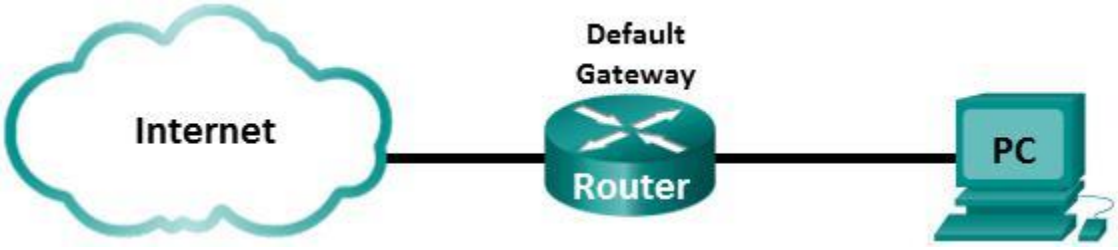
AMC Lab 1 **Instruction** **Deadline: 3.12.2020**

Name: _____Rubaiya Kabir Pranti_____

Simple Network
Internet Access Analysis
VoIP / Video Streaming Analysis



Tasks:

Task 1-3 cover topics from Bachelor level networking classes and helps you to refresh networking knowledge and skills. Task 4 focuses on VoIP streaming.

Task1 - Simple Network and Connectivity Testing

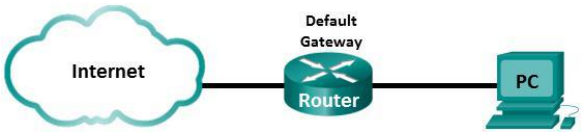
Task 2 – Analyze Latency in your LAN Task 3 –

Internet Connectivity Task 4 – Examine VoIP

Streaming

Task1 - Simple Network and Connectivity Testing

Topology



Part 1: Configure PC Hosts and test connectivity with ICMP Ping

Step 1: Connect your PC to your LAN, via cabled LAN or WLAN, with Default Gateway.

Step 2: IP address information on the PCs.

a. If you run DHCP, DHCP will provide connectivity settings. If you use static IP addresses, manually configure IP address, subnet mask, and default gateway, which fit to your local topology.

b. Use the command prompt window to verify the PC settings and connectivity.

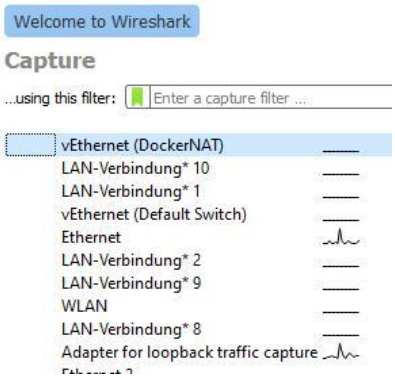
- Host IP address: **IPv4 Address: 192.168.1.103**
- IP network mask: **Subnet Mask: 255.255.255.0**
- Default Gateway IP address: **Default Gateway: 192.168.1.1**
- Record your host MAC address: **Physical Address: 30-5A-3A-8B-D8-AD**

