Задание «Протокол ARP. Адресация уровня Data Link (OSI).»

- **Загружать решения в виде ZIP архива в проект Smart LMS под названием HW2.**
- **В архиве должна быть папка, которая включает в себя проект на согласованном языке (см. ниже) и отчет[1] (DOCX/PDF формат).**
- **Название ZIP архива и папки в нем должно совпадать с Вашей фамилией.**

Мотивация. Для углубленного изучения протокола ARP и адресации уровня L2 (Data Link), необходима практика в виде разработки небольшой программной утилиты для выполнения задач, связанных с протоколом ARP и адресацией уровня L2 (Data Link).

Нужно использовать сетевую библиотеку PCAP (версия для Windows – NPCAP). Это чрезвычайно популярная библиотека. Например, утилита nmap (фигурирует в нескольких известных художественных фильмах), использует сетевую библиотеку PCAP: https://nmap.org/movies/ Ее также используют много современных больших проектов (в том числе WireShark): https://en.wikipedia.org/wiki/Pcap

На семинаре № 3 уже была освещена библиотека PCAP/NPCAP, ее цели, основное API. Также было мини-задание по разработке программы прослушивания трафика с помощью библиотеки PCAP.

Нужно использовать только PCAP (low-level API).
Использование других библиотек с целью замены PCAP запрещено.
Код на другом языке программирования, который не был согласован, либо без низкоуровневого PCAP не принимается.

Язык программирования.
Библиотека PCAP (или bindings для нее) доступна для разных языков.

Java https://github.com/kaitoy/pcap4j
https://www.jnetpcap.com/
https://github.com/jpcap/jpcap
https://github.com/slytechs-repos/jnetpcap-bindings/tree/main
Rust https://github.com/rust-pcap/pcap
C++ https://pcapplusplus.github.io/
C https://github.com/the-tcpdump-group/libpcap
Go https://github.com/miekg/pcap
https://pkg.go.dev/github.com/alicebob/pcap
https://github.com/dreadl0ck/gopcap

Другой язык – по согласованию с преподавателем семинарских занятий.

[1] *Примечание: без заполнения ключевых разделов отчета, напр. об использовании утилит или ИИ, работа может быть оценена с коэффициентом 0.5 (т. е., по умолчанию будет считаться, что не были выполнены ключевые требования задания либо был использован ИИ, шаблон отчета находится в конце этого файла)*

Хотя библиотека PCAP очень популярна и находится в активной разработке (оригинальный репозиторий – см. выше для языка C), она является довольно низкоуровневой. Поэтому для языков более высокого уровня репозитории PCAP bindings могут выглядеть устаревшими – это не означает, что сама по себе библиотека устаревшая. В то же время PCAP bindings выпущенные годы назад обычно хорошо работают.

Если у Вас возникают трудности с PCAP bindings, вы можете использовать гибридный подход: собрать данные для расчета статистики (см. ниже) с помощью одного языка (выгрузив их в файл), а обрабатывать собранные данные с помощью другого языка.

Оценивание задания. Во-первых, код будет анализироваться на корректность и точность выполнения требований этого задания. Далее, будет выполнен анализ полученных результатов и отчета. Затем будет учитываться качество кода (форматирование, обработка corner cases, иные стандартные элементы качества кода).

Содержание задания.
Реализовать приложение с использованием библиотеки PCAP. Управление приложением должно происходить путем ввода пользователем команд с консоли (как именно – подумайте и спроектируйте самостоятельно).
Ваше приложение должно уметь выполнять следующие задачи:

- Захват всех пакетов ARP и вывод их на консоль (включите «неразборчивый (*PROMISCUOUS*)» режим Вашего сетевого интерфейса) – интерпретируйте формат и содержимое захваченных кадров (в лекции по протоколу ARP, содержится описание заголовка и протокола ARP. Используйте эту информацию для интерпретации захваченных кадров).
- Выяснить MAC адрес Вашего роутера с помощью ARP запроса (IP роутера можно найти на подключенном устройстве, например ноутбуке с помощью утилит командной строки для формирования корректного заголовка ARP).
- Соберите статистику и напечатайте в консоль. За заданное пользователем время (вводится в консоли) после запуска Вашего приложения:
    - Сколько Ethernet фреймов передается? А сколько пакетов ARP?
    - Сколько уникальных MAC адресов обнаружено?
        - Почему? * (нужно ответить, см. шаблон отчета)
    - Сколько было широковещательных сообщений Ethernet? Из них, сколько с использованием протокола ARP?
    - Сколько было Gratuitous ARP Requests?
    - Сколько пар было ARP targeted requests и responses (нужно сопоставить во времени requests и соответствующие для них responses)?
    - Каков объем данных в байтах, который был передан между Вашим устройством и роутером (padding тоже учитываем в поле Data)?
        - Каким образом Вы можете рассчитать объем данных в байтах? * (нужно ответить, см. шаблон отчета)

Контрольные вопросы:
Для каких целей нужны *ARP targeted requests/responses*? Какова их структура?
Аналогичный вопрос о *Gratuitous ARP Request* и их структуре.

