Introduction to Web Science

Assignment 1

PD Dr. Matthias Thimm   Ipek Baris Schlicht

thimm@uni-koblenz.de   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: 17.11.2020, CEST 23:59

For the programming tasks, please do not add your code on the PDF. You need to submit only the .ipynb or .py files.

In case a team member has not contributed to the assignment, please do not include the name in the PDF.

1 Introduction to Python Programming   20 points

1.1   10 points

In this task, you will write a simple python script that does the following:

1. Generate a random number sequence of **100** values that are **between 0 to 1000**. Make sure that each of element in the sequence is type of `float` and use **42** as random seed.

2. Print each of the element in the sequence.

3. The elements in the sequence denote the degrees. Perform sine and cosine operation on them and store the values in two different arrays named SIN and COS respectively.

4. Plot the values of SIN and COS in two different colors and shapes. The plot must have labeled axes and legend that contain plausible information of the task.

Only `numpy`, `random` and `matplotlib` are allowed for this task.

1.2   10 points

Write another simple python script that does the following:

1. Read sample text (`sample.txt`) and store in `TEXT` variable.

2. Count the frequency of each word in `TEXT` by filtering out any punctuation (e.g `.`, `!`) and number. If a word has uppercase letters, change them to lowercase.

3. Plot the frequency distribution of words that occurs more than once, in an descending order. The plot must have labeled axes and legend that contain plausible information of the task. Apply the necessary settings for readable axes' information.

Only `string` and `matplotlib` are allowed for this task.

For the programming tasks, you can use Google Colab. However, if you use your computer, make sure that the version of Python is 3.6 or 3.7.

2 Ethernet Frame   20 points

An Ethernet Frame is of the given structure:

Preamble	Destination MAC address	Source MAC address	Type/Length	User Data	Frame Check Sequence
8	6	6	2	46-1500	4

Table 1: Ethernet Frame Structure with associated sizes in Bytes

Given below are two Ethernet frames.

```
aa aa aa aa aa aa aa ff        10 52 99 a5 42 d7 02 55
74 31 59 a8 86 dd aa 31        89 45 63 81 23 05 03 88
e2 41 31 83 b2 83 41 09        00 00 00 00 00 31 c0 a8
02 67 00 00 18 ca 70 46


aa aa aa aa aa aa aa ff        41 21 65 66 aa 01 41 92
12 43 00 de 08 06 00 31        00 09 03 13 53 71 58 12
97 53 13 12 54 13 90 31        00 00 00 00 00 31 c0 a8
02 67 00 00 63 c5 63 3c
```

Find for both Ethernet frames:

1. Destination MAC Address

2. Source MAC Address

3. What protocol is inside the data payload?

3 Research tasks   20 points

In this task you should do additional research extending the lecture. Please keep the citation rules in mind.

3.1 Collision

A computer tries to send data over an Ethernet network. However, after sending three packages the computer detects a collision. What happens next?

Describe in your own words how the Ethernet Collision Detection Algorithm handles a detected collision and highlight how the algorithm stops.

3.2 IPv6

In a few sentences please research the differences between IPv4 and IPv6 and explain the advantages of IPv6.

4 Routing Table 20 points

4.1 Solution:

Based on the schematic representation from Figure 1 the routing table is as shown in table 2
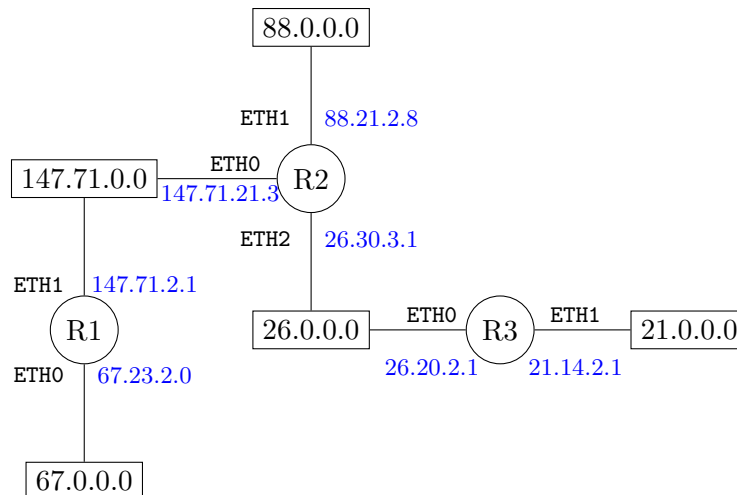

Figure 1: Routing schematic representation

Table 2: Routing Table for 1

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	26.30.3.1	eth2	88.0.0.0	26.30.3.1	eth0
26.0.0.0	147.71.21.3	eth1	67.0.0.0	141.71.2.1	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

4.2 Solution:

The Routing schematic representation for table 3 is shown in figure 2.
The steps in path traversal if a packet which is generated from 67.0.0.0 network and heading for 26.0.0.0 network are:

1. When a packet is generated at **67.0.0.0** for destination **26.0.0.0**, the host will look up the routing table for the next hop which is at IP **67.68.3.1** through interface **eth0** in order to reach router 1 (represented by R1).

