

## ВЕРИФИКАЦИЯ ПРОТОКОЛОВ ETHERNET

## VERIFICATION OF ETHERNET PROTOCOLS

**Аннотация.** Выполнена верификация протоколов Ethernet, представленных сетями Петри. Использована естественная декомпозиция сети Петри на функциональные подсети, определяемая правилами построения модели. Инвариантность моделей, включающих произвольное число рабочих станций доказана на основе инвариантности модели отдельной рабочей станции, являющейся функциональной подсетью.

**Summary.** Verification of Ethernet protocols represented with Petri nets is implemented. Natural decomposition of Petri net into functional subnets defined by model construction rules is used. Invariance of models with an arbitrary number of workstations is proved on the base of invariance of separate workstation model that is functional subnet.

Верификация телекоммуникационных протоколов является традиционной областью применения сетей Петри [1]. Первые работы, представляющие методы верификации протоколов с помощью сетей Петри, появились в начале восьмидесятых годов прошлого столетия [2]. Действительно, большинство сетевых протоколов предполагают асинхронный характер взаимодействия систем, что затрудняет их описания с помощью последовательных моделей, таких, например, как блок-схема.

При исследовании протоколов, как правило, решают две основные задачи: доказательство корректности протокола и оценку его эффективности. Первую задачу называют еще верификацией протокола. Она действительно является приоритетной, поскольку наличие дефектов в исходных спецификациях представляет собой один из наиболее дорогостоящих типов ошибок. Поскольку, если некорректный протокол будет реализован программно либо аппаратно, то затраты на исправление ошибок могут быть значительными.

В процессе верификации, обычно, устанавливается возможность неограниченно повторять процесс взаимодействия систем, используя при этом элементы ограниченной емкости. В