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: India
Group Member : Jalpa Patel, Amani Gaddamedi, Jasvindar Kaur

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 of `www.uni-koblenz-landau.de` is **141.26.200.8**

```
jp@Jappi-PC:~$ dig www.uni-koblenz-landau.de

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.uni-koblenz-landau.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36428
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 6, ADDITIONAL: 11

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

;; ANSWER SECTION:
www.uni-koblenz-landau.de. 14267 IN     A       141.26.200.8

;; AUTHORITY SECTION:
de.                     2447    IN      NS      f.nic.de.
de.                     2447    IN      NS      s.de.net.
de.                     2447    IN      NS      n.de.net.
de.                     2447    IN      NS      a.nic.de.
de.                     2447    IN      NS      l.de.net.
de.                     2447    IN      NS      z.nic.de.

;; ADDITIONAL SECTION:
a.nic.de.               2447    IN      A       194.0.0.53
a.nic.de.               2447    IN      AAAA    2001:678:2::53
f.nic.de.               2447    IN      A       81.91.164.5
f.nic.de.               2447    IN      AAAA    2a02:568:0:2::53
l.de.net.               2447    IN      A       77.67.63.105
l.de.net.               2447    IN      AAAA    2001:668:1f:11::105
n.de.net.               2447    IN      A       194.146.107.6
n.de.net.               2447    IN      AAAA    2001:67c:1011:1::53
s.de.net.               2447    IN      A       195.243.137.26
z.nic.de.               2447    IN      A       194.246.96.1

;; Query time: 1 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Nov 14 16:15:46 CET 2016
;; MSG SIZE  rcvd: 384
```

Figure 1: IP address using dig Command

2. **SOA** : An SOA(State of Authority) Record is the most essential part of a Zone file. The SOA record is a way for the Domain Administrator to give out simple information about the domain like, how often it is updated, when it was last updated, when to check back for more info, what is the admins email address and so on. A Zone file can contain only one SOA Record.

3. SOA record

Figure 2: SOA record in University Network

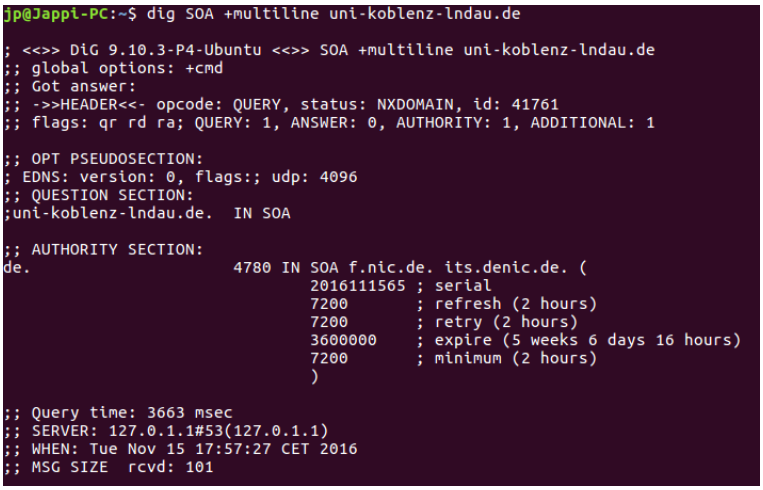

Figure 3: SOA record in Home Network

Following is description of SOA records in Uni network :

Figure 4: SOA record in Home Network

SOA Records		
Mname	dnsvw01.uni-koblenz-landau.de	This defines the primary master name server for this domain
Rname	root.dnsvw01.uni-koblenz-landau.de	This is the email address of the administrator for this zone.
Serial	2016110401	This field shows how many times the zone has been updated.The recommended value for this is a 10-digit number in the form YYYYMMDDnn (year, month, date, revision).
Refresh	14400	This is the amount of time that the slave DNS server will wait before polling the master for zone file changes.
Retry	900	The number of seconds that the primary name server(s) should wait, if an attempt to refresh failed, before making another attempt to refresh.
Expire	604800	The number of seconds that lets the secondary name server(s) know how long they can hold the information before it is no longer considered authoritative.
Minimum (TTL)	14400	This is the amount of time that the name server will cache a name error if it cannot find the requested name in this file.
Name	uni-koblenz-landau.de	This is the root of the zone. This specifies that the zone file is for the domain.
Type	SOA	The SOA is the indicator that this is a Start of Authority record.
Class	IN	The ”IN” portion means internet
TTL	3940	It is basically a timer. A caching name server can use previously queried results to answer questions until the TTL value runs out.

