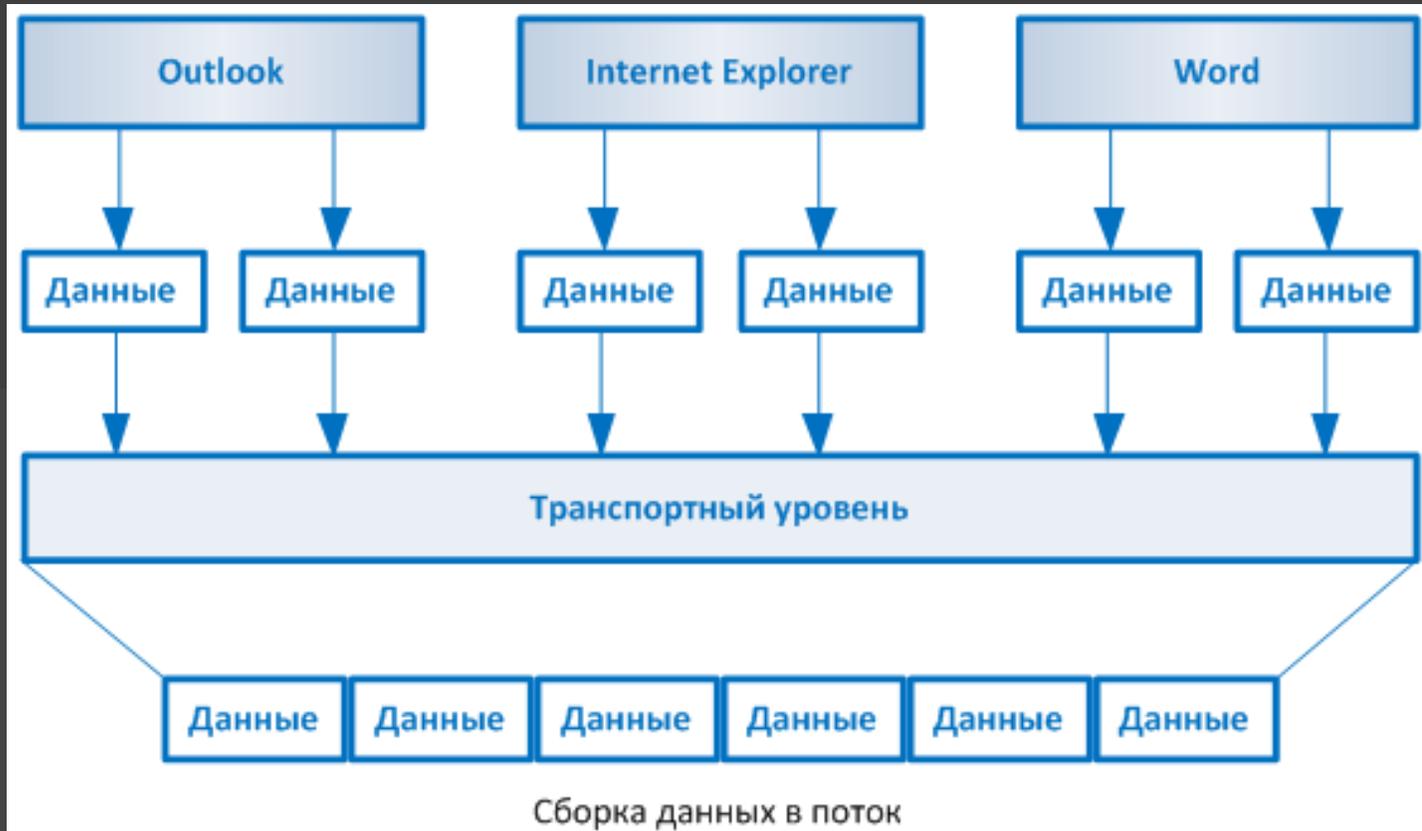


# Протоколы TCP и UDP

# Введение



**Что такое транспортный уровень?**

- Посредник между приложениями и сетью.
- Передаёт данные из программы в программу.

