

Научная статья
УДК 004.942

DOI: 10.31799/2949-0693-2022-2-81-90

Применение сетей Петри для анализа таблиц расписания коммуникационных протоколов

Надежда Юрьевна Чумакова¹

✉ nadezhda.chumakova@guap.ru, orcid.org/0000-0002-8979-979X

Валентин Леонидович Оленев¹

valentin.olenev@guap.ru, orcid.org/0000-0002-1817-2754

¹ Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Российская Федерация

Аннотация. На работу бортовых вычислительных сетей напрямую влияет правильное распределение канальных ресурсов. Механизм планирования, реализуемый в различных коммуникационных протоколах, позволяет осуществлять данное распределение при помощи таблиц расписания. При проектировании бортовых сетей необходимо составить корректную таблицу расписания. Поэтому важными задачами являются анализ и проверка корректности таблиц расписания. В данной статье представлен подход к анализу и проверке таблиц расписания с применением такого инструмента исследования систем, как сети Петри.

Ключевые слова: бортовые сети, SpaceWire, коммуникационные протоколы, СТП-ИСС, сеть Петри, моделирование

Благодарности. Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, соглашение № FSRF-2020-0004 «Научные основы построения архитектур и систем связи бортовых информационно-вычислительных комплексов нового поколения для авиационных, космических систем и беспилотных транспортных средств».

Для цитирования: Чумакова Н. Ю., Оленев В. Л. Применение сетей Петри для анализа таблиц расписания коммуникационных протоколов // Инновационное приборостроение. 2022. Т. 1, № 2. С. 81–90. DOI: 10.31799/2949-0693-2022-2-81-90.

Original article

Application of Petri nets for the analysis of schedule tables of communication protocols

Nadezhda Yu. Chumakova¹

✉ nadezhda.chumakova@guap.ru, orcid.org/0000-0002-8979-979X

Valentin L. Olenev¹

valentin.olenev@guap.ru, orcid.org/0000-0002-1817-2754

¹ Saint Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russian Federation

Abstract. The correct distribution of channel resources directly affects the operation of on-board computer networks. The scheduling mechanism implemented in various communication protocols allows this distribution to be carried out using scheduling tables. In the design of onboard networks, it is extremely important to draw up a correct schedule table. Therefore, an important task is the analysis and verification of the schedule tables. This article presents an approach to the analysis and verification of schedule tables using such a system research tool as Petri nets.

Keywords: on-board networks, SpaceWire, communication protocols, STP-ISS, Petri net, modeling

Acknowledgements. The work was supported by the Ministry of Science and Higher Education and of the Russian Federation, grant agreement no. FSRF-2020-0004 «Scientific basis for architectures and communication systems development of the onboard information and computer systems new generation in aviation, space systems and unmanned vehicles».

For citation: Chumakova N. Yu., Olenev V. L. Application of Petri nets for the analysis of schedule tables of communication protocols. *Innovacionnoe priborostroenie* = *Innovative Instrumentation*. 2022;1(2):81–90. (In Russ.). DOI: 10.31799/2949-0693-2022-2-81-90.

Введение

Бортовые вычислительные сети современных космических аппаратов состоят из большого числа элементов с различной функциональностью, связанных в единую инфраструктуру. Задача проектирования бортовых сетей является крайне

сложной, при этом ее сложность возрастает с увеличением числа устройств в сети, влекущим за собой увеличение количества информационных потоков, а также возрастание объема требований и ограничений к сети.

При проектировании бортовых вычислительных сетей одной из важнейших проблем является

проблема планирования канальных ресурсов. Планирование канальных ресурсов позволяет решать такие задачи, как упорядочивание информационных потоков в сети, предотвращение конфликтов при одновременном использовании узлами ограниченных ресурсов сети, а также обеспечение в большей степени гарантированных задержек для передачи пакетов в сети.

Планирование особенно актуально для бортовых сетей на основе стандарта SpaceWire [1]. SpaceWire является одной из самых распространенных современных технологий, использующихся для бортовых сетей космических аппаратов. В данной статье будет рассмотрен механизм «Планирование» протокола СТП-ИСС [2] – одного из транспортных протоколов, работающих в сетях SpaceWire.

Механизм «Планирование» протокола СТП-ИСС позволяет составлять таблицы расписания для узлов в сети [3]. С помощью данных таблиц в сети осуществляется распределение канальных ресурсов во времени.

Некорректная таблица расписания может повлечь за собой образование конфликтов в сети, большие задержки при передаче данных, невозможность передачи данных за допустимое время и многие другие ошибки. Поэтому проверка корректности таблиц расписания является крайне важной.

В данной статье будет рассматриваться подход к анализу и проверке корректности таблиц расписания при помощи сетей Петри [4]. Сети Петри были выбраны в качестве инструмента анализа, так как они позволяют провести тщательную и точную проверку таблиц расписания.

Механизм «Планирование» протокола СТП-ИСС

SpaceWire – международный стандарт, описывающий построение бортовых сетей передачи,

обработки и управления информацией на борту космических аппаратов [5]. В сетях SpaceWire при помощи механизмов распространения меток времени (тайм-кодов) можно выполнять синхронизацию работы датчиков в реальном времени с точностью до микросекунд, выполнять синхронизацию таймеров, а также синхронно получать информацию по единому сигналу управления [5–7].