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:

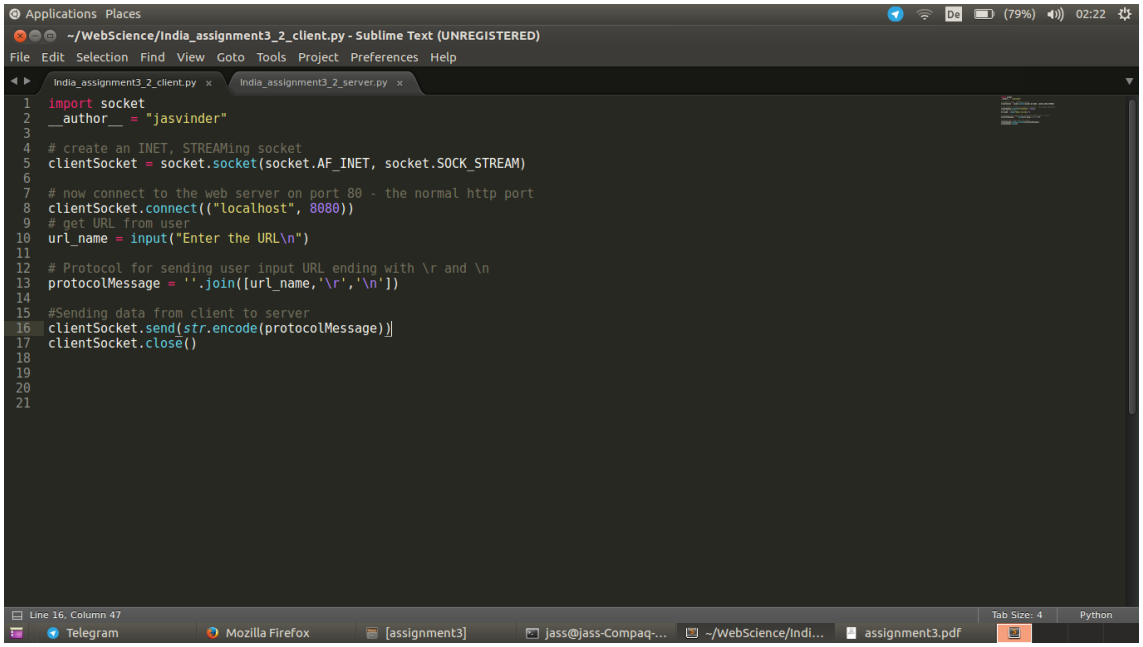

Figure 5: Client code

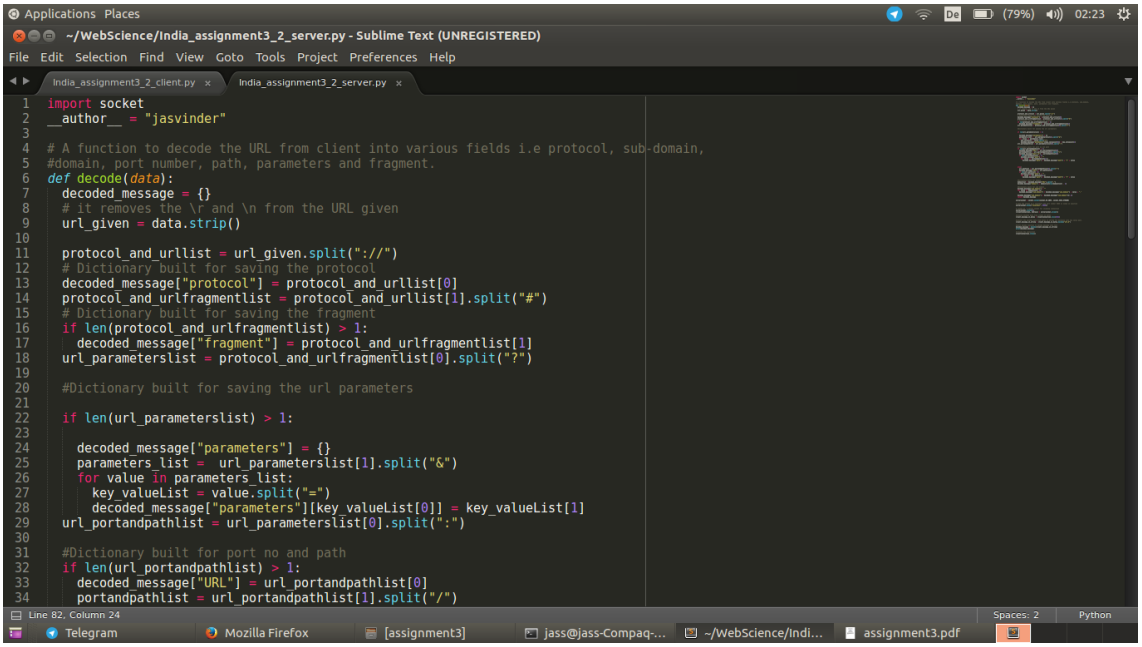

Figure 6: Server code Part 1

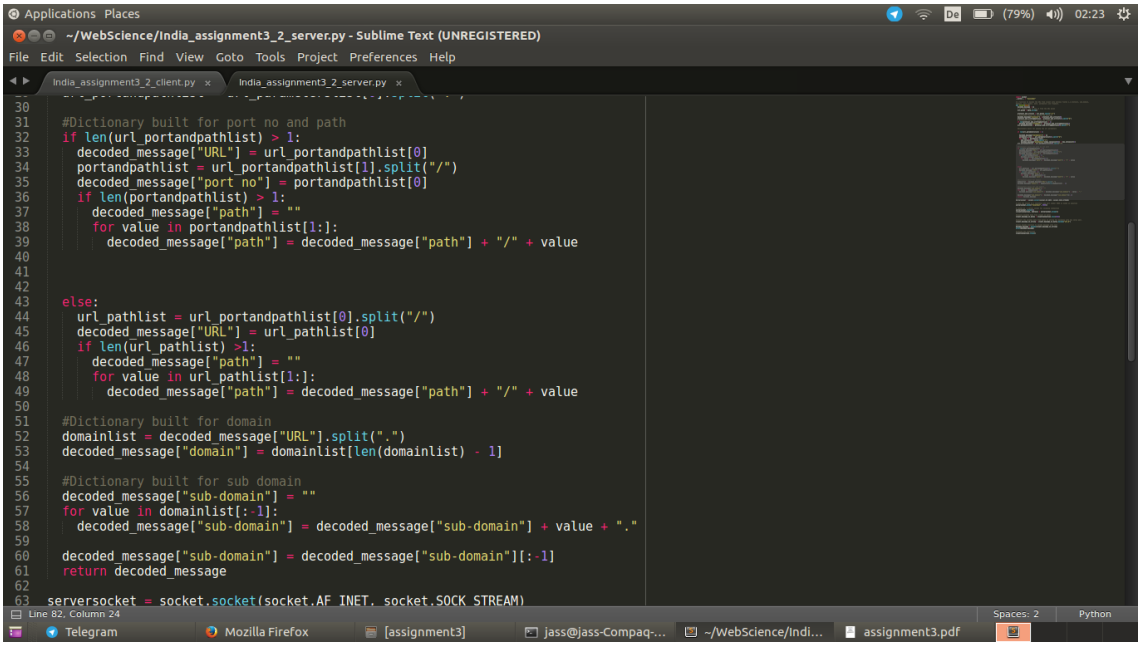

