**Wireshark Group Assignment**

Group Members: Mark White and Vivienne Quist

The two online meeting spaces our group selected were: Zoom, for the video session, and Microsoft Teams for the text session.

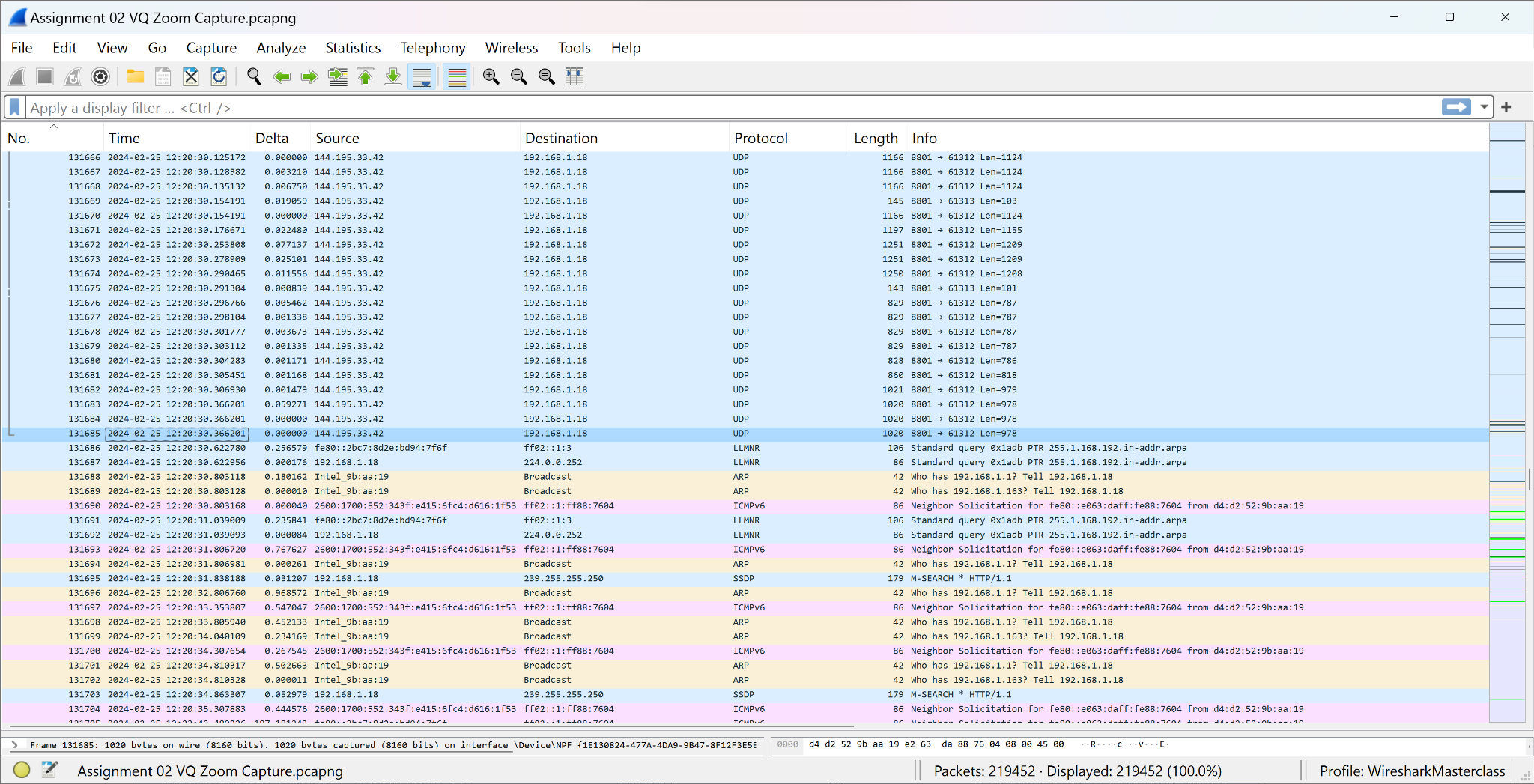
When utilizing Zoom, we noticed the data from the call uses a combination of UDP and TCP protocols connected using port 8801. We recognized the data is being transmitted through an intermediate server, since it is not being sent to our IP addresses but rather to an intermediate destination (144.195.33.42) .