# Роль транспортных протоколов

- Доставка данных
- Контроль ошибок
- Разделение потоков
- Работа с портами



# Два главных протокола

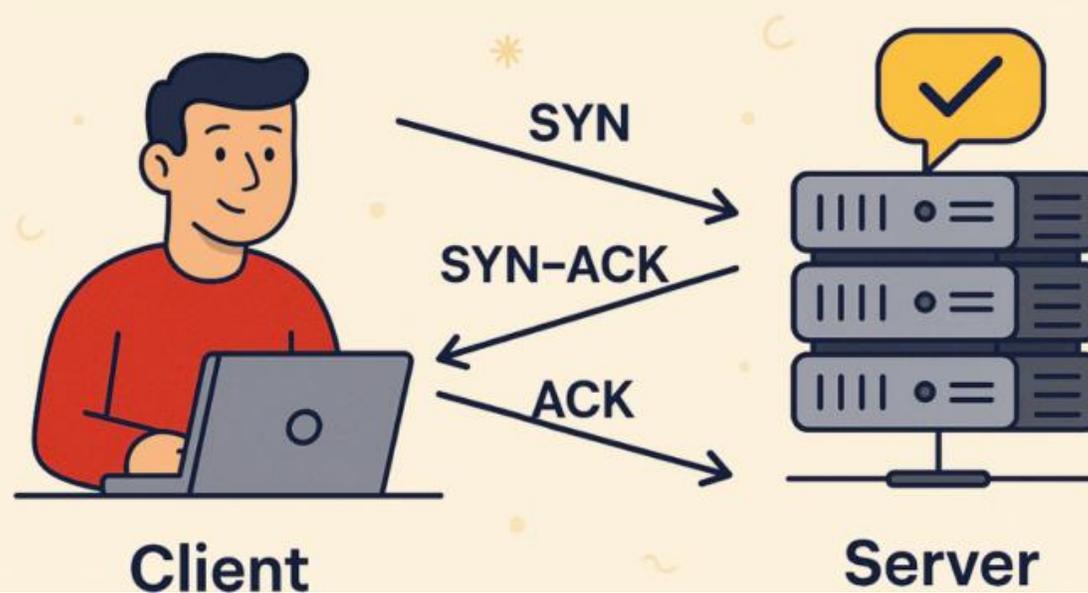


# Пример из жизни



# Протокол TCP

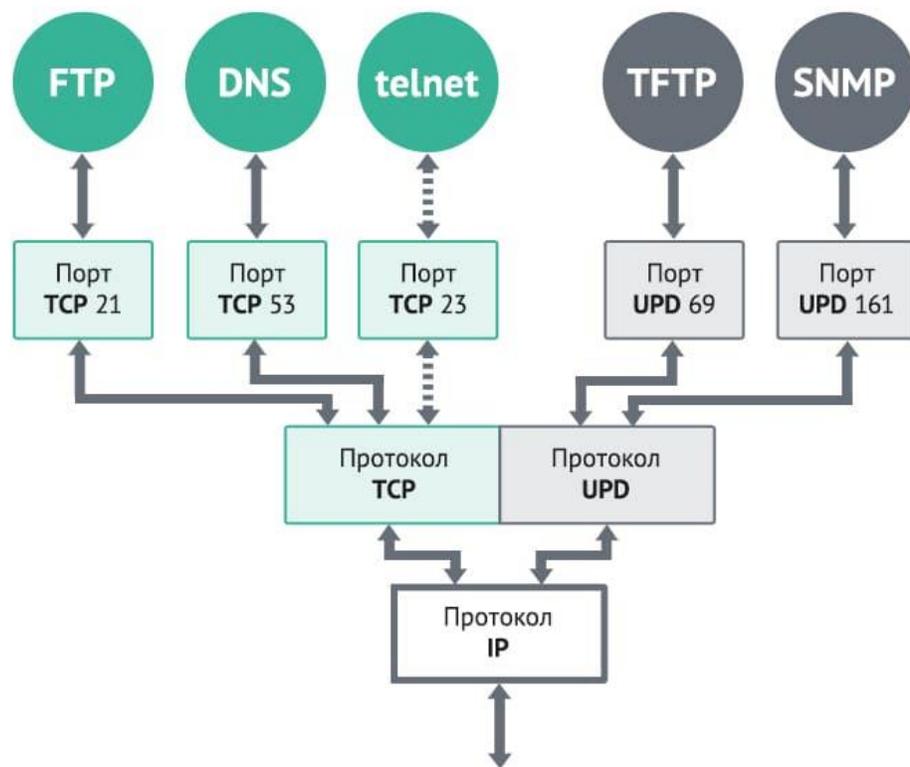
## TCP Handshake process



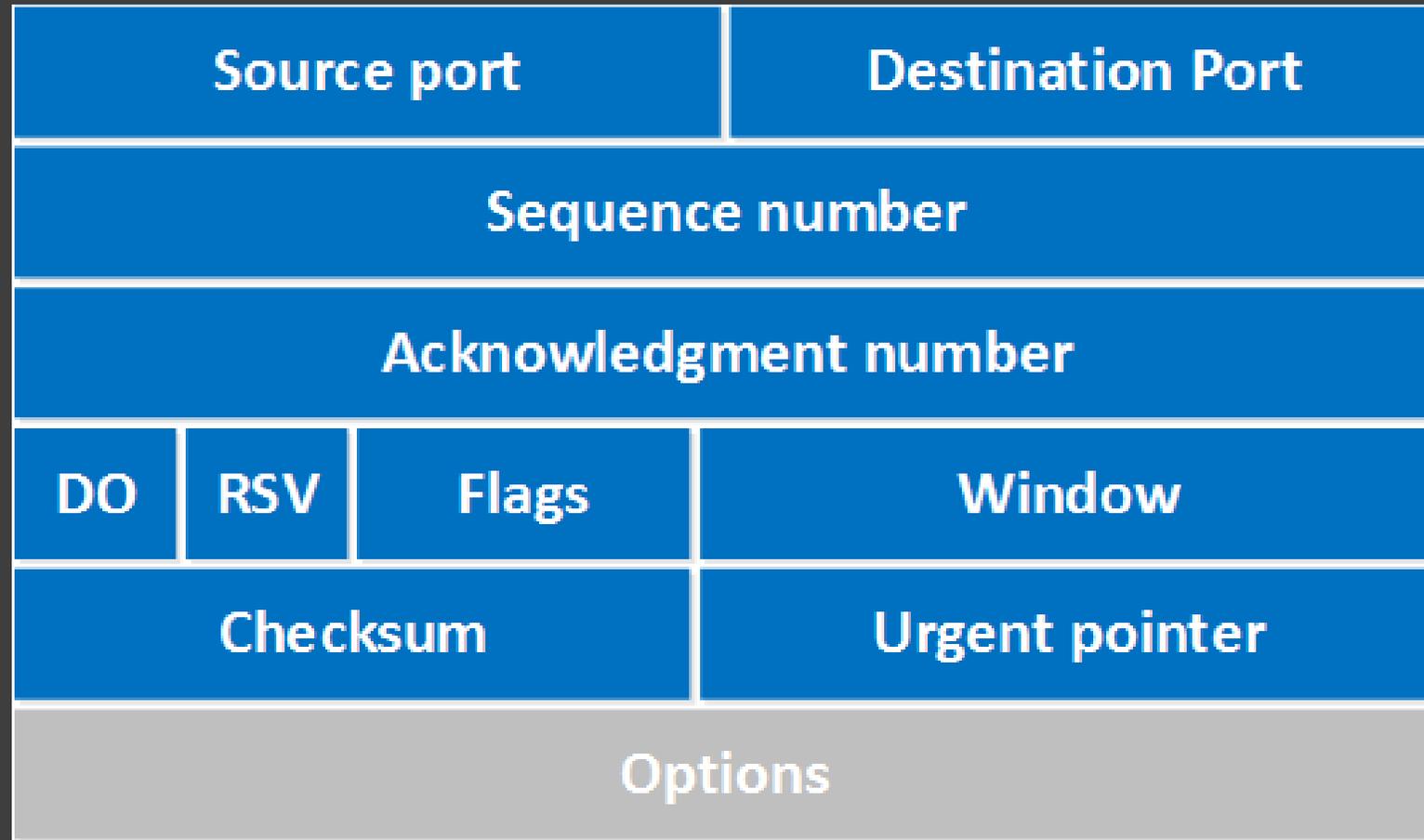
# Основные особенности TSP

- Дуплексность и потоки
- Мультиплексирование (номера портов)
- Управление потоком

