Task Report

Name: Mohamed Elmeselhy

Submitted to: Eng. Mohamed ElSamanoudy

Eng. Noha Gamal

Eng. Ola Alaa

Code Structure:

1. File Organization

1.1 Main.py: Main entry point for the application. It reads configurations from an input file, generates packets based on the specified payload type, and writes them to an output file.

1.2 generation.py: Contains classes for packet generation, including Ethernet-Generation and ECPRI-Generation.

1.3 packet.py: Defines data classes for representing Ethernet and ECPRI packets.

1.4 file_handler.py: Implements file handling operations, including reading, and writing configuration files.

1.5 logger.py: Provides logging functionality for the application.

2. Files explanation

2.1 Main.py: Main entry point for the application. It reads configurations from an input file, generates packets based on the specified payload type, and writes them to an output file.

- Checks if the input is correct.
- Clears the log file.
- Assign paths for config and output files.
- Check read and write permissions.

Based on config file main will decide if it generates ECPRI or Ethernet

2.2 file_handler.py: Implements file handling operations, including reading, and writing configuration files.

- Define enumeration class represents different payload types.
- File handler class with methods for reading and writing files.
- Check read and write permissions.
- Read from config file.
- Write buffer to output file.

2.3 generation.py: Contains classes for packet generation, including Ethernet-Generation and ECPRI-Generatio

2.3.1 Abstract Base Class - Generation:

- Serves as the base class for packet generation.
- Contains abstract methods for generating single packets, single bursts, and writing bursts to a file.

2.3.2 Ethernet Generation (Generation):

- Generates Ethernet packets with preamble, start of frame, destination address, source address, Ether Type, payload, and CRC.

2.3.3 ECPRI Generation (Generation):

- Generates ECPRI packets with a predefined first byte, message type, maximum packet size, and payload.

2.3.4 Burst Generation and Timing:

- Generates bursts of packets based on configured parameters such as burst size and burst duration.
- Handles timing constraints for bursts using time-based calculations.

2.4 packet.py: Defines data classes for representing Ethernet and ECPRI packets.

2.4.1Data Classes:

- The code defines two data classes: Packet and its subclasses ECPRI and Ethernet.
- Packet class holds generic attributes common to all types of packets.
- ECPRI and Ethernet are subclasses of Packet, tailored for ECPRI and Ethernet packets respectively.

2.4.2Attributes:

- Packet class includes attributes such as stream duration, burst size, burst period, and max packet size.
- ECPRI class adds attributes specific to ECPRI packets like protocol version, reserved field, and C field.
- Ethernet class adds attributes specific to Ethernet packets such as the number of Inter-Frame Gaps (IFGs), source and destination MAC addresses, and Ethernet Type field.

2.4.3Properties:

- Both ECPRI and Ethernet classes have properties to calculate various fields of their respective packet types.
- These properties compute values like the first byte of an ECPRI packet, message type, max packet size in bytes, payload size, Inter-Frame Gap (IFG), Start of Frame (SOF), and preamble.

2.4.5Methods:

- The classes utilize properties to dynamically compute field values based on packet attributes, eliminating the need for explicit method calls.
- Data Conversion:
- The code uses methods like to_bytes() and byte string concatenation to convert data between different formats such as binary strings and byte arrays.
- Integration with Other Modules:
- The code integrates with other modules such as sys for system-related functionalities.

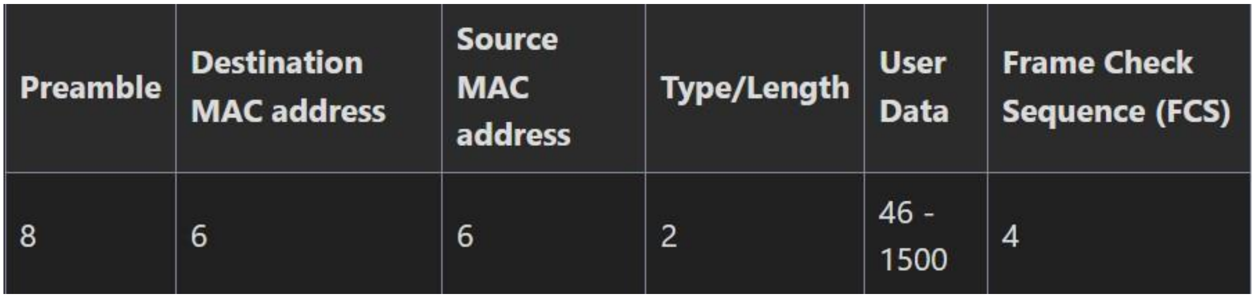
2.5 logger.py: Provides logging functionality for the application.