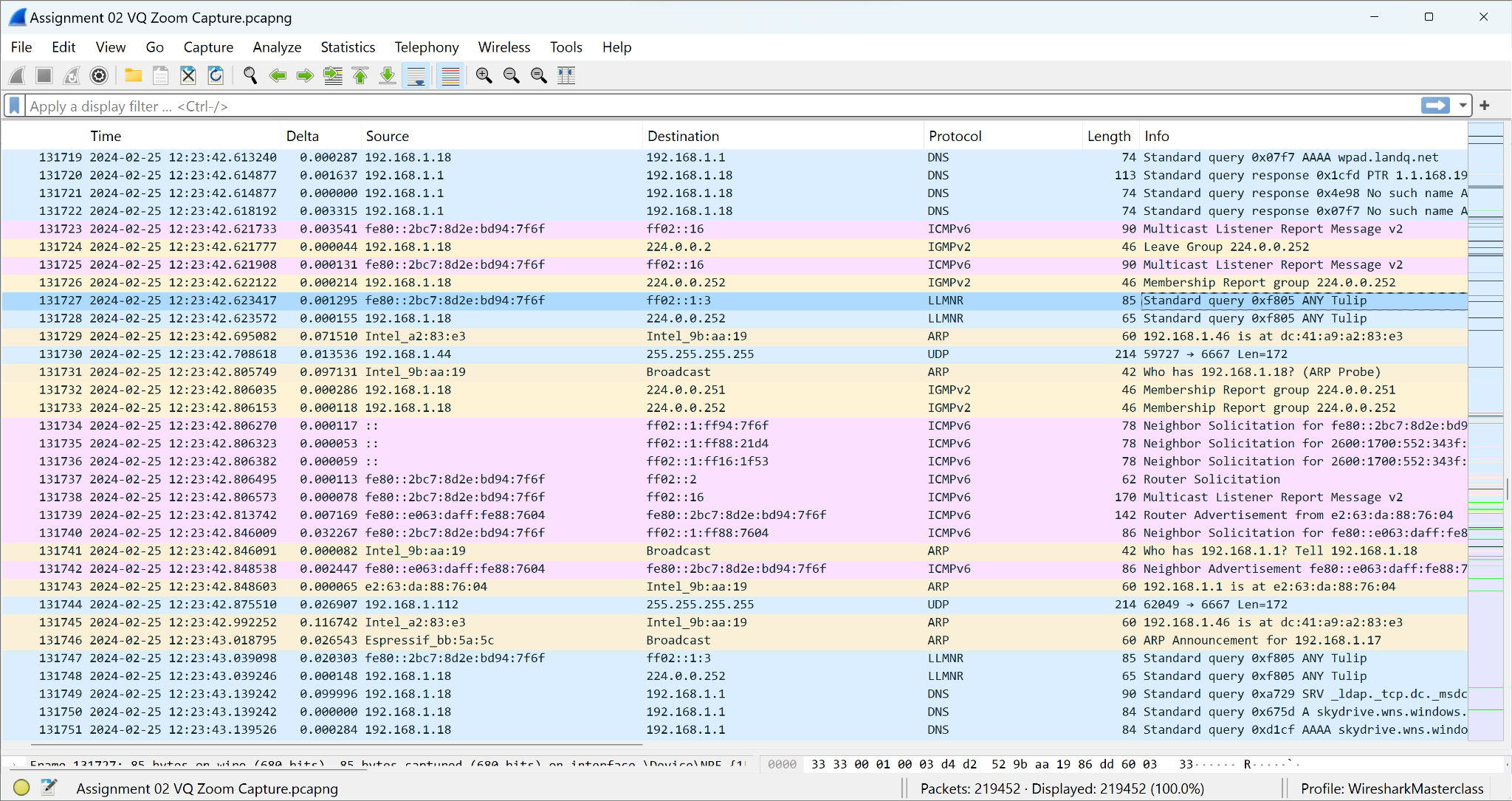
To introduce latency and lost packets Vivienne went outside to establish a poor connection and we looked at the difference in capturing packets. At a certain point, the call dropped, and the laptop disconnected from the network. At this point, Wireshark on Vivienne’s side showed a stop in the UDP packages at 12:20:30, which was followed by tens of thousands of packets, primarily DNS requests and TCP packets, before it became almost entirely large blocks of Bad TCP connections at 12:24:07, as the applications on her computer, including Zoom attempted to reestablish the connection. The minutes between the end of the UDP packages and the beginning of the Bad TCP blocks reflect the reality that the Zoom call was one of the first things sacrificed in light of the bad connection, although Zoom also spent some time after the call dropped attempting to reconnect. As Mark remained in the call during the latency test, the UDP packages continued to send as normal while Vivienne’s face cam was frozen. Once Vivienne disconnected large blocks of bad TCP connections were apparent from Mark’s end.