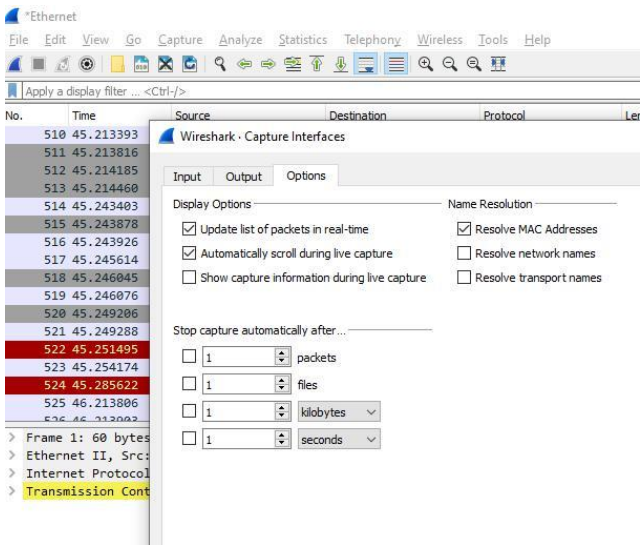
```
Physical Address. . . . . . . . . : 30-5A-3A-8B-D8-AD
DHCP Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::7d6a:3967:8df6:d5e4%12(Preferred)
IPv4 Address. . . . . . . . . . . : 192.168.1.103(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

Part 2: Capture and Analyze Local ICMP Data in Wireshark

Step 1: Start Wireshark and begin capturing data.

a. Start **Wireshark** and select the **Interface**.
By selecting an interface, you **start** a capture.





Note: If multiple interfaces are listed and you are unsure which interface to check, you use **Capture ➔ Options**, where you also find information on the MAC addresses of interface

You should select **automatically scroll during live capture,** if not active**.**

b. Ping your Default Gateway (max. 4 times) and stop the capture. (If not successful, adjust your network setup until it works.)

Step 2: Examine the captured data

a. Filter ICMP traffic in your Wireshark capture.

b. Check the **1st ICMP Echo request** PDU frames in the top section of Wireshark. Record the following:

- Source IP address: **192.168.1.103 (notebook IP address)**
- Destination IP address: **192.168.1.1 (default gateway)**

With this PDU frame still selected in the top section, navigate to the middle section. Click the plus sign to the left of the Ethernet II row to view the Destination and Source MAC addresses.

- Does the Source MAC address match your PC's interface? **Yes**
  Source MAC address in middle section: **ASUSTekC_8b: d8:ad or 30: 5a: 3a: 8b: d8: ad**
  Physical address of notebook from command prompt: **30-5A-3A-8B-D8-AD**

- Record the Destination MAC address,
  which is the MAC address of your Default Gateway.
  Destination MAC address: **TendaTec_69: 1b: e0 or d8: 32: 14: 69: 1b: e0**

Check the ICMP detailed information

- Which hex number represents message type Echo Request (ping)?
  Type: 0×08, Code: 0×00

c. Select the **Ethernet frame**, which contains the **1st ICMP Echo reply** message

- Which MAC address is now destination MAC address?
  ASUSTekC_8b: d8:ad or 30: 5a: 3a: 8b: d8: ad
- How many Bytes have been captured in total? **74 Bytes**
- How many Bytes are in the Ethernet header? **14 Bytes**

Which Ethernet header fields are shown?
Source MAC, Destination MAC address and data type
1.Source MAC address: **TendaTec_69:1b: e0 (d8: 32: 14: 69: 1b: e0)**
2.Destination MAC address: **ASUSTekC_8b: d8: ad (30: 5a: 3a: 8b: d8: ad),**
3.Frame Type: **IPv4 (0x0800)**

Which Ethernet protocol field is not displayed? Explain, why this Ethernet protocol information is missing:

Answer: FCS is missing in Ethernet protocol information. Thus Ethernet trailer FCS is not displayed in Wireshark capture. **If there is no significant error captured or bad checksum detected while transferring frame, there is no FCS shown in Wireshark capture. Ethernet hardware, NIC strips it if the checksum is not correct.**

Why is the Ethernet preamble missing in this capture?

Answer: The Preamble contains 7 bytes and 1 byte of Start Frame Delimiter (SFD) and altogether do synchronization process between the sender and receiver. If a new frame is sent from source, the receiver gets sign to get ready to receive a new frame by the help of preamble and SFD**. If in Wireshark, there is no preamble shown, it means that both of them are working out of frame because they are not actually part of frame. Preamble doesn't carry any useful data and is not received like others. The 8 byte of preamble is also not stored in memory.**
