CNT 4713 – Project 3

Overview:

The goal of this project is to practice UDP socket programing and understand binary packet structures by developing a simplified DNS lookup client. You must create your own socket and cannot use any existing DNS library. You may use Python3, Java, C++, or C as the programming language.

Background:

The Domain Name System (DNS) protocol is a UDP based application layer protocol for domain name resolution. DNS serves as the Internet's phone book: every time you access the Internet, your computer performs DNS lookups to retrieve various information associated with a domain name.

The DNS RFC [1] is the official document that defines the protocol. Chapter 2.4 of the textbook [2] explains DNS in detail, including its resolution processes and message formats.

Instructions:

Your DNS lookup client should use iterative queries to resolve a given domain name. It should be started by typing:

`mydns domain-name root-dns-ip` or
`java mydns domain-name root-dns-ip` or
`python mydns.py domain-name root-dns-ip`

where `domain-name` is the domain name (e.g., cs.fiu.edu) to be resolved and `root-dns-ip` is the IPv4 address (e.g., 202.12.27.33) of a root DNS server on the Internet. The list of current root DNS servers can be found at: https://www.iana.org/domains/root/servers. **Please make sure that your program name is exactly mydns. Submissions will be graded by an automated script. Unrecognized program names will result in a zero.**

Your DNS lookup client should first query the specified root DNS server, which will return one or more intermediate DNS servers (in the Authority and Additional Information sections of the reply). Your client must automatically pick one intermediate server and query it. This process should continue until one or more IP address of the specified domain name are obtained (the IPs are in the *Answers* section of the reply). Then your DNS client should stop.

The DNS iterative resolution process is explained in Chapter 2.4 of the textbook, and is illustrated in Figure 1.

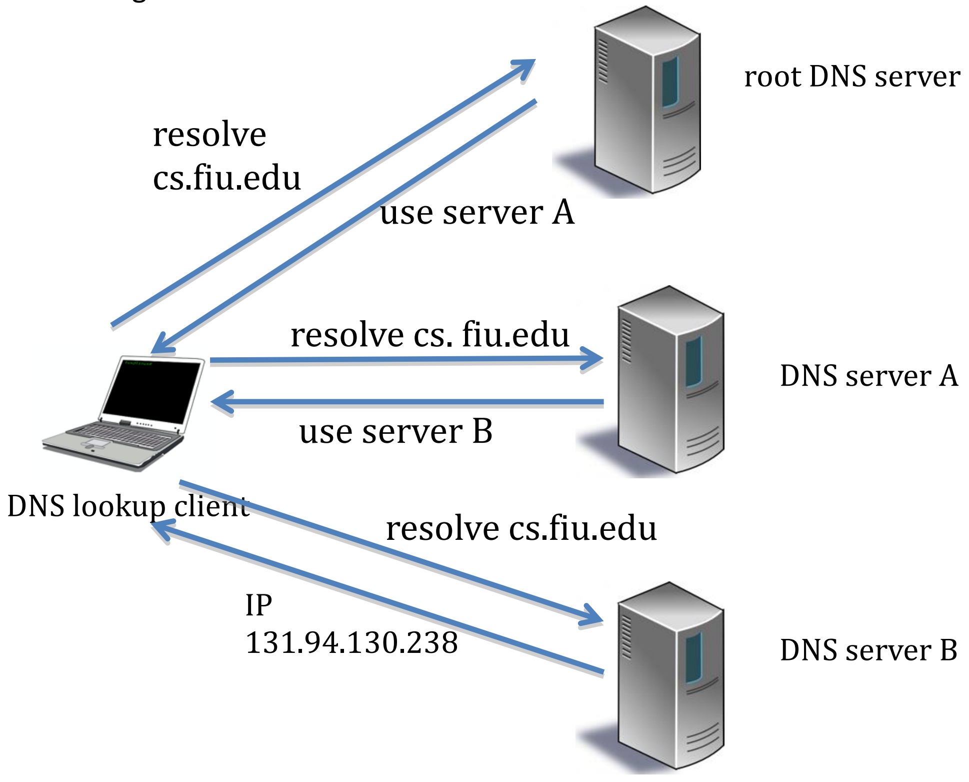


Figure 1. DNS iterative resolution process.

