Лабораторная работа №12

Пример моделирования простогомпротокола передачи данных

Клюкин Михаил Александрович

Содержание

# 1 Цель работы

Реализовать простой протокол передачи данных в CPN Tools.

# 2 Задание

1. Реализовать простой протокол передачи данных в CPN Tools.
2. Вычислить пространство состояний, сформировать отчет о нем и построить граф.

# 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 декларации мод/ели (рис. 1).

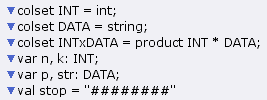


Рис. 1: Задание деклараций

Состояние Send имеет тип INTxDATA и следующую начальную маркировку (в соответствии с передаваемой фразой).

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 1’1. 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. Также необходимо получать информацию с подтверждениями о получении данных. От перехода Send Packet к состоянию NextSend дуга с выражением n, обратно — k.

Построим начальный граф (рис. 2).

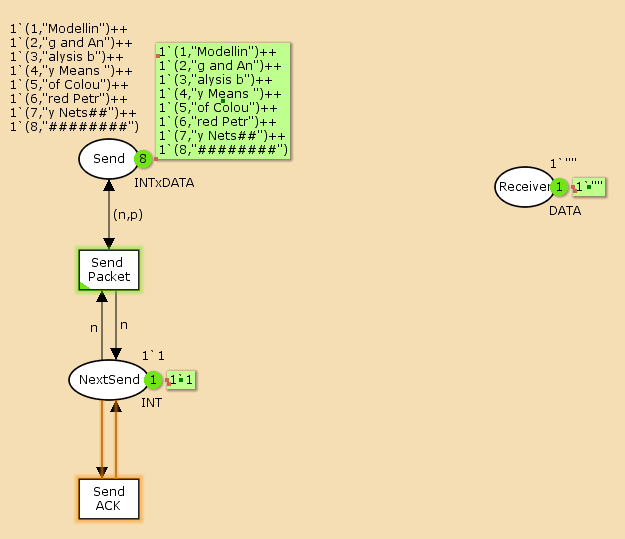


Рис. 2: Начальный граф

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 (если n=k и мы не получили стоп-байт, то направляем в состояние строку и к ней прикрепляем p, в противном случае посылаем толко строку). На переходах Transmit Packet и Transmit ACK зададим потерю пакетов. Для этого на интервале от 0 до 10 зададим пороговое значение и, если передаваемое начение превысит этот порог, то считаем, что произошла потеря пакета, если нет, то передаём пакет дальше. Для этого задаём вспомогательные состояния SP и SA с типом Ten0 и начальным значением 1’8, соединяем с соответствующими переходами (рис. 3).

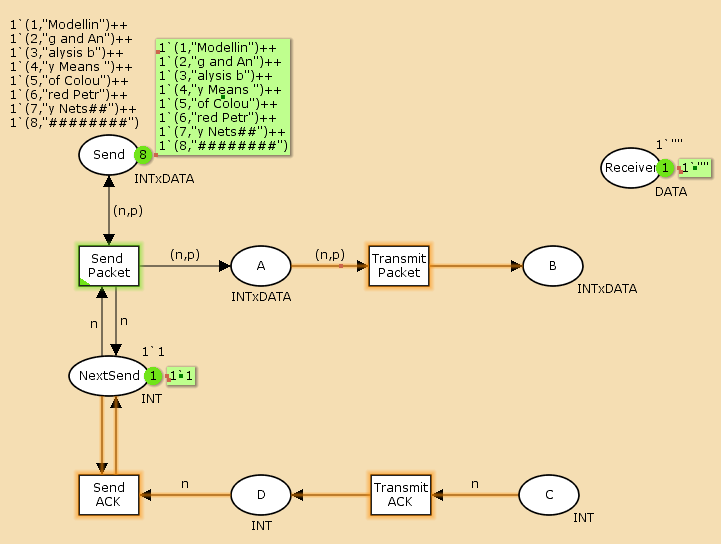


Рис. 3: Добавление промежуточных состояний

В декларациях задаём (рис. 4).

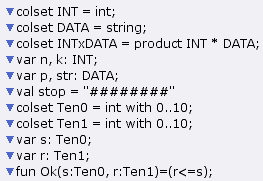


Рис. 4: Задание деклараций

Таким образом, получим модель простого протокола передачи данных. 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).