After going down from data link layer to physical layer, preamble and SFD are added to frame.

d. Examine **IP packet** in the **1st ICMP Echo Reply** message.

- Which size (in Bytes) does the IP packet have? **60 Bytes**
-
- How many Bytes are in the IP header? **20 Bytes**

- Which protocol is signaled in the IP header?
  - Protocol field (hex value): **0×01** Protocol field (decimal value): **1**
  - Protocol name: **ICMP**

e. Examine **ICMP message** in the **1st ICMP Echo Reply** message.

Which hex number represents message type **Echo reply**? **Type: 0×00 and Code: 0×00**

- How many Bytes of ICMP has been sent?

  ICMP header: **8 Bytes** ICMP payload: **32 Bytes**

Task 2 – Analyze Latency in your LAN

Step 1: Latency in your LAN

a. Restart Wireshark, Ping your Default Gateway (max. 4 times) again, and stop capture.

b. Form Wireshark capture, record the average Round Trip Time (RTT) for one ICMP Echo Request / ICMP Echo Reply Response pair:

Answer: By using View > Time Display Format> Seconds since Beginning of capture (milliseconds)
RTT in my network is calculated as:
RTT1=0.004s, RTT2= 0.003s, RTT3=0.003s, RTT3=0.001s, RTT3=0.002s, RTT3=0.002s, RTT3=0.002s, RTT3=0.003s, RTT3=0.002s, RTT3=0.002s, RTT3=0.003s, RTT3=0.001s, RTT3=0.005s, RTT3=0.006s, RTT3=0.003s, RTT3=0.026s (also by using Cntrl+T)
Total RTT=68ms
Average RTT for one ICMP Echo Request / ICMP Echo Reply Response pair: 68/16=4.25ms

Step 2: Propagation delay

a. Estimate the length of the cable path from your PC to your Default Gateway:

b. **Answer**: In my home, there is WLAN (wireless network) activated that's why I can't measure any solid cable's length from my PC to my default gateway (router).
Therefore,
Estimated distance between my notebook and router is, l=10m

c. Calculate the corresponding propagation delay t_{pd}:
Answer: We know,
Propagation delay, **tpd= l / C** where l = distance between notebook and router=10m and
C= velocity of the microwave through wireless link =c= c0 = 300,000 km/s=3×10^8 m/s
Therefore, t_{pd}= l / C = 10m / (3×10^8) m/s = 3.33×10^{-8} s=33.3 ns

d. Record how many times an ICMP message is transmitted over this cable for one RTT: **two times**

e. Calculate the cumulative propagation delay $\sum t_{pd}$ for one ICMP Echo / ICMP Echo Reply pair:
2×33.3 ns=66.6 ns

Step 3: Transmission time

a. Record the data rate R of your local network:

Answer:
Data rate in my local area network is= 150 Mbps

b. From Wireshark capture, record the size of an Ethernet frame carrying the ICMP Echo Request message and do not forget to add the Ethernet FCS:
For ICMP ECHO Request, Frame size=74 Bytes and FCS= 4 Bytes
74+4 =78 Bytes

c. Calculate the transmission time t_t of this Ethernet frame:

Answer:
Transmission time, $t_t = M / R$, where M = Size of one Ethernet frame =78 bytes=78*8=624 bits and R= link bit rate=150 Mbps=1.5×10^8 bit per second
Therefore, $t_t = M / R = 624 \text{ bits} / 1.5\times10^8 \text{ bps} = 4.16\times10^{-6}$ s=4.16 µs

d. Record, how many NICs must transmit this Ethernet frame to reach your default gateway: **two NICs**

Admin State	State	Type	Interface Name
Enabled	Disconnected	Dedicated	**Ethernet 2**
Disabled	Disconnected	Dedicated	Ethernet
Enabled	Connected	Dedicated	**Wi-Fi**

e. Calculate the cumulative latency created by all transmission times $\sum t_t$ for one ICMP Echo / ICMP Echo Reply pair:
2×4.16 µs=8.32 µs

Step 4: Latency discussion

a. Which type of delay, transmission time t_t or propagation delay t_{pd}, has the highest influence on the ping round-trip-time (RTT) in this scenario?