2.5.1Methods:

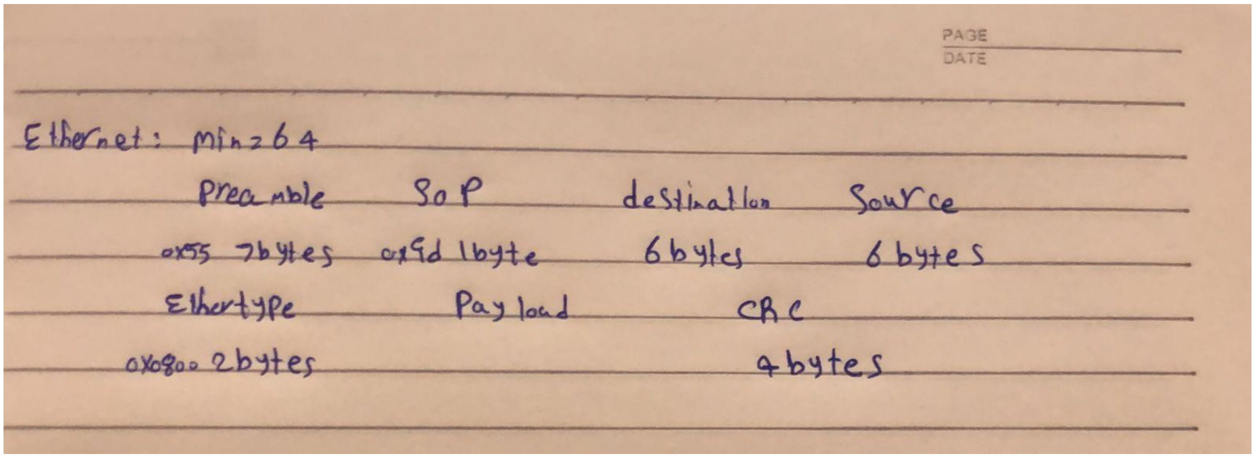
- clear_log(): Clears the log file by creating an empty file.
- log(message): Appends a message to the log file.

3. Packet structure

3.1 Ethernet:



Preamble	Destination MAC address	Source MAC address	Type/Length	User Data	Frame Check Sequence (FCS)
8	6	6	2	46 - 1500	4

