

Моделирование и анализ свойств протоколов распространения данных в распределенных системах с использованием раскрашенных сетей Петри

В. Г. Агаджанова, Д. Ю. Чалый

*Ярославский государственный университет им. П. Г. Демидова
E-mail: agadvika@yandex.ru, chaly@uniyar.ac.ru*

Формализм раскрашенных сетей Петри и их применение является актуальным направлением исследований в современной науке, что подтверждается наличием большого количества статей по этой тематике. Многие авторы этих статей используют CPNTools в качестве инструментального средства моделирования и анализа построенных моделей. В нашей статье рассматриваются протоколы из семейства Gossip-протоколов, которые широко используются для распространения информации в современных распределенных системах.

Основным результатом работы является оригинальная модель протокола и анализ ее функционирования на примере нескольких топологий коммуникационных сетей.

Ключевые слова: раскрашенные сети Петри, сети Петри, CPNTools, Gossip-протокол, Gossip-based protocol, Coloured Petri Nets.

Введение

Одним из ярких примеров применения сетей Петри является работа [1]. В ней на простом модельном примере рассмотрено применение вложенных сетей Петри для моделирования распределенных систем. Авторами было определено расширение формализма вложенных сетей Петри за счет приписывания некоторым переходам дополнительных условий срабатывания. Приведены условия, при которых использование условий срабатывания на переходах сохраняет свойства базовой модели, в частности разрешимость некоторых поведенческих свойств.

В работе [2] авторы предлагают проводить имитационное моделирование и построение графа достижимости для NP-сетей путем перевода NP-сетей в раскрашенные сети Петри и использования инструментария CPNTools в качестве виртуальной машины для исполнения и средства автоматического анализа исходных NP-сетей.

В работе [3] авторы промоделировали и проанализировали протокол управления передачей (TCP) с помощью раскрашенных сетей Петри (CPN). Они представили свою CPN-модель и примеры того, как можно изучить проблемы производительности протокола TCP. Показали, как можно модифицировать эту модель, чтобы представить протокол ARTCP. Авторы утверждают, что их модель может использоваться в качестве основы для построения формальных моделей будущих модификаций протокола TCP.

В работе [6] авторы поставили перед собой цель изучить полезность раскрашенных сетей Петри (CPN) и CPNTools для анализа безопасности защищенного модуля Trusted Platform Module (TPM). В качестве примеров они взяли Object-specific authorization protocol (OSAP) и Session Key Authorization Protocol (SKAP).

Целью работы [7] являлась разработка комплексной модели связи для сети взаимодействующих спутников, при использовании сетей Петри, позволяющих реконфигурировать подсети в тот момент, когда появляются спутниковые ошибки. Для моделирования взаимодействия спутников в сети использовались раскрашенные сети Петри (CPN).

В статье [8] описано применение моделирования многоагентных систем. Авторы используют многоуровневый подход, основанный на раскрашенных сетях Петри для моделирования сложных одновременных разговоров среди агентов в многоагентных системах.

Все это демонстрирует, что проблематика коммуникационных протоколов, их моделирование и анализ являются актуальной проблемой, а раскрашенные сети Петри представляют собой подходящий формализм для проведения анализа этих объектов.

Протокол Gossip

Протокол Gossip — это протокол для распределенных систем, состоящих из равноправных узлов. Данный протокол используют в распределенных системах потому, что его специфика проста и он имеет удобную структуру, в отличие от ряда других

протоколов распределенных систем. Чаще всего алгоритм данного протокола конструируют таким образом, что узлы взаимодействуют главным образом с соседними узлами и лишь иногда с узлами, которые находятся далеко. Этим протоколам необходимо обеспечить достаточную степень связности, чтобы избежать риска отключения узла от сети.

Gossiping — это альтернатива классическому методу отправки пакетов, который использует случайный выбор, чтобы снизить потребление ресурса в коммуникационной системе. Вместо того чтобы без разбора выполнять пересылку данных на всех своих соседей, узел передает данные только на одного случайно выбранного соседа. Если Gossiping-узел получает данные от соседа, он может переслать данные обратно соседу, если он случайным образом выбрал его.

