

به نام خدا



درس شبکه‌های کامپیوتری

نیم‌سال اول ۰۴-۰۵

استاد: دکتر دولتی

تیم آموزشی: امیرحسن جعفرآبادی، نیکی سپاسیان، دوین غربی

دانشکده مهندسی کامپیوتر

مهلت تحویل: ۲۰ آبان ۱۴۰۴

پروژه سری اول - پیاده‌سازی سرور DNS محلی

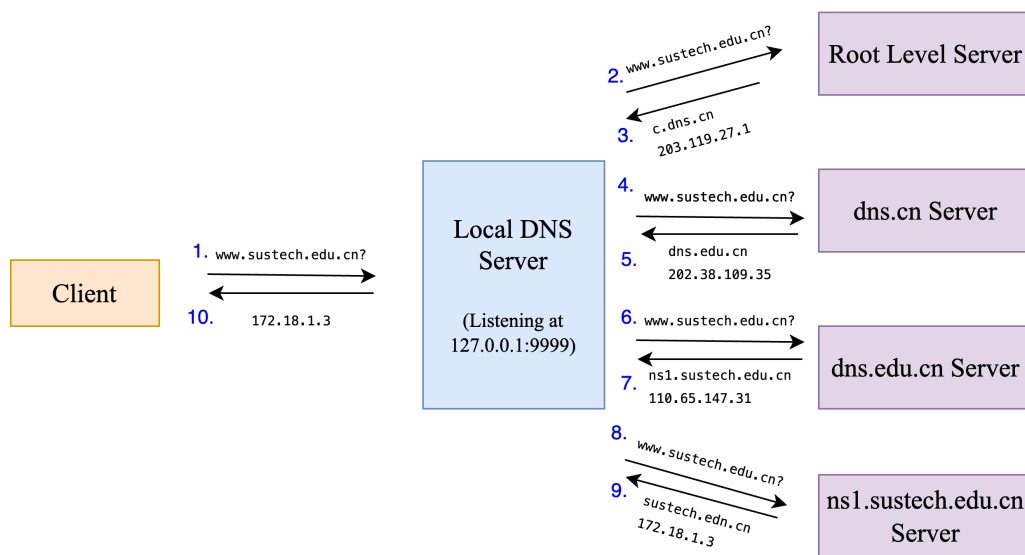
- هر پروژه می‌تواند حداکثر تا پنج روز با تأخیر تحویل داده شود. در صورت تأخیر، به ازای هر یک از سه روز اول، ۱۰ درصد از نمره پروژه کسر می‌شود. برای دو روز بعدی، به ازای هر روز ۲۰ درصد از نمره کسر خواهد شد. پس از روز پنجم، نمره پروژه صفر در نظر گرفته می‌شود.
- از به‌کارگیری ابزارهای مبتنی بر هوش مصنوعی و مدل‌های زبانی در حل تمارین جداً خودداری فرمایید. تشابه قابل توجه میان پاسخ‌های دو یا چند نفر، منجر به **لحاظ نمره صفر** برای تمام افراد خواهد شد.
- ارسال پاسخ‌ها صرفاً باید از طریق تمرین تعیین شده در **کلاس درس در سامانه CW** انجام شود. پاسخ‌هایی که از طریق ایمیل، پیام یا ... ارسال شوند مورد پذیرش قرار نخواهند گرفت.
- در صورت وجود هرگونه ابهام در صورت سؤال یا نحوه‌ی ارسال، از طریق سامانه‌ی CW سوالات خود را مطرح کنید.

Local DNS Server (نمره ۱۰۰) ۱.

در این تمرین، شما باید یک local DNS server به زبان پایتون پیاده‌سازی کنید که از iterative DNS query پشتیبانی کند.

توجه: در این پروژه استفاده از کتابخانه‌های جانبی مانند dnspython، dnslib یا هر کتابخانه دیگری که به طور خاص برای تجزیه بسته‌های DNS طراحی شده است مجاز نمی‌باشد. استفاده از کتابخانه‌های استاندارد پایه مانند socket و IO ممانعتی ندارد.

