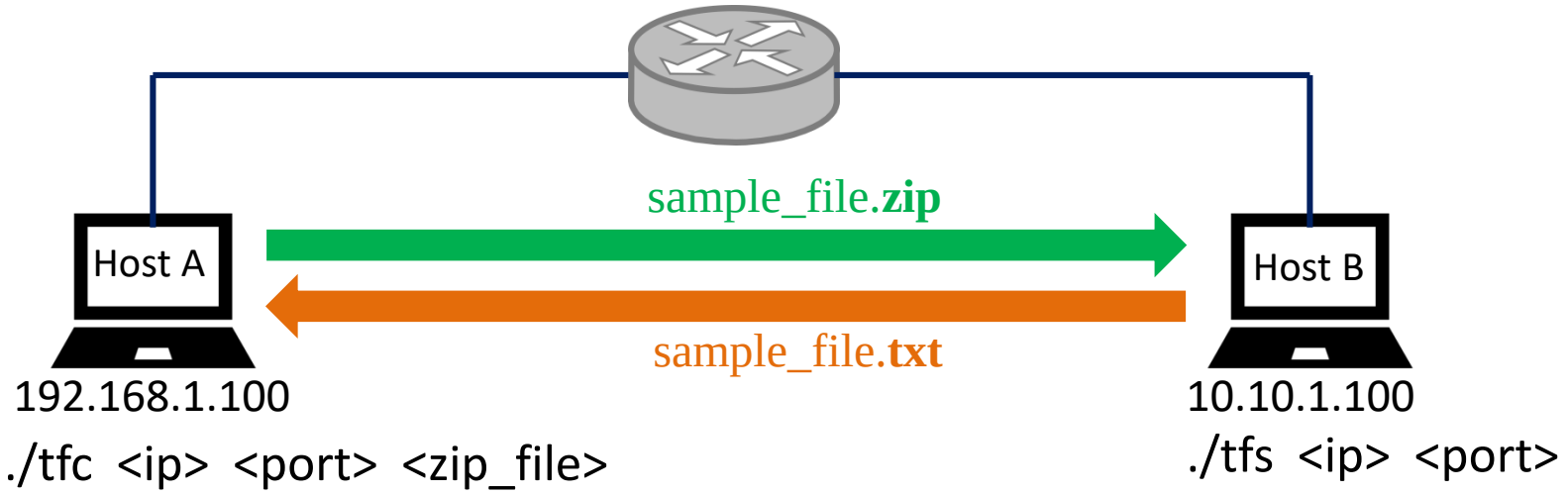
Project 2 (File Transfer/Processing using TCP)



Step 1) Read the given zip file and send it to the server program at host-b.

Step 2) Unzip the received file & send it back to the client.

Step 3) Client side program displays the total duration of the file transfer process in milliseconds.

Step 4) Check the checksum of the original file and the received file (using **md5sum** Linux command)

Project 2 (specifications)

In the second project, you will develop a client/server program for file transfer/processing based on Stream (TCP) sockets. The server program which is called **tfs.c** (TCP File Server) listens on a given **TCP IP address/Port number** at **host-b**. The input arguments of this program are an **IP address** and **Port number**.

On the other side (**host-a**), the client program is called **tfc.c** (TCP File Client) and has 3 input arguments including the **IP address** and **Port number** of the server side as well as the **Name of a zipped file**. The client program sends the given file to the TCP file server and the server program receives and stores the file, first. Then, it unzips the received file and start sending back the received unzipped file to the TCP client program. A sample zip file will be provided for the students to test their code.

The client program receives the unzipped file and store it as a new file. The newly received file must be as same as its original version. To check this, you may use **md5sum** Linux command to calculate/compare the checksum values of the original and received files. Moreover, the client side program (**tfc.c**) must display the total duration (time) of file transfer with millisecond (ms) precision after receiving the file.

After implementation of the introduced programs, you need to write a report and answer the provided questions in the next page.

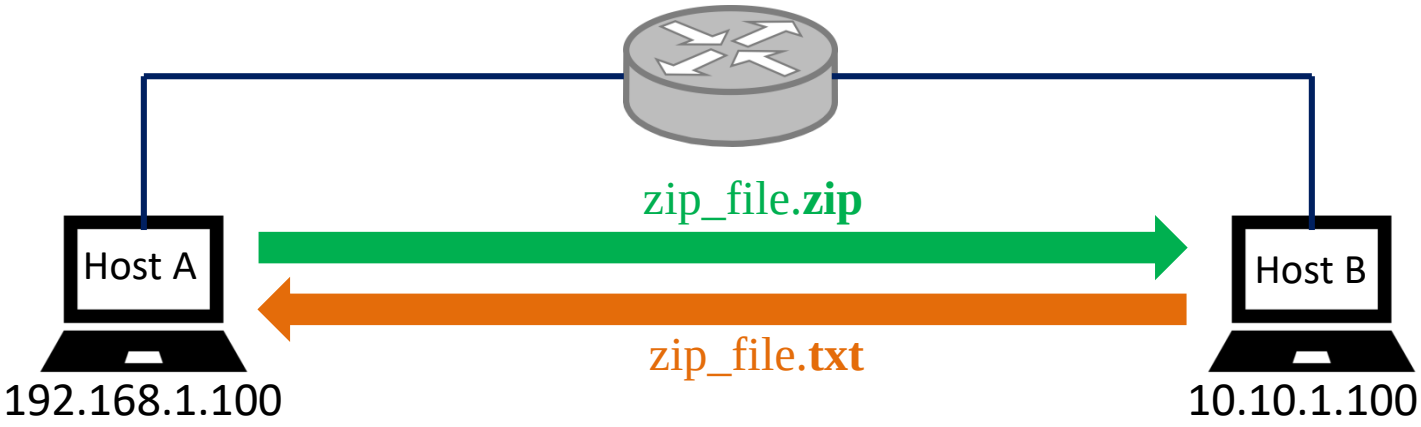
Project 2 (questions)

1. As the first set of results, make sure that the router has no prior configuration to drop the packets and no loss occurs during the file transfer process at the router. You may use '**tc qdisc show**' at the router to check the existing configuration. Then, use **Wireshark** to capture the transferred traffic between two hosts at outgoing interface of router. After finishing the file transfer, use the captured traffic to count the number of transmitted/received packets at both hosts.

2. In the next step, set the packet drop ratio to 20% (using tc command) at the router and then run **Wireshark** to capture the traffic at the ongoing interface of the router. After finishing the file transfer process, count the total number of transmitted/received packets at the both hosts, again. Also, count the total number of retransmitted packets (by the both hosts) for this scenario. You can either use **Wireshark filters** or your own client/server programs to do this task.

3. After getting the results for the explained scenarios, write a short and concise analysis and justify your observations and the outcomes. You can use proper snapshots and graphs to elaborate your justification in the report.

Sample Syntax



./tfc 10.10.1.100 9090 test.zip

./tfs 10.10.1.100 9090

Marking Scheme

5% project report

5% 5 questions from lab materials (since Session 6)

10% running the code (successful file transfer) and answer the related questions