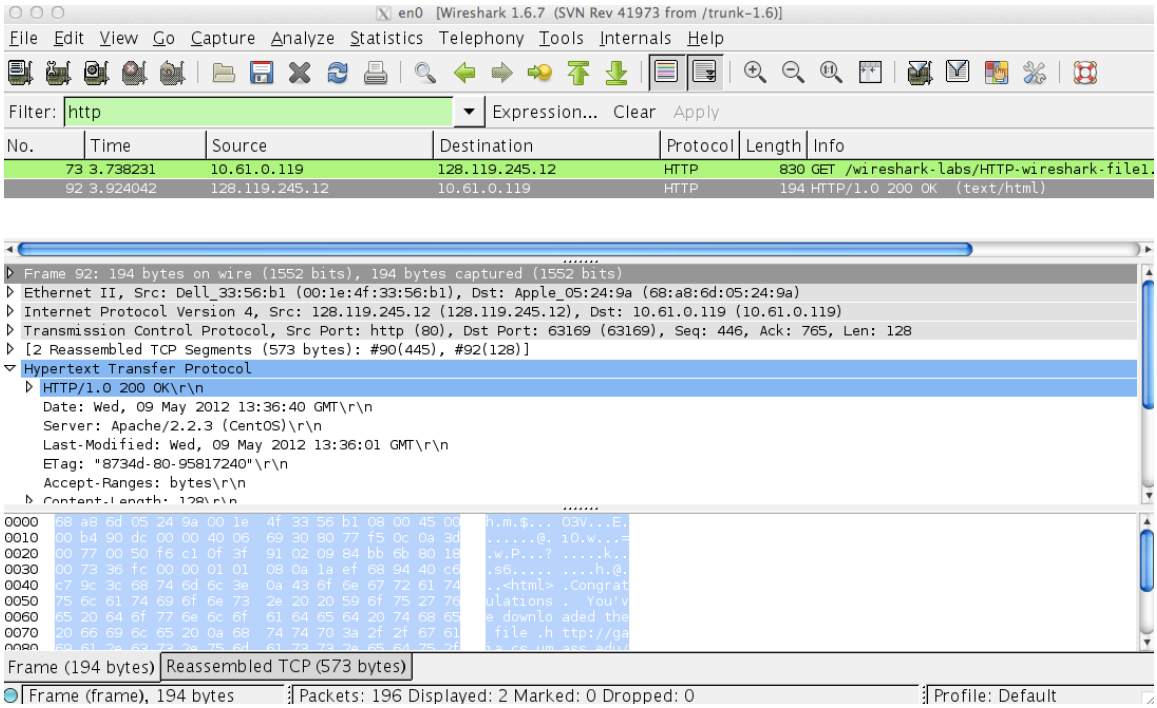


图1：浏览器检索到http://gaia.cs.umass.edu/wireshark-labs/HTTPwireshark-file1.html后的Wireshark显示

图1的示例显示，在数据包列表窗口中捕获了两条HTTP消息：一条是GET请求（浏览器向gaia.cs.umass.edu服务器发送的请求），另一条是服务器返回给浏览器的响应消息。数据包内容窗口则详细展示了所选消息的具体信息（此处为高亮显示的HTTP OK响应）。需要说明的是，由于HTTP消息通过TCP数据段传输，而该数据段又嵌入IP数据报中，最终封装在以太网帧内，因此Wireshark会同步显示帧结构、以太网协议、IP地址和TCP数据包等完整信息。我们需要尽量减少显示的非HTTP数据量（当前研究聚焦于HTTP协议，其他协议将在后续实验中探讨）。因此，请确保帧结构、以太网、IP和TCP信息栏最左侧的方框显示加号或右上三角形（表示隐藏未显示的信息），而HTTP信息栏则需显示减号或左下三角形（表示所有HTTP消息信息均已完整显示）。

---

2 下载zip文件http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip并解压http-ethereal-trace-1。该zip文件中的网络流量记录由作者在计算机上运行Wireshark时采集，具体操作步骤请参考Wireshark实验指南。下载完成后，将文件导入Wireshark后，通过文件菜单选择打开，然后选中http-ethereal-trace-1流量记录文件进行查看。最终显示效果应与图1所示内容基本一致。（在不同的操作系统和不同版本的Wireshark中，Wireshark用户界面的显示方式略有不同）。

（注意：您应该忽略所有针对favicon.ico的HTTP GET请求和响应。如果看到对这个文件的引用，说明浏览器正在自动询问服务器是否拥有一个小型图标文件，该文件应显示在浏览器中当前显示的URL旁边。在本次实验中，我们将忽略对这个烦人的文件的所有引用。）