Answer: **Propagation delay** has the highest influence on the ping round-trip-time (RTT) in this scenario. Propagation delay depends on the distance between source and destination.

b. Discuss other devices or SW processes which has influence on the RTT.

There are other factors those has influence on RTT. These are:
Other processes:
-processing delay (It depends on the distance between source and destination.)
-queuing delay
-encoding delay
Devices: Intermediate routers or servers
These delays depend on number of hops between sender and receiver.
Overall, distance and transmission media between server and host along with number of network hops, traffic in media, interference present in circuit, the speed of intermediary devices and server response time effects RTT.

Task 3 –Internet Connectivity

Part 1: Examine Web Access

Step 1: Request a Website

a. Start your preferred Browser, but do not request any URL.

b. Start Wireshark and capture without any filter and automatic scroll during live capture.

c. From command line delete DNS cache.
(Windows **ipconfig /flushdns** or Linux **sudo systemd-resolve --flush-caches**).

d. Switch to your Browser and request the Website http://www.nt.th-koeln.de/vogt/bs.html

e. Stop your Wireshark capture.

Step 2: Examine the Wireshark capture

a. Which information is asked for in your **DNS REQUEST**?
As DNS request, IP address of www.nt.th-koeln.de is asked.

b. Record the answer given by the DNS RESPONSE.

DNS first response to query www.nt.th-koeln.de:
1. www.nt.th-koeln.de: type CNAME, class IN, cname plesk-02-ext.cit-vip.fh-koeln.de
DNS second response to query www.nt.th-koeln.de:
2. www.nt.th-koeln.de: type A, class IN, addr 139.6.10.107
also for DNS query:
ilias.th-koeln.de: type A, class IN
DNS Response:
plesk-02-ext.cit-vip.fh-koeln.de: type A, class IN, addr 139.6.10.107 and etc.

- Record the IP address which is associated with www.nt.th-koeln.de:
139.6.10.107
- To which local network device has the DNS REQUEST been sent in your LAN?
Check the destination MAC address to solve this.
To Router (default gateway) which has 192.168.1.1 IP address. Router can work as both DNS server and DHCP to which DNS REQUEST comes.
Router's destination MAC address TendaTec_69:1b: e0 (d8: 32: 14: 69: 1b: e0).

c. Check the HTTP request message.

- Record the HTTP method sent in the **HTTP REQUEST**:
GET method (GET /vogt/bs.html HTTP/1.1)
- Record the destination IP address: **139.6.10.107**
- Record the remote TCP Port: port **80**
- Record the local TCP Port: port **57499** (while sending GET request and having response)
- Record the local network device to which the HTTP REQUEST has been sent. Check the destination MAC address to solve this.
The router's (default gateway) destination MAC address TendaTec_69:1b: e0 (d8:14: 69: 1b: e0) is observed. Through this local area device- router, HTTP GET REQUEST was sent.

Part 2: Network Route to a Website

Step 1: Examine ping to a Website

a. Start a new Wireshark capture without saving the previous data. From command line issue **ping -4 www.cisco.com (**Windows**)** or **ping www.cisco.com (**Linux**)**

Important note: Use the "-4" option of the ping command to exclude IPv6 addresses in this step. Finally stop the Wireshark capture.

Examine the ICMP request-response pairs. Is the ping successful?
Yes

- Which IP time-to-live (TTL) value is received in the ICMP ECHO REPLY message?
  TTL=56

Note: When an IP packet is sent, the source sets the TTL value in each IP packet. In WinOS TTL usually starts with 128, in UNIX/Linux it starts with 64. With each router hop the TTL is decremented by 1.

