

# Introduction to Web Science

## Assignment 3

PD Dr. Matthias Thimm

[thimm@uni-koblenz.de](mailto:thimm@uni-koblenz.de)

Ipek Baris Schlicht

[ibaris@uni-koblenz.de](mailto:ibaris@uni-koblenz.de)

Kenneth Skiba

[kennethskiba@uni-koblenz.de](mailto:kennethskiba@uni-koblenz.de)

Institute of Web Science and Technologies

Department of Computer Science

University of Koblenz-Landau

Submission until: 01.12.2020, CEST 23:59

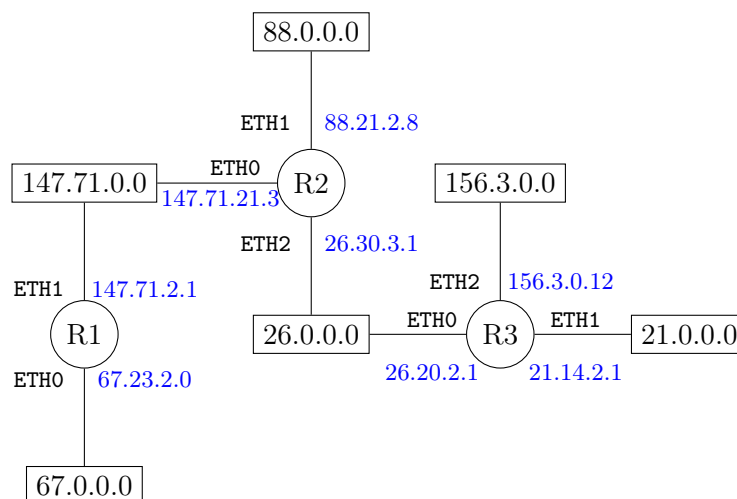
# 1 Recursive Query: DNS

20 Points

For this task we will extend the routing table from the fourth task on Assignment 1.

In the following schema rectangles represent the networks, with there name inside. The circles are the routers. An edge between a router and a network means, that a router is part of this network and has the MAC address written at the edge in blue, while the interface is written in black.

In the routing table below you find an entry for every router. One entry in the routing table of a router contains a three tuple of Destination, Next Hop and Interface.



**Figure 1:** Routing schematic representation

Router 1			Router 2			Router 3		
Destination	Next Hop	Interface	Destination	Next Hop	Interface	Destination	Next Hop	Interface
67.0.0.0	67.23.2.0	eth0	147.71.0.0	147.71.21.3	eth0	26.0.0.0	26.20.2.1	eth0
147.71.0.0	147.71.2.1	eth1	88.0.0.0	88.21.2.8	eth1	21.0.0.0	21.14.2.1	eth1
88.0.0.0	147.71.21.3	eth1	26.0.0.0	26.30.2.1	eth2	88.0.0.0	26.30.3.1	eth0
26.0.0.0	147.71.21.3	eth1	67.0.0.0	147.71.2.1	eth0	147.71.0.0	26.30.3.1	eth0
21.0.0.0	147.71.21.3	eth1	21.0.0.0	26.20.2.1	eth2	67.0.0.0	26.30.3.1	eth0
156.3.0.0	147.71.21.3	eth1	156.3.0.0	26.20.2.1	eth2	156.3.0.0	156.3.0.12	eth2

**Table 1:** Routing table

Let us assume a client with the following IP address 67.4.5.2 wants to resolve the following domain `subdomain.webscienceexample.com` using the DNS.

You can further assume the root name server has the IP address of 88.8.2.1 and the name-server for `com` has the IP address 156.3.20.2. Finally the domain is handled by a name server with the IP of 21.155.36.7.

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.