Протокол СТП-ИСС является транспортным протоколом, разработанным для применения в сетях SpaceWire. Данный протокол обладает рядом ключевых особенностей, отличающих его от других протоколов транспортного уровня. Такими особенностями являются:

- возможность передачи данных как с установкой транспортного соединения и управлением потоком, так и без нее;

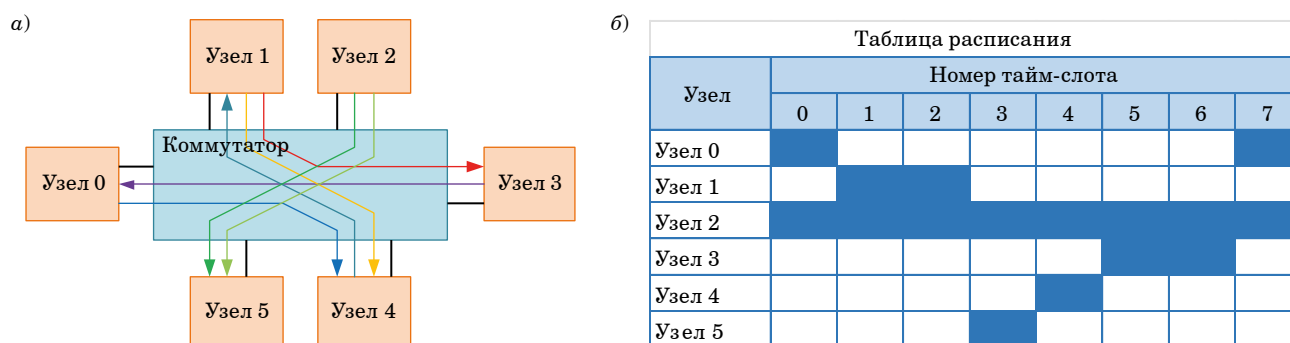
- поддержка механизмов качества сервиса (Quality of Service, QoS) «С приоритетом», «Гарантированная доставка данных», «Негарантированная доставка данных», «Планирование»;

- обнаружение дублированных пакетов команд управления [2] и др.

Для всей сети SpaceWire, в состав которой входят узлы с протоколом СТП-ИСС с включенным механизмом «Планирование», создается единое расписание в виде таблицы расписания. Оно регламентирует передачу данных для каждого узла в разрешенные для него временные интервалы (тайм-слоты). На рис. 1 показан пример сети и таблицы расписания для данной сети.

Расписание и длительность временного интервала задаются на этапе конфигурации и хранятся в каждом узле сети. Таймер временного интервала используется для отсчета времени до окончания текущего временного интервала на локальном узле [3].

Таблица расписания описывает одну эпоху. Эпоха состоит из определенного числа временных интервалов. Каждый узел сети с включенным



• Рис. 1. Пример потоков в сети (а) и таблицы расписания (б)
• Fig. 1. Example of data flows in network (a) and scheduling table (b)

механизмом «Планирование» имеет право отправлять данные только в течение тех временных интервалов, которые разрешены для данного узла. В конце временного интервала узел, отправляющий пакет, должен остановить передачу данных только после того, как завершится передача текущего пакета. Если на передающей стороне есть данные для передачи, но узлу не разрешено отправлять данные в текущий временной интервал, то узел должен ожидать наступления следующего временного интервала [8].

Сети Петри и их подклассы

Для анализа таблиц расписания мы будем применять сети Петри.

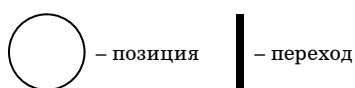
Сети Петри – это инструмент исследования систем. Теория сетей Петри позволяет моделировать систему ее математическим представлением в виде сети Петри. Структура сети Петри определяется ее позициями, переходами, входной и выходной функциями.

Сети Петри представляются в виде двудольного ориентированного мультиграфа с двумя типами вершин – позициями и переходами. Графическое представление вершин показано на рис. 2.

Вершины графа соединяются дугами (стрелками). Дуги ориентированы и могут быть направлены либо от переходов к позициям, либо от позиций к переходам. Дуги не могут соединять две вершины одного типа (две позиции либо два перехода), так как граф сети Петри является двудольным. Также в сетях Петри допустимо использование кратных дуг между двумя вершинами. Это означает, что позиция и переход могут быть связаны более чем одной дугой.

«Фишка» (или «маркер») – это примитивное понятие сетей Петри. Фишка может символизировать собой какой-либо объект, перемещающийся по сети Петри. Присвоение фишек позициям в сети называется маркировкой сети Петри. Фишки могут присваиваться только позициям, но не переходам. Число и положение фишек в сети может изменяться в зависимости от ее структуры, а также меняться в процессе выполнения сети.

Выполнение сети Петри – это запуск ее переходов. Запуск перехода представляет собой уда-



• Рис. 2. Типы вершин сети Петри
• Fig. 2. Types of Petri net vertices

ление фишек из входных позиций данного перехода и добавление фишек в выходные позиции данного перехода. Переход может быть запущен только в том случае, если он разрешен.

Разрешенным является переход, для которого выполняется следующее правило: каждая входная позиция данного перехода должна иметь число фишек, большее или равное числу дуг, направленных из позиции в переход. Если хотя бы в одной из входных позиций число фишек меньше числа дуг, то условие не выполняется и переход не может быть запущен.

В один момент модельного времени может сработать только один переход, любой из разрешенных [4, 9].

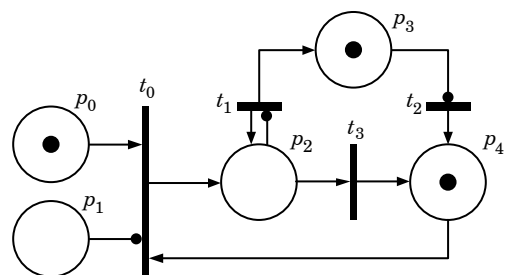
Подклассы сетей Петри. Существуют различные подклассы сетей Петри: ординарные, ингибиторные, с приоритетами, стохастические и др. Для анализа таблиц расписания протокола СТП-ИСС будут использоваться два подкласса – ингибиторные и временные сети Петри.