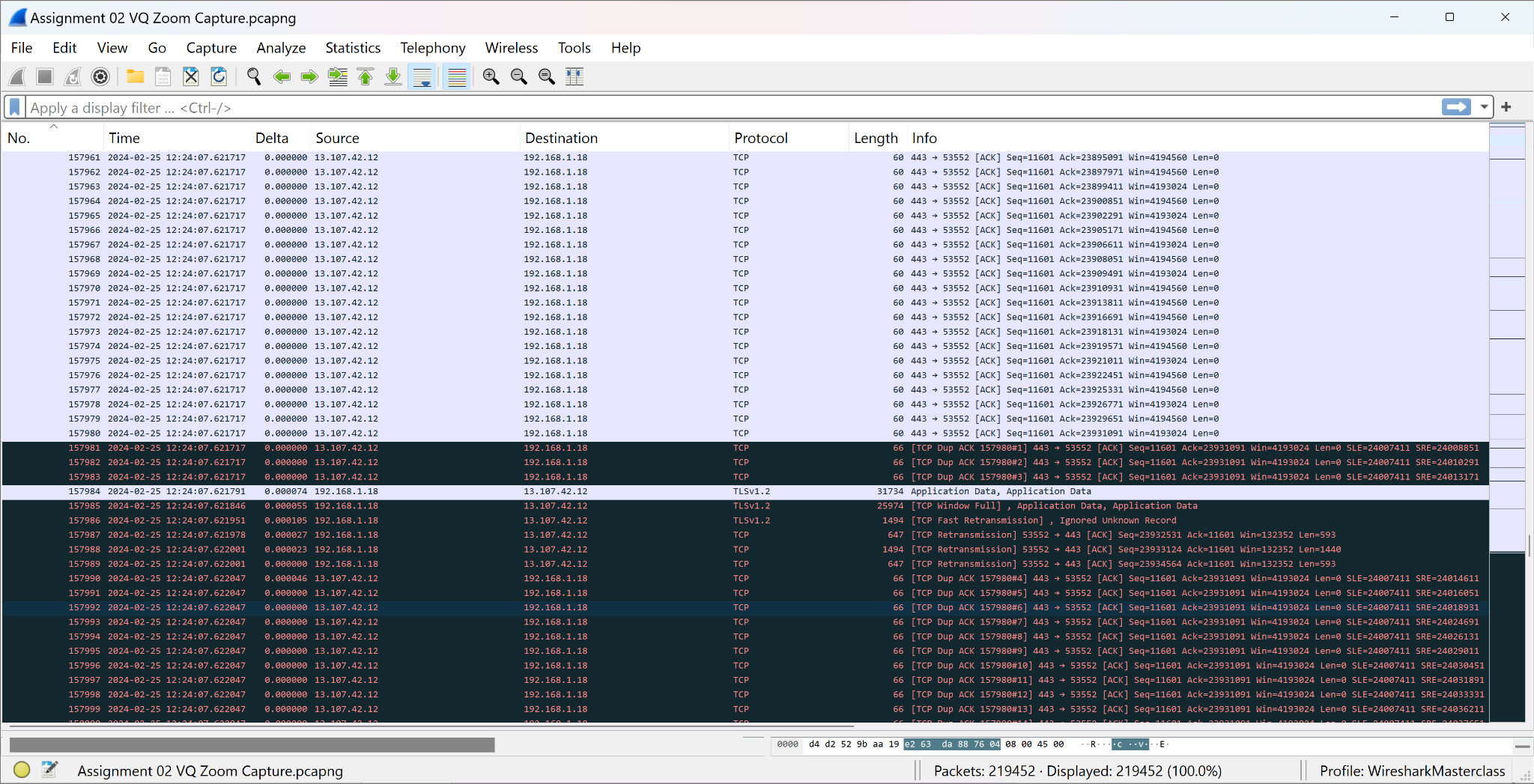
When utilizing Microsoft Teams, we identified several possible servers for the Teams messaging interactions. We settled on addresses 52.112.95.104 (Mark) and 52.123.187.144 (Vivienne), being the most probable candidates. Packages exchanged between these server addresses and home IP addresses mainly followed TCP or TLSv1.2/TLSv1.3. Based on the Wireshark capture it looks like sending text-based communications differs significantly from video- or audio-based communications in that while video/audio communications appears to use a combination of TCP and UDP, text communications depend mostly on TCP connections. Within the Wireshark capture, the handshake interactions between client and server also appear, and the duration between these interactions indicates that Teams uses persistent TCP connections with a set timeout interval, so that the delay in real-time exchanges can be reduced, without being burdensome to the network and servers long-term.

