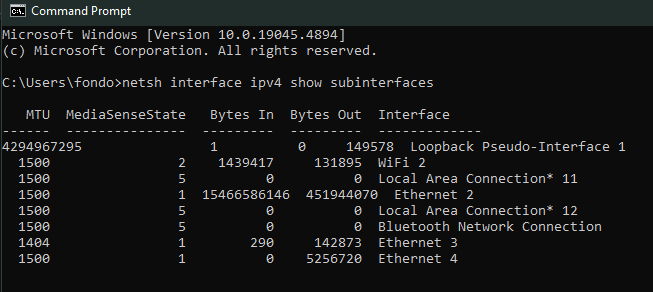
1. How can you determine the MTU of your network card, and what steps will you follow to accomplish this?

- Finding out the Maximum Transmission Unit (MTU) of a network card can help optimize network performance and troubleshoot issues.

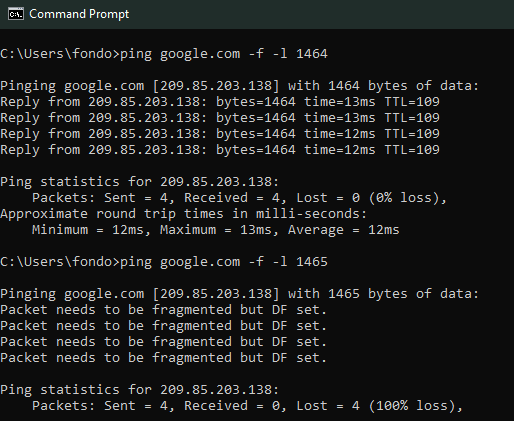
-Command: “netsh interface ipv4 show subinterfaces”

This command displays a list of network interfaces as well as their MTU values.



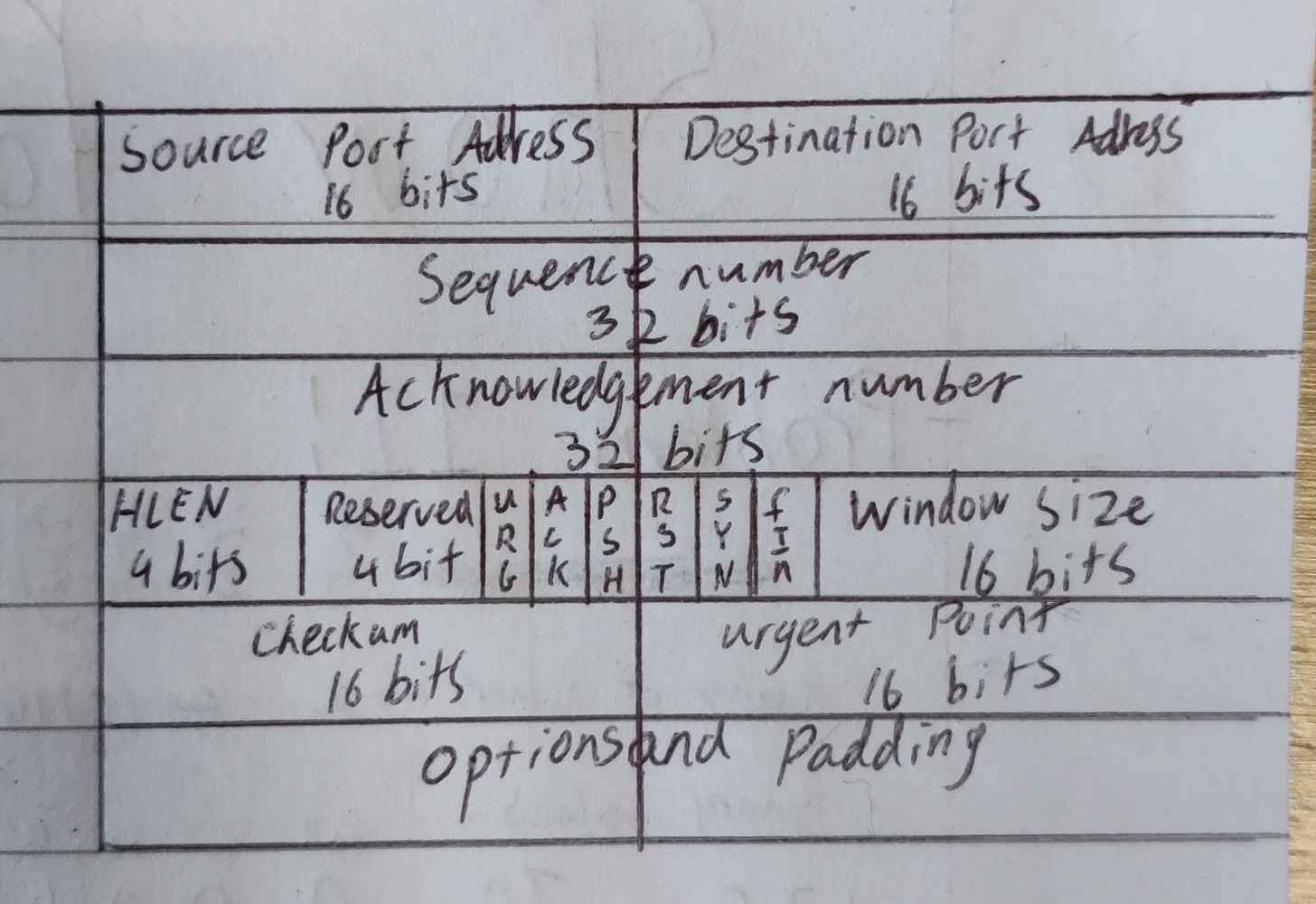
Here the MTU value for my Ethernet 2 which I use exclusively on this PC is 1500.

