

Лабораторная работа №12

Имитационное моделирование

Серёгина Ирина Андреевна

Содержание

1	Цель работы	3
2	Задание	4
3	Теоретическое введение	5
4	Выполнение лабораторной работы	6
5	Выводы	15

1 Цель работы

Построить модель простого протокола передачи данных и ознакомиться с принципом её работы.

2 Задание

1. Построить модель простого протокола передачи данных.
2. Выполнить упражнение.

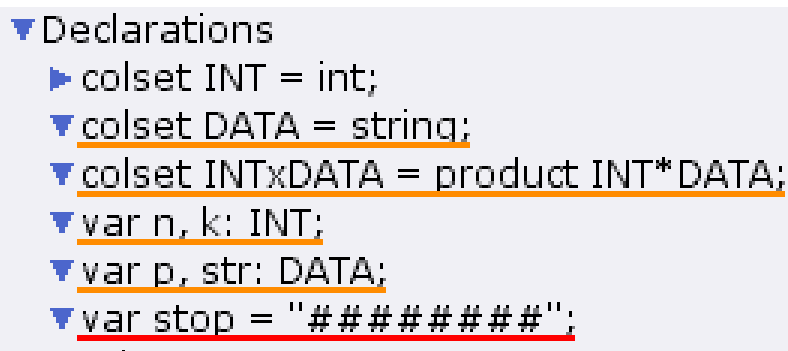
3 Теоретическое введение

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.

4 Выполнение лабораторной работы

Выполняю лабораторную работу с помощью утилиты `crntools`. Основные состояния: источник (Send), получатель (Receiver). Действия (переходы): отправить пакет (Send Packet), отправить подтверждение (Send ACK). Промежуточное состояние: следующий посылаемый пакет (NextSend)

Зададим декларации модели (рис. 4.1).



```
▼ Declarations
  ► colset INT = int;
  ▼ colset DATA = string;
  ▼ colset INTxDATA = product INT*DATA;
  ▼ var n, k: INT;
  ▼ var p, str: DATA;
  ▼ var stop = "#####";
```

Рис. 4.1: Декларации модели

Стоповый байт ("#####") определяет, что сообщение закончилось. Состояние Receiver имеет тип DATA и начальное значение 1"" (т.е. пустая строка, поскольку состояние собирает данные и номер пакета его не интересует). Состояние NextSend имеет тип INT и начальное значение 11. Поскольку пакеты представляют собой кортеж, состоящий из номера пакета и строки, то выражение у двусторонней дуги будет иметь значение (n,p). Кроме того, необходимо взаимодействовать с состоянием, которое будет сообщать номер следующего посылаемого пакета данных. Поэтому переход Send Packet соединяем с состоянием NextSend двумя дугами с выражениями n (рис. 4.2).

