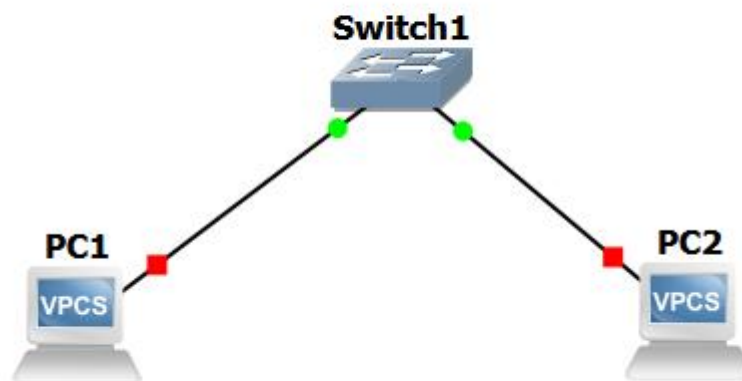


Лабораторная работа №1

«Освоение инструментария для выполнения работ, построение простой сети»

Для начала была построена простая сеть, состоящая из одного коммутатора и двух компьютеров:



PC1 console	
ip 192.168.0.10/24	
PC2 console	
ip 192.168.0.10/24	
<pre>PC1> ip 192.168.0.10/24 Checking for duplicate address... PC1 : 192.168.0.10 255.255.255.0 PC1> █</pre>	<pre>VPCS is free software, distributed under the terms of the "BSD" licence. Source code and license can be found at vpcs.sf.net. For more information, please visit wiki.freecode.com.cn. Press '?' to get help. Executing the startup file PC2> ip 192.168.0.20/24 Checking for duplicate address... PC2 : 192.168.0.20 255.255.255.0 PC2> █</pre>

После запуска симуляции была выполнена команда ping на PC1.

PC1 console
ping 192.168.0.20

```
PC1> ping 192.168.0.20

84 bytes from 192.168.0.20 icmp_seq=1 ttl=64 time=0.221 ms
84 bytes from 192.168.0.20 icmp_seq=2 ttl=64 time=0.245 ms
84 bytes from 192.168.0.20 icmp_seq=3 ttl=64 time=0.291 ms
84 bytes from 192.168.0.20 icmp_seq=4 ttl=64 time=0.246 ms
84 bytes from 192.168.0.20 icmp_seq=5 ttl=64 time=0.277 ms
```

Вследствие выполнения данной команды был получен ответ, свидетельствующий об успешном установлении соединения между PC1 и PC2, ответ представлен на скриншоте выше.

Стоит отметить, что на момент выполнения данной команды трафик на всех линках уже перехватывался с помощью Wireshark.

На первом скриншоте отмечен трафик, перехваченный с Switch1 ⇔ PC1, на втором – с Switch1 ⇔ PC2:

arp					
No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	Private_66:68:01	Broadcast	ARP	64 Who has 192.168.0.20? Tell 192.168.0.10
2	0.000217	Private_66:68:00	Private_66:68:01	ARP	64 192.168.0.20 is at 00:50:79:66:68:00

arp					
No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	Private_66:68:01	Broadcast	ARP	64 Who has 192.168.0.20? Tell 192.168.0.10
2	0.000170	Private_66:68:00	Private_66:68:01	ARP	64 192.168.0.20 is at 00:50:79:66:68:00

