МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

Сети и телекоммуникации

Лабораторная работа №2

Расчет контрольной суммы заголовка протокола IP

ПРОВЕРИЛ:

_____ Гай В.Е.

СТУДЕНТ:

Козменкова Е.П.
18 В-2

Нижний Новгород
2021 г.

Цель:

Изучить формат заголовка пакета IP и на примере разобрать механизм вычисления 16-битовой контрольной суммы, использующейся для обнаружения ошибок в заголовке протокола IP.

Ход работы:

Для подготовки к работе попробую разобрать одну из задач в методичке:

№ вар.	Пакет IPv4
1	0000 00 13 8f 13 b7 f8 d8 50 e6 a2 37 61 08 00 45 00 0010 00 34 6e 86 40 00 40 06 00 00 ac 10 64 29 40 e9 0020 a2 5f d7 82 01 bb 78 ea 6c bb 3c 25 ac 7a 80 10 0030 00 ed 69 cf 00 00 01 01 08 0a e4 51 97 c8 17 1b 0040 dd c5

В задании написано, что пакет начинается с заголовка Ethernet. Посмотрим, как выглядит этот заголовок:

6 байт	6 байт	2 байта	46-1500 байт	4 байта
DA	SA	T	Данные	FCS

Где DA – MAC-адрес узла назначения, SA – MAC-адрес узла отправителя, T – код протокола (08 00 - IP), FCS – контрольная сумма.

Заголовок IP, в свою очередь, выглядит так:

Теперь рассмотрю пакет из задания:

№ вар.	Пакет IPv4
1	0000 00 13 8f 13 b7 f8 d8 50 e6 a2 37 61 08 00 45 00 0010 00 34 6e 86 40 00 40 06 00 00 ac 10 64 29 40 e9 0020 a2 5f d7 82 01 bb 78 ea 6c bb 3c 25 ac 7a 80 10 0030 00 ed 69 cf 00 00 01 01 08 0a e4 51 97 c8 17 1b 0040 dd c5

00 13 8f 13 b7 f8 – MAC-адрес получателя;
d8 50 e6 a2 37 61 – MAC-адрес отправителя;
08 00 – код протокола (IP);
С 45 начинается поле данных – заголовок IP-пакета:
4 – номер версии протокола IP (IPv4);
5 – длина заголовка (пять 32-битных слов);
00 – тип сервиса: приоритет пакета (первые три бита) - 0, критерии выбора маршрута (задержка, пропускная способность и надежность) – так же 0;
00 34 – общая длина IP-пакета;
6e 86 – идентификатор пакета;
40 00 – флаги и смещение фрагмента: первые три бита (флаги) – 0 1 0, где 2-й бит – флаг DF, который запрещает маршрутизатору фрагментировать пакет; так как пакет не фрагментируется, поле смещения – 0;
40 – время жизни пакета (в секундах)
06 – протокол верхнего уровня (TCP)
00 00 – контрольная сумма заголовка, которую мне предстоит посчитать
ac 10 64 29 – IP-адрес источника
40 e9 a2 5f – IP-адрес назначения
Следующие 32 байта – другие данные кадра

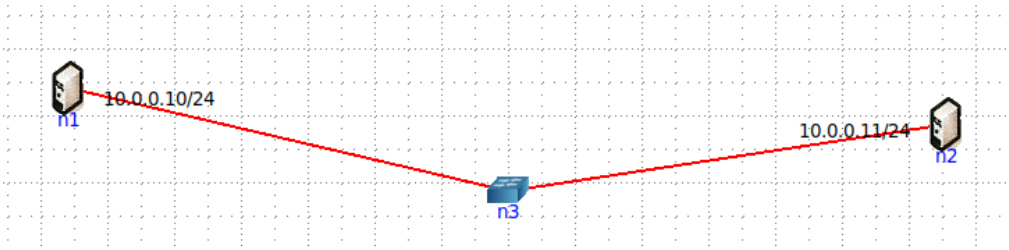
Для подсчета контрольной сумма заголовка IP-пакета, разобьем его на слова по 16 бит:

4500	0034
6E86	4000
4006	0000
AC10	6429
40E9	A25F

Просуммируем:
$(4500)_{16} + (0034)_{16} + (6E86)_{16} + (4000)_{16} + (4006)_{16} + (0000)_{16} + (AC10)_{16} + (6429)_{16} + (40E9)_{16} + (A25F)_{16} = (32741)_{16}$
Результат сложения превышает 16 разрядов, разобью его на два слова и посчитаю еще раз:
$(0003)_{16} + (2741)_{16} = (2744)_{16}$

Найду контрольную сумму:
$CS_{IP} = (FFFF)_{16} – (2744)_{16} = (D8BB)_{16}$

Теперь можно перейти к рассмотрению реальных IP пакетов.
Сеть:



Запущу ping с компьютера 10.0.0.11:

```
root@n2:/tmp/pycore.38007/n2.conf# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.142 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.092 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.125 ms
^C
--- 10.0.0.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2034ms
rtt min/avg/max/mdev = 0.092/0.119/0.142/0.024 ms
```

Wireshark на компьютере 10.0.0.10 (перехвачу и прочитаю пакет ICMP):

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::90aa:27ff:feb…	ff02::2	ICMPv6	70	Router Solicitation from 92:aa:27:b3:b7:ee
2	11.245283326	00:00:00_aa:00:01	Broadcast	ARP	42	Who has 10.0.0.10? Tell 10.0.0.11
3	11.245320294	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	10.0.0.10 is at 00:00:00:aa:00:00
4	11.245336243	10.0.0.11	10.0.0.10	ICMP	98	Echo (ping) request id=0x0023, seq=1/256, ttl=64 (rep
5	11.245360089	10.0.0.10	10.0.0.11	ICMP	98	Echo (ping) reply id=0x0023, seq=1/256, ttl=64 (req
6	12.256530714	10.0.0.11	10.0.0.10	ICMP	98	Echo (ping) request id=0x0023, seq=2/512, ttl=64 (rep
7	12.256559306	10.0.0.10	10.0.0.11	ICMP	98	Echo (ping) reply id=0x0023, seq=2/512, ttl=64 (req
8	12.288358804	fe80::200:ff:feaa:0	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:00
9	13.187236486	fe80::10d0:dfff:fe7…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" quest
10	13.280124414	10.0.0.11	10.0.0.10	ICMP	98	Echo (ping) request id=0x0023, seq=3/768, ttl=64 (rep
11	13.280156716	10.0.0.10	10.0.0.11	ICMP	98	Echo (ping) reply id=0x0023, seq=3/768, ttl=64 (req
12	16.384098750	00:00:00_aa:00:00	00:00:00_aa:00:01	ARP	42	Who has 10.0.0.11? Tell 10.0.0.10
13	16.384173007	00:00:00_aa:00:01	00:00:00_aa:00:00	ARP	42	10.0.0.11 is at 00:00:00:aa:00:01
14	18.097937992	fe80::90aa:27ff:feb…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" quest
15	26.624153608	fe80::10d0:dfff:fe7…	ff02::2	ICMPv6	70	Router Solicitation from 1e:e4:a1:ef:05:e7
16	26.624230325	fe80::200:ff:feaa:1	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:01

```
▸ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_aa:00:01 (00:00:00:aa:00:01), Dst: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
▸ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 10.0.0.10
▸ Internet Control Message Protocol
```

```
0000  00 00 00 aa 00 00 00 00  00 aa 00 01 08 00 45 00   ........ ......E·
0010  00 54 5d db 40 00 40 01  c8 b9 0a 00 00 0b 0a 00   ·T]·@·@· ........
0020  00 0a 08 00 db 52 00 23  00 01 74 d1 5d 60 00 00   .....R·# ··t·]`··
0030  00 00 8b 84 00 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

Кадр Ethernet:

```
0000  00 00 00 aa 00 00 00 00  00 aa 00 01 08 00 45 00   ........ ......E·
0010  00 54 5d db 40 00 40 01  c8 b9 0a 00 00 0b 0a 00   ·T]·@·@· ........
0020  00 0a 08 00 db 52 00 23  00 01 74 d1 5d 60 00 00   .....R·# ··t·]`··
0030  00 00 8b 84 00 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

00 00 00 aa 00 00 – MAC-адрес получателя;

00 00 00 aa 00 01 – MAC-адрес отправителя;

08 00 – код протокола (IP);

