

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВПО «ЗабГУ»)  
Факультет: Энергетический  
Кафедра: Информатики, вычислительной техники и прикладной математики

## **КУРСОВОЙ ПРОЕКТ**

По дисциплине: Сети и телекоммуникации

На тему: Программа для генерации и приёма OAMPDU трафика

Выполнил студент группы ИВТ–18–2, Долгов Александр Артемович

Руководитель работы: доцент кафедры ИВТ и ПМ, Машкин Владимир  
Анатольевич

Чита  
2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВПО «ЗабГУ»)  
Факультет: Энергетический  
Кафедра: Информатики, вычислительной техники и прикладной математики

## **ЗАДАНИЕ**

на курсовой проект

По дисциплине: Сети и телекоммуникации

Студенту: Долгову Александру Артемовичу

Специальности (направления подготовки): Информатика и вычислительная техника

- 1 Тема курсовой работы: Программа для генерации и приёма OAMPDU трафика
- 2 Срок подачи студентом законченной работы: 10.06.2021
- 3 Исходные данные к работе: описание предметной области

Дата выдачи задания: 18.03.2021

Руководитель курсовой работы \_\_\_\_\_ / Машкин В.А. /  
(подпись, расшифровка подписи)

Задание принял к исполнению

«18» марта 2021 г.

Подпись студента \_\_\_\_\_ /Долгов А. А. /

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВПО «ЗабГУ»)  
Факультет: Энергетический  
Кафедра: Информатики, вычислительной техники и прикладной математики

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

По дисциплине: Сети и телекоммуникации

На тему: Программа для генерации и приёма OAMPDU трафика

Выполнил студент группы ИВТ–18–2, Долгов Александр Артемович

Руководитель работы: доцент кафедры ИВТ и ПМ, Машкин Владимир  
Анатольевич

Чита

2021

УТВЕРЖДАЮ

Зав.кафедрой \_\_\_\_\_

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Наименование раздела курсового проекта	Месяц, неделя															
	Март				Апрель				Май				Июнь			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Утверждение темы			+													
2. Защита темы в предметной области					+											
3. Утверждение функционала							+									
4. Защита ТЗ								+								
5. Анализ данных									+							
6. Разработка алгоритмов										+						
7. Защита программной части												+				
8. Защита реферативной части													+			
9. Публичная защита курсового проекта														+		

План выполнен: руководитель \_\_\_\_\_  
(подпись, расшифровка подписи)

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

## РЕФЕРАТ

Пояснительная записка – 24с., рисунки – 8, источники – 5 таблицы – 1.

OAMPDU, АГРЕГИРОВАНИЕ, SCAPY, PANDAS, РЕГУЛЯРНЫЕ  
ВЫРОЖЕНИЯ.

В данной работе рассматривается процесс создания программ, позволяющих отправлять и отлавливать пакеты OAMPDU трафика, от одного компьютера к другому. В работе определены методы разработки и описан процесс их применения при создании программ.

# СОДЕРЖАНИЕ

## Оглавление

РЕФЕРАТ .....	5
ВВЕДЕНИЕ.....	7
1. Анализ поставленной задачи .....	8
1.1 Описание предметной области .....	8
1.2 Постановка задачи .....	8
2. Теоретическая часть.....	9
2.1 Основные определения.....	9
2.2 Средства реализации .....	13
3. Анализ данных .....	14
3.1 Входные данные.....	14
3.2 Промежуточные данные.....	14
4. Программная реализация .....	15
4.1 Структура приложения.....	15
4.2 Алгоритм .....	15
5. Документирование .....	16
5.1 Руководство пользователя.....	16
ЗАКЛЮЧЕНИЕ .....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	18

## **ВВЕДЕНИЕ**

Цель работы – сделать программу для генерации и приема трафика OAMPDU. Изучить предметную область OAMPDU. Используя знания, полученные в ходе лекций, практических работ и самообучения реализовать программу для генерации и отправки трафика OAMPDU

В данной работе для генерации протокола используется язык программирования Python, вместе с Scapy.

## **1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ**

### **1.1 Описание предметной области**

В работе будет показана генерация, отлов и отправка трафика OAMPPDU. Работа программы представлена в двух экземплярах, одна для отлова, вторая для генерации и отправки трафика. В программе для генерации нужно ввести некоторые данные для генерации и отправки. В программе для отлова трафика нужно вести время. В течение введённого времени программа будет отлавливать пакеты и записывает данные в файл и в таблицу.