Вариации протокола Gossip:

- Pushepidemic — узел передает имеющуюся информацию случайно выбранному соседу;
- Pullepidemic — узел запрашивает информацию у смежных узлов.

Модель протокола Gossip

Система CPNTools позволяет создавать иерархические модели, т. е. модель внутри модели. На рис. 1 видно, что в нашей модели содержатся еще три модели, они представлены на основной странице Top в виде переходов Init, Add и Check. В первом переходе происходит инициализация, во втором — добавление (при надобности) новых элементов в систему, в третьем — осуществляется проверка и передача информации узлам, которые ранее ею не обладали.

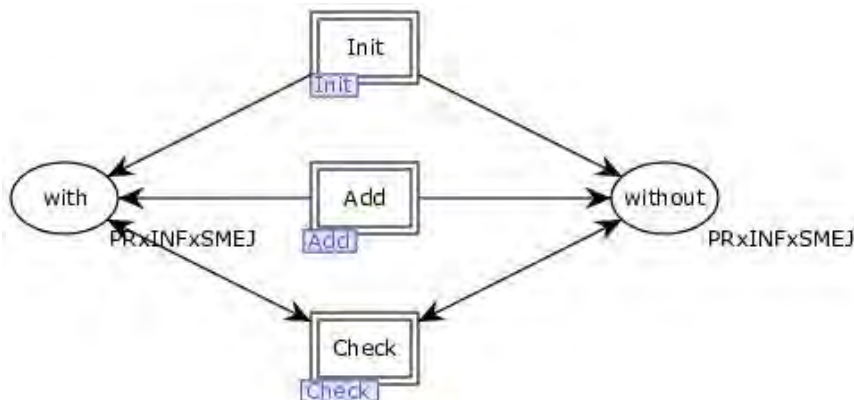


Рис. 1. Модель протокола Gossip

Как видим, позиции `with` и `without` имеют тип `PRxSMEJxINF` — он моделирует узел, список смежности и данные о наличии информации. Причем `with` содержит фишки, моделирующие узлы, обладающие информацией, а `without` — фишки, моделирующие узлы, не обладающие информацией. Рассмотрим подробнее подсеть, в которой происходит передача информации.

Основной подсетью, моделирующей обмен информацией, является `Check`, здесь происходит передача информации от узлов, содержащих информацию, к узлам, не содержащим информацию.

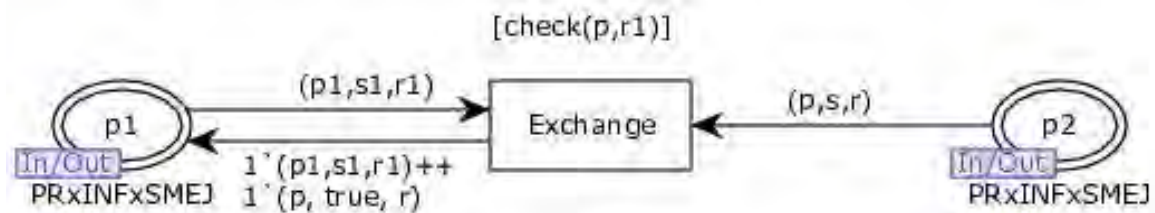


Рис. 2. Модель передачи информации от одного узла другому

Узел `p1` — это тот, в котором содержатся узлы, имеющие новую информацию. Узел `p2` — узел, в котором содержатся узлы, не содержащие новой информации. При срабатывании перехода `Exchange` мы вызываем функцию `check` (рис.3).

```

▼ fun check (p, r) =
  if r <> nil
  then (if hd r = p
        then true
        else check (p, tl r))
  else false

```

Сеть случайным образом выбирает один элемент из `p1` и один элемент из `p2` и далее проверяет, есть ли у элемента `p1` в списке смежных элемент `p2`. Этим занимается функция `check`.

Рис. 3. Функция `check`

Если она находит этот элемент, тогда у элемента из узла `p2` изменится значение `inf` на `true` и он переходит в узел `p1`. Если не находит, она берет следующий элемент.

Сеть продолжает свою работу до тех пор, пока все элементы из узла `p2` не станут обладать информацией и не перейдут в узел `p1`.

