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.

Team: **Papa**
Team members:

1. Brigitte Aznar

2. Bonasmitha Behura

3. Ilia Tugushi

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".

1. Calculating the sum of the header
   4500 + 062A + 42A1 + 8001 + 4210 + C0A8 + 0001 + C0A8 + 0003 = **2D130**

2. Transform the result into binary
   0010 1101 0001 0011 0000

3. Add the first 4 bits to the rest of the value
   0010 + 1101 0001 0011 0000 = **1101 0001 0011 0010**

4. Flip the 1s and 0s to get the check sum
   1101 0001 0011 0010
   0010 1110 1100 1101

5. Transform the previous result back into hex
   The check sum is **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
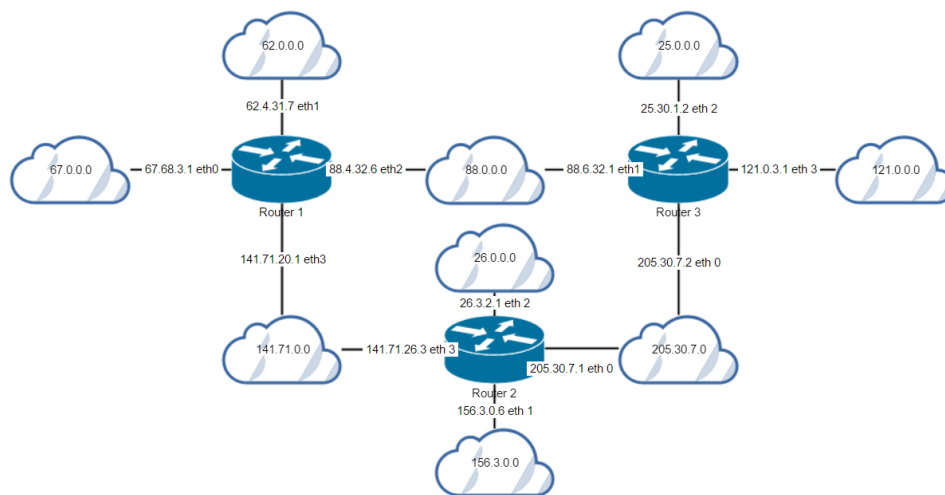
Figure 1: Network diagram

1.

2. a) Goes from 67.0.0.0 network to Router 1

    b) Gets out of Router 1 through eth2 to the 88.0.0.0 network

    c) Gets into Router 3 trough eth 1

    d) Finally it's sent to the 25.0.0.0 network through eth2 and finds 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]

