Network traffic sniffing: Wireshark tool learning

General description

In this task, students will gain hands-on experience with network traffic sniffing using Wireshark, a powerful network protocol analyzer. The objective of this task is to understand how network communication works at a low level and to analyze the traffic exchanged between clients and a server in a simulated network environment.

Task

1. Setup:
   - Ensure that Wireshark is installed on your computer. You can download Wireshark from the official website: https://www.wireshark.org/
   - Set up a network environment with at least one client and one server. The clients and server use TCP for communication.
   - Make sure that the client and server applications are running and communicating with each other over TCP.
2. Capturing TCP Traffic:
   - Open Wireshark on your computer.
   - Select the network interface through which the TCP communication between the client and server is happening. This could be Ethernet, Wi-Fi, or any other network interface. **!IMPORTANT! If you're executing client and server on the same machine, please choose Adapter for loopback traffic capture.**
   - Start capturing TCP traffic by filtering for TCP packets and/or filtering server's port number in Wireshark.
3. Task:
   - For both assignments 2 and 3 you should analyze the network traffic.
   - Results must be presented in a separate doc file where you should explain all communication steps.
   - Identify and analyze different types of TCP traffic, such as the TCP handshake, TCP data segments, TCP acknowledgments, and TCP connection termination.
   - Observe how the client and server exchange data during various stages of TCP communication, such as connection establishment, data transfer, and connection termination.
   - Note any abnormalities or unexpected behavior in the TCP traffic, such as retransmissions, out-of-order packets, or TCP protocol errors. In general, there is a low chance of such issues on your computers that's why here we have an **extra task**: find the maximum size of TCP packet by changing your sources.
   - Document your findings and observations in a report, highlighting key insights and lessons learned from the TCP traffic analysis.
4. Conclusion:
   - Conclude the task by summarizing your observations and insights gained from analyzing the TCP traffic.
   - Reflect on the importance of TCP traffic sniffing in understanding and troubleshooting TCP-based network issues.
   - Discuss any challenges faced during the task and how they were overcome.

Theory

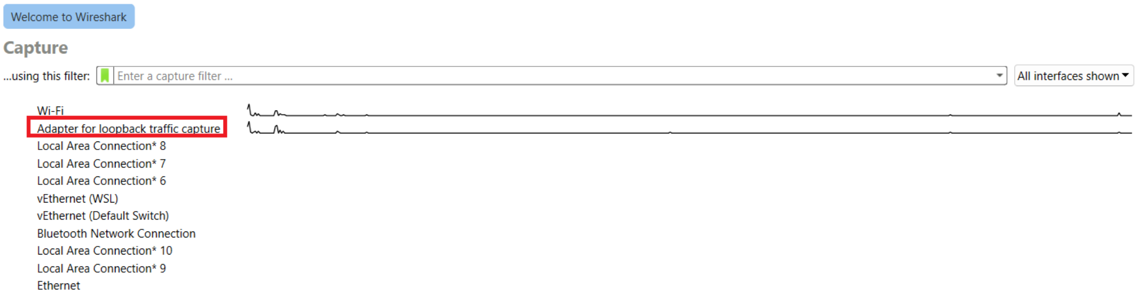
1. **TCP (Transmission Control Protocol)**:
   - TCP is a reliable, connection-oriented protocol used for transmitting data over networks.
   - It provides mechanisms for establishing and terminating connections, as well as for ensuring the reliable delivery of data packets.
2. **TCP Handshake**:
   - The TCP handshake is a three-way process used to establish a connection between a client and a server.
   - It involves the exchange of SYN (synchronize) and ACK (acknowledge) packets between the client and server to agree on initial sequence numbers and establish communication parameters.
3. **TCP Segments**:
   - TCP data is transmitted in segments, each consisting of a header and optional payload (data).
   - The TCP header contains control information such as source and destination port numbers, sequence numbers, acknowledgment numbers, and flags.
4. **TCP Flags**:
   - TCP uses control bits, or flags, in the TCP header to control various aspects of communication.
   - Common TCP flags include SYN (synchronize), ACK (acknowledge), FIN (finish), RST (reset), and PSH (push).
5. **Wireshark**:
   - Wireshark is a popular open-source network protocol analyzer used for analyzing network traffic in real-time.
   - It captures packets traversing a network interface and provides detailed information about each packet, including headers, payloads, and protocol information.
6. **Packet Sniffing**:
   - Packet sniffing refers to the process of capturing and analyzing network packets as they travel across a network.
   - Packet sniffers like Wireshark can intercept and analyze packets to inspect network traffic, troubleshoot network issues, and identify security threats.
7. **OSI Model**:
   - The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstraction layers.
   - Each layer represents a specific set of functions and protocols, facilitating communication between devices over a network.
8. **TCP/IP Model**:
   - The TCP/IP model is a more simplified and widely used networking model, which combines the functionalities of the OSI model into four layers: Application, Transport, Internet, and Link.
   - TCP and IP are the primary protocols used in the TCP/IP model for communication between devices on a network.
9. **Port Numbers**:
   - Port numbers are used to identify specific endpoints (applications) within a host in TCP communication.

