

# **Отчёт по лабораторной работе №12**

**Пример моделирования простого протокола передачи данных**

Ощепков Дмитрий Владимирович НФИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Определим декларации</b>	<b>9</b>
<b>5</b>	<b>Упражнение</b>	<b>11</b>
<b>6</b>	<b>Выводы</b>	<b>13</b>

# Список иллюстраций

3.1	граф системы . . . . .	8
5.1	Граф состояний . . . . .	11
5.2	Граф состояний . . . . .	12

## Список таблиц

# 1 Цель работы

Реализовать простой протокол передачи данных

## 2 Задание

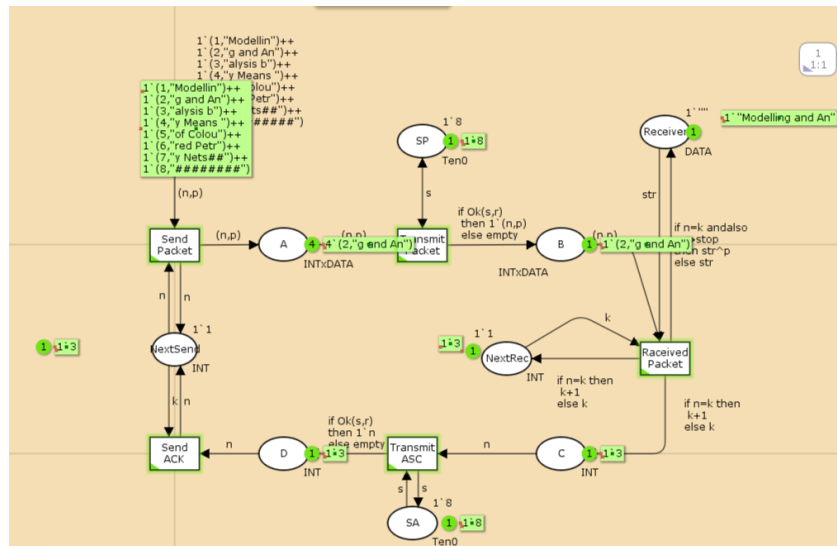
Рассмотрим ненадёжную сеть передачи данных, состоящую из источника, получателя. Перед отправкой очередной порции данных источник должен получить от получателя подтверждение о доставке предыдущей порции данных. Считаем, что пакет состоит из номера пакета и строковых данных. Передавать будем сообщение «Modelling and Analysis by Means of Coloured Petry Nets», разбитое по 8 символов.

### 3 Выполнение лабораторной работы

Основные состояния: источник (Send), получатель (Receiver). Действия (переходы): отправить пакет (Send Packet), отправить подтверждение (Send ACK). Промежуточное состояние: следующий посылаемый пакет (NextSend).

Зададим промежуточные состояния (A, B с типом INTxDATA, C, D с типом INTxDATA) для переходов (рис. 12.2): передать пакет Transmit Packet (передаём  $(n, p)$ ), передать подтверждение Transmit ACK (передаём целое число  $k$ ).

Таким образом, получим модель простого протокола передачи данных (рис. 12.3). Пакет последовательно проходит: состояние Send, переход Send Packet, состояние A, с некоторой вероятностью переход Transmit Packet, состояние B, попадает на переход Receive Packet, где проверяется номер пакета и если нет совпадения, то пакет направляется в состояние Received, а номер пакета передаётся последовательно в состояние C, с некоторой вероятностью в переход Transmit ACK, далее в состояние D, переход Receive ACK, состояние NextSend (увеличивая на 1 номер следующего пакета), переход Send Packet. Так продолжается до тех пор, пока не будут переданы все части сообщения. Последней будет передана стоппоследовательность.