2. The router looks up the destination IP of the packet and confirm it's actual destination and lookup in the routing table for the next hop to reach that network which is **141.71.20.1** though interface **eth2** and reaches **147.71.0.0**.

3. Upon receiving this packet the network will repeat the process of looking up in the routing table and redirect to router 2 (represented by R2) by taking the next hop at **141.71.26.3** through interface **eth1**.

4. Since router 2 is directly connected to network **26.0.0.0** the last hop is taken at **26.3.2.1** through interface **eth2**.

The Routing schematic representation to send a packet which is generated from 67.0.0.0 network and heading for 26.0.0.0 network is shown in figure 3.

Table 3: Routing Table

| Router 1			Router 2			Router 3		
Destination	Next Hop	Interface	Destination	Next Hop	Interface	Destination	Next Hop	Interface
67.0.0.0	67.68.3.1	eth0	205.30.7.0	205.30.7.1	eth0	205.30.7.0	205.30.7.2	eth0
88.0.0.0	88.4.32.6	eth1	141.71.0.0	141.71.26.3	eth1	88.0.0.0	88.6.32.1	eth1
141.71.0.0	141.71.20.1	eth2	26.0.0.0	26.3.2.1	eth2	26.0.0.0	205.30.7.1	eth0
26.0.0.0	141.71.26.3	eth2	67.0.0.0	141.71.20.1	eth1	141.71.0.0	205.30.7.1	eth0
205.30.7.0	88.6.32.1	eth1	88.0.0.0	141.71.20.1	eth1	67.0.0.0	88.4.32.6	eth1
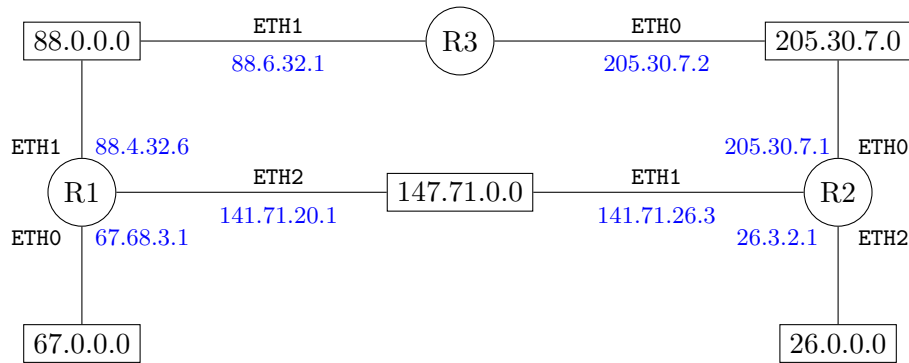
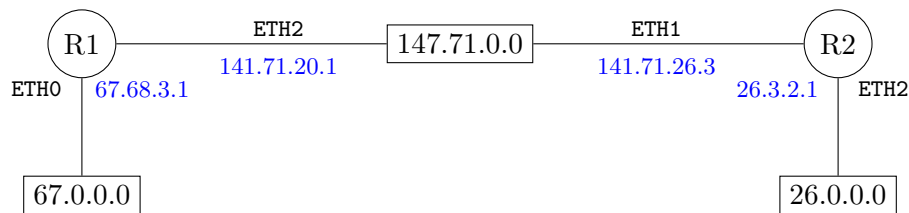**Figure 2:** Routing schematic representation

Figure 3: Routing schematic representation of path if a packet is generated from 67.0.0.0 network and heading for 26.0.0.0 network.

6

Important Notes

Submission

- Solutions have to be submitted to the SVN repository. Use the directory name `groupname/assignmentassignmentnumber/` 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 SVN 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 pdfLaTeX template was adapted by Jun Sun based on the LuaLaTeX version by Lukas Schmelzeisen.

LaTeX

Use `pdflatex or LuaLatex combiler for assignment_X.tex` to build your PDF.