1. Ингибиторные сети Петри. В сетях Петри, помимо обычных дуг, могут использоваться ингибиторные дуги. Сети Петри, в которых используются ингибиторные дуги, называют ингибиторными сетями Петри.

Ингибиторная дуга – это дуга, осуществляющая проверку на нулевую разметку, т. е. проверку отсутствия фишек в позиции. Ингибиторные дуги могут быть направлены только от позиций к переходам. Кратность таких дуг всегда равна единице. На графе ингибиторные дуги отображаются в виде дуг с небольшими кругами вместо стрелок.

Пример ингибиторной сети Петри приведен на рис. 3. В данном примере в сети Петри имеется три ингибиторных дуги: из p_1 в t_0 , из p_2 в t_1 , из p_3 в t_2 .

В ингибиторной сети разрешенным является переход, для которого выполняется следующее правило: каждая входная позиция, соединенная с переходом t обычной дугой с кратностью w , содержит не менее w фишек, а каждая входная



• Рис. 3. Пример ингибиторной сети Петри
• Fig. 3. Example of an Inhibitor Petri net

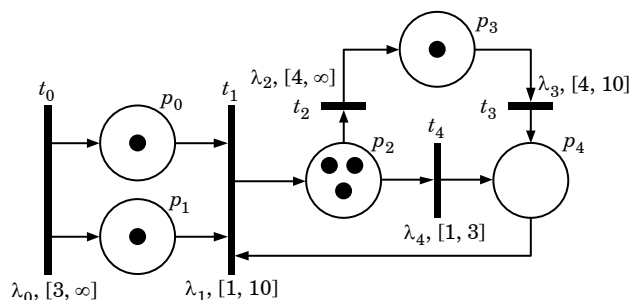
позиция, соединенная с переходом t ингибиторной дугой, имеет нулевую разметку, т. е. не содержит фишек [4].

В примере, показанном на рис. 3, в соответствии с правилом срабатывания переходов в ингибиторных сетях Петри, разрешены переходы t_0 и t_1 .

2. Временные сети Петри представляют собой сети Петри, в которых переходы ограничены дополнительным временным условием. При этом время в сети рассматривается не как абсолютная величина, а как временные единицы – часы, секунды, наносекунды и др.

Существуют различные типы временных сетей Петри: с мягким временем, с жестким временем и др. Для анализа таблиц расписания будут использованы сети Петри с жестким временем.

В сетях Петри с жестким временем переходы обладают задержкой срабатывания λ , выполняемой после того, как переход станет разрешенным.



• Рис. 4. Пример сети Петри с жестким временем
• Fig. 4. Example of a Petri net with hard time

Помимо задержек, для переходов устанавливается временной интервал, в течение которого переход должен сработать: (α, β) , где α – нижний, β – верхний предел срабатывания перехода, т. е. минимальное и максимальное время срабатывания перехода после того, как он стал разрешенным [4].

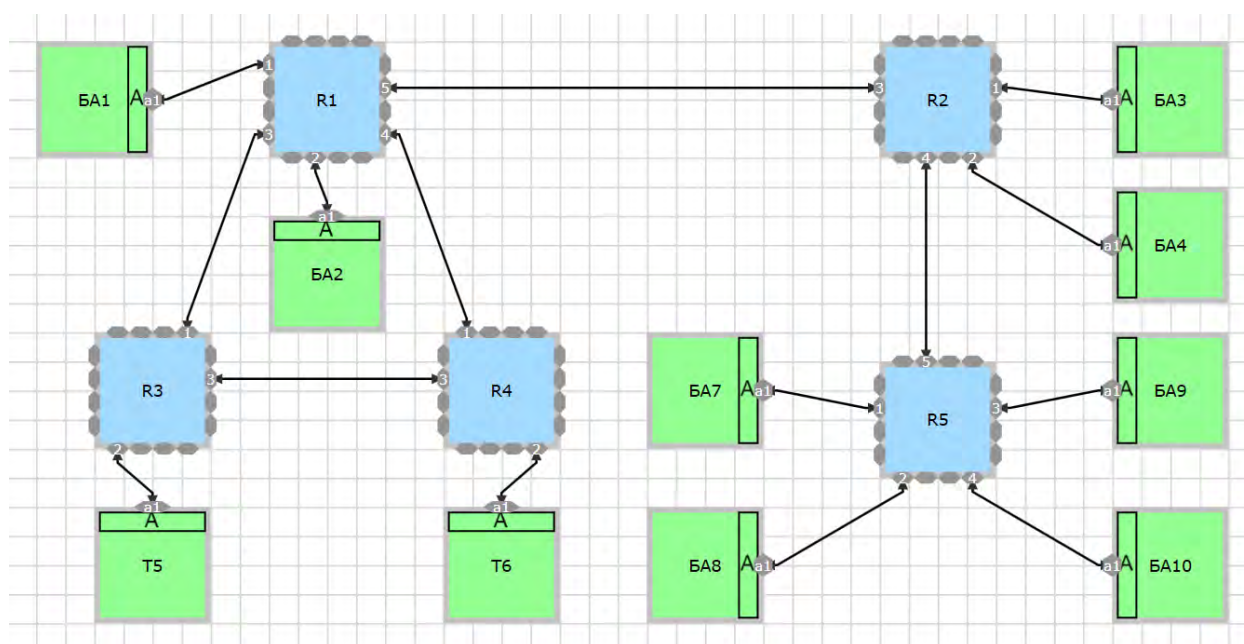
Пример сети Петри с жестким временем показан на рис. 4. В примере каждый переход сети обладает интервалом срабатывания (например, для перехода t_1 интервал срабатывания – $[1, 10]$) и задержкой λ_n . Множество, содержащее задержки переходов, может выглядеть следующим образом:

$$\Lambda(t) = \{3, 1, 1, 0, 2\}.$$

Применение сетей Петри для анализа таблиц расписания

Таблица расписания напрямую зависит от топологии сети, потоков данных в сети, а также от маршрутов передачи данных. При изменении любого из этих параметров требуется также изменение таблицы расписания. Поэтому анализ таблицы расписания проводится для конкретной топологии сети с заданными маршрутами передачи данных и параметрами потоков данных.

