Introduction to Web Science

Assignment 3

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Luxembourg

Submission until: November 16, 2016, 10:00 a.m.
Tutorial on: November 18, 2016, 12:00 p.m.

The main objective of this assignment is for you understand different concepts that are associated with the ”Web”. In this assignment we cover two topics: 1) DNS & 2) Internet.

These tasks are not always specific to “Introduction to Web Science”. For all the assignment questions that require you to write a code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Team Name: Echo
Team Members: Hanadi Tamimi, Keya Kashem, Md Jakaria Nawaz

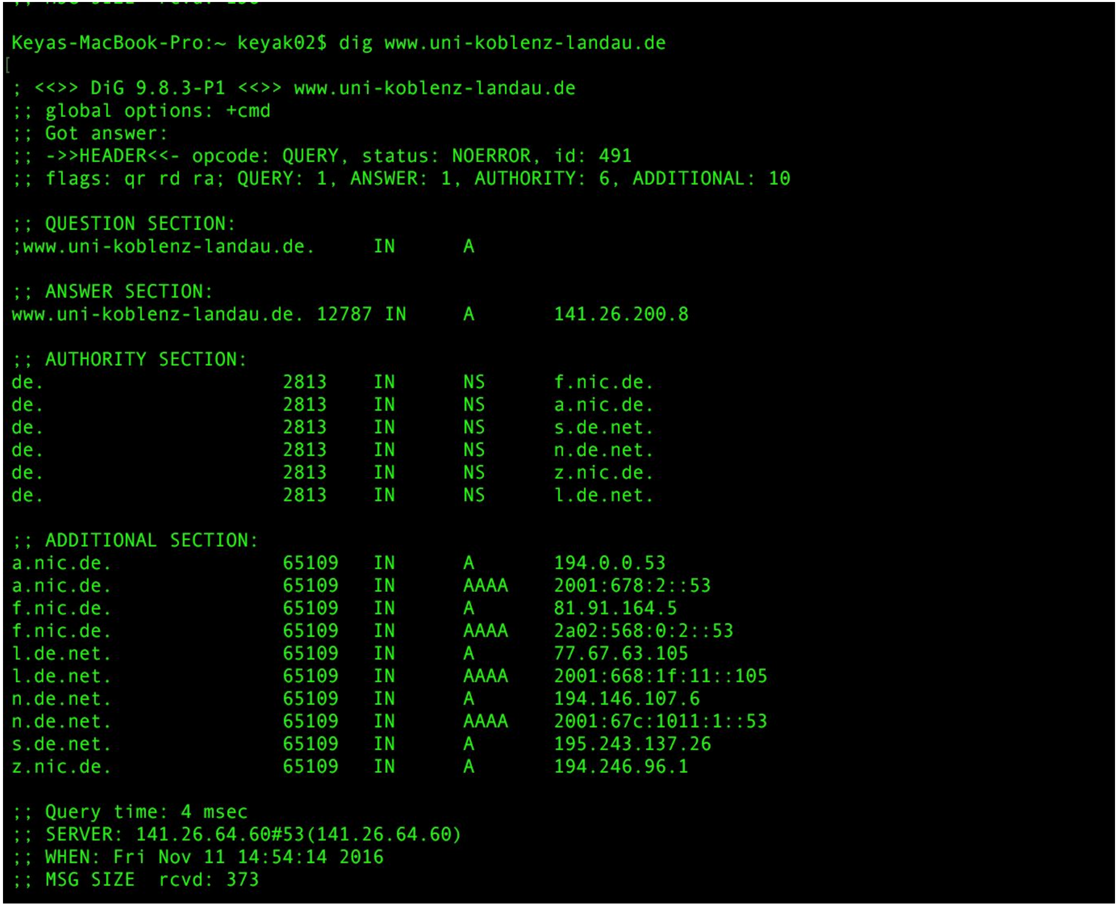
1 DIG Deeper (5 Points)

Assignment 1 started with you googling certain basic tools and one of them was ”*dig*”.

1. Now using that dig command, find the IP address of `www.uni-koblenz-landau.de`
2. In the result, you will find ”SOA”. What is SOA?
3. Copy the SOA record that you find in your answer sheet and explain each of the components of SOA with regards to your find. Merely integrating answers from the internet wont fetch you points.