$$p = 10kb$$

$$D_1 = 20ms = 0.02s$$

$$B_1 = 10Mbps$$

$$D_2 = 30ms = 0.03s$$

$$B_2 = 1Mbps$$

$$W = 2 * B * D$$

$$W_1 = 2 * 10Mb/s * 0.02s = 0.40Mb = 400Kb$$

$$WP_1 = 400kb/10kb = \textbf{40 packages}$$

$$W_2 = 2 * 1Mb/s * 0.03s = 0.06Mb = 60Kb$$

$$WP_2 = 60kb/10kb = \textbf{6 packages}$$

For WAN 1 the window package size is 40 and for WAN 2 is 6

** being W the window size in Kb
WP the number of packages per window
B the bandwidth
D delay

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**]
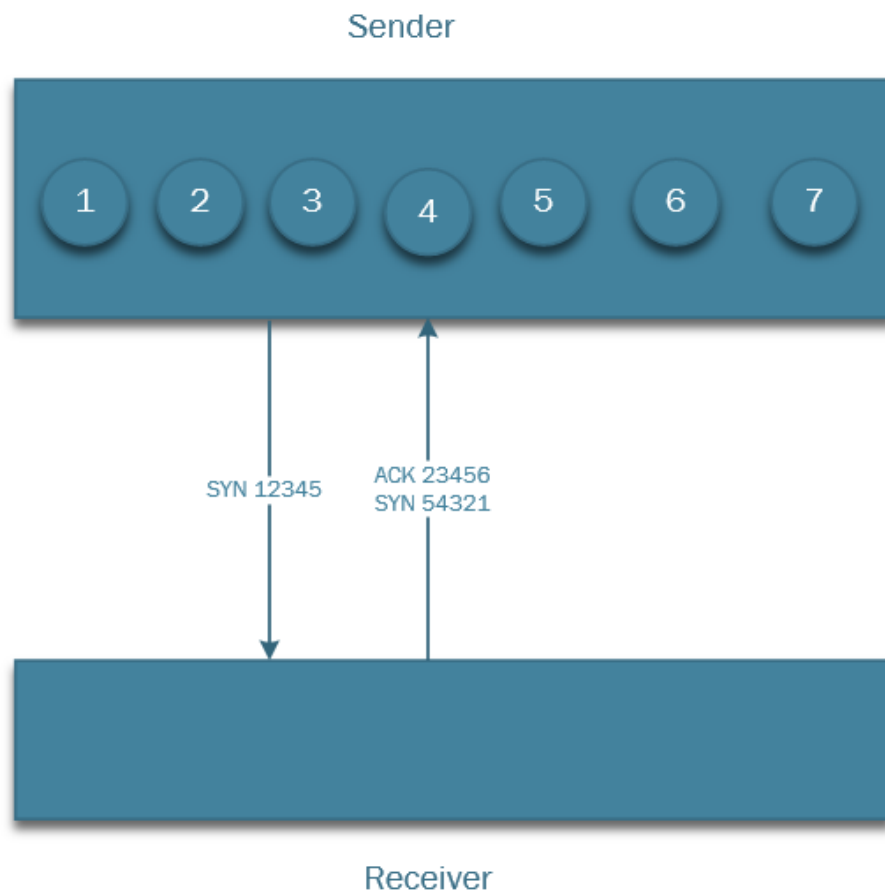
Figure 2: Initiates protocol by sending SYN segment and receives acknowledgment back
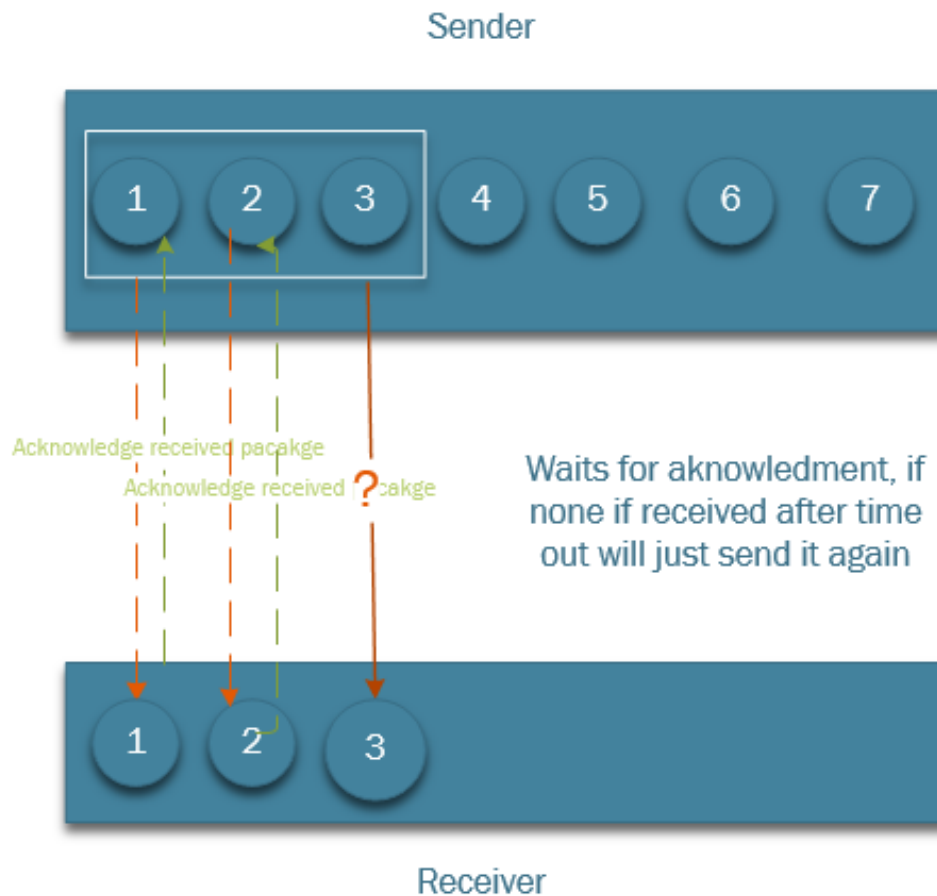
Figure 3: Now the sender sends all the packages in the windows simultaneously and the receiver send the acknowledgement back. But 3rd package never gets to the receiver, so the sender waits until its timeout expires for acknowledgement
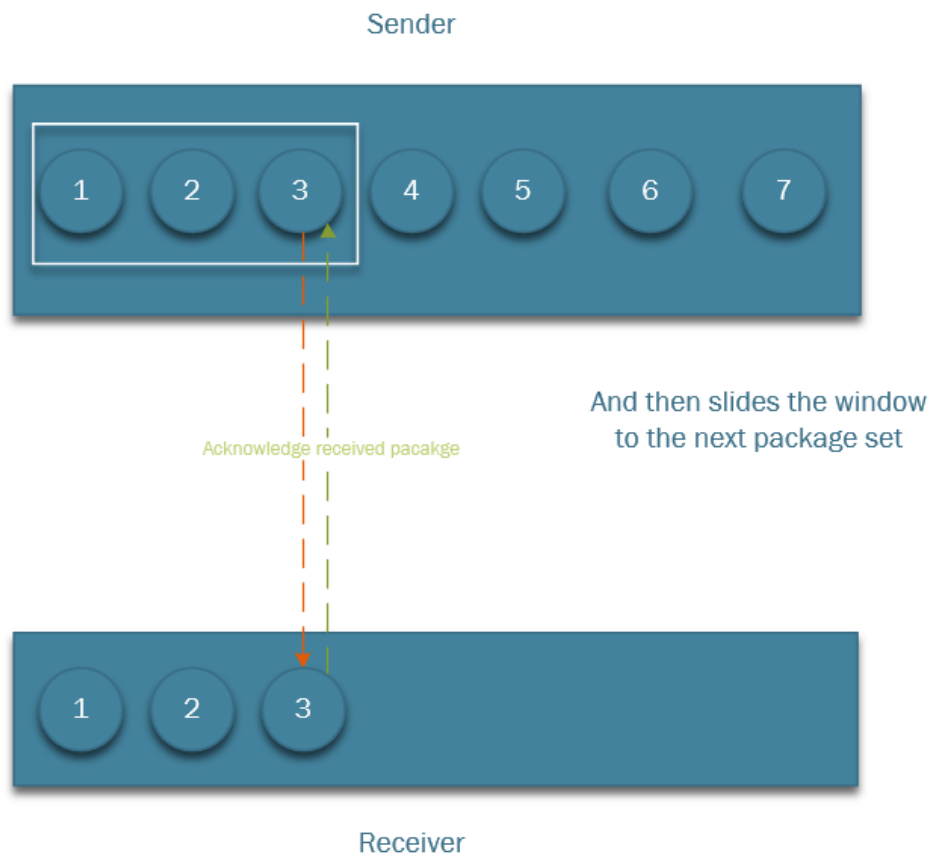
Sender