Рассмотрим пример. На рис. 5 показан пример топологии сети, работающей посредством протокола СТП-ИСС с включенным механизмом «Планирование».



• Рис. 5. Топология сети
• Fig. 5. Network topology

При помощи системы проектирования и моделирования бортовых сетей SANDS [10] для данной топологии была сгенерирована таблица расписания (рис. 6) со следующими параметрами:

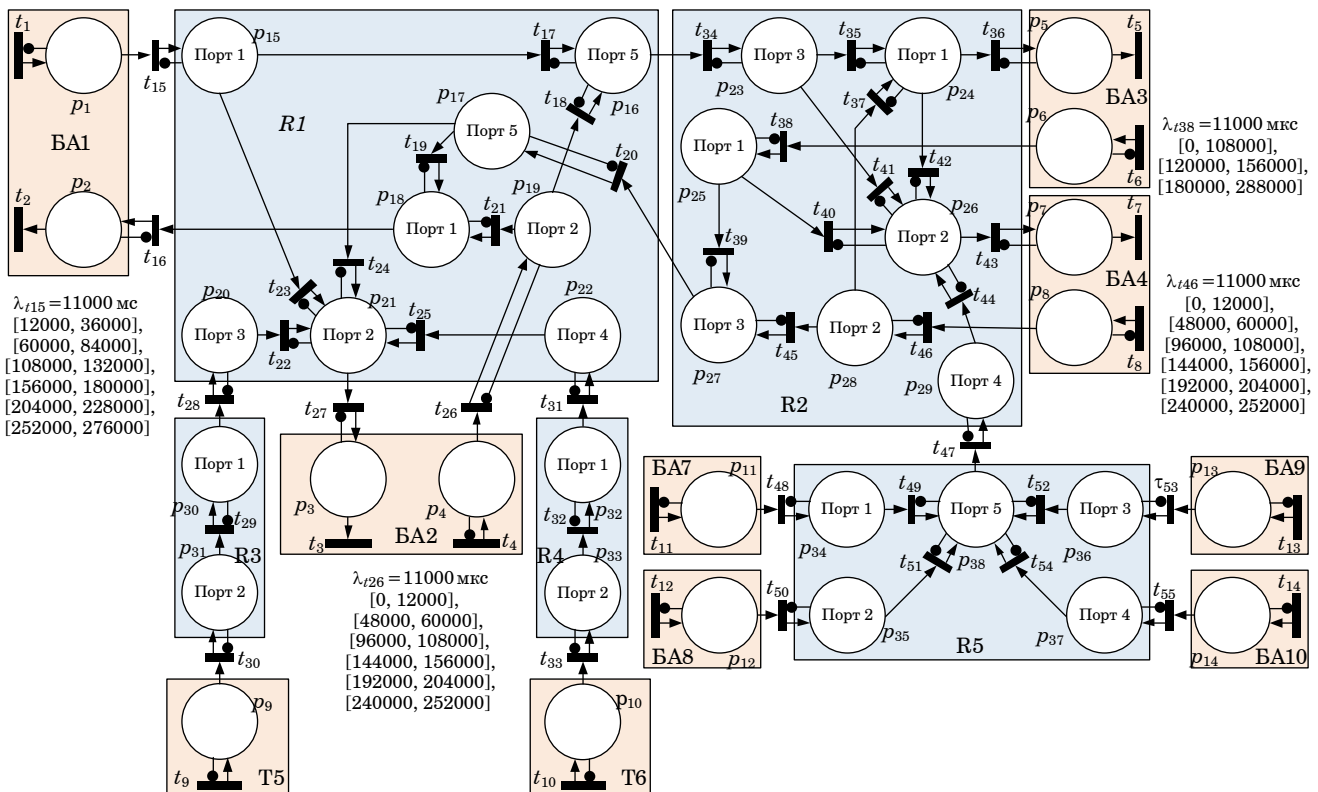
- длительность одной эпохи: 300 000 мкс (0,3 с);
- длительность временного интервала: 12 000 мкс (0,012 с);
- расписание отправки:
 - а) для узла БА2: 0, 4, 8, 12, 16, 20;
 - б) для узла БА1: 1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21, 22;
 - в) для узла БА3: 0-8, 10-13, 15-23;
 - г) для узла БА4: 0, 4, 8, 12, 16, 20.

ID узла	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
БА2	■				■				■				■				■				■				
БА1		■	■			■	■			■	■		■	■			■	■			■	■			
БА3	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
БА4	■				■				■				■				■				■				
Начало временного интервала, мкс	0	12000	24000	36000	48000	60000	72000	84000	96000	108000	120000	132000	144000	156000	168000	180000	192000	204000	216000	228000	240000	252000	264000	276000	288000

• Рис. 6. Таблица расписания
• Fig. 6. Scheduling table

В таблице расписания отображены только узлы с включенным механизмом «Планирование» – БА1, БА2, БА3 и БА4. Остальные узлы могут отправлять данные в любой момент времени.

