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.

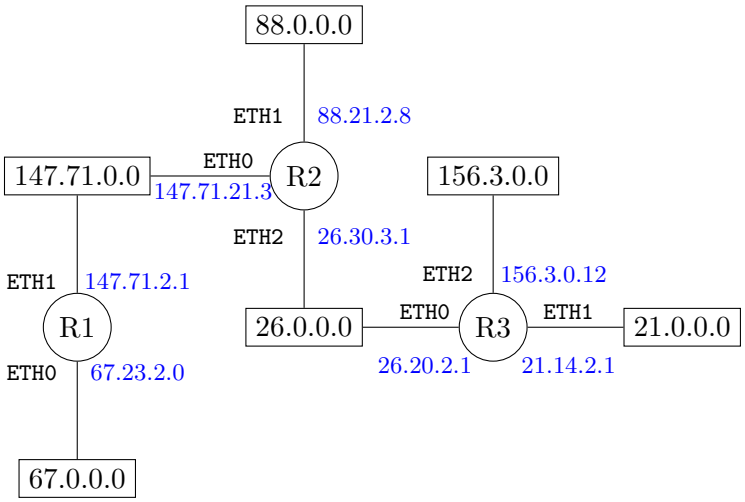

Figure 1: Routing schematic representation

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.

2.2 Example to show usage of the four layers:

-

- When transfering data like a file from one computer to another computer, we have a protocol on the application layer which enables a process to process communication channel.
- The file is split into segments encapsulated into a TCP segment on the transport layer. The transport layer protocol is used to establish a host to host connection.
- Each TCP segment is packed inside an IP packet which can then be routed between various routers across networks. The routing happens on the internet layer which interconnects different networks on the path of the packet.
- In order to send data from one router to another in a network a linked layer protocol is used. IP packets are then encapsulated inside frames. Ethernet or DSL are typical linked layer protocols.
- As the data arrives the encapsulation process is reversed in order to assemble the original file.

REFERENCE: https://commons.wikimedia.org/wiki/Data-Flow-of-the-Internet-Protocol-Suite