### **1.2 Постановка задачи**

Задачей проекта является разработка приложения для отправки OAMPPDU пакета.

Программа для генерации пакетов включает в себя: данные для генерации пакета.

Данные для генерации трафика: `dst`, `src`, `data`, `partner_state`, `actor_state`.



## 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 2.1 Основные определения

Итак, есть такой класс протоколов, которые призваны контролировать различные аспекты Ethernet, иначе говоря, Flow Control. Они являются частью стандарта 802.3 и делятся на два вида:

- **Быстрые.** Они должны отрабатывать моментально для предотвращения снижения производительности и перерыва в предоставлении сервисов вообще. Как правило они реализуются аппаратно. Собственно, представитель этого класса — *механизм PAUSE* – когда порт устройства получает трафика больше, чем может обработать, он отсылает PAUSE Frame противоположному узлу с просьбой понизить скорость отправки. Механизм «Паузы» — это технология для полнодуплексных линков. Для Half-Duplex существует *Back Pressure*. Он весьма строгий и не просто заставляет притормозить, а блокирует передачу данных, потому что фактически сообщает передающему коммутатору, что имеет место коллизия и нужно прекратить свои тщетные попытки. Но кого это сейчас волнует?

- **Медленные** – те самые Slow Protocols. У них не такие большие аппетиты на частоту отправки и задержки. Они реализуются программно. ОАМ относится именно к этому виду. Для него используется специальный MAC-адрес: – 0180-c200-0002, и EtherType 8809.

#### Ограничения

Во-первых, на такого вида протоколы накладываются следующие ограничения:

- 1) Передавать не более 10 кадров в секунду

2) Максимальное число протоколов с EtherType 8809 — десять. Теоретически их может быть больше, но для них уже будет указан другой тип в заголовке Ethernet.

3) Размер кадра «медленных» протоколов ограничен 128 байтами. \*Не сказать, что выполняется честно — OAMPDU, например, спокойно может превышать этот размер.

Вышеуказанные ограничения преследуют простую цель — уменьшить объём служебного трафика в сети. Как для протоколов более высокого уровня существуют специально выделенные мультикастовые IP-адреса (224.0.0.5 для OSPF, например), так и «медленным» протоколам назначили отдельный мультикастовый MAC-адрес: 01-80-C2-00-00-02.

Данный MAC-адрес принадлежит диапазону выделенному ISO/IEC 15802-3 для протоколов, ограниченных одним линком. Фактически это означает, что кадры, передающиеся на данный адрес не могут быть перенаправлены за пределы данного конкретного линка.

Вполне возможно, что могут существовать «медленные» протоколы, которые требуют юникастовой пересылки кадра — это не воспрещается. То есть адрес 01-80-C2-00-00-02 — это не требование — это пожелание. «Медленный» протокол может использовать другой MAC-адрес, но никакой другой протокол, кроме «медленного», не может использовать данный (01-80-C2-00-00-02).

Как я уже заметил выше, тип протоколов группы «Slow Protocols» (EtherType) — 8809. Далее определяются подтипы — subtypes для конкретных представителей данного класса:

## Подтипы

Подтип	Предназначение
0	Неиспользуемый запрещённый подтип
1	Link Aggregation Control Protocol (LACP)
2	Link Aggregation—Marker Protocol
3	Operations, Administration, and Maintenance (OAM)
4-9	Зарезервировано
10	Organization Specific Slow Protcol (OSSP)
11–255	Неиспользуемый запрещённый подтип

Таблица 1 – Подтипы

Как видите, выделено 8 бит под подтип. О типе 10 поговорим чуть ниже, а с остальными всё понятно: 1 — обычный знакомый нам LACP

2 — также часть механизма LAG для балансировки нагрузки, упорядоченного получения кадров и оптимизации управления линками LAG.

3 — протоколы Ethernet OAM.

По сути, два известных нам сейчас протокола, которые используют этот стандарт — это LACP и OAM, но вообще-то есть возможность создать собственный «медленный» протокол по своим нуждам. Это и есть тот самый пункт десять — OSSP — Organization Specific Slow Protocol.