Рассмотрим две модификации модели. Модель со временем (рис. 4) и модель с ограниченным количеством передаваемых сообщений (рис. 5).

На рис. 4 представлена первая модификация нашей модели. О ней говорилось в пункте 2. С помощью этого улучшения можно теперь проследить, какое количество времени требуется программе, для того чтобы завершить процесс. В данном языке знак

@- является знаком времени. Время меняется в зависимости от задаваемых параметров. Время можно увидеть в списке деклараций, рядом с количеством шагов.

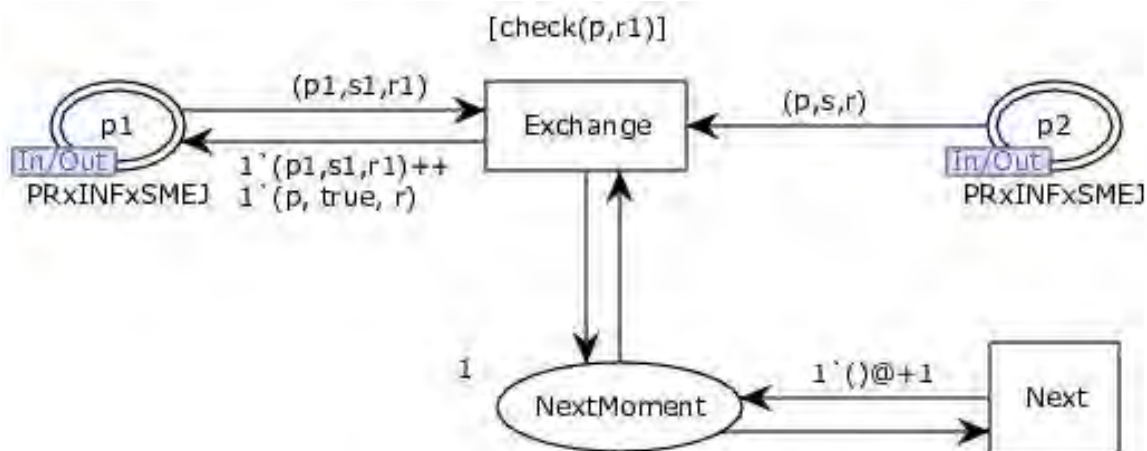


Рис. 4. Модификация модели

За определенный момент времени переход Exchange успевает сработать множество раз, при добавлении позиции NextMoment сеть сама решает, когда наступает следующий момент времени. В NextMoment из Exchange поступает фишка, и она дальше перемещается в переход Next. Значит, наступил другой момент времени, у перехода Next увеличивается счетчик, а фишка возвращается обратно в сеть. На рис. 5 мы видим модифицированную подсеть Check.

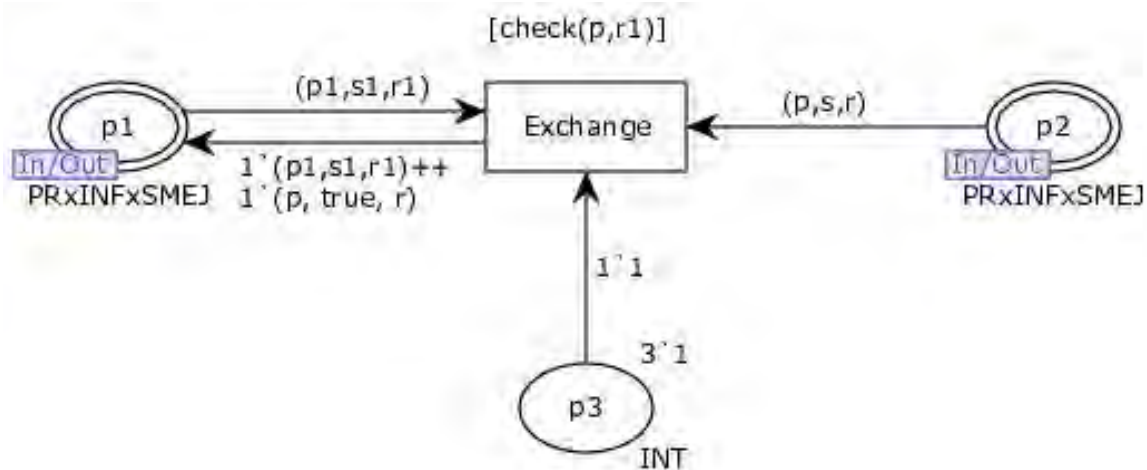


Рис. 5. Модификация модели с ограничением количества передаваемых сообщений

На данной подсети появилась новая позиция $p3$. Как видим, она имеет обычный числовой тип данных и имеет фишку 3^*1 . Это

