General information

The measurement tasks can be carried out during or after the lab. In addition, questions can be asked during/after lab (via Teams or e-mail).

To do the tasks Wireshark (or another packet analyzer) is needed. Additionally, for task *4. Passive traffic measurement*, a .pcap file has to be downloaded from the webpage of the measurement. (Processing this .pcap file may take some time.) Since the file is relatively large, it is advised to process it using command line tools, especially if a computer with lower performance is used.

Exercise *3. VoIP traffic filtering and analysis* and *4. Passive traffic measurement* can be done in pairs. If it is done so, please indicate this in the lab report. The other exercises have to be carried out individually.

The lab report has to be submitted in a week after the lab via e-mail, to meszarosa@hit.bme.hu.

The lab report has to contain the evaluation of the results. Simply presenting the data is not enough! It is recommended to use figures for the evaluation. The report should be submitted in pdf format.

Any program can be used to do the tasks. The applications mentioned in the lab guide are simply recommendations. However, obtaining proper results are the responsibility of the sudent. Processing the data can also be done using any tool (Matlab, Excel, arbitrary programming language). The programs/commands used for the lab have to be given in the lab report. Simple commands used during the tasks should be presented in the report. Longer command sequences/programs should be submitted in a separate file.

Additional remarks for the tasks

Task 1.3

For this task, a reasonable choice can be, e.g., a news site (nytimes.com, bbc.com, etc).

The downloaded objects, the corresponding download times, and other information can be tracked using the Network tab of Developer tools (Ctrl + Shift + i).

Https uses encryption. To be able to see the unencrypted version of data, Wireshark has to know the corresponding encryption keys. You can do this with the following (or similar) steps (on Windows):

1. Close Google Chrome.
2. Open the window for environmental variables. The environmental variables can be edited, e.g., by running sysdm.cpl (Win + R and type sysdm.cpl) and choosing the Advanced tab in the window that pops up.

3. Create a new environmental variable with the name SSLKEYLOGFILE and a freely chosen path (including a file name) where the SSL keys will be stored. E.g. C:\Users\cloud\sslkeys.pms
4. Open Wireshark.
5. In the Edit/Preferences option of Wiresharkban, in the Protocols tab, check the „Reassemble out-of-order segments" option for the TCP protocol.
6. Open Google Chrome and open a page that uses HTTPS.
7. In the Edit/Preferences option of Wireshark, in the Protocols tab, go to the TLS protocol and set the (Pre)-Master-Secret log filename to the generated file. (sslkeys.pms in the example)

Some antivirus software can interfere with Wireshark, therefore it is recommended to turn them offf during this task.

Reopen the previously opened HTTPS page. In Wireshark, filter the packets related to the download of this page. If everything was done correctly, you should be able to see the unencrypted messages in the bottom window of Wireshark. Compare the object sizes with those shown by the Network tab in the Developer tools of the browser. To do this, you should create a new column in Wireshark containing the size of the reassambled segment sizes. This can be done under the Edit/Preferences menu option by choosing the Appearance tab and creating a new column under the Column option which contains the tls.reassembled.length field.

Task 2.1

The term "appropriately long" (video/download) is not a precise description. Think of the number of samples you need to create good enough statistics, that can also be processed without much difficulty. It is recommended to export the capture in CSV format using the Export Packet Dissections in the File menu, then use a different tool to create the statistics.

Task 2.2

The arrival time of packets is not the same as the inter-arrival time of packets.
Instead of a histogram you can also use the empirical distribution. The main goal is to analyze the behaviour using our statistical tools (e.g. think about what to do do with the outliers).

Task 3.1

Any VoIP application can be used (Skype, Teams, Viber, Zoom, etc. – do not measure Discord twice). Many VoIP program applies silence suppression, therefore it is recommended to transfer voice during the conversation.

As a bonus task you can also examine a video call in addition.

Task 4.

Information about the trace file used for the passive measurement:

The measurement is 15 minutes long, contains around 5 million packets, with maximum packet size of 1500 bytes.
The measurement was carried out on the link of the 157.253.0.0/16 network. During the analysis of the host, only hosts in this network should be considered.
For the length of the packets export the ip.len field. Not every packet has an ip.len field. These packets should be considered to have an unknown packet size, and should be mentioned separately in the statistics.
The TCP and UDP port numbers can be between 0 and 65535 – including 0.

IMPORTANT!
In the original .pcap file, several frame lengths are incorrect, therefore make sure to export the ip.len field (which contains the packet length).
It is also important that the trace contains ICMP messages, thus, if you use tshark for exporting, you should apply the "-E occurrence=f" option, otherwise it will export two ip.len fields.

When using default settings, tshark sends its output to the standard output. We can write this into a file.
An example for using tshark:
```
tshark -r mycap.pcap -E occurrence=f -T fields -e ip.len > mydat.dat
```

To analyze the data, we need the following:

- time information
- packet length
- protocol of the packet
- source/destination port for TCP and UDP

Throughput: To get a smooth curve, it is generally advised to use some kind of sliding window, but for this lab you can simply calculate a throughput for every second separately and create a figure from this.

Protocol statistics: The trace contains several protocols. From these you only need to show TCP, UDP, ICMP and DNS separately, the others can be aggregated under "other protocols". (Of course, you can examine these as well if you wish to.)

Port statistics: Only applicable for the TCP and UDP protocols, but for these it should be done separately. Also check what kind of application corresponds to these port numbers.

Traffic distribution of hosts: For this, it may be advisable to use SplitCap. In this case make sure that only the hosts of the 157.253.0.0/16 network are in the statistics.

When using SplitCap, the recommended workflow (of course you can do it differently if you want to):
1. Split the original .pcap file according to hosts.
2. From the small .pcap files extract the relevant fields and save them into text file(s) with a script that uses tshark.
3. Process the text file(s).

Create a figure that shows what percentage of hosts is responsible for what percentage of the traffic. Provide the data for some special points spearately (e.g.. 90% of traffic).

Flow statistics: For this task using SplitCap is recommended. A potential workflow can be the same as for the hosts. The trace contains exactly 50.000 flows, therefore writing the .pcap files into text files can take quite a while (hours). Therefore it is recommended to to this separately.