PART2

In this homework, you are expected to implement a **UDP-based** "**Reliable Data Transfer**" (**RDT**) protocol of **your own (not same as RDT 3.0, etc. in the textbook and literature)** that supports **pipelining and multi-homing**. You are also expected to modify your routing implementations based on the requirements of the file transmission. Note that your protocol should be implemented by considering your own approach and design. You will concentrate on the above topology in part A which each node has disjoint paths. The **source** will be the host **s,** and the **destination** will be the host **d**.

1. Specifications

- You can use any programming language to implement your code, as soon as you explain the usage of it.
- All nodes should handle the links in a reliable fashion.
- Assume that a sender and receiver as an application layer implementation on the node "s" and the node "d" are running.
- Your source node "s" will send this large file (exactly 5 MBytes) to the destination node "d".
- **The file transmission is performed for two experiments: The source node will send the specified file to the destination node by using two different paths by conducting two different experiments:**
    - **The first path will be the shortest one based on the Dijkstra Algorithm that you found in the first part, and the second one will use the other available links "at the same time". These two different transmissions should be handled by using the same script. You can give the number of experiments as a parameter to your scripts and run them for two experiments.**
    - **For Experiment 1**, your hosts (nodes) should route each of the packet based on the shortest path. This means that your file transmission will be over the shortest-path from the source node (s) to the destination node (d). The naming convention for the file can be for this transmission like input1 on the node "s" and output1 on the node "d".
    - **In Experiment 2,** the source node exploits the remaining available links of (two) for a copy of this input file. The naming convention for the file can be like input2 on the node "s" and output2 on the node "d" for this transmission.
        - Exploit disjoint links between the source and destination. We will use the advantage of two links while transferring the file, exploiting these multiple paths will allow us to transfer the file faster. This part will provide us a **multi-homed protocol** that you are expected to implement.
        - **Assume that the shortest path is s-->r3-->d for experiment 1, and, then the second file transmission (for experiment 2) routing information should be like the following: (Use these paths at the same time) for the first**

path: s --> r1 -->d and the second path s-->r2-->d, and for response messages also follow the inverse of these paths. The links between the r1 and r2; r2 and r3 will not be used.

- ***The network may be faced with one type of failure which is link failure. One of the links between the source node "s" and two other nodes (r1 and r2) can be down. Your reliable and multi-homed implementation should be aware of the links and update the routing and modifying the reliable transmission over the available link. This part will be applied only for the second experiment (Experiment 2).***
-
- ***Please write generic code since the network topology can be changed while evaluating your scripts: the links can be down. To down of a link:***

```
ip link set dev <interface> down
```

-
  - **For two experiments:**
    - The files will be protected with a **checksum** in the assessment. In other words, the input file at the source side, and the transmitted file at the destination side should be exactly the same.
    - You will implement and develop your own RDT protocol on top UDP. Therefore, you will use your own reliable protocol to send a large file from s to d. Then, the communications will be based on your RDT protocol.
    - Develop your reliable transport protocol that supports multi-homing by using UDP sockets.
    - Develop a packet-based protocol.
    - Design your packet structure that will go into the UDP payload. **The maximum packet sizes of your RDT protocol (header + payload) can be at most 1000 bytes.**
    - **Here, your codes will be graded based on only the above topology.**
    - **Only one script will run in each host**

2. Experimental Results

>>> You are expected to plot the following figure for **two experiments above** by using your own reliable protocols for each experiment.

- x-axis: **packet loss percentage (5%, 15%, 38%)**, y-axis: **file transfer time** with 95% confidence intervals. Keep the delay the same as 3 ms. Packet loss will be applied for all links. Notice that initial configurations should also be applied explained below. In this figure, the results of your RDT protocol should be represented as a line chart or bar graph. You will present your results by considering all the experiments. (**All of these configurations should be applied to all links of all the nodes in the topology.**)

```
tc qdisc change dev [INTERFACE] root netem loss 5% delay 3ms

tc qdisc change dev [INTERFACE] root netem loss 15% delay 3ms

tc qdisc change dev [INTERFACE] root netem loss 38% delay 3ms
```

Notice that, for plotting the **confidence interva**l you will have to repeat each experiment multiple times, say *n*. The margin of error less must be than 2.5%. The z-score is 1.96 for 95% confidence. We expect that the standard error (deviation) will become smaller as you increase *n*. You will have to compute the value of *n* in your experiments based on the standard error you achieve.

