

DNS server实验

搭建在OpenNetLab上的DNS服务器

- 实验背景与目标
- 实验准备
- 实验任务
- 代码测评
- 实验结果提交与展示
- Tips

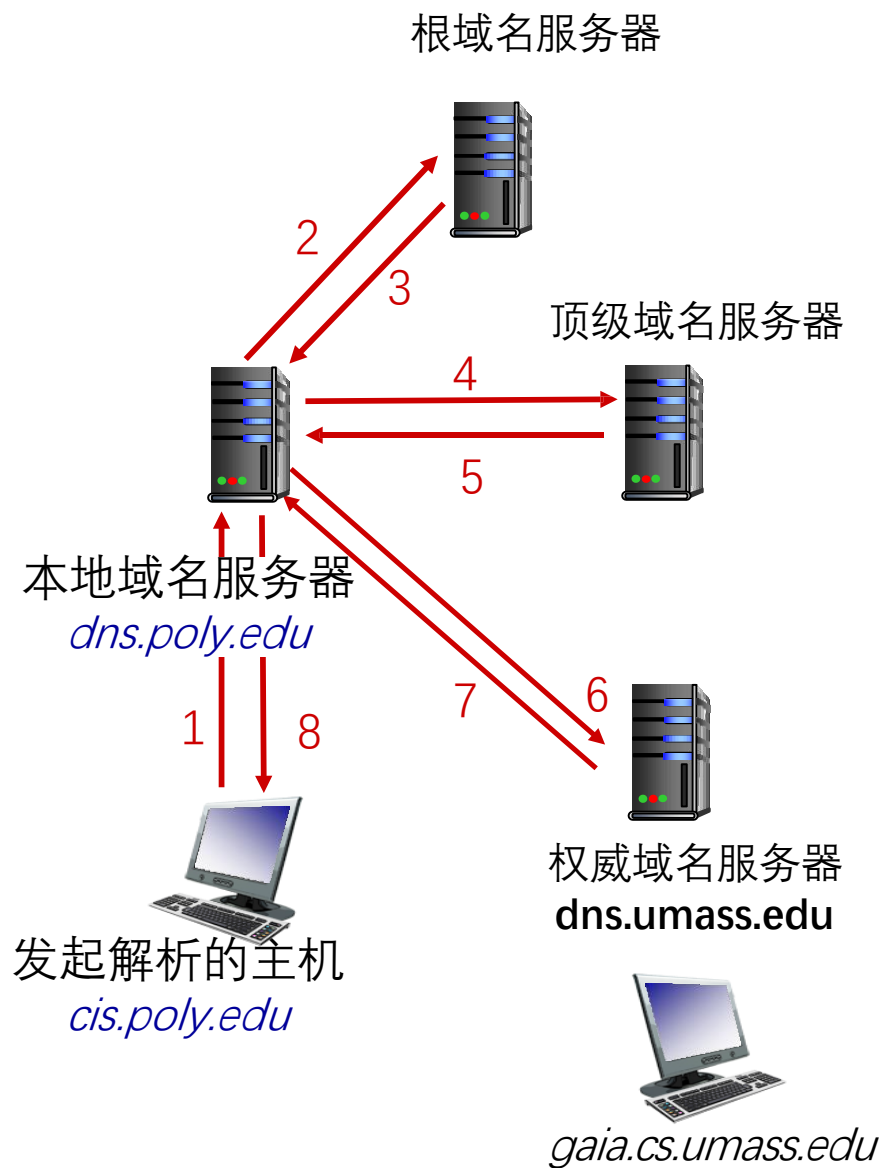
实验背景： 域名解析示例

- 在cis.poly.edu的主机希望获得 gaia.cs.umass.edu的IP地址

实际解析过程：

- 主机到本地域名服务器是递归的
- 剩余过程是迭代的
- 如图所示

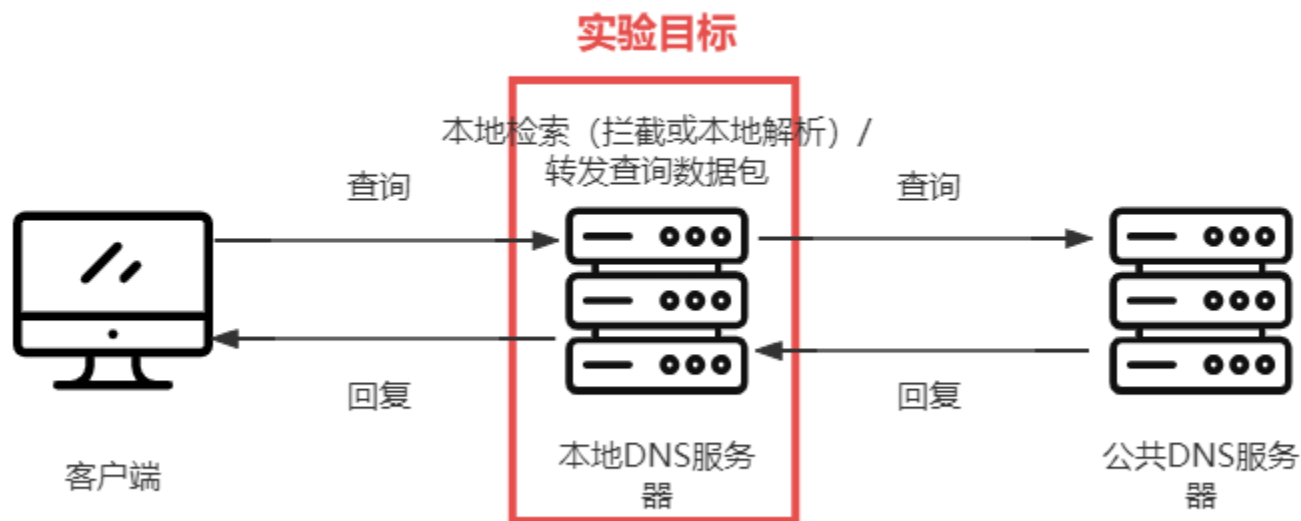
* 如果不了解DNS，请仔细阅读《第二章 应用层》PPT第2.4节



实验目标

本实验需要实现一个搭建在OpenNetLab上的DNS服务器，该服务器具备以下功能：

- 拦截特定域名
- 返回域名对应的IP地址



实验准备

- 注册OpenNetLab账号
- 推荐的实验环境：
 - Linux操作系统，Python版本3.6以上
 - 方法一：学校VLAB实验平台<https://vlab.ustc.edu.cn/>创建虚拟机
 - 方法二：VMware Workstation或VirtualBox安装Linux操作系统
- 在Linux系统中安装onl库