The DNS message formats are explained in Section 4 of the RFC and also Chapter 2.4 of the textbook. To simplify the programming tasks, your DNS lookup client only needs to understand two types of resource records in the reply:

- *NS*, which means the record contains an intermediate name server for the domain name being queried. In this case, your client must also check the *Additional Information* section to find out the IP of the intermediate name server, in order to send a query to the server. If multiple intermediate name servers are included, your client only needs to pick one to query.
- *A*, which means the record contains the IP of the domain name being queried; your DNS client can stop once such a reply is received.

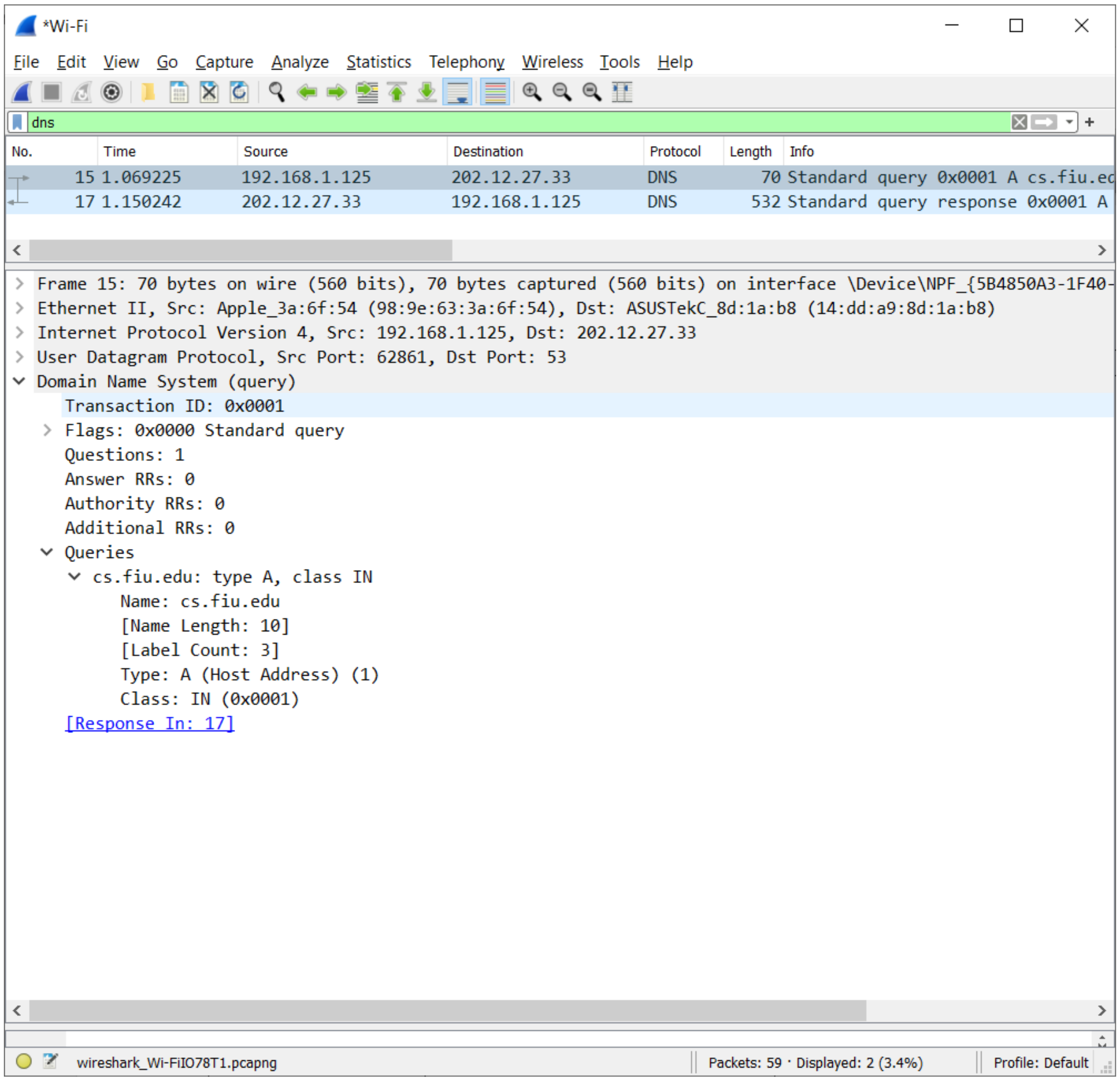
Your DNS lookup client should print all the intermediate results on the screen. For each iterative step, your client must clearly display the content of the reply message, and the DNS server chosen to query next. A sample output is attached in the appendix.