## 4 Определим декларации

```
colset INT = int; colset DATA = string; colset INTxDATA = product INT * DATA; var n,  
k: INT; var p, str: DATA; val stop = "#####";  
colset Ten0 = int with 0..10; colset Ten1 = int with 0..10; var s: Ten0; var r: Ten1;
```

- ▼ Declarations
  - ▶ Standard priorities
  - ▶ Standard declarations
  - ▼ colset INT = int;
  - ▶ colset DATA
  - ▶ colset INTxDATA
  - ▶ var n k
  - ▶ var p str
  - ▶ val stop
  - ▶ colset Ten0
  - ▶ colset Ten1
  - ▶ var r
  - ▶ var s
  - ▶ fun Ok
- ▶ Monitors
  - main

## 5 Упражнение

Вычислите пространство состояний. Сформируйте отчёт о пространстве состояний и проанализируйте его. Постройте граф пространства состояний.

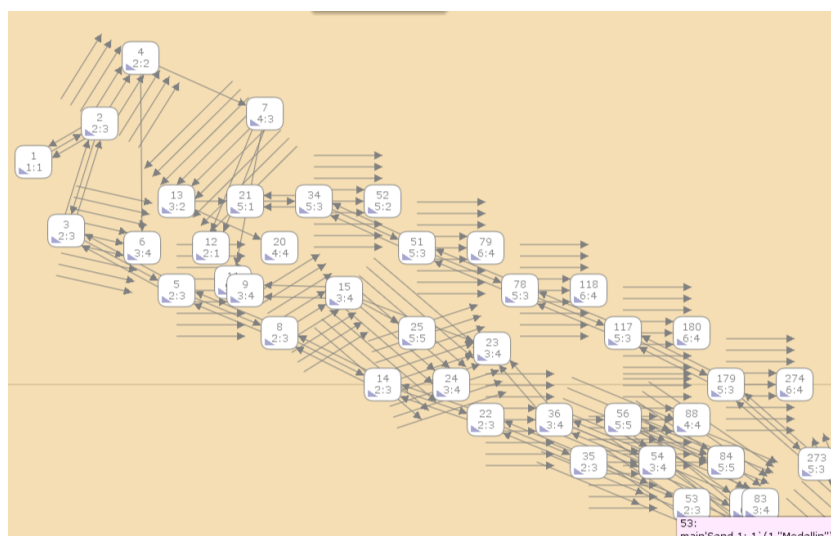


Рис. 5.1: Граф состояний

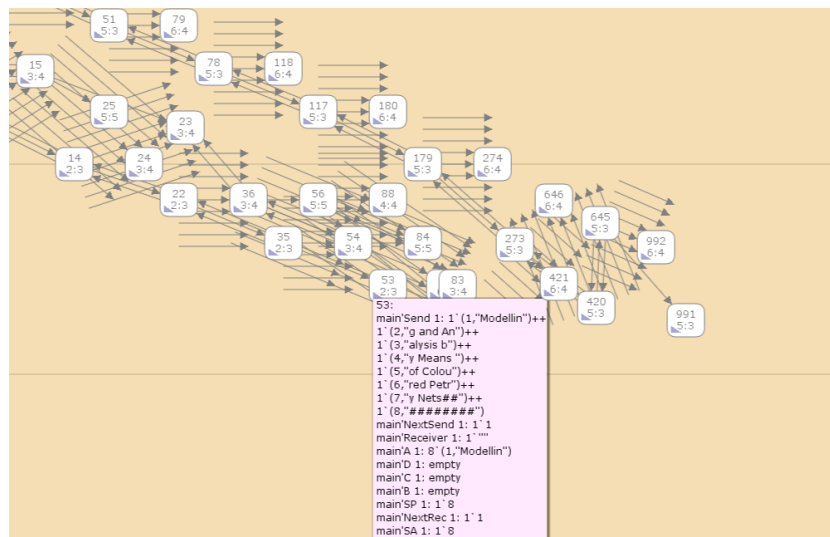


Рис. 5.2: Граф состояний

## 6 Выводы

Реализовал простой протокол передачи данных.