Построение временной ингибиторной сети Петри. На рис. 7 показана сеть Петри, представляющая собой модель работы сети, топология которой приведена на рис. 5. Для ее построения были использованы два подкласса сетей Петри: ингибиторные сети Петри и временные сети Петри с жестким временем. Для обозначения данной сети будет использоваться сокращение «ВИСП» – «временная ингибиторная сеть Петри».



• Рис. 7. Временная ингибиторная сеть Петри
• Fig. 7. Timed Inhibitor Petri Net

При проектировании ВИСП учитывали маршруты передачи данных в сети, приведенные в табл. 1.

Построенная ВИСП включает в себя 38 позиций. Позиции p_1 – p_{14} соответствуют портам узлов сети. Остальные позиции соответствуют портам коммутаторов.

В моделируемой сети (рис. 5) все порты являются двунаправленными и могут выполнять как прием, так и передачу данных. Для моделирования данной особенности в спроектированной ВИСП порты представляются в виде двух позиций. Например, для моделирования работы порта узла БА1 используется позиция p_1 , моделирующая процесс передачи данных через порт узла, а также позиция p_2 , моделирующая процесс приема данных через порт узла.

Некоторые порты представлены только одной из двух позиций. Такой вариант представления выбирался в случае, если в сети отсутствуют маршруты передачи данных, включающие в себя прием или передачу данных по некоторому порту. Например, в сети существует маршрут, на котором узел Т5 выполняет отправку данных, но в сети отсутствуют маршруты, на которых узел Т5 выполняет прием данных. Поэтому порт узла представляется только одной позицией – p_9 , моделирующей процесс передачи данных через порт

- Таблица 1. Маршруты передачи данных в сети
- Table 1. Data Routes in Network

Передатчик	Приемник	Маршрут
БА1	БА2	БА1 – R1 – БА2
	БА3	БА1 – R1 – R2 – БА3
	БА4	БА1 – R1 – R2 – БА4
БА2	БА1	БА2 – R1 – БА1
	БА3	БА2 – R1 – R2 – БА3
	БА4	БА2 – R1 – R2 – БА4
БА3	БА1	БА3 – R2 – R1 – БА1
	БА2	БА3 – R2 – R1 – БА2
	БА4	БА3 – R2 – R1 – БА4
БА4	БА1	БА4 – R2 – R1 – БА1
	БА2	БА4 – R2 – R1 – БА2
	БА3	БА4 – R2 – R1 – БА3
БА7	БА4	БА7 – R5 – R2 – БА4
БА8	БА4	БА8 – R5 – R2 – БА4
БА9	БА4	БА9 – R5 – R2 – БА4
БА10	БА4	БА10 – R5 – R2 – БА4
Т5	БА2	Т5 – R3 – R1 – БА2
Т6	БА2	Т6 – R4 – R1 – БА2

узла Т5. При добавлении нового маршрута ВИСП, при необходимости, нужно модифицировать.

В ВИСП есть две группы переходов:

1) переходы t_1 – t_{14} – отображают внутреннюю работу узлов. Данные переходы имеют задержку λ , равную 1000 мкс;

2) переходы t_{15} – t_{55} – моделируют задержку при передаче данных между портами, а также обработку принимаемых и отправляемых через порты данных, маршрутизацию и т. д. Для примера было принято, что все пакеты имеют равную длину и могут быть обработаны коммутатором, приняты через порт, либо переданы через порт за 11 000 мкс (0,011 с). Поэтому данные переходы имеют одинаковую задержку λ , равную 11 000 мкс.

Построенная ВИСП является временной сетью Петри с жестким временем, и для всех переходов в сети, помимо задержки, определены интервалы времени, в течение которых данные переходы могут сработать. Значения временных интервалов приведены в табл. 2. Для переходов t_{15} , t_{26} , t_{38} и t_{46} выбранные временные интервалы соответствуют тайм-слотам, разрешенным для отправки соответствующим узлам в таблице расписания (рис. 6).

Ингибиторные дуги в построенной ВИСП выполняют функцию проверки возможности пере-

- Таблица 2. Интервалы времени для переходов ВИСП
- Table 2. Time slots for TIPN transitions

Переход	Значение	Интервалы
t_{15}	Передача данных от узла БА1	[12 000, 36 000], [60 000, 84 000], [108 000, 132 000], [156 000, 180 000], [204 000, 228 000], [252 000, 276 000]
t_{26}	Передача данных от узла БА1	[0, 12 000], [48 000, 60 000], [96 000, 108 000], [144 000, 156 000], [192 000, 204 000], [240 000, 252 000]
t_{38}	Передача данных от узла БА3	[0, 108 000], [120 000, 156 000], [180 000, 288 000]
t_{46}	Передача данных от узла БА4	[0, 12 000], [48 000, 60 000], [96 000, 108 000], [144 000, 156 000], [192 000, 204 000], [240 000, 252 000]
Другие переходы	Передача данных между узлами и коммутаторами в сети, внутренняя работа узлов	[0, 300 000]

дачи и приема данных через порты узлов и коммутаторов. Передача данных выполняется только в случае, если в данный момент через порт не выполняется передача данных, т. е. если в соответствующей позиции, обозначающей порт, отсутствует фишка. Аналогично выполняется прием данных.

Анализ таблицы расписания при помощи ВИСП. Анализ таблицы расписания сводится к анализу достижимости конкретных маркировок путем построения полного дерева достижимости, включающего в себя все возможные маркировки спроектированной ВИСП, либо к построению отдельных веток дерева достижимости, отображающих конкретные примеры работы сети.

