Introduction to Web Science

Assignment 2

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 9, 2016, 10:00 a.m.
Tutorial on: November 11th, 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: uniform
Group members: Pradip Giri, Jalak Arvind Kumar Pansuriya, Madhu Rakhal Magar

1 IP Packet (5 Points)

Consider the IPv4 packet that is received as:

```
4500 062A 42A1 8001 4210 XXXX C0A8 0001 C0A8 0003
```

Consider `XXXX` to be the check sum field that needs to be sent with the packet.

Please provide a step-by-step process for calculating the "Check Sum".

Answer:

1. Sum of the packet header:
   4500 + 062A = 4B2A
   Like this sum up other headers
   4B2A + 8001 + 4210 + XXXX + C0A8 + 0001 + C0A8 + 0003 = 2D130
2. Add carry '2' to the rest of the result
   D130 + 2 = D132
3. Convert the result into binary
   1101 0001 0011 0010
4. Get the Check Sum by flipping 1s and 0s
   0010 1110 1100 1101
5. Convert the last result back into hexadecimal
   Check Sum = 2ECD

2 Routing Algorithm (10 Points)

UPDATE. The bold fonted numbers have been updated on Monday Nov. 7th. (If you already have done so feel free to use the old numbers. But the solution with the old version will be more complex than the solution with the updated numbers.)

You have seen how routing tables can be used to see how the packets are transferred across different networks. Using the routing tables below of Router 1, 2 and 3:

1. Draw the network `[6 points]`
2. Find the shortest path of sending information from 67.68.2.10 network to 25.30.3.13 network `[4 points]`

Table 1: Router 1

Destination	Next Hop	Interface
67.0.0.0	67.68.3.1	eth 0
62.0.0.0	62.4.31.7	eth 1
88.0.0.0	88.4.32.6	eth 2
141.**71**.0.0	141.**71**.20.1	eth 3
26.0.0.0	141.71.26.3	eth 3
156.3.0.0	141.71.26.3	eth 3
205.**30.7**.0	141.71.26.3	eth 3
25.0.0.0	88.6.32.1	eth 2
121.0.0.0	88.6.32.1	eth 2

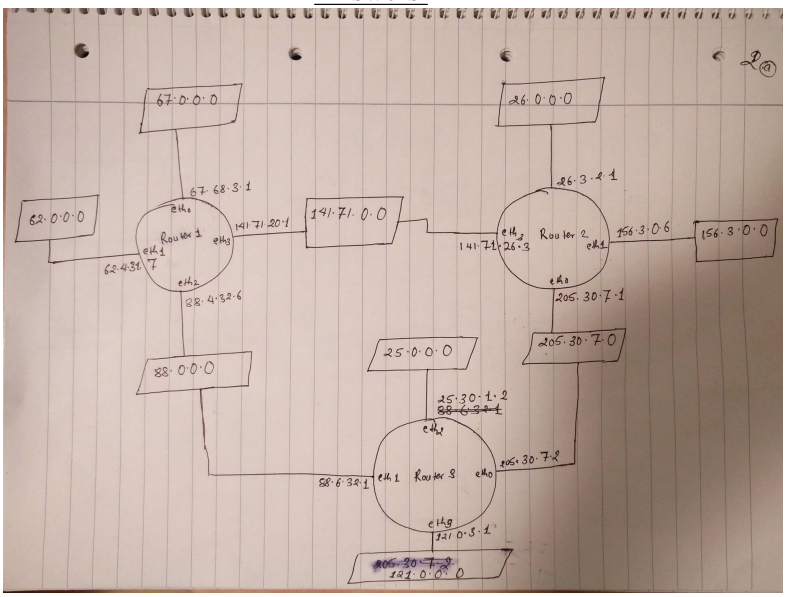
Table 2: Router 2

Destination	Next Hop	Interface
141.**71**.0.0	141.71.26.3	eth 3
205.**30.7**.0	205.**30.7**.1	eth 0
26.0.0.0	26.3.2.1	eth 2
156.**3**.0.0	156.3.0.6	eth 1
67.0.0.0	141.**71**.20.1	eth 3
62.0.0.0	141.**71**.20.1	eth 3
88.0.0.0	141.**71**.20.1	eth 3
25.0.0.0	205.30.7.2	eth 0
121.0.0.0	205.30.7.2	eth 0