مرور کلی



شکل ۱: مثالی از iterative DNS query برای دریافت آدرس www.sustech.edu.cn

در این پروژه، باید یک local DNS server پیاده‌سازی کنید که روی آدرس 127.0.0.1:9999 گوش می‌دهد. هنگامی که یک درخواست DNS دریافت می‌شود، سرور محلی DNS یک درخواست به root server ارسال می‌کند. سپس بر اساس پاسخ root server، درخواست بعدی را به سرور بعدی ارسال می‌کند و این کار را تا زمانی که نتیجه نهایی را دریافت کند (به صورت iterative مشابه مثالی که در شکل ۱ آورده شده است) ادامه می‌دهد. در نهایت، نتیجه به client بازگردانده می‌شود.

client: برای آزمایش درستی سرور محلی شما از ابزار dig استفاده می‌شود. DNS برای مثال، می‌توانید با وارد کردن دستور زیر در خط فرمان، یک درخواست DNS به سرور محلی گوش‌دهنده روی پورت ۹۹۹۹ ارسال کنید:

```
dig @127.0.0.1 www.sustech.edu.cn a -p 9999
```

dig بعد از دریافت پاسخ معتبر آن را چاپ خواهد کرد. در ادامه نمونه‌ای از خروجی dig آورده شده است:

```
>>> DiG 9.10.6 <<>> @127.0.0.1 www.sustech.edu.cn a -p 9999
; (1 server found)
;; global options: +cmd
```

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30163
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.sustech.edu.cn. IN A

;; ANSWER SECTION:
www.sustech.edu.cn. 3600 IN CNAME sustech.edu.cn.
sustech.edu.cn. 3600 IN A 172.18.1.3

;; AUTHORITY SECTION:
sustech.edu.cn. 3600 IN NS ns1.sustech.edu.cn.
sustech.edu.cn. 3600 IN NS ns2.sustech.edu.cn.

;; ADDITIONAL SECTION:
ns1.sustech.edu.cn. 3600 IN A 172.18.1.92
ns1.sustech.edu.cn. 3600 IN AAAA 2001:da8:201d::42:92
ns2.sustech.edu.cn. 3600 IN A 172.18.1.93
ns2.sustech.edu.cn. 3600 IN AAAA 2001:da8:201d::42:93

;; Query time: 3467 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Thu Mar 07 18:33:30 CST 2024
;; MSG SIZE rcvd: 380
```

Local DNS Server: همانطور که پیشتر اشاره شد، سرور محلی DNS روی آدرس 127.0.0.1:9999 گوش می‌دهد. از آنجا که استفاده از کتابخانه‌هایی مانند dnspython و dnslib مجاز نیست، شما باید خودتان فرآیند تجزیه و ساخت بسته‌های DNS را پیاده‌سازی کنید. یک فایل پایه با نام local_dns_server.py در اختیار شما قرار داده شده است که شامل یک سرور ساده UDP تحت عنوان MyLocalDNSServerHandler می‌باشد. شما باید پروسه iterative DNS query خواسته شده را در تابع handle این کلاس پیاده‌سازی کرده و خروجی مد نظر را بازگردانید. لطفاً به توضیحات موجود در کامنت‌های فایل مذکور توجه فرمایید.

دستورالعمل پروژه

محیط اجرا

پایتون

توصیه می‌شود از نسخه‌ی Python 3.8 استفاده کنید. نسخه‌های بالاتر ممکن است موجب بروز خطا شوند. می‌توانید از دستور زیر برای ساخت محیط مورد نظر استفاده نمایید:

```
conda create -n assign1 python=3.8
```

Dig

ابزار dig یک ابزار خط فرمان برای ارسال درخواست به سرورهای DNS برای دریافت اطلاعات درباره‌ی یک نام دامنه است. dig در سیستم‌عامل‌های Linux و macOS به صورت پیش فرض نصب شده است. برای نصب آن در Windows می‌توانید به این راهنما مراجعه نمایید.

تجزیه بسته DNS

از آنجا که اجازه استفاده از کتابخانه‌های جانبی وجود ندارد، باید بسته‌های DNS را به صورت دستی تجزیه کنید. قالب یک بسته‌ی DNS به شکل زیر است:

Header	
Question	the question for the name server
Answer	RRs answering the question
Authority	RRs pointing toward an authority
Additional	RRs holding additional information

درخواست‌ها و پاسخ‌های DNS هر دو دارای همین ساختار کلی می‌باشند. یک بسته‌ی DNS شامل بخش‌های header، question، answer، authority و additional است. سه بخش آخر با استفاده از رکوردهای منبع (RR) ساخته می‌شوند. بنابراین در فایل local_dns_server.py نیز برای نمایش کامل یک بسته DNS، ۴ کلاس تحت عنوان‌های DNSHeader، DNSMessage، DNSQuestion و DNSRR تعریف شده‌اند.

تجزیه DNS Header (۱۰ نمره)

در گام اول باید DNS Header را تجزیه کنید.

											1	1	1	1	1	1
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
	ID															
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
QR	Opcode					AA	TC	RD	RA		Z	RCODE				
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
	QDCOUNT															
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
	ANCOUNT															
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
	NSCOUNT															
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																
	ARCOUNT															
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----																

یک DNS header به طور معمول شامل ۱۲ بایت است. بایت‌های ۰-۱ شناسه (ID)، بایت‌های ۲-۳ پرچم‌ها (flags) و بایت‌های ۴-۱۱ شماره‌ها هستند. شما باید header را تجزیه کرده و اطلاعات را در کلاس DNSHeader ذخیره کنید. این کلاس شامل ویژگی‌های زیر است:

```
self.id : int = 0
self.flag : bytes = b'' # maintain the 2 bytes of flags
self.qdcount : int = 0 # num of entries in the question section
self.ancount : int = 0 # num of RRs in the answer section
self.nscount : int = 0 # num of RRs in the authority records section
self.arcount : int = 0 # num of RRs in the additional records section
```

ابتدا باید تابع `from_wire(data:bytes)->DNSHeader` را تکمیل کنید. این تابع توسط `test.py` فراخوانی می‌شود که بایت‌های خام یک DNS header را به عنوان ورودی دریافت کرده و یک شیء از نوع `DNSHeader` باز می‌گرداند. برای تسهیل پروسه دیباگ، یک نمونه `DNS header` در فایل `./raw_packet/header.raw` قرار داده شده است. بنابراین این فایل تنها شامل ۱۲ بایت ابتدایی از یک بسته DNS می‌باشد. در صورت نیاز می‌توانید متغیرهای تابع `__init__` را تغییر دهید، اما متغیرهای ورودی تابع `from_wire` باید دست‌نخورده باقی بمانند.

می‌توانید برای بررسی صحت خروجی در این بخش، فایل `test.py` را اجرا کنید. در صورتی که تجزیه بخش DNS header به درستی انجام شده باشد، خروجی زیر را مشاهده خواهید کرد:

```
python test.py
**header test passed**
.FF
=====
FAIL: test_question (__main__.TestDNSResolver)
-----
Traceback (most recent call last):
  File "test.py", line 36, in test_question
    self.assertEqual(type(question), DNSQuestion)
AssertionError: <class 'NoneType'> != <class 'local_dns_server.DNSQuestion'>

=====
FAIL: test_whole_msg (__main__.TestDNSResolver)
-----
Traceback (most recent call last):
  File "test.py", line 44, in test_whole_msg
    self.assertEqual(type(msg), DNSMessage)
AssertionError: <class 'NoneType'> != <class 'local_dns_server.DNSMessage'>