Рассмотрим пример. Выполняется передача данных в сети по следующим маршрутам:

от узла БА1 к узлу БА2, через коммутатор R1. По данному маршруту передается 1 пакет;
от узла БА2 к узлу БА1, через коммутатор R1. По данному маршруту передается 2 пакета;
от узла БА3 к узлу БА2 через коммутаторы R2 и R1. По данному маршруту передается 2 пакета;
от узла БА4 к узлу БА1 через коммутаторы R2 и R1. По данному маршруту передается 2 пакета;
от узла Т5 к узлу БА2 через коммутаторы R3 и R1. По данному маршруту передается 1 пакет;

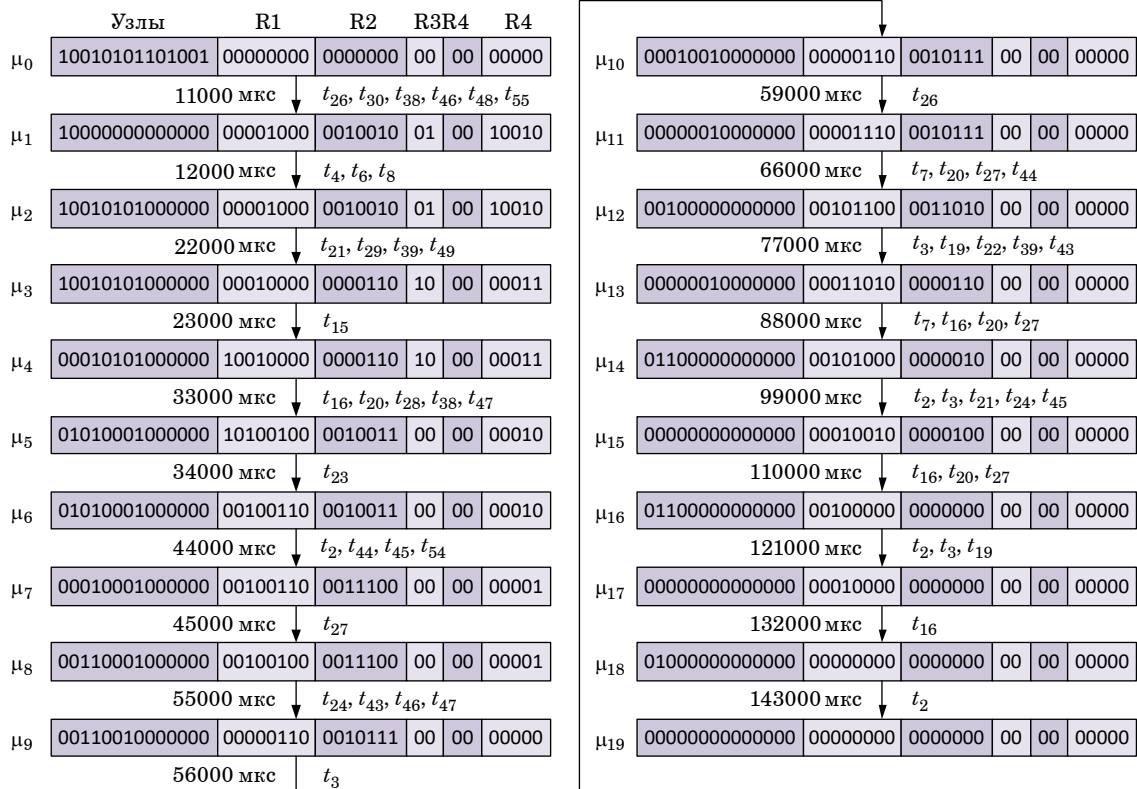
от узла БА7 к узлу БА4 через коммутаторы R5 и R2. По данному маршруту передается 1 пакет;

от узла БА10 к узлу БА4 через коммутаторы R5 и R2. По данному маршруту передается 1 пакет