# Порты TCP



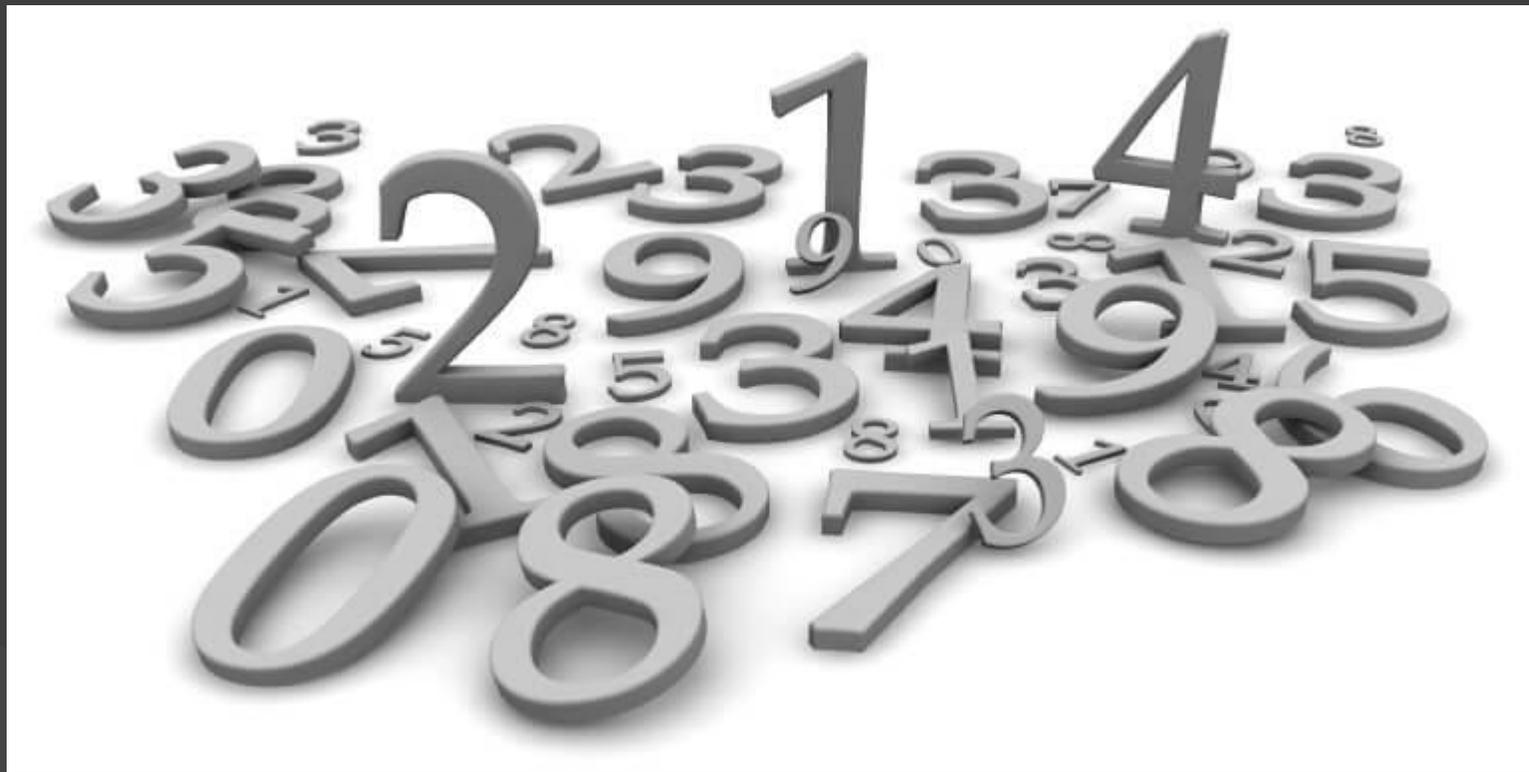
# Что такое сегмент TCP



# Зачем нужны сегменты

- Исходный порт
- Порт получателя
- Порядковый номер
- Номер подтверждения
- Смещение данных
- Управляющие флаги
- Размер окна
- Контрольная сумма
- Указатель срочных данных
- Опции