**Solution:**

1. The client with IP address **67.4.5.2** reaches the router **R1** in order to issue a **DNS request** to the root name server with the destination IP address **88.8.2.1**. Now the router **R1** looks into its routing table and finds the next hop to be **147.71.2.3** in order to reach the network **88.0.0.0** and reaches the Router **R2**. Now since router **R2** is directly connected to the network **88.0.0.0**, it delivers the **IP packet** requesting the IP address of **subdomain.webscienceexample.com** to the root name server at **88.8.2.1**.
2. The root name server responds with the referral to top level domain **.com** with IP address **156.3.20.2**. Now this IP packet is routed back from the destination **88.8.2.1** to client at **67.4.5.2**. This packet from router **R2** reaches router **R1** with the next hop **147.71.2.1**. Since the network **67.0.0.0** is directly connected to router **R1**, the packet gets delivered to the client at **67.4.5.2**.
3. The client sends another DNS request to the name server with IP **156.3.20.2**. This IP packet reaches router **R1**. The router repeats the process of looking into its table and routes the packet to router **R2** through next hop **147.71.2.3**. Now the router 2 looks into its table and routes the packet to router **R3** through next hop **26.20.2.1**. Router **R3** delivers the IP packet to the name server at destination **156.3.20.2**.
4. The name server now responds with an IP packet consisting of the address of the name server **webscienceexample.com** to the client which is now acting as the destination at **67.4.5.2**. This packet is routed from **R3** to router **R2** with the hop **26.30.3.1**. The packet from router **R2** reaches router **R1** through next hop **147.71.2.1**. The router **R1** delivers the packet to the client.
5. Now the client sends the request to the name server **webscienceexample.com** at destination with IP **21.155.36.7**. This packet is routed from **R1** to **R2** through next hop **147.71.2.3**. The packet is routed from **R2** to **R3** through next hop **26.20.2.1**. The request is delivered to the name server through **R3**.
6. The name server sends an IP packet with requested information to the client at destination **67.4.5.2**. This packet is routed from **R3** to router **R2** with the hop **26.30.3.1**. The packet from router **R2** reaches router **R1** through next hop **147.71.2.1**. The router **R1** delivers the packet to the client.

## 2 Internet Architecture

**20 Points**

1. Explain in your own words the four layer of the Internet architecture.
2. Formulate an example to show the usage of the aforementioned layers.

### 3 World Wide Web

**14 Points**

1. Name two different motivation points for the construction of the world wide web and explain why these points were necessary.
2. Describe five design principles of the world wide web with your own words.

## 4 Python Programming

**26 points**

### 4.1 URL Parser

**13 points**

Write a Python script called as `urlparser.py`. The script parses an url into the segments that are explained in the lecture **Internet vs WWW**, and additionally extracts top-level domains as one segment <sup>1</sup>. When you execute the script (e.g `python -m urlparser https://west.uni-koblenz.de/studying/ws2021`) at the command-line, a dictionary containing the url and its segments should be returned. For the optional parts, you may use `None` values.

Take a screenshot of the terminal output of your script for the following URLs

1. `https://www.facebook.com/photo.php?fbid=2068026323275211&set=a.269104153167446&type=3&theater`
2. `http://www.blog.google.uk:1000/path/to/myfile.html?key1=value1&key2=value2#InTheDocument`
3. `https://www.overleaf.com/9565720ckjijuhzpbccsd#/347876331/`
4. `ftp://root@west.uni.koblenz.de`
5. `https://west.uni-koblenz.de/studying/ws2021`

You are not allowed to use any specific libraries that help in url parsing and regular expressions.

### 4.2 Simple HTTP Web Client

**13 points**

You are asked to write a simple HTTP client (`httpclient.py`) that takes a URL and is able to download that webpage from the World Wide Web and store it on your hard drive (in the same directory as your python code is running). The program should also print out the complete HTTP header of the response and store the header in a *separate file*.

You are allowed to use 1) socket, 2) `urlparser.py` that you implement for the Question 4.1 3) sys for reading input from the command-line.

Run your code on the following urls. Analyse the responses: (1) what is the response code (2) explain briefly why does the website respond with that code?

1. `http://example.com`
2. `http://example.com/test.html`
3. `https://west.uni-koblenz.de/research/projects`

**Warning:** Please don't make consecutive calls to the websites.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Top-level\\_domain](https://en.wikipedia.org/wiki/Top-level_domain)