You may focus on only IPv4 and ignore IPv6.

A Python3 based program skeleton is provided in Canvas for you to get started. The skeleton shows how to construct a DNS query, and how to parse the Header section and Question section of a DNS response. In addition to the header that has been parsed in the skeleton, you also need to parse the resource records in the DNS response. **Please carefully read Section 4 of the RFC [1] on how to parse a DNS response. In particular, Section 4.1.4. of the RFC explains a message compression scheme that makes name parsing more challenging.**

Test:

You can use Wireshark [3] to capture the sent and received DNS messages, verify that the message format is correct, and analyze different field values, as shown in Figure 2.

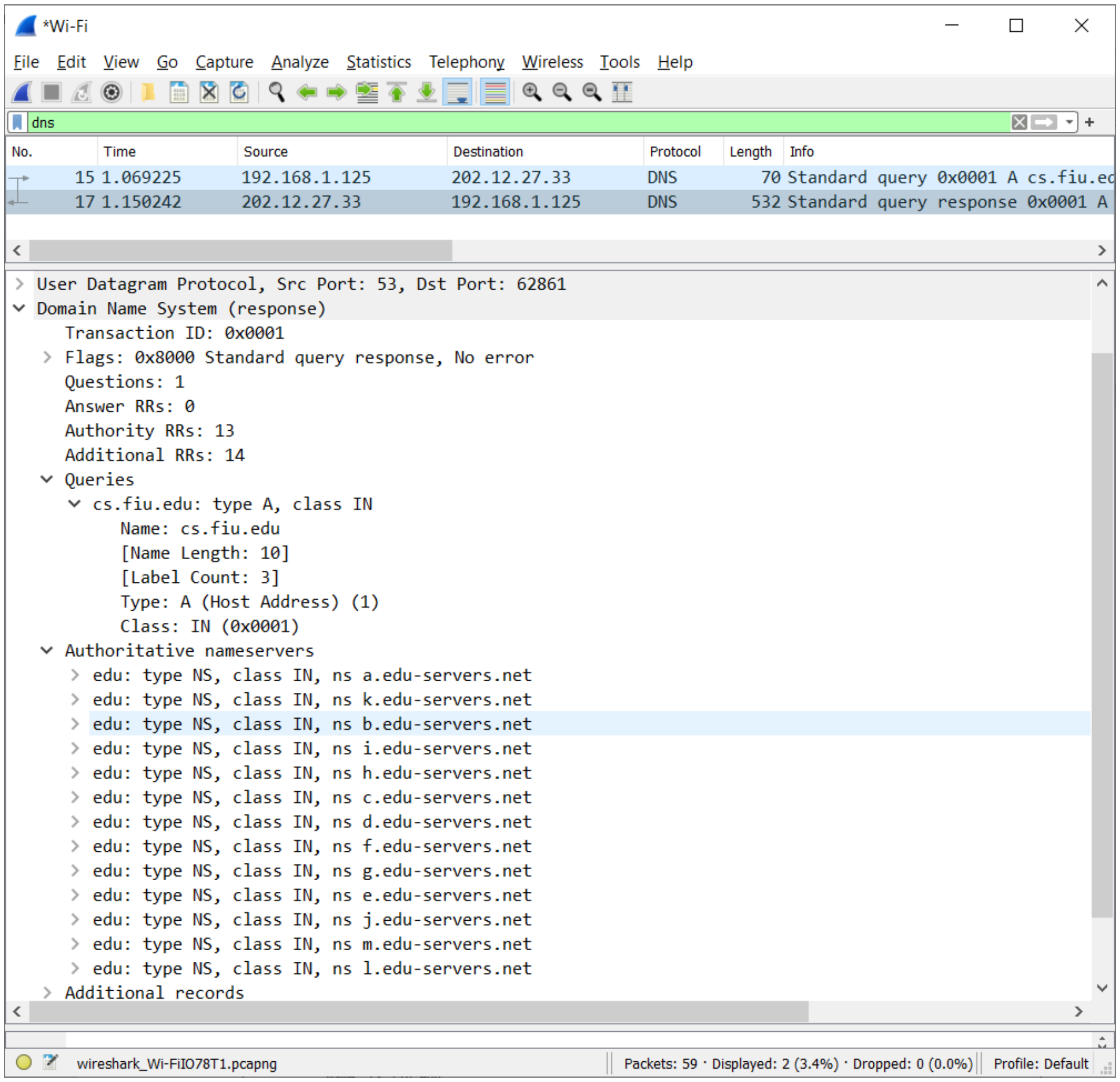

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
dns
No.   Time      Source         Destination    Protocol Length Info
   15 1.069225  192.168.1.125  202.12.27.33   DNS          70 Standard query 0x0001 A cs.fiu.ec
   17 1.150242  202.12.27.33   192.168.1.125  DNS         532 Standard query response 0x0001 A

> Frame 15: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{5B4850A3-1F40-
> Ethernet II, Src: Apple_3a:6f:54 (98:9e:63:3a:6f:54), Dst: ASUSTekC_8d:1a:b8 (14:dd:a9:8d:1a:b8)
> Internet Protocol Version 4, Src: 192.168.1.125, Dst: 202.12.27.33
> User Datagram Protocol, Src Port: 62861, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0001
  > Flags: 0x0000 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v cs.fiu.edu: type A, class IN
        Name: cs.fiu.edu
        [Name Length: 10]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 17]

wireshark_Wi-FiIO78T1.pcapng    Packets: 59 · Displayed: 2 (3.4%)    Profile: Default
```

(a) DNS query analyzed by Wireshark.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.   Time       Source          Destination     Protocol  Length  Info
   15 1.069225   192.168.1.125   202.12.27.33    DNS           70  Standard query 0x0001 A cs.fiu.ed
   17 1.150242   202.12.27.33    192.168.1.125   DNS          532  Standard query response 0x0001 A