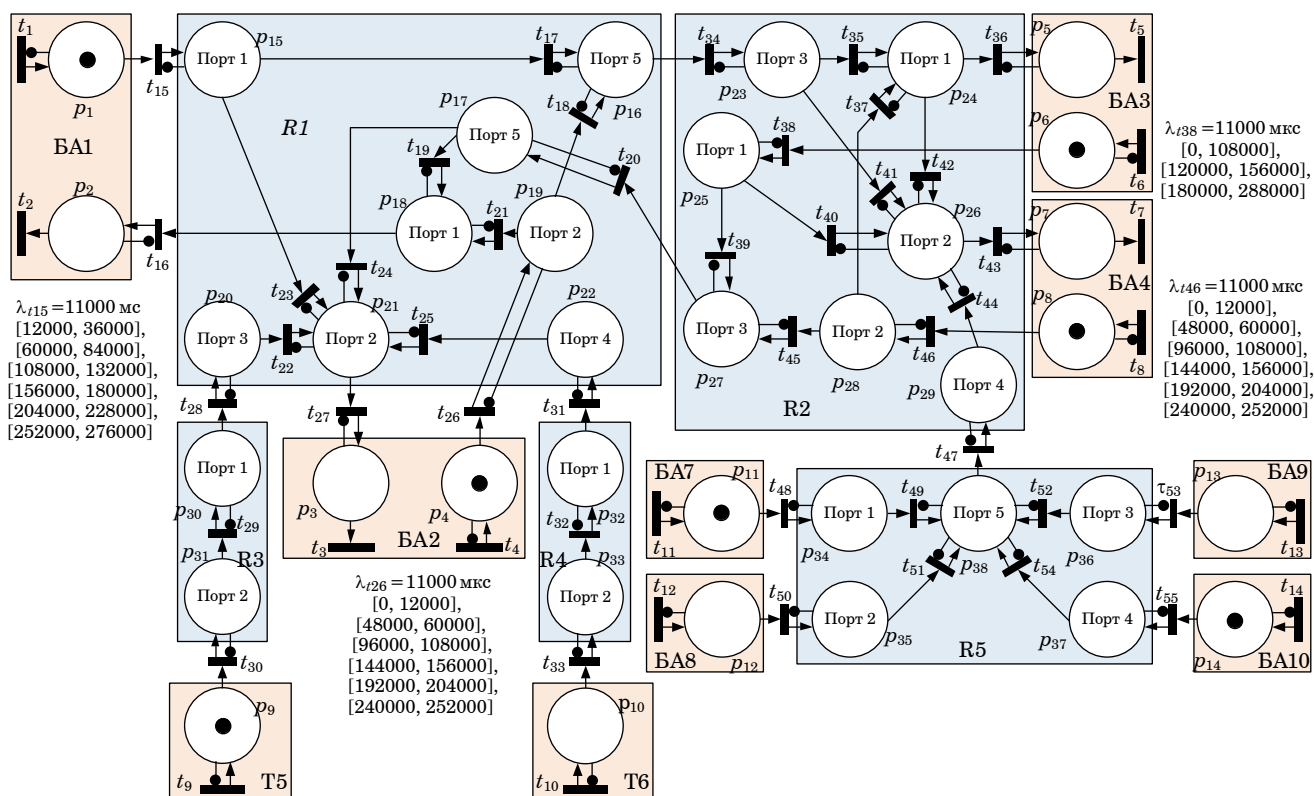
Передача будет считаться успешной в случае, если будет достигнута нулевая маркировка, т. е. в ВИСП не останется фишек. Нулевая маркировка означает, что все узлы-отправители передали все данные, а все узлы-получатели приняли все данные.

На рис. 8 показана одна ветка дерева достижимости, отображающая пример работы сети.

Согласно правилам выполнения сетей Петри, в один момент модельного времени может сработать только один из разрешенных переходов. Рассматриваемая ВИСП имеет большое число переходов, что влечет за собой больше число маркировок. Для упрощения отображения дерева было принято решение группировать переходы. Например, из начальной маркировки μ_0 могут быть запущены переходы $t_{26}, t_{30}, t_{38}, t_{46}, t_{48}$ и t_{55} . Данные переходы не влияют на работу друг друга и срабатывают друг за другом с небольшим временным промежутком. Маркировка μ_1 будет являться результатом последовательного запуска переходов $t_{26}, t_{30}, t_{38}, t_{46}, t_{48}$ и t_{55} , промежуточные маркировки после запуска каждого из перечисленных переходов на рис. 8 не отображаются.



• Рис. 8. Ветка дерева достижимости
• Fig. 8. Branch of reachability tree



• Рис. 9. Начальная маркировка
• Fig. 9. Initial marking

Рассмотрим некоторые маркировки данной ветки.

1. Начальная маркировка μ_0 . Узлы БА1, БА2, БА3, БА4, Т5, БА7 и БА10 имеют данные для отправки. Начальная маркировка показана на рис. 9.

2. Маркировка μ_1 . На рис. 9 отображено время срабатывания переходов. Текущее время (11 000 мкс) не входит в разрешенный интервал для перехода t_{15} , поэтому переход не срабатывает. Данный переход представляет собой порт узла БА1, выполняющий отправку данных. Следовательно, узел БА1 не выполняет отправку данных. Остальные переходы, соответствующие узлам, имеющим данные для отправки, срабатывают.

3. Маркировка μ_4 . Текущее время (23 000 мкс) входит в разрешенный интервал для перехода t_{15} , поэтому переход срабатывает, соответственно, узел БА1 начинает передачу данных.

4. Маркировка μ_5 . Срабатывает переход t_{16} . Срабатывание данного перехода означает, что узел БА1 принял пакет. Далее, после срабатывания перехода t_2 (маркировка μ_7), пакет будет обработан узлом.

5. Маркировка μ_{19} . В ходе работы сети все пакеты были отправлены узлами-отправителями и приняты узлами-получателями. В сети не осталось фишек.

В данном примере все узлы отправляют данные в соответствии с расписанием (см. рис. 6). Например, узел БА2 (переход t_{26}) выполняет отправку в 11 000 мкс (соответствует разрешенному тайм-слоту № 0) и в 59 000 мкс (соответствует разрешенному тайм-слоту № 4). В ходе работы сети все данные были переданы. Можно сделать вывод, что для данного примера таблица расписания корректна.

Заключение

Нами был представлен подход к анализу и проверке корректности таблиц расписания при помощи сетей Петри. Данный подход включает в себя два основных этапа: построение сети Петри и анализ путем построения дерева достижимости.

В статье рассмотрен пример анализа таблицы расписания для сети, работающей в соответствии с протоколом СТП-ИСС – транспортным протоколом, разработанным для сетей SpaceWire. Для данного примера построена сеть Петри. Для моделирования передачи данных использовали временные сети Петри с жестким временем. Этот подкласс сетей Петри позволил смоделиро-

вать задержки при передаче данных через порты устройств, а также передачу данных в соответствии с расписанием. Для проверки возможности использования портов для передачи и приема данных в спроектированной сети Петри также использовали ингибиторные дуги.

