Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

ОТЧЁТ

По лабораторной работе № 5

На тему

**ДЕТЕРМИНИРОВАННЫЕ МЕТОДЫ ДОСТУПА К МОНОКАНАЛУ**

Выполнил: Проверил:

Студент группы 050503 Преподаватель

Сикорин А. С. Марцинкевич В. А.

Минск 2022

1. **Цель работы:**

Изучить методы доступа к каналам СрПД в компьютерных сетях. Разработать программу, реализующую процедуру детерминированного доступа к моноканалу по алгоритму Token ring.

1. **Задание к лабораторной работе:**

Написать программу передачи данных через COM-порты упрощенным алгоритмом Token ring в соответствии с требованиями.

**3. Теоретическая часть**

Зачастую в компьютерных сетях станции разделяют единый ресурс — общий канал передачи данных, моноканал. При «столкновении» данных из разных передатчиков возникают коллизии в связи с физической реализацией сети. Стоит задача организовать работу станций таким образом, чтобы станции не мешали друг другу, редко попадая в трудноразрешимые ситуации, или же ослабить контроль за порядком передачи данных, что часто ведет к конфликтам, которые, однако, легко разрешаются. Детерминированные методы доступа к моноканалу работают в соответствии с первым вариантом, а конкретно в алгоритме Token ring между станциями передается специальный кадр — маркер, который дает принявшей его станции право отсылать собственные данные. Если станция не имеет данных для пересылки или проверка приоритетов ее данных говорит о том, что маркер больше пригодится другой станции, то принявшая маркер обязана передать его в кольцо незамедлительно, в ином случае станция передает свои данные и запускает таймер, чтобы через какое-то время освободить маркер. Те станции, что не имеют маркера, в этот момент могут лишь принимать и отсылать чужие данные. Если станция приняла кадр, предназначенный ей, то она изменяет биты в поле AC(access control), сигнализирующие о том, что кадр воспринят корректно, и отсылает кадр обратно в кольцо, чтобы адресант уничтожил его или повторил, если биты не выставлены. При этом идет изменение приоритетов кадров.

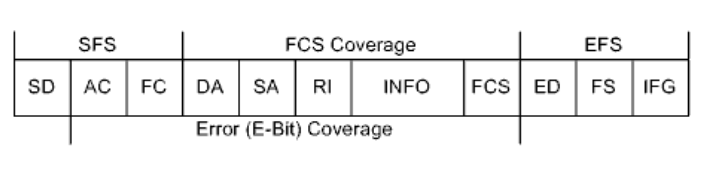


Рис. 1 – формат кадра Token ring

Программа эмулирует кольцо из четырех станций, одна из которых имеет маркер изначально. Станции генерируют собственные сообщения и складывают их в очередь, при этом принимают и передают чужие кадры. Каждая станция имеет свой уникальный приоритет. Если станция не имеет кадра для передачи или приоритеты кадров требуют передачи маркера, то маркер освобождается и переходит к следующей станции.

1. **Практическая часть**

#!/usr/bin/env python3

from dataclasses import dataclass

TOTAL\_STATIONS = 4

@dataclass

class Sequence:

priority: int = 0

is\_frame: bool = False

destination\_address: int = None

source\_address: int = None

message: str = None

is\_address\_recognized: bool = False

is\_frame\_copied: bool = False

def \_\_repr\_\_(self):

if self.is\_frame:

return f"Frame(" \

f"P={self.priority} " \

f"T={int(self.is\_frame)} " \

f"DA={self.destination\_address} " \

f"SA={self.source\_address} " \

f"INFO='{self.message}' " \

f"A={int(self.is\_address\_recognized)} " \

f"C={int(self.is\_frame\_copied)})"

return f"Token(" \

f"P={self.priority} T={int(self.is\_frame)})"

def main():

current\_sequence = Sequence()

stations\_queue: dict[int, Sequence] = {i: None for i in range(TOTAL\_STATIONS)}

while True:

for station\_id in range(TOTAL\_STATIONS):

print(f"\nStation №{station\_id}")

print(f"Stations queue={stations\_queue}")

print(f"Get sequence={current\_sequence}")

if current\_sequence.destination\_address == station\_id:

print(f"Message has arrived: {current\_sequence.message}")

current\_sequence.is\_address\_recognized = True

current\_sequence.is\_frame\_copied = True

if current\_sequence.source\_address == station\_id:

print(f"Remove consumed message: {current\_sequence.message}")

if stations\_queue[station\_id] is not None:

current\_sequence = stations\_queue[station\_id]

stations\_queue[station\_id] = None

else:

current\_sequence = Sequence()

if stations\_queue[station\_id] is None:

message = input("Enter new station message (`n` for skip):").strip()

if message != "n":

destination\_address = int(input("Enter destination station address:"))

new\_sequence = Sequence(station\_id, True,

destination\_address, station\_id, message,

False, False)

if current\_sequence.is\_frame and current\_sequence.priority < new\_sequence.priority:

print(f"Store current frame={current\_sequence} in queue")

stations\_queue[station\_id] = current\_sequence

else:

print("Capture token")

current\_sequence = new\_sequence

if \_\_name\_\_ == '\_\_main\_\_':

main()