# Нумерация сегментов



# Зачем нужна нумерация



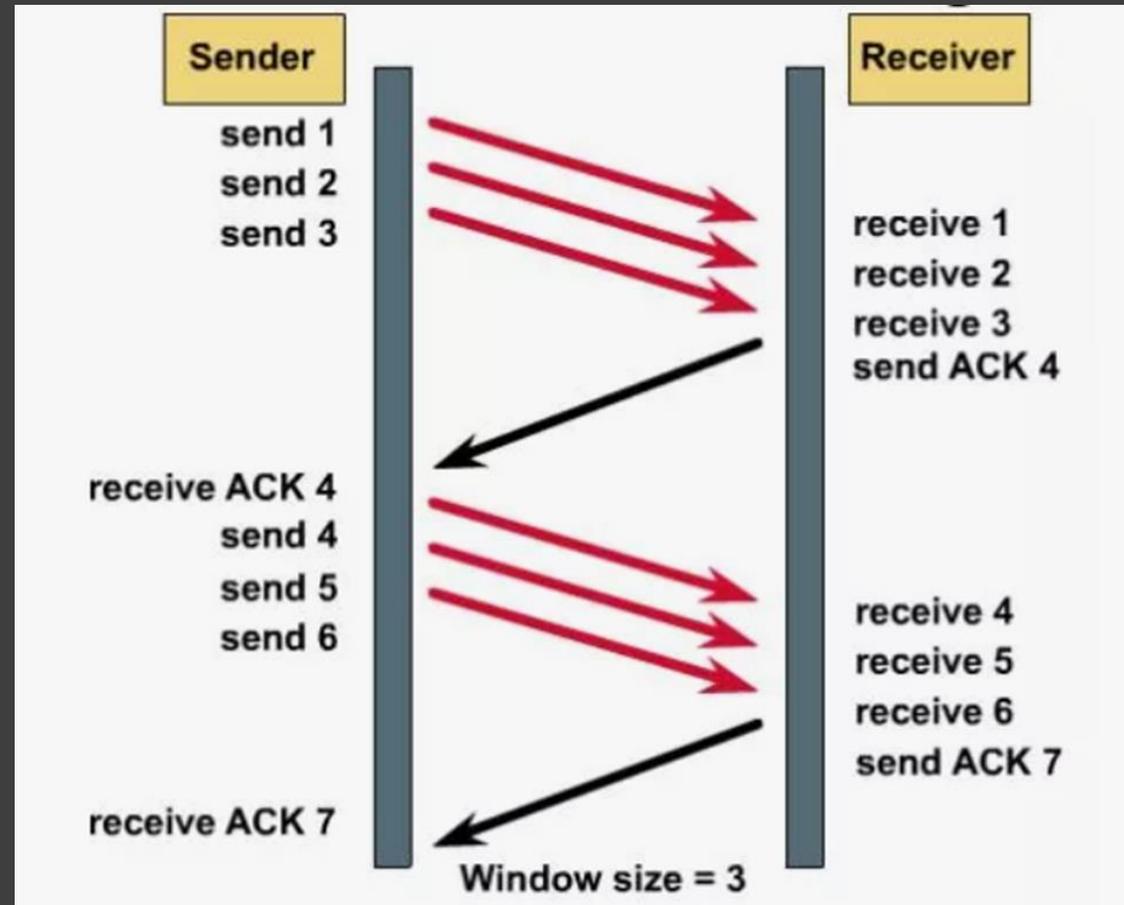
# Установление соединения



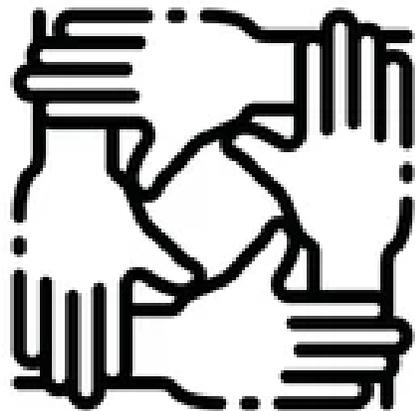
# Передача данных



# Управление потоком и скользящее окно



# Завершение сеанса



# Практическая часть(ТСП)

The screenshot displays the GNS3 network simulator interface. On the left, the 'All devices' panel lists various network components such as ATM switch, c3745, Cloud, Ethernet hub, Ethernet switch, Frame Relay switch, NAT, and VPCS. The main workspace shows a topology with two routers, R1 and R2, connected by a single link. The right side of the interface features a 'Topology Summary' panel showing the console for R1 (telnet localhost:5000) and R2 (telnet localhost:5001). Below it, a 'Servers Summary' panel shows the system resources for the host 'whitakker', including CPU usage at 29.5% and RAM usage at 51.2%. At the bottom, a console window displays the GNS3 management console text: 'Running GNS3 version 2.2.45 on Windows (64-bit) with Python 3.10.11 Qt 5.15.2 and PyQt 5.15.10. Copyright (c) 2006-2025 GNS3 Technologies. Use Help -> GNS3 Doctor to detect common issues.' followed by a prompt '=>'.

# Настройка сервера и клиента

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#hostname SERVER
SERVER(config)#ip domain-name test.local
SERVER(config)#int fa0/0
SERVER(config-if)#ip address 192.168.1.1 255.255.255.0
SERVER(config-if)#no shutdown
