

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

---

Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

# Телекоммуникационные технологии

## Отчёт по лабораторной по работе №12

выполнил:  
Биттиев М.Х.  
Группа:  
5130901/10201  
Преподаватель:  
Н. В. Богач

Санкт-Петербург

2024

## Содержание

1. Теоритическая основа	3
2. Схемы в GNU RADIO	3
3. Тестирование	4
4. Вывод	7

# 1. Теоритическая основа

Пакетная коммуникация — это метод передачи данных, при котором информация разбивается на небольшие блоки, называемые пакетами, которые затем передаются по коммуникационному каналу от отправителя к получателю. Каждый пакет содержит не только полезную нагрузку данных, но и дополнительную информацию, необходимую для передачи и обработки пакета.

Статья на странице "Packet Communications" в документации GNU Radio объясняет основные концепции и компоненты, связанные с пакетной коммуникацией. Вот основные понятия, которые обычно связаны с пакетной коммуникацией :

- **Пакеты:** Пакеты представляют собой блоки данных, которые содержат полезную нагрузку, например информацию для передачи или команды. Каждый пакет имеет заголовок, который содержит дополнительную информацию, такую как адрес назначения, адрес отправителя, контрольные суммы и другие поля.
- **Модуляция и демодуляция пакетов:** перед передачей пакеты могут быть модулированы для преобразования их в аналоговый или цифровой сигнал, который можно передать по физическому каналу связи. На стороне получателя сигнал демодулируется для извлечения пакетов.
- **Обнаружение и декодирование пакетов:** Получатель должен выполнить обнаружение пакетов, чтобы определить начало и конец каждого пакета в потоке данных. Затем происходит декодирование пакетов, чтобы извлечь полезную нагрузку и другую информацию из заголовков.
- **Контрольные суммы и исправление ошибок:** Пакетная коммуникация часто включает контрольные суммы или другие методы проверки целостности данных, которые позволяют обнаруживать ошибки передачи данных. В случае обнаружения ошибок можно предпринять меры для исправления ошибок или повторной передачи пакета.
- **Режимы передачи:** Пакетная коммуникация может быть реализована в различных режимах, таких как однонаправленная передача, двунаправленная передача или многопользовательская передача.