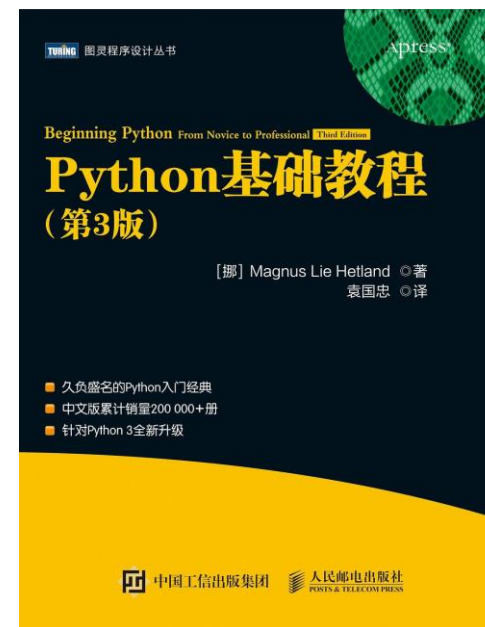
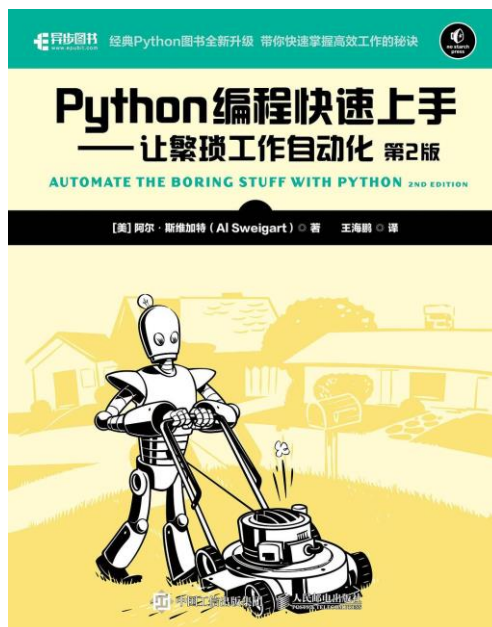
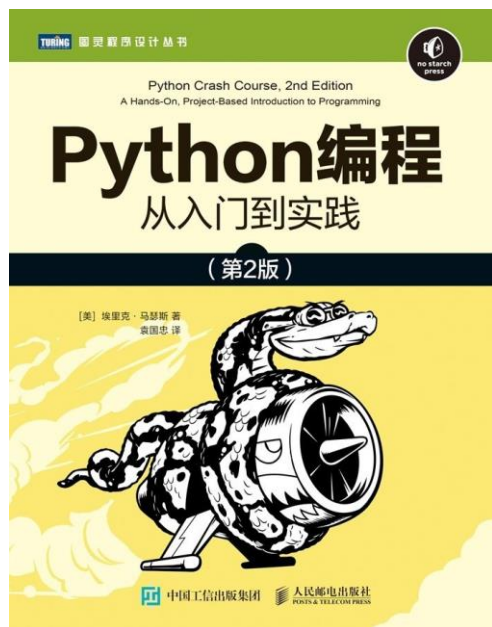
pip install onl-edu

* OpenNetLab的使用请参考《网络教学平台学生使用手册》

实验准备

对于Python零基础的同学：

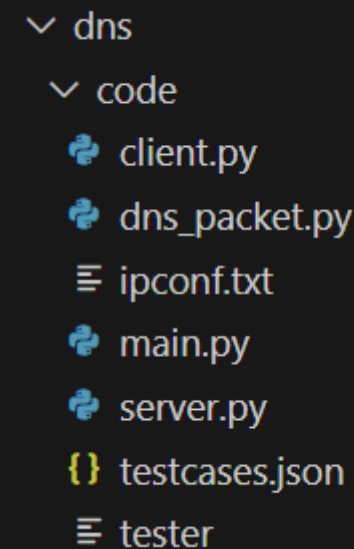
- Python入门书籍



- Bilibili、YouTube上各种Python入门课程

实验准备

- 下载[dns.zip](#)并解压，得到以下文件夹：
 - code文件夹:包含基础的实验代码模板
 - client.py:客户端代码，无需修改
 - server.py:服务器代码， *TODO*部分待编写
 - dns_packet.py:数据包解析和编码代码，无需修改
 - main.py:本地调试代码
 - tester:可执行的本地测评文件
 - ipconf.txt,testcases.json:配置文件，无需修改



```
▼ dns
  ▼ code
    client.py
    dns_packet.py
    ipconf.txt
    main.py
    server.py
    testcases.json
    tester
```

实验任务

学生需要完成的工作：

1. 完成编写server.py代码模板中 *TODO* 部分的代码片段
2. 进行本地测试
3. 将server.py代码提交到在线平台进行远程测评
4. 完成实验报告，准备实验结果展示