SERVER(config-if)#n
*Mar 1 00:01:34.123: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Mar 1 00:01:35.123: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
SERVER(config-if)#no keepalive
SERVER(config-if)#exit
SERVER(config)#line vty 0 4
SERVER(config-line)#password cisco
SERVER(config-line)#login
SERVER(config-line)#transport input telnet
SERVER(config-line)#exit
SERVER(config)#enable secret cisco
SERVER(config)#end
SERVER#wr
*Mar 1 00:02:56.235: %SYS-5-CONFIG_I: Configured from console by console
SERVER#wr mem
Building configuration...
[OK]
```

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#hostname CLIENT
CLIENT(config)#int fa0/0
CLIENT(config-if)#ip addr 192.168.1.2 255.255.255.0
CLIENT(config-if)#no shutdown
CLIENT(config-if)#exit
CLIENT(config)#
*Mar 1 00:04:16.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
o up
*Mar 1 00:04:17.047: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
et0/0, changed state to up
CLIENT(config)#ip name-server 192.168.1.1
CLIENT(config)#end
CLIENT#wr
*Mar 1 00:04:41.635: %SYS-5-CONFIG_I: Configured from console by console
CLIENT#wr mem
Building configuration...
[OK]
```

# Генерация трафика

```
CLIENT#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 24/40/64 ms
CLIENT#telnet 192.168.1.1
Trying 192.168.1.1 ... Open

User Access Verification

Password:
SERVER>exit

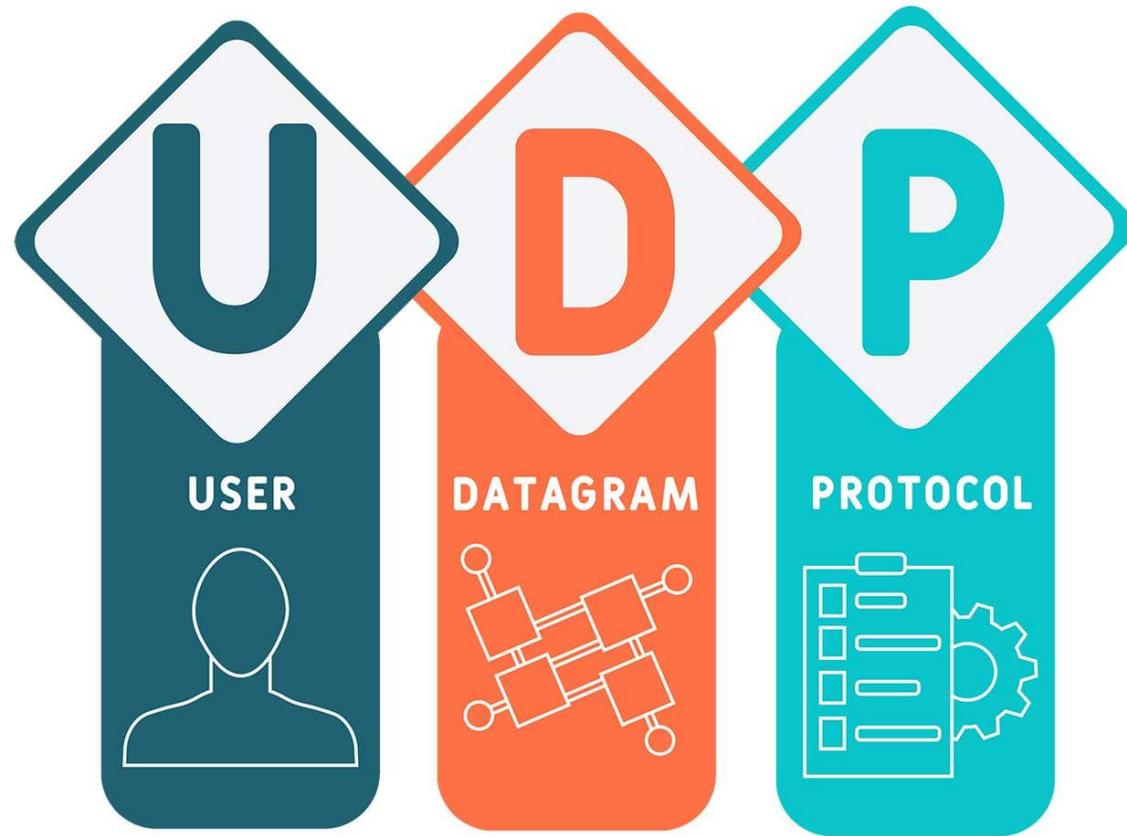
[Connection to 192.168.1.1 closed by foreign host]
```