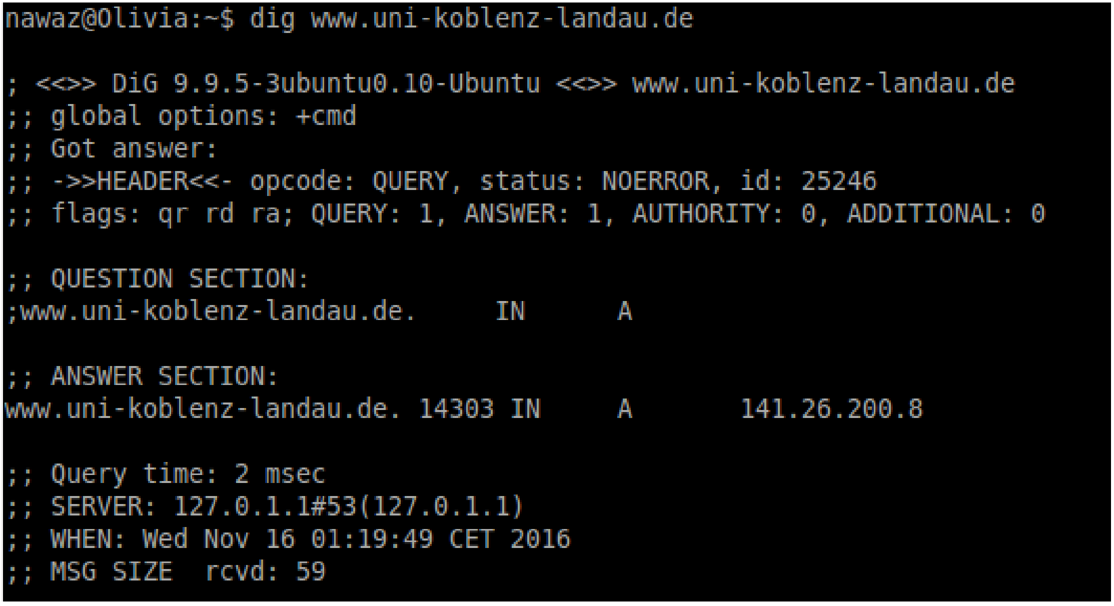
Try the experiment once from University network and once from Home network and see if you can find any differences and if so, clarify why.

Answers:
(1) IP Address for www.uni-kobelenz-landau.de is 141.26.200.8
The result is same for both network.
Network 1



```
Keyas-MacBook-Pro:~ keyak02$ dig www.uni-koblenz-landau.de
[
; <<>> DiG 9.8.3-P1 <<>> www.uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 491
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 6, ADDITIONAL: 10

;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      A

;; ANSWER SECTION:
www.uni-koblenz-landau.de. 12787 IN     A       141.26.200.8

;; AUTHORITY SECTION:
de.                     2813    IN      NS      f.nic.de.
de.                     2813    IN      NS      a.nic.de.
de.                     2813    IN      NS      s.de.net.
de.                     2813    IN      NS      n.de.net.
de.                     2813    IN      NS      z.nic.de.
de.                     2813    IN      NS      l.de.net.

;; ADDITIONAL SECTION:
a.nic.de.               65109   IN      A       194.0.0.53
a.nic.de.               65109   IN      AAAA    2001:678:2::53
f.nic.de.               65109   IN      A       81.91.164.5
f.nic.de.               65109   IN      AAAA    2a02:568:0:2::53
l.de.net.               65109   IN      A       77.67.63.105
l.de.net.               65109   IN      AAAA    2001:668:1f:11::105
n.de.net.               65109   IN      A       194.146.107.6
n.de.net.               65109   IN      AAAA    2001:67c:1011:1::53
s.de.net.               65109   IN      A       195.243.137.26
z.nic.de.               65109   IN      A       194.246.96.1

;; Query time: 4 msec
;; SERVER: 141.26.64.60#53(141.26.64.60)
;; WHEN: Fri Nov 11 14:54:14 2016
;; MSG SIZE  rcvd: 373
```

