Introduction to Web Science

Assignment 1

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: November 2, 2016, 10:00 a.m.
Tutorial on: November 4th, 2016, 12:00 p.m.

The main objective of this assignment is for you to use different tools with which you can understand the network that you are connected to or you are connecting to in a better sense. These tasks are not always specific to "Introduction to Web Science". For all the assignment questions that require you to write a code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Group name: mike
Group members: Anish Girijashivaraj, Shohel Ahamad, Slobodan Kocevski

1 Ethernet Frame (5 Points)

Ethernet Frame is of the given structure:

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

Figure 1: Ethernet Frame Structure

Given below is an Ethernet frame without the Preamble and the Frame Check Sequence.

```
00 27 10 21 fa 48 00 13     10 e8 dd 52 08 06 00 01
08 00 06 04 00 01 00 13     10 e8 dd 52 c0 a8 02 01
00 00 00 00 00 00 c0 a8     02 67
```

Find:

1. Source MAC Address
2. Destination MAC Address
3. What protocol is inside the data payload?
4. Please mention what the last 2 fields hold in the above frame.

Answer:

1. 00 13 10 e8 dd 52
2. 00 27 10 21 fa 48
3. 0x0806 Address Resolution Protocol
4. The last two fields in the Address Resolution protocol are reserved for the protocol address of the intended receiver.

2 Cable Issue (5 Points)

Let us consider we have two cables of 20 meters each. One of them is in a 100MBps network while the other is in a 10MBps network. If you had to transfer data through each of them, how much time it would take for the first bit to arrive in each setting? (For your calculation you can assume that the speed of light takes the same value as in the videos.) Please provide formulas and calculatoins along with your results.

Answer:

1. Cable length 20 meters
   100MBps = 100 Million bits per second
   For 1 bit the time will be
   (1 second/100 Million bits per second) = 0.00000001 seconds.
   There are 1000000000 nanoseconds in one second, so we get 10 nanoseconds for one bit of transaction.

   The speed of light is 299792458 meters per second = electromagnetic waves.
   So, in 100MBit cable, 1 bit will travel
   (299 792 458 meters / 1 second)*(10 nanoseconds) = 2.99792458 meters in one clock cycle .
   So, to travel 20 meters it will take
   (20*10)/2.99792458 = 66.71 nanoseconds

2. Cable length 20 meters
   10MBps = 10 Million bits per seconds
   For 1 bit the time will be
   (1second/10 Million bits per second) = 0.0000001 seconds
   There are 1000000000 nanoseconds in one second, so we get 100 nanoseconds for one bit of transaction.

   The speed of light is 299792458 Million meters per second = electromagnetic waves. So, in 10 MBps, 1 bit will travel
   (299 792 458 meters / 1 second)*(100 Nano Second) =29.9792458 meters in one clock cycle.
   So, to travel 20 meters it will take
   (20*100)/29.9792458 = 66.71 Nano Second.

3 Basic Network Tools (10 Points)

Listed below are some of the commands which you need to ”google” to understand what they stand for:

1. *ipconfig / ifconfig*
2. *ping*
3. *traceroute*
4. *arp*
5. *dig*

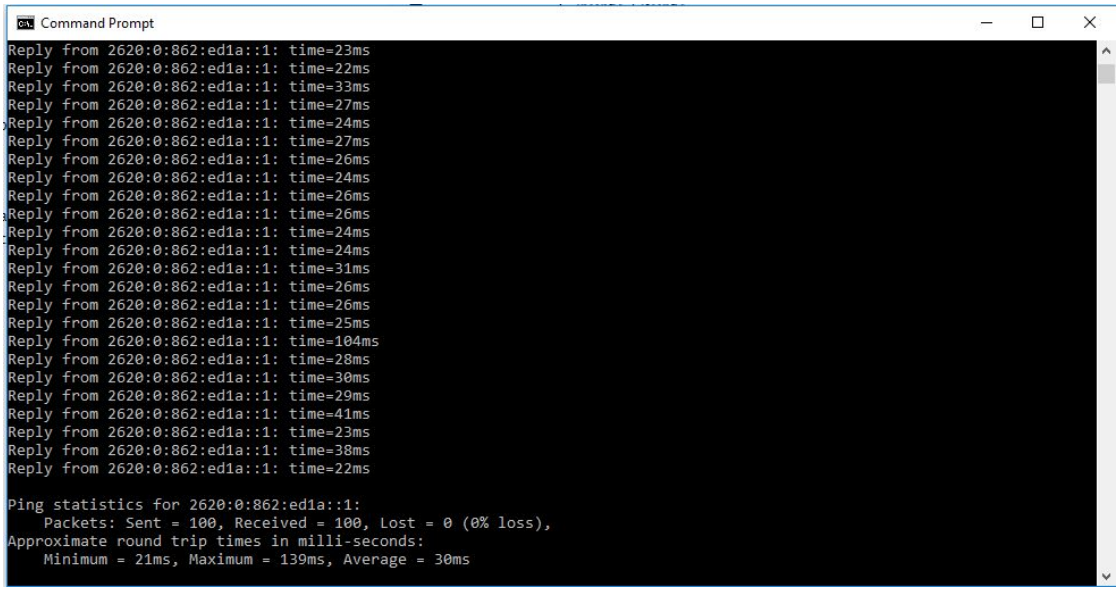
Consider a situation in which you need to check if `www.wikipedia.org` is reachable or not. Using the knowledge you gained above to find the following information:

1. The *% packet loss* if at all it happened after sending 100 packets.
2. *Size* of the packet sent to *Wikipedia* server
3. *IP address* of your machine and the *Wikipedia* server
4. *Query Time* for DNS query of the above url.
5. Number of *Hops* in between your machine and the server
6. MAC address of the device that is acting as your network gateway.