通过分析HTTP GET请求和响应消息中的信息，回答以下问题。作答时请打印出GET请求和响应消息（具体操作方法可参考Wireshark入门实验教程），并标注在消息中找到答案的具体位置。提交作业时，请在输出内容中标注说明，确保清晰显示答案信息的来源位置（例如：课堂上我们会要求学生用钢笔在纸质试卷上做标记，或在电子文档中用彩色字体标注）。

1. 您的浏览器运行的是HTTP版本1.0或1.1吗？服务器运行的是什么版本的HTTP?
2. 您的浏览器显示它能够接受服务器的哪些语言（如果有）？
3. 您的计算机的IP地址是什么？gaia.cs.umass.edu服务器上?
4. 服务器向浏览器返回的状态代码是什么?
5. 您检索的HTML文件上次在服务器上修改的时间是什么时候?
6. 有多少字节的内容被返回到您的浏览器?
7. 通过检查数据包内容窗口中的原始数据，您是否在数据中看到任何未在数据包列表窗口中显示的标头？如果是，请说出一个。

在回答上述第5题时，您可能会惊讶地发现刚下载的文档，在您访问前一分钟内刚被修改过。这是因为（针对这个特定文件），gaia.cs.umass.edu服务器会将文件的最后修改时间设置为当前时间，并且每分钟执行一次。因此，如果两次访问之间间隔一分钟，文件就会显示最近被修改过，于是浏览器就会下载一个“新”版本的文档。

2. HTTP CONDITIONAL GET/response交互

回想一下，文本的第2.2.5节中提到，大多数web浏览器执行对象缓存，因此在检索HTTP对象时执行条件GET。在执行以下步骤之前，请确保您的浏览器的缓存为空。（要在Firefox中执行此操作，请选择“工具”->“清除最近历史记录”并选中“缓存”框，或者对于Internet Explorer，请选择“工具”->“Internet选项”->“删除文件”；这些操作将从浏览器的缓存中删除缓存文件。）现在执行以下操作：

- 启动您的web浏览器，并确保清除浏览器的缓存，如上文所述。
- 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。

- 再次快速地在浏览器中输入相同的URL（或简单地选择浏览器上的刷新按钮）
- 停止Wireshark数据包捕获，并在显示过滤器规格说明窗口中输入“HTTP”，以便以后在数据包列表窗口中仅显示捕获的HTTP消息。
- （注：如果无法在实时网络连接上运行Wireshark，可以使用http-ethereal-trace-2数据包跟踪来回答以下问题；请参见脚注1。此跟踪文件是在作者的一台计算机上执行上述步骤时收集的。）

回答下列问题

8. 检查浏览器对第一个HTTP GET请求的内容
   服务器。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了文件的内容？如何判断?
10. 现在检查浏览器向服务器发送的第二个HTTP GET请求的内容。您是否在HTTP GET中看到“IF-MODIFIED-SINCE：”行？如果是，那么“IF-MODIFIED-SINCE：”标头之后有哪些信息?
11. 服务器响应第二个HTTP GET时返回的HTTP状态代码和短语是什么？服务器是否明确返回了文件的内容？请解释。

3. 检索长文档

在我们之前的示例中，检索到的文档都是简单且简短的HTML文件。接下来，让我们看看下载一个较长的HTML文件会发生什么。请执行以下操作：

- 启动您的web浏览器，并确保清除浏览器的缓存，如上文所述。
- 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
  你的浏览器应该显示相当长的美国权利法案。
- 停止Wireshark数据包捕获，并在显示过滤器规格说明窗口中输入“HTTP”，以便仅显示捕获的HTTP消息。
- （注：如果无法在实时网络连接上运行Wireshark，可以使用http-ethereal-trace-3数据包跟踪来回答以下问题；请参见脚注1。此跟踪文件是在作者的一台计算机上执行上述步骤时收集的。）

在数据包列表窗口中，您应该看到HTTP GET消息，然后是针对HTTP GET请求的多数据包TCP响应。这个多数据包响应值得稍加解释。回顾第2.2节（参见正文图2.9）可知，HTTP响应消息由状态行、头部信息、空行和实体内容四部分组成。在本次HTTP GET请求中，

在响应中，实体主体即为完整的请求HTML文件。以当前案例为例，该HTML文件长度较长，其4500字节的体积已超出单个TCP数据包的承载能力。因此，系统会将整个HTTP响应拆分为多个数据块，每个数据块分别封装在独立的TCP段中（参见正文图1.24）。最新版Wireshark会将每个TCP段单独标记为独立数据包，并通过显示信息栏中的“重组PDU的TCP段”标识，明确指出原始HTTP响应被拆分为多个TCP数据包的事实。早期版本则使用“续传”提示来表示HTTP消息内容被分割成多个TCP段。需要特别说明的是：HTTP协议本身并不包含“续传”这类消息！