Внимание:

- Программа при старте должна выводить на консоль перечень команд, которые она поддерживает.
- Каждое под-задание данного задания должно реализовываться в отдельном классе/методе.
- Должно быть четко видно, в каком месте исходного кода реализована та либо иная функция**.**
- Обязательно выполнить отслеживание ARP пакетов в WireShark (достаточно найти несколько пакетов и понять, что они сгенерированы Вашим приложением или роутером).
- **Можно создать текстовый файл, который содержит перечень функций и указание на место кода, где функция реализована.**

Ход работы и фрагменты кода в помощь.

- Установите WireShark, в который входят драйвера для WinPCAP.
- Запустите WireShark либо ipconfig -all и найдите
    - IP адрес и MAC адрес своего сетевого интерфейса, по которому Ваше устройство подключено к Интернет
    - Свой IP адрес нужно будет указать вместо "192.168.1.6" в InetAddress.*getByName*("192.168.1.6")
    - Свой MAC адрес нужно будет указать вместо “ fe:00:01:02:03:04” в MacAddress.getByName("fe:00:01:02:03:04");
- Ваши IP и MAC можно задать в виде константы в отдельном текстовом файле (не в коде)

Пример для Java

- создавать проект maven/gradle
- использовать PCAP4J, зависимости для Maven проекта:

```
<dependency>
    <groupId>org.pcap4j</groupId>
    <artifactId>pcap4j-core</artifactId>
    <version>[1.0, 2.0)</version>
</dependency>
<dependency>
    <groupId>org.pcap4j</groupId>
    <artifactId>pcap4j-packetfactory-static</artifactId>
    <version>[1.0, 2.0)</version>
</dependency>
```

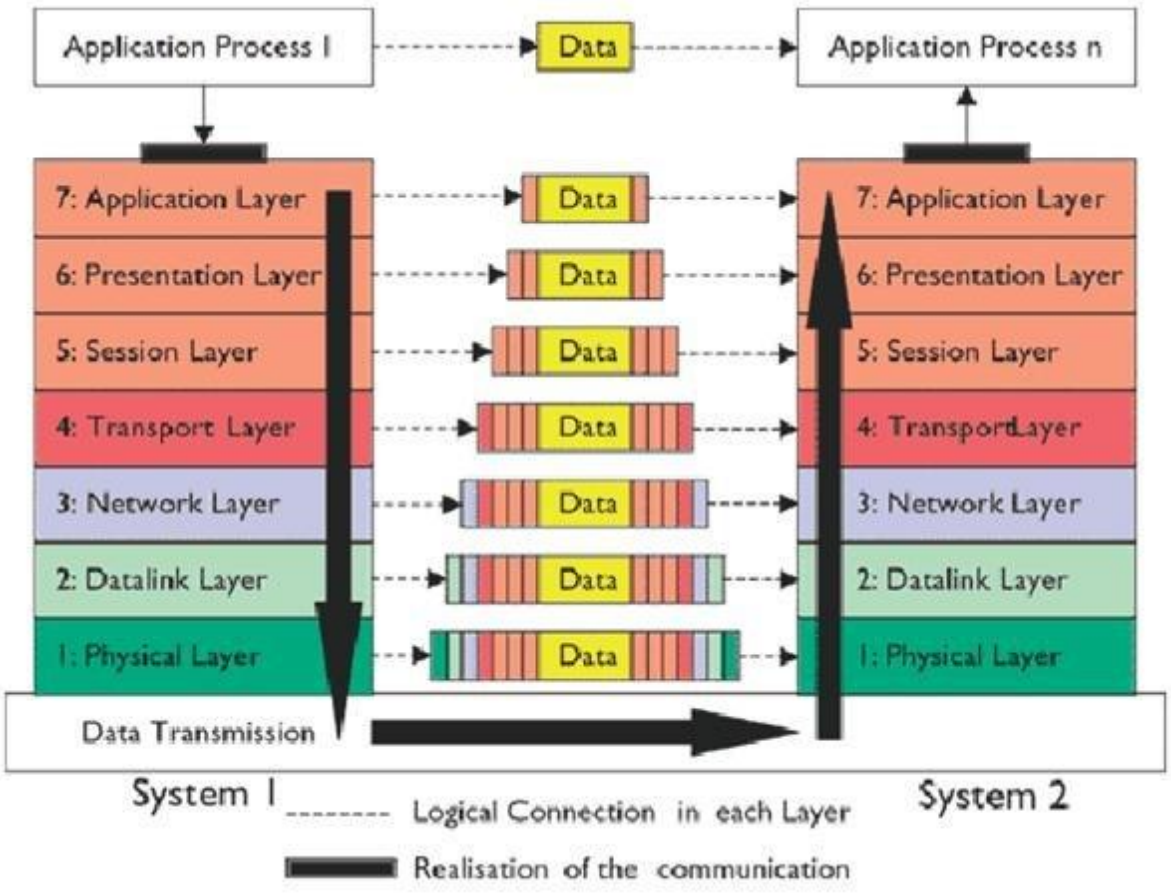
Минимальный код на Java для работы с WinPCAP (ведется захват всех сообщений по протоколу ARP):

```
package networks2025;

import org.pcap4j.core.*;
import org.pcap4j.packet.ArpPacket;
import org.pcap4j.packet.Packet;

import java.io.EOFException;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.concurrent.TimeoutException;


public class PcapARP {
    public static void main(String args[]) throws UnknownHostException,
PcapNativeException, EOFException, TimeoutException, NotOpenException {

        InetAddress addr = InetAddress.getByName("192.168.1.6");
        PcapNetworkInterface nif = Pcaps.getDevByAddress(addr);

        int snapLen = 65536;
        PcapNetworkInterface.PromiscuousMode mode =
PcapNetworkInterface.PromiscuousMode.PROMISCUOUS;
        int timeout = 10000;
        PcapHandle handle = nif.openLive(snapLen, mode, timeout);

        while(true) {
            Packet packet = handle.getNextPacketEx();

            ArpPacket arpPacket = packet.get(ArpPacket.class);
            if(arpPacket !=null) {
                System.out.println(arpPacket);
            }
        }
    }
}
```