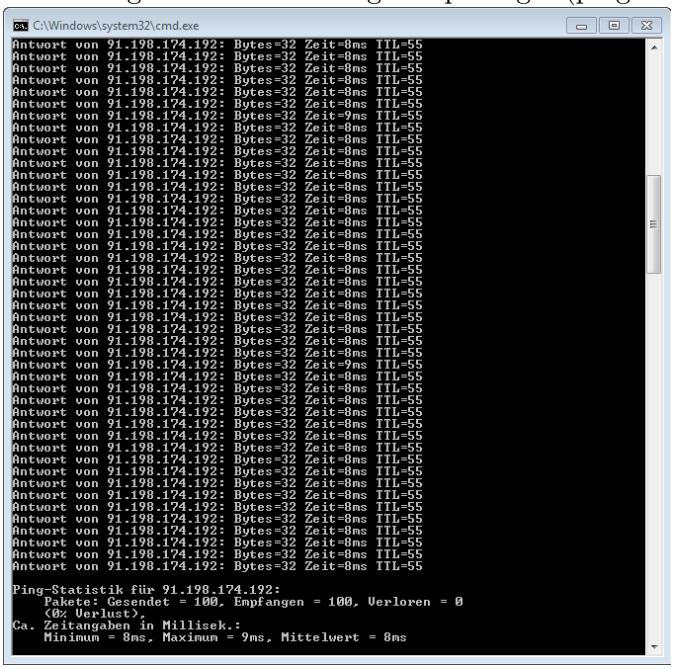
Do this once in the university and once in your home/dormitory network. With your answers, you must paste the screen shots to validate your find.

Answer:

1. The *% packet loss* if at all it happened after sending 100 packets.
   a) Home network:
   0% Package loss after sending 100 packages (ping -n 100 www.wikipedia.org)

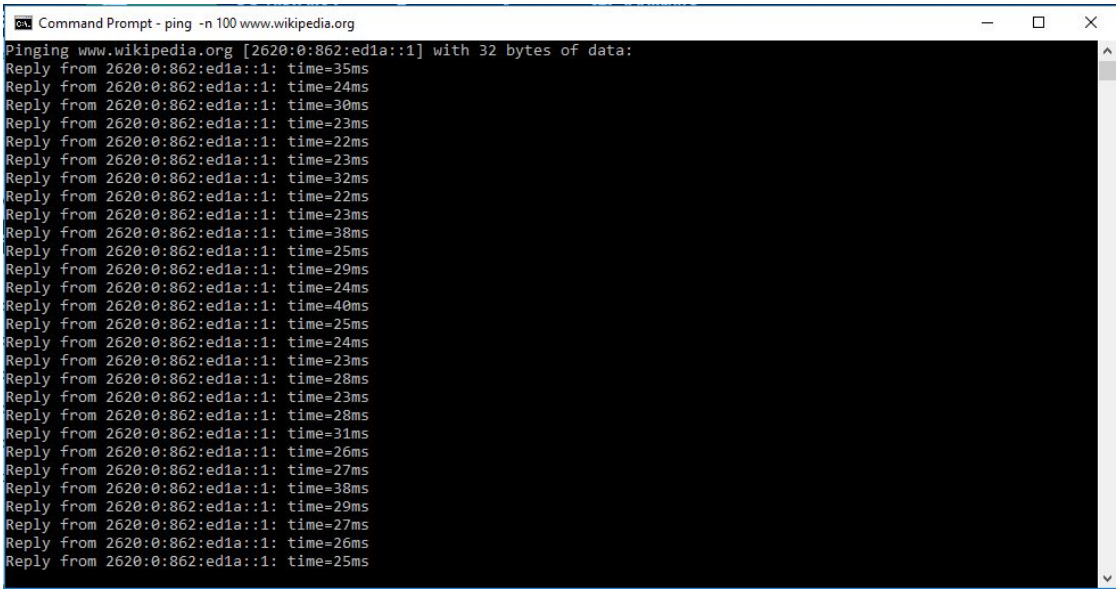


b) University's network:
0% Package loss after sending 100 packages (ping -n 100 www.wikipedia.org)

