

Лабораторная работа №12

Дисциплина: Имитационное моделирование

Пронякова Ольга Максимовна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	Декларации	7
2.2	Модель простого протокола передачи данных	9
2.3	запуск Модели простого протокола передачи данных	10
2.4	Отчет о пространстве состояний	11

Список таблиц

1 Цель работы

Рассмотреть ненадёжную сеть передачи данных, состоящую из источника, получателя.

2 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 модели(рис.2.1).

```

▼ Declarations
  ▼ colset INT = int;
  ► colset DATA
  ► colset INTxDATA
  ► var n k
  ► var p str
  ▼ val stop = "#####";
  ▼ colset Ten0 = int with 0..10
  ► colset Ten1
  ▼ var s: Ten0;
  ▼ var r: Ten1;
  ▼ fun Ok(s:Ten0, r:Ten1)=(r<=
► Monitors
Modeling

```

Рис. 2.1: Декларации

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

стро- ки, 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 .

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 11 (один пакет), связываем его дугами с переходом Receive Packet. Причём к переходу идёт дуга с выражением k , от перехода — $\text{if } n=k \text{ then } k+1 \text{ else } k$. 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 $\text{if } n=k \text{ then } k+1 \text{ else } k$. От перехода Receive Packet к состоянию Receiver: $\text{if } n=k \text{ and also } p \neq \text{stop then } str \wedge p \text{ else } str$ (если $n=k$ и мы не получили стоп-байт, то направляем в состояние строку и к ней прикрепляем p , в противном случае посылаем только строку). На переходах Transmit Packet и Transmit ACK зададим потерю пакетов. Для этого на интервале от 0 до 10 зададим пороговое значение i , если передаваемое значение 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 18, соединяем с соответствующими переходами.(рис.2.2).

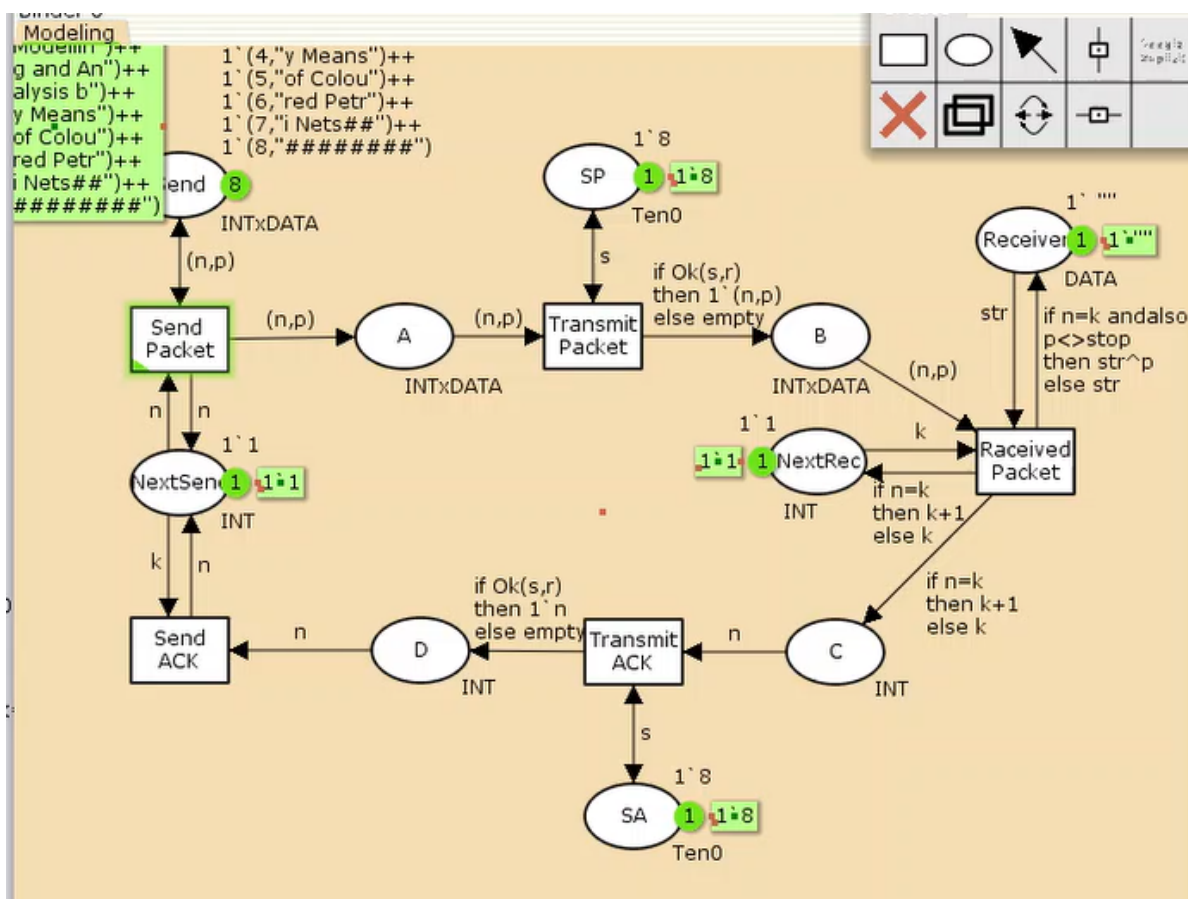


Рис. 2.2: Модель простого протокола передачи данных

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.2.3).