However, when I go to ping, my outgoing packet size maxes out at 1464 before needing to be fragmented, shown here:

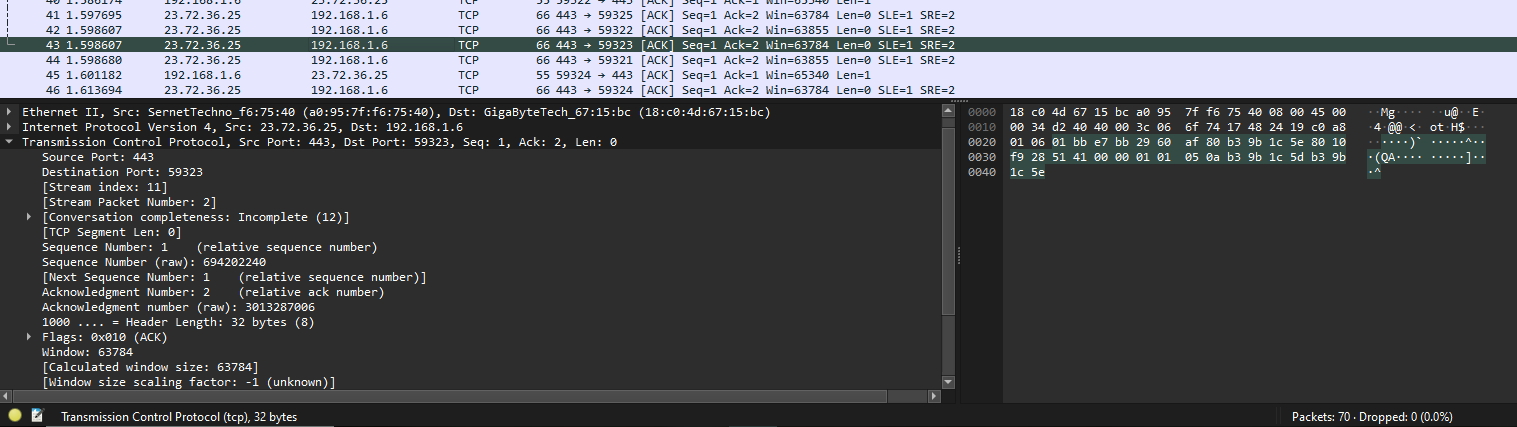


The remaining size is used for the overhead and that is why it cannot be utilized in the same way.

