|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Лабораторная работа № 4 | | |
| по дисциплине «Операционные системы и компьютерные сети» | | |
| **Анализ структуры кадра технологии Ethernet** | | |
|  | | |
|  | Бригада 11 | уваров артём |
| Группа ПМ-23 | румянцев артёмаааааа |
| Место для ввода текста. |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Кобылянский Валерий Григорьевич |
|  | сивак мария алексеевна |
| Новосибирск, 2024 | | |

1. **Цель**

Спроектировать и реализовать программу, выполняющую анализ структуры кадра/фрейма технологии Ethernet.

1. **Задание**

1. Разработать и отладить программу, выполняющую анализ потока кадров. Потоки кадров представлены в виде файлов двоичного формата, место нахождения которых уточняется у преподавателя. В кадрах отсутствует преамбула и контрольная сумма, для исходящего кадра длина может быть меньше минимальной. Каждая бри-гада выполняет обработку одного файла с именем ethersXX.bin, где ХХ – номер бригады.

При выполнении работы в дистанционном режиме в обязательном порядке вы-полнить анализ файлов ethers06.bin и ethers07.bin.

Требования к программе:

• предусмотреть возможность ввода имени файла с клавиатуры;

• обеспечить вывод на экран по каждому кадру: номер, размер, тип, IP- адресов (основную информацию заголовка IP-пакета), MAC- адресов (основную информацию заголовка кадра); также необходимо вывести итоговые результаты обработки: общее число обработанных кадров и число кадров каждого типа (IP, ARP, Novell, SNAP и т.д.);

Алгоритм определения формата кадров можно найти в Приложении 2. Там же находится вся необходимая информация по форматам кадров Ethernet, IP- пакетам и TCP-сегментам.

**Текст программы:**

#include <iostream>

#include <vector>

#include <fstream>

#include <windows.h>

#include <map>

#include <iomanip>

using namespace std;

void printMacAddresses(ostream& output, const UCHAR\* data, int startIndex) {

for (UCHAR i = 0; i < 5; i++) {

output << setw(2) << setfill('0') << (int)data[startIndex + i] << ':';

}

output << setw(2) << setfill('0') << (int)data[startIndex + 5] << endl;

}

void printIpAddresses(ostream& output, UCHAR\* data, int startIndex) {

for (int i = 0; i < 3; i++) {

output << (int)data[startIndex + i] << '.';

}

output << (int)data[startIndex + 3] << endl;

}

int main()

{

int n\_dix = 0, n\_novell = 0, n\_snap = 0, n\_llc = 0, n\_ipv4 = 0, n\_arp = 0;

ofstream output("frames\_info.txt");

string filename;

string filepath;

cout << "Enter the filename: " << endl;

cin >> filename;

//filename = "ethers07.bin";

output << "File name: " << filename << endl;

filepath = "./stud\_ethers\_var2\_1/";

ifstream input(filepath + filename, ios::binary);

if (!input) {

cerr << "File opening error" << endl;

return 1;

}

vector<unsigned char> data(istreambuf\_iterator<char>(input), {}); // Запись всего файла в вектор

input.close();

int byte = 0;

int n\_frames = 0;

while (byte < data.size()) {

n\_frames++;

output << "Frame " << n\_frames << endl;

output << hex << uppercase;

output << "Recipient's MAC address: ";

printMacAddresses(output, data.data(), byte);

output << "Sender's MAC address: ";

printMacAddresses(output, data.data(), byte + 6);

output << dec;

unsigned short type\_bytes = (data[byte + 12] << 8) + data[byte + 13]; // Определение типа фрейма

int size;

if (type\_bytes > 0x05DC) {

output << "Type of frame: Ethernet DIX (Ethernet II)" << endl;

n\_dix++;

if (type\_bytes == 0x0800) {

output << "Protocol: IPv4" << endl;

n\_ipv4++;

output << "Sender's IP address: ";

printIpAddresses(output, data.data(), byte + 26);

output << "Recipient's IP address: ";

printIpAddresses(output, data.data(), byte + 30);

size = (data[byte + 16] << 8) + data[byte + 17] + 14;

output << "Size: " << size << " bytes" << endl;

byte += size;

}

else if (type\_bytes == 0x0806) {

output << "Type of frame: ARP" << endl;

n\_arp++;

output << hex << uppercase;

output << "Sender's MAC address: ";

printMacAddresses(output, data.data(), byte + 22);

output << dec;

output << "Sender's IP address: ";

printIpAddresses(output, data.data(), byte + 28);

output << hex << uppercase;

output << "Recipient's MAC address: ";

printMacAddresses(output, data.data(), byte + 32);

output << dec;

output << "Recipient's IP address: ";

printIpAddresses(output, data.data(), byte + 38);

size = 42;

output << "Size: " << size << " bytes" << endl;

byte += size;

}

else

byte += type\_bytes + 14;

}

else {

unsigned short LLC = (data[byte + 14] << 8) + data[byte + 15];

if (LLC == 0xFFFF) {

n\_novell++;

output << "Type of frame: Ethernet Raw 802.3" << endl;

size = type\_bytes + 14;

}

else if (LLC == 0xAAAA) {

n\_snap++;

output << "Type of frame: Ethernet SNAP" << endl;

size = type\_bytes + 14;

}

else {

n\_llc++;

output << "Type of frame: Ethernet 802.2/LLC" << endl;

size = type\_bytes + 14;

}

output << "Size: " << size << " bytes" << endl;

byte += size;

}

output << endl;