- How many router hops may be passed on the return path from cisco.com?
  As CISCO uses Linux operating system also. For this OS, we know, TTL=64
  64-56=8 hops maybe passed from cisco.com on return path.
- Your OS assumption: **Linux/UNIX in cisco server.**

Step 2: Examine the network path to a Website

a. Start a new Wireshark capture without saving the previous data.

b. From command line issue **tracert -4 www.cisco.com** and finally stop the Wireshark capture.

- How many hops do you get by traceroute?
  9 hops (**from my notebook to cisco server**).

  Compare this result with a). The ICMP TTL exceeded is generated by which OS?

  Answer:
  (Tracing route to e2867.dsca.akamaiedge.net [104.122.16.62]
  over a maximum of 30 hops:)
  Comparing with a), CDN of Akamai (initiated sending from cisco.com) sends TTL exceeded replies using 6 hops through to me from other IP addresses.

c. Examine the ICMP request-response pairs.

- In the 1st ICMP ECHO REQUEST, which TTL has been set? **TTL= 1**
  Which ICMP response message has been received? **(TIME-TO-LIVE EXCEEDED) Time to live exceeded in transit**
- From which IP address?
  Router's IP address 192.168.1.1

  How many times was this test repeated with the same TTL?
  Every same TTL was repeated 3 times

d. Look for the ICMP ECHO REQUEST with TTL+1 value (**In Win OS often the 5th ICMP request**).

- Which TTL has been set now? - **TTL=2 was seen in ICMP 4TH ECHO REQUEST**
- Which ICMP response message has been received to this? - **TIME-TO-LIVE EXCEEDED**
- From which IP addresses? - **103.127.177.45**

e. Continue the evaluation of changing TTL values in ICMP requests

- For how many different TTL values do you get ICMP TTL EXCEEDED?

  Answer: There is TTL=9 in whole for ICMP ECHO REQUESTS. But after 9th TTL, the echo reply was destination unreachable instead of TTL EXCEEDED. And upon 8th TTL, the reply was always as TTL EXCEEDED**. In short, up to TTL=1 to TTL=8 values I get ICMP TTL EXCEEDED.**

- By which other ICMP response than ICMP TTL EXCEEDED does traceroute stop the search of the path? Check the last response to the traceroute requests.

  Answer: Other than ICMP EXCEEDED**,** the last responses using the tracert command are from **ICMP ECHO REPLY**.

- Describe the mechanism which is used by traceroute to find the path from source to destination.
  Answer:
  Traceroute traces number of intermediary devices on the media from source to destination. Firstly, traceroute sends UDP packet to the destination with TTL=1. TTL means Time-to-Live which limits the life time of data packet in network. It helps to count hops. However, when first router receives first UDP packet with TTL=1, it reduces it by 1, like TTL=1-1=0 and drops the packet and then send a ICMP error message as Time Exceeded to source/sender. Through this, traceroute calculates round trip time also. After getting the error message, sender sends two more packets to the 2nd router and again 2nd router makes TTL=1-1=0 sending same ICMP error messages. Like this, this process happens for three times before source sends next packet by incrementing TTL by 1 that is TTL=2. Until the UDP packet reaches its final destination, in this same way, traceroute keeps tracking average RTT and the IP addresses of routers and other devices with names and upon reaching the destination, ICMP error message: Time exceeded is not sent anymore to source. Lastly, ICMP error message Destination unreachable is sent to source to denote that UDP packet has been reached to destination. In Wireshark, ICMP ECHO packet is sent instead of UDP packet.