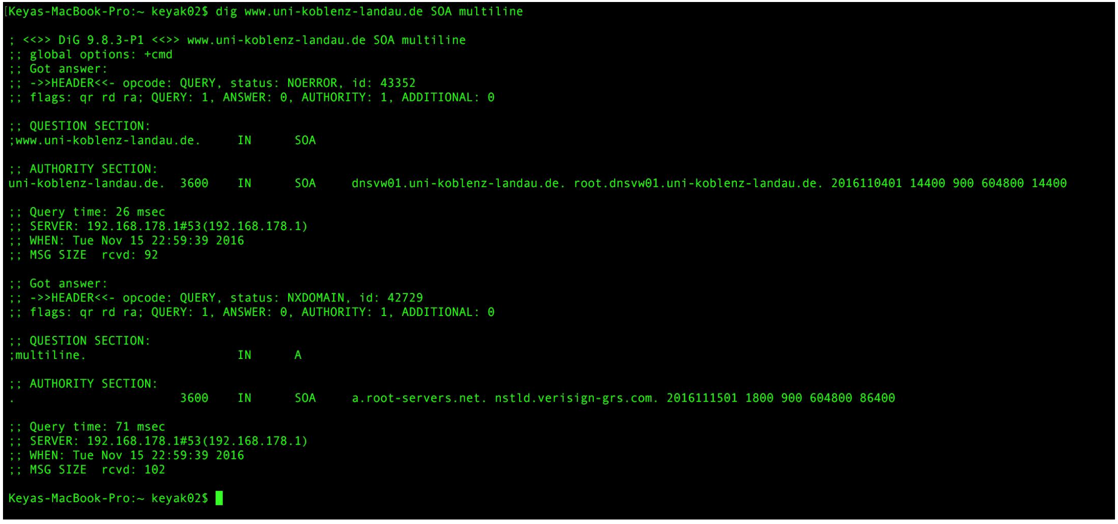
Network 2



```
nawaz@Olivia:~$ dig www.uni-koblenz-landau.de

; <<>> DiG 9.9.5-3ubuntu0.10-Ubuntu <<>> www.uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25246
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      A

;; ANSWER SECTION:
www.uni-koblenz-landau.de. 14303 IN     A       141.26.200.8

;; Query time: 2 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Nov 16 01:19:49 CET 2016
;; MSG SIZE  rcvd: 59
```

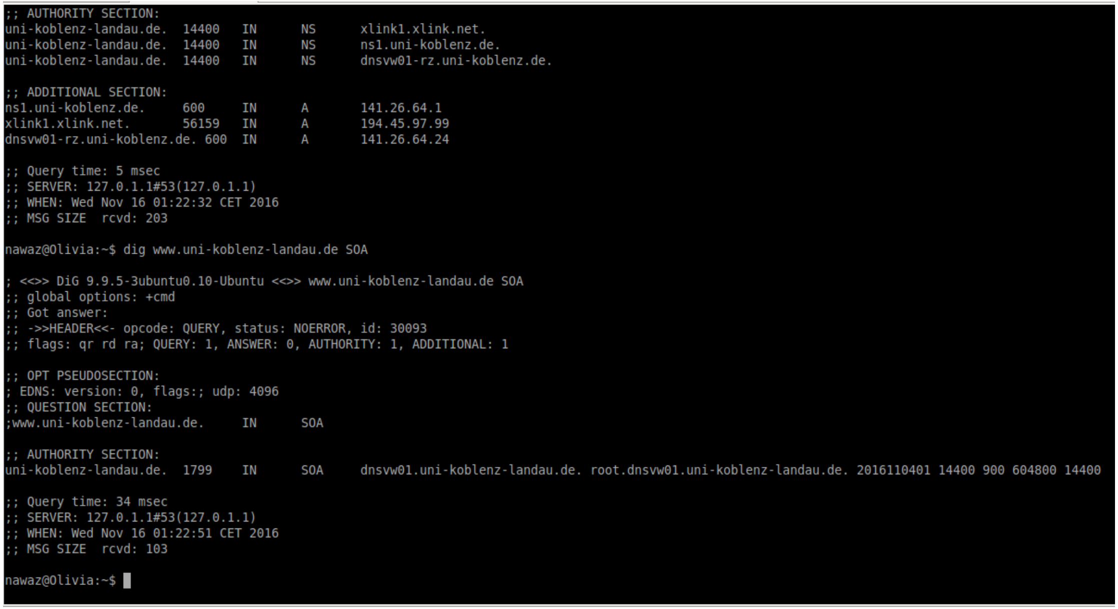
(2) SOA: The full form of SOA is Start Of Authority. An SOA record is a Start of Authority. Every domain must have a Start of Authority record at the cutover point where the domain is delegated from its parent domain. It is an important component of a zone file in Domain Name System (DNS). An SOA record contains important information for the management of the zone, in particular the zone transfer.The SOA record consists of 7 fields.

