Лабораторная работа №12

Имитационное моделирование

Серёгина Ирина Андреевна

Содержание

# 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 cpntools. 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. 1).

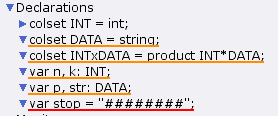


Рис. 1: Декларации модели

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. 2).

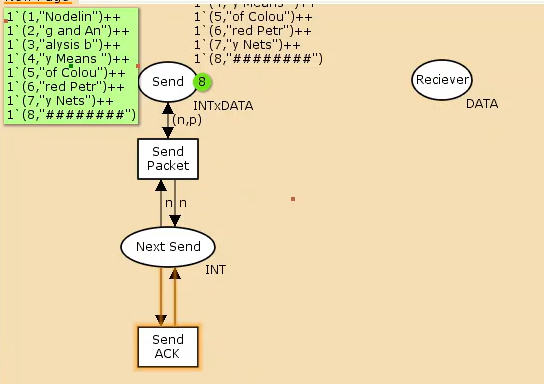


Рис. 2: Начальный граф

Зададим промежуточные состояния (A, B с типом INTxDATA, C, 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 B и C с переходом Receive Packet. От состояния B к переходу Receive Packet — выражение (n,p), от перехода Receive Packet к состоянию C — 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. 3).

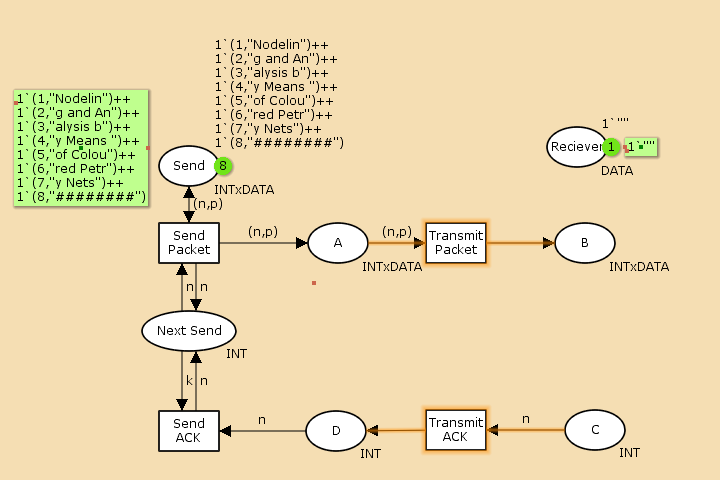


Рис. 3: Граф с промежуточными состояниями

Задаем дополнительные декларации и функцию (рис. 4).

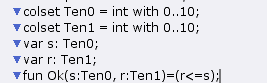


Рис. 4: Дополнительные декларации модели

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. 5).

