

## 最终检查点：将所有内容整合在一起

到期的：下课（12 月 3 日下午 5 点）

## 0 合作政策

编程作业必须是你自己的作品：您必须编写您提交的编程作业的所有代码，除了我们作为作业的一部分提供给您的代码。请不要从 Stack Overflow、GitHub 或其他来源复制粘贴代码。如果您根据在 Web 或其他地方找到的示例编写自己的代码，请在您提交的源代码的注释中引用 URL。

与他人合作：在本次作业中，您将与实验室伙伴合作，建立一个网络连接您以前的实验室和他们的实验室。与以往的实验室一样，您不得向他们或任何其他人展示您的代码，不得查看任何其他人的代码，也不得查看前几年的解决方案。我们确实会请您与您的实验室伙伴或其他学生讨论您的代码在做什么（这可能需要进行调试）。如果您与其他学生讨论作业，请在您提交的源代码或您的写作中的评论中提及他们的名字。请参阅课程管理讲义了解更多详细信息，如果不清楚，请在 Piazza 上询问。

广场：请随意在 Piazza 上提问，但请不要发布任何源代码。

## 1 概述

到本课程的这个阶段，您已经实现了互联网基础设施的很大一部分。从实验室 0（可靠字节流）到实验室 1-4（传输控制协议）、实验室 5（IP/以太网网络接口）和实验室 6（IP 路由器），您已经完成了大量编码工作！

在这个实验室里，你不需要做任何编码（假设你之前的实验室运行良好）。相反，为了完成你的成就，你需要使用你之前的所有实验都用来创建一个真实的网络，其中包括你的网络堆栈（主机和路由器）与班上另一名学生实现的网络堆栈对话。

本实验以两人一组的方式进行。您需要与实验伙伴（班上的另一名学生）一起工作。请利用实验课寻找实验伙伴，如果您不能参加实验课，则可以利用 Piazza。如有必要，同一名学生可以多次担任“实验伙伴”。

## 2 入门

1. 确保你已经提交了实验 6 的所有解决方案。请不要修改自由海绵目录，或 webget.cc. (请不要

添加您的代码所依赖的额外文件。)