Furthermore, the Wireshark data for the Zoom call and Teams messages did not only differ in protocol tendencies but in size as well. The packets related to the Teams messages tended to range from dozens of bytes to several thousand bytes, varying depending on the purpose of the packets, and the size of the message being sent. The packets identified as being related to the Zoom session, on the other hand, were usually between 800 and 1300 bytes. This makes sense, as for video/audio calls it is expected that the quality of the video and audio be consistent and stable, which would require the regularity and homogeneousness of the UDP packets sent during the Zoom call. Text-based communications on the other hand, do not require data to be continuously transmitted to the other party, and the size of the messages sent may vary. This is reflected in the variation between the different TCP messages sent during the Teams conversation, as even though they utilized a persistent connection, the usage of this connection could vary wildly minute-to-minute, or second-to-second.

As we looked through the packets being sent when using Teams we noticed the destination address would often change completely, unlike when we used Zoom which had an intermediate server. Once we sent a few messages, waited, and then sent more, the destination address would often change. We concluded that there could be many reasons the service is set up this way, as teams may use a distributed network infrastructure. To possibly provide load balance, ensure high availability, and flexibility. Overall our group enjoyed using wireshark and getting a better grasp of how data is moved from one location to another.







The above screenshots show three key moments in Vivienne’s Wireshark capture of the Zoom call: the first showing just before the connection became unstable and the loss of connection, the second showing a typical moment after the initial loss of connection, and the third at which point any level of internet connection was lost, leading applications to send and resend packets until they receive a response.

