

实验室检查点 6: 构建 IP 路由器

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: 您不得向其他人展示您的代码, 不得查看其他人的代码, 也不得查看前几年的解决方案。您可以与其他学生讨论作业, 但不得抄袭任何人的代码。如果您与其他学生讨论作业, 请在您提交的源代码的评论中注明他们的名字。请参阅课程管理讲义了解更多详细信息, 如果有任何不清楚的地方, 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

在本周的实验室检查点中, 你将在现有网络接口。路由器有一些网络接口, 并可以在其中任何一个接口上接收互联网数据报。路由器是根据路由表: 一组规则, 告诉路由器, 对于任何给定的数据报,

- 通过什么接口发送出去
- 下一跳的 IP 地址

你的任务是实现一个路由器, 它可以为任何给定的数据报找出这两个东西。(你不需要实现路由表, 例如 RIP、OSPF、BGP 或 SDN 控制器——只是关注路由表。

你的路由器实现将使用 Sponge 库和一个新的路由器课程, 以及在模拟网络中检查路由器功能的测试。实验 6 以您实施的接口来自实验室 5, 但确实不是使用你在实验 0-4 中实现的 TCP 堆栈。IP 路由器不必了解任何有关 TCP、ARP 或以太网 (仅 IP) 的知识。我们预计你的实现将需要大约 25-30 行代码。

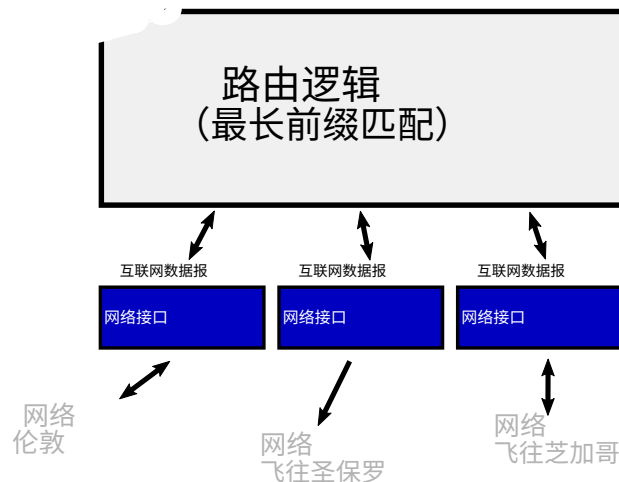


图 1: 路由器包含多个网络接口, 可以在其中任何一个接口上接收 IP 数据报。路由器会将其收到的任何数据报转发到下一跳, 并在适当的出站接口上转发。路由表会告诉路由器如何做出此决定。

2 入门

1. 确保你已经提交了实验 5 的所有解决方案。请不要修改自由海绵目录, 或webget.cc. (并且请不要添加您的代码所依赖的额外文件。) 否则, 您可能会在合并 Lab 6 启动代码时遇到麻烦。
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 5 的起始代码 `git 合并 origin/lab6-startercode`。
4. 在您 的建造目录, 编译源代码: `制作` (例如, 你可以运行 `制作-j4` 在编译时使用四个处理器)。
5. 外面建造目录, 打开并开始编辑writeups/lab6.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 实现路由器

在本实验中, 您将实现路由器可以进行以下操作的类:

- 跟踪路由表 (转发规则或路由的列表), 以及
- 转发收到的每个数据报:
 - 到正确的下一跳
 - 正确的传出网络接口。

您的实施将被添加到[路由器](#)和[路由器](#)骨架文件。在开始编码之前，请查看[新的文档路由器班级](#)。

以下是您将实现的两种方法以及我们对每种方法的期望：

空白添加路线（**常量**`uint32_t`路线前缀，
常量`uint8_t`前缀长度，**常量**选修的<地
 址>下一跳，**常量**`size_t`接口编号）；

此方法将路由添加到路由表。您需要在路由器类来存储此信息。此方法需要做的就是保存路线以供以后使用。

路线的各个部分代表什么？

路由是一种“匹配动作”规则：它告诉路由器*如果*数据报发往特定网络（某个 IP 地址范围），并且*如果*该路线被选为最具体的匹配路线，*然后*路由器应该将数据报转发到特定接口上的特定下一跳。

“匹配”：数据报是否发往该网络？这路由前缀和 前缀长度一起指定可能包含数据报目的地的 IP 地址范围（网络）。路由前缀是一个 32 位数字 IP 地址。前缀长度是 0 到 32 之间的数字（含）；它告诉路由器多少个最高有效位的路由前缀非常重要。例如，要表达网络“18.47.0.0/16”的路由（这匹配任何32位IP地址，前两个字节为18和47），路由前缀将是 305070080 ($18 \times 2^{24} + 47 \times 2^{16}$)，以及前缀长度将是 16。任何发往“18.47.xy”的数据报都将匹配。