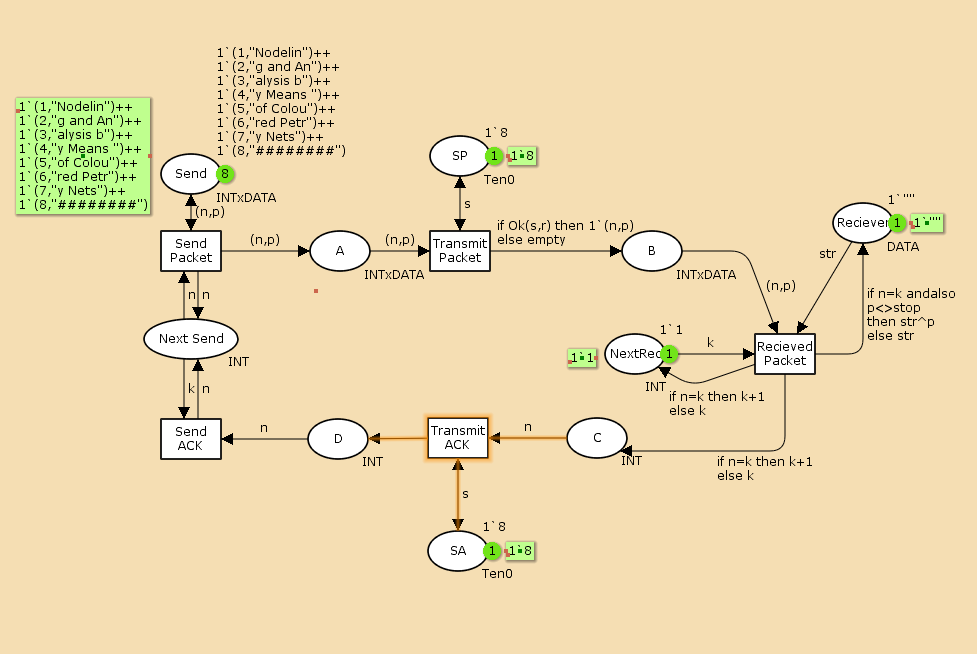


Рис. 5: Модель простого протокола передачи данных

Запускаем нашу модель и видим, что она работает (рис. 6).

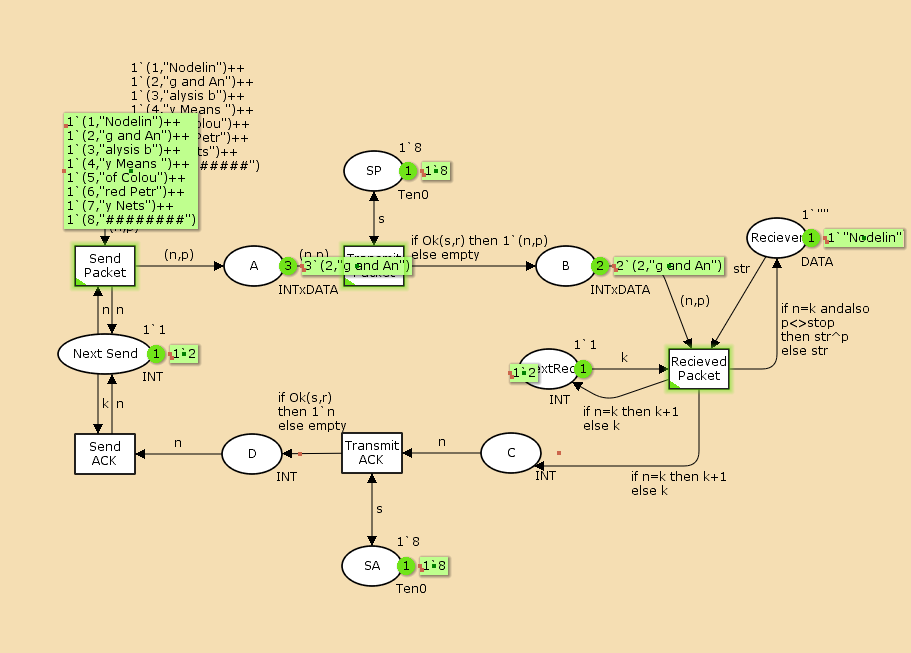


Рис. 6: Запуск модели

Упражнение. Вычислите пространство состояний. Сформируйте отчёт о пространстве состояний и проанализируйте его. Постройте граф пространства состояний (рис. 7).

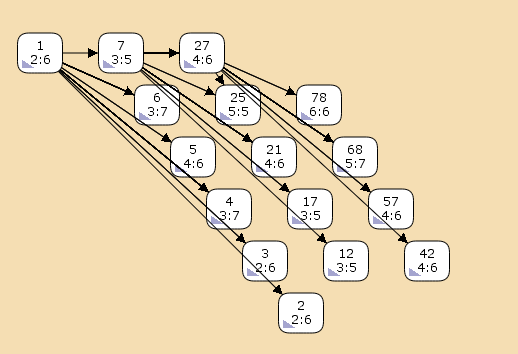


Рис. 7: Начало графа пространства состояний

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`"Nodeling 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`"Nodeling 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.