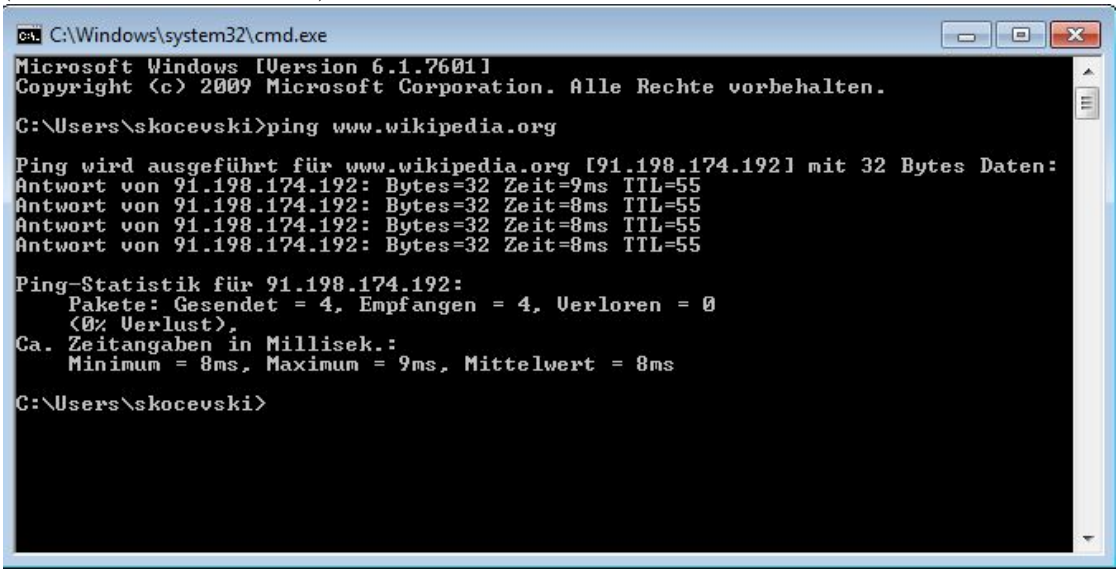


2. Size of the packet sent to Wikipedia server

a) Home network:
Size of the package sent to www.wikipedia.org is 32bytes (ping www.wikipedia.org)

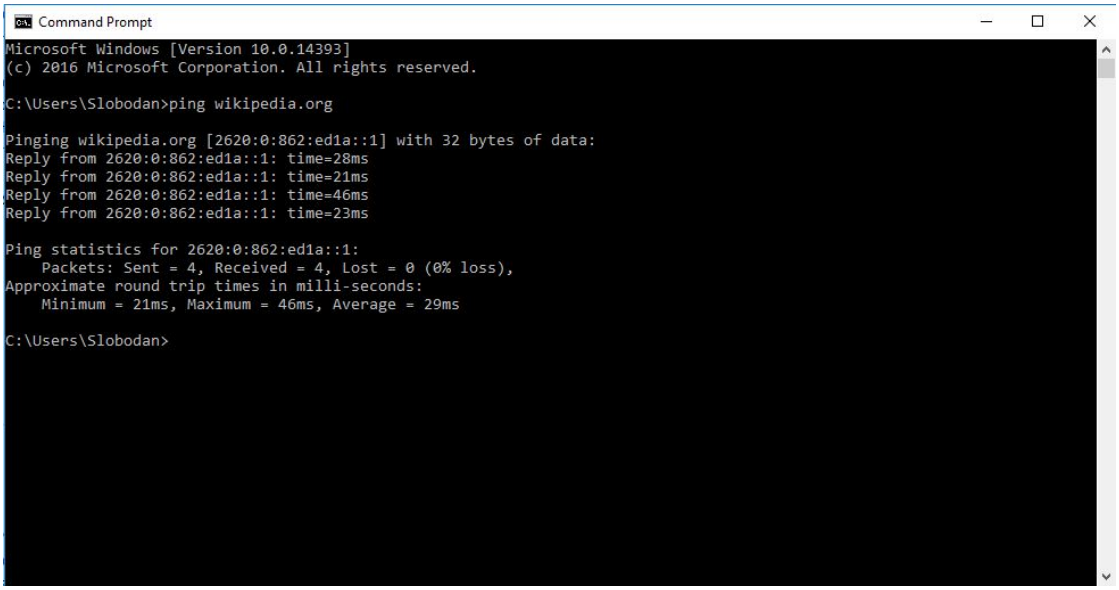


b) University's network: Size of the package sent to www.wikipedia.org is 32bytes (ping www.wikipedia.org)