“动作”：如果路线匹配并且被选中，该做什么。如果路由器 *直接地* 连接到相关网络，下一跳将为空选修的。在这种情况下，下一跳是数据报的目标地址。但如果路由器通过其他路由器连接到相关网络，则下一跳将包含路径上下一个路由器的 IP 地址。接口号给出路由器的索引网络接口那当然用于发送数据报 m 到下一跳。您可以使用

界面（接口号） 方法。

空白路由一数据报 (互联网数据报&dgram);

这是轮胎与道路接触的地方。此方法需要路由一数据报到达下一跳，从适当的接口发出。它需要实现 IP 路由器的“最长前缀匹配”逻辑来找到最好的路线。这意味着：

- 这路由器搜索路由表以查找与数据报的目标地址匹配的路由。这里的“匹配”是指最重要的前缀长度目标地址的位与最高有效位相同前缀长度位的 路由前缀。
- 在匹配的路由中，路由器选择具有最大的价值 前缀长度。这是最长前缀匹配路线。
- 如果没有匹配的路由，路由器将丢弃数据报。
- 路由器减少数据报的 TTL（生存时间）。如果 TTL 已经为零，或在减少后达到零，则路由器应丢弃数据报。
- 哦否则，路由器发送修改后的数据包 内存在适当的接口（
界面（接口号）.发送数据报()）到适当的下一跳。

互联网的设计在这里有一个美妙之处（或者至少是一个成功的抽象）：路由器从不考虑 TCP、ARP 或以太网帧。路由器甚至不知道链路层是什么样子。路由器只考虑互联网数据报，只通过网络接口抽象。当遇到诸如“如何解析链路层地址？”或“链路层是否有自己不同于 IP 的寻址方案？”或“链路层帧的格式是什么？”或“数据报的有效负载是什么意思？”等问题时，路由器根本不关心。

4 测试

您可以通过运行来测试您的实现
在特定的模拟网络中，如图所示2。

[检查 lab6](#) .这将测试你的路由器

5 问答

- 我应该使用什么数据结构来记录路由表？
由你决定！但不必太过疯狂。每个数据报都需要哦（否）工作，其中否是路由表中条目的数量。如果你想做更高效的事情，我们鼓励你获得一个可行的实现