Можно заметить, что было перехвачено по два пакета. Первый пакет является запросом PC1 на получение MAC-адреса устройства с IPv4 192.168.0.20. Подробнее данный пакет рассмотрен на двух скриншотах ниже:

```

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....1. .... = IG bit: Group address (multicast/broadcast)
  Source: Private_66:68:01 (00:50:79:66:68:01)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
  [Stream index: 0]
  Padding: 00000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: Private_66:68:01 (00:50:79:66:68:01)
  Sender IP address: 192.168.0.10
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
  Target IP address: 192.168.0.20

0000  ff ff ff ff ff ff 00 50 79 66 68 01 08 06 00 01  .....P yfh....
0010  08 00 06 04 00 01 00 50 79 66 68 01 c0 a8 0a  .....P yfh....
0020  ff ff ff ff ff ff c0 a8 00 14 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

0000	ff ff ff ff ff ff	00 50 79 66 68 01	08 06 00 01P yfh.....
0010	08 00 06 04 00 01	00 50 79 66 68 01	c0 a8 00 0aP yfh.....
0020	ff ff ff ff ff ff	c0 a8 00 14	00 00 00 00 00 00
0030	00 00 00 00 00 00	00 00 00 00 00 00	00 00 00 00

ARP пакет состоит из 64 байт.

Первые 6 байт являются MAC-адресом назначения (в данном случае мы запрашиваем все устройства сети).

Следующие 6 байт представляют из себя MAC-адрес отправителя запроса.

Следующие 2 байта устанавливают тип данных (в данном случае ARP пакет).

Затем идут 2 байта, обозначающие тип сети, в нашем случае: 1 – Ethernet.

Следующие 2 байта – тип протокола (08 00 – IPv4).

Затем идут 2 байта длин адресов: 6 байт для MAC (06) и 4 байта для IP (04).

После этого указаны 2 байта типа операции (00 01 – ARP запрос).

Затем аналогично указываются 6 байт MAC-адреса отправителя и 4 байта IP-адреса (PC1).

После этого указан неизвестный MAC-адрес (6 байт), который мы хотим узнать, и IP-адрес устройства (4 байта), чей MAC-адрес мы запрашиваем (PC2).

Для того, чтобы размер пакета составлял 64 байта, в конце просто доставляются нули.

После данного запроса в ответ с PC2 был отправлен ARP пакет на PC1, который позволил первому компьютеру узнать MAC-адрес второго. На скриншоте ниже представлена более подробная информация:

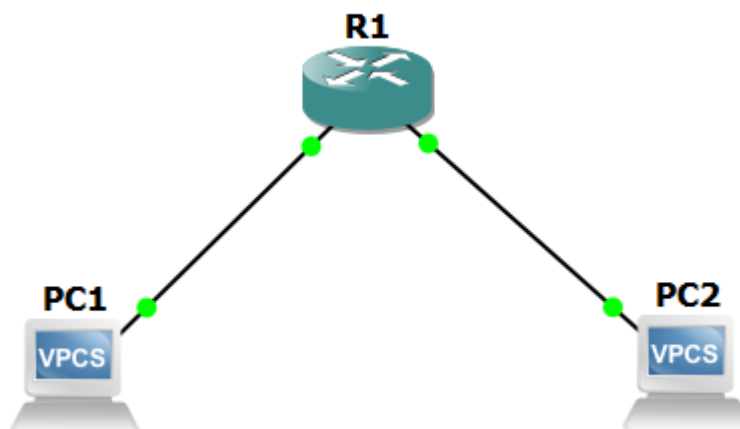
<pre> Frame 2: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0 Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01) Destination: Private_66:68:01 (00:50:79:66:68:01) Source: Private_66:68:00 (00:50:79:66:68:00) Type: ARP (0x0806) [Stream index: 1] Padding: 00000000000000000000000000000000 Frame check sequence: 0x00000000 [unverified] [FCS Status: Unverified] Address Resolution Protocol (reply) Hardware type: Ethernet (1) Protocol type: IPv4 (0x0800) Hardware size: 6 Protocol size: 4 Opcode: reply (2) Sender MAC address: Private_66:68:00 (00:50:79:66:68:00) Sender IP address: 192.168.0.20 Target MAC address: Private_66:68:01 (00:50:79:66:68:01) Target IP address: 192.168.0.10 </pre>				
0000	00 50 79 66 68 01	00 50 79 66 68 00	08 06 00 01	Pyfh P yfh.....
0010	08 00 06 04 00 02	00 50 79 66 68 00	c0 a8 00 14P yfh.....
0020	00 50 79 66 68 01	c0 a8 00 0a	00 00 00 00 00 00	Pyfh.....
0030	00 00 00 00 00 00	00 00 00 00 00 00	00 00 00 00

Вот пакет, полученный PC1 в свою очередь:

```
▶ Frame 2: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
▼ Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01)
  ▼ Destination: Private_66:68:01 (00:50:79:66:68:01)
    .... ..0. .... = LG bit: Globally unique address (factory default)
    .... ..0. .... = IG bit: Individual address (unicast)
  ▼ Source: Private_66:68:00 (00:50:79:66:68:00)
    .... ..0. .... = LG bit: Globally unique address (factory default)
    .... ..0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
  [Stream index: 1]
  Padding: 00000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Sender IP address: 192.168.0.20
  Target MAC address: Private_66:68:01 (00:50:79:66:68:01)
  Target IP address: 192.168.0.10

0000  00 50 79 66 68 01 00 50  79 66 68 00 08 06 00 01  .Pyfh..P yfh....
0010  08 00 06 04 00 02 00 50  79 66 68 00 c0 a8 00 14  .....P yfh....
0020  00 50 79 66 68 01 c0 a8  00 0a 00 00 00 00 00 00  .Pyfh.....
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
```

После теста, проведенного выше, коммутатор в сети был заменен на маршрутизатор:



Сначала был настроен маршрутизатор R1. Для этого сначала с помощью команды «config» был запущен конфигурационный режим, с помощью «interface ...» был совершен переход в режим настройки соответствующего интерфейса. С помощью «ip address ...» установили соответствующему порту адрес, а благодаря «no shutdown» активируем данный интерфейс.

R1 console
config interface FastEthernet0/0 ip address 192.168.1.1 255.255.255.0 no shutdown