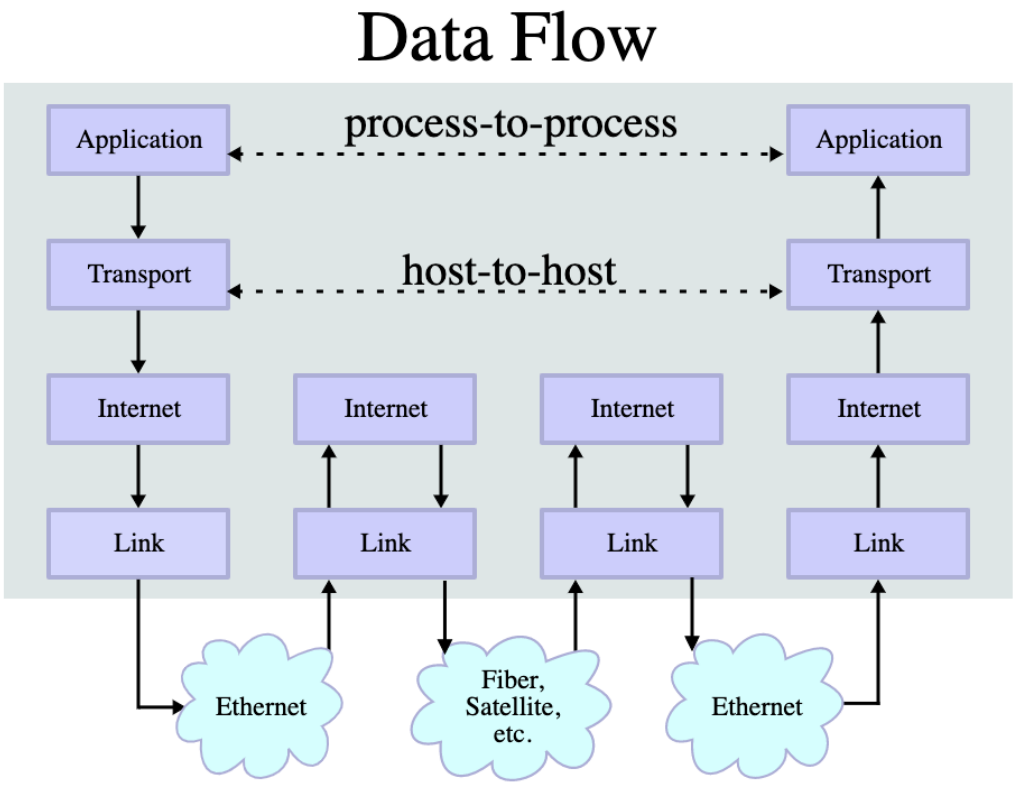

Figure 2: Data Flow of the Internet Protocol Suite

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

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

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

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

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

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

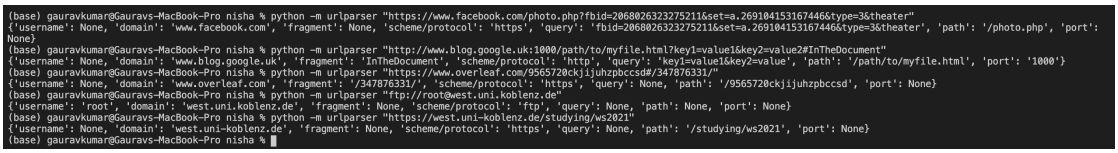

Figure 3: Screenshot of the url parsing

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.

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

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?

a) http://example.com

b) http://example.com/test.html

c) https://west.uni-koblenz.de/research/projects

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

```
(base) gauravkumar@Gauravs-MacBook-Pro nisha % python -m httpclient.py/
Enter url: "https://west.uni-koblenz.de/research/projects"
{'username': None, 'domain': 'west.uni-koblenz.de', 'fragment': None, 'scheme/protocol': 'https', 'query': None, 'path': '/research/projects', 'port': None}
HTTP/1.1 403 Forbidden
Server: nginx/1.18.0
Date: Mon, 30 Nov 2020 15:15:00 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive


file created
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: No module named httpclient.py/
```

Figure 4: 403 status code

```
(base) gauravkumar@Gauravs-MacBook-Pro nisha % python -m httpclient.py/
Enter url: "https://www.sqlite.org"
{'username': None, 'domain': 'www.sqlite.org', 'fragment': None, 'scheme/protocol': 'https', 'query': None, 'path': None, 'port': None}
HTTP/1.1 301 Permanent Redirect
Connection: keep-alive
Date: Mon, 30 Nov 2020 15:13:00 +0000
Location: http://stackoverflow.com/index.html
Content-length: 0


file created
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: No module named httpclient.py/
(base) gauravkumar@Gauravs-MacBook-Pro nisha % python -m httpclient.py/
Enter url: "https://www.sqlite.org/lang_select.html"
{'username': None, 'domain': 'www.sqlite.org', 'fragment': None, 'scheme/protocol': 'https', 'query': None, 'path': '/lang_select.html', 'port': None}
HTTP/1.1 301 Permanent Redirect
Connection: keep-alive
Date: Mon, 30 Nov 2020 15:13:30 +0000
Location: http://stackoverflow.com/index.html
Content-length: 0


file created
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: No module named httpclient.py/
```

Figure 5: 301 status code

```
(base) gauravkumar@Gauravs-MacBook-Pro nisha % python -m httpclient.py/
Enter url: "http://data.pr4e.org/"
{'username': None, 'domain': 'data.pr4e.org', 'fragment': None, 'scheme/protocol': 'http', 'query': None, 'path': '/', 'port': None}
HTTP/1.1 400 Bad Request
Date: Mon, 30 Nov 2020 15:13:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 308
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

file created
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: No module named httpclient.py/
(base) gauravkumar@Gauravs-MacBook-Pro nisha % python -m httpclient.py/
Enter url: "http://data.pr4e.org/romeo.txt"
{'username': None, 'domain': 'data.pr4e.org', 'fragment': None, 'scheme/protocol': 'http', 'query': None, 'path': '/romeo.txt', 'port': None}
HTTP/1.1 400 Bad Request
Date: Mon, 30 Nov 2020 15:14:10 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 308
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

file created
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: No module named httpclient.py/
```

Figure 6: 400 status code