Topic: Implementing RDT over UDP

#
# Instructions for Task 1. Preparation and Preliminary Study
#
Implement the testbed which contains 2 directly connected Linux VMs with 100Mbps link speed (must be configured on Linux VM), and apply the delay and packet loss using "tc" command. For the reference of "tc" command, see the following man page.
http://manpages.ubuntu.com/manpages/hirsute/man8/tc.8.html

# Deliverables of Task 1
1. Perform 10 attempts of FTP (SCP may perform a bit slower than FTP) to transmit 100MB data (*Remark) from the client to the server under the following two conditions: 1) without delay and packet loss, and 2) with 50 ms delay and 5% packet loss for each direction. Report the overall throughput, and time taken to complete to transfer 100 MB data of each attempt in the form of the table or the graph.

*Remark: Download the 100MB data via the following URL.
https://www.dropbox.com/s/fp6knjmayc0kgir/CS3543_100MB?dl=0

# Evaluation Criteria of Task 1
1. Quality and technical justifiability of pictorial illustration and written report.

#
# Instructions for Task 2
# Implementing "our-UDP-FTP"
#
1. Implement "our-UDP-FTP": a simple UDP based FTP tool on the application data transfer . Consider "our-UDP-FTP" as a miniature & UDP version of FTP to transmit a file over UDP and to measure the throughput of file transfer from the client to the server.

2. "Our-UDP-FTP" must have reliable data transfer (RDT) features because UDP does not provide reliable data transfer. The RDT features offered by "Our-UDP-FTP" can include, but not limited to, packet loss detection, acknowledgement, packet retransmission, flow control, congestion control, etc. The goal of "our-UDP-FTP"" is to achieve the best available throughput of the file transfer under adverse conditions with delay and packet loss. As a UDP application, "our-UDP-FTP" shall use "sendto()" to send the application data over UDP.

3. For measurement purpose, "Our-UDP-FTP" must be able to monitor the packet count and the byte count of file chunks which are successfully received at the server.

# Deliverables of Task 2
1. Provide the pictorial illustration of the application header as per the example of various header formats that were covered in the course.
2. Provide the written report and the pictorial illustration of the features that your group implements as part of RDT. Highlight what is special about the features.
3. Perform 10 attempts of "our-UDP-FTP" to transmit 100MB data (*1) from the client to the server under the following two conditions: 1) without delay and packet loss, and 2) with 50 ms delay and 5% packet

```
loss for each direction. Report the overall throughput, and time taken to complete to transfer 100 MB data of each attempt in the form of the table or the graph.
(New) 3. Take the packet capture during the experiment to prove the performance of your system with 3rd-party tools such as Wireshark. The throughput you report and that Wireshark tell should be more or less consistent.
(New) 4. Submit the source code.

#
# Evaluation Criteria of Task 2
#
1. Quality and technical justifiability of pictorial illustration and written report.
2. Is the idea or design principle of RDT features justifiable?
3. Does the file transfer complete using Our-UDP-FTP?
4. Is the file transfer using Our-UDP-FTP faster than the regular FTP or not?
5. Is the source code copied from "GeeksforGeeks" or other similar online resources?

# Chances of getting lower marks
1. Plagiarism not accepted.
2. If a group falls back to implement GBN or SR as RDT feature, such a group may receive less marks than the other groups that implement original and better RDT features.
```