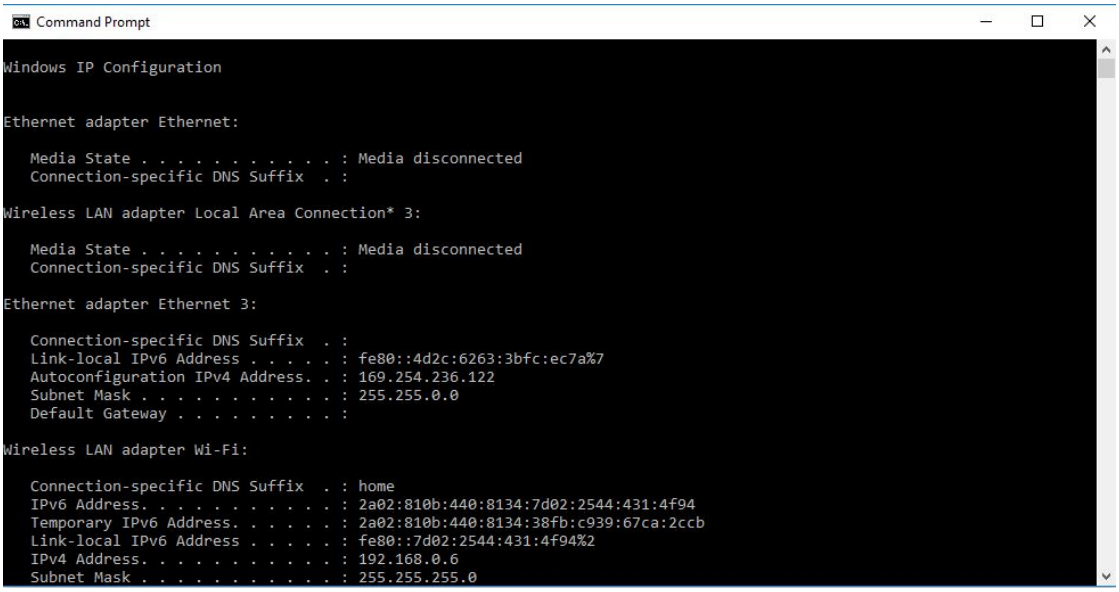


3. IP address of your machine and the Wikipedia server

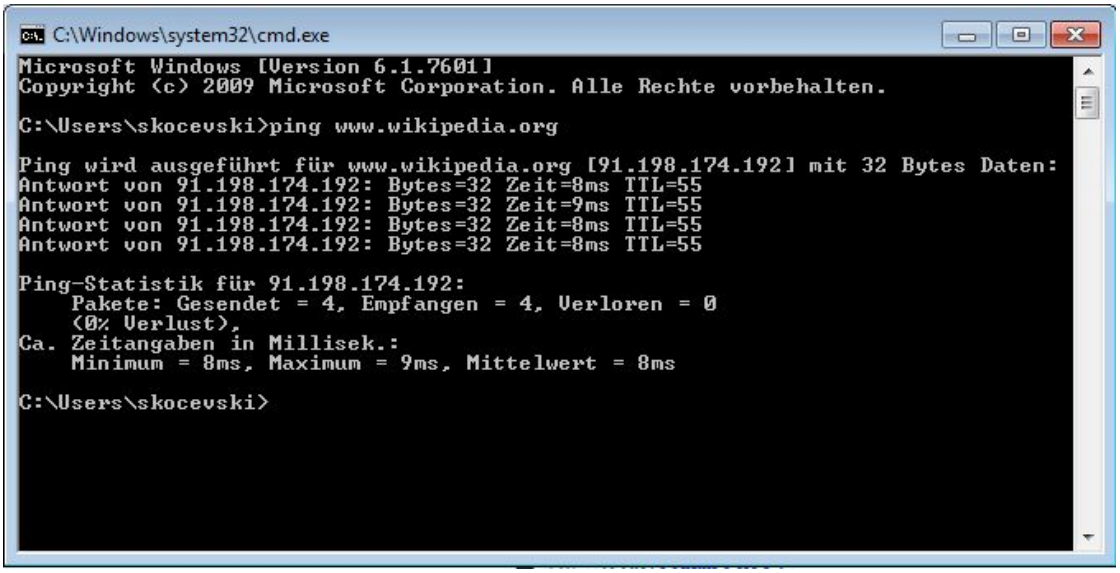
a) Home network: (ping www.wikipedia.org)
The IP address of wikipedia.org server is [2620:0:862:ed1a::1]



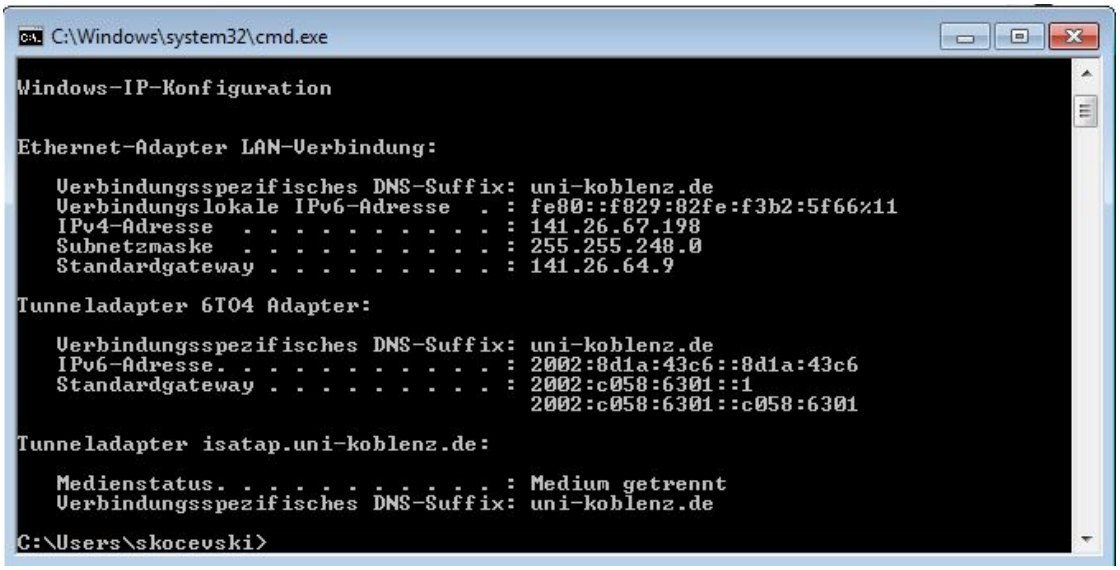
The IP address of local machine is 192.168.0.6



b) University's network: (ping www.wikipedia.org)
The IP address of wikipedia.org server is 91.198.174.192