And then slides the window
to the next package set

Acknowledge received pacakge

Receiver

Figure 4: If the timeout is reached then he tries to send the package again, this time it gets its acknowledgment 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

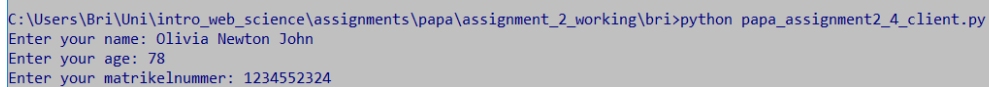Provide a snapshot of the results along with the code.

For the server side:

```
 1: #papa_assignment2_4_server.py
 2: import socket
 3: import json
 4:
 5: def Main():
 6:
 7:     socket_server = socket.socket()
 8:     socket_server.bind(('localhost', 8080))
 9:
10:     socket_server.listen(1)
11:     conn, addr = socket_server.accept()
12:     print("Connection from: " + str(addr))
13:     data = conn.recv(1024)
14:     data = data.decode('utf-8')
15:     if not data:
16:         print('no data received')
17:         return
```

```
18:
19:      data = json.loads(data)
20:      print('Name:  {}'.format(data['name'] ))
21:      print('Age: {}'.format(data['age'] ))
22:      print('Matrikelnummer: {}'.format(data['matrikel'] ))
23:
24:      conn.close()
25:
26:
27: if __name__ == '__main__':
28:      Main()
```

For the client side:

```
 1: #papa_assignment2_4_client.py
 2: import socket
 3: import json
 4: def Main():
 5:      host = 'localhost'
 6:      port = 8080
 7:
 8:      name     = input('Enter your name: ')
 9:      age      = input('Enter your age: ')
10:      m_nummer = input('Enter your matrikelnummer: ')
11:
12:      clientSocket = socket.socket()
13:      clientSocket.connect((host, port))
14:
15:      data = {}
16:      data['name'] = name
17:      data['age'] = age
18:      data['matrikel'] = m_nummer
19:      json_data = json.dumps(data)
20:      clientSocket.send(json_data.encode('utf-8'))
21:
22:      clientSocket.close()
23:
24:
25: if __name__ == '__main__':
26:      Main()
```

```
C:\Users\Bri\Uni\intro_web_science\assignments\papa\assignment_2_working\bri>python papa_assignment2_4_client.py
Enter your name: Olivia Newton John
Enter your age: 78
Enter your matrikelnummer: 1234552324
```

Figure 5: Client results

```
C:\Users\Bri\Uni\intro_web_science\assignments\papa\assignment_2_working\bri>python papa_assignment2_4_server.py
Connection from: ('127.0.0.1', 2611)
Name: Olivia Newton John
Age: 78
Matrikelnummer: 1234552324
```

Figure 6: Server results

We used json because is a very clean way to send and receive data, it is also safe to decode/encode.

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