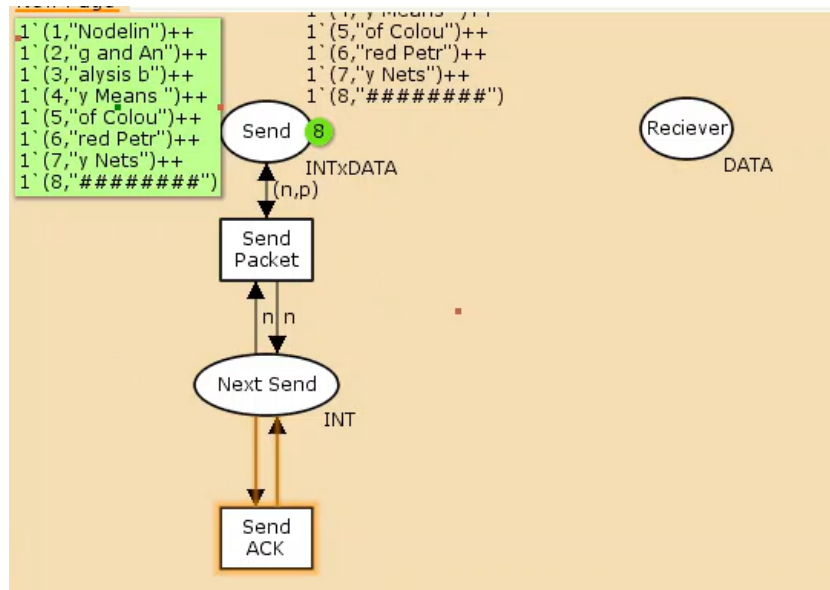


Рис. 4.2: Начальный граф

Зададим промежуточные состояния (А, В с типом INTxDATA, С, D с типом INTxDATA) для переходов: передать пакет Transmit Packet (передаём (n,p)), передать подтверждение Transmit ACK (передаём целое число k). Добавляем переход получения пакета (Receive Packet). От состояния Receiver идёт дуга к переходу Receive Packet со значением той строки (str), которая находится в состоянии Receiver. Обратно: проверяем, что номер пакета новый и строка не равна стоп-биту. Если это так, то строку добавляем к полученным данным. Кроме того, необходимо знать, каким будет номер следующего пакета. Для этого добавляем состояние NextRec с типом INT и начальным значением 1'1 (один пакет), связываем его дугами с переходом Receive Packet. Причём к переходу идёт дуга с выражением k, от перехода — if n=k then k+1 else k. Связываем состояния В и С с переходом Receive Packet. От состояния В к переходу Receive Packet — выражение (n,p), от перехода Receive Packet к состоянию С — выражение if n=k then k+1 else k. От перехода Receive Packet к состоянию Receiver: if n=k andalso p<>stop then str^p else str (рис. 4.3).

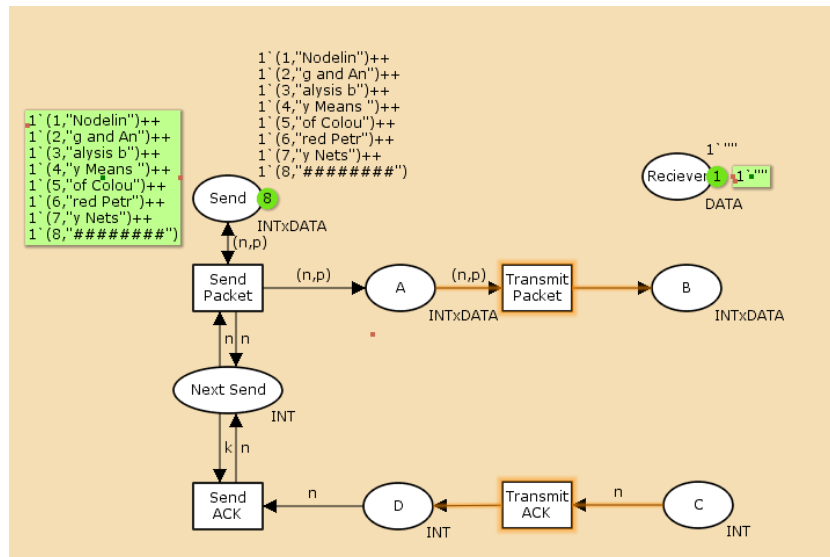


Рис. 4.3: Граф с промежуточными состояниями

Задаем дополнительные декларации и функцию (рис. 4.4).

```

▼ colset Ten0 = int with 0..10;
▼ colset Ten1 = int with 0..10;
▼ var s: Ten0;
▼ var r: Ten1;
▼ fun Ok(s:Ten0, r:Ten1)=(r<=s);

```

Рис. 4.4: Дополнительные декларации модели

Задаём выражение от перехода Transmit Packet к состоянию B: if Ok(s,r) then 1(n,p) else empty
 Задаём выражение от перехода Transmit ACK к состоянию D: if Ok(s,r) then 1n else empty
 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 стоп- последовательность (рис. 4.5).