```

```
> User Datagram Protocol, Src Port: 53, Dst Port: 62861
v Domain Name System (response)
    Transaction ID: 0x0001
  > Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 13
    Additional RRs: 14
  v Queries
    v cs.fiu.edu: type A, class IN
        Name: cs.fiu.edu
        [Name Length: 10]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  v Authoritative nameservers
    > edu: type NS, class IN, ns a.edu-servers.net
    > edu: type NS, class IN, ns k.edu-servers.net
    > edu: type NS, class IN, ns b.edu-servers.net
    > edu: type NS, class IN, ns i.edu-servers.net
    > edu: type NS, class IN, ns h.edu-servers.net
    > edu: type NS, class IN, ns c.edu-servers.net
    > edu: type NS, class IN, ns d.edu-servers.net
    > edu: type NS, class IN, ns f.edu-servers.net
    > edu: type NS, class IN, ns g.edu-servers.net
    > edu: type NS, class IN, ns e.edu-servers.net
    > edu: type NS, class IN, ns j.edu-servers.net
    > edu: type NS, class IN, ns m.edu-servers.net
    > edu: type NS, class IN, ns l.edu-servers.net
  > Additional records
```

```
wireshark_Wi-FiI078T1.pcapng    Packets: 59 · Displayed: 2 (3.4%) · Dropped: 0 (0.0%)    Profile: Default
```

(b) DNS response analyzed by Wireshark.

Figure 2. Using Wireshark to analyze DNS messages.

Verify the correctness of the lookup results by typing

```
nslookup domain-name 8.8.8.8
```

References:

- [1] DNS RFC, http://www.ietf.org/rfc/rfc1035.txt.
- [2] DNS iterative resolution and message formats, Chapter 2.4 of textbook.
- [3] Wireshark, https://www.wireshark.org/download.html.

Group:

You may work in a group of up to three. Only one submission is necessary for a group.

Submission:

Submit a copy of your final code to Canvas. (Source code only, no NetBeans/Eclipse… project files)

Submit a readme.txt to Canvas with the following information:

1. Member names and IDs
2. Language used: Python3/Java/C++/C
3. Compiling instructions (not necessary for Python): command-line compiling instructions only, no NetBeans/Eclipse… compiling.

