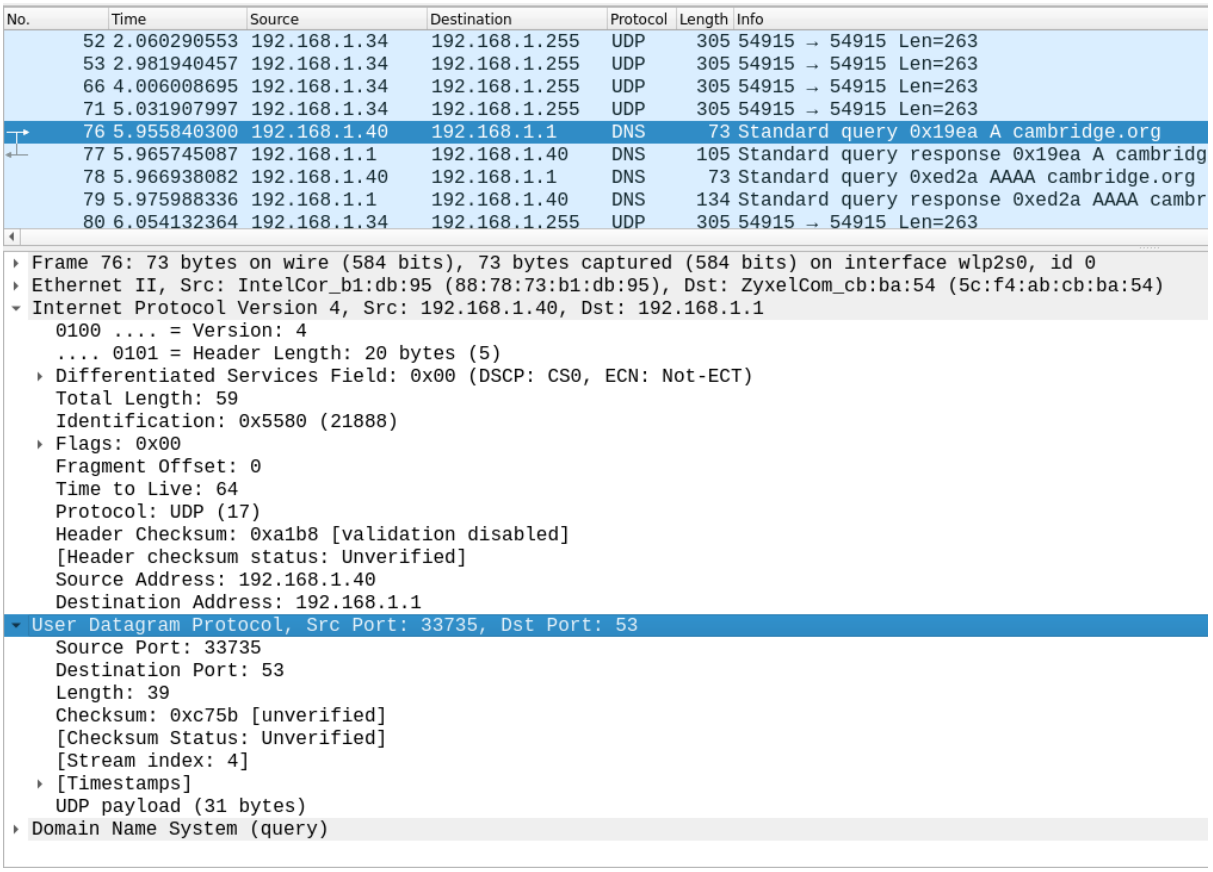
Wireshark: UDP

1. В заголовке 4 поля: Source Port, Destination Port, Length, Checksum
2. Каждое поле по 2 байта
3. Length – суммарная длина пакета (39 = 2*4 + 31, где 31 – размер данных)
4. Максимальное значение длины 2^16-1, из них 8 байт на заголовок => максимальное количество данных 2^16-9 = 65527 байт
5. Раз под порт отводится два байта, то максимальное значение 2^16 - 1 = 65535
6. Protocol: UDP (17). В 16-ричной системе это 11.
7. Порты src и dst поменялись местами.



Задачки

Задача 1

Рассмотрим ситуацию со стороны получателя. Обозначим за A максимальный номер пакета, для которого мы выслали ACK.
Заметим, во-первых, что отправитель не будет передавать нам пакеты с номерами <= A-n. где n – размер окна. Действительно, раз отправитель прислал пакет A, он обязан был сдвинуть свое окно так, чтобы оно задевало пакет A, а значит ACK для всех пакетов <= A-n были получены. Во-вторых, отправитель не пришлет нам пакет с

номером >A+n, поскольку на пакет A+1 мы еще не выслали ACK, а значит окно отправителя не может содержать пакеты с номерами >A+n.
Таким образом, в каждый момент времени получатель ожидает пакет из диапазона [A-n+1, A+n], длина которого 2*n. Если мы кодируем номера этих пакетов различными числами (2^k >= 2n), то мы точно будем различать все ситуации.
Допустим теперь, что у A+n и какого-то другого пакета B совпали номера в нашей нумерации. Мы не отличим ситуации, когда
а) Отправитель послал пакеты A-n+1 … A, получил все ACK, послал пакеты A+1 … A+n, и из них к нам пришел только последний.
б) Отправитель послал пакеты A-n+1 … A, все ACK потерялись, отправитель посылает пакеты повторно, из них доходит только B.
Таким образом, максимальный возможный размер окна n = 2^(k-1).

Задача 2

Пусть размер окна n. Используемая пропускная способность канала $R0 = \frac{nL}{RTT + L/R}$.

Условие: $R0 \geq 0.98R$, то есть $\frac{nL}{RTT + L/R} \geq 0.98R$

$n \geq 0.98R * RTT/L + 0.98 = 0.98 * 10^{9} * 30 * 10^{-3}/(1,5 * 8 * 10^{3}) = 2450.98$
То есть размер окна должен быть хотя бы 2451.