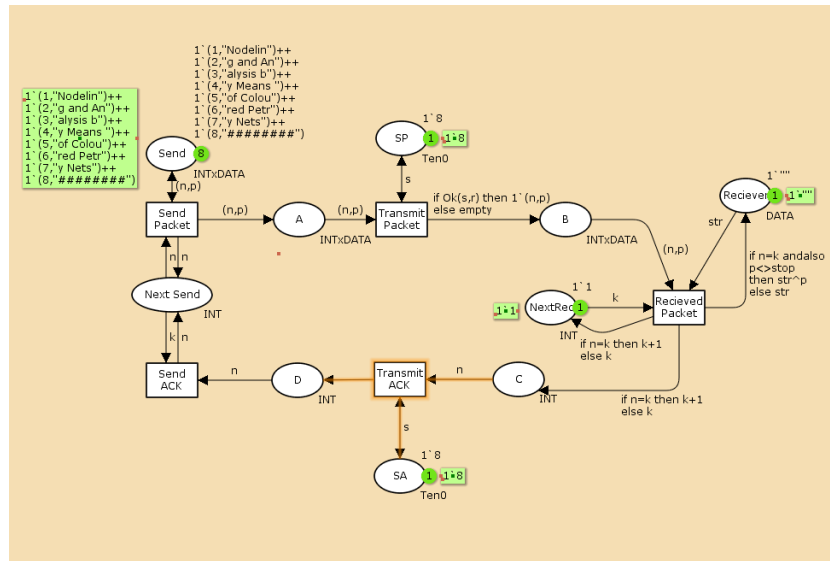


Рис. 4.5: Модель простого протокола передачи данных

Запускаем нашу модель и видим, что она работает (рис. 4.6).

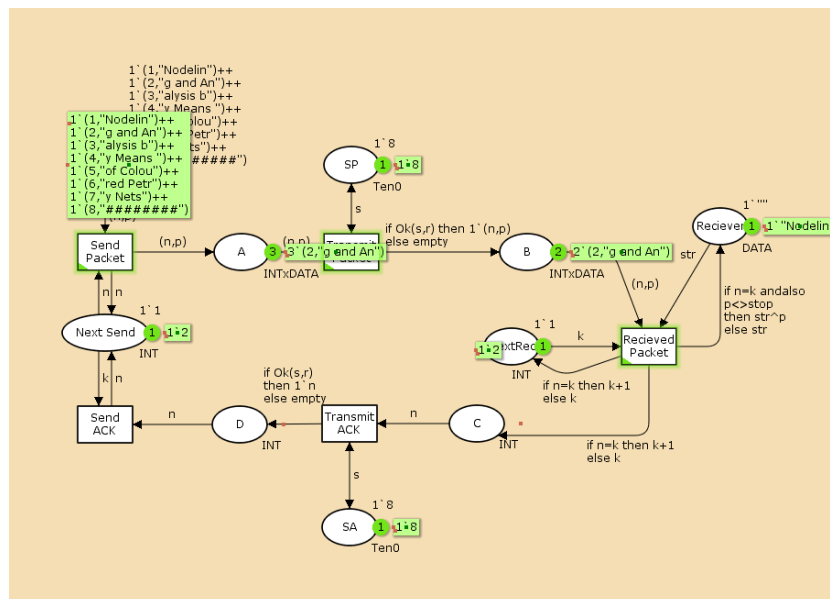


Рис. 4.6: Запуск модели

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

стояний (рис. 4.7).