Grading:

Item	Percentage
Send query to root DNS server	10%
Parse reply from root DNS server	15%
Display server reply content	15%
Extract intermediate DNS server IP	15%
Send query to intermediate servers	15%
Parse reply from intermediate servers	15%
Display IPs for queried domain name	15%

Code plagiarism will be reported for academic dishonesty.

Appendix. Sample output:

```
$ ./mydns cs.fiu.edu 202.12.27.33

----------------------------------------------------------------
DNS server to query: 202.12.27.33
Reply received. Content overview:
        0 Answers.
        6 Intermediate Name Servers.
        7 Additional Information Records.
Answers section:
Authority Section:
        Name : edu      Name Server: l.edu-servers.net
        Name : edu      Name Server: a.edu-servers.net
        Name : edu      Name Server: f.edu-servers.net
        Name : edu      Name Server: c.edu-servers.net
        Name : edu      Name Server: g.edu-servers.net
        Name : edu      Name Server: d.edu-servers.net
Additional Information Section:
        Name : a.edu-servers.net       IP : 192.5.6.30
        Name : c.edu-servers.net       IP : 192.26.92.30
        Name : d.edu-servers.net       IP : 192.31.80.30
        Name : f.edu-servers.net       IP : 192.35.51.30
        Name : g.edu-servers.net       IP : 192.42.93.30
        Name : l.edu-servers.net       IP : 192.41.162.30
        Name : g.edu-servers.net

-------------------------------------------------------------
DNS server to query: 192.5.6.30
Reply received. Content overview:
        0 Answers.
        5 Intermediate Name Servers.
        5 Additional Information Records.
Answers section:
Authority Section:
        Name : fiu.edu          Name Server: ns.fiu.edu
        Name : fiu.edu          Name Server: ns3.fiu.edu
        Name : fiu.edu          Name Server: ns1.fiu.edu
        Name : fiu.edu          Name Server: drdns.fiu.edu
        Name : fiu.edu          Name Server: ns4.fiu.edu
Additional Information Section:
        Name : ns.fiu.edu       IP : 131.94.205.10
        Name : ns3.fiu.edu      IP : 131.94.226.10
        Name : ns1.fiu.edu      IP : 131.94.7.220
        Name : drdns.fiu.edu    IP : 131.94.69.36
        Name : ns4.fiu.edu      IP : 131.95.205.12

-------------------------------------------------------------
DNS server to query: 131.94.205.10
Reply received. Content overview:
        0 Answers.
        4 Intermediate Name Servers.
        4 Additional Information Records.
Answers section:
Authority Section:
        Name : cs.fiu.edu       Name Server: goedel.cs.fiu.edu
        Name : cs.fiu.edu       Name Server: sagwa-ns.cs.fiu.edu
        Name : cs.fiu.edu       Name Server: offsite.cs.fiu.edu
        Name : cs.fiu.edu       Name Server: zorba-ns.cs.fiu.edu
Additional Information Section:
	Name : offsite.cs.fiu.edu	IP : 131.94.68.228
	Name : sagwa-ns.cs.fiu.edu	IP : 131.94.134.129
	Name : zorba-ns.cs.fiu.edu	IP : 131.94.134.130
	Name : goedel.cs.fiu.edu	IP : 131.94.133.43

----------------------------------------------------------------
DNS server to query: 131.94.68.228
Reply received. Content overview:
	1 Answers.
	6 Intermediate Name Servers.
	4 Additional Information Records.
Answers section:
	Name : cs.fiu.edu	IP: 131.94.130.238
Authority Section:
	Name : cs.fiu.edu	Name Server: zorba-ns.v6.cs.fiu.edu
	Name : cs.fiu.edu	Name Server: goedel.cs.fiu.edu
	Name : cs.fiu.edu	Name Server: offsite.cs.fiu.edu
	Name : cs.fiu.edu	Name Server: sagwa-ns.cs.fiu.edu
	Name : cs.fiu.edu	Name Server: sagwa-ns.v6.cs.fiu.edu
	Name : cs.fiu.edu	Name Server: zorba-ns.cs.fiu.edu
Additional Information Section:
	Name : goedel.cs.fiu.edu	IP : 131.94.133.43
	Name : offsite.cs.fiu.edu	IP : 131.94.68.228
	Name : sagwa-ns.cs.fiu.edu	IP : 131.94.134.129
	Name : zorba-ns.cs.fiu.edu	IP : 131.94.134.130
```