Network 1



```
[Keyas-MacBook-Pro:~ keyak02$ dig www.uni-koblenz-landau.de SOA multiline

; <<>> DiG 9.8.3-P1 <<>> www.uni-koblenz-landau.de SOA multiline
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43352
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      SOA

;; AUTHORITY SECTION:
uni-koblenz-landau.de.  3600    IN      SOA     dnsvw01.uni-koblenz-landau.de. root.dnsvw01.uni-koblenz-landau.de. 2016110401 14400 900 604800 14400

;; Query time: 26 msec
;; SERVER: 192.168.178.1#53(192.168.178.1)
;; WHEN: Tue Nov 15 22:59:39 2016
;; MSG SIZE  rcvd: 92

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 42729
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;multiline.                     IN      A

;; AUTHORITY SECTION:
.                       3600    IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2016111501 1800 900 604800 86400

;; Query time: 71 msec
;; SERVER: 192.168.178.1#53(192.168.178.1)
;; WHEN: Tue Nov 15 22:59:39 2016
;; MSG SIZE  rcvd: 102

Keyas-MacBook-Pro:~ keyak02$
```

Network 2



```
;; AUTHORITY SECTION:
uni-koblenz-landau.de.  14400   IN      NS      xlink1.xlink.net.
uni-koblenz-landau.de.  14400   IN      NS      ns1.uni-koblenz.de.
uni-koblenz-landau.de.  14400   IN      NS      dnsvw01-rz.uni-koblenz.de.

;; ADDITIONAL SECTION:
ns1.uni-koblenz.de.     600     IN      A       141.26.64.1
xlink1.xlink.net.       56159   IN      A       194.45.97.99
dnsvw01-rz.uni-koblenz.de. 600  IN      A       141.26.64.24

;; Query time: 5 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Nov 16 01:22:32 CET 2016
;; MSG SIZE  rcvd: 203

nawaz@Olivia:~$ dig www.uni-koblenz-landau.de SOA

; <<>> DiG 9.9.5-3ubuntu0.10-Ubuntu <<>> www.uni-koblenz-landau.de SOA
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30093
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.uni-koblenz-landau.de.     IN      SOA

;; AUTHORITY SECTION:
uni-koblenz-landau.de.  1799    IN      SOA     dnsvw01.uni-koblenz-landau.de. root.dnsvw01.uni-koblenz-landau.de. 2016110401 14400 900 604800 14400

;; Query time: 34 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Nov 16 01:22:51 CET 2016
;; MSG SIZE  rcvd: 103

nawaz@Olivia:~$ █
```

(3) The SOA record of www.uni-koblenz-landau.de is described below-

- Primary- The primary name server for the domain - dnsvw01.uni-koblenz-landau.de

- Mail address - The responsible party for the domain - root.dnsvw01.uni-koblenz-landau.de

- serial number- A timestamp that changes whenever the domain is updated - 2016110401

- Refresh - The number of seconds before the zone should be refreshed - 14400

- Retry - The number of seconds before a failed refresh should be retried - 900

- Expire - The upper limit in seconds before a zone is considered no longer authoritative - 604800

- TTL -How long a resolver should consider a negative result for a subdomain to be valid before retrying - 14400

2 Exploring DNS (10 Points)

In the first part of this assignment you were asked to develop a simple TCP Client Server. Now, using **that** client server setup. This time a url should be send to the server and the server will split the url into the following:

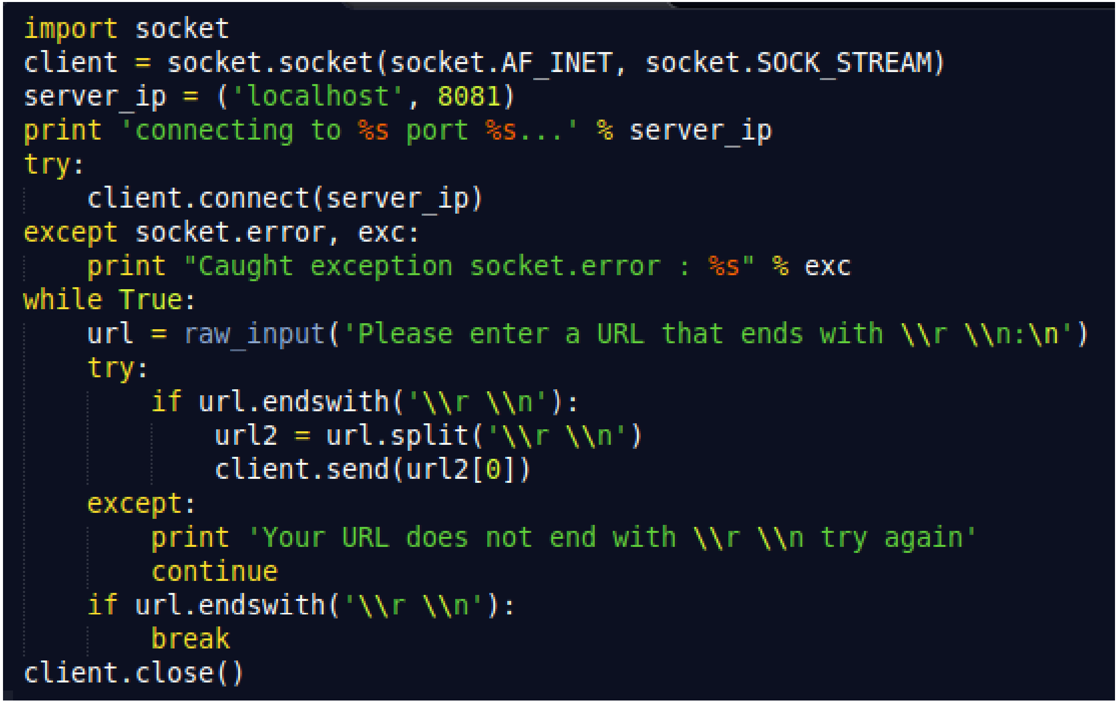
`http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#InTheDocument`