实验任务

处理DNS请求：

server.py从配置文件ipconf.txt中读取IP地址-域名的映射列表，并按照以下方式处理DNS查询：

- **拦截**：如果查询的域名在列表中，并且其关联的IP地址是“0.0.0.0”，对该域名进行拦截，向客户端返回“域名不存在”的报错消息。
- **本地解析**：如果查询的域名在列表中，并且有一个有效的关联IP地址，就回复该IP地址。
- **中继**：如果域名不在列表中，将对应的DNS请求转发给公网上的DNS服务器，当收到该服务器的应答时，将应答转发给客户端。

实验任务

处理DNS请求：

需要补充的代码：

```
async def recv_callback(self, data: bytes):  
    ...  
    TODO: 处理DNS请求, data参数为DNS请求数据包对应的字节流  
    1. 解析data得到构建应答数据包所需要的字段  
    2. 根据请求中的domain name进行相应的处理:  
        2.1 如果domain name在self.url_ip中, 构建对应的应答数据包, 发送给客户端  
        2.2 如果domain name不在self.url_ip中, 将DNS请求发送给public DNS server  
    ...
```

实验任务

处理DNS请求：

*TODO*部分对应的伪代码：

```
# process DNS requests
function recv_callback(data){
    resolve data to DNS frame recvd
    if recvd is a query message{
        if recvd.qname is in url_ip table{
            if url_ip[recvd.qname]="0.0.0.0"{
                generate a reply error data
            }
            else{
                generate response data
            }
        }
        else{
            send query message to public DNS server
            receive data from public server
        }
        send data to client
    }
}
```