Заголовок IP-пакета:

```
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.0.11, Dst: 10.0.0.10
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Trans
    Total Length: 84
    Identification: 0x5ddb (24027)
  Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xc8b9 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.0.11
    Destination: 10.0.0.10
```

```
0000  00 00 00 aa 00 00 00 00  00 aa 00 01 08 00 45 00   ........ ......E·
0010  00 54 5d db 40 00 40 01  c8 b9 0a 00 00 0b 0a 00   ·T]·@·@· ........
0020  00 0a 08 00 db 52 00 23  00 01 74 d1 5d 60 00 00   ·····R·# ··t·]`··
0030  00 00 8b 84 00 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

4 – номер версии протокола IP (IPv4);

5 – длина заголовка (пять 32-битных слов);

00 – тип сервиса: приоритет пакета (первые три бита) - 0, критерии выбора маршрута (задержка, пропускная способность и надежность) – так же 0;

00 54 – общая длина IP-пакета;

5d db – идентификатор пакета;

40 00– флаги и смещение фрагмента: первые три бита (флаги) – 0 1 0, где 2-й бит – флаг DF, который запрещает маршрутизатору фрагментировать пакет; так как пакет не фрагментируется, поле смещения – 0;

40 – время жизни пакета (в секундах – 64 с)

01 – протокол верхнего уровня (ICMP)

c8 b9– контрольная сумма заголовка, с которой буду сравнивать посчитанную

0a 00 00 0b – IP-адрес источника

0a 00 00 0a – IP-адрес назначения

Дальше идут параметры ICMP протокола:

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xdb52 [correct]
    [Checksum Status: Good]
    Identifier (BE): 35 (0x0023)
    Identifier (LE): 8960 (0x2300)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Response frame: 5]
    Timestamp from icmp data: Mar 26, 2021 15:20:04.000000000 MSK
    [Timestamp from icmp data (relative): 0.034045606 seconds]
  Data (48 bytes)
      Data: 8b840000000000001011121314151617181 91a1b1c1d1e1f...
      [Length: 48]
```

```
0000  00 00 00 aa 00 00 00 00  00 aa 00 01 08 00 45 00   ........ ......E·
0010  00 54 5d db 40 00 40 01  c8 b9 0a 00 00 0b 0a 00   ·T]·@·@· ........
0020  00 0a 08 00 db 52 00 23  00 01 74 d1 5d 60 00 00   ·····R·# ··t·]`··
0030  00 00 8b 84 00 00 00 00  00 00 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

Посчитаю контрольную сумму:

4500	0054
5ddb	4000
4001	0000
0a00	000b
0a00	000a

Просуммируем:

$(4500)_{16} + (0054)_{16} + (5ddb)_{16} + (4000)_{16} + (4001)_{16} + (0000)_{16} + (0a00)_{16} + (000b)_{16} + (0a00)_{16} + (000a)_{16} = (13745)_{16}$

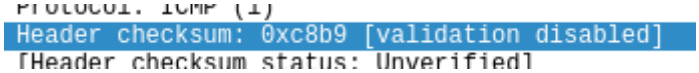
Результат сложения превышает 16 разрядов, разобью его на два слова и посчитаю еще раз:

$(0001)_{16} + (3745)_{16} = (3746)_{16}$

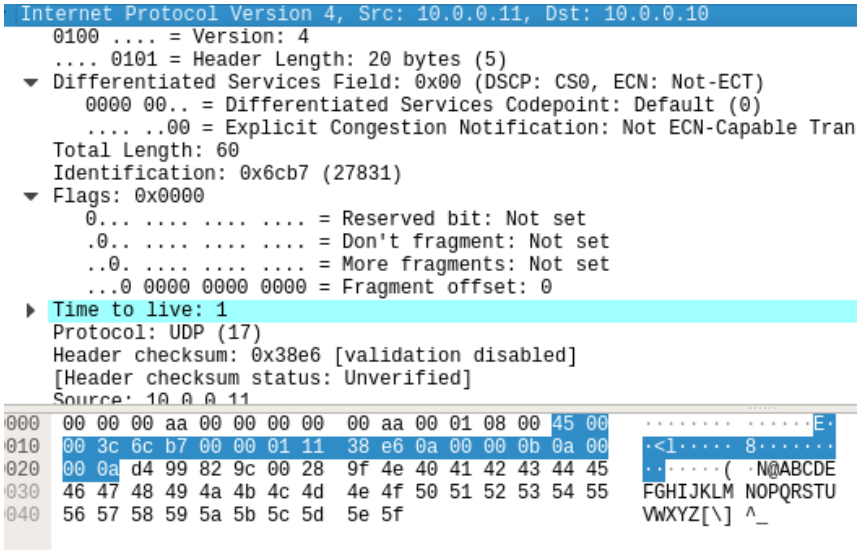
Найду контрольную сумму:

$CS_{IP} = (FFFF)_{16} – (3746)_{16} = (C8B9)_{16}$

Результат совпадает с контрольной суммой заголовка:



Теперь запущу traceroute для получения UDP пакета:



Отличия от предыдущего перехваченного пакета:

00 3c – общая длина IP-пакета;

6c b7 – идентификатор пакета;

00 00– флаги и смещение фрагмента: первые три бита (флаги) – 0 0 0; поле смещения – 0;

01 – время жизни пакета (в секундах – 1 с)

11 – протокол верхнего уровня (UDP - 17)

38 e6– контрольная сумма заголовка, с которой буду сравнивать посчитанную

0a 00 00 0b – IP-адрес источника

0a 00 00 0a – IP-адрес назначения

Посчитаю контрольную сумму:

4500	003c
6cb7	0000
0111	0000
0a00	000b
0a00	000a

Просуммируем:

$(4500)_{16} + (003c)_{16} + (6cb7)_{16} + (0000)_{16} + (0111)_{16} + (0000)_{16} + (0a00)_{16} + (000b)_{16} + (0a00)_{16} + (000a)_{16} = (C719)_{16}$

Найду контрольную сумму:

$CS_{IP} = (FFFF)_{16} – (C719)_{16} = (38E6)_{16}$

Результат совпадает с контрольной суммой.