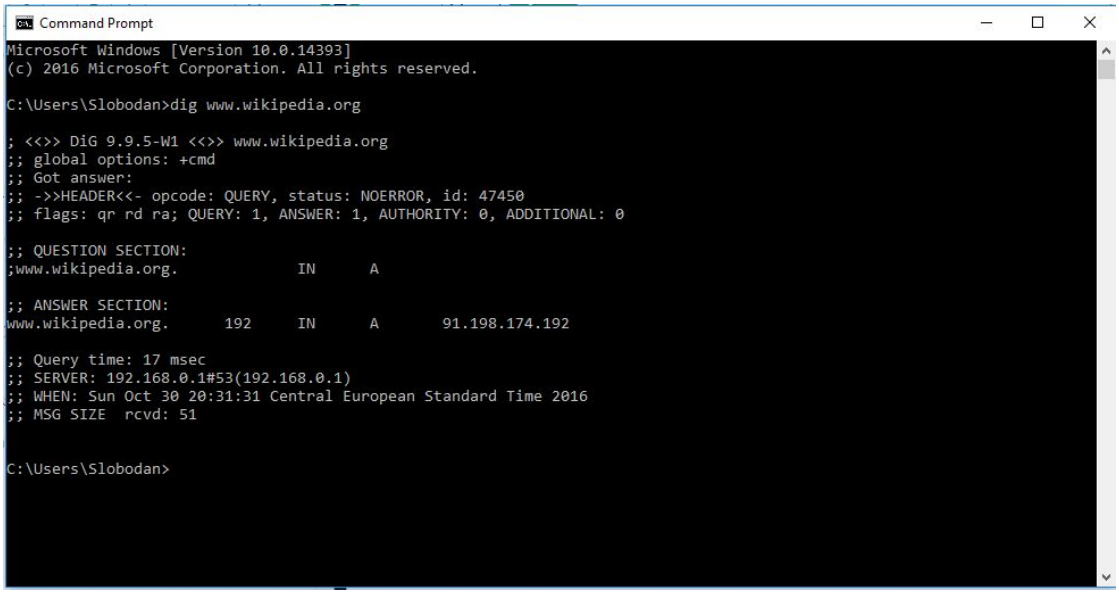


The IP address of local machine is 141.26.67.198

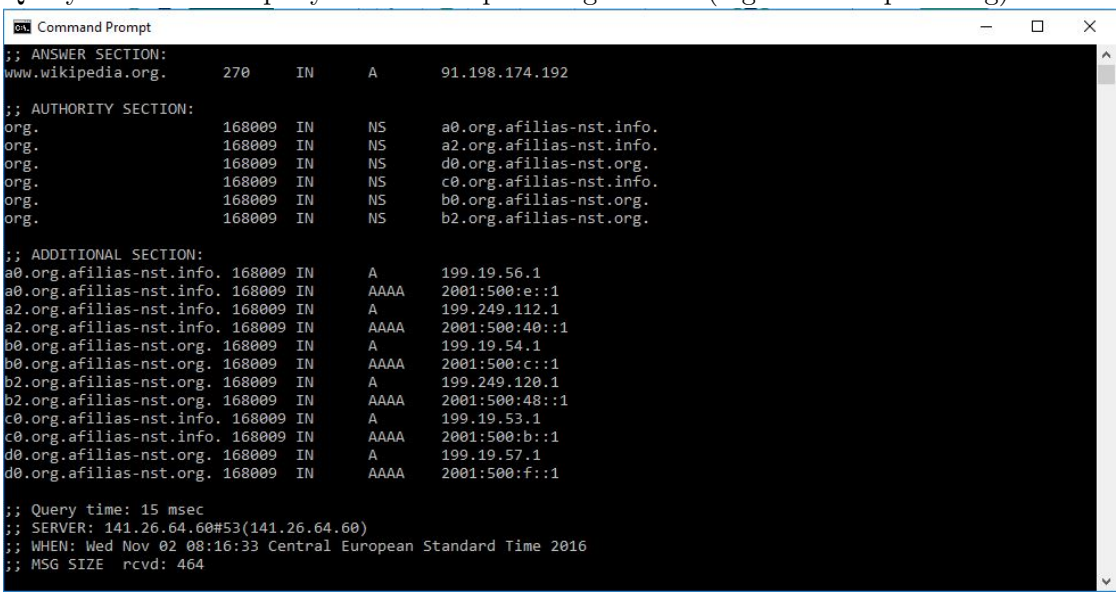


4. Query Time for DNS query of the above url.

   a) Home network:
   Query time for DNS query of www.wikipedia.org: 17msec (dig www.wikipedia.org)

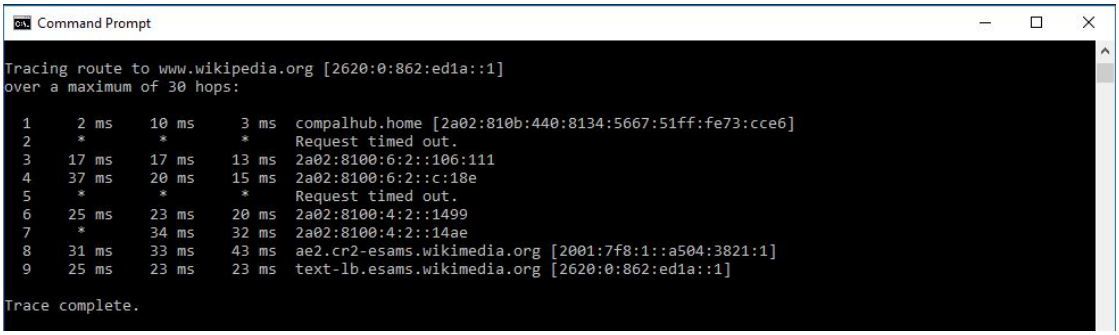


b) University's network
Query time for DNS query of www.wikipedia.org: 15msec (dig www.wikipedia.org)