2. Draw a TCP header. Capture packets using Wireshark and explain the fields for a particular TCP packet captured. Try to explain the purpose of each field.



Here is my chosen captured packet which is using TCP (Transmission Control Protocol):



### **Source Port**

* The source port represents the port number on the sender's side. In this case, it is port 443.

### **Destination Port**

* The destination port is the port number on the recipient's side. This port (59323) is dynamically assigned, often used by a client in communication with a server.

### **Stream Index**

* This number tracks the stream of packets exchanged between the source and destination in the same session.

### **Stream Packet Number: 2**

* This indicates that this is the second packet in the conversation within the stream.

### **Conversation Completeness: Incomplete**

* This field shows whether the entire TCP conversation has been captured or not. In this case, it is incomplete, meaning some parts of the conversation are missing.

### **TCP Segment Len**

* The length of the TCP payload is 0, indicating this is a control packet (an acknowledgment) with no data.

### **Sequence Number:**

* **Relative Sequence Number: 1**
  + The sequence number keeps track of the order of the bytes sent.
* **Raw Sequence Number: 694202240**
  + The actual sequence number of this packet.

### **Next Sequence Number:**

* **Relative Next Sequence Number**
  + This is what the next sequence number will be after this packet.

### **Acknowledgment Number:**

* **Relative Acknowledgment Number**
  + The acknowledgment number indicates which byte the receiver is expecting next.

### **Header Length: 32 bytes**

* This indicates the size of the TCP header, in this case, 32 bytes. The header includes information such as flags, source/destination ports, and sequence numbers.

### **Flags: 0x010 (ACK)**

* This flag indicates the purpose of the packet. The flag 0x010 means that this is an acknowledgment (ACK) packet.

### **Window: 63784**

* The window size specifies how much data the sender can accept from the receiver before it must stop and wait for acknowledgment of the data.

### **Calculated Window Size: 63784**

* This field is the same as the "Window" field above but calculated to be user-friendly.

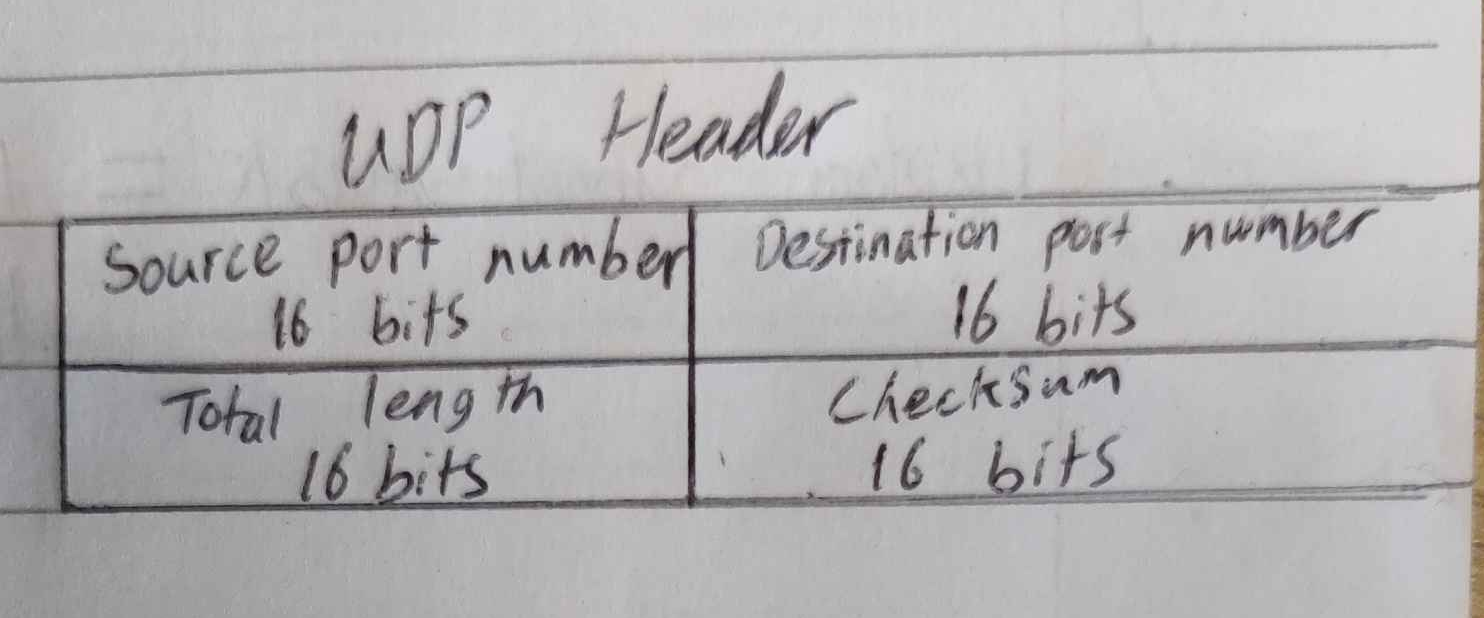
### **Window Scaling Factor: -1 (unknown)**