Пример отправки ARP пакетов (PCap4J):

https://github.com/kaitoy/pcap4j/blob/v1/pcap4j-sample/src/main/java/org/pcap4j/sample/SendArpRequest.java

Для того, чтобы отправить ARP пакет, нам нужен Ethernet фрейм. Поэтому нужно создать ARP пакет и поместить его в Ethernet фрейм, только потом отправить фрейм (см. рисунок ниже).



Шаблон отчета

ОТЧЕТ ПО HW2
ФИО, группа, email

Цель работы: *укажите цель работы своими словами*

Я претендую на оценку до 5 баллов, был использован генеративный ИИ
(далее следует информация из декларации согласно
https://www.hse.ru/studyspravka/ai_guidelines/)
ЛИБО
Я претендую на оценку до 10 баллов, генеративный ИИ не был использован для работы над этим заданием.

Если Вам согласовали язык не из списка выше, напишите здесь кто и когда согласовал Вам выполнение работы на заданном языке. Если Вы выполняли работу не на одном языке, кратко опишите мотивацию и архитектуру решения.

Были использованы такие данные для работы:
здесь необходимо привести факты, свидетельствующие о том, что Ваше приложение и роутер в действительности коммуницировали, например:

- скриншоты утилит командной строки с информацией об IP роутера и клиента
- Скриншоты WireShark
- иные факты

Преподаватель может запросить и другие подтверждения при необходимости, в Ваших интересах привести полную картину экспериментальной установки.

Также необходимо привести скриншоты выполнения приложением каждой из задач:

- вывод на консоль
- WireShark

Скриншоты (а) WireShark и (б) вывода на консоль:

- «Захват всех пакетов ARP…» - скриншот работы WireShark, скриншот работы Вашего приложения с выводом пакетов
- «Выяснить MAC адрес…» - скриншоты WireShark (скриншот – соответствующий зарос, другой скриншот – ответ на запрос), скриншот работы Вашего приложения с выводом результата
- Статистика (8 показателей) – скриншоты, иллюстрирующие широковещательные сообщения Ethernet с пакетом ARP, Gratuitous ARP Requests, ARP targeted requests / responses

На скриншотах не только должны быть видны строчки с пакетами/фреймами, нужно в окне просмотра фрейма развернуть раздел с ARP заголовком и убедиться, что содержимое ARP заголовка попало в скриншот

Ответьте на вопросы, помеченные *