-----
Ran 3 tests in 0.01s

FAILED (failures=2)
```

تجزیه DNS Question (۱۰ نمره)

پس از تجزیه DNS header نوبت به DNS question می‌رسد. در این بخش نیز یک نمونه `DNS question` در مسیر `./raw_packet/question.raw` فراهم شده است. تضمین می‌شود که تمامی بسته‌های DNS تنها شامل یک

question خواهند بود. اطلاعات موجود در DNS question به طور کلی عبارتند از:

- self.qname: رشته‌ای که نام دامنه را نمایش می‌دهد.
- self.qtype: عددی که نوع درخواست را نشان می‌دهد.
- self.qclass: عددی که کلاس درخواست را نشان می‌دهد.

برای اطلاعات بیشتر درباره فرمت و قالب بخش DNS question، می‌توانید به بخش ۴.۱.۲ از RFC1035 مراجعه نمایید.

بعد از تکمیل توابع from_wire و __init__ در کلاس DNSQuestion، مشابه قبل می‌توانید برای بررسی صحت خروجی تجزیه شده از فایل test.py استفاده نمایید.

تحلیل پیام DNS (۳۰ نمره)

پیام DNSMessage شامل ویژگی‌های زیر است:

- self.header: یک شیء از کلاس DNSHeader
- self.question: یک شیء از کلاس DNSQuestion (در تمام تست‌ها فقط یک سؤال وجود دارد)
- self.answer: فهرستی از اشیاء DNSRR
- self.authority: فهرستی از اشیاء DNSRR
- self.additional: فهرستی از اشیاء DNSRR

شما باید تابع from_wire را پیاده‌سازی کنید تا کل بسته DNS را تجزیه کند. مهم‌ترین بخش، تجزیه رکوردهای RR در بخش‌های answer، authority و additional است. همه رکوردهای RR دارای یک قالب یکسان هستند. قالب رکورد RR در بخش ۴/۱/۳ از سند RFC1035 آمده است:

<https://www.ietf.org/rfc/rfc1035.txt>

کلاس DNSRR دارای ویژگی‌های زیر است:

- self.name: رشته‌ای که نام دامنه را نمایش می‌دهد.
- self.type: عددی که نوع رکورد را مشخص می‌کند.
- self.class_: عددی که کلاس رکورد را نشان می‌دهد.
- self.ttl: عددی که زمان ماندگاری (time to live) رکورد را نشان می‌دهد.
- self.rdlength: طول داده RDATA بر حسب بایت.
- self.rdata: رشته یا شیء بایت که مقدار RDATA را نمایش می‌دهد.

نکته ۱: نام دامنه می‌تواند به سه صورت نمایش داده شود: قالب عادی، قالب فشرده (compressed) یا ترکیبی از هر دو. شما باید هر دو حالت را پشتیبانی کنید. بخش ۴/۱/۴ از RFC1035 با عنوان Message compression را مطالعه کنید.

نکته ۲: تنها نیاز است نوع‌های زیر از رکوردهای RR را پردازش کنید: A، NS، CNAME و AAAA.

- برای نوع A: مقدار self.rdata باید یک رشته از آدرس IPv4 باشد (برای مثال: 183.2.172.42)
- برای نوع AAAA: مقدار self.rdata باید یک رشته از آدرس IPv6 باشد (برای مثال: 2001:0DB8:AC10:FE01::)

- برای نوع‌های NS و CNAME: مقدار self.rdata باید یک نام دامنه باشد.
 - برای نوع (41) OPT می‌توانید مقدار self.rdata را برابر رشته «root» قرار دهید (این حالت در تصحیح تست نخواهد شد).
 - سایر نوع‌ها می‌توانند مقدار rdata به صورت بایت داشته باشند.
- زمانی که پیاده‌سازی توابع from_wire در کلاس‌های DNSHeader، DNSQuestion و DNSMessage را تمام کردید، می‌توانید با اجرای دستور زیر نتیجه را آزمایش کنید:

```
python test.py
```

در صورت موفقیت در تجزیه کامل بسته DNS، خروجی زیر را مشاهده خواهید کرد:

```
python test.py
**header test passed**
.**question test passed**
.**whole msg test passed**
.
```

```
-----
Ran 3 tests in 0.001s
```

```
OK
```

این خروجی به معنای گذر از تمام تست‌هاست. در هنگام تصحیح، تست دیگری با داده‌های متفاوت اجرا می‌شود تا عملکرد شما ارزیابی گردد.

پیاده‌سازی سرور DNS (۴۰ نمره)

شما باید تابع handle در کلاس MyLocalDNSServerHandler را تکمیل کنید تا یک سرور DNS از نوع تکراری (iterative) پیاده‌سازی شود. فرآیند تکرار زمانی خاتمه می‌یابد که رکوردی از نوع A در بخش پاسخ (answer) وجود داشته باشد. اگر تنها رکورد CNAME در بخش پاسخ وجود دارد، باید درخواست جدیدی به سرور دامنه CNAME ارسال کرده و بخش پاسخ جدید را به پاسخ قبلی اضافه کنید.

تنها رکورد CNAME در بخش پاسخ

برای مثال، زمانی که دستور زیر اجرا می‌شود:

```
dig @127.0.0.1 www.baidu.com a -p 9999
```

ممکن است تنها یک رکورد از نوع CNAME در پاسخ وجود داشته باشد که به دامنه www.a.shifen.com اشاره می‌کند. در این صورت، باید درخواست جدیدی به سرور مربوط به این دامنه ارسال کرده و رکوردهای پاسخ جدید را به پاسخ اولیه اضافه نمایید.