- Well-known port numbers (0 to 1023) are reserved for common services such as HTTP (port 80) and HTTPS (port 443), while registered and dynamic ports (1024 to 65535) are used for other applications.

10. **TCP Retransmission**:
    - TCP retransmission occurs when a data segment is not acknowledged by the receiver within a specified timeout period.
    - The sender retransmits the unacknowledged segment to ensure reliable delivery, helping to recover from packet loss or network congestion.

Examples

1. How to choose the correct interface:



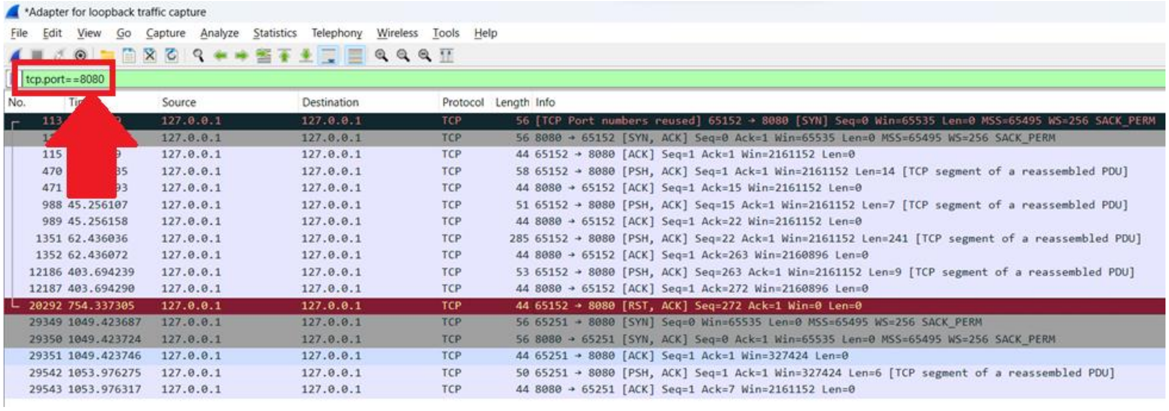
Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ... All interfaces shown

```
Wi-Fi
Adapter for loopback traffic capture
Local Area Connection* 8
Local Area Connection* 7
Local Area Connection* 6
vEthernet (WSL)
vEthernet (Default Switch)
Bluetooth Network Connection
Local Area Connection* 10
Local Area Connection* 9
Ethernet
```