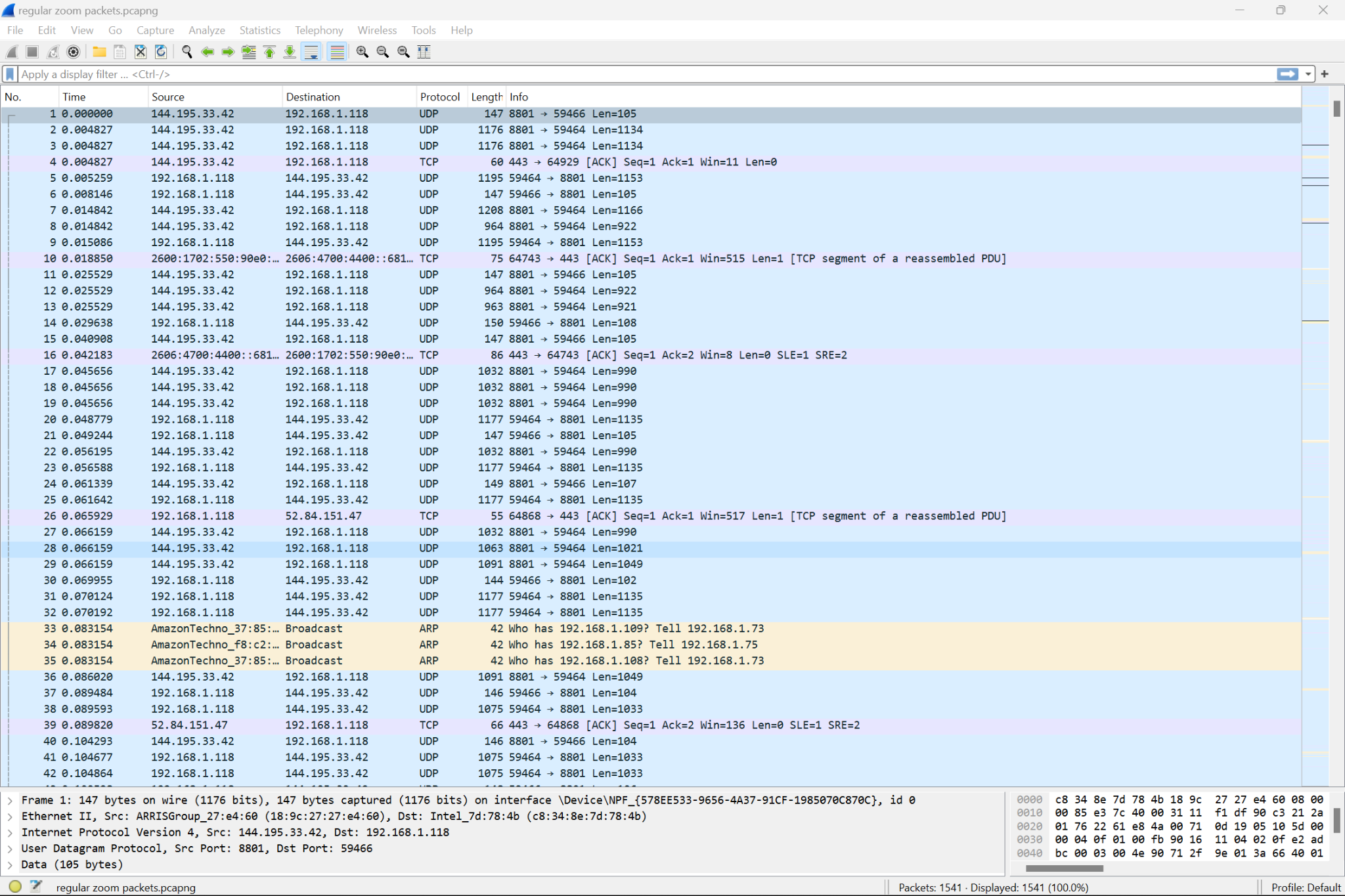


Fig. A: Good Connection

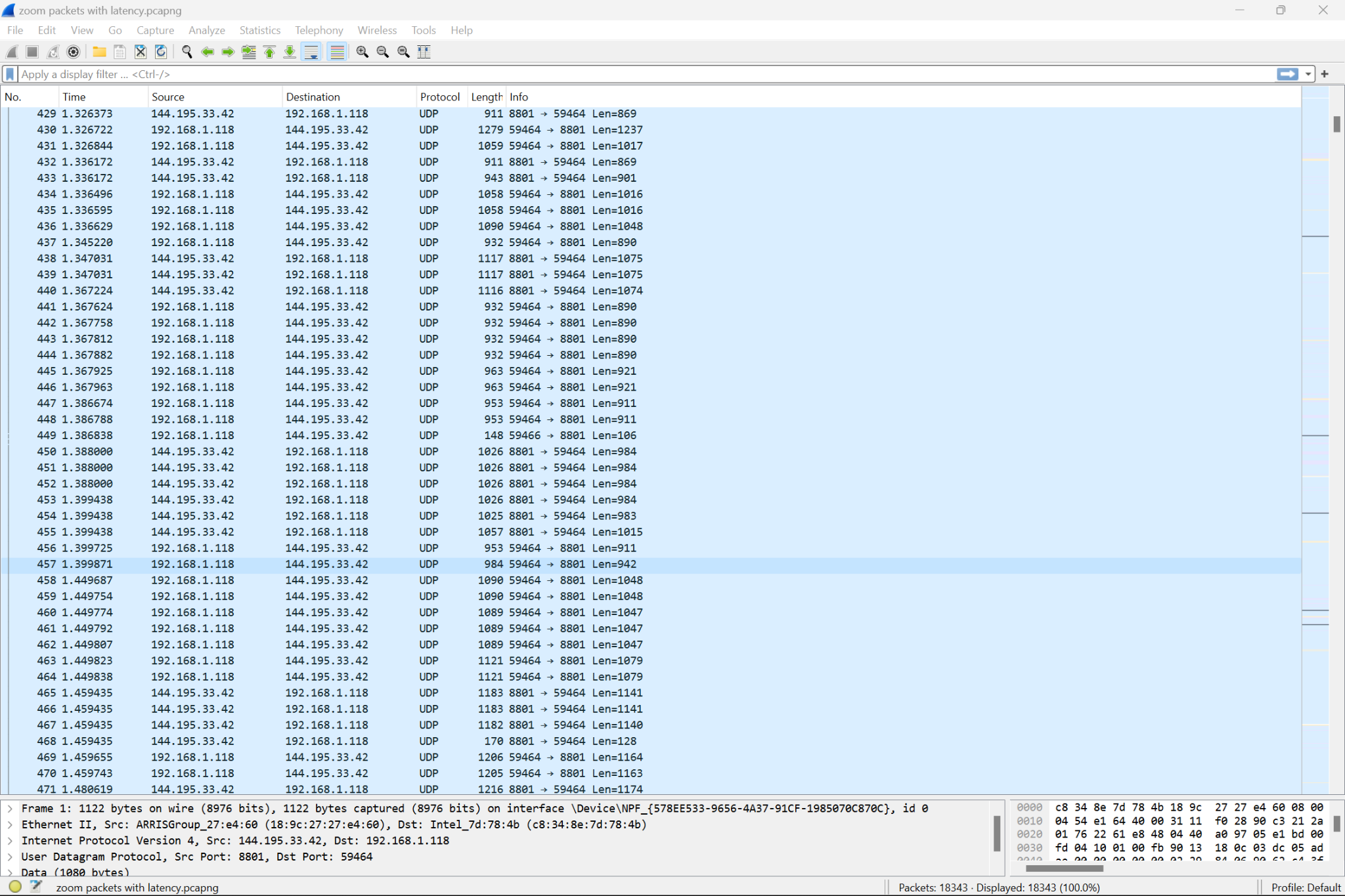


Fig. B: Unstable Connection

