1. **Wireshark: UDP**

1) UDP-заголовок содержит 4 поля: Source Port, Destination Port, Length, Checksum.

2) Каждое поле UDP-заголовка имеет длину 2 байта.

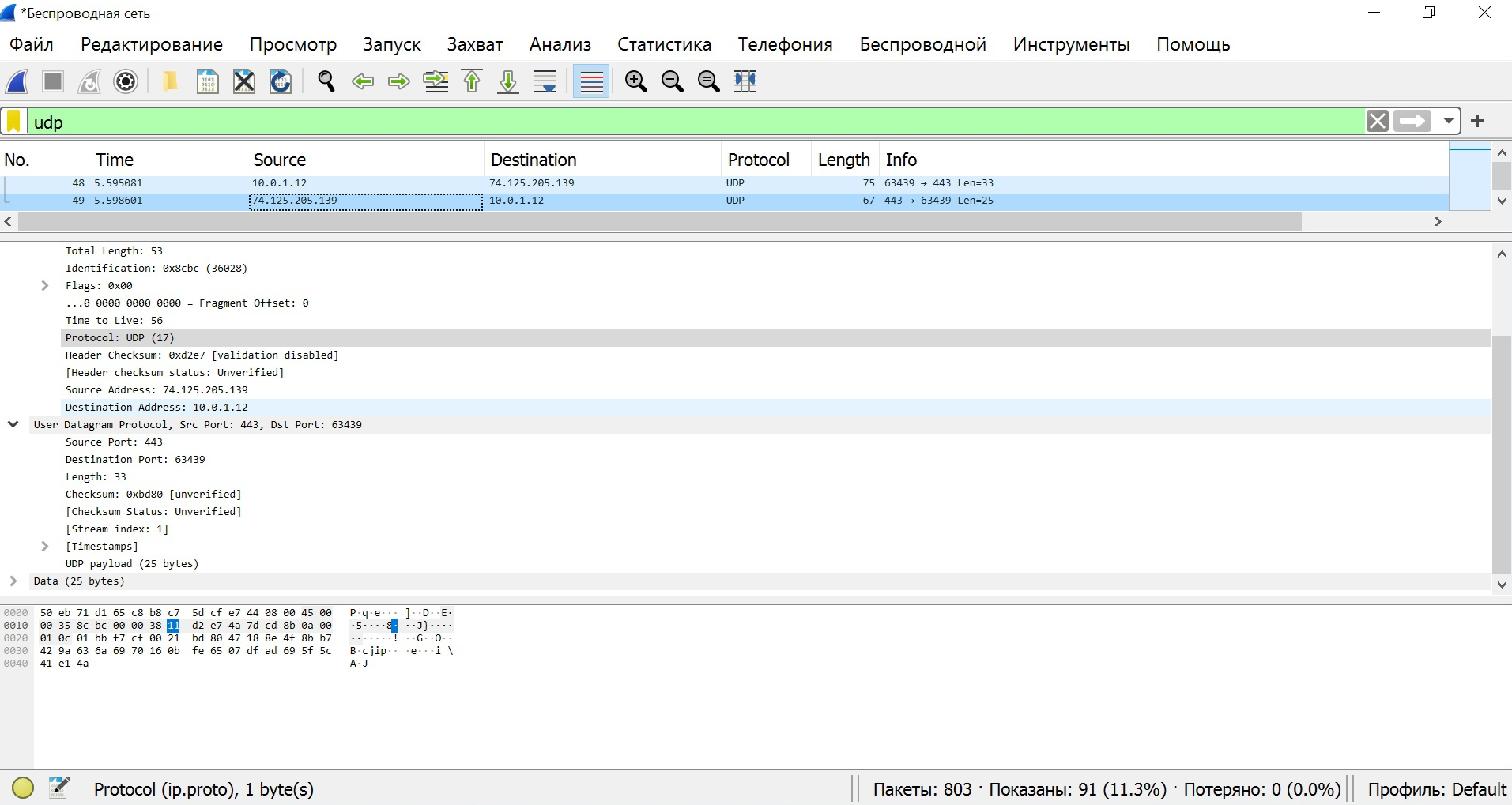
3) Значение в поле Length - это суммарная длина заголовка и полезной нагрузки.

4) Понимаем, что в Length суммарная длина, а на её значение отводится 2 байта, т.е. можно передать значения до 2^16 - 1 = 65535, но 8 байт занимает заголовок, значит на полезную часть остаётся 65535 - 8 = 65527 байт.

5) Так как длина поля с номером отправителя 2 байта, то его максимальное значение 2^16 - 1 = 65535.

6) Номер протокола для протокола UDP равен 17 (в 10СС) и 11 (в 16СС)

7) Source Port отправляемого = Destination Port ответного, Source Port ответного = Destination Port отправляемого



**3. Задачки**

**Задача 1**

Пусть (m) - максимальный номер пакета для которого был выслан ACK. Тогда понимаем, что мы его получили, а значит окно отправителя было сдвинуто так что оно задевало этот пакет, т.е. пакеты с номерами меньшими (m-n+1) отправляться не будут. Также понимаем, что на (m+1) пакет ACK не высылался, а значит окно отправителя не может задевать пакеты с номерами больше (m+n). Значит мы можем получать пакеты с номерами от (m-n+1) до (m+n). Их здесь (2n).

Предположим теперь, что (2^k) < (2n). Тогда для пакета с номером (m+n) найдётся пакет с таким же номером среди пакетов от (m-n+1) до (n), пусть он имеет номер (x).

Рассмотрим 2 наши ситуации:

(а) К отправителю пришли ACK от пакетов с (m-n+1) до (n) и он отправил пакеты с (n+1) до (m+n). Но дошёл только (m+n).

(б) К получателю пришли пакеты с (m-n+1) до (n), но обратно не дошёл не один ACK. Тогда отправитель снова пошлёт пакеты с номерами с (m-n+1) до (n). И дойдёт только пакет с номером (x).

В таком случае для получателя эти ситуации идентичны, но пакеты были получены разные, значит чтобы избежать такой ситуации надо чтобы (2^k) >= (2n), а значит при максимальное n получаем при (2^k) = (2n), т.е. (n) = (2^k-1).

**Задача 2**

Значит размер окна должен быть хотя бы 2451 пакет.