1. Protocol
2. Domain
3. Sub-Domain
4. Port number
5. Path
6. Parameters
7. Fragment

The Protocol for sending the URL will be a string terminated with \r \n.

P.S.: You are **not** allowed to use libraries like `urlparse` for this question. You will also not use ”Regular Expressions” for this.

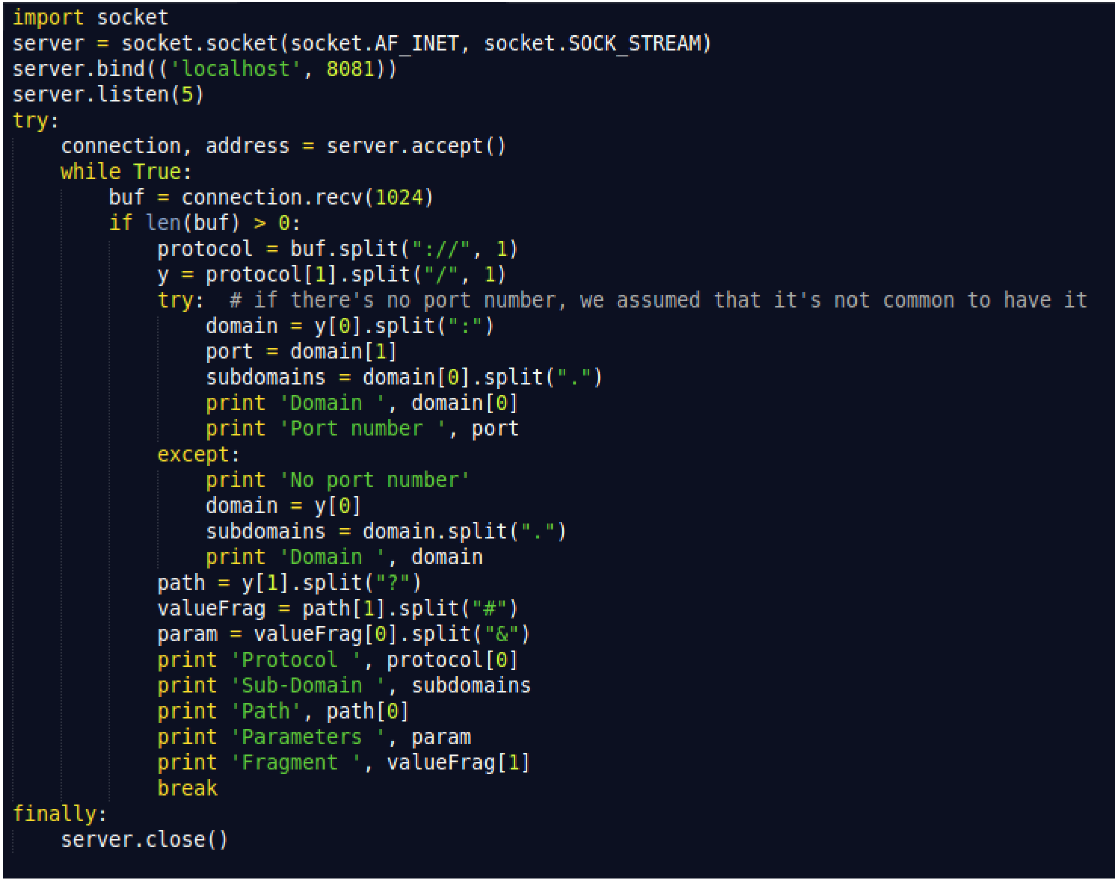
Answer: Code (client.py):



```
import socket
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_ip = ('localhost', 8081)
print 'connecting to %s port %s...' % server_ip
try:
    client.connect(server_ip)
except socket.error, exc:
    print "Caught exception socket.error : %s" % exc
while True:
    url = raw_input('Please enter a URL that ends with \\r \\n:\n')
    try:
        if url.endswith('\\r \\n'):
            url2 = url.split('\\r \\n')
            client.send(url2[0])
    except:
        print 'Your URL does not end with \\r \\n try again'
        continue
    if url.endswith('\\r \\n'):
        break
client.close()
```