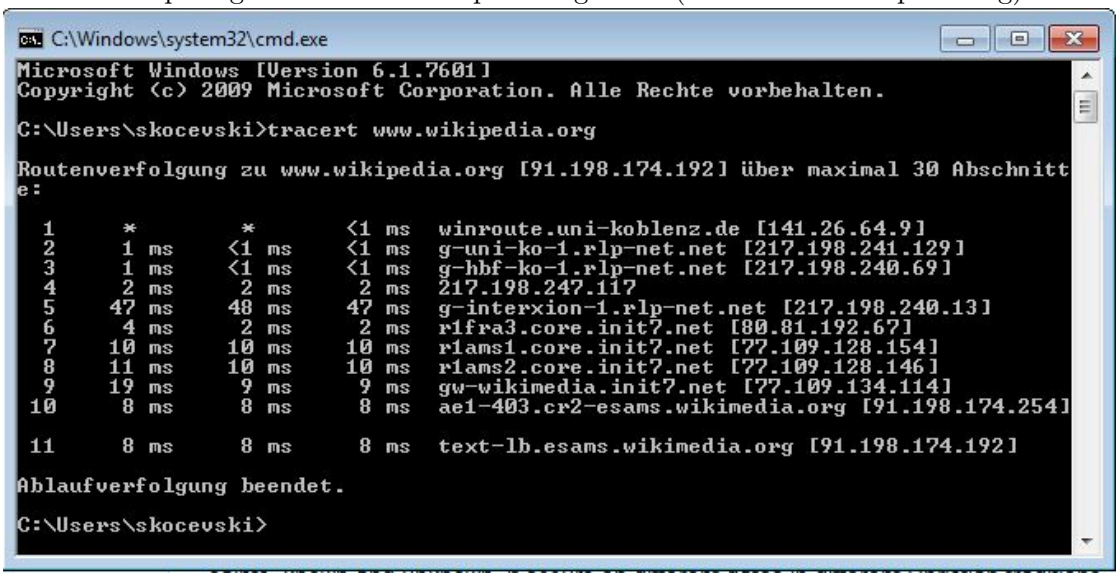


5. Number of *Hops* in between your machine and the server.

a) Home network:
It took 9 Hops to get to the www.wikipedia.org server (tracert www.wikipedia.org)

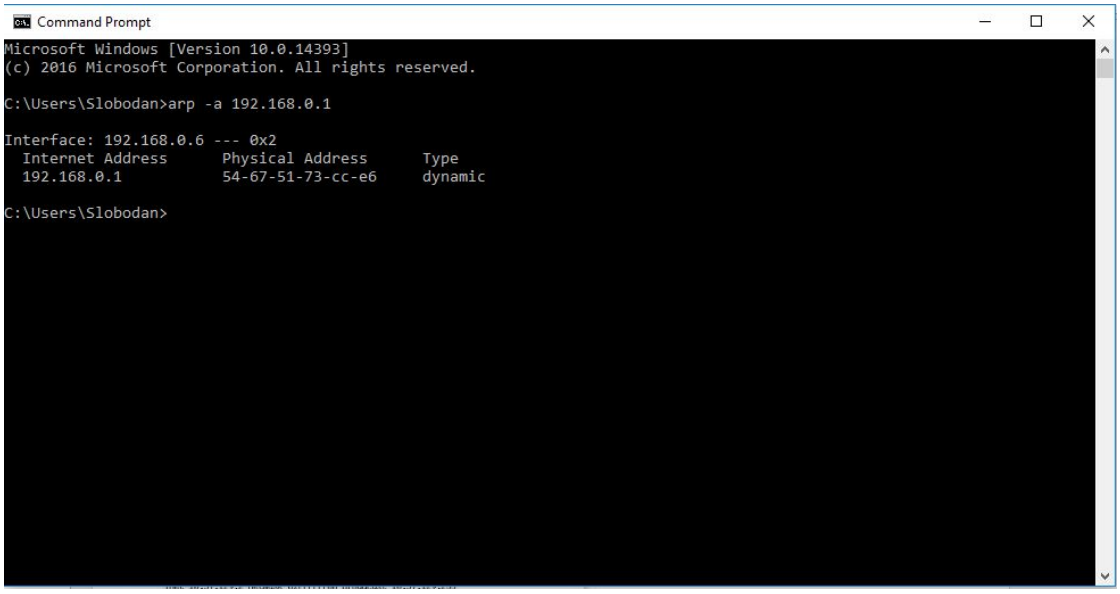


b) University's network:
It took 11 Hops to get to the www.wikipedia.org server (tracert www.wikipedia.org)



6. MAC address of the device that is acting as your network gateway

a) Home network:
MAC address of the network gateway device:54-67-51-73-cc-e6



b) University's network:
MAC address of the network gateway device D4-3D-7E-F1-97-18

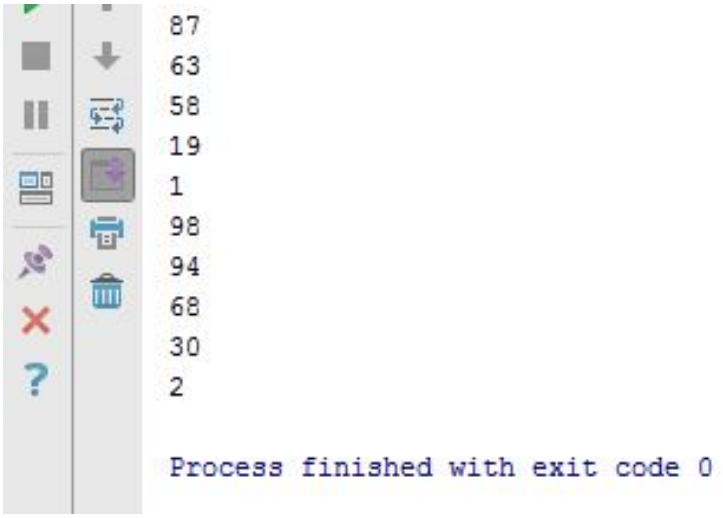
4 Simple Python Programming (10 Points)

Write a simple python program that does the following:

1. Generate a random number sequence of 10 values between 0 to 90.
2. Perform `sine` and `cosine` operation on numbers generated.
3. Store the values in two different arrays named SIN & COSIN respectively.
4. Plot the values of SIN & COSIN in two different colors.
5. The plot should have labeled axes and legend.

