

University of Pittsburgh

Project Proposal:  
Implement a TDMA Scheduler in Linux

Frank Fan & Enoch Li

CS 1980

Dr. Luis Oliveria

February 20, 2023

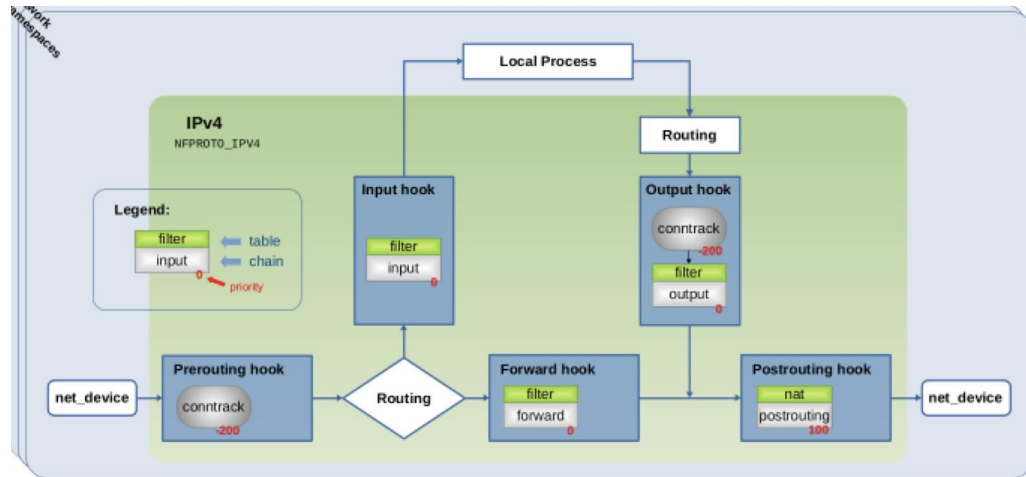
## I. Introduction

### a. Background

In recent years, the widespread use of wireless networks has led to an increase in the demand for high-speed and reliable Wi-Fi connections. However, in most wireless communications protocols, such as Wi-Fi, radio waves are transmitted from multiple devices into the air. If left at their own accord, they will interfere with each other, causing a higher number of missed messages and poor overall performance. Therefore, mechanisms have been developed to manage and organize such communications.

Oliveria et al. proposed a protocol in the paper “Managing High Loads in WiFi with Automatic Synchronization and Bandwidth Control” for managing high loads in Wi-Fi by using a combination of automatic synchronization and bandwidth control. The proposed protocol uses a novel approach to deal with generic traffic and improve the performance of highly loaded WiFi networks.

To generalize these mechanisms, the authors suggest two possible approaches: implementing a kernel-space module or implementing the solution in user-space using TUN/TAP interface. The former approach involves intercepting traffic and allowing it to flow when permitted, providing greater control but potentially sacrificing performance. The latter approach involves less control but a simpler implementation.



The primary objective of this project is to explore these two scenarios by developing a simple kernel module and a simple user-space solution using tun/tap interfaces. The project will analyze the performance of each approach using various performance metrics such as communication latency and difficulty of implementation, to determine which approach should be pursued further. By exploring these two scenarios, the project will gain a deeper understanding of the potential trade-offs between control and performance in managing wireless communication protocols.

b. Expected outcome

The expected outcome of this project is to develop and analyze the two different approaches for implementing the protocol proposed by Oliveria et al. for managing high loads in Wi-Fi with automatic synchronization and bandwidth control. Specifically, the project aims to develop a simple kernel module and a simple user-space solution using tun/tap interfaces, which will be evaluated based on various performance metrics. By comparing these two approaches, the project aims to determine which approach provides better control and performance for managing wireless communication protocols.

c. Expected impact

The impact of this project is twofold. First, it will help to further develop and optimize the protocol proposed by Oliveria et al. for managing high loads in Wi-Fi with automatic synchronization and bandwidth control. By exploring different implementation approaches, the project can help identify the more effective and efficient way to manage wireless communication protocols, having potential benefits in terms of reducing interference, improving performance, and enhancing the reliability of wireless communication systems.

Second, the project can provide valuable reference for further research in the field of wireless communication. The project could help to promote ongoing research and innovation in the field of wireless communication, leading to continued improvement in performance, reliability, and overall efficiency. Ultimately, this could have a range of applications in areas such as telecommunications, the internet of things, and mobile computing, helping to advance the state of art in wireless communication and drive new developments in the world.