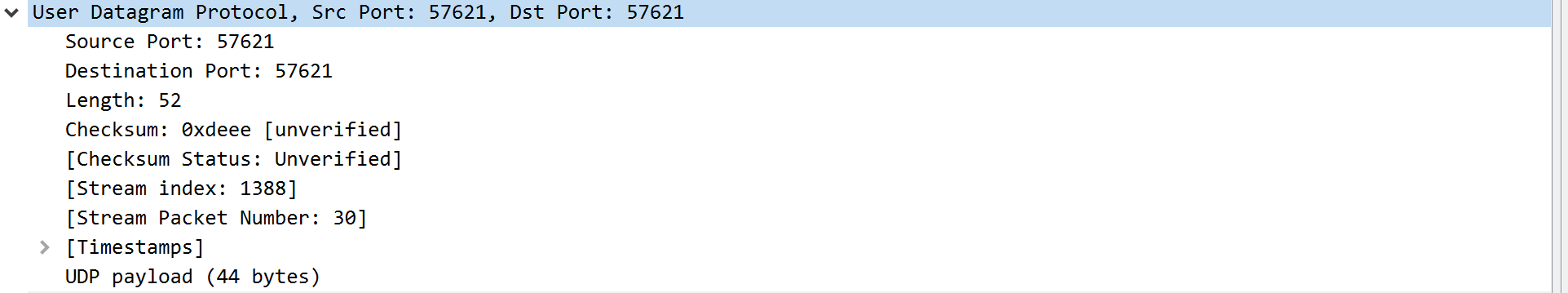
* This field is used to scale the TCP window size for large transfers.

### **Urgent Pointer: 0**

* The urgent pointer is set to 0, meaning there is no urgent data in this packet. The urgent pointer is used when specific data needs to be processed immediately by the receiving application.

3. Draw a UDP header. Capture packets using Wireshark and explain the fields for a particular UDP packet captured. Try to explain the purpose of each field.





### **Source Port: 57621**

* This is the source port number on the sender's device, from where the UDP packet originates.

### **Destination Port: 57621**

* The destination port is the port number on the recipient's side.

### **Length: 52**

* The length field indicates the size of the UDP packet, including both the header and the data (payload). In this case, the total length is 52 bytes.

### **Checksum: 0xdeee unverified**

* The checksum is a value used to verify the integrity of the UDP packet. It is calculated by the sender and used by the receiver to check for errors during transmission.

### **Checksum Status: Unverified**

* This field indicates that the checksum has not been verified yet.

### **Stream Index: 1388**

* The stream index groups packets within the same communication flow, enabling easier tracking of related UDP packets.

### **Stream Packet Number: 30**

* This is the 30th packet in this specific stream, meaning 29 packets have already been exchanged in this communication.

### **Timestamps**

* Provides information about the precise time the packet was sent and received.

### **UDP Payload (44 bytes)**

* The actual data carried by the UDP packet is 44 bytes long. The UDP header (typically 8 bytes) is part of the overall packet length of 52 bytes (as indicated by the "Length" field). The payload is the information being transmitted.