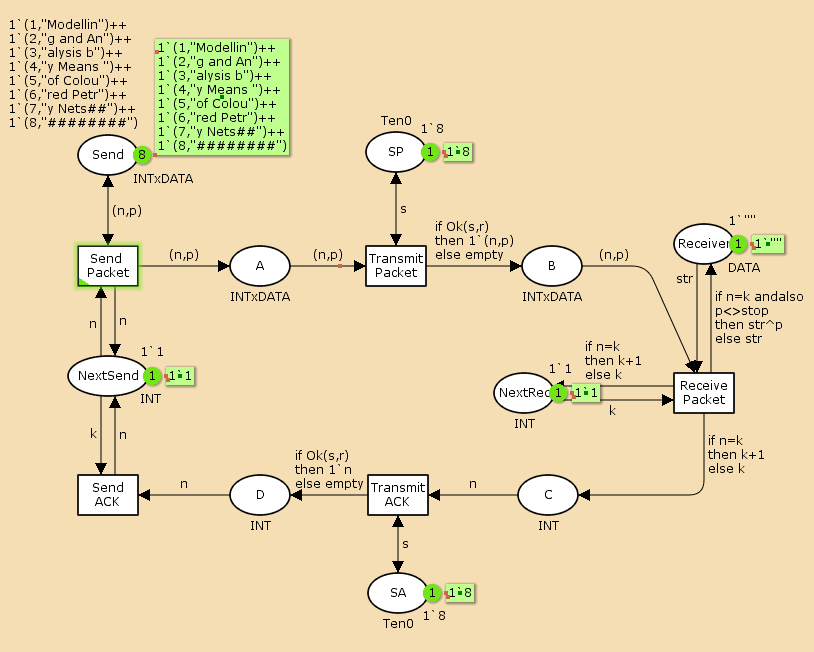


Рис. 5: Модель простого протокола передачи данных

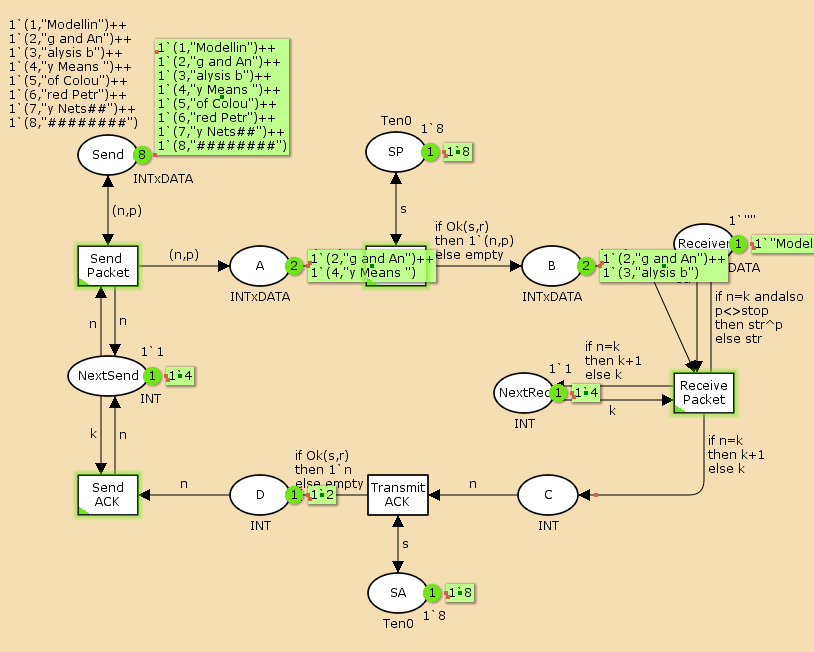


Рис. 6: Запуск модели простого протокола передачи данных

# 4 Упражнение

Вычислим пространство состояний. Для этого сформируем код пространства состояний, который создается, когда используется инструмент “Войти в пространство состояний системы”. Сформируем отчет о пространстве состояний и проанализируем его.

