Лабораторная работа 12

Пример моделирования простого протокола передачи данных

Акопян Сатеник

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Смоделировать простой протокол передачи данных.

2 Задание

Рассмотрим ненадёжную сеть передачи данных, состоящую из источника, получате- ля. Перед отправкой очередной порции данных источник должен получить от полу- чателя 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

1. Зададим декларации модели (рис. 3.1).

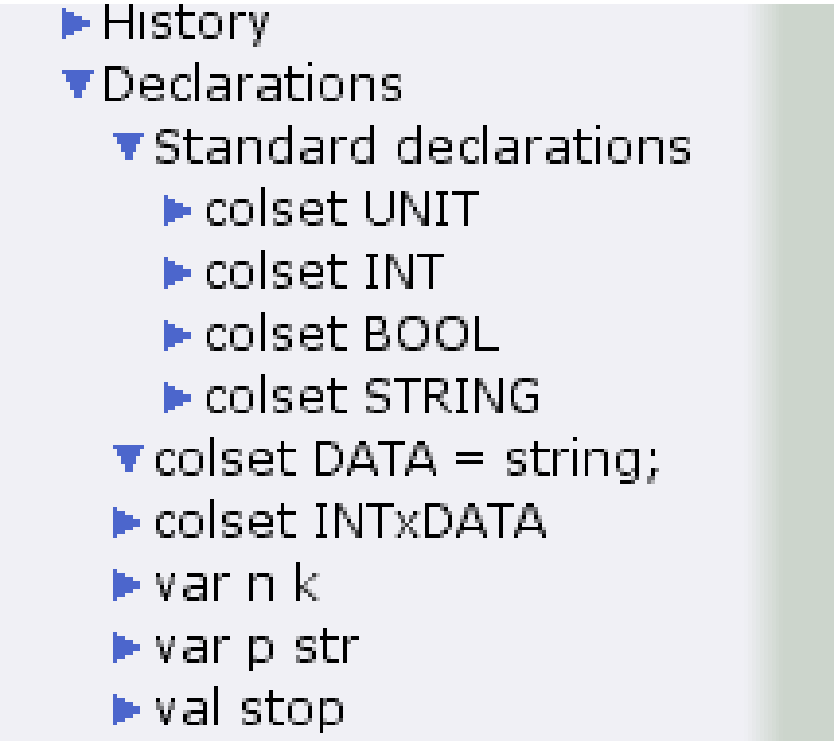

Рис. 3.1: декларации модели

2. 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 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 (рис. 12.1). Также необходимо получать информацию с подтверждениями о получении дан- ных. От перехода Send Packet к состоянию NextSend дуга с выражением n, обратно — k. (рис. 3.2)

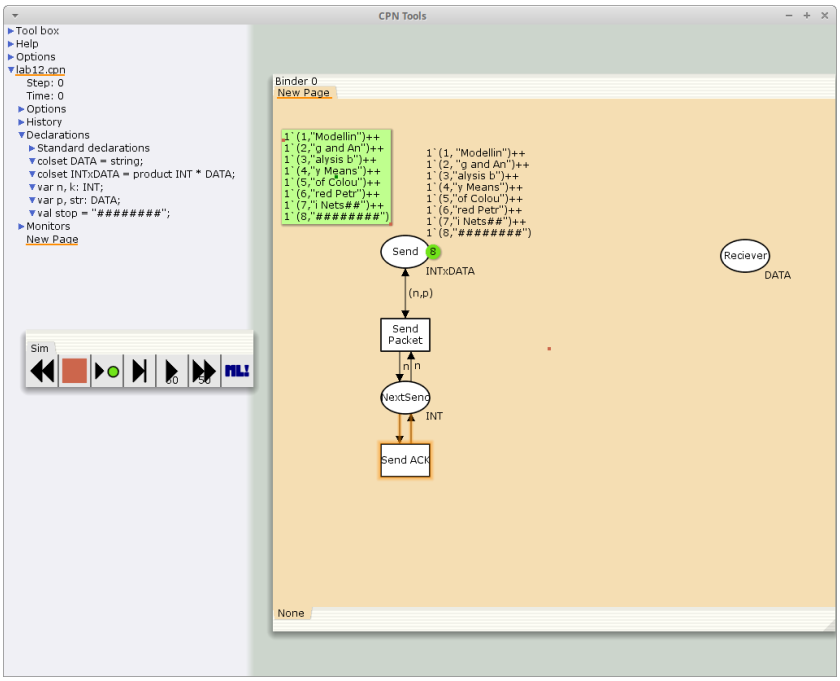

Рис. 3.2: Начальный граф

3. 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). 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