# Перехват трафик и анализ

1	0.000000	c4:02:2d:64:00:00	c4:02:2d:64:00:00	LOOP	60 Reply
2	6.186415	c4:02:2d:64:00:00	Broadcast	ARP	60 Who has 192.168.1.1? Tell 192.168.1.2
3	6.202463	c4:01:0e:50:00:00	c4:02:2d:64:00:00	ARP	60 192.168.1.1 is at c4:01:0e:50:00:00
4	8.170804	192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) request id=0x0000, seq=1/256, ttl=255 (reply in 5)
5	8.186679	192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) reply id=0x0000, seq=1/256, ttl=255 (request in 4)
6	8.201775	192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) request id=0x0000, seq=2/512, ttl=255 (reply in 7)
7	8.249794	192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) reply id=0x0000, seq=2/512, ttl=255 (request in 6)
8	8.265524	192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) request id=0x0000, seq=3/768, ttl=255 (reply in 9)
9	8.281631	192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) reply id=0x0000, seq=3/768, ttl=255 (request in 8)
10	8.297261	192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) request id=0x0000, seq=4/1024, ttl=255 (reply in 11)
11	8.312897	192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) reply id=0x0000, seq=4/1024, ttl=255 (request in 10)
12	9.998745	c4:02:2d:64:00:00	c4:02:2d:64:00:00	LOOP	60 Reply
13	20.000234	c4:02:2d:64:00:00	c4:02:2d:64:00:00	LOOP	60 Reply
14	24.452390	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [SYN] Seq=0 Win=4128 Len=0 MSS=1460
15	24.468077	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=1460
16	24.483916	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [ACK] Seq=1 Ack=1 Win=4128 Len=0
17	24.499031	192.168.1.1	192.168.1.2	TELNET	66 Telnet Data ...
18	24.499031	192.168.1.2	192.168.1.1	TELNET	63 Telnet Data ...
19	24.515002	192.168.1.1	192.168.1.2	TELNET	96 Telnet Data ...
20	24.515002	192.168.1.2	192.168.1.1	TCP	60 [TCP Dup ACK 16#1] 27717 → 23 [ACK] Seq=10 Ack=1 Win=4128 Len=0
21	24.530691	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
22	24.530691	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
23	24.546386	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
24	24.546386	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
25	24.562503	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
26	24.562503	192.168.1.2	192.168.1.1	TELNET	63 Telnet Data ...
27	24.733679	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [ACK] Seq=25 Ack=67 Win=4062 Len=0
28	24.765029	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [ACK] Seq=67 Ack=25 Win=4104 Len=0
29	25.592738	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
30	25.779424	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
31	25.810732	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [ACK] Seq=67 Ack=27 Win=4102 Len=0
32	25.906005	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
33	26.092225	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
34	26.123591	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [ACK] Seq=67 Ack=29 Win=4100 Len=0
35	26.202205	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
36	26.420964	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [ACK] Seq=67 Ack=30 Win=4099 Len=0
37	26.530873	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
38	26.546830	192.168.1.1	192.168.1.2	TELNET	63 Telnet Data ...
39	26.748165	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [ACK] Seq=32 Ack=76 Win=4053 Len=0
40	30.015152	c4:02:2d:64:00:00	c4:02:2d:64:00:00	LOOP	60 Reply
41	30.812662	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
42	30.828713	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
43	30.936890	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
44	30.952952	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
45	31.060663	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
46	31.076582	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
47	31.218270	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
48	31.234298	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
49	31.452293	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [ACK] Seq=36 Ack=80 Win=4049 Len=0
50	31.732854	192.168.1.2	192.168.1.1	TELNET	60 Telnet Data ...
51	31.748904	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...