Step 3: Decision on Webservers OS

a. Comparing the ping test and the traceroute test, which OS is run on the Webserver?
Answer:
Ping test:
After pinging, TTL=56 is observed.
Traceroute test:
After trace routing, I have got 9 hops.
If I add this two outputs, I can assume the OS which runs cisco server.
56+9=65
Now, for measuring same hop/router twice, 9-1=8 hops will be taken.
56+8=64
TTL=64 stands for Linux kernel. Also 64 stands for some other OS too.

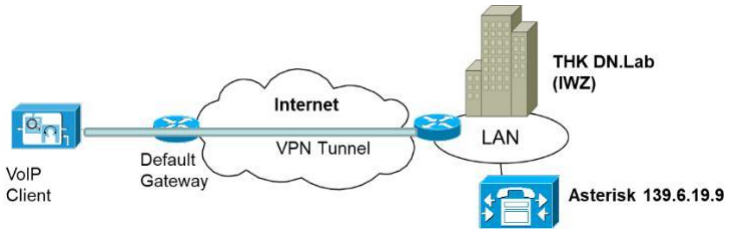
Task 4 – Examine VoIP Streaming

Background / Scenario

VoIP is an IP streaming technology of private Enterprise Networks, public Next Generation Networks (NGN) and public Next Generation Mobile Networks (NGMN). It uses SIP/SDP protocol for signaling, and RTP encapsulation of audio samples.

In this lab you will setup your own VoIP client and will test VoIP sessions and throughput with the DN.Lab Asterisk VoIP server. For this lab, IP VPN connectivity to TH Köln, network 139.6.0.0 /16, is required.

Note: The DN.Lab VoIP server has very limited connectivity and cannot be used for any public calling.



Part 1: Connectivity to DN.Lab

Step 1: Setup VPN connection

a. To get connectivity to the DN.Lab VoIP server you must implement a VPN connection to TH Köln. English and German instructions are found here: https://www.th-koeln.de/hochschule/vpn---virtual-private-network_26952.php.

Step 2: Setup VoIP client

a. There are many VoIP clients available for different OS platforms, and you can work with your preferred one. Some SIP clients are:

WinOS	MacOS	Linux
Linphone (Audio + Video)	Linphone (Audio + Video)	Linphone (Audio + Video)
PhonerLite (Audio)	Telephone	Zoiper
Bria	X-Lite	
Phoner	Bria	
NinjaLite	Zoiper	
X-Lite		
Zoiper		

The VoIP client requirements for this lab are

- Voice codecs: PCM G.711 µ-law, GSM full rate, Speex 8 kHz sampling and Speex 16 kHz sampling
- Video codec: H.264 (VGA 640x480 resolution)

b. Install the VoIP client of your choice. Register user 6001 with the DN.Lab Asterisk server. You must have a VPN connection to THK, to get access to the Asterisk server. The following instructions were tested with the free and open **Linphone** client.

- Available SIP accounts:
  SIP User: sip:6001@139.6.19.9 (password 6001)
  Voice response: sip:6999@139.6.19.9
  Echotest: sip:6998@139.6.19.9
- SIP Domain 139.6.19.9
- Transport UDP
- SIP Port 5060
- RTP Ports allowed: from 7078-32767

Step 3: Test VoIP connectivity

a. Start Wireshark capture and ensure you select the internal interface without to read unencrypted SIP and RTP messages.

b. Call extension 6999 at the DN.Lab VoIP server. You should receive the "hello world" announcement and some "monkeys".

c. Stop the Wireshark capture when the call is ended.

d. Record the Wireshark filter options to filter one IP flow, which transmits the audio stream from the DN.Lab VoIP server to your VoIP client: **RTP mainly**

e. Apply this filter options in Wireshark and record the following:

- DN.Lab server socket: **139.6.19.9: 10166**
- VoIP client socket: **139.6.215.176: 7078**
- RTP payload type: **ITU-T G.711 PCMU (0)**

Part 2: Examine VoIP service

Step 1: Test RTT to DN.Lab VoIP Server

a. Close Wireshark application and **disconnect** from VPN

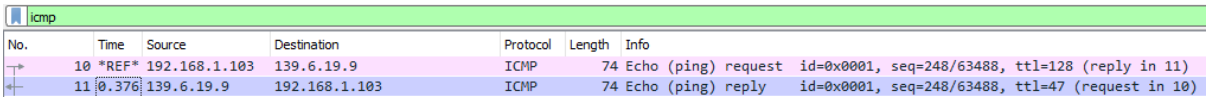
b. Estimate **propagation delay t_{pd}**

- Estimate the distance of your location to our lab (address: Betzdorfer Straße 2, D-50679 Cologne). You may use OpenStreetsMap (www.openstreetmap.org) to find a route.