## 2. Схемы в GNU RADIO

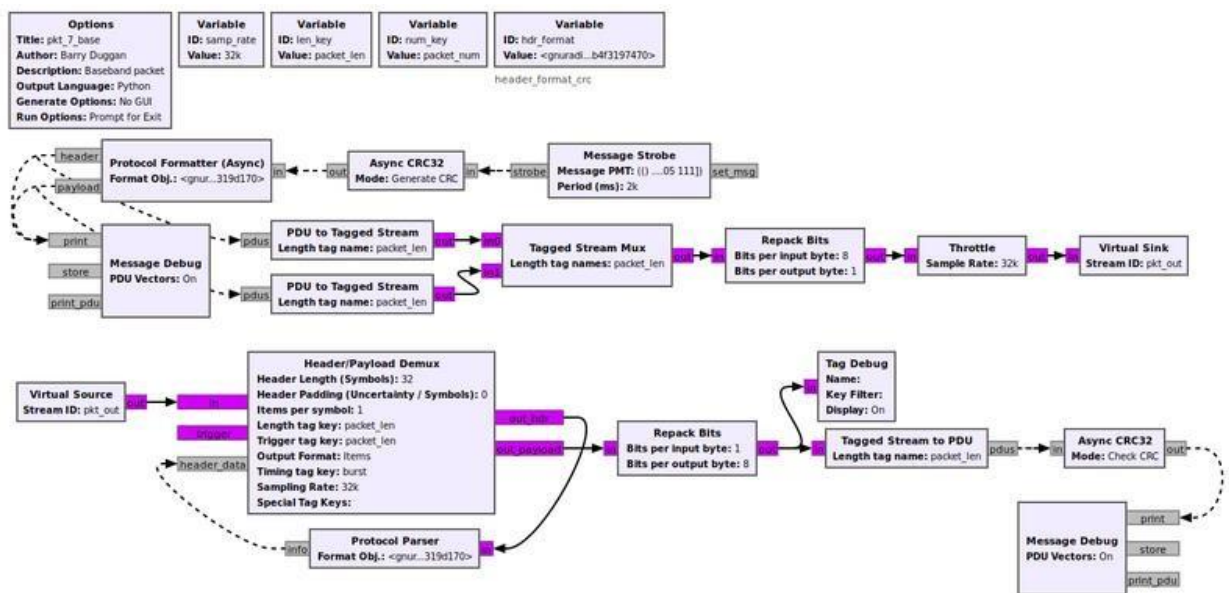


Рис 2.1. Схема базовой обработки пакетов

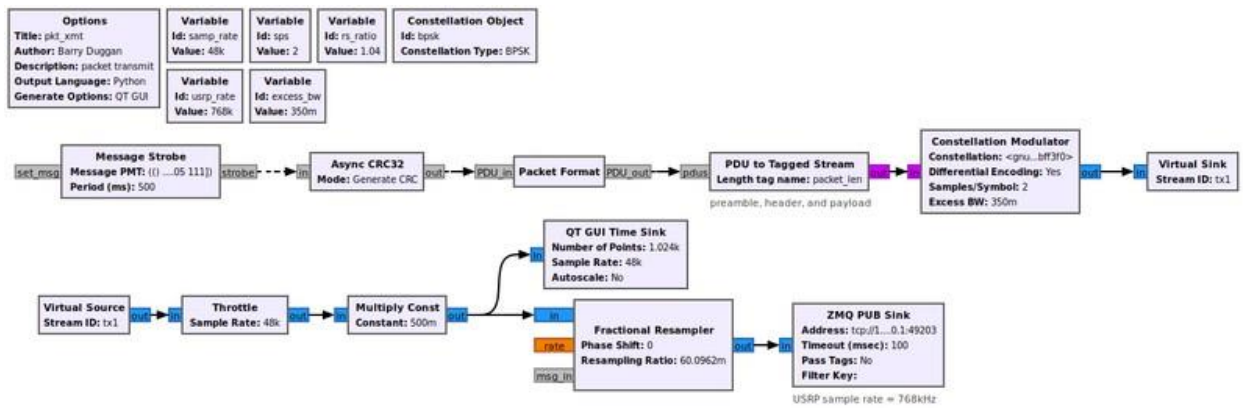


Рис. 2.2 Схема передатчика сигнала

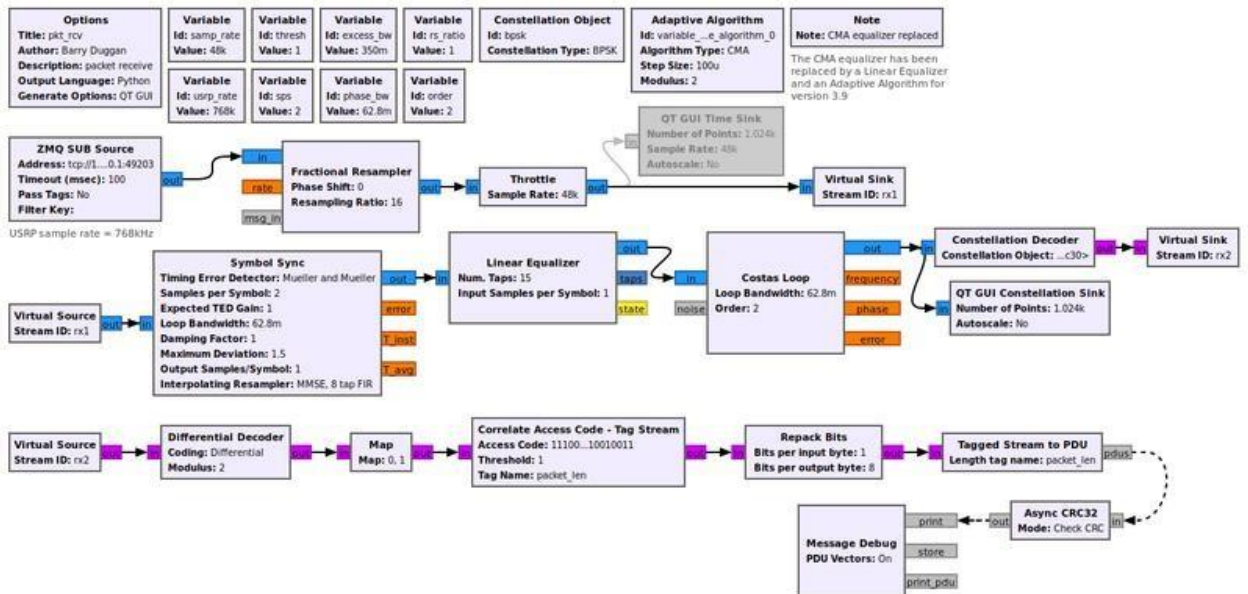


Рис 2.3. Схема приемника сигнала

### 3. Тестирование

Запустим посмотрим на работу базовой схемы обработчика сигналов, общая цель заключается в создании и запуске графа потока GNU Radio для обработки и передачи пакетов данных. Различные блоки выполняют операции по форматированию, мультиплексированию, демультиплексированию, расчету CRC и отладке передаваемых данных.