Git link (client.py):
https://github.com/jakaria-nawaz/Echo/blob/master/Echo/assignment3/client.py/

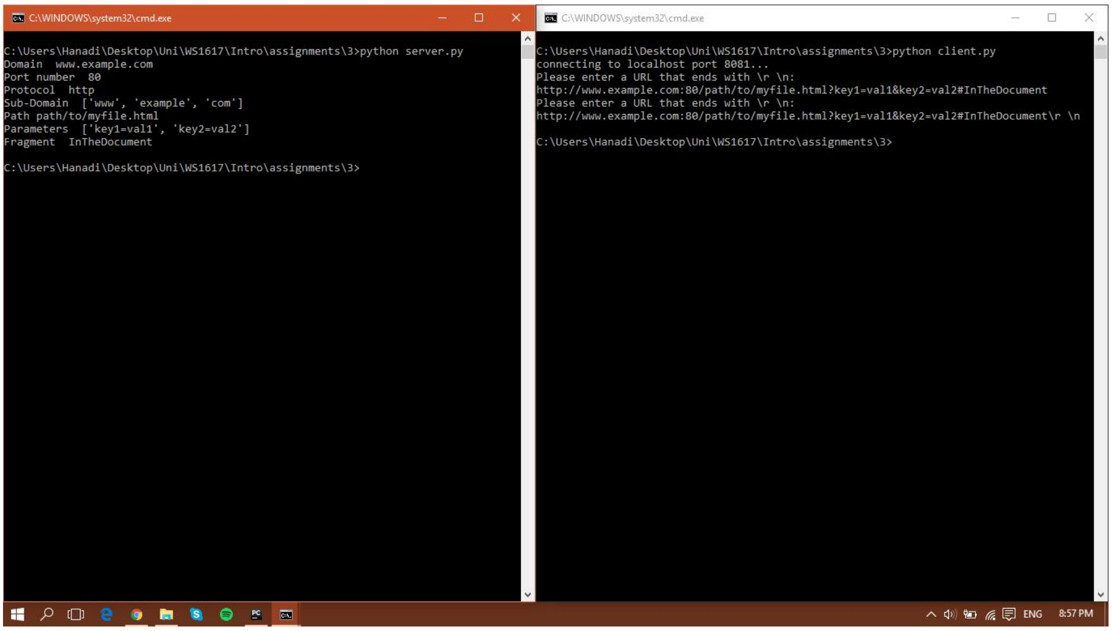
Code (server.py):



```
import socket
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(('localhost', 8081))
server.listen(5)
try:
    connection, address = server.accept()
    while True:
        buf = connection.recv(1024)
        if len(buf) > 0:
            protocol = buf.split("://", 1)
            y = protocol[1].split("/", 1)
            try:  # if there's no port number, we assumed that it's not common to have it
                domain = y[0].split(":")
                port = domain[1]
                subdomains = domain[0].split(".")
                print 'Domain ', domain[0]
                print 'Port number ', port
            except:
                print 'No port number'
                domain = y[0]
                subdomains = domain.split(".")
                print 'Domain ', domain
            path = y[1].split("?")
            valueFrag = path[1].split("#")
            param = valueFrag[0].split("&")
            print 'Protocol ', protocol[0]
            print 'Sub-Domain ', subdomains
            print 'Path', path[0]
            print 'Parameters ', param
            print 'Fragment ', valueFrag[1]
            break
finally:
    server.close()
```

Git link (server.py):
https://github.com/jakaria-nawaz/Echo/blob/master/Echo/assignment3/server.py/