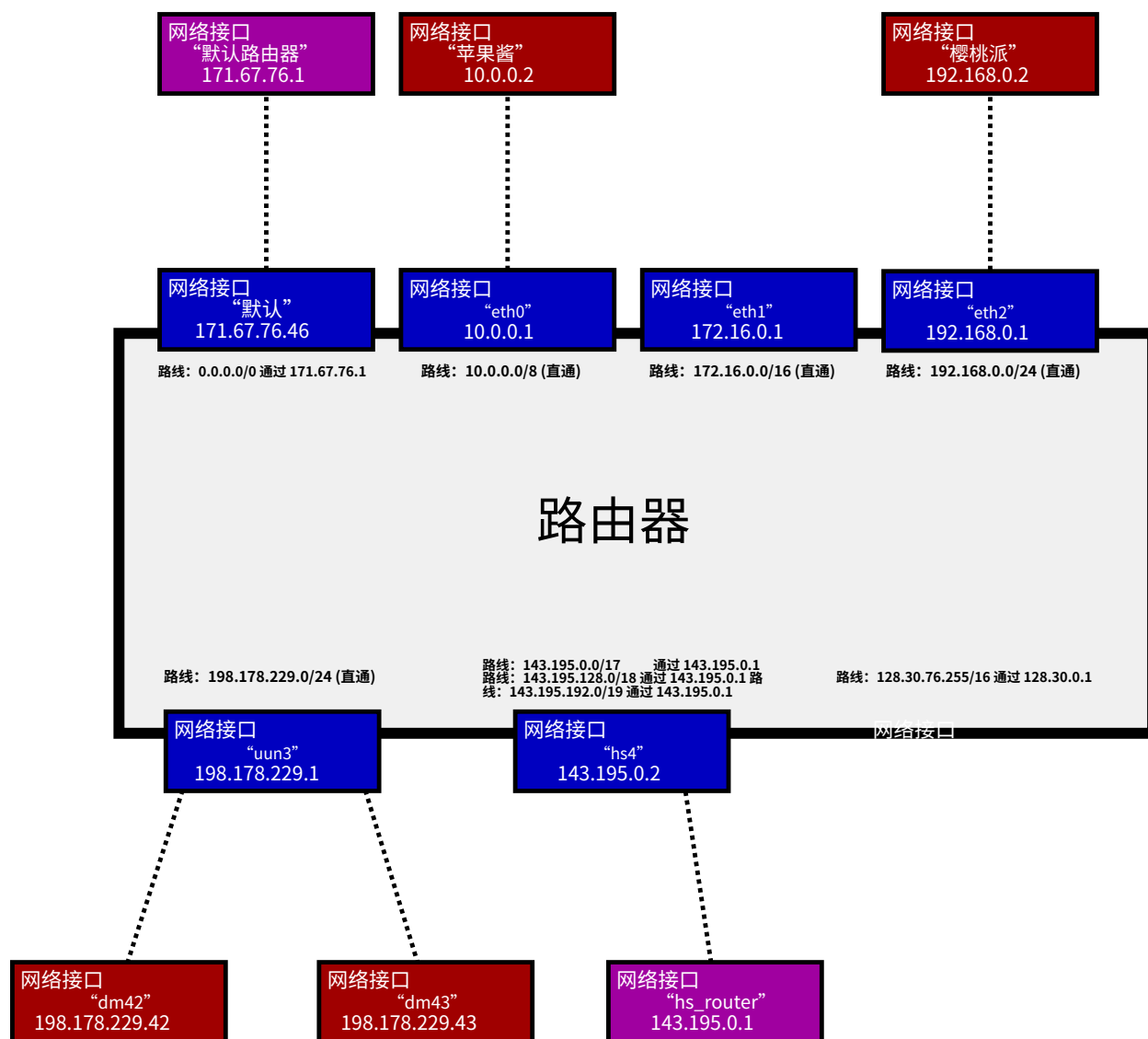


图 2: 所用的模拟测试网络应用程序/网络模拟器工具, 也由 [检查 lab6](#)。 (有趣的事实: 尤恩网络是 [大卫·马齐埃 \(David Mazières\)](#) 的互联网片段, 1993年分配。这whois工具或链接的网站可用于查找谁控制每个 IP 地址分配。

在优化之前，请首先仔细记录和评论您选择实施的任何内容。

- 如何将以地址对象形式出现的 IP 地址转换为可以写入 ARP 消息的原始 32 位整数？

使用地址::ipv4 数字()方法。

- 如何将原始 32 位整数形式的 IP 地址转换为地址对象？

使用地址::来自 ipv4 数字()方法。

- 如何比较最重要的否位（其中 $0 \leq \text{否} \leq 32$ ）一个 32 位 IP 地址，其中最高有效位否另一个 32 位 IP 地址的位数？

这可能是这项作业中最“棘手”的部分——让逻辑正确。也许值得用 C++ 编写一个小的测试程序（一个简短的独立程序）或向 Sponge 添加测试来验证您对相关 C++ 运算符的理解并仔细检查您的逻辑。

回想一下，在 C 和 C++ 中，它可以产生未定义的行为将 32 位整数移位 32 位。

在 `cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo` 并使用清理程序编译你的代码
在 `cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo` 尝试捕捉任何未定义的行为
在您提交之前，请在您的代码中输入

您可以通过在建造目录。

`./apps/网络模拟器` 从

- 如果路由器没有到达目的地的路由，或者 TTL 为零，它不应该向数据报的源发送回 ICMP 错误消息吗？

在现实生活中，这确实会有所帮助。但在本实验中没有必要这样做 — 丢弃数据报就足够了。（即使在现实世界中，在这些情况下，也并非每个路由器都会将 ICMP 消息发送回源。）

- 如何运行该实验室的测试套件？

`检查 lab6` (两个测试)。或者你可以使用以下方法运行整个测试套件 `进行检查`
(161 次测试)。

- 此 PDF 版本发布后，我可以在哪里阅读更多常见问题解答？

请查看网站 (<https://cs144.github.io/lab常见问题.html>) 和 Piazza 经常光顾。

6 提交

1. 在您提交的作品中，请仅对 `进行更改` 和 `cc fi` 位于顶层的 `libsponge`。在这些文件中，请根据需要随意添加私有成员，但请不要更改民众任何类的接口。

2. 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。

3. 在提交任何作业之前，请按顺序运行以下内容：

- (一) 制作格式 (规范化编码风格)
- (二) git 状态 (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) 检查 lab6 (确保自动化测试通过)

4. 撰写报告writeups/lab6.md。该文件应为大约 20 到 50 行的文档，每行不超过 80 个字符，以方便阅读。报告应包含以下部分：

(一个) 程序结构和设计。描述代码中体现的高级结构和设计选择。您无需详细讨论从起始代码中继承的内容。利用这个机会突出重要的设计方面，并更详细地介绍这些方面，以便您的评分助教理解。强烈建议您使用小标题和大纲使本文尽可能易于阅读。请不要简单地将您的程序翻译成一段英文。

(二) 实施挑战。描述您发现最麻烦的代码部分并解释原因。反思您如何克服这些挑战以及是什么帮助您最终理解了让您感到困难的概念。您如何尝试确保您的代码保持您的假设、不变量和先决条件，以及在哪些方面您觉得这很容易或很难？您如何调试和测试您的代码？

(三) 餘下的错误。尽可能指出并解释代码中存在的任何错误（或未处理的边缘情况）。

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。