## Пример вывода показаний

```
***** VERBOSE PDU DEBUG PRINT *****
()
pdu length = 4 bytes
pdu vector contents =
0000: 0d 10 00 3a
*****

***** VERBOSE PDU DEBUG PRINT *****
()
pdu length = 13 bytes
pdu vector contents =
0000: 47 4e 55 20 52 61 64 69 6f 8f aa 09 f1
*****

-----

Tag Debug:
Input Stream: 00
  Offset: 13  Source: n/a      Key: packet_len  Value: 13
  Offset: 13  Source: n/a      Key: packet_num   Value: 1
-----

***** VERBOSE PDU DEBUG PRINT *****
((packet_num . 1))
pdu length = 9 bytes
pdu vector contents =
0000: 47 4e 55 20 52 61 64 69 6f
*****
```

Рис 3.1. Часть лога

Из предоставленного вывода можно сделать следующие наблюдения:

1. Вывод начинается с раздела "\*\*\*\*\* VERBOSE PDU DEBUG PRINT \*\*\*\*\*", который указывает на детальную отладочную информацию о передаче пакетов.
2. Первый блок вывода отображает информацию о PDU (Protocol Data Unit) длиной 4 байта. PDU представляет собой блок данных, который может быть передан через коммуникационный канал. Здесь показан вектор содержимого PDU, представленного в шестнадцатеричном формате.
3. Второй блок вывода отображает информацию о следующем PDU длиной 13 байт. Опять же, показан вектор содержимого PDU в шестнадцатеричном формате.
4. После блока с векторами PDU следует раздел "Tag Debug", который содержит дополнительную информацию о передаче данных. Здесь указано, что входящий поток имеет значение "00", и есть две метки с их соответствующими значениями ключей и смещениями в потоке.
5. Последний блок вывода отображает информацию о третьем PDU длиной 9 байт. В этом случае также показан вектор содержимого PDU в шестнадцатеричном формате. Кроме того, указано, что это PDU с атрибутом "packet\_num", и его значение равно 1.

Общий вывод из этого вывода заключается в том, что происходит передача пакетов данных. Каждый пакет имеет свою длину и содержит определенное содержимое,

представленное в шестнадцатеричном формате. Также присутствует дополнительная отладочная информация и метки, которые могут быть использованы для анализа и обработки переданных пакетов.

Теперь запустим Pkt\_rcv.grc который будет являться нашим приемником сигнала, и pkt\_xmt.py - потоковый граф для передачи пакетов данных.