}

output << "Number of frames: " << n\_frames << endl << endl;

output << "Number of frames of each type:\n" << endl;

output << "Ethernet DIX (Ethernet II): " << n\_dix << endl;

//if (n\_dix > 0) {

output << " Protocols: " << endl;

output << " IPv4: " << n\_ipv4 << endl;

output << " ARP: " << n\_arp << endl;

// }

output << "Ethernet Raw 802.3 (Novell 802.3): " << n\_novell << endl;

output << "Ethernet SNAP: " << n\_snap << endl;

output << "Ethernet 802.2 / LLC: " << n\_llc << endl;

output.close();

return 0;

}

2. Анализ кадра с номером 11:

Кадр с номером 11 из файла ethers11.bin

00 02 16 09 FA 40 00 90 27 A1 36 D0 08 00 45 00 02 04 32 54 40 00 40 06 A0 D3 C3 3E 02 0B 51 B5 4E CE 00 16 90 FD E8 C9 48 0B D3 F6 13 B0 80 18 BF 28 6E 0A 00 00 01 01 08 0A 35 22 C0 74 08 96 0D F4 00 00 01 BC 09 1F 00 00 00 95 00 00 00 07 73 73 68 2D 72 73 61 00 00 00 01 23 00 00 00 81 00 B4 70 1C D8 A8 B4 42 1B 47 B0 66 E4 F4 3F 77 8A 86 90 CE C2 77 A3 9E 44 07 3B 2C 16 74 FD FB 92 C6 C8 56 36 AD 91 EC 25 87 53 A7 47 C2 AC 98 C9 28 5C 57 4B B0 E7 96 88 6A 22 D9 D2 DA 2E 60 DF 60 BA 67 4D 1D 6E 50 4A 77 C0 64 C4 FD 7B 9F 49 C5 72 7A 04 12 3E 36 12 BF C6 C6 94 E9 D8 C4 32 0B EF A2 5F E2 F8 2B 4D 7C 11 27 C8 08 F7 19 BF 07 38 D1 94 E8 89 D2 2F 70 65 D7 05 04 82 90 3D 00 00 00 81 00 C8 D1 79 F7 F2 9D 9B C1 58 A4 9F FE 89 B1 B0 0C A2 C3 2C 1F FB C6 69 7F D7 3E C8 14 64 F1 F1 9A 76 D2 94 0D A9 8B 2C 9E 69 F0 2B F5 70 D4 49 9A 33 4E A5 15 D3 62 E2 12 54 C2 E6 E1 2A 9E 10 1A B1 97 70 6E 4D B2 78 39 C0 24 39 88 DC 37 E8 F4 32 0D A5 6D 5E 54 C8 C5 FB 54 12 BD C9 AE 4E 2C 57 85 E0 B4 F4 FD 09 FF 16 DD 2A E9 50 C2 7F 2B 8C 59 7F 31 CE 87 EB D4 21 E8 16 8A E0 BA CC 84 00 00 00 8F 00 00 00 07 73 73 68 2D 72 73 61 00 00 00 80 90 32 C2 AF B2 1D F9 EE 6C 10 65 CF FB 69 DF 59 6F B7 19 DD AC 50 BB DB 3A 45 3A C6 CC D3 EB 23 CF AF 98 A3 ED 1B 7A C0 F1 84 75 BB 8D CD 1A 9E 72 73 25 D3 A8 09 C2 55 18 13 54 F2 00 58 55 16 5C 2C 60 A0 00 B0 EF C6 70 14 BA 04 DB 05 D5 15 E4 49 6E 81 FF CA D0 23 D3 43 5F 68 42 1E 9A 7F B4 D9 29 50 B2 5C E1 23 80 E1 BB B9 8D 3E DD 93 A1 46 04 05 00 EE 5A 0B 13 31 D8 67 FC 62 49 E8 00 00 00 00 00 00 00 00 00 00 00 00 0C 0A 15 00 00 00 00 00 00 00 00 00 00

Кадр начинается по адресу 0x00000C60. Его содержимое:

Заголовок Ethernet

00 02 16 09 FA 40 – MAC-адрес получателя: 00:02:16:09:FA:40

00 90 27 A1 36 D0 – MAC-адрес отправителя: 00:90:27:A1:36:D0

08 00 – Тип/Длина: протокол IPv4

Заголовок IPv4

45 – Версия + IHL: IPv4 (Версия 4) и длина заголовка 5 (20 байт)

00 – DSCP + ECN: Код точки обслуживания по умолчанию и отсутствие уведомления о перегрузке

02 04 – Общая длина: 516 байт для всего пакета IPv4

32 54 – Идентификация: 32 54

40 00 – Флаги + Смещение фрагмента: флаг "Не фрагментировать" и смещение фрагмента 0

40 – TTL: 64

06 – Протокол: TCP

A0 D3 – Контрольная сумма заголовка: A0 D3

C3 3E 02 0B – IP-адрес отправителя: 195.62.2.11

51 B5 4E CE – IP-адрес получателя: 81.181.78.206

Сегмент TCP

00 16 – Порт отправителя: 22

90 FD – Порт получателя: 37117

E8 C9 48 0B – Код позиции в сообщении

D3 F6 13 B0 – Номер октета, который должен прийти следующим

80 18 – Hlen, Резерв и Флаги: Длина заголовка 32 байта, установлен флаг ACK

BF 28 – Размер окна: 48936

6E 0A – Контрольная сумма

00 00 – Указатель важной информации: нет срочных данных

Далее содержится 32 бита опций (необязательно) и данные.