Result(Screenshot):



```
C:\Users\Hanadi\Desktop\Uni\WS1617\Intro\assignments\3>python server.py
Domain  www.example.com
Port number  80
Protocol  http
Sub-Domain  ['www', 'example', 'com']
Path path/to/myfile.html
Parameters  ['key1=val1', 'key2=val2']
Fragment  InTheDocument

C:\Users\Hanadi\Desktop\Uni\WS1617\Intro\assignments\3>
```

```
C:\Users\Hanadi\Desktop\Uni\WS1617\Intro\assignments\3>python client.py
connecting to localhost port 8081...
Please enter a URL that ends with \r \n:
http://www.example.com:80/path/to/myfile.html?key1=val1&key2=val2#InTheDocument
Please enter a URL that ends with \r \n:
http://www.example.com:80/path/to/myfile.html?key1=val1&key2=val2#InTheDocument\r \n

C:\Users\Hanadi\Desktop\Uni\WS1617\Intro\assignments\3>
```

3 DNS Recursive Query Resolving (5 Points)

You have solved the ”Routing Table” question in Assignment 2. We updated the routing tables once more. resulting in the following tables creating the following topology

Table 1: Routing Table

Router1				**Router2**				**Router3**		
Destination	Next Hop	Interface		Destination	Next Hop	Interface		Destination	Next Hop	Interface
67.0.0.0	67.68.3.1	eth 0		205.30.7.0	205.30.7.1	eth 0		205.30.7.0	205.30.7.2	eth 0
62.0.0.0	62.4.31.7	eth 1		156.3.0.0	156.3.0.6	eth 1		88.0.0.0	88.6.32.1	eth 1
88.0.0.0	88.4.32.6	eth 2		26.0.0.0	26.3.2.1	eth 2		25.0.0.0	25.03.1.2	eth 2
141.71.0.0	141.71.20.1	eth 3		141.71.0.0	141.71.26.3	eth 3		121.0.0.0	121.0.3.1	eth 3
26.0.0.0	141.71.26.3	eth3		67.0.0.0	141.71.20.1	eth 3		156.3.0.0	205.30.7.1	eth 0
156.3.0.0	88.6.32.1	eth 2		62.0.0.0	141.71.20.1	eth 3		26.0.0.0	205.30.7.1	eth 0
205.30.7.0	141.71.26.3	eth 3		88.0.0.0	141.71.20.1	eth 3		141.71.0.0	205.30.7.1	eth 0
25.0.0.0	88.6.32.1	eth 2		25.0.0.0	205.30.7.2	eth 0		67.0.0.0	88.4.32.6	eth 1
121.0.0.0	88.6.32.1	eth 2		121.0.0.0	205.30.7.2	eth 0		62.0.0.0	88.4.32.6	eth 1

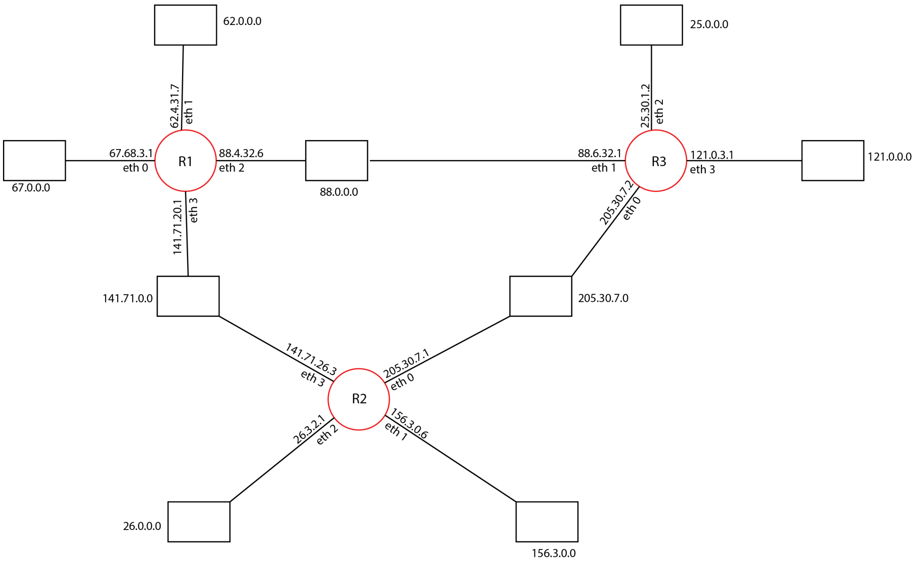


Figure 1: DNS Routing Network

Let us asume a client with the following ip address 67.4.5.2 wants to resolve the following domain `subdomain.webscienceexampledomain.com` using the DNS.

You can further assume the root name server has the IP address of 25.8.2.1 and the name-server for `webscienceexampledomain.com` has the IP address 156.3.20.2. Finally the sub-domain is handled by a name server with the IP of 26.155.36.7.

