Introduction to Web Science

Assignment 2

Prof. Dr. Steffen Staab   René Pickhardt

staab@uni-koblenz.de   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 Bravo
Daniyal Akbari (akbari@uni-koblenz.de)
Shriharsh Ambhore (ashriharsh@uni-koblenz.de)
Kandhasamy Rajasekaran (kandhasamy@uni-koblenz.de)

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. Add up all the 16 bits together except the checksum 4500 + 062A + 42A1 + 8001+ 4210 + C0A8 + 0001 + C0A8 + 0003 = 2 D130
2. If there is a carry then add it
2 is the carry. So 2 + D130 = D132
3. If there is still a carry then add that as well.
there is no carry now
4. Get the One's complement - 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

3 Network Drawing

1)
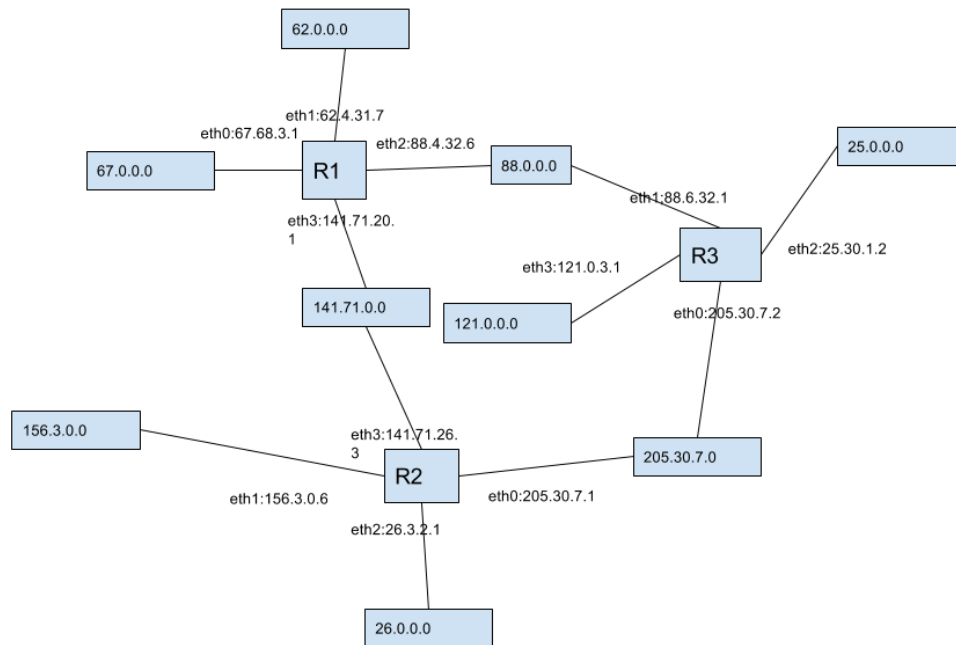
2) Shortest path from 67.68.2.10 to 25.30.3.13

67.68.2.10 to 67.68.3.1 (R1 eth0)
R1 to 88.6.33.2.1 (R3 eth1)
R3 to 25.30.3.13

4 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`]

The transmission time of a data packet in first WAN is (10 Kb)/(10 Mbps) = 1msec.

The transmission time of a data packet in second WAN is (10 Kb)/(1 Mbps) = 10msec.

The total delay time is 20+30 = 50 msec. Thus an RTT for sending a data packet and receiving an acknowledgment for it is 50+1+10 = 61 msec.

To go as fast as possible, First WAN must use a window that is at least as large as the bandwidth-delay product, which is (10 Mbps)(61 msec) = 610000 Bits.

The question asks for how many data packets, which is given by:

Ceiling[610000 b / (10 Kb/pkt)] = 61 pkts

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`]

(a)

(b)
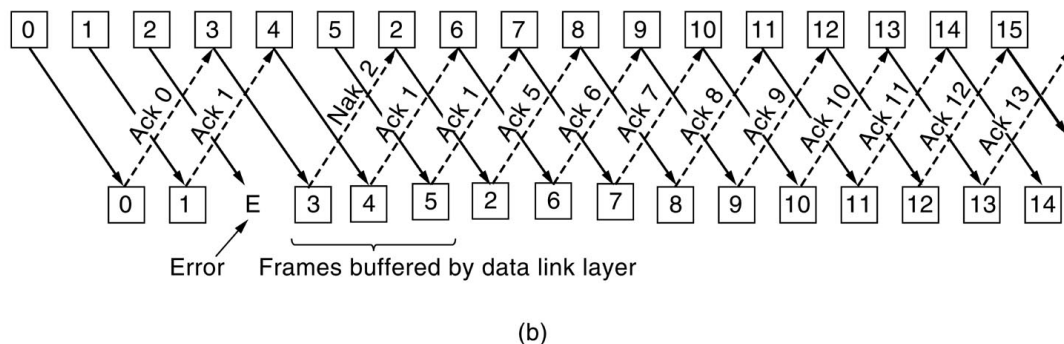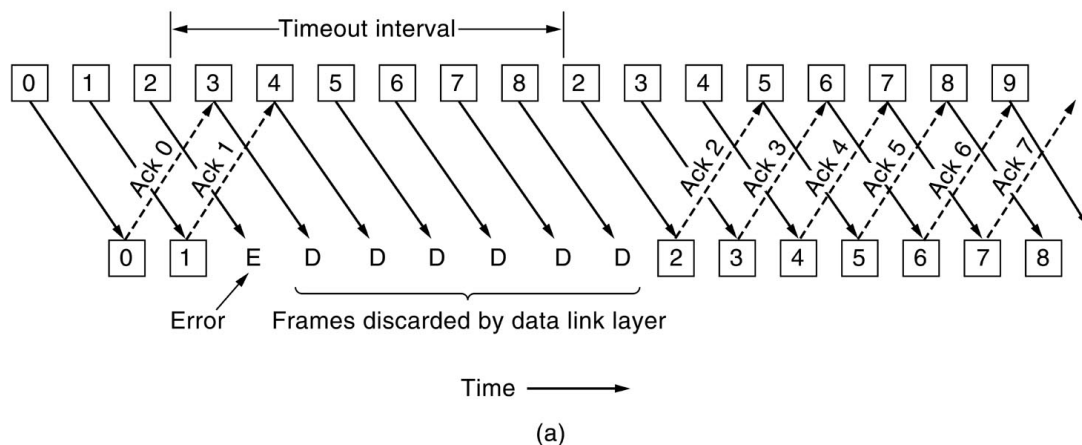
Receiver B window size = 1 Discard all frames after error No ack - they will eventually get re-sent Can be lot of re-sends Wastes lot of bandwidth if error rate high.

DA needs buffers in memory to hold frames waiting for ack (might need to re-transmit them). Acks may get lost/damaged. "ack n" is interpreted by A as "ack everything up to n". Each frame has its own timer. Easy to simulate multiple timers in software.

Selective repeat (SR) (b) above

Receiver B window size = n Good frames after error are buffered (because can't send them to Network Layer NB yet, can't get rid of them) Bad frame 2 times out and is re-sent. At a timeout, sender A only sends the oldest un-ack-ed frame, not a sequence.

Or (in figure) B sends a Negative Ack (NAK) to provoke a re-send of frame 2 instead of waiting for timeout - may improve performance. If you know it will timeout, why wait? Notice B doesn't ack 3-5 while it is buffering them. It sends back "Ack 1" which means "Everything ok up to and including 1". Once 2 gets through, B can send a whole bunch (2-5) to NB layer and can then jump to "Ack 5" ("Everything ok up to and including 5")

7

5 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.

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