CPN Tools state space report for:  
/home/openmodelica/12.cpn  
Report generated: Sat Apr 26 11:03:24 2025  
  
  
 Statistics  
------------------------------------------------------------------------  
  
 State Space  
 Nodes: 11109  
 Arcs: 167988  
 Secs: 300  
 Status: Partial  
  
 Scc Graph  
 Nodes: 5833  
 Arcs: 138605  
 Secs: 32  
  
  
 Boundedness Properties  
------------------------------------------------------------------------  
  
 Best Integer Bounds  
 Upper Lower  
 Net'A 1 19 0  
 Net'B 1 9 0  
 Net'C 1 6 0  
 Net'D 1 5 0  
 Net'NextRec 1 1 1  
 Net'NextSend 1 1 1  
 Net'Receiver 1 1 1  
 Net'SA 1 1 1  
 Net'SP 1 1 1  
 Net'Send 1 8 8  
  
 Best Upper Multi-set Bounds  
 Net'A 1 19`(1,"Modellin")++  
15`(2,"g and An")++  
10`(3,"alysis b")++  
5`(4,"y Means ")  
 Net'B 1 9`(1,"Modellin")++  
7`(2,"g and An")++  
5`(3,"alysis b")++  
2`(4,"y Means ")  
 Net'C 1 6`2++  
5`3++  
3`4++  
1`5  
 Net'D 1 5`2++  
3`3++  
2`4++  
1`5  
 Net'NextRec 1 1`1++  
1`2++  
1`3++  
1`4++  
1`5  
 Net'NextSend 1 1`1++  
1`2++  
1`3++  
1`4++  
1`5  
 Net'Receiver 1 1`""++  
1`"Modellin"++  
1`"Modelling and An"++  
1`"Modelling and Analysis b"++  
1`"Modelling and Analysis by Means "  
 Net'SA 1 1`8  
 Net'SP 1 1`8  
 Net'Send 1 1`(1,"Modellin")++  
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  
 Net'A 1 empty  
 Net'B 1 empty  
 Net'C 1 empty  
 Net'D 1 empty  
 Net'NextRec 1 empty  
 Net'NextSend 1 empty  
 Net'Receiver 1 empty  
 Net'SA 1 1`8  
 Net'SP 1 1`8  
 Net'Send 1 1`(1,"Modellin")++  
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  
 3910 [9999,9998,9997,9996,9995,...]  
  
 Dead Transition Instances  
 None  
  
 Live Transition Instances  
 None  
  
  
 Fairness Properties  
------------------------------------------------------------------------  
 Net'Receive\_Packet 1 No Fairness  
 Net'Send\_ACK 1 No Fairness  
 Net'Send\_Packet 1 Impartial  
 Net'Transmit\_ACK 1 No Fairness  
 Net'Transmit\_Packet 1 Impartial

Из отчета можно видеть:

* 11109 состояний и 167988 переходов между ними,
* границы значений для каждого элемента: промежуточные состояния A, B, вспомогатальные состояния SP, SA, NextRex, NextSend, Reciever, Send,
* границы в виде мультимножеств,
* маркировка home для всех состояний,
* маркировка dead равная 4675 – состояния, в которых нет включенных переходов.

Сформируем граф пространства состояний (рис. 7).

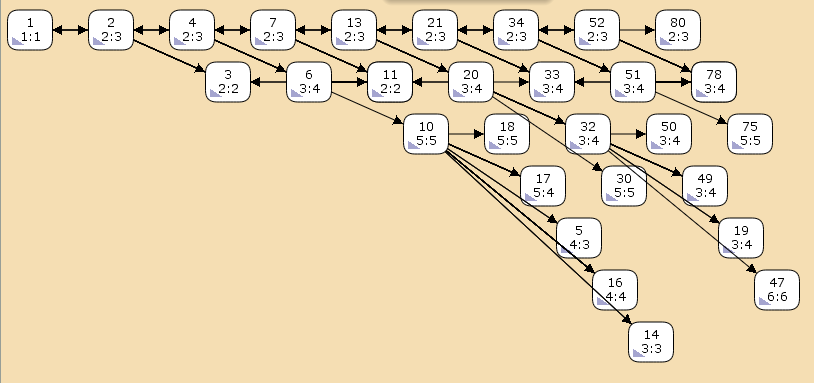


Рис. 7: Пространство состояний для модели простого протокола передачи данных

# 5 Выводы

В процессе выполнения лабораторной работы реализовали простой протокол передачи данных в CPN Tools, провели анализ его пространства состояний.

# Список литературы