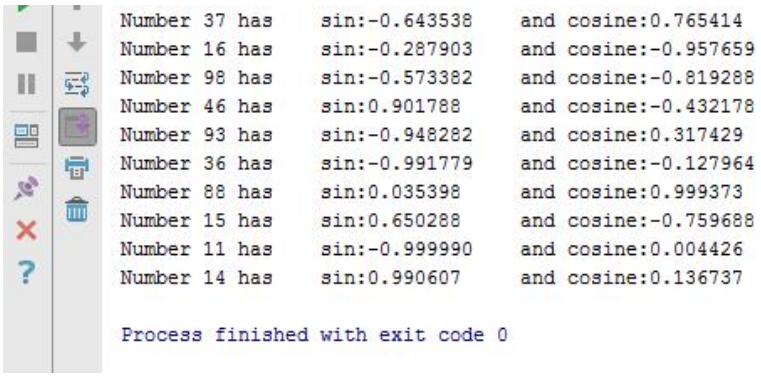
Answers:

1. import random
   x = 0
   while (x < 10):
   print(random.randint(0, 99))
   x += 1

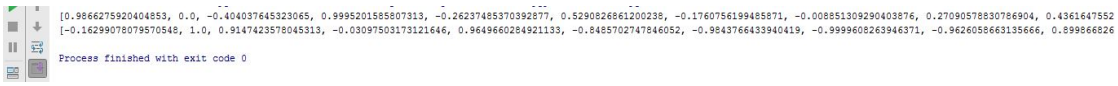


2. import math
   import random
   x = 0
   while (x < 10):
   y = random.randint(0, 99)
   s = math.sin(y)
   c = math.cos(y)

