Introduction to Web Science

Assignment 1

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: 17.11.2020, CEST 23:59

Team: Bravo
Members:
Gaurav Kumar (220200656)
Pavithree Shetty (220200661)
Nisha Sharma (220202359)
Oishi Chatterjee (220203213)

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.

1.1.1 Solution

Attached python files ($Assignment$_1.1)

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.

1.2.1 Solution

Attached python files ($Assignment$_1.2 and samlpe.txt)

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 payloade?

2.1 Ethernet Frame : Solution

For both Ethernet frames, destination MAC address, source MAC address and protocol details are as follows:

Table 2: Destination MAC address, Source MAC address and Protocol

	Ethernet 1	Ethernet 2
1. Destination MAC Address	10 52 99 a5 42 d7	41 21 65 66 aa 01
2. Source MAC Address	02 55 74 31 59 a8	41 92 12 43 00 de
3. Protocol	Internet Protocol Version 6 (IPv6)	Address Resolution Protocol (ARP)

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 Solution:

Carrier-sense multiple access with collision detection (CSMA/CD) is the collision detection algorithm implemented in Ethernet. Procedure:

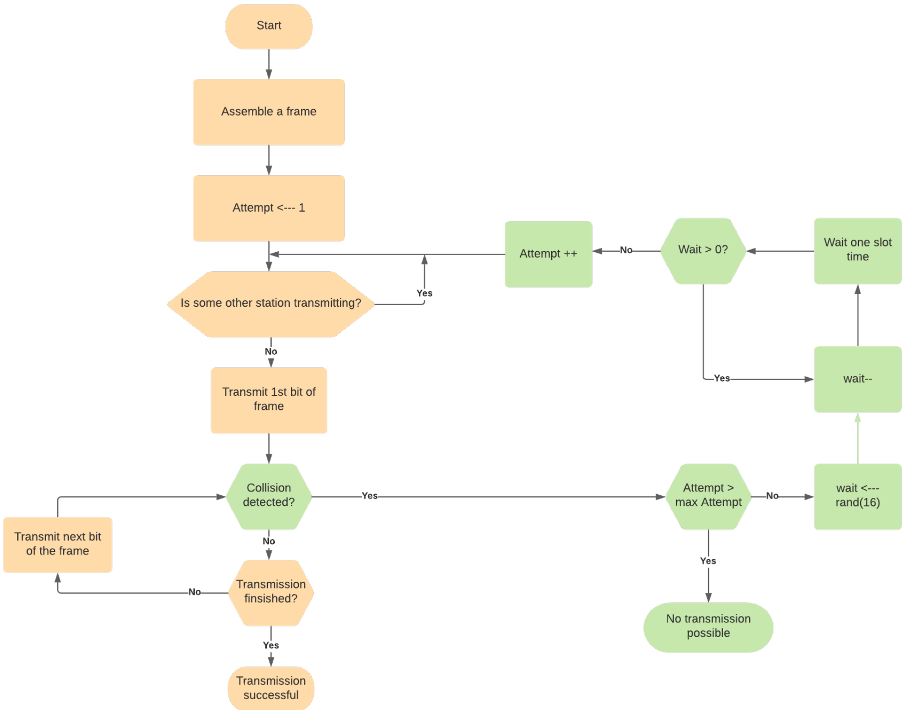

Figure 1: CSMA/CD

- The host assembles the frame to be transmitted with necessary header fields.
- Once the frame is available for transmission, the medium is checked if its idle, if yes the first package is transmitted.
- In the absence of any collision the host continues to transmit the frame to its completion.
- Let us assume a collision has been detected after the transmission of first 3 packages. Now the collision detection procedure is initialised.

- The moment collision is detected, a jam signal is transmitted instead of the frame to notify all the devices the occurance of the collision.
- The jam signal is transmitted for a minimum packet time so that it transmits through entire Ethernet length notifying all the devices about the collision.
- The algorithm checks if the maximum attempt is reached if yes, the algorithm stops here telling the sender further transmission of data is not possible and the algorithm halts.
- If the maximum attempts is not reached,the wait counter is assigned with a random number, and it waits for one slot time (512 clock cycles - minimum frame size is 46+12 bytes) this is the minimum time required to occupy entire ethernet frame.
- All the devices will be notified about the collision at the same time so while we refrain from transmitting the other devices will also do the same. So the idea here is that random numbers of these devices will be different.
- As we wait for one slot time we decrement the wait counter, and increment the attempt counter and check if the medium is idle for transmission, if yes continue transmission. If not we wait for another slot time till wait counter equals 0 or maximum attempt is reached.

