Introduction to Web Science

Assignment 3

PD Dr. Matthias Thimm
thimm@uni-koblenz.de

Ipek Baris Schlicht
ibaris@uni-koblenz.de

Kenneth Skiba
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

Team: Bravo
Members:
Gaurav Kumar (220200656)
Pavithree Shetty (220200661)
Nisha Sharma (220202359)

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.

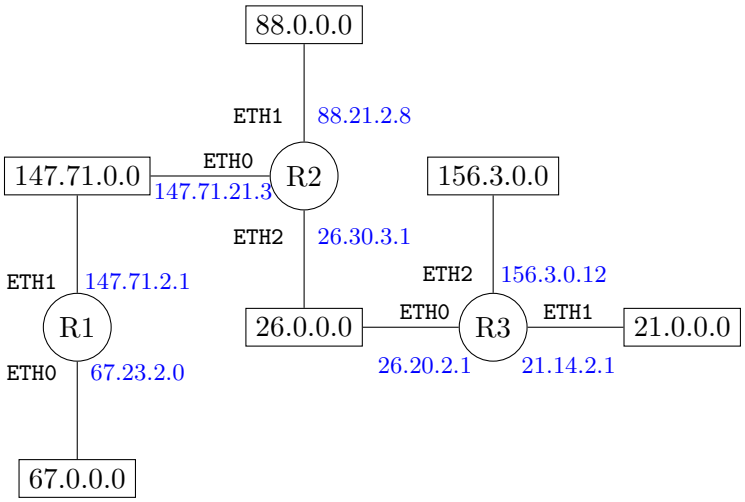

Figure 1: Routing schematic representation

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.

1.1 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 a 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 **147.71.21.3**. 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 **147.71.21.3**. 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.

2.1 Four layers of the Internet Protocol suite:

a) **First Layer - Link Layer**

- It is a protocol or group of methods that operate only on host's link.
- The link is the physical and logical network component used to interconnect hosts or nodes in the network.
- For example, how ethernet worked on a Local area network.

b) **Second Layer - Internet Layer**

- It is a group of internetworking methods, protocols and specifications that are used to transport datagrams or packets from the originating host across network boundaries, if possible to the destination host specified by a network address or IP address.
- It derives its name from function of facilitating internetworking, connecting multiple networks with each other through gateways and routers.

c) **Third Layer - Transport Layer**

- It provides end-to-end communication services for applications within a layered architecture of network components and protocols.
- It also provides convenient services such as connection-oriented data stream support, reliability, flow control, and multiplexing.

d) **Fourth Layer - Application Layer**

- It is an abstraction layer reserved for process-to-process communications across an Internet Protocol computer network.
- It uses transport layer protocols to establish process-to-process connections via ports. For example, Domain Name System, Simple Mail Transfer Protocol, etc.

REFERENCE: https://commons.wikimedia.org/wiki/File:Data_Flow_of_the_Internet_Protocol_Suite.PNG