3. Reports

- Use latex (you will use the same template) to write the report. It should not be less than 6 pages and more than 8-9 pages.
- **Please write your problem & solution, the methodology to overview of your codes, which mechanisms use to provide** reliability, your scripts, and your **approaches. In your reports try to analyze and demonstrate your results. Differences between the first and second submissions type, and put some screenshots to show the multihoming part.**
- **You are expected to discuss and compare the file transmission time durations, the implementation details of the first and the second submission type.**

- **In your reports,** you should explain all of the developments and implementations, strategies, approaches, their reasons for your own protocol. For instance: How do you provide **multi-homing**, **pipelining** (if you use go-back-n, or selective repeat or your method, etc. Explain them), which mechanisms you use to implement RDT(**checksum**, **re-transmission, ACK/NACK mechanism, sequence number, buffering, etc**.) and which of such mechanisms you use to **handle the loss**, and how it affects your implementation.
- Any design issue or algorithm selection is based on your decisions as long as it provides the transmission of a large file between two hosts in a reliable and pipelining multi-homed approach. It is important that you should explain your solutions to the problems clearly and explicitly. You should share your related code segments (for multihoming handler and pipelining) and explain them in your report.
- Please also be sure that you are expected to make a **read me file**. For its content refer to the assignment text. Your configurations step by step, synchronization of the nodes, your first IP tables, all of the tc/netem usages, etc. In addition to your how to execute your scripts for each experiment and for each step.

- Please put your coding comments into all of your scripts. All of them should be explained the reasons for the usage.

- After obtaining the results and plotting these figures; report the figures and write your comments about the figures. You are expected to write at least one paragraph per figure including your technical discussions. Ensure that the figures are self-explanatory: labels, captions, and units are clearly stated in the figures. Use the paragraph to draw the attention to the most significant points of the figure, give specifics but do not repeat details. Ensure image clarity. Use legends.

You also have to provide the code, and "how to run" within a README file. At the top of the README file, you should write ID, name, and surname of both group members as commented out. ReadMe File should also include all configuration commands step by step (from time synchronization to using routing command). For the network configuration part show your IP tables, your used commands to make the configuration and other settings. Please explain all of the steps you follow to execute your scripts for each experiment.

4. Submission

Use ODTUCLASS to submit the homework. There will be two submission steps: one of them is the submission of your implementation files and the other one is your report submission. The first step will be performed under this Assignment and the other one will be under the Turnitin Assignment. Thus,

**** Submit your codes, latex files, and README file to **this assignment (TP-2 Implementation Submission)** as a single file named as TP_part2_##.tar.gz. Then,

**** Submit your report to "**TP-2 Report Submission**" as Tp_part2_##.pdf.

Do not forget the replace ## with your group number.

This assignment is a Turnitin assignment. According to METU regulations, we will tolerate up to 20% of similarity results.

If your result is bigger than 20%, your assignment will be evaluated as zero.

ONLY ONE GROUP MEMBER SHOULD SUBMIT THE HOMEWORK.

Your source codes will be checked with the codes with different resources in terms of academic dishonesty.

- # 5. Suggestions

* While plotting the figures, MatLab, Octave or Gnuplot will ease your job if you can set up the output accordingly.

* Each group member may get different grades. Sharing the workload is OK, but do not separate the coding part & the reporting part completely. **Express clearly in the document who has done what!**

* You can use any programming language.

* You can use any idea or approach unless it is specified in this document.

- # 6. Evaluation (TENTATIVELY)

1- Reliable Data Transfer protocol implementation with a proper checksum for the above topology for Experiment 1 **(15) points**. In detail, (15) points for your RDT protocol with **a proper checksum** for the above topology for Experiment 2. In other words, your input file should be transmitted to the destination as it is. If your code is successful regarding **multi-homing and pipelining**, you will get **10,10 points respectively**. **These points also include the explanation of your mechanisms, design, etc. in your reports.**

2- The figure and the associated paragraph will deserve **10 points**.

2.1 Legend, axis labels, caption and units missing: -2 points
2.2 Paragraph is not specific or repeats details: -4 points
2.3 The paragraph derives wrong conclusions or not explanatory reflecting key learning points -4 points

Therefore, there is one figure for two experiments: 10 **points.**

4. Link Characterization, link failure handler, and explanation with screenshots to present the down of a link and the transmission on one of the other links: **5 points.**

5- Discussion about multi-homing implementation in detail explicitly **10 points**.

6.- Latex and clear ReadMe File 10 points.

7. Deadline

The deadline is the 27th of December 2019 at 23:55.