Для анализа таблицы расписания была построена ветвь дерева достижимости, реализующая конкретный пример работы сети. Данный пример показывает, что в сети возможна передача данных в соответствии с расписанием, за-

давшим в анализируемой таблице расписания. Построение полного дерева достижимости позволит рассмотреть все возможные сценарии передачи данных в моделируемой сети в соответствии с таблицей расписания и провести исчерпывающий анализ корректности таблицы расписания.

Предложенный подход может быть применен для анализа таблиц расписания не только протокола СТП-ИСС, но и других коммуникационных протоколов со схожим механизмом планирования.

СПИСОК ИСТОЧНИКОВ

1. SpaceWire Standard. ECSS-E-ST-50-12C // Space engineering SpaceWire – Links, nodes, routers and networks. 2008. P. 137.
2. Разработка, анализ и проектирование транспортного протокола СТП-ИСС для бортовых космических сетей SpaceWire / Ю. Е. Шейнин [и др.] // Исследования наукограда: науч. журн. 2016. № 1–2.
3. Коробков И. Л. Метод планирования канальных ресурсов в бортовых сетях SpaceWire с технологией TDMA // Радиопромышленность. 2019. Т. 29, № 4. С. 44–53.
4. Оленев В. Л. Моделирование систем: учеб. пособие. СПб.: ГУАП, 2015. 95 с.
5. Тюнягин Д. В. Распределенная система управления легких космических аппаратов, построенная на базе технологий «Система на кристалле» и SpaceWire // Вестн. СибГАУ. 2009. № 2. С. 182–187.
6. Иванов М. А., Кириллов К. Ю. Бортовой комплекс управления для искусственного спутника Луны // Исследования наукограда: науч. журн. 2014. № 1 (7). С. 4–11.
7. Ханов В. Х. Сетевые технологии для бортовых систем космического аппарата: опыт разработки // Доклады ТУСУР. 2014. № 2 (32). С. 287–293.
8. Design and Simulation of Onboard SpaceWire Networks / V. Olenov, I. Lavroskaya, Yu. Sheynin, I. Korobkov // Proceeding of the 24th conference of FRUCT association / Moscow Technical University of Communications and Informatics (MTUCI). Moscow, 2019. P. 291–299.
9. Котов В. Е. Сети Петри. М.: Наука, 1984. 160 с.
10. SANDS tool for design and simulation of onboard networks / V. L. Olenov, I. L. Korobkov, N. Y. Chumakova, N. I. Sinyov // Proceedings of 2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). IEEE. SPb., 2021. С. 1–10.

REFERENCES

1. SpaceWire Standard. ECSS-E-ST-50-12C. Space engineering SpaceWire – Links, nodes, routers and networks. 2008. 137 p.
2. Sheynin Yu. E., Olenov V. L., Lavrovskaya I. Y. et al. Development, analysis and design of the STP-ISS transport protocol for SpaceWire onboard space networks. Issledovaniya naukoграда = Science city research. 2016;(1–2):21–30. (In Russ.).
3. Korobkov I. A. Channel resource scheduling method in SpaceWire on-board networks with TDMA technology. Radiopromishlennost = Radio industry. 2019;(4):44–53. (In Russ.).
4. Olenov V. L. Modeling of the systems: study guide. SPb., SUAI; 2015, 95 p. (In Russ.).
5. Tunagin D. V. Distributed control system for light spacecraft based on System-on-a-Chip and SpaceWire technologies. Vestnik SibGAU = Bulletin of SibSAU. 2009;(2):182–187. (In Russ.).
6. Ivanov M. A., Kirilov K. Yu. On-board control complex for an artificial moon satellite. Issledovaniya naukoграда = Science city research. 2014;(7):4–11. (In Russ.).
7. Hanov V. H. Network technologies for spacecraft onboard systems: development experience. Doklady TUSUR = TUSUR reports. 2014;(32):287–293. (In Russ.).
8. Olenov V., Lavroskaya I., Sheynin Yu., Korobkov I. Design and Simulation of Onboard SpaceWire Networks. Proceeding of the 24th conference of FRUCT association. Moscow Technical University of Communications and Informatics (MTUCI). Moscow, 2019, pp. 291–299. (In Russ.).

9. Kotov V. E. Petri nets. Moscow: Nauka; 1984. 160 p. (In Russ.).
10. Olenov V. L., Korobkov I. L., Chumakova N. Y., Sinyov N. I. SANDS tool for design and simulation of onboard networks. Proceedings of 2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). IEEE. SPb., 2021, pp. 1–10. (In Russ.).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Чумакова Надежда Юрьевна, ассистент кафедры аэрокосмических компьютерных и программных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Область научных интересов – моделирование, бортовые вычислительные сети, коммуникационные протоколы.

Оленев Валентин Леонидович, кандидат технических наук, доцент, заведующий кафедрой аэрокосмических компьютерных и программных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Область научных интересов – бортовые сети, встроенные системы, моделирование, формальная верификация, космические аппараты и техника, технология SpaceWire/SpaceFibre.

Chumakova Nadezhda Yu., Assistant at the Department of Aerospace Computer and Software Systems, St. Petersburg State University of Aerospace Instrumentation.

Research interests – modeling, on-board networks, communication protocols.

Olenov Valentin L., PhD in Technical Sciences, Associate Professor, Head at the Department of Aerospace Computer and Software Systems, St. Petersburg State University of Aerospace Instrumentation.

Research interests – on-board networks, embedded systems, modeling, formal verification, spacecraft and equipment, SpaceWire/SpaceFibre technology.

Поступила в редакцию 26.10.2022

Поступила после рецензирования 02.11.2022

Принята к публикации 21.11.2022

Received 26.10.2022

Revised 02.11.2022

Accepted 21.11.2022