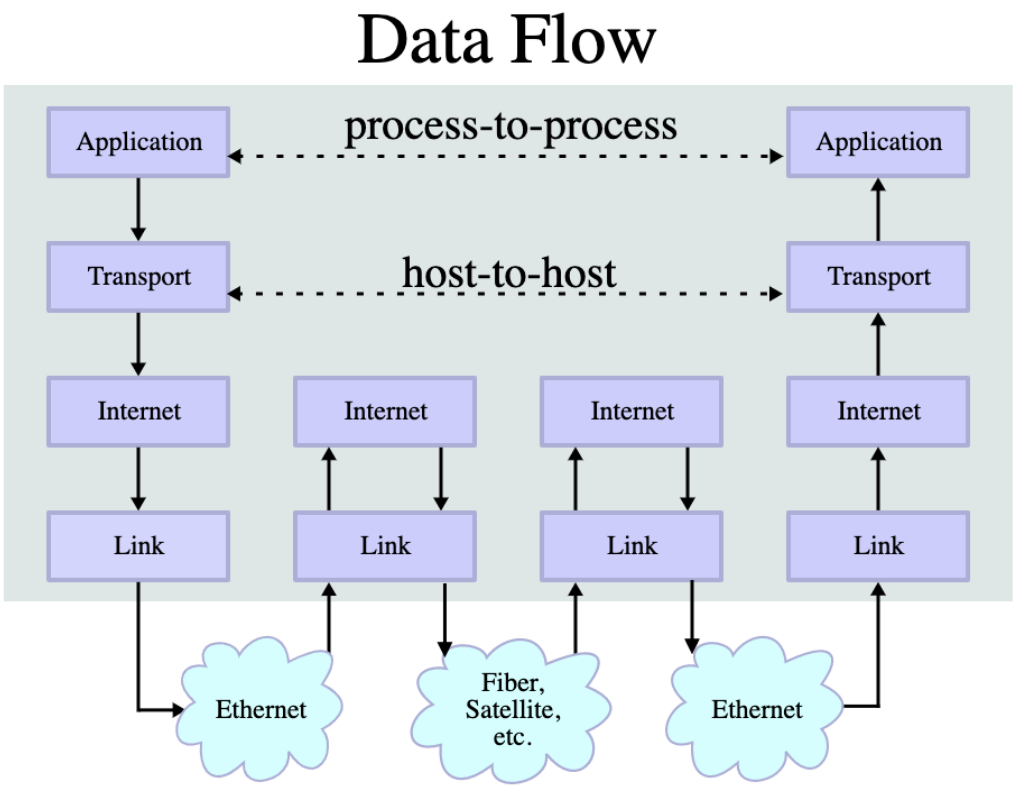

Figure 2: Data Flow in the four layers

2.2 Example to show usage of the four layers:

Let's suppose we're using Skype on a laptop. We're messaging our friend, who's using Skype on their phone from a different network.

- Skype, as a network-connected application, uses **Layer 4 (Application)** protocols like Telnet. If we send our friend a picture of our cat, Skype would be using the File Transfer Protocol (FTP).
- **Layer 3 (Transport)** receives data from Layer 4 and segments it. Each segment, or data unit, has a source and destination port number, as well as a sequence number. The port number ensures that the segment reaches the correct application. The sequence number ensures that the segments arrive in the correct order.
- **Layer 2** (Internet) transmits data segments between networks in the form of packets. When we message our friend, this layer assigns source and destination IP addresses to the data segments. Our IP address is the source, and our

friend's is the destination. Layer 2 also determines the best paths for data delivery.

- **Layer 1 (Link)** receives packets from Layer 2. Whereas Layer 3 performs logical addressing (IPv4, IPv6), Layer 1 performs physical addressing. It adds sender and receiver MAC addresses to the data packet to form a data unit called a frame. Layer 1 enables frames to be transported via local media (e.g. copper wire, optical fiber, or air). This layer is embedded as software in our computer's Network Interface Card (NIC). In short, Layer 1 allows the upper network layers to access media, and controls how data is placed and received from media.

3 World Wide Web 14 Points

3.1 Two different motivation points for the construction of the world wide web

a) **Decentralize Network**

- Earlier every data was organized and transmitted through a centralised system.
- Slowed the process of transmission and information exchange between two host.
- Not everyone had the freedom to contribute for the development of this system.
- To give Power to people rather than only a central admistration.

b) **Exchange of information**

- The major issue faced while data transmission was the exchange of information was very restricted due to various contraints.
- Both host was required to have same system which were compatible with each other for data transmission.
- Both host was required to have same operating system, software etc for successful data transmission.
- To create a network which is independent of any technical restriction

c) **Crossing network boundaries**

- Another issue which movitate Tim Berners Lee to create was being able to connect to system beyond network boundaries.
- It was time consuming and required user to remember a lot of commands for transmission.
- To be able to reach people far away from the host.

d) **Combining Hypertext with network**

- To be able to combine the capability of hypertext with transmission of data over the network.

3.2 Five design principles of the world wide web

a) **Test of Independent invention**

- In a centralized system if there are two similar ideas, one would be discared thereby lossing the efforts and idea

- World wide web gave the opportunity to those similar ideas to coexist by integrating them smoothly and imporve those idea with incremental efforts

b) **HyperText**

- The Hypertext was combined with networking capability to create world wide web

c) **Networking Capability**

- The ability to reach beyond the network boundaries.

d) **URL**

- Global naming convention for every website which is easy to remember
- The URL need not be registered, only the domain names are required to register

e) **No single point of failure**

- Any subdomain may no longer be available/down, however web as a whole will not stop working.

f) **Scaling**

- World wide web has the capabiliy to scale as and when the number of host/users starts increasing.

g) **Simple Language**

- It was developed using html which is very simple and minimum syntax to remember, which helps people contribute easily.

h) **Effortless**

- Accessing and contributing was made effortless with around 80% interaction and only 20% efforts

i) **Easy to provide content using http**

j) **Can work independently and contribute anonymously**

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 [1]. 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

a) https://www.facebook.com/photo.php?fbid=2068026323275211&set=a.269104153167446&type=3&theater