代码测评

- **本地调试：**

在本机运行 *main.py* 程序，使用本地的一个测试用例对DNS服务器的正确性进行评测并输出运行日志。可以借助 *main.py* 进行代码调试。

python3 main.py

- **本地测评：**

在本机运行 *tester* 程序，使用本地的测试用例对DNS服务器的正确性进行评测。

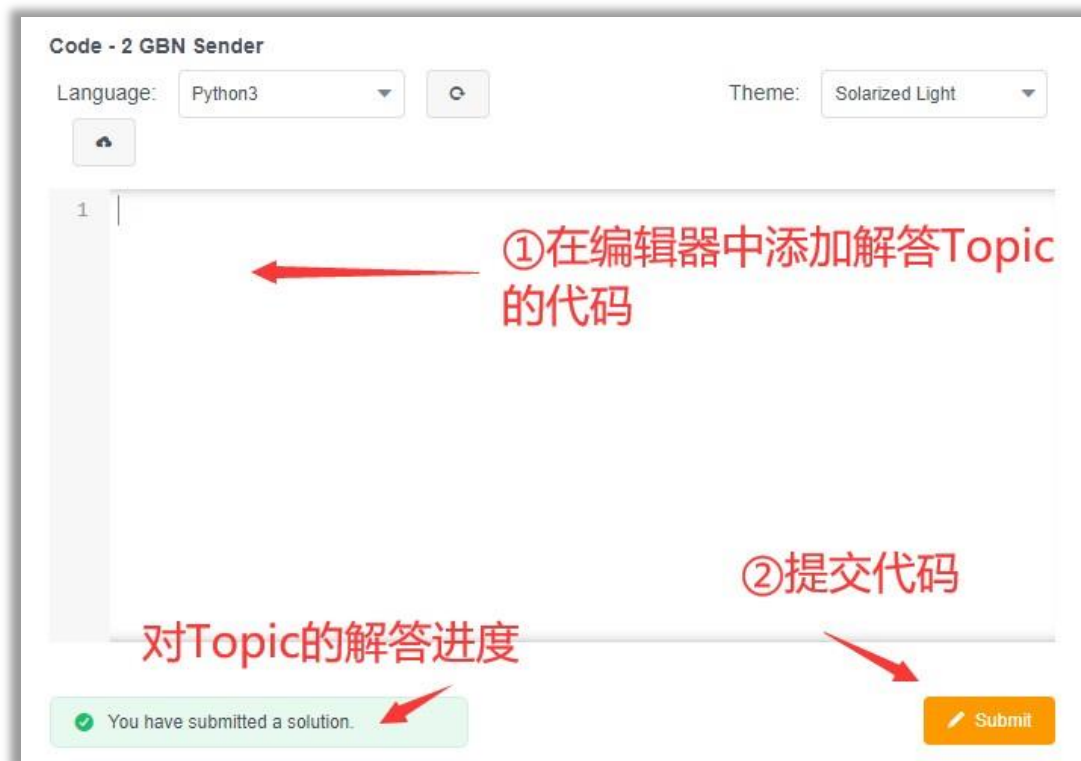
./tester

- **远程测评：**

在OpenNetLab上提交代码进行远程评测。

OpenNetLab

- 选择实验DNS Server
- 提交补充后的完整的server.py代码



OpenNetLab

- ## • 查看评估结果

9种状态:

- **COMPILE ERROR**
- **WRONG ANSWER**
- **ACCEPTED**
- **RUNTIME ERROR**
- **SYSTEM ERROR**
- **NETWORK TIMEOUT**
- **PENDING**
- **EVALUATING**
- **PARTIALLY ACCEPTED**

Submissions

Status All

When	ID	Status	Topic	Running server	Language	Author
2022-8-4 01:55:11	106aebff5f1a	Accepted	1	http://192.168.179.128:8000	Python3	cjy
2022-8-4 01:45:25	533cc646f071	Judging	1	http://192.168.179.128:8000	Python3	cjy

<

1

>

Welcome to OpenNetLab Project

实验结果提交

1. 在OpenNetLab上提交代码

代码提交次数与测试样例通过率会影响实验最终评分

推荐进行本地调试，不鼓励根据远程测试样例结果进行调试

禁止代码抄袭，单人独立完成

代码远程测评提交截止时间：2023.11.27,23:59。

*临近DDL平台容易崩溃导致提交失败，请勿踩点提交。不接受以踩点失败、提交失败为理由的补交。

实验结果提交

2. 以PDF或DOC格式在bb系统作业区提交实验报告

单人独立完成，报告文件命名格式：学号_姓名_实验一.pdf/docx

实验报告需要包括以下内容：

- TODO部分补充的代码
- 对补充代码的解释
- 对未通过的远程测试用例的解释（通过率100%则不需要写）
- tester程序在本地执行的输出结果，以截图方式呈现
- 代码在OpenNetLab上的最终评估结果，以截图方式呈现

报告提交截止时间：2023.11.27,23:59。

*临近DDL平台容易崩溃导致提交失败，请勿踩点提交。不接受以踩点失败、提交失败为理由的补交。

实验结果展示

进行实验结果展示需要包括：

- 对代码的解释
- 程序在本地的执行和输出

展示时间：暂未确定

Tips

- dns_packet.py: 阅读RFC1035, 理解DNS请求和应答报文中各个字段的含义, 理解dns_packet.py是如何实现解析和生成DNS数据包。
- client.py: DNSClient类和DNSServer类都继承了UDPDevice类, server如何向client发送数据包可以参考client发送DNS请求数据包。
- tester: 可以带参数“-h”来查看帮助信息, 更详细的使用说明见[Testing页面](#)。

./tester -h

Tips

- server.py: server通过socket与公共DNS server建立通信连接，以发送接收DNS数据包。以下代码片段表示DNS server向上游DNS服务器（指定为"223.5.5.5"， port 53）发送DNS查询，并监听本地的1088端口得到查询结果。

```
self.name_server = ("223.5.5.5", 53)
self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
self.server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
self.server_socket.bind(("", 1088))
self.server_socket.setblocking(True)
```

- 请仔细阅读代码模板和[实验文档](#)。

联系方式

实验相关问题请在钉钉群内@助教

- 实验1 DNS内容相关问题
 - 钉钉群：助教-王涔鹄
- 实验评测平台相关问题
 - 钉钉群：助教-杨帆