2. 在实验室作业存储库中，运行实验室作业的最新版本。

`git 获取` `检索`

3. 运行以下命令下载实验 7 的代码

`git 合并 origin/lab7-startercode`。

4. 在您的建造目录，编译源代码：

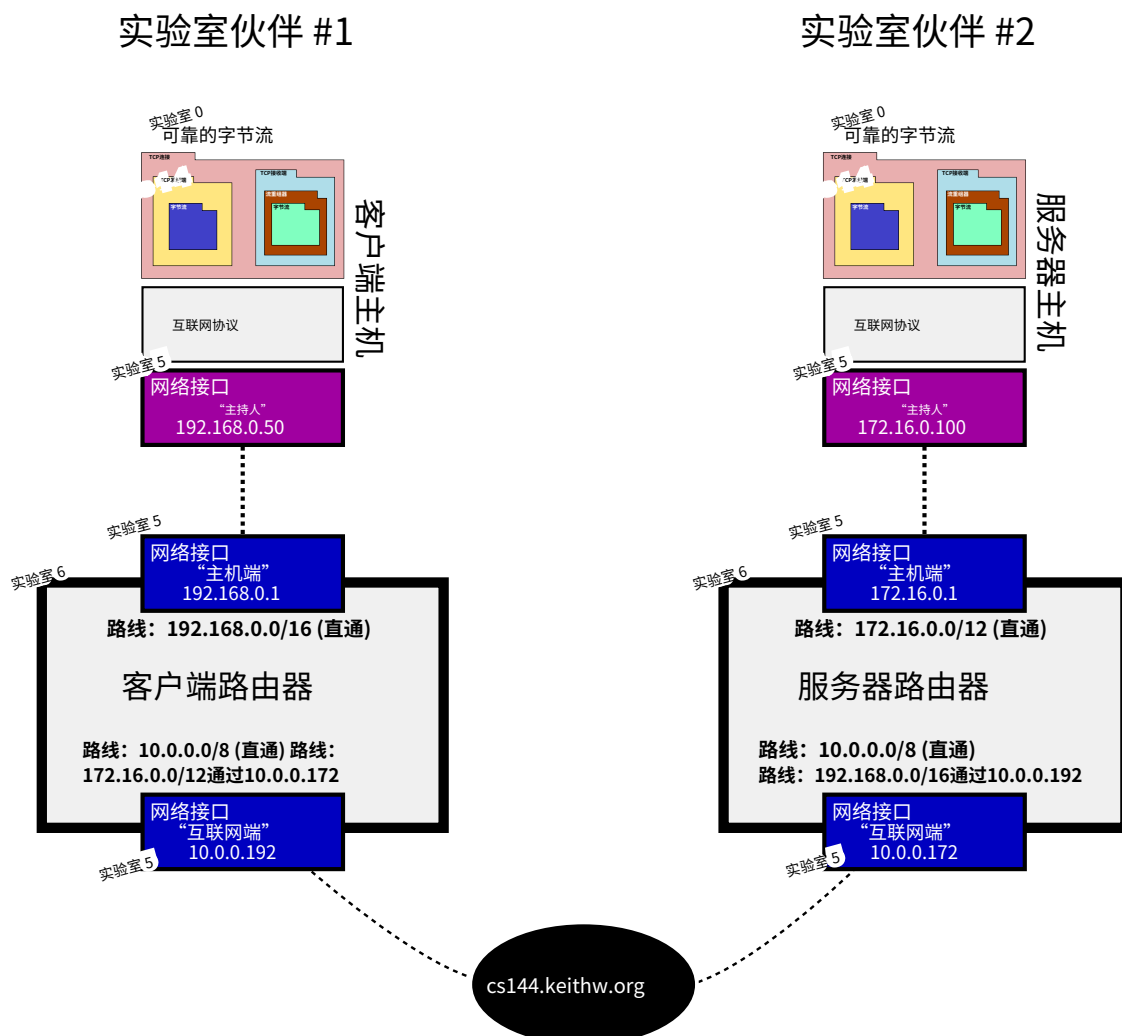
`制作-j4` 在编译时使用四个处理器)。

`制作` (例如，你可以运行

5. 外面建造目录，打开并开始编辑writeups/lab7.md fi这是您的实验室报告模板，将包含在您的提交内容中。

### 3. 网络

在本实验中，您将创建一个真实的网络，该网络将您自己实现的网络堆栈与班上其他学生实现的网络堆栈相结合。你们每个人都将贡献一个主机（包括您的可靠字节流、您的 TCP 实现和您的 NetworkInterface）和一个路由器：



因为您或您的实验室伙伴很可能位于网络地址转换器后面，所以双方之间的网络连接将通过中继服务器（cs144.keithw.org）流动。

我们已经将您的代码整合到一个新应用程序中，您可以在构建/应用程序/lab7。运行它的步骤如下：

1. 在与实验室伙伴一起执行这些步骤之前，亲自尝试一下。通过在虚拟机上使用两个不同的窗口或终端，您可以同时扮演客户端和服务器的角色。这样，您的网络将包含两个代码副本（主机和路由器），它们会相互通信。这比与陌生人交谈更容易调试！

一旦这些步骤对你有效，然后和实验室伙伴一起尝试。决定你们俩中谁将充当“客户端”，谁将充当“服务器”。一旦成功，你们可以随时交换角色并重试。

2. 要使用中继，请选择随机偶数介于 1024 和 64000 之间。这标识了您的实验室组，并且需要与同时工作的任何其他实验室组不同，因此请选择一个随机数。并且它必须是偶数。对于其余这些示例，我们假设您选择了“3000”：

3. 从“build”目录，“server”学生运行：

```
./apps/lab7 服务器 cs144.keithw.org 3000
```

（将“3000”替换为您的实际数字）。

如果一切顺利，“服务器”将打印如下输出：

```
$ ./apps/lab7 服务器 cs144.keithw.org 3000
```

```
调试：网络接口具有以太网地址 02:00:00:5e:61:17 和 IP 地址 172.16.0.1 调试：网络接口具有以太网地址
02:00:00:cd:e7:e0 和 IP 地址 10.0.0.172 调试：在接口 0 上添加路由 172.16.0.0/12 => (直接)
```

```
调试：在接口 1 上添加路由 10.0.0.0/8 => (直接) 调试：在接口 1 上添加路由
```

```
192.168.0.0/16 => 10.0.0.192
```

```
调试：网络接口具有以太网地址 5a: 75: 4e: 8b: 20:00 和 IP 地址 172.16.0.100 调试：正在监听传入连接...
```

4. 从“build”目录中，“client”学生运行：

```
./apps/lab7 客户端 cs144.keithw.org 3001
```

（将“3001”替换为您的随机数，再加一）。如果一切顺利，“客户

端”将打印如下输出：

```
$ ./apps/lab7 客户端 cs144.keithw.org 4001
```

```
调试：网络接口具有以太网地址 02:00:00:41:c7:5b 和 IP 地址 192.168.0.1 调试：网络接口具有以太网地址
02:00:00:e6:66:d9 和 IP 地址 10.0.0.192 调试：在接口 0 上添加路由 192.168.0.0/16 => (直接)
```

```
调试：在接口 1 上添加路由 10.0.0.0/8 => (直接) 调试：在接口 1 上添加路由
```

```
由 172.16.0.0/12 => 10.0.0.172
```

```
调试：网络接口具有以太网地址 26:05:12:4a:8a:c9 和 IP 地址 192.168.0.50 调试：从 192.168.0.50:57005 连接...
```

```
调试：连接到 172.16.0.100:1234... 成功连接到
```

```
172.16.0.100:1234。
```