51	31.748904	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
52	31.858626	192.168.1.1	192.168.1.2	TELNET	60 Telnet Data ...
53	31.874600	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [FIN, PSH, ACK] Seq=88 Ack=38 Win=4091 Len=0
54	31.890542	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [ACK] Seq=38 Ack=89 Win=4041 Len=0
55	31.906595	192.168.1.2	192.168.1.1	TCP	60 27717 → 23 [FIN, PSH, ACK] Seq=38 Ack=89 Win=4041 Len=0
56	31.922303	192.168.1.1	192.168.1.2	TCP	60 23 → 27717 [ACK] Seq=89 Ack=39 Win=4091 Len=0
57	33.343385	c4:01:0e:50:00:00	CDP/VTP/DTP/PAgP/UD...	CDP	364 Device ID: SERVER.test.local Port ID: FastEthernet0/0
58	33.373314	c4:02:2d:64:00:00	CDP/VTP/DTP/PAgP/UD...	CDP	353 Device ID: CLIENT Port ID: FastEthernet0/0

# UDP



# Где нужен UDP



# Недостатки UDP

## Ключевые недостатки UDP:

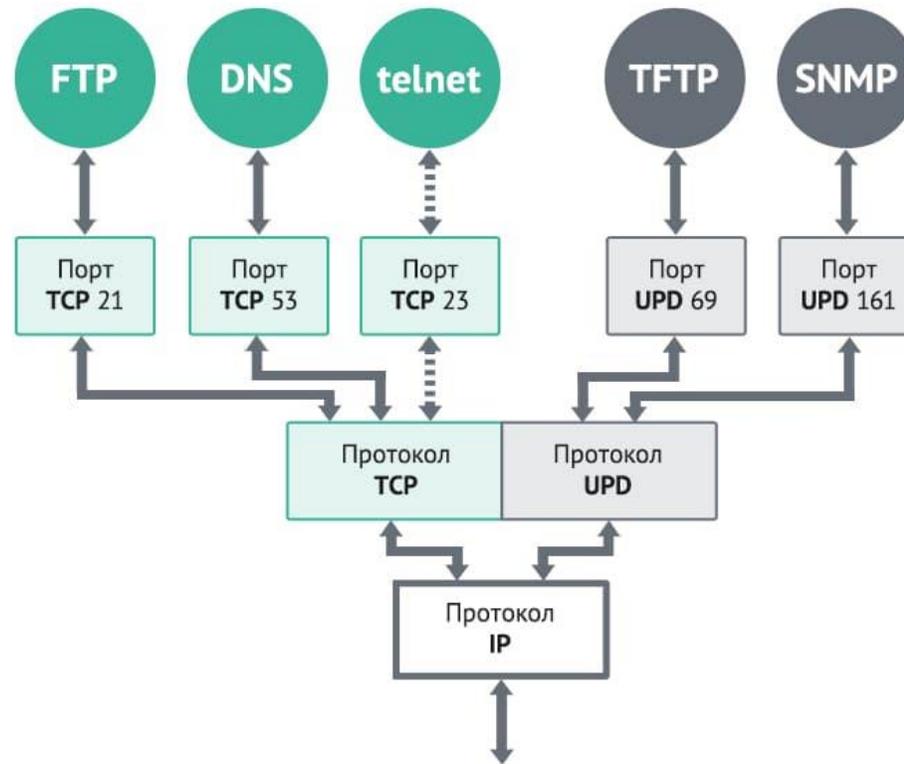
- 1. Нет гарантии доставки. Пакеты могут потеряться, и протокол не предпримет попыток их повторной отправки.
- 2. Нет контроля порядка. Пакеты могут приходить к получателю в произвольном порядке, а не в том, в котором были отправлены.
- 3. Нет контроля перегрузки. Отправитель может завалить получателя пакетами, вызывая перегрузку сети и потерю данных, так как механизма "притормозить" не существует.
- 4. Нет установления соединения. Это значит, что нет встроенной защиты от подмены отправителя (спуфинга) и выше риск получения нежелательного трафика.
- 5. Надежность ложится на приложение. Все проблемы, которые TCP решает на уровне протокола (упорядочивание, целостность потока, управление потоком), в случае с UDP должны реализовывать сами программисты в коде своего приложения, если это необходимо.

# Заголовок UDP

## Заголовок UDP

16 битный порт источника	16 битный порт получателя
16 битный UDP заголовок	16 битная проверка суммы UDP
Данные	

# Служебные порты UDP



# Практическая часть(UDP)

The screenshot displays the GNS3 network simulator interface. On the left, the 'All devices' panel lists various network components such as ATM switch, c3745, Cloud, Ethernet hub, Ethernet switch, Frame Relay switch, NAT, and VPCS. The main workspace shows a topology with two routers, R1 and R2, connected by a single link. The right side of the interface features a 'Topology Summary' panel showing the console for R1 (telnnet localhost:5000) and R2 (telnnet localhost:5001). Below it, a 'Servers Summary' panel shows the system resources for the host 'whitakker', including CPU usage at 29.5% and RAM usage at 51.2%. At the bottom, a console window displays the GNS3 management console text: 'GNS3 management console. Running GNS3 version 2.2.45 on Windows (64-bit) with Python 3.10.11 Qt 5.15.2 and PyQt 5.15.10. Copyright (c) 2006-2025 GNS3 Technologies. Use Help -> GNS3 Doctor to detect common issues. =>'