  Answer: After using this map, distance is found as **9982 km** between my location and addressed location.

- Calculate the propagation delay t_{pd} for that route:
  We know,
  Propagation delay, $t_{pd} = l / C$
  where l =distance between two locations =9982 km and
  C= velocity of the signal through fiber optic cabling =$2/3\ c_0 = 2/3 \times 300.000$ km per second=200,000 km/s=2×10^8 m/s
  Therefore, $t_{pd} = l / C = 9982\ km / (2\times10^8)\ m/s = 0.0499\ s = 49.91ms = 0.05\ s =$ **50ms**

c. Restart Wireshark and ping DN.Lab VoIP server and record RTT from Wireshark capture.

- When you filter "ICMP" and select "View -> Time display format -> Seconds since previous displayed packet" you can easily read the **RTT: 0.376s= 376ms (first RTT)**
- **using Cntrl+T**



- Record the **size** of the Ethernet frame carrying the ICMP Packet (add the Ethernet FCS):
  Total size of Ethernet frame carrying ICMP Packet=74 Bytes.
  Ethernet Checksum: 4 Bytes
  Total ICMP frame=74+4=78 Bytes

d. Traceroute to DN.Lab VoIP server and record number of hops to DN.Lab VoIP server

- Number of hops: **30 hops (seems like there are more hops than 30)**

e. Calculate the transmission t_t time of a ping.
For simplicity we assume R=1000Mbps and take the frame size of f).

VoIP client side: (from my pc)
Transmission time, $t_t = M/R$
where M = Size of the frame =78 bytes=78*8=624 bits and (from c)
R= Bit rate=150 Mbps=1.5 $\times 10^8$ bit per second (recorded)
Therefore, $t_t = M / R = 624$ bits $/ 1.5 \times 10^8$ bps = 4.16×10^{-6} s=4.16 µs

VoIP server side: (from lab)
Transmission time, $t_t = M/R$
where M = Size of the frame =78 bytes=78*8=624 bits and (from c)
R= Bit rate=1000 Mbps= 10^9 bit per second
Therefore, $t_t = M / R = 624$ bits $/ 10^9$ bps = 6.24×10^{-6} s=6.24 µs

f. Add the propagation delay and all transmission times at each hop to and from the DN.Lab VoIP server:

- **Transmission time (to and from) $\sum t_t$ (all hops): 4.16 µs × 30 + 6.24 µs × 17= 230.88 µs**
  64-47=17 hops are used from DN.Lab to VoIP client.
  TTL=30 from VoIP client to the DN.Lab VoIP server as captured
  TTL=47 from the DN.Lab VoIP server to VoIP client as captured

- **Propagation delay (to and from) $\sum t_{pd}$ (all links):**
  333ms + 50ms+ 50ms= 433ms
  2×433ms=866ms (to and from)

  Description:

  Firstly, through Wireless link from VoIP client to default gateway:

  Propagation delay, $t_{pd} = l / C$
  where l =distance between two locations =9982 km and
  C= velocity of the signal through fiber optic cabling = c_0 = 300.000 km per second=300,000 km/s=3×10^8 m/s
  Therefore, $t_{pd} = l / C = 9982$ km $/ (3 \times 10^8)$ m/s = 0.03327 s = 0.333s= 333ms

  Secondly, through Fiber optic link through Bangladesh to Koeln:

  Propagation delay, $t_{pd} = l / C = 50$ms (calculated in part2_b)

  Thirdly, through twisted pair cabling inside of DN.lab:

  Propagation delay, $t_{pd} = l / C = 50$ms (same as fiber optic cabling)

- **Network latency (to and from): $\sum t_t + \sum t_{pd}$: 230.88 µs (0.23088ms) + 866ms = 866.23088ms**
- Calculate the share (in %) of network latency in the total RTT:

  RTT1=376ms
  RTT2=371ms
  RTT3=373ms
  RTT4=359ms (from Wireshark)
  Total RTT=1479ms

Network latency in the total RTT:

Network latency ($\sum t_t + \sum t_{pd}$) / Total RTT
=866.23088ms / 1479ms=0.5856×100=58.56%

- Name devices and processes, which add latency to the RTT, not given by t_t or t_{pd} :
  Answer:
  Devices: **Intermediate routers or servers**
  Other processes:
  -Propagation delay- It depends on the distance between source and destination.
  -Processing delay
  -Queuing delay