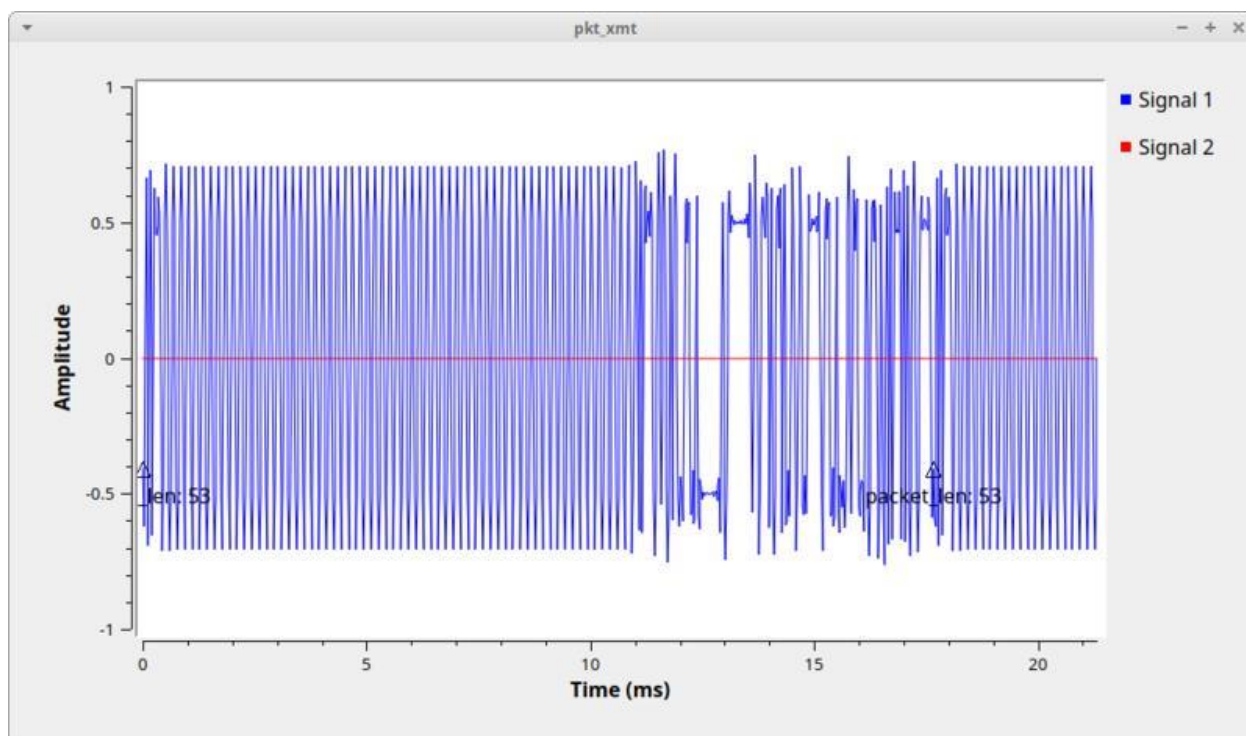


Рис 3.2. Выходной сигнал передатчика

```
***** VERBOSE PDU DEBUG PRINT *****  
(  
pdu length = 9 bytes  
pdu vector contents =  
0000: 47 4e 55 20 52 61 64 69 6f  
*****
```

Рис 3.3. Выходные данные приемника с консоли

Анализируя вывод, можно сделать следующие наблюдения:

1. pdu length = 9 bytes: Размер пакета составляет 9 байт.
2. pdu vector contents: Содержимое пакета представлено в шестнадцатеричном формате. Каждый символ соответствует одному байту данных.
3. 0000: 47 4e 55 20 52 61 64 69 6f: Пакет содержит следующие значения (в шестнадцатеричном формате): 47 4e 55 20 52 61 64 69 6f. Это можно интерпретировать как ASCII-коды символов: "GNU Radio". Каждое значение представляет байт данных.

Таким образом, пакет, который был передан, содержит строку "GNU Radio" в виде байтовых данных.

## 4. Вывод

Работа с пакетной передачей позволяет осуществлять передачу данных в виде пакетов, которые содержат информацию о своей структуре и содержимом. Некоторые выводы, которые можно сделать в работе с пакетной передачей, включают:

- Пакетная передача обеспечивает структурирование данных: Пакеты могут содержать информацию о типе данных, длине пакета, заголовках и полезных данных. Это позволяет передавать данные с определенным форматированием и организацией, что облегчает их обработку на приемной стороне.
- Возможность передачи метаданных: Пакеты могут содержать дополнительные метаданные, такие как временные метки, идентификаторы, контрольные суммы и другую информацию, которая может быть полезна для обработки и проверки целостности данных.
- Управление потоком данных: Пакеты могут использоваться для управления потоком данных. Например, можно использовать пакеты-заголовки для указания начала и конца сообщений или для управления приоритетом передачи данных.
- Обнаружение и восстановление ошибок: Пакеты могут содержать контрольные суммы или другие механизмы для обнаружения ошибок при передаче данных. Это позволяет обнаруживать и исправлять ошибки для обеспечения целостности и достоверности данных.
- Мультиплексирование и демультиплексирование: Пакетная передача позволяет объединять несколько потоков данных в один и передавать их по одному каналу. На приемной стороне эти потоки могут быть разделены и обработаны независимо.
- Эффективность использования ресурсов: Пакетная передача может быть эффективнее использовать доступные ресурсы, такие как пропускная способность канала связи или вычислительные ресурсы, позволяя упаковывать данные в пакеты и передавать их в оптимальном формате.

В целом, работа с пакетной передачей предоставляет более гибкие и мощные инструменты для передачи, обработки и управления данными в системах связи и обработки сигналов. Это позволяет эффективно использовать ресурсы и обеспечивает надежную передачу данных с учетом их структуры и контроля целостности.