```

[(base) → ~ dig @127.0.0.1 www.baidu.com a -p 9999
;; Warning: Message parser reports malformed message packet.

; <<>> DiG 9.10.6 <<>> @127.0.0.1 www.baidu.com a -p 9999
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53401
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.baidu.com.                IN      A

;; ANSWER SECTION:
www.baidu.com.                1200    IN      CNAME   www.a.shifen.com.

;; Query time: 236 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Fri Mar 08 10:52:11 CST 2024
;; MSG SIZE rcvd: 90

```

Only query www.baidu.com

wrong!

```

[(assign1) → CS305-Assignment1 git:(main) * dig @127.0.0.1 www.baidu.com a -p 9999

```

```

; <<>> DiG 9.10.6 <<>> @127.0.0.1 www.baidu.com a -p 9999
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6665
;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 9

```

```

;; QUESTION SECTION:

```

```

;www.baidu.com.                IN      A

```

```

;; ANSWER SECTION:

```

```

www.baidu.com.                1200    IN      CNAME   www.a.shifen.com.
www.a.shifen.com.            300     IN      A        182.61.200.7
www.a.shifen.com.            300     IN      A        182.61.200.6

```

right

```

;; AUTHORITY SECTION:

```

```

a.shifen.com.                1200    IN      NS       ns1.a.shifen.com.
a.shifen.com.                1200    IN      NS       ns2.a.shifen.com.
a.shifen.com.                1200    IN      NS       ns3.a.shifen.com.
a.shifen.com.                1200    IN      NS       ns4.a.shifen.com.
a.shifen.com.                1200    IN      NS       ns5.a.shifen.com.

```

```

;; ADDITIONAL SECTION:

```

```

ns1.a.shifen.com.            300     IN      A        110.242.68.42
ns2.a.shifen.com.            600     IN      A        220.181.33.32
ns3.a.shifen.com.            300     IN      A        36.155.132.12
ns3.a.shifen.com.            300     IN      A        153.3.238.162
ns4.a.shifen.com.            300     IN      A        14.215.177.229
ns4.a.shifen.com.            300     IN      A        111.20.4.28
ns5.a.shifen.com.            600     IN      A        180.76.76.95
ns5.a.shifen.com.            600     IN      AAAA     240e:bf:b801:1006:0:ff:b04f:346b
ns5.a.shifen.com.            600     IN      AAAA     240e:940:603:a:0:ff:b08d:239d

```

```

;; Query time: 456 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Thu Mar 21 15:42:06 CST 2024
;; MSG SIZE rcvd: 660

```


نبودن بخش Additional در پاسخ

در برخی موارد، پاسخ دریافتی فاقد بخش additional است، یعنی آدرس IP سروری که باید در مرحله بعدی مورد پرس‌وجو قرار گیرد در آن وجود ندارد. اما در بخش authority رکوردهایی از نوع NS وجود دارند که شامل نام دامنه سرور هستند. در این حالت، باید یک پرس‌وجوی جدید برای به‌دست آوردن آدرس IP این دامنه انجام دهید و پس از آن، پرس‌وجوی بعدی را ادامه دهید.

تست سرور محلی DNS

می‌توانید دامنه‌هایی مانند `www.baidu.com`، `www.example.com`، `www.bilibili.com` و سایر دامنه‌ها را آزمایش کنید.

نکته: برای اجرای سرور محلی DNS، دستور زیر را در یک ترمینال اجرا کنید:

```
python local_dns_server.py
```

سپس در ترمینال دیگر با استفاده از `dig` آن را تست کنید.

گزارش (۱۰ نمره)

گزارش شما باید شامل موارد زیر باشد:

- نام و شماره دانشجویی
- توضیح کدهای اضافه شده
- نحوه مدیریت حالت وجود تنها رکورد CNAME در بخش پاسخ (به همراه کد و توضیح)

فایل‌های مورد نیاز برای ارسال

- فایل `local_dns_server.py` (تغییر نام ندهید)
- فایل گزارش با نام `SID_Name_report.pdf`، برای مثال:

`12345678_AhmadAhmadi_report.pdf`

موفق باشید!