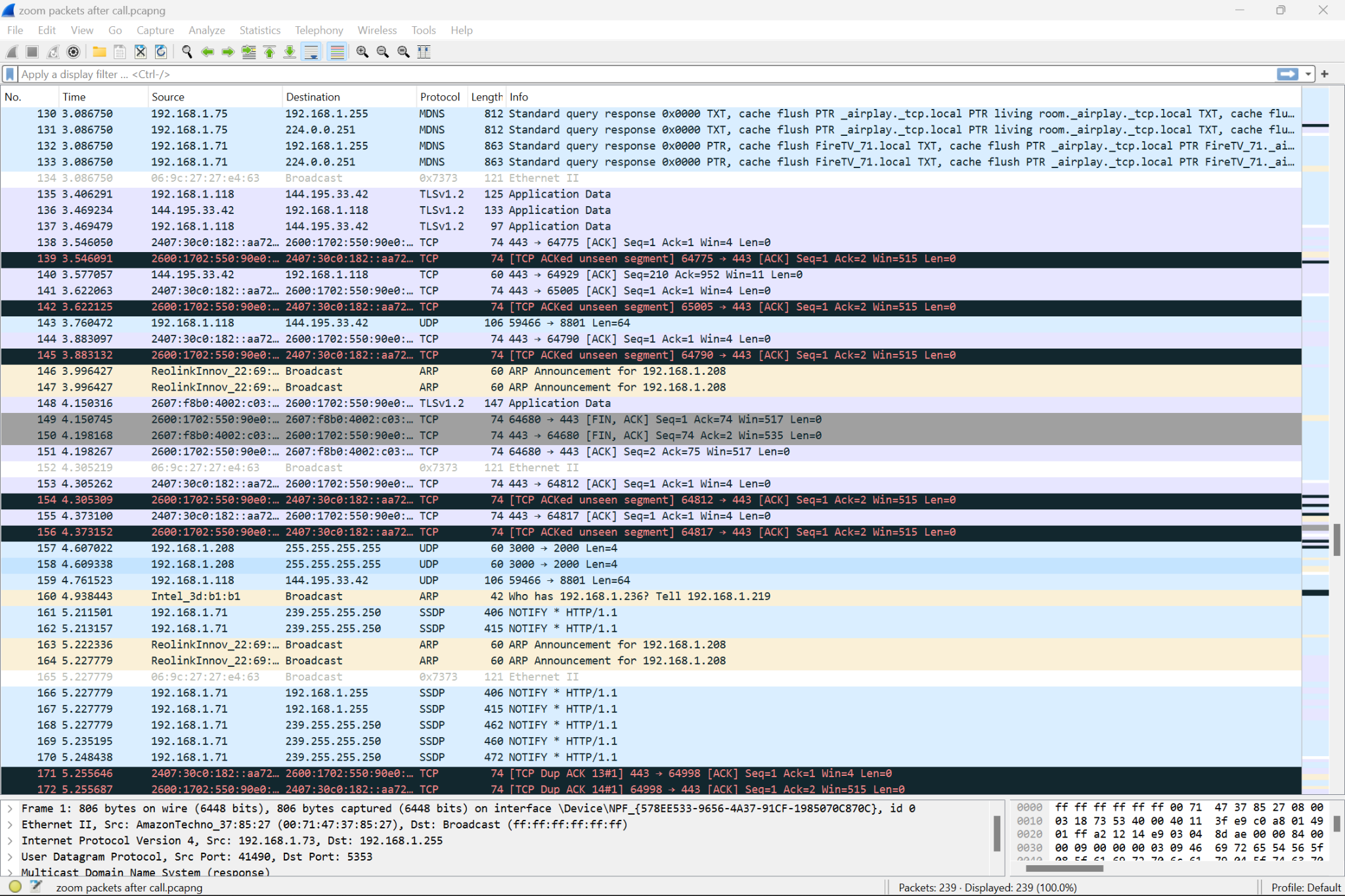


Fig. C: Lost Connection

The three screenshots above, show Mark’s point of view using Wireshark as we established a good connection, an unstable connection, and as we lost connection through our Zoom call.