```
exit

interface FastEthernet1/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
```

На компьютерах же были установлены соответствующие IP-адреса и шлюзы:

PC1 console

```
ip 192.168.1.10/24 192.168.1.1
```

```
PC1> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.10 255.255.255.0 gateway 192.168.1.1
```

PC2 console

```
ip 192.168.2.10/24 192.168.2.1
```

```
PC2> ip 192.168.2.10/24 192.168.2.1
Checking for duplicate address...
PC2 : 192.168.2.10 255.255.255.0 gateway 192.168.2.1

PC2> █
```

После настройки было проверено соединение между PC1 и PC2 посредством команды «ping»:

PC1 console

```
ping 192.168.2.10
```

Стоит отметить, что на скриншоте ниже представлен результат уже второго выполнения данной команды:

```
PC1> ping 192.168.2.10

84 bytes from 192.168.2.10 icmp_seq=1 ttl=63 time=14.904 ms
84 bytes from 192.168.2.10 icmp_seq=2 ttl=63 time=47.702 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=63 time=18.473 ms
84 bytes from 192.168.2.10 icmp_seq=4 ttl=63 time=17.766 ms
84 bytes from 192.168.2.10 icmp_seq=5 ttl=63 time=18.195 ms
```

Ниже представлены скриншоты перехваченного трафика.

PC1 ⇔ R1:

arp icmp					
No.	Time	Source	Destination	Protocol	Length Info
2	0.000039	cc:01:58:9b:00:00	Broadcast	ARP	60 Gratuitous ARP for 192.168.1.1 (Reply)
3	0.000044	cc:01:58:9b:00:00	Broadcast	ARP	60 Gratuitous ARP for 192.168.1.1 (Reply)
10	44.935707	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.10 (Request)
12	45.936343	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.10 (Request)
13	46.937242	Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.1.10 (Request)
19	90.167729	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1388, seq=1/256, ttl=64 (no response found!)
20	92.168845	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1588, seq=2/512, ttl=64 (no response found!)
21	92.186624	cc:01:58:9b:00:00	Broadcast	ARP	60 Who has 192.168.1.10? Tell 192.168.1.1
22	92.186708	Private_66:68:01	cc:01:58:9b:00:00	ARP	60 192.168.1.10 is at 00:50:79:66:68:01
23	94.169815	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1788, seq=3/768, ttl=64 (reply in 24)
24	94.189930	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1788, seq=3/768, ttl=63 (request in 23)
25	95.191014	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1888, seq=4/1024, ttl=64 (reply in 26)
26	95.208134	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1888, seq=4/1024, ttl=63 (request in 25)
27	96.209143	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1988, seq=5/1280, ttl=64 (reply in 28)
28	96.224664	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1988, seq=5/1280, ttl=63 (request in 27)
30	99.983701	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1d88, seq=1/256, ttl=64 (reply in 31)
31	99.994086	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1d88, seq=1/256, ttl=63 (request in 30)
32	100.994378	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1e88, seq=2/512, ttl=64 (reply in 33)
33	101.012221	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1e88, seq=2/512, ttl=63 (request in 32)
34	102.012472	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1f88, seq=3/768, ttl=64 (reply in 35)
35	102.030830	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1f88, seq=3/768, ttl=63 (request in 34)
36	103.031617	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x2088, seq=4/1024, ttl=64 (reply in 37)
37	103.049405	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x2088, seq=4/1024, ttl=63 (request in 36)
38	104.049752	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x2188, seq=5/1280, ttl=64 (reply in 39)
39	104.067835	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x2188, seq=5/1280, ttl=63 (request in 38)

PC2 ⇔ R1:

arp icmp					
No.	Time	Source	Destination	Protocol	Length Info
2	0.010061	cc:01:58:9b:00:10	Broadcast	ARP	60 Gratuitous ARP for 192.168.2.1 (Reply)
3	0.010085	cc:01:58:9b:00:10	Broadcast	ARP	60 Gratuitous ARP for 192.168.2.1 (Reply)
9	38.180935	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.2.10 (Request)
10	39.181788	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.2.10 (Request)
11	40.181871	Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.2.10 (Request)
15	64.641801	cc:01:58:9b:00:10	Broadcast	ARP	60 Who has 192.168.2.10? Tell 192.168.2.1
16	64.641904	Private_66:68:00	cc:01:58:9b:00:10	ARP	60 192.168.2.10 is at 00:50:79:66:68:00
18	66.645783	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1588, seq=2/512, ttl=63 (reply in 19)
19	66.645887	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1588, seq=2/512, ttl=64 (request in 18)
20	68.649129	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1788, seq=3/768, ttl=63 (reply in 21)
21	68.649279	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1788, seq=3/768, ttl=64 (request in 20)
22	69.667238	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1888, seq=4/1024, ttl=63 (reply in 23)
23	69.667376	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1888, seq=4/1024, ttl=64 (request in 22)
24	70.683829	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1988, seq=5/1280, ttl=63 (reply in 25)
25	70.683970	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1988, seq=5/1280, ttl=64 (request in 24)
27	74.453218	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1d88, seq=1/256, ttl=63 (reply in 28)
28	74.453394	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1d88, seq=1/256, ttl=64 (request in 27)
29	75.471338	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1e88, seq=2/512, ttl=63 (reply in 30)
30	75.471499	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1e88, seq=2/512, ttl=64 (request in 29)
31	76.489948	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x1f88, seq=3/768, ttl=63 (reply in 32)
32	76.490081	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x1f88, seq=3/768, ttl=64 (request in 31)
33	77.508529	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x2088, seq=4/1024, ttl=63 (reply in 34)
34	77.508634	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x2088, seq=4/1024, ttl=64 (request in 33)
35	78.526995	192.168.1.10	192.168.2.10	ICMP	98 Echo (ping) request id=0x2188, seq=5/1280, ttl=63 (reply in 36)
36	78.527117	192.168.2.10	192.168.1.10	ICMP	98 Echo (ping) reply id=0x2188, seq=5/1280, ttl=64 (request in 35)

Здесь:

ARP пакеты 2 и 3 отправляются для уведомления всей сети о том, что адреса 192.168.1.1 и 192.168.2.1 принадлежат маршрутизатору.

ARP пакеты 10, 11, 12, 13 используются для проверки конфликтов на этапе установки адресов для PC1 (192.168.1.10) и PC2 (192.168.2.10).

ARP пакеты 15, 16, 21, 22 аналогичны пакетам, разобранным в схеме с коммутатором. Они отправляются на этапе первого «ping».

Остальные ICMP пакеты являются ping запросами между PC1 и PC2. Здесь можно отметить пакеты 19 и 20, в которых ответ не был получен по причине того, что PC1 ещё не был известен MAC-адрес PC2.

Далее для примера был рассмотрен отдельно взятый ICMP пакет:

```

> Frame 23: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: cc:01:58:9b:00:00 (cc:01:58:9b:00:00)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.2.10
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x8815 (34837)
  > 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: ICMP (1)
  Header Checksum: 0x6e2f [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.1.10
  Destination Address: 192.168.2.10
  [Stream index: 0]
> Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x0881 [correct]
  [Checksum Status: Good]
  Identifier (BE): 6024 (0x1788)
  Identifier (LE): 34839 (0x8817)
  Sequence Number (BE): 3 (0x0003)
  Sequence Number (LE): 768 (0x0300)
  [Response frame: 24]
> Data (56 bytes)
  Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f
  [Length: 56]

```

0000	cc 01 58 9b 00 00	00 50 79 66 68 01	08 00 45 00	..X...P yfh...E.
0010	00 54 88 15 00 00	40 01 6e 2f c0 a8 01 0a	c0 a8	.T...@ n/.....
0020	02 0a 08 00 08 81	17 88 00 03 08 09 0a 0b	0c 0d
0030	0e 0f 10 11 12 13 14 15	16 17 18 19 1a 1b 1c 1d	
0040	1e 1f 20 21 22 23 24 25	26 27 28 29 2a 2b 2c 2d		.. !"# \$% &'()*+,-
0050	2e 2f 30 31 32 33 34 35	36 37 38 39 3a 3b 3c 3d		./012345 6789:;<=
0060	3e 3f			>?

ICMP пакет, в свою очередь, состоит из 98 байт.

Сначала идет 14 байт Ethernet заголовка:

6 байт MAC-адреса получателя;

6 байт MAC-адреса отправителя;

4 байта типа протокола (08 00 – IPv4);

Затем идет 20 байт IP заголовка:

1 байт: 4 – тип протокола (IPv4), 5 – длина заголовка 20 байт;

1 байт приоритета (00 – обычный приоритет);

2 байта длины пакета (00 54 = 84 байта, включая данные);

2 байта с идентификатором пакета (88 15);

1 байт флагов (00);

1 байт, указывающий на фрагментирование пакета (00 – пакет целый);

1 байт ttl (время жизни пакета в сети) (40 = 64 прохода);

1 байт протокола (01 – ICMP);

2 байта контрольной суммы заголовка (be 2f);

4 байта IP-адреса отправителя;

4 байта IP-адреса получателя;

После этого идет ICMP заголовок (8 байт):

1 байт типа (08 – Echo Request);

1 байт кода (00 для ping);

2 байта контрольной суммы ICMP (08 81);

2 байта идентификатора (17 88);

2 байта номера последовательности (00 03);

В конце идет 48 байт, которые являются полезной нагрузкой.