Table 3: Router 3

Destination	Next Hop	Interface
205.**30.7**.0	205.30.7.2	eth 0
88.0.0.0	88.6.32.1	eth 1
25.0.0.0	25.30.1.2	eth 2
121.0.0.0	121.0.3.1	eth 3
156.**3**.0.0	205.**30**.7.1	eth 0
26.0.0.0	205.**30**.7.1	eth 0
141.0.0.0	205.**30**.7.1	eth 0
67.0.0.0	88.4.32.6	eth 1
62.0.0.0	88.4.32.6	eth 1

Answers: 2.1



2.2 Here is the shortest path description to go from 67.68.2.10 network to 25.30.3.13

1. When the information is sent from 67.68.2.10 computer, it goes to the Router 1 via eth0 whose IP address is 67.68.3.1 .
2. When the router 1 gets the information it looks the IP header of information(i.e from Ethernet frame) and sends the data to the 88.0.0.0 network via eth2 whose IP address is 88.4.32.6 .
3. The information gets into the router3 via eth1 whose IP address is 88.6.32.1 .
4. Now the router 3 looks in the IP header of information and sends the data to 25.0.0.0 network via eth2 whose IP address is 25.30.1.2 .
5. Finally data arrived at 25.30.3.13

3 Sliding Window Protocol (10 Points)

Sliding window algorithm, which allows a sender to have more than one unacknowledged packet "in flight" at a time, improves network throughput.

Let us consider you have 2 Wide Area Networks. One with a bandwidth of 10 Mbps (Delay of 20 ms) and the other with 1 Mbps (Delay of 30 ms) . If a packet is considered to be of size 10kb. Calculate the window size of number of packets necessary for Sliding Window Protocol. [`5 points`]

Answers:
Given:
The bandwidth of first WAN (B1) = 10 Mbps
The bandwidth of second WAN (B2) = 1 Mbps
Delay for first WAN (D1) = 20 ms = 0.02 s
Delay for second WAN(D2) = 30 ms = 0.03 s
Packet size (P) = 10 Kb
Now, We can calculate the window size by using formula below: Window Size(WS) = 2 * Bandwidth(B) * Delay(D)
So,
Window size for first WAN = 2 * B1*D1
= 2 * 10 Mbps * 0.02s
= 0.040 Mb
= 400 Kb

Window size for second WAN = 2*B2*D1
= 2 * 1Mbps * 0.03s
= 0. 06 Mb
= 60 Kb
Now,
The window size of number of packets for first WAN = 400 Kb / 10 Kb
= 40 packets
The window size of number of packets for second WAN = 60 Kb / 10 Kb
= 6 packets

Since you now understand the concept of Window Size for Sliding Window Protocol and how to calculate it, consider a window size of 3 packets and you have 7 packets to send. Draw the process of `Selective Repeat Sliding Window Protocol` where in the 3rd packet from the sender is lost while transmission. Show diagrammatically how the system reacts when a packet is not received and how it recuperates from that scenario. [`5 points`]