# Настройка сервера и клиента

```
SERVER#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SERVER(config)#ntp master 1
SERVER(config)#ntp source fa0/0
```

```
CLIENT#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CLIENT(config)#ntp server 192.168.1.1
CLIENT(config)#ntp source fa0/0
CLIENT(config)#exit
CLIENT#wr mem
*Mar  1 00:07:50.015: %SYS-5-CONFIG_I: Configured from console by console
CLIENT#wr mem
Building configuration...
[OK]
CLIENT#show ntp status
Clock is synchronized, stratum 2, reference is 192.168.1.1
nominal freq is 250.0000 Hz, actual freq is 249.9999 Hz, precision is 2**18
reference time is C02944DF.08423A3D (00:07:59.032 UTC Fri Mar 1 2002)
clock offset is 17.9669 msec, root delay is 28.05 msec
root dispersion is 27.34 msec, peer dispersion is 9.35 msec
```

# Перехват и анализ трафика

8	46.438762	192.168.1.2	192.168.1.1	NTP	90 NTP Version 3, client
9	46.454659	192.168.1.1	192.168.1.2	NTP	90 NTP Version 3, server

```
> Frame 8: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface -, id 0
> Ethernet II, Src: c4:02:2d:64:00:00 (c4:02:2d:64:00:00), Dst: c4:01:0e:50:00:00 (c4:01:0e:50:00:00)
< Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 76
    Identification: 0x0000 (0)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 255
    Protocol: UDP (17)
    Header Checksum: 0x378d [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.2
    Destination Address: 192.168.1.1
  > User Datagram Protocol, Src Port: 123, Dst Port: 123
  > Network Time Protocol (NTP Version 3, client)
```

---

```
0000  c4 01 0e 50 00 00 c4 02 2d 64 00 00 08 00 45 c0  ...P... -d...E.
0010  00 4c 00 00 00 00 ff 11 37 8d c0 a8 01 02 c0 a8  .L..... 7.....
0020  01 01 00 7b 00 7b 00 38 50 1f 1b 02 06 ee 00 00  ...{...8 P.....
0030  06 30 00 00 06 ef c0 a8 01 01 c0 29 44 e5 08 60  .0.....)D..`
0040  59 74 c0 29 44 e5 0b a4 e7 cc c0 29 44 e5 08 60  Yt.)D... ..)D...
0050  59 74 c0 29 45 24 ff a7 6f 1f  Yt.)E$.. o.
```

# TCP vs UDP

Критерий	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Основная концепция	Надёжный, ориентированный на соединение протокол. Гарантирует доставку, целостность и порядок данных.	Быстрый, не требующий соединения протокол. Работает по принципу «отправил и забыл» без гарантий.
Установление соединения	Требуется (3-этапное рукопожатие: SYN, SYN-ACK, ACK). Создаёт виртуальный канал.	Не требуется. Каждый пакет (датаграмма) отправляется независимо.
Надёжность	Высокая. Использует подтверждения (ACK), повторные передачи (Retransmission) и контрольные суммы. Гарантирует, что данные дойдут целыми и в правильном порядке.	Низкая/Отсутствует. Нет подтверждений, повторных передач или гарантий доставки. Пакеты могут быть потеряны, продублированы или прийти не по порядку.
Упорядочивание	Да. Использует порядковые номера (Sequence Numbers), чтобы собирать байты на стороне получателя в исходном порядке.	Нет. Не имеет встроенного механизма упорядочивания. Приложение должно само обрабатывать порядок, если это важно.
Управление потоком (Flow Control)	Да. Использует механизм скользящего окна (Sliding Window). Получатель указывает размер окна (Window Size), чтобы отправитель не перегружал его буфер.	Нет. Отправитель может посылать данные с любой скоростью, рискуя переполнить буфер получателя.

Управление перегрузкой (Congestion Control)	Да. Сложные алгоритмы (TCP Reno, CUBIC) для динамического снижения скорости при обнаружении перегрузки в сети.	Нет. Может усугублять перегрузку, так как не снижает скорость передачи при потере пакетов.
Заголовок	Большой и сложный (от 20 до 60 байт). Содержит множество полей для управления соединением (номера, флаги, окно и т.д.).	Маленький и фиксированный (8 байт). Только порты, длина и контрольная сумма. Низкие накладные расходы.
Скорость и задержка	Относительно медленнее, выше задержка (латентность). Затраты на установление соединения, подтверждения, упорядочивание и управление.	Очень быстрый, минимальная задержка. Отсутствие накладных расходов на управление делает его идеальным для реального времени.
Передача данных	Поток байтов (byte-stream). Не сохраняет границы сообщений. Приложение само должно их определять.	Датаграммы (message-oriented). Сохраняет границы сообщений. Каждая отправленная датаграмма прибывает как отдельное сообщение.
Тип связи	Только точка-точка (unicast).	Поддерживает unicast, multicast (многоадресную) и broadcast (широковещательную) рассылку.
Контрольная сумма	Обязательна. Проверяет целостность заголовка и данных.	Необязательна (может быть равна 0).

# Заключение