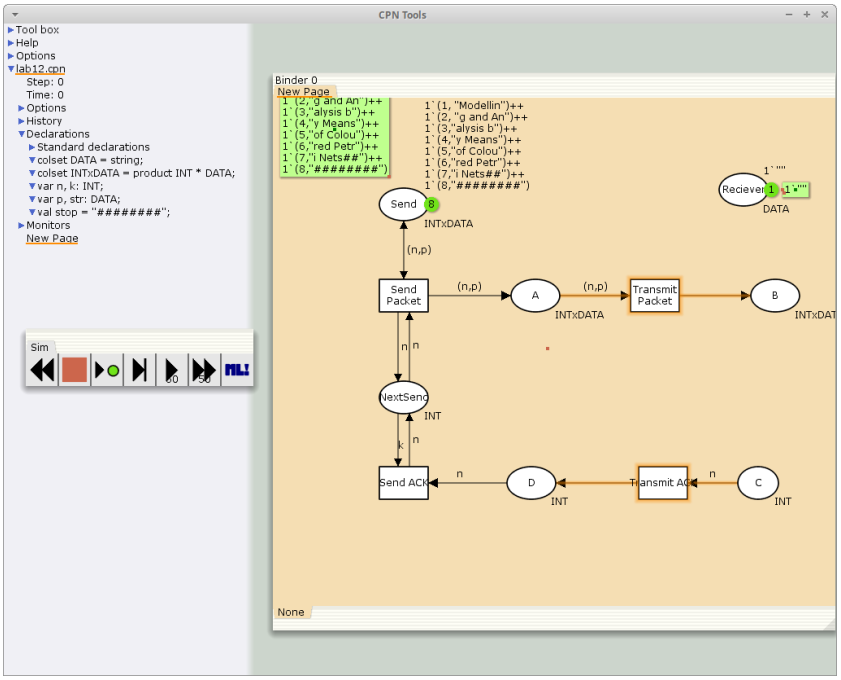

Рис. 3.3: Добавление промежуточных состояний

4. На переходах Transmit Packet и Transmit ACK зададим потерю пакетов. Для этого на интервале от 0 до 10 зададим пороговое значение и, 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 1‘8, соединяем с соответствующими переходами.

В декларациях задаём:

```
colset Ten0 = int with 0..10;
colset Ten1 = int with 0..10;
```

```
var s: Ten0;
var r: Ten1;
```

(рис. 3.4)

Таким образом, получим модель простого протокола передачи данных (рис. 3.5).

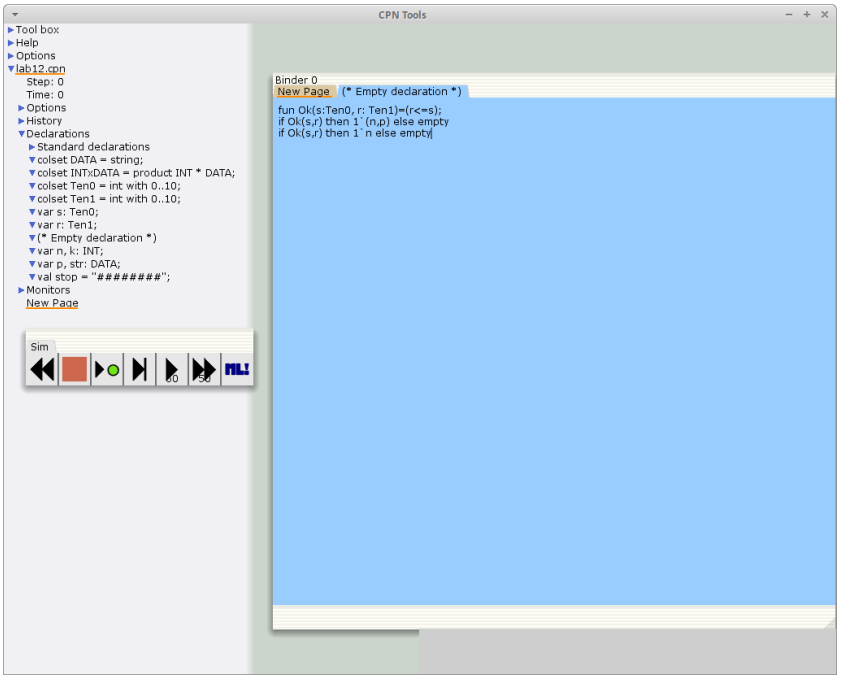

Рис. 3.4: Декларации модели

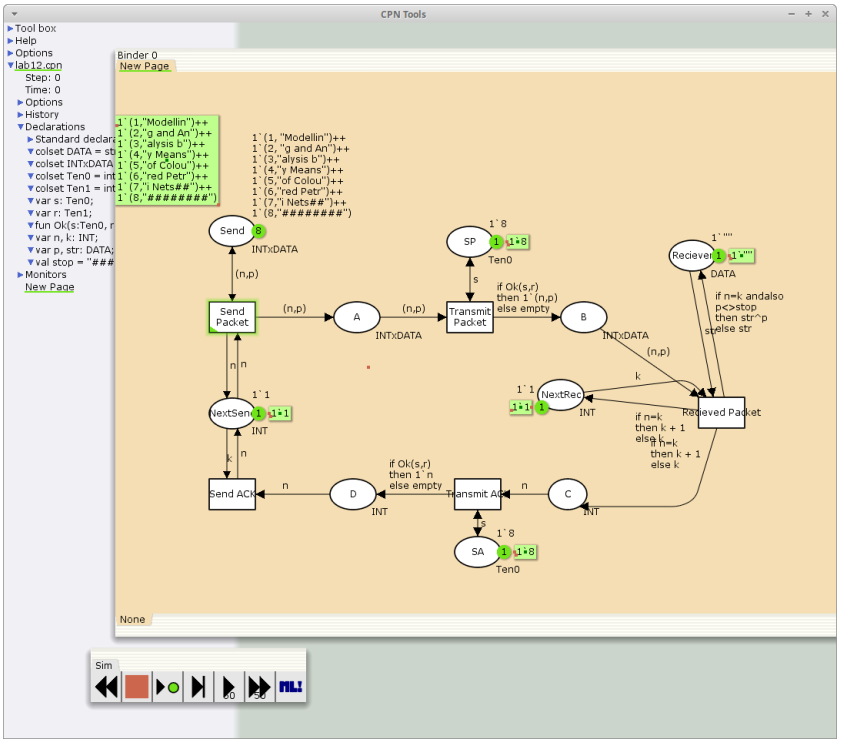

Рис. 3.5: Модель простого протокола передачи данных

4 Выводы

В результате был смоделирован простой протокол передачи данных.

Список литературы