Figure 7: Server code Part 2

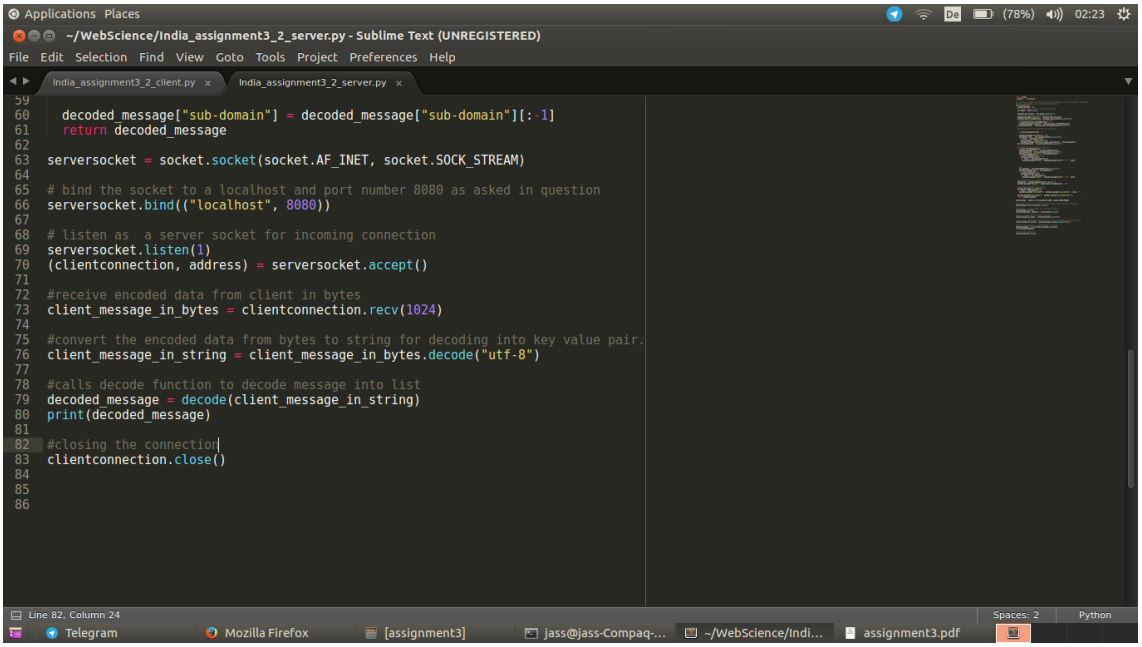

Figure 8: Server code Part 3

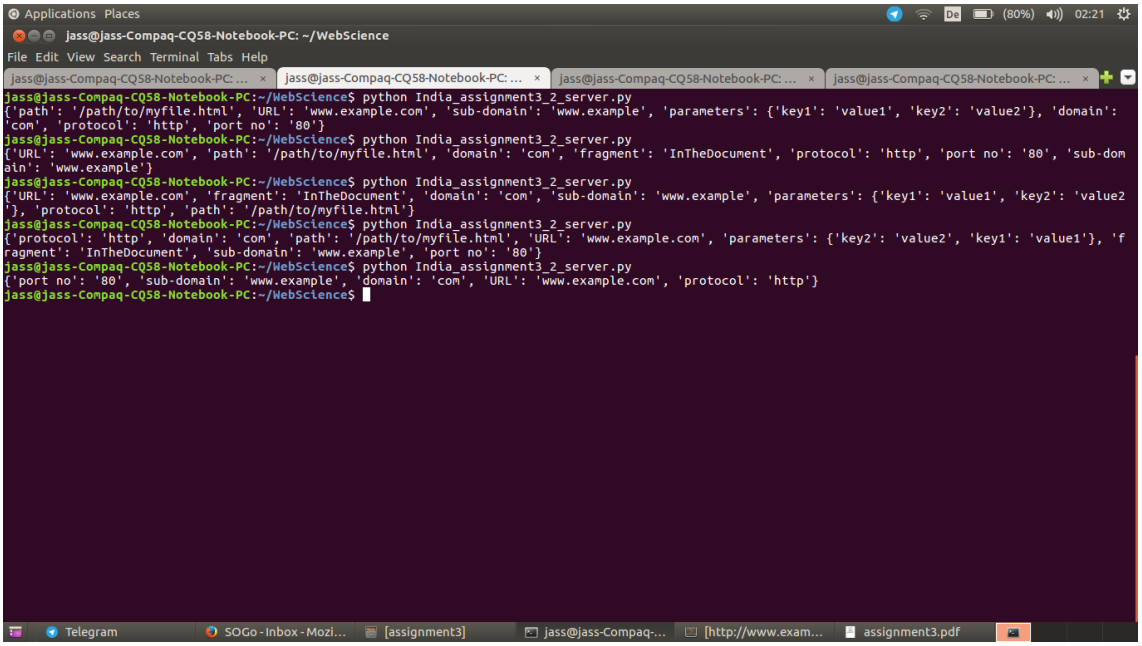

Figure 9: Server output

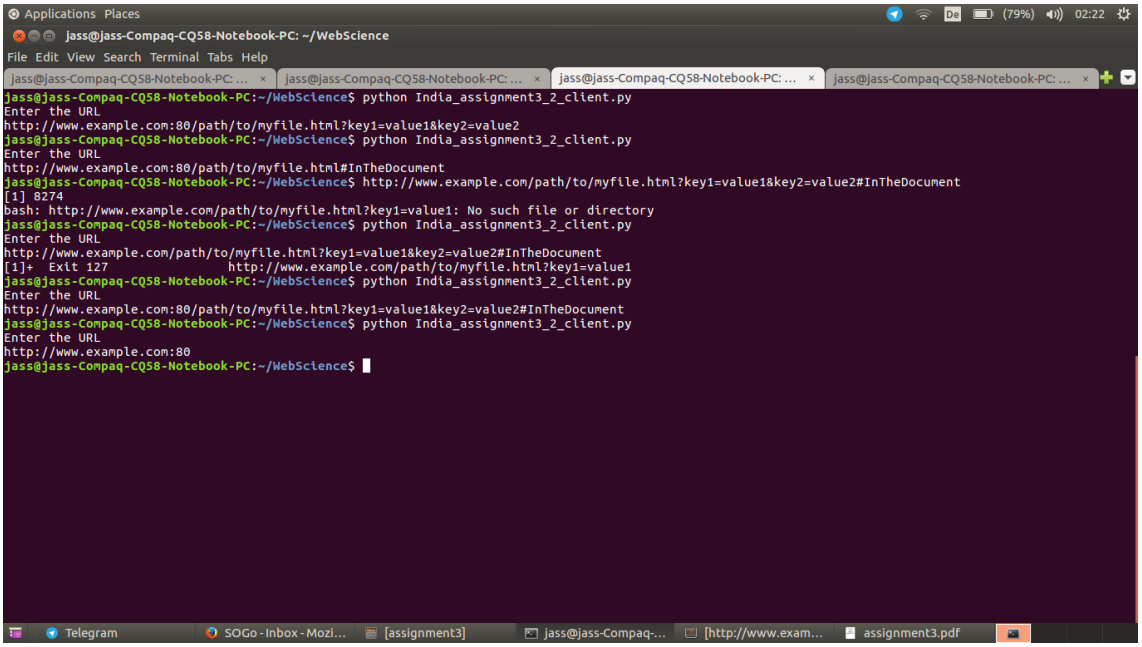

Figure 10: Client output