2. How to filter the traffic when the server is working on port 8080:

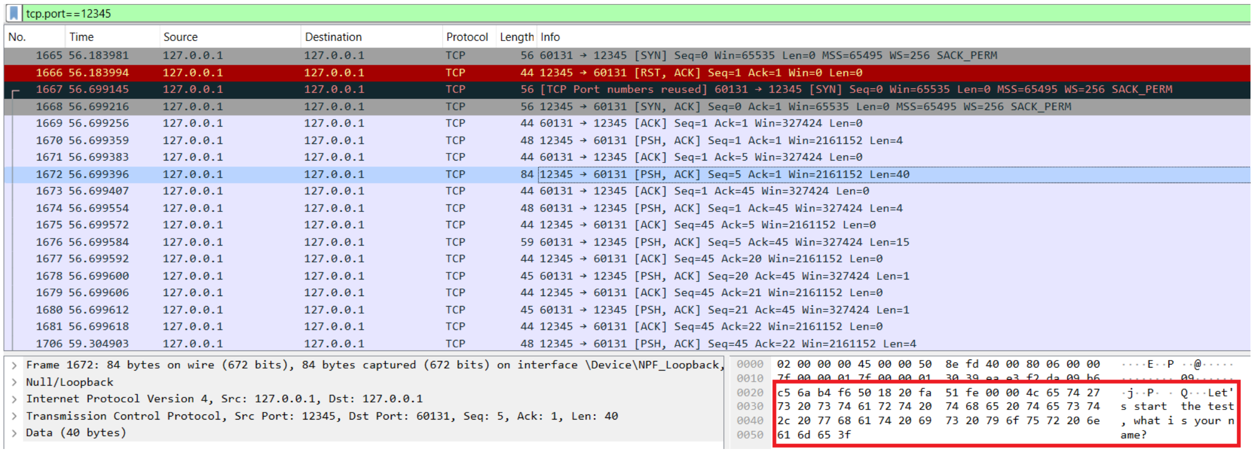


*Adapter for loopback traffic capture

tcp.port==8080

```
No.    Time            Source      Destination  Protocol  Length  Info
  113                  127.0.0.1   127.0.0.1    TCP       56 [TCP Port numbers reused] 65152 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
  1                    127.0.0.1   127.0.0.1    TCP       56 8080 → 65152 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
  115          9       127.0.0.1   127.0.0.1    TCP       44 65152 → 8080 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
  470          35      127.0.0.1   127.0.0.1    TCP       58 65152 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=14 [TCP segment of a reassembled PDU]
  471          93      127.0.0.1   127.0.0.1    TCP       44 8080 → 65152 [ACK] Seq=1 Ack=15 Win=2161152 Len=0
  988 45.256107        127.0.0.1   127.0.0.1    TCP       51 65152 → 8080 [PSH, ACK] Seq=15 Ack=1 Win=2161152 Len=7 [TCP segment of a reassembled PDU]
  989 45.256158        127.0.0.1   127.0.0.1    TCP       44 8080 → 65152 [ACK] Seq=1 Ack=22 Win=2161152 Len=0
 1351 62.436036        127.0.0.1   127.0.0.1    TCP      285 65152 → 8080 [PSH, ACK] Seq=22 Ack=1 Win=2161152 Len=241 [TCP segment of a reassembled PDU]
 1352 62.436072        127.0.0.1   127.0.0.1    TCP       44 8080 → 65152 [ACK] Seq=1 Ack=263 Win=2160896 Len=0
12186 403.694239       127.0.0.1   127.0.0.1    TCP       53 65152 → 8080 [PSH, ACK] Seq=263 Ack=1 Win=2161152 Len=9 [TCP segment of a reassembled PDU]
12187 403.694290       127.0.0.1   127.0.0.1    TCP       44 8080 → 65152 [ACK] Seq=1 Ack=272 Win=2160896 Len=0
20292 754.337305       127.0.0.1   127.0.0.1    TCP       44 65152 → 8080 [RST, ACK] Seq=272 Ack=1 Win=0 Len=0
29349 1049.423687      127.0.0.1   127.0.0.1    TCP       56 65251 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
29350 1049.423724      127.0.0.1   127.0.0.1    TCP       56 8080 → 65251 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
29351 1049.423746      127.0.0.1   127.0.0.1    TCP       44 65251 → 8080 [ACK] Seq=1 Ack=1 Win=327424 Len=0
29542 1053.976275      127.0.0.1   127.0.0.1    TCP       50 65251 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=6 [TCP segment of a reassembled PDU]
29543 1053.976317      127.0.0.1   127.0.0.1    TCP       44 8080 → 65251 [ACK] Seq=1 Ack=7 Win=2161152 Len=0
```