означает, что переход Exchange сработает только 3 раза и затем сеть остановится. Поскольку сеть случайным образом выбирает узлы, которым необходимо передать информацию, при каждом срабатывании сети без информации оказываются различные узлы.

Заключение

В ходе работы в системе CPNTools была построена модель Gossip-протокола, который широко используется для распространения информации в современных распределенных системах. Данная модель была модифицирована до модели Gossip-протокола со временем, что позволило проанализировать модель по количеству затрачиваемого времени. Была сделана еще одна модификация модели, в которой вводились некоторые ограничения относительно количества передаваемых сообщений в системе. На примере нескольких топологий коммутируемых сетей построенная модель была проанализирована по многим параметрам, все результаты и анализ приведены в работе.

Ссылки

1. Ломазова И. А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. М.: Научный Мир, 2004. 208 с.

2. Dworzanski L., Lomazova I. CPN tools-assisted simulation and verification of nested Petri nets // Automatic Control and Computer Sciences. 2013. Volume 47. Issue 7. P. 393–402.

3. Chaly D., Sokolov V. An Extensible Coloured Petri Net Model of a Transport Protocol for Packet Switched Networks // Lecture Notes in Computer Science. 2003. Volume 2763. P. 66–75.

4. Чалый Д. Ю. Моделирование и анализ сетевых транспортных протоколов с помощью раскрашенных сетей Петри: дисс. ... канд. физ.-мат. наук. Ярославль, 2006. 148 с.

5. Башкин В. А. Функциональное программирование на языке SML: методические указания. Ярославль: ЯрГУ, 2007. 39 с.

6. Analysis of two authorization protocols using Colored Petri Nets / Y. Seifi, S. Suriadi, E. Foo, C. Boyd // International Journal of Information Security. 2014. Volume 14. Issue 3. P. 221–247.

7. Einafshar A., Razavi B., Sassani F. Integrated Reconfiguration of Multi-Satellite Network Communication Using Colored Petri Nets // Integrated Systems: Innovations and Applications. 2015. P. 3–28.

8. Ebadi T., Purvis M., Purvis M. A Colored Petri Net Model to Represent the Interactions between a Set of Cooperative Agents // *Lecture Notes in Computer Science*. 2012. Volume 6573. P. 141–152.

9. Дмитриев В. Н., Тушнов А. С., Сергеева Е. В. Имитационное моделирование системы мониторинга многозвенной сети передачи данных // *Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика*. 2013. № 2. С. 86–91.

УДК 519.681.5: 519.682

Тестирование библиотеки RPMlib на примере задачи о распределении тепла

С. С. Алексеев, В. В. Васильчиков

*Ярославский государственный университет им. П. Г. Демидова
E-mail: gwblaize@gmail.com, vasilch@uniyar.ac.ru*

В статье описывается параллельный алгоритм решения задачи о распределении тепла, ориентированный на использование библиотеки рекурсивно-параллельного программирования RPMlib, приводятся результаты численного эксперимента.

Ключевые слова: параллельные вычисления, рекурсия, .NET.

Введение. Цели исследования

В данной работе описывается параллельный алгоритм решения задачи о распределении тепла, ориентированный на использование библиотеки рекурсивно-параллельного (РП) программирования RPMlib, а также результаты его тестирования с использованием ее текущей версии [1, 2]. Устройство и основные возможности библиотеки RPMlib описаны в [3, 4], там же рассмотрены результаты ее тестирования на примере задачи о клике.

Эти результаты наглядно продемонстрировали возможность достижения очень значительного ускорения при работе в параллельном режиме за счет использования предлагаемых компонентов.

© Алексеев С. С., Васильчиков В. В., 2015