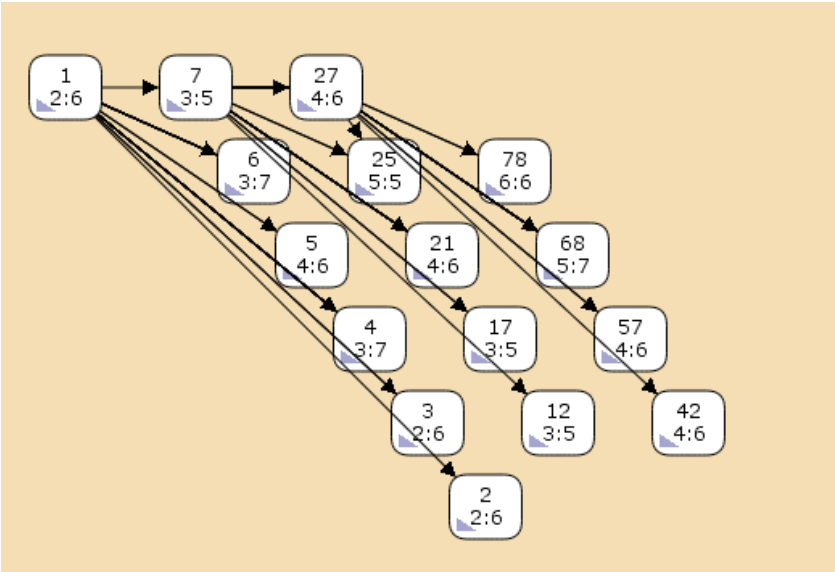


Рис. 4.7: Начало графа пространства состояний

CPN Tools state space report for:
/home/openmodelica/12 lab.cpn
Report generated: Sat Apr 26 14:27:41 2025

Statistics

State Space

Nodes: 21098
Arcs: 411044
Secs: 300
Status: Partial

Scc Graph

Nodes: 9653
Arcs: 328195

Secs: 9

Boundedness Properties

Best Integer Bounds

	Upper	Lower
New_Page'A 1	17	0
New_Page'B 1	9	0
New_Page'C 1	17	0
New_Page'D 1	12	0
New_Page'NextRec 1	1	1
New_Page'Next_Send 1	1	1
New_Page'Reciever 1	1	1
New_Page'SA 1	1	1
New_Page'SP 1	1	1
New_Page'Send 1	8	8

Best Upper Multi-set Bounds

New_Page'A 1	17` (8, "#####")
New_Page'B 1	1` (7, "y Nets")++
8` (8, "#####")	
New_Page'C 1	12`8++
6`9	
New_Page'D 1	12`8++
4`9	
New_Page'NextRec 1	1`8++
1`9	
New_Page'Next_Send 1	

```

1`8++
1`9
New_Page'Reciever 1 1`"Modeling and Analysis by Means of Coloured Petry Nets"
New_Page'SA 1      1`8
New_Page'SP 1      1`8
New_Page'Send 1    1`(1,"Nodelin")++
1`(2,"g and An")++
1`(3,"alysis b")++
1`(4,"y Means ")++
1`(5,"of Colou")++
1`(6,"red Petr")++
1`(7,"y Nets")++
1`(8,"#####")

```

Best Lower Multi-set Bounds

```

New_Page'A 1      empty
New_Page'B 1      empty
New_Page'C 1      empty
New_Page'D 1      empty
New_Page'NextRec 1 empty
New_Page'Next_Send 1
                    empty
New_Page'Reciever 1 1`"Modeling and Analysis by Means of Coloured Petry Nets"
New_Page'SA 1      1`8
New_Page'SP 1      1`8
New_Page'Send 1    1`(1,"Nodelin")++
1`(2,"g and An")++
1`(3,"alysis b")++
1`(4,"y Means ")++

```

1` (5,"of Colou")++

1` (6,"red Petr")++

1` (7,"y Nets")++

1` (8,"#####")

Home Properties

Home Markings

None

Liveness Properties

Dead Markings

6023 [21098,21097,21096,21095,21094,...]

Dead Transition Instances

None

Live Transition Instances

None

Fairness Properties

New_Page'Recieved_Packet 1

No Fairness

New_Page'Send_ACK 1 No Fairness

New_Page'Send_Packet 1 Impartial

New_Page'Transmit_ACK 1

No Fairness

New_Page'Transmit_Packet 1

Impartial

5 Выводы

Я построила модель простого протокола передачи данных и ознакомилась с принципом её работы.