3. How to check the data that is transferred over TCP protocol (application protocol details)?



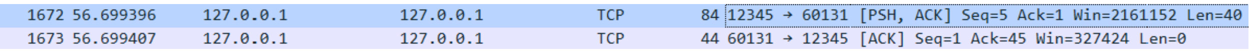
tcp.port==12345

```
No.   Time       Source      Destination  Protocol  Length  Info
1665 56.183981   127.0.0.1   127.0.0.1    TCP       56 60131 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
1666 56.183994   127.0.0.1   127.0.0.1    TCP       44 12345 → 60131 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1667 56.699145   127.0.0.1   127.0.0.1    TCP       56 [TCP Port numbers reused] 60131 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
1668 56.699216   127.0.0.1   127.0.0.1    TCP       56 12345 → 60131 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
1669 56.699256   127.0.0.1   127.0.0.1    TCP       44 60131 → 12345 [ACK] Seq=1 Ack=1 Win=327424 Len=0
1670 56.699359   127.0.0.1   127.0.0.1    TCP       48 12345 → 60131 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=4
1671 56.699383   127.0.0.1   127.0.0.1    TCP       44 60131 → 12345 [ACK] Seq=1 Ack=5 Win=327424 Len=0
1672 56.699396   127.0.0.1   127.0.0.1    TCP       84 12345 → 60131 [PSH, ACK] Seq=5 Ack=1 Win=2161152 Len=40
1673 56.699407   127.0.0.1   127.0.0.1    TCP       44 60131 → 12345 [ACK] Seq=1 Ack=45 Win=327424 Len=0
1674 56.699554   127.0.0.1   127.0.0.1    TCP       48 60131 → 12345 [PSH, ACK] Seq=1 Ack=45 Win=327424 Len=4
1675 56.699572   127.0.0.1   127.0.0.1    TCP       44 12345 → 60131 [ACK] Seq=45 Ack=5 Win=2161152 Len=0
1676 56.699584   127.0.0.1   127.0.0.1    TCP       59 60131 → 12345 [PSH, ACK] Seq=5 Ack=45 Win=327424 Len=15
1677 56.699592   127.0.0.1   127.0.0.1    TCP       44 12345 → 60131 [ACK] Seq=45 Ack=20 Win=2161152 Len=0
1678 56.699600   127.0.0.1   127.0.0.1    TCP       45 60131 → 12345 [PSH, ACK] Seq=20 Ack=45 Win=327424 Len=1
1679 56.699606   127.0.0.1   127.0.0.1    TCP       44 12345 → 60131 [ACK] Seq=45 Ack=21 Win=2161152 Len=0
1680 56.699612   127.0.0.1   127.0.0.1    TCP       45 60131 → 12345 [PSH, ACK] Seq=21 Ack=45 Win=327424 Len=1
1681 56.699618   127.0.0.1   127.0.0.1    TCP       44 12345 → 60131 [ACK] Seq=45 Ack=22 Win=2161152 Len=0
1706 59.304903   127.0.0.1   127.0.0.1    TCP       48 12345 → 60131 [PSH, ACK] Seq=45 Ack=22 Win=2161152 Len=4
```

```
> Frame 1672: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface \Device\NPF_Loopback,
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 12345, Dst Port: 60131, Seq: 5, Ack: 1, Len: 40
> Data (40 bytes)
```

```
0000  02 00 00 00 45 00 00 50  8e fd 40 00 80 06 00 00   ····E··P ··@·····
0010  7f 00 00 01 7f 00 00 01  30 39 ea e3 f2 da 09 b6   ········ 09······
0020  c5 6a b4 f6 50 18 20 fa  51 fe 00 00 4c 65 74 27   ·j··P· · Q···Let'
0030  73 20 73 74 61 72 74 20  74 68 65 20 74 65 73 74   s start  the test
0040  2c 20 77 68 61 74 20 69  73 20 79 6f 75 72 20 6e   , what i s your n
0050  61 6d 65 3f                                        ame?
```

4. [**Push**, **Ack**nowledge] -> [**Ack**nowledge] sequence:



```
1672 56.699396    127.0.0.1    127.0.0.1    TCP    84 12345 → 60131 [PSH, ACK] Seq=5 Ack=1 Win=2161152 Len=40
1673 56.699407    127.0.0.1    127.0.0.1    TCP    44 60131 → 12345 [ACK] Seq=1 Ack=45 Win=327424 Len=0
```

Evaluation

Detailed traffic analysis for the assignment 4, in testing session you should include 2 clients connection and transferring of 1 large file	7
Analyze one [PSH, ACK]->[ACK] sequence in details (explain each part of the message)	2
Find the maximum TCP packet size on your system either by changing the code or by finding it in active session	1
Total	10

For 1 communication session you should be able to explain what do we see in the Wireshark, otherwise the total grade will be reduced.

Questions

1. Explain the TCP handshake process in detail, including the purpose of each step and the role of sequence numbers and acknowledgment numbers.
2. Which information seen in the Wireshark relates to the TCP protocol?
3. Which information seen in the Wireshark relates to the Application protocol?
4. Describe the differences between TCP and UDP protocols. Discuss the advantages and disadvantages of using TCP for reliable data transmission compared to UDP.
5. How does Wireshark capture network traffic, and what are the different methods available for filtering and analyzing captured packets? Provide examples of Wireshark filters commonly used to analyze TCP traffic.
6. Describe the purpose and functionality of TCP flags (control bits) such as SYN, ACK, FIN, and RST. How are these flags used to manage TCP connections and control the flow of data?
7. Discuss the role of sequence numbers and acknowledgment numbers in TCP data transmission. How do these numbers ensure reliable and ordered delivery of data segments?
8. What is the OSI model, and how does it relate to TCP/IP? Describe each layer of the OSI model and explain how TCP/IP protocols operate within these layers.

Links

1. Wireshark official documentation: https://www.wireshark.org/docs/
2. Internetworking With TCP/IP by Douglas E. Comer : https://www.homeworkforyou.com/static_media/uploadedfiles/Douglas%20E.%20Comer%20-%20Internetworking%20with%20TCP_IP%20Volume%20One.%201-Addison-Wesley%20(2013).pdf

3. Computer Systems: A Programming Perspective: https://github.com/iWangMu/Book-CSAPP/blob/master/_Attachments/Computer_Systems_A_Programmers_Perspective(3rd).pdf Chapters 11-12