3.2 IPv6 : Solution

Differences between IPv4 and IPv6 are as follows:

Table 3: IPv4 vs IPv6

IPv4	IPv6
32-bit IP address with numeric addressing method	128 bit IP address with alphanumneric addressing method
Address resolution Protocol is used to map MAC address	Neighbour Dicovery protocol is used to map MAC address
supports Broadcast Data transmission	Supports multicast Data trasnmission .
DHCP or manual congiguration	Supports Auto-configuration
Both host and the routers perform fragmentation	Only Sender performs fragmentation
checksum field is present	checksum field is absent
It generates 4 billion address	It generates 340 undecillion addresses

Advantages of IPv6:

- Autoconfiguration:
  As soon as the device is powered up, IPv6 generates an IP address and places itself in the network. The moment it finds an IPv6 router it generates a local address and a globally routable address. In contrast, in IPv4 the devices has to be added manually where autoconfiguartion is not supported.
- Efficient packet processing:
  In IPv4 the checksum is recalculated at every hop thus increasing packet processing time. IPv6 contains no checksum field so recalculation of cheksum is avoided .

- Directed Data Flows:
  IPv6 supports multicast rather than broadcast. Multicast allows bandwidth-intensive packet flows to be sent to multiple destinations simultaneously, saving network bandwidth.
- Efficient routing:
  In IPv6 the packet fragmentation is handled by the sender and the size of routing tables are comparatively smaller than IPv4 routing tables.
- Multicasting :
  As IPv6 supports multicasting, therefore overall network bandwidth is reduced by transmitting bandwidth intensive packets to multiple destinations simultaneously.
- Security :
  IPSec security is implemented which provides authentication and protects data integrity.

4 Routing Table 20 points

4.1 Solution:

Based on the schematic representation from Figure 2 the routing table is as shown in table 4

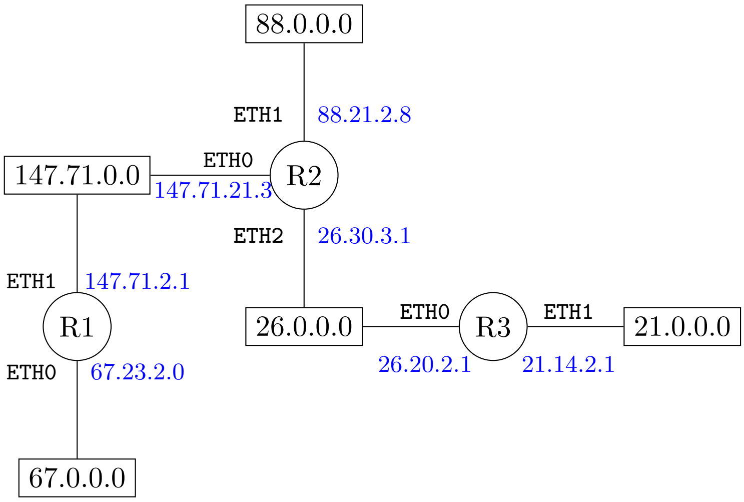

Figure 2: Routing schematic representation

Table 4: Routing Table for 2

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 5 is shown in figure 3.
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 4.

Table 5: 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

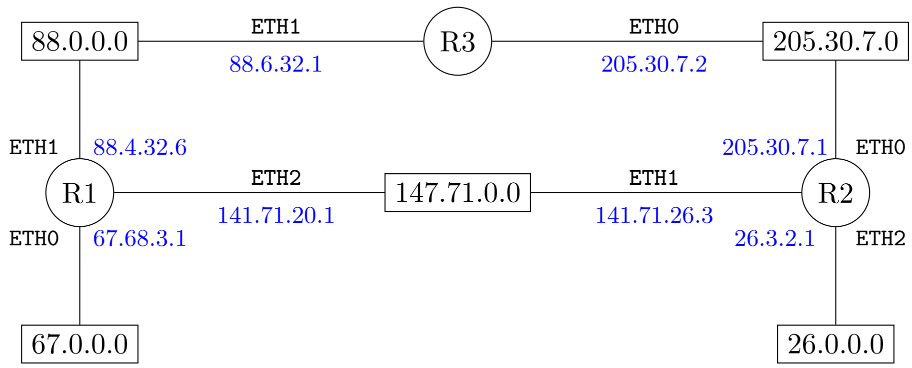

Figure 3: Routing schematic representation

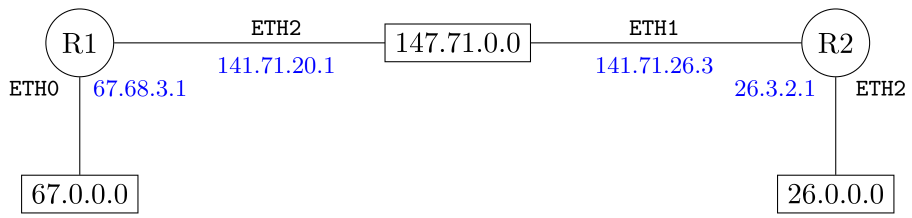

Figure 4: Routing schematic representation of path if a packet is generated from 67.0.0.0 network and heading for 26.0.0.0 network.