

# CS144: 计算机网络导论 2025年秋季

## 检查点 7 草案: 构建互联网 + 创意项目!

截止日期: 课程结束时 (12月5日, 晚上11:59)

## 0 合作政策

合作政策: 检查点7要求你与尽可能多的组员合作——你需要连接起来构建一个课程"互联网", 也许还需要一起调试。强烈鼓励讨论数据包和想法! 但与之前一样, 我们坚持明确的界限: 请不要查看其他学生的代码或向任何人 (包括聊天机器人或以前的学生) 请求或接收代码。与之前的检查点一样, 请在你的报告中完全披露合作者、LLM提示或任何灰色地带——这保护你。

## 1 概述

到目前为止, 你已经实现了互联网基础设施的大部分。从检查点0 (可靠字节流), 到检查点1-3 (传输控制协议), 检查点5 (IP/以太网网络接口) 和检查点6 (IP路由器), 你已经掌握了互联网的许多核心基础设施!

为了庆祝你的成就, 你将使用所有之前的实验成果来设计和运行一个"检查点7顶点互联网", 其中包括你的网络栈 (主机和路由器) 与课程中其他学生实现的网络栈通信。请使用实验课来协调和连接, 或者如果你无法参加, 使用EdStem来协调。

## 2 开始

确保已提交所有解决方案。请不要修改src目录顶层之外的任何文件, 也不要修改webget.cc和ip\_raw.cc。否则你可能无法合并检查点7的起始代码。

在实验作业仓库内, 运行 `git fetch --all` 以获取实验作业的最新版本。

通过运行 `git merge origin/check7-startercode` 下载检查点7的起始代码。

确保构建系统已正确设置: `cmake -S . -B build`

记住, 如果遇到问题, 你可以构建"清理" (错误检查) 版本的应用程序: `cmake -S . -B build -DSANITIZED_APPS=True`

编译源代码: `cmake --build build`

提醒: 请在工作时在本地Git仓库中进行频繁的小提交。如果你需要帮助确保做对了, 请向同学或教学人员寻求帮助。你可以使用`git log`命令查看你的Git历史。

## 3 阅读并理解起始代码

作为检查点7起始代码的一部分, 我们给了你两个新的"应用程序", 它们使用你的ByteStream、TCP实现、NetworkInterface和Router。

`apps/tcp_eth_udp.cc` 实现了一个"全栈"端点: 你的TCP实现创建TCP段, 我们的代码将其封装在互联网数据报中。这些被交给你的NetworkInterface, 后者将它们封装在以太网帧中。tcp\_eth\_udp工具将这些以太网帧封装在用户数据报中并通过正常互联网发送。

`apps/fun_router` 通过给你的Router一个命令行界面来实现兼容的IP路由器。命令行标志可以创建接口并向路由器的路由表添加路由。Router的NetworkInterface也发送以太网帧, 封装在用户数据报中, 通过互联网传输。

阅读这两个工具的源代码，努力理解它们如何工作以及它们在做什么。

一个关键点：这两个工具都发送IP-in-Ethernet-in-UDP-in-IP。因此，有两层IP地址在起作用：在最外层通过真实互联网发送的互联网数据报中使用的"物理"地址，以及你将在本检查点中设计的"检查点7顶点互联网"中使用的"虚拟"地址。

当你使用这些工具时，"物理"地址将对应于可以到达另一个学生计算机的真实IP地址。这可以包括你之前在检查点1和3中使用的CS144虚拟专用网络；从这些应用程序的角度来看，那也是"真实"的互联网。

相比之下，"虚拟"地址将是你自己的发明——我们希望你在本检查点中设计一个具有有趣拓扑、尽可能多的路由器路由数据包以及适当路由表的互联网，以便许多学生可以同时通过这个"虚拟"互联网相互通信。

#### 4 在你自己的VM上创建一个微型互联网

首先，我们将在你的计算机内重现历史性的1969年"首次ARPANET连接"（互联网的前身）在UCLA和斯坦福之间。你将创建一个虚拟拓扑如下所示的迷你互联网，通过以下方式在两者之间发送TCP段：

IP地址为50.9.8.7的"斯坦福"上的TCP服务器

IP地址为80.6.5.4的"UCLA"上的TCP客户端

通过一个具有两个接口的路由器：一个用于斯坦福的网络（具有指向50/8范围的路由），一个用于UCLA的网络（80/8）

这些IP数据报将被封装在以太网帧中，以太网帧又在用户数据报中，用户数据报又在"物理"IP数据报中。目前，这些"物理"IP数据报都将留在你的计算机内部，其内部IP地址为127.0.0.1。