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

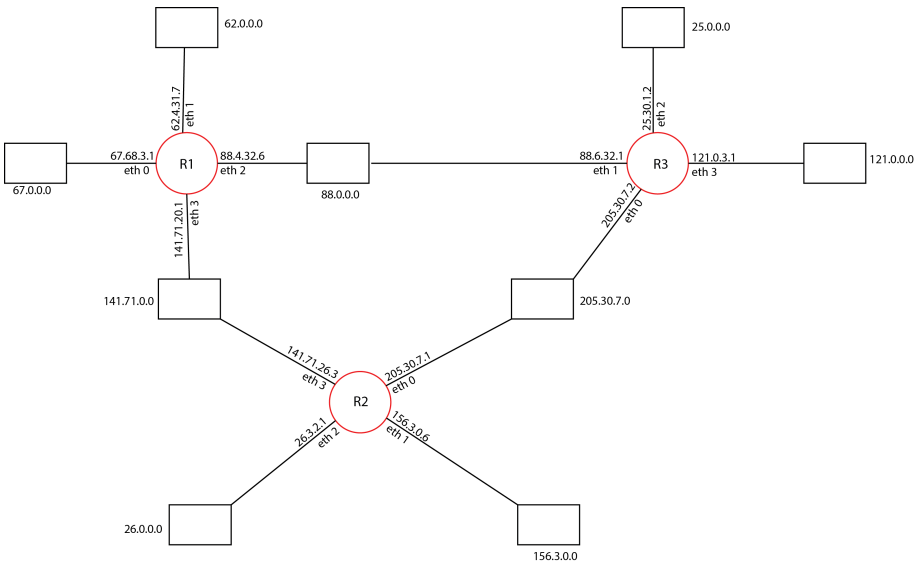

Figure 11: DNS Routing Network

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:

The client with the ip address 67.4.5.2 sends the request to the root with ip address 25.8.2.1 requesting the address of subdomain.webscienceexampledomain.com the packet is shown in fig a .
this request goes from 67.0.0.0 to r1 to 88.0.0.0 to r3 to 25.0.0.0 network