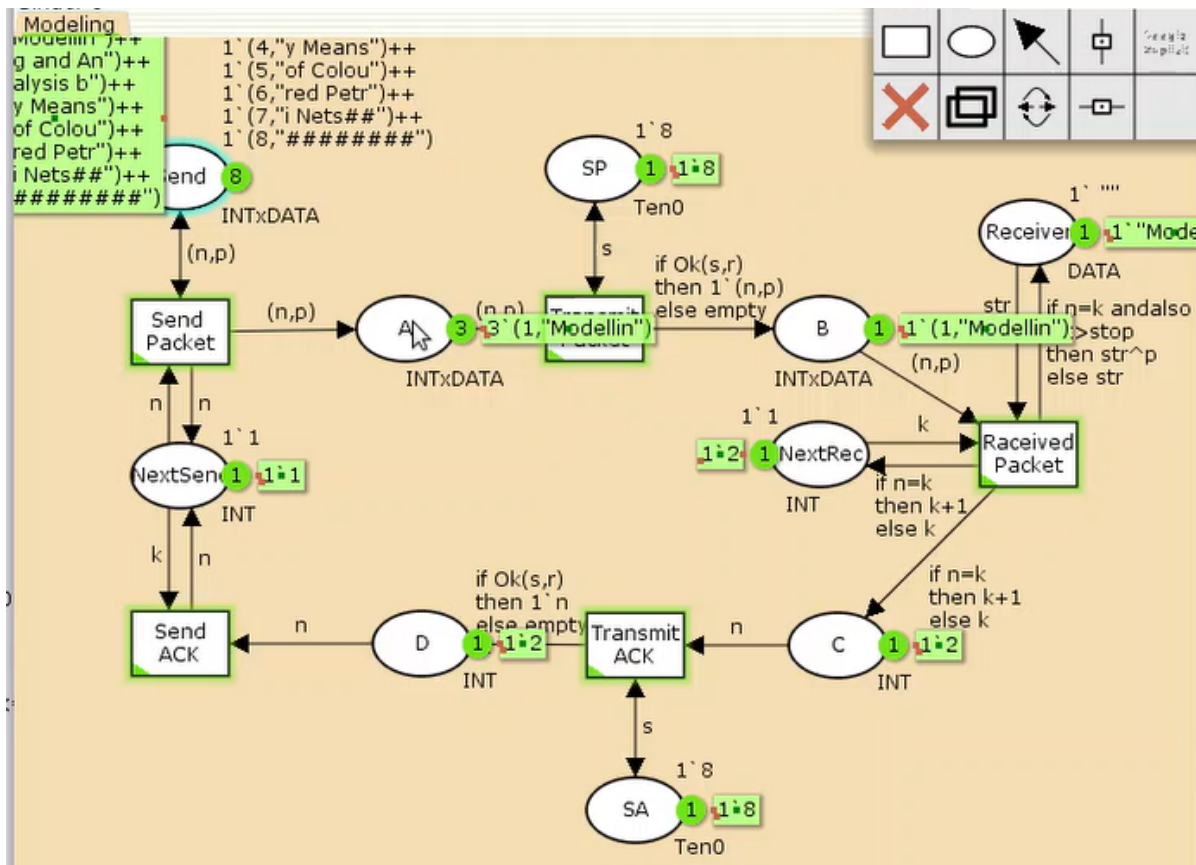


Рис. 2.3: запуск Модели простого протокола передачи данных

Сформировала отчет о пространстве состояний(рис.2.4).

Файл Правка Поиск Вид Документ Справка		
CPN Tools state space report for: <unsaved net> Report generated: Thu Apr 24 21:20:04 2025		
Statistics		

State Space		
Nodes:	16587	
Arcs:	257020	
Secs:	300	
Status:	Partial	I
Scc Graph		
Nodes:	8724	
Arcs:	213233	
Secs:	14	
Boundedness Properties		

Best Integer Bounds		
	Upper	Lower
Modeling'A 1	20	0
Modeling'B 1	10	0
Modeling'C 1	6	0
Modeling'D 1	5	0
Modeling'NextRec 1	1	1
Modeling'NextSend 1	1	1
Modeling'Receiver 1	1	1
Modeling'SA 1	1	1
Modeling'SP 1	1	1
Modeling'Send 1	8	8
Best Upper Multi-set Bounds		
Modeling'A 1	20` (1,"Modellin")++	
16` (2,"g and An")++		
11` (3,"alysis b")++		
6` (4,"y Means")++		
1` (5,"of Colou")		
Modeling'B 1	10` (1,"Modellin")++	
8` (2,"g and An")++		
5` (3,"alysis b")++		
3` (4,"y Means")		
Modeling'C 1	6` 2++	
5` 3++		
3` 4++		
2` 5		

Рис. 2.4: Отчет о пространстве состояний

3 Выводы

Рассмотрела ненадёжную сеть передачи данных, состоящую из источника, получателя.

Список литературы