

Задача “Разбор сетевых данных”

Дано:
Файл с пакетами трафика, захваченного с сетевой карты (стек протоколов вымышленный). Описание сетевой модели и протоколов представлено ниже.

Требуется:
Создать программу на C++, собирающую по данному файлу следующую статистику:

- 1. Количество пакетов NETWORK V1
- 2. Количество пакетов NETWORK V2
- 3. Количество уникальных адресов устройств NETWORK V1
- 4. Количество уникальных адресов устройств NETWORK V2
- 5. Количество пакетов TRANSPORT V1
- 6. Количество пакетов TRANSPORT V2
- 7. Количество пакетов TRANSPORT V1 с ошибочной контрольной суммой
- 8. Количество пакетов TRANSPORT V2 с ошибочной контрольной суммой
- 9. Количество уникальных портов TRANSPORT V1
- 10. Количество уникальных портов TRANSPORT V2
- 11. Количество сессий TRANSPORT V2
- 12. Количество пакетов TEXT
- 13. Количество пакетов JSON
- 14. Количество пакетов BINARY

Обязательна сборка статистики до пункта 11 (включительно).

- Что оценивается:
- 1) Работоспособность программы
 - 2) Стабильность при работе с большим объемом входных данных
 - 3) Скорость работы программы
 - 4) Качество кода
 - 5) Оптимизация по памяти и многопоточность

Сетевая модель

#	Уровень	Протокол
3	Application	JSON, TEXT, BINARY
2	Transport	TRANSPORT V1, TRANSPORT V2
1	Network	NETWORK V1, NETWORK V2

Иерархия протоколов



Порядок байт

Порядок байт в пакетах big-endian.
Например, рассмотрим число **123456789** в десятичной системе счисления.
Это же число в двоичной системе счисления: **00000111 01011011 11001101 00010101**.
Будет записано в пакете так:

Порядок байт в пакете	Значение
1	00000111
2	01011011
3	11001101
4	00010101

Сетевой уровень

NETWORK V1

	Bit							
Byte	0	1	2	3	4	5	6	7
0	Version(0x01)							
1	Source Address							
2								
3								
4								
5	Destination Address							
6								
7								
8								
9	Protocol							
10	Data Size							
11								
12	Header Checksum							
13								
14-N	Data							

NETWORK V2

Byte	Bit							
	0	1	2	3	4	5	6	7
0	Version(0x02)							
1	Source Address							
2								
3								
4								
5								
6								
7	Destination Address							
8								
9								
10								
11								
12								
13	Protocol							
14	Data Size							
15								
16	Header Checksum							
17								
18-N	Data							

Для протоколов NETWORK V1 и NETWORK V2 гарантируется отсутствие ошибок в заголовках и, соответственно, не может быть пакетов с ошибочным значением контрольной суммы Header Checksum.

Описание полей протоколов сетевого уровня

Поле	Описание
Version	Версия протокола
Source Address	Адрес отправителя
Destination Address	Адрес получателя
Protocol	Протокол верхнего уровня. Пакет протокола верхнего уровня помещается в поле Data. Значения поля: 1. 0x01 - TRANSPORT V1 2. 0x02 - TRANSPORT V2
Data Size	Размер полезных данных
Data	Полезные данные
Header Checksum	Контрольная сумма для проверки целостности заголовка. Сумма всех байт заголовка по модулю 2^{16}

Транспортный уровень

TRANSPORT V1

	Bit							
Byte	0	1	2	3	4	5	6	7
0	Source Port							
1								
2	Destination Port							
3								
4	Data Size							
5								
6-N	Data							
N+1	Checksum							
N+2								

TRANSPORT V2

	Bit							
Byte	0	1	2	3	4	5	6	7
0	Source Port							
1								
2	Destination Port							
3								
4	Fragment Number							
5								
6								
7								
8							L	F
9	Data Size							
10								
11-N	Data							
N+1	Checksum							
N+2								

Сессией протокола TRANSPORT V2 называется набор пакетов с последовательными порядковыми номерами фрагментов Fragment Number (включая пакеты с флагами F и L), переданных от одной конечной точки (конкретные Source Address и Source Port) до другой (конкретные Destination Address и Destination Port). В файле входных данных пакеты, относящиеся к одной сессии, могут лежать не по порядку, могут быть разделены пакетами других сессий. А также пакеты с некоторыми порядковыми номерами фрагментов вообще могут отсутствовать, в таком случае сессия не может быть собрана.

Описание полей протоколов транспортного уровня

Поле	Описание
Source Port	Порт отправителя
Destination Port	Порт получателя
Fragment Number	Порядковый номер фрагмента [0, 2^32].
F	Первый пакет в сессии
L	Последний пакет в сессии
Data Size	Размер полезных данных
Data	Полезные данные

Checksum	Контрольная сумма для проверки целостности пакета. Сумма всех байт пакета транспортного уровня по модулю 2^{16}
----------	---

Прикладной уровень

TEXT - текст в формате ASCII

JSON - текстовые данные в формате JSON

BINARY - неструктурированные бинарные данные