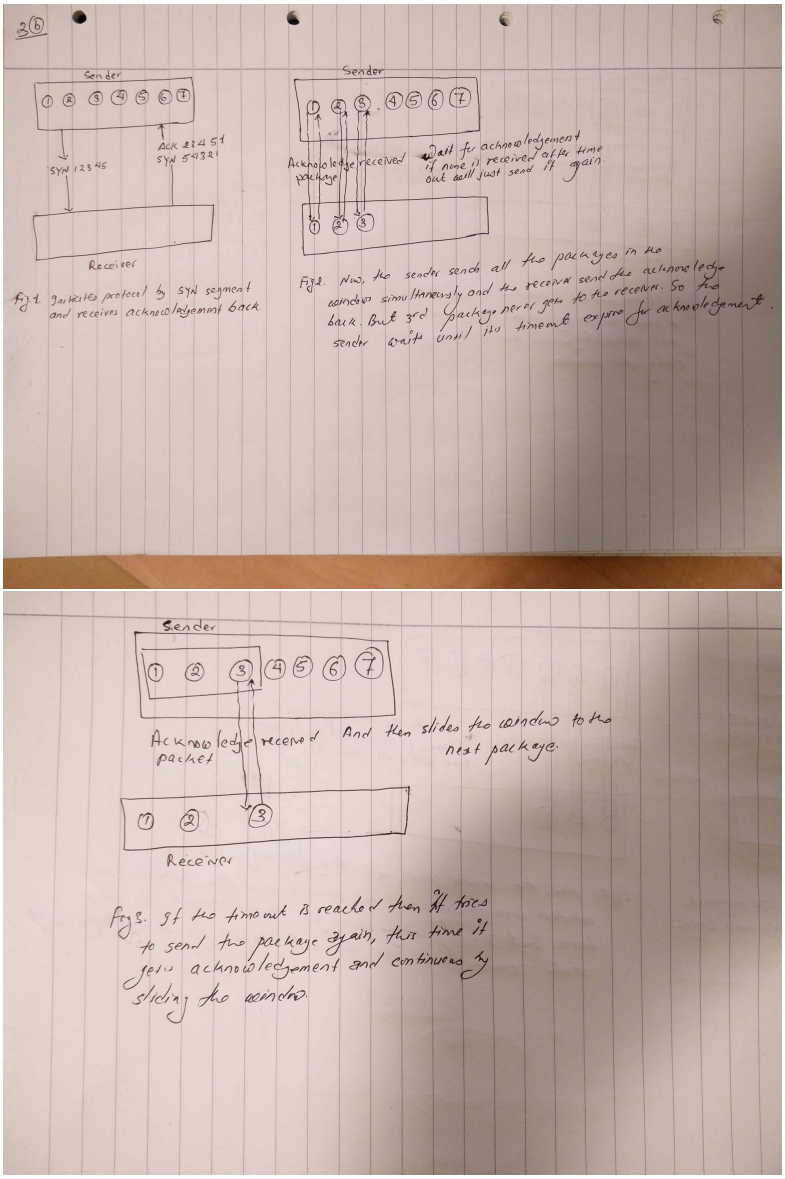


Fig1. Initiates protocol by SYN segment and receives acknowledgement back.

Fig2. Now, the sender sends all the packages in the window simultaneously and the receiver sends the acknowledge back. But 3rd package never goes to the receiver. So the sender waits until the timeout expires for acknowledgement.

Fig3. If the timeout is reached then it tries to send the package again, this time it gets acknowledgement and continues by sliding the window.

4 TCP Client Server (10 Points)

Use the information from the socket documentation and create: [4 points]

1. a simple TCP Server that listens to a
2. Client

Note: Please use port 8080 for communication on localhost for client server communication.

Given below are the following points that your client and server must perform: [6 points]

1. The *Client* side asks the user to input their name, age & *matrikelnummer* which is then sent to the server all together.
2. Develop a protocol for sending these three information and subsequently receiving each of the information in three different lines as mentioned in the below format. Provide reasons for the protocol you implemented.
3. Format the output in a readable format as:
   Name: Korok Sengupta;
   Age: 29;
   Matrikelnummer: 21223ert56

Provide a snapshot of the results along with the code.

Answer

1. Client

```
#!/usr/bin/env python
# pylint: disable-msg=C0103
"""
    Simple python programme which asks for
    name age and Matrikelnummer nummber
"""
import socket
from collections import OrderedDict

BUFFER_SIZE = 1024
HOST = '127.0.0.1'
PORT = 8080


print("Please enter your name")
name = input()
```

```
17:
18: print("Please enter your age")
19: age = input()
20:
21: print("Please enter your Matrikelnummer")
22: matrikelnummer = input()
23: data = OrderedDict()
24: data['name'] = name
25: data['age'] = age
26: data['matrikelnummer'] = matrikelnummer
27:
28: s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29: s.connect((HOST, PORT))
30: s.send(str(data).encode())
31: response = s.recv(BUFFER_SIZE)
32: s.close()
```

```
~/d/a/w/u/assignment-2 git:master ››› python client.py
Please enter your name
madhu rakhal magar
Please enter your age
30
Please enter your Matrikelnummer
216203676
```

2. Server

```
1: # pylint: disable-msg=C0103
2: """ Simple Web Server in Python
3:     using socket
4: """
5: import socket
6: from collections import OrderedDict
7:
8: HOST = '127.0.0.1'
9: PORT = 8080
10: BUFFER_SIZE = 1024
11:
12: s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13: s.bind((HOST, PORT))
14: s.listen(1)
15: conn, addr = s.accept()
16: while True:
17:   data = conn.recv(BUFFER_SIZE)
18:   if not data: break
19:   info = eval(data)
20:   for item in info.items():
21:     print(item[0].title() + " : " + item[1].title())
22:   conn.send(data)
```

```
23: conn.close()
```

```
~/d/a/w/u/assignment-2 git:master ››› python server.py
Name : Madhu Rakhal Magar
Age : 30
Matrikelnummer : 216203676
```

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment2/` 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".

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, go to settings and change the LaTeXengine to `LuaLaTeX` in case you encounter any error