完整的"虚拟"和"物理"地址如下所示。花一些时间研究这个图表并理解它——如果一开始看起来很多也没关系。

虚拟拓扑：

```
"stanford" interface: 50.0.0.1
"ucla" interface: 80.0.0.1
route: 50/8 (direct)
route: 80/8 (direct)
tcp_eth_udp (TCP server, local: 50.9.8.7:80) <-- fun_router --> tcp_eth_udp (TCP client, local: 80.6.5.4:80)
client route: 0/0 via 50.0.0.1
server route: 0/0 via 80.0.0.1
```

物理拓扑：

```
"stanford" interface: UDP port 9050, peer: UDP 127.0.0.1:2050
"ucla" interface: UDP port 9080, peer: UDP 127.0.0.1:2080
```

要使用fun\_router和tcp\_eth\_udp应用运行此拓扑，你需要在VM上打开四个终端。

在第一个终端上，运行 `sudo tshark -ni lo -d udp.port==2080,eth 'port 2080'` 来监视飞行中的数据包。-d `udp.port==2080,eth`

告诉tshark将用户数据报有效载荷解释为以太网，可能内部有IP和TCP。

在第二个终端上运行路由器：

```
./build/apps/fun_router \
  interface:stanford:50.0.0.1:9050:127.0.0.1:2050 \
  interface:ucla:80.0.0.1:9080:127.0.0.1:2080 \
  route:50.0.0.0:8:stanford route:80.0.0.0:8:ucla
```

在第三个终端上，运行TCP服务器（在"斯坦福"）：

```
./build/apps/tcp_eth_udp server 2050 50.0.0.1 50.9.8.7:80
```

在第四个终端上，运行TCP客户端（从"UCLA"）：

```
./build/apps/tcp_eth_udp_client 2080 127.0.0.1:9080 80.0.0.1 80.6.5.4 50.9.8.7:80
```

如果一切正常，你的"tshark"终端将显示ARP交换然后是TCP连接建立。你的"路由器"终端将显示两个方向上的大量数据包。你的TCP客户端和服务终端将显示已建立的连接（如你在检查点3中看到的）。

尝试在UCLA终端中输入"LO"并按Enter。它是否到达了斯坦福？

## 5 如果遇到问题...

使用tshark终端（或wireshark）以及路由器的调试输出来调试。

考虑构建endtoend程序的"清理"（错误检查）版本。它将找到你代码中许多未定义行为和无效地址使用的实例。（见上面的说明。）

运行完整的单元测试套件（包括本学期同学贡献的新测试）：`cmake --build build --target test`

## 6 三人互联网

现在你已经在自己的计算机内有了一个可工作的迷你互联网，尝试与两个伙伴在CS144课程网络上做同样的练习。一个人运行"路由器"，一个人运行TCP客户端，第三个人运行TCP服务器。你可能需要重新加入CS144课程网络（使用检查点1中的说明，并访问 <https://cs144.net>）。

更改物理地址以匹配你的实际物理地址（以10.144开头）。你不需要更改任何虚拟地址。

## 7 构建更大的互联网

在接下来的三周和两次实验课中，努力设计你能设计的最大的虚拟互联网（最多的路由器和主机）。每个学生将运行一个网络，我们希望你们中的许多人将连接起来创建一个大型的互联网网络。相互分配IP块和IP块的子网，并设置适当的路由表以连接几个学生运行的路由器形成"有趣的"拓扑。安排在此网络上同时发生多个TCP连接。在你的实验报告中，包括互联网拓扑的图表，包含所有相关地址、路由表和连接——以及运行每个路由器或端点的学生姓名。

## 8 创意项目

最后，请将webget、ip\_raw、tcp\_ipv4、tcp\_udp\_eth和fun\_router程序作为起点，编写一个涉及计算机网络的创意程序。这可以是一个小游戏、聊天程序、小型电子邮件客户端或Web浏览器、traceroute/mtr程序、Web服务器、某种疯狂的路由器，或任何与本课程主题相关的、能激发你灵感的东西。

## 9 提交

以PDF格式撰写报告。请包括：

你的三人互联网的描述。谁参与了（请包括SUnet ID），各自扮演什么角色？你能在所有方向上交换数据吗？你能发送一个1-MB的文件并让它完全相同地到达吗（使用与检查点3中相同的过程）？是否发生了有什么有趣的事情（或者你是否发现了需要修复的错误）？

你的"更大互联网"的描述。请绘制拓扑图（手绘并拍照也可以）。包括相关的地址范围、路由表和连接，以及运行每个路由器或端点的学生的姓名/SUnet ID。你在此网络上运行了什么工作负载/流？有什么惊喜吗？

描述你做的创意项目！根据适当情况显示截图或描述它的功能。包括你为此编写的最重要或最有趣的代码片段。而且...为什么这很酷（在你看来）？

你对本课程的任何反思或对未来年份CS144实验中包含的未来材料的建议。你想看到关于拥塞控制、实时音频或视频的检查点吗？

CA们和Keith都渴望阅读你们所有人的作品。感谢大家度过了一个美好的学期，节日快乐！