  -Encoding delay
  These delays depend on number of hops between sender and receiver.
  Overall, distance and transmission media between server and host along with number of network hops, traffic in media, interference present in circuit, the speed of intermediary devices and server response time effects RTT.

Step 2: VoIP Echo test

a. Connect to VPN again. Close and restart Wireshark capture.

b. Call the echo test extension 6998 at the DN.Lab VoIP server and perform some echotest.

- Estimate the perceived echo delay:
  Assumption from capture:
  Echo delay=original sound+ time span before first echo sound
  =2ms (using seconds since beginning of capture time format and further calculation)
  + [Time since previous frame: 0.021015000 seconds (first echo response)]
  =2ms+21.015ms
  =23.015ms

- Filter RTP of this echotest. Can you read the echo delay from Wireshark capture?
  From delta time in timestamps area, we can get echo delay.

Step 3: VoIP Throughput

a. For this test, you call the extension 6002 from your VoIP client and capture all packets sent and received by your VoIP client. It will ring up to 30 seconds before it will hang up. While it is ringing, RTP flows are already running.

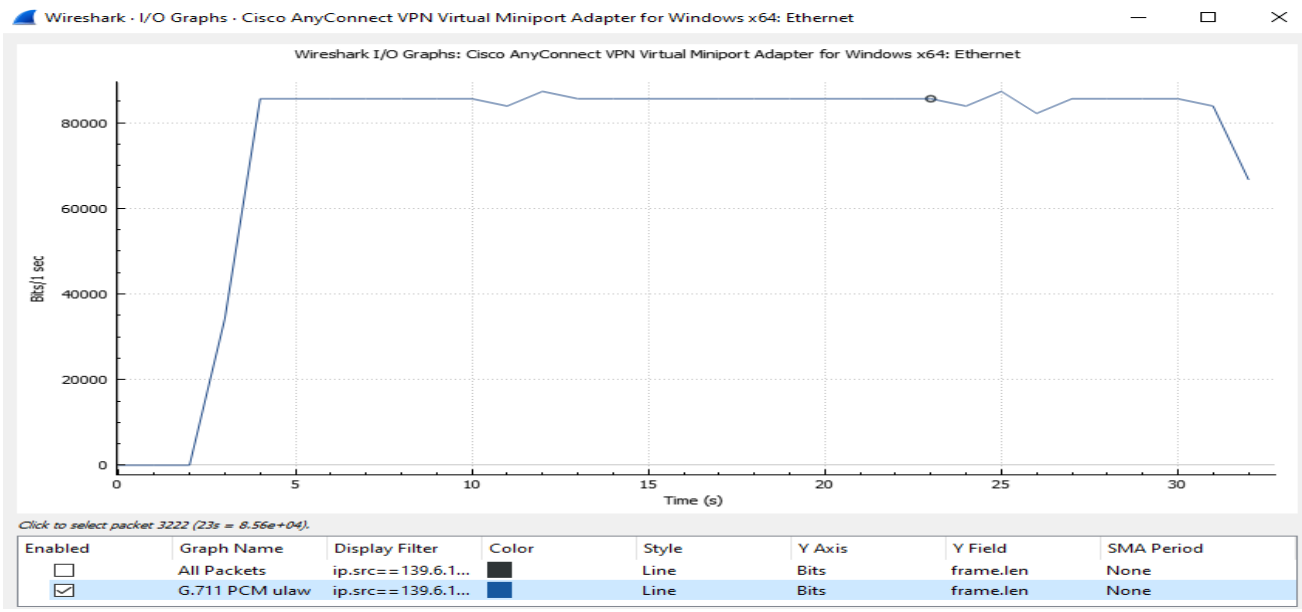
b. For each of the following codecs you will

- Configure the client only with the required audio codec, switch-off other audio codecs. Start a Wireshark capture and call to extension 6002. Stop the capture when the call ended.
- Only for the G.711 codec, save the Wireshark capture of this test in .**pcapng** format.
- Filter the IP flow of the required codec from your VoIP client to the destination.
- Display the IP flow throughput with “I/O graph” in Bit per Second (!) and record the throughput of the selected codec in kbps.

c. Record the codec throughput

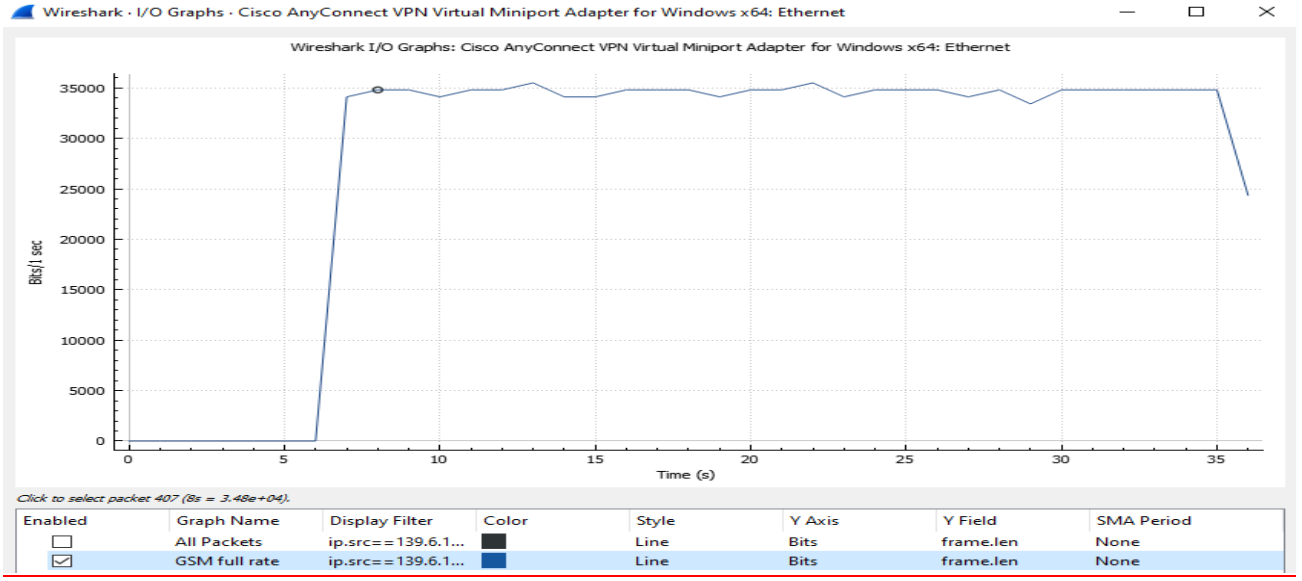
G.711 PCM ulaw:

Throughput recorded = 8.56×10^4 bit/sec = 85600 bps= 85.6 kbps



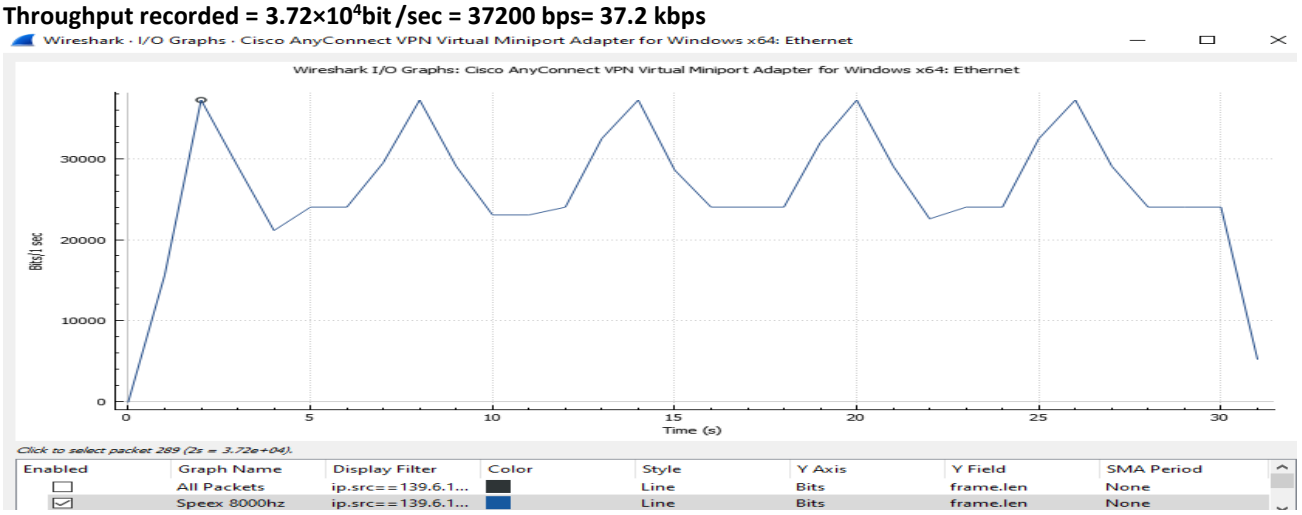
Audio GSM Full Rate:

Throughput recorded = 3.48×10^4bit /sec = 34800 bps= 34.8 kbps



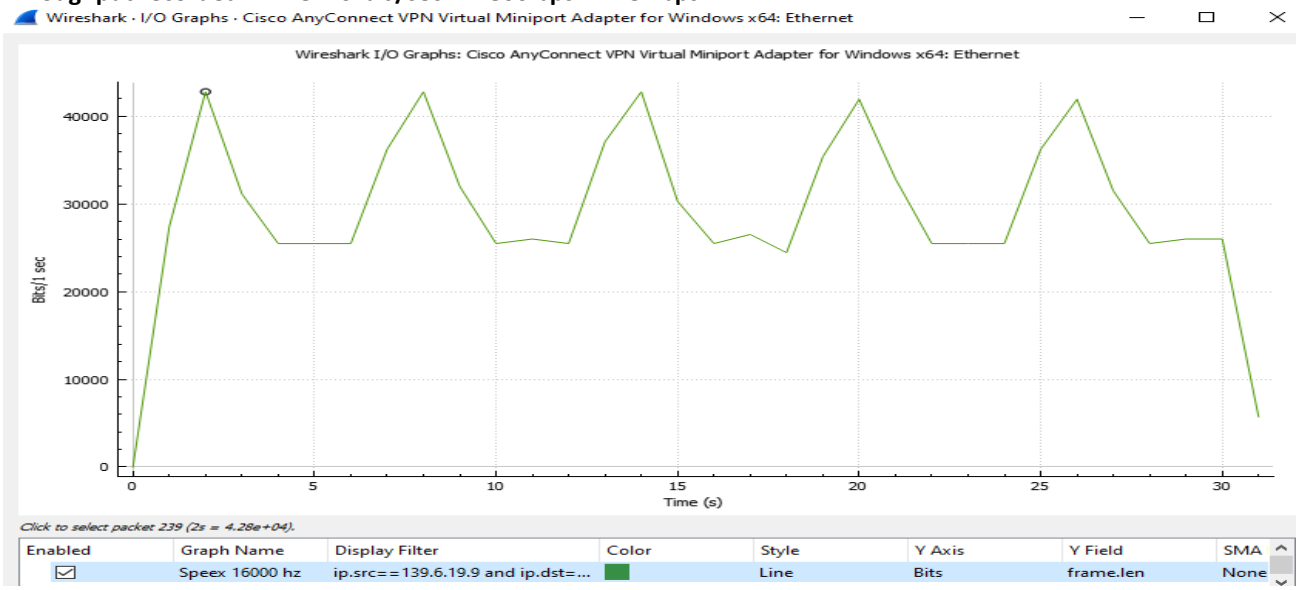
Audio Speex 8000 Hz:

Throughput recorded = 3.72×10^4bit /sec = 37200 bps= 37.2 kbps



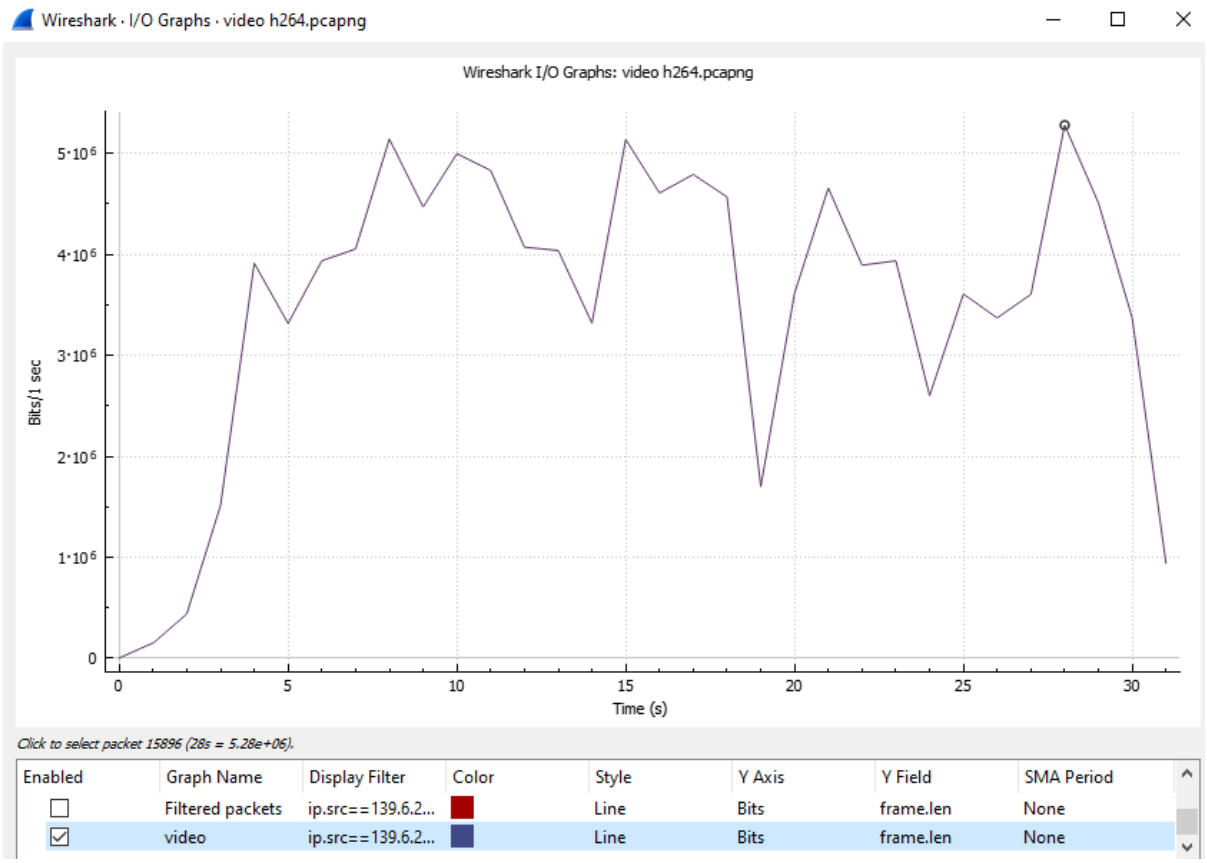
Audio Speex 16000 Hz:

Throughput recorded = 4.28×10^4 bit /sec= 42800 bps= 42.8 kbps



Video H.264 640*480):

Throughput recorded = 5.28×10^6 bit /sec = 5.28 mbps (using Speex 16khz parallel with video H.264 as only H.264 codec does not work)



Codec	Ethernet level throughput in kbps
Audio G.711 PCM µ-law	85.6 kbps
Audio GSM Full Rate	34.8 kbps
Audio Speex 8000 Hz	37.2 kbps
Audio Speex 16000 Hz	42.8 kbps
Video H.264 (640x480)	5.28 mbps

d. Review Speex audio codec. Explain why the throughput does not double from 8 kHz sampling to 16 kHz sampling although we generate twice samples per second.
Throughput is not doubled while doubling from 8 kHz sampling to 16 kHz sampling. Because sampling is doubled in application layer. And on layer 2, other fields, like the RTP header, TCP and UDP header, IP header, and Ethernet header do not get doubled. Although packets have now larger samples as of twice samples per second, but the headers are always the same on data link layer. So on layer 2, headers don't get doubled while transmitting. And that is why in Speex 1600 KHz codec, throughput is not doubled from Speex 8 KHz codec.

Checkout

When you successfully finished this Lab, save your results and answers in this PDF file.

Upload your results and the capture file (.pcapng) of Tast4 / Part2 / Step3 in Ilias Lab Solutions.