回答下列问题

12. 您的浏览器发送了多少个HTTP GET请求消息？跟踪记录中包含帐单或权利的GET消息的哪个数据包编号？
13. 跟踪中包含相关状态代码和短语的哪个数据包编号对HTTP GET请求的响应如何？
14. 响应中的状态代码和短语是什么？
15. 需要多少个包含数据的TCP段才能传输单个HTTP回应和《权利法案》的文本？

4. 带有嵌入对象的HTML文档

现在我们已经看到Wireshark如何显示捕获的大型HTML文件的数据包流量，我们可以看看当浏览器下载一个包含嵌入对象的文件时会发生什么，即一个包含其他对象（在下面的例子中是图像文件）的文件，这些对象存储在另一个服务器上。

执行以下操作：

- 启动您的web浏览器，并确保清除浏览器的缓存，如上文所述。
- 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
  您的浏览器应当显示一个包含两张图片的简短HTML文件。这两个图片在基础HTML文件中被引用，也就是说，图片本身并不包含在HTML文件中，而是通过下载的HTML文件中的图片URL来呈现。正如教材中所述，您的浏览器需要从指定网站获取这些标志。我们的出版商标志是从gaia.cs.umass.edu网站获取的。我们第5th 版的封面图片（这是我们最喜欢的封面之一）存储在caite.cs.umass.edu服务器上。
  （cs.umass.edu内部有两个不同的web服务器）。
- 停止Wireshark数据包捕获，并在显示过滤器规格说明窗口中输入“HTTP”，以便仅显示捕获的HTTP消息。

- （注：如果无法在实时网络连接上运行Wireshark，可以使用http-ethereal-trace-4数据包跟踪来回答以下问题；请参见脚注1。此跟踪文件是在作者的一台计算机上执行上述步骤时收集的。）

回答下列问题

16. 您的浏览器发送了多少个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连续下载了这两个图像，还是并行从两个网站下载的？请解释。

5 HTTP验证

最后，让我们尝试访问一个需要密码保护的网站，并检查该网站交换的HTTP消息序列。URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html 是受密码保护的。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
受密码保护的站点。请执行以下操作：

- 请确保按照上文所述清除浏览器的缓存，然后关闭浏览器。接着启动浏览器
- 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
  在弹出框中输入所需的用户名和密码。
- 停止Wireshark数据包捕获，并在显示过滤器规格说明窗口中输入“HTTP”，以便以后在数据包列表窗口中仅显示捕获的HTTP消息。
- （注：如果无法在实时网络连接上运行Wireshark，可以使用http-ethereal-trace-5数据包跟踪来回答以下问题；请参见脚注2。此跟踪文件是在作者的一台计算机上执行上述步骤时收集的。）

现在让我们来分析Wireshark的输出结果。建议您先通过查阅http://frontier.userland.com/stories/storyReader$2159网站上《HTTP访问认证框架》这份通俗易懂的资料，系统学习HTTP认证的相关知识。

回答下列问题

18. 服务器对浏览器发出的初始HTTP GET消息做出什么响应（状态代码和短语）？
19. 当浏览器第二次发送HTTP GET消息时，HTTP GET消息中会包含什么新字段?

您输入的用户名（wireshark-students）和密码（network）编码在字符串（d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=）后面

在客户端HTTP GET报文中的“Authorization：Basic”头部显示，虽然看似你的用户名和密码经过加密处理，但实际上它们只是采用Base64编码格式。请注意：这些信息并未加密！要验证这一点，请访问http://www.motobit.com/util/base64-decoder-encoder.asp网站输入Base64编码字符串d2lyZXNoYXJrLXN0dWRlbnRz进行解码。瞧！你已成功将Base64编码转换为ASCII编码，此时就能看到你的用户名了！若想查看密码，请输入字符串剩余部分Om5ldHdvcms=并点击解码按钮。由于任何人都可以下载类似Wireshark的工具来嗅探网络适配器传输的数据包（不仅限于自己的），而且任何人都能将Base64转换为ASCII（你刚才就是这么操作的！），由此可见：除非采取额外防护措施，否则网站上的简单密码根本无法保障安全。

不要害怕！正如我们将在第八章看到的，有一些方法可以使WWW访问更加安全。但是，显然我们需要一些超越基本HTTP身份验证框架的东西！