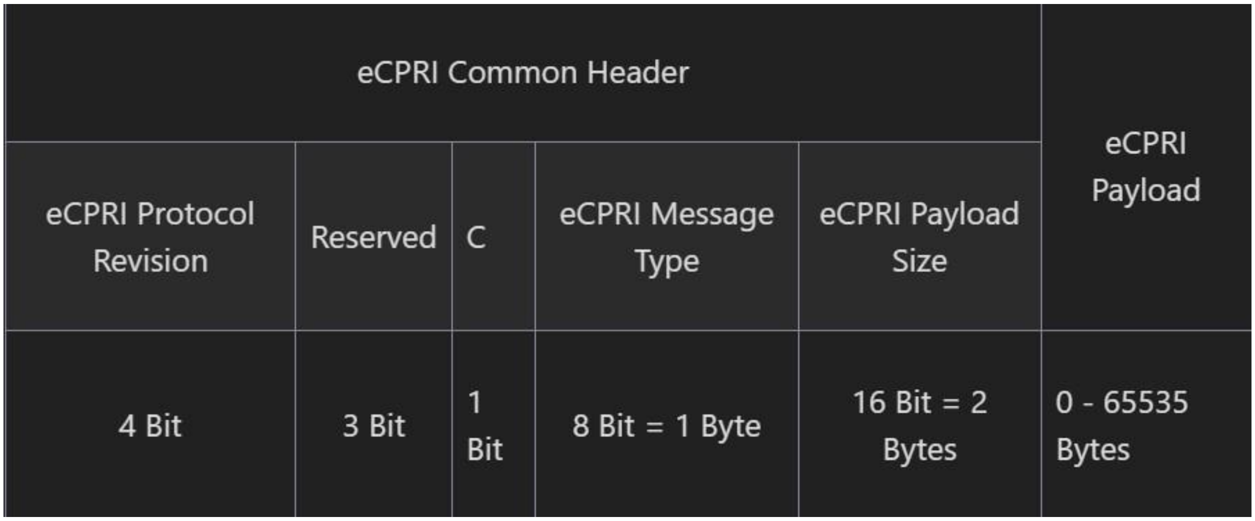


PAGE
DATE

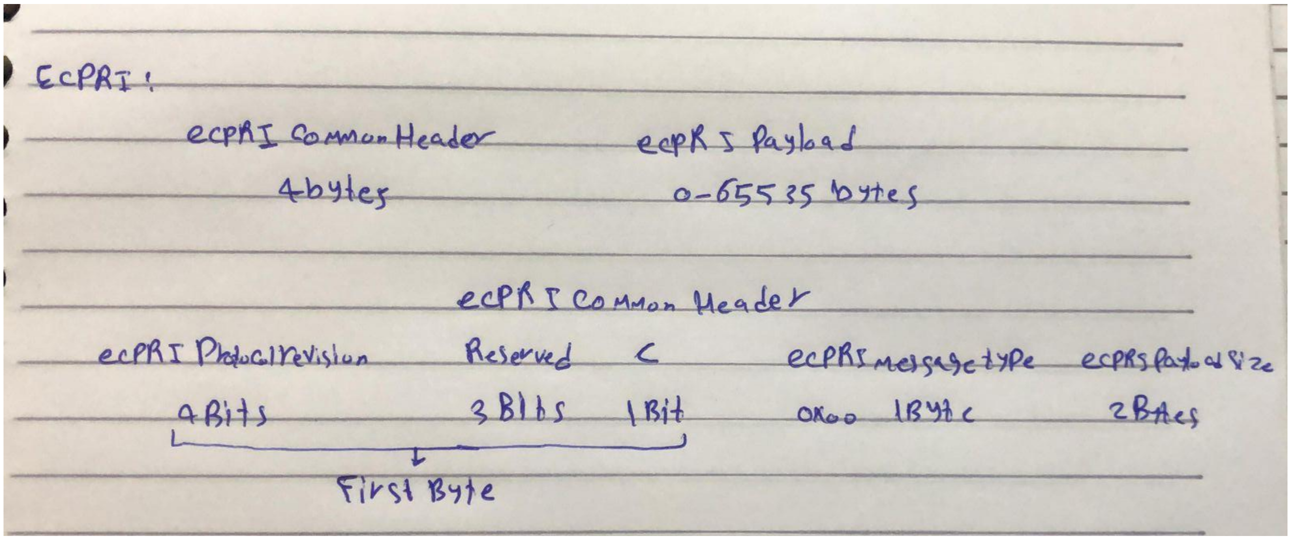
Ethernet: min=64

Preamble	SoP	destination	Source
0x55 7bytes	0x9d 1byte	6bytes	6 bytes
Ethertype	Payload	CRC	
0x0800 2bytes		4bytes	

3.2 ECPRI



eCPRI Common Header					eCPRI Payload
eCPRI Protocol Revision	Reserved	C	eCPRI Message Type	eCPRI Payload Size	
4 Bit	3 Bit	1 Bit	8 Bit = 1 Byte	16 Bit = 2 Bytes	0 - 65535 Bytes



ECPRI:

ecPRI Common Header	ecPRI Payload
4bytes	0-65535 bytes

ecPRI Common Header

ecPRI Photocolrevision	Reserved	C	ecPRI message type	ecPRI Payload size
4Bits	3Bits	1Bit	0x00 1Byte	2Bytes

$\underbrace{\text{4Bits} \quad \text{3Bits} \quad \text{1Bit}}_{\text{First Byte}}$