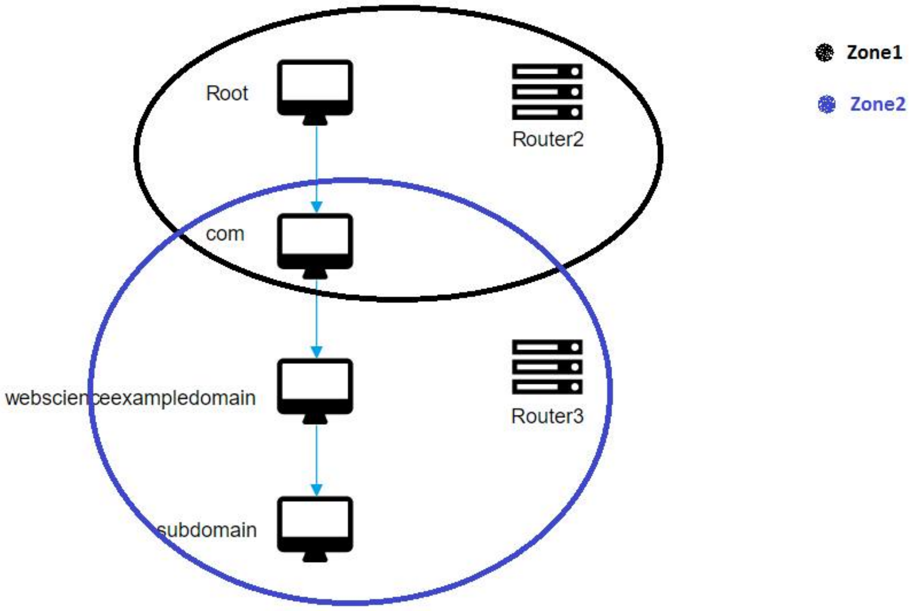
Please explain how the traffic flows through the network in order to resolve the recursive DNS query. You can assume ARP tables are cached so that no ARP-requests have to be made.

Hint: You can start like this:

67.4.5.2 creates an IP packet with the source address XXXXXX an destination address YYYYY inside there is the DNS request. This IP packet is send as an ethernet frame to ZZZZZ. ZZZZZ receives the frame and forwards the encapsulated IP packet to ....

Also you can assume the DNS requests and responses will fit inside one IP packet. You also don't have to write down the specific DNS requests and responses in hex.

Answer:
In this solution we have followed what was explained in the video of "DNS address resolution" and mixed it with the routing table algorithm that we learnt in previous weeks.
Therefore, we are assuming that (Router1), which is resposible for the network that the client is connected to, to be the ISP.
The ISP is going to send all the requests and receive them until it receives the final correct response then sends it to the client.
Moreover, we are assuming (Router2) and (Router3) to be DNS servers.
We are also assuming that "webscienceexampledomain" and "com" is in a zone (Zone2) with the "subdomain", and that zone is considered to be handled by (Router2); since (Router2) is responsible for both of them according to the routing tables.
Also, there's another assumed zone (Zone1 that includes the "root server" and the ".com" domain which is handled by (Router3). We added ".com" in the zone to make it overlap with the whole scenario. Since the question also hasn't given us an IP for the .com domain.



As client 67.4.5.2 wants to resolve the domain ”subdomain.webscienceexampledomain.com” using DNS. First, the client will issue a request to the ISP, asking for the address of the domain ”subdomain.webscienceexampledomain.com”.
The ISP will forward that request to the root server with IP 25.8.2.1 that exists under (Router3). (Router3) as a DNS will know that it doesn’t have that domain in its networks. Therefore it will send back a response to the ISP, in that response will be the IP address of the DNS that the ISP should send the request to, which is now 156.3.20.2
The ISP sends a request to the ”webscienceexampledomain.com” with the IP 156.3.20.2, which is under (Router2). (Router2) as a DNS will be asked for the address of the domain ”subdomain.webscienceexampledomain.com”. (Router2) (DNS) knows that this domain lies in its network and therefore is able to resolve its IP address. (Router2) will send back a response to the ISP with the IP address of the requested domain.
The ISP sends back a response to the client 67.4.5.2 with the IP address of the domain ”subdomain.webscienceexampledomain.com”
There is another description given hop by hop means how the request will travel, which hop after which hop. It is uploaded in the git as text file.
Here is the link:
https://github.com/jakaria-nawaz/Echo/blob/master/Echo/assignment3/hop-description(quses3).txt/

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment3/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of ”Web Information Retrieval”.

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.