并且“服务器”将再打印一行：

来自 192.168.0.50:57005 的新连接。

5.如果你看到预期的输出，一切顺利——两台计算机已成功交换了 TCP 握手！

(a) 为自己鼓掌（保持适当的社交距离）——这是你应得的！

(b) 现在该交换数据了。在其中一个窗口中键入，然后查看另一个窗口中出现的输出。尝试反向键入。

(c) 尝试结束连接。完成后键入 - D。当每一方都这样做时，它将结束该方向的出站字节流的输入，同时继续接收传入数据，直到对等方结束其自己的字节流。验证是否发生这种情况。

(d) 当双方都结束了他们的字节流，并且一方已经停留了几秒钟后，两个程序都应该正常退出。

6.如果你没有看到预期的输出，可能是时候打开“调试模式”了。运行“lab7”程序并添加一个附加参数：在命令行末尾附加“debug”。这将打印出正在交换的每个以太网帧，并且您可以看到所有 ARP 和 TCP/IP 帧。

7.一旦您在自己的计算机上的两个窗口之间建立了网络，就可以与实验室伙伴一起尝试相同的步骤（以及他们自己的实现）。

## 4 发送文件

一旦看起来您可以进行基本的对话，请尝试通过网络发送文件。同样，您可以自己尝试，如果一切顺利，然后与实验室伙伴一起尝试。方法如下：

到写一个一兆字节的随机文件到“/tmp/big.txt”：

```
dd if=/dev/urandom bs=1M count=1 of=/tmp/big.txt
```

让服务器在接受传入连接后立即发送文件 n：

```
./apps/lab7 服务器 cs144.keithw.org 偶数 < /tmp/大.txt
```

让客户端关闭其出站流并下载文件：

```
</dev/null ./apps/lab7 客户端 cs144.keithw.org 奇数 > /tmp/big-received.txt
```

比较两个文件并 m 确保它们相同：

```
sha256sum /tmp/big.txt 或者 sha256sum /tmp/big-received.txt
```

如果 SHA-256 哈希值匹配，您几乎可以肯定文件已正确传输。

## 5 发挥创造力

如果一切顺利，我们鼓励您做一些有创意的事情，在您的文章中加入一些有趣的内容，以获得一些（象征性的）额外学分。请随意修改 lab7.cc 程序。您可以创建一个更复杂的网络，同时涉及更多学生，或者做一些我们没有预料到的事情。（要清楚：这根本不是必需的。）

## 6 提交

1. 撰写报告 writeups/lab7.md。该文件应为大约 30 到 70 行的文档，每行不超过 80 个字符，以便于阅读。报告应包含以下部分：

- 独奏部分

- 您的实现是否成功地开始和结束与其自身的另一个副本的对话？
- 它是否成功传输了一个一兆字节的文件，并且接收时内容完全相同？
- 请描述需要进行哪些代码更改（如果有）才能通过这些步骤。

- 团体部分

- 您的实验室合作伙伴是谁（以及他们的 SUNet ID 是什么，例如温斯坦（winstein））？
- 您的实现是否成功地开始和结束彼此之间的对话（每个实现充当“客户端”或“服务器”）？
- 您是否成功在两个实现之间传输了 1MB 的文件，并且接收时内容相同？
- 请描述您或您的实验室伙伴需要进行哪些代码更改（如果有）才能通过这些步骤。

- 创意部分

- 如果您为我们的“创意挑战”做了任何贡献，请夸耀一下！

2. 如果您确实需要更改源代码，请仅对 进行更改。时 和 。cc fi 位于顶层的 libspoonage。在这些文件中，请根据需要随意添加私人成员，但请不要更改公共接口。

3. 请不要添加额外的文件——自动评分器不会查看它们，并且您的代码可能无法编译。

4. 在提交任何作业之前，请按顺序运行以下内容：

- (一)  （规范化编码风格）
- (二)  （检查未提交的更改 - 如果有，请提交！）

(三) 制作 (确保代码可以编译)

(四) 进行检查 (要运行完整的测试套件——请确保你没有破坏任何测试)

5. 请填写完成该作业所花费的小时数以及其他评论。
6. 准备提交时, 请按照以下说明操作<https://cs144.github.io/提交>。提交前请确保您已完成所有想要完成的工作。
7. 如果实验课上出现任何问题, 请尽快告知课程工作人员, 或在 EdStem 上发布问题。