The root respose back with the address of webscienceexampledomain.com with ip address as 156.3.20.2 packet is shown in fig -b

the Client sends the request to the webscienceexampledomain.com requesting the address of subdomai packet is shown in fig-c
this request goes from 67.0.0.0 to r1 to 141.71.0.0 to r2 to 156.3.0.0 network

The webscienceexampledomain.com sends the subdomain address as response as 26.155.36.7 this is shown in fid-d

the client sends request to subdomainwebscienceexampledomain.com of ip 26.155.36.7 from the 67.0.0.0 to r1 to141.71.0.0 to r2 to 26.0.0.0 packet network shown in figure e

The recursive DNS Query for the given condition is shown in following figure:

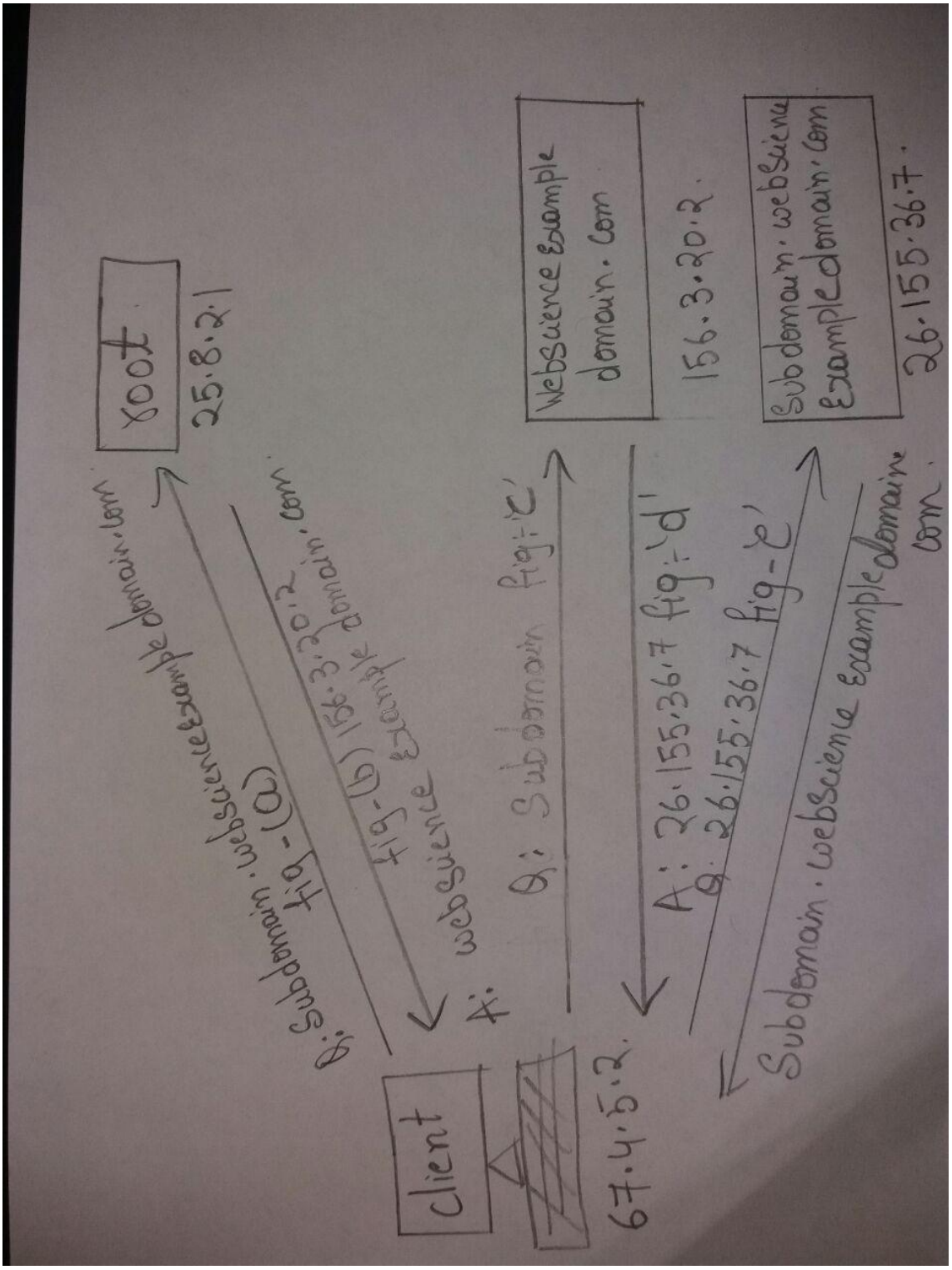

Figure 12: The Recursive query flow

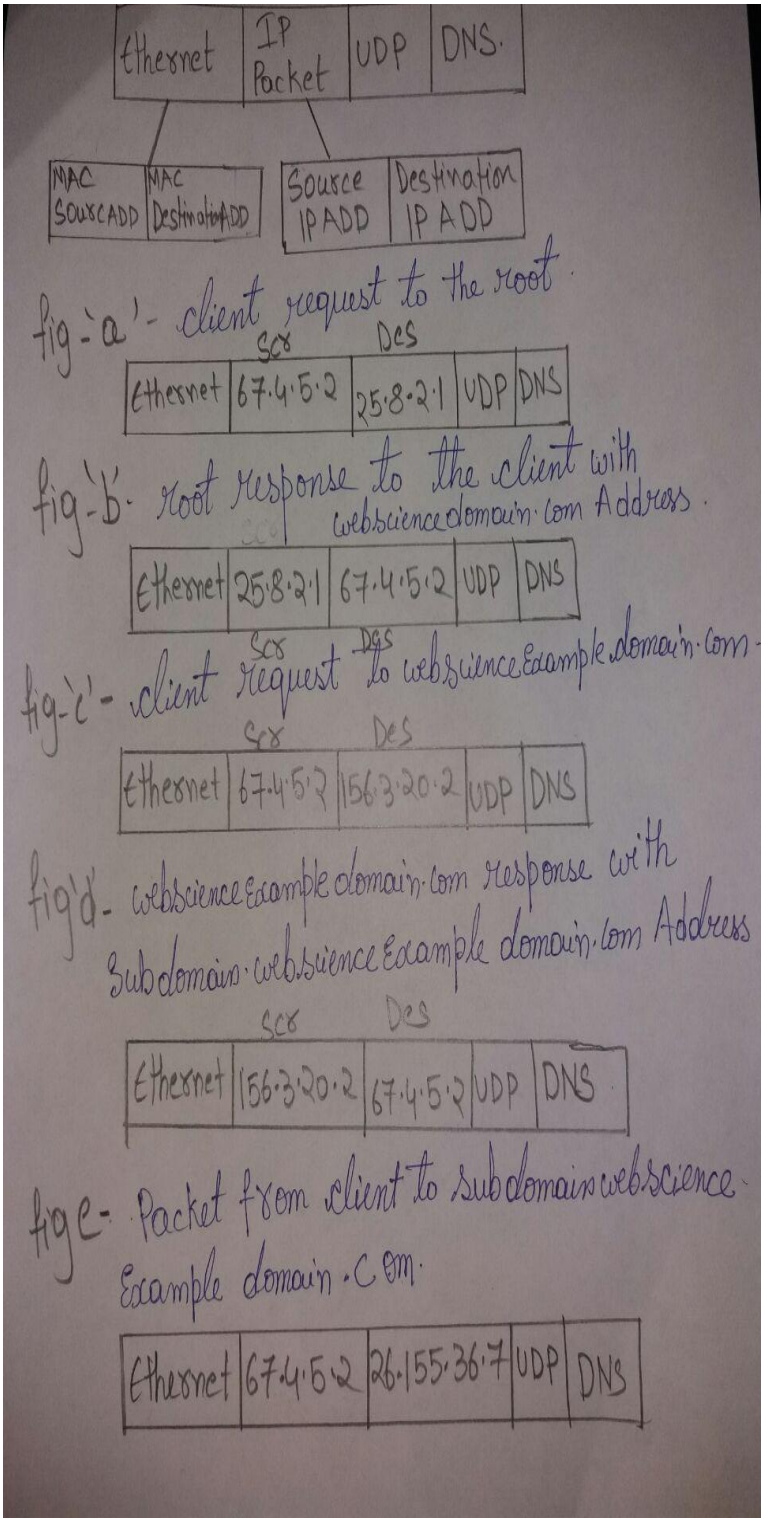

Figure 13: The packet structure with the query

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`.