print("Number %d has sin:%f and cosine:%f" % (y, s, c))
x += 1

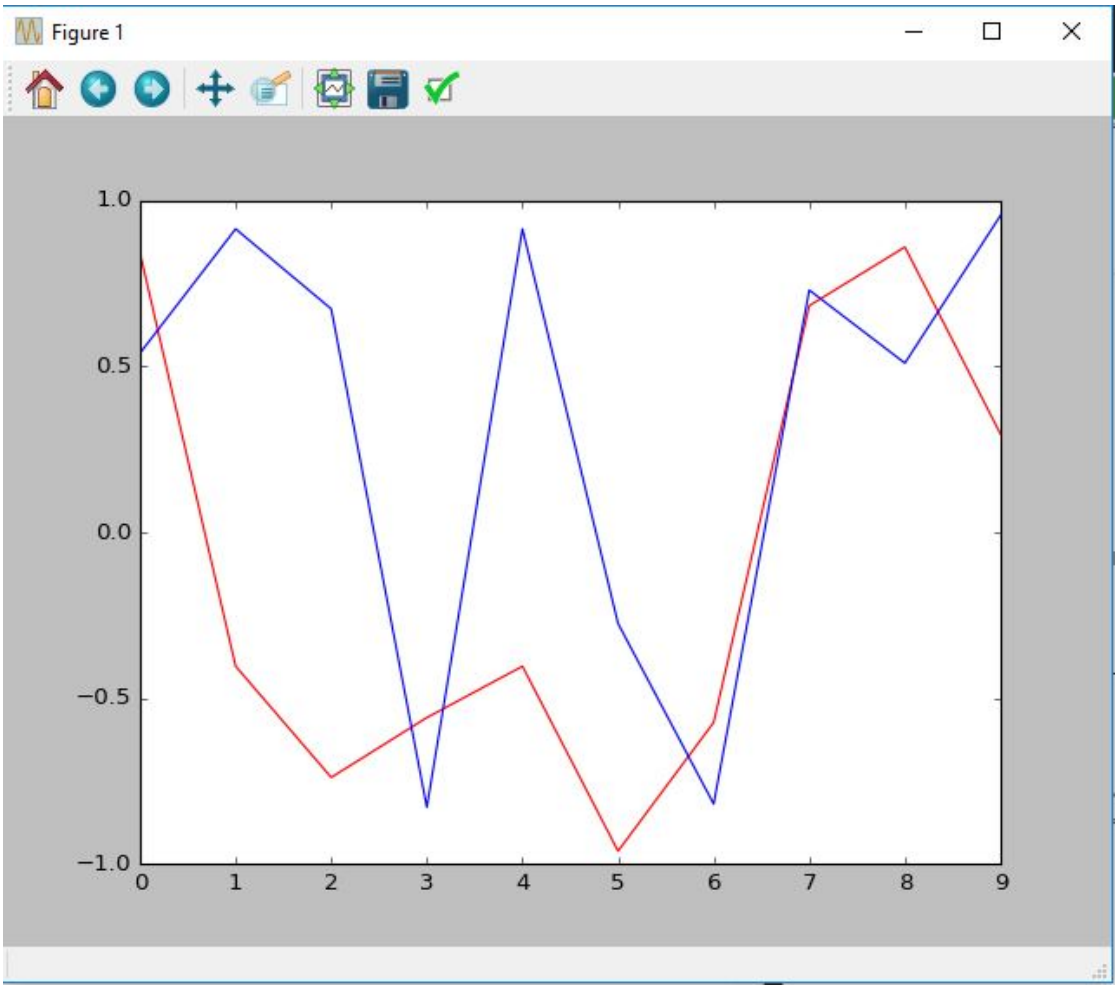


3. from matplotlib import pyplot as plt
import math
import random
x = 0
sin = []
cosin = []
while (x < 10):
y = random.randint(0, 99)
s = math.sin(y)
c = math.cos(y)
sin.insert(x, s)
cosin.insert(x, c)
x += 1
print(sin)
print(cosine)



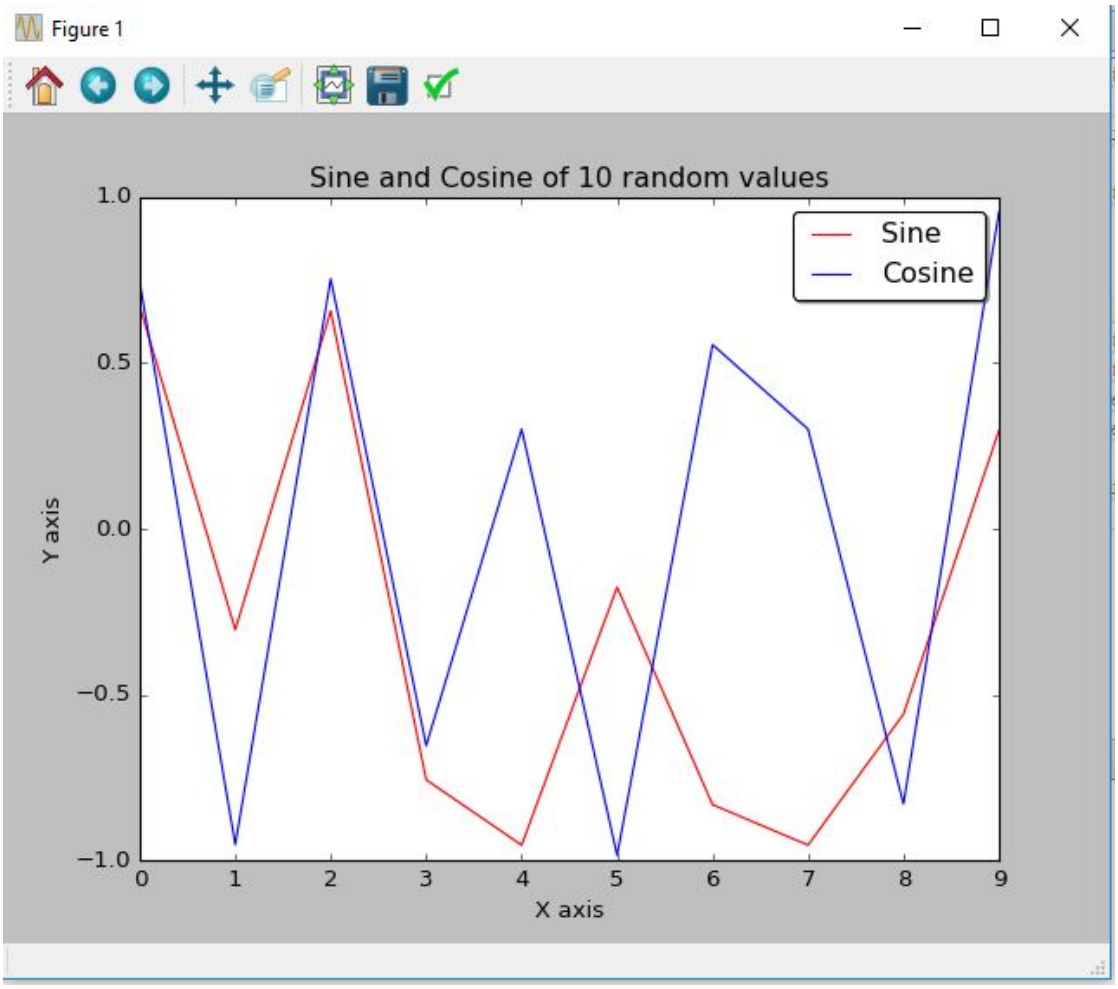
4. from matplotlib import pyplot as plt
import math
import random
x = 0
sin = []
cosin = []
while (x < 10):
y = random.randint(0, 99)
s = math.sin(y)
c = math.cos(y)

```
sin.insert(x, s)
cosin.insert(x, c)
x += 1
plt.plot(sin, color="red")
plt.plot(cosin, color="blue")
plt.show()
```



5.
```
from matplotlib import pyplot as plt
import math
import random
x = 0
sin = []
cosin = []
while (x < 10):
y = random.randint(0, 99)
```

```
s = math.sin(y)
c = math.cos(y)
sin.insert(x, s)
cosin.insert(x, c)
x += 1
plt.plot(sin, color="red", label="Sine")
plt.plot(cosin, color="blue", label="Cosine")
plt.title('Sine and Cosine of 10 random values')
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.legend( shadow=True, fancybox=True)
plt.show()
```



Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment1/` 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 github 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 latex template was created by Lukas Schmelzeisen for the tutorials of "Web Information Retrieval".