## II. Project Design

a. Technologies/Framework

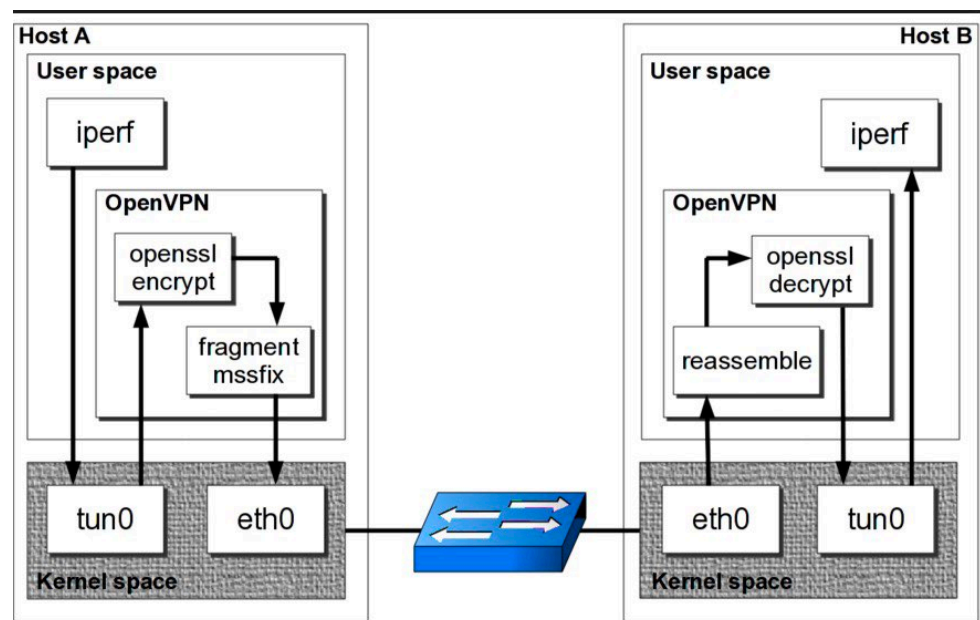
- Programming language: the kernel module and user-space solution for this project will be primarily developed in C.
- Knowledge of Linux kernel development, NFQUEUE, netfilter.
- Network protocols: TCP/IP, UDP
- TUN/TAP interfaces: the user-space solution will use TUN/TAP interfaces.
- Source code management: Git will be used for managing and versioning of the project code.
- Network analysis tools
- Testing environments

b. Project architecture

The project architecture will consist of two main components: a kernel-space module and a user-space solution using tun/tap interfaces.

The kernel-space module will be responsible for intercepting traffic and only allowing it to flow when permitted. This module will be implemented in C and will have direct access to the operating system's resources, allowing for fine-grained control over traffic flow.

The user-space solution will be implemented using tun/tap interfaces, which provide a virtual network device that can be used to send and receive packets. This solution will also be implemented in C, but will operate in user-space, which may lead to decreased performance but will simplify implementation.



To compare the performance of these two solutions, we will use metrics such as communication latency and difficulty of implementation. We will also test the scalability of each solution by simulating a large number of devices communicating simultaneously.

- c. Foreseen Issues
  - Technical difficulties
    - Developing a kernel module and a user-space solution can be technically challenging, especially when dealing with wireless communication protocols that our group is not familiar with. The project may require significant effort to overcome issues related to memory management, synchronization, and performance optimization.
  - Project scope

- The project may require a significant amount of time and effort to fully explore and evaluate the two implementation approaches because finding the optimal balance between the various metrics to be measured may require extensive testing and analysis. It may be necessary to limit the scope of the project in order to ensure that it is feasible to complete the project within the available time and resources.

### III. Expected Timeline

The development of the project will be divided into three phases (subjected to change):

- Phase 0 (4 weeks): Research and Setup
  - Configure Ubuntu on virtual machine for setup
  - Research TDMA scheduling and kernel module development
  - Implement a basic kernel module for receiving a packet and sending it out right away
  - Research TUN/TAP interfaces and user-space solutions
  - Implement a basic user-space program using TUN/TAP interfaces for receiving a packet and sending it out right away
  - Stretch goals:
    - Explore other scheduling mechanisms such as FDMA and CDMA
    - Develop a more advanced kernel module with configurable parameters
    - Implement a more complex user-space solution using netfilter
- Phase 1 (3 weeks): Performance Analysis
  - Continue to develop the kernel module and user-space solution to support TDMA scheduling
  - Set up testing environment and gather performance metrics for kernel module
  - Set up testing environment and gather performance metrics for user-space solution
  - Analyze performance metrics and compare results
  - Stretch goals:
    - Explore performance in different scenarios (e.g. high traffic, varying message sizes)
    - Analyze resource usage of both solutions
- Phase 2 (2-3 weeks): Conclusion and Reporting
  - Finish up implementation of the programs and performance analysis
  - Draw conclusion based on performance metrics and resource usage
  - Finalize report and presentation
  - Stretch goals:
    - Conduct additional testing and analysis to further validate findings
    - Investigate potential real-world applications for TDMA scheduling in Linux

#### IV. Communication

- Communication with sponsors.
  - Weekly in-person meeting scheduled on Wednesday with Dr. Oliveria.
- Communication within team
  - Discord.

Communication between the group members with visibility to the POC is conducted through Discord.
  - WeChat.

Communication within the group members is conducted through WeChat.
  - Zoom.

Zoom meetings will be held between the group members to discuss more complex issues regarding the project.