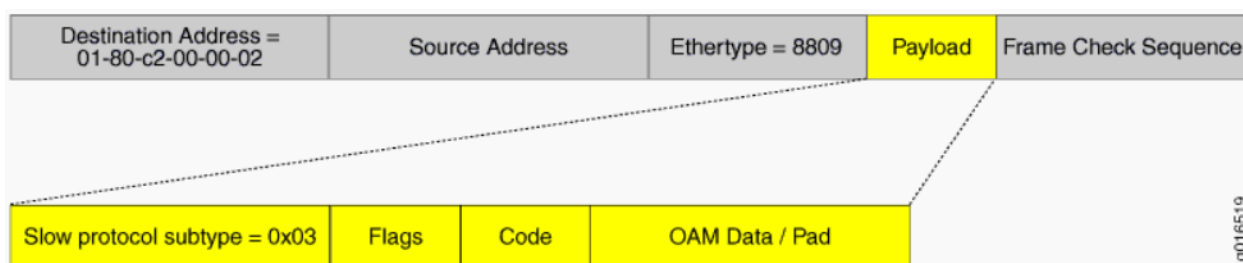


Рисунок 1 – Формат OAMPDU

Поле Flags используется для информирования партнера о локальном состоянии. Это состояние используется при обнаружении и удаленном обнаружении сбоев. Поле Code обозначает тип пакета OAM. Формат поля OAM Data/Pad состоит из элементов TLV.

Поддерживаются четыре типа сообщений OAM:

1. Информационный PDU OAM — PDU OAM переменной длины, который используется для процесса обнаружения. Этот PDU OAM содержит локальную, удаленную и специфичную для организации информацию.

2. PDU OAM уведомления о событии — PDU OAM переменной длины, который используется для мониторинга канала. Этот тип PDU OAM может передаваться несколько раз, чтобы повысить вероятность успешного приема, например, в средах, которые приводят к ошибкам в старших битах. PDU OAM уведомления о событии также включают временную метку, указывающую время, когда они инициируются.

3. PDU управления петлей OAM — предварительно заданный PDU OAM длиной 64 байта для включения или отключения команды удаленной проверки по петле.

4. Зависимый от поставщика OAM PDU — OAM PDU переменной длины, который позволяет добавлять в OAM расширения, зависящие от поставщика.

## **2.2 Средства реализации**

Язык программирования Python будет использован как основа для написания программы.

Scapy как инструмент, для создания и работы с сетевыми пакетами. Программа написана на языке Python.

Pandas - программная библиотека на языке Python для обработки и анализа данных.

### **3. АНАЛИЗ ДАННЫХ**

#### **3.1 Входные данные**

К входным данным генерации пакета относятся: `dst(destination)`, `src(source)`, `data`, `partner_state`, `actor_state`. Эти данные служат для генерации протокола и запрашиваются в специальных полях в программе.

#### **3.2 Промежуточные данные**

К промежуточным данным генерации пакета относится таблица с данными о пакете. Эта таблица состоит из: `dst`, `src`, `data`, `partner_state`, `actor_state`, `subtype`, `type`.

## 4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

### 4.1 Структура приложения

В результате работы с программой была придумана структура, по которой велась разработка.

Модуль (OAM\_main.py) – это точка вхождения в программу генерации.

### 4.2 Алгоритм

В данной работе реализован следующий алгоритм: **алгоритм отправки и генерации пакета**. Программа считывает введенные данные, генерирует пакет OAM и отправляет. После данные можно просмотреть в программе Wireshark.

## 5. ДОКУМЕНТИРОВАНИЕ

### 5.1 Руководство пользователя

В работе используется программа для генерации и отправки OAMPDU трафика.

Программа для генерации трафика, включается по модулю (OAM\_main.py). Как программа включилась нужно ввести команду **sendp(e)**, иначе программа не сможет отправить пакет по технологии OAM.

Также можно изменить значение переменной **data** для отправки другого сообщения.



## **ЗАКЛЮЧЕНИЕ**

В результате была разработана программа для отлова OAMPDU трафика.

Разработана система отправки пакетов, с помощью Scapy.

В результате работы изучена программная область, OAMPDU, Slow Protocols и Scapy.

В связи с нехваткой времени не все функции, и алгоритмы были реализованы. Например: Удобный интерфейс программы и сохранение отправленных OAMPDU пакетов в отдельный файл.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основы компьютерных сетей Тема №8. Протокол агрегирования каналов: Etherchannel [Электронный ресурс] – URL <https://habr.com/ru/post/334778/>
2. Что такое Slow Protocols [Электронный ресурс] – URL <https://habr.com/ru/post/180761/>
3. scapy.contrib.lacp [Электронный ресурс] – URL <https://scapy.readthedocs.io/en/latest/api/scapy.contrib.lacp.html>