

Wireshark: UDP

1. В заголовке 4 поля: Source Port, Destination Port, Length, Checksum
2. Каждое поле по 2 байта
3. Length – суммарная длина пакета ($39 = 2 \cdot 4 + 31$, 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.

No.	Time	Source	Destination	Protocol	Length	Info
52	2.060290553	192.168.1.34	192.168.1.255	UDP	305	54915 → 54915 Len=263
53	2.981940457	192.168.1.34	192.168.1.255	UDP	305	54915 → 54915 Len=263
66	4.006008695	192.168.1.34	192.168.1.255	UDP	305	54915 → 54915 Len=263
71	5.031907997	192.168.1.34	192.168.1.255	UDP	305	54915 → 54915 Len=263
76	5.955840300	192.168.1.40	192.168.1.1	DNS	73	Standard query 0x19ea A cambridge.org
77	5.965745087	192.168.1.1	192.168.1.40	DNS	105	Standard query response 0x19ea A cambridg
78	5.966938082	192.168.1.40	192.168.1.1	DNS	73	Standard query 0xed2a AAAA cambridge.org
79	5.975988336	192.168.1.1	192.168.1.40	DNS	134	Standard query response 0xed2a AAAA cambr
80	6.054132364	192.168.1.34	192.168.1.255	UDP	305	54915 → 54915 Len=263

Frame 76: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface wlp2s0, id 0

Ethernet II, Src: IntelCor_b1:db:95 (88:78:73:b1:db:95), Dst: ZyxelCom_cb:ba:54 (5c:f4:ab:cb:ba:54)

Internet Protocol Version 4, Src: 192.168.1.40, Dst: 192.168.1.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 59

Identification: 0x5580 (21888)

Flags: 0x00

Fragment Offset: 0

Time to Live: 64

Protocol: UDP (17)

Header Checksum: 0xa1b8 [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.1.40

Destination Address: 192.168.1.1

User Datagram Protocol, Src Port: 33735, Dst Port: 53

Source Port: 33735

Destination Port: 53

Length: 39

Checksum: 0xc75b [unverified]

[Checksum Status: Unverified]

[Stream index: 4]

[Timestamps]

UDP payload (31 bytes)

Domain Name System (query)

Задачи

Задача 1

Рассмотрим ситуацию со стороны получателя. Обозначим за А максимальный номер пакета, для которого мы послали ACK.

Заметим, во-первых, что отправитель не будет передавать нам пакеты с номерами $\leq A-n$, где n – размер окна. Действительно, раз отправитель прислал пакет А, он обязан был сдвинуть свое окно так, чтобы оно задевало пакет А, а значит ACK для всех пакетов $\leq A-n$ были получены. Во-вторых, отправитель не пришлет нам пакет с

номером $>A+n$, поскольку на пакет $A+1$ мы еще не послали 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 $2n$. Если мы кодируем номера этих пакетов различными числами ($2^k \geq 2n$), 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 $A-n+1 \dots A$, получил все ACK, послал пакеты $A+1 \dots A+n$, и из них к нам пришел только последний.

б) Отправитель послал пакеты $A-n+1 \dots A$, 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 $R_0 = \frac{nL}{RTT + L/R}$.

Условие: $R_0 \geq 0.98R$, 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.