Figure 3: Screenshot of the url prsing

b) http://www.blog.google.uk:1000/path/to/myfile.html?key1=value1&key2=value2#InTheDocument

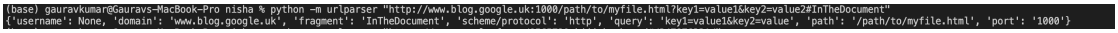

Figure 4: Screenshot of the url prsing

c) https://www.overleaf.com/9565720ckjijuhzpbccsd#/347876331/

Figure 5: Screenshot of the url prsing

d) ftp://root@west.uni.koblenz.de

Figure 6: Screenshot of the url prsing

e) https://west.uni-koblenz.de/studying/ws2021

Figure 7: Screenshot of the url prsing

[1] https://en.wikipedia.org/wiki/Top-level_domain

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

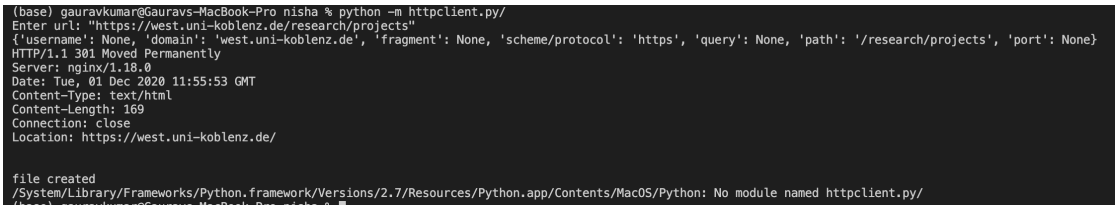

Figure 8: 301 status code

- **https://west.uni-koblenz.de/research/projects**, returned code **301**, this domain is moved to a new URL and is no longer available for use at the given URL.

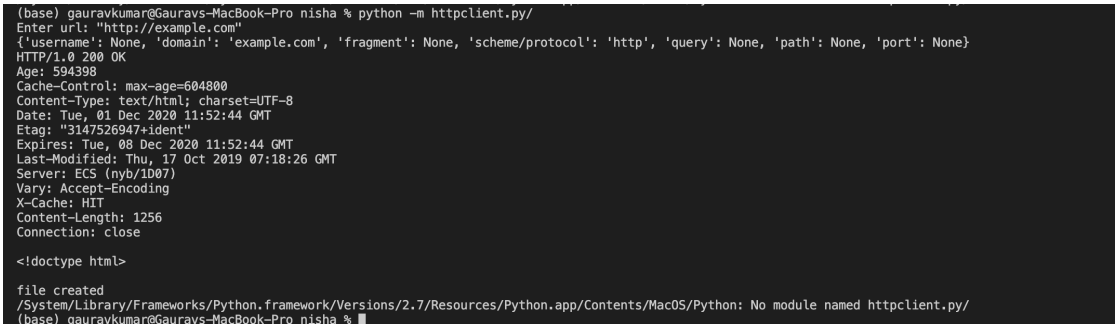

Figure 9: 200 status code

- For url **http://example.com**, returned code **200**, This signifies successful connection to the given website.

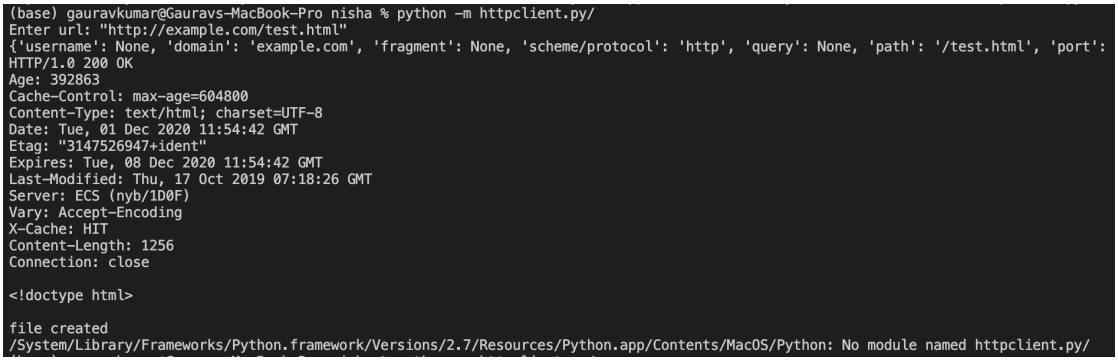

Figure 10: 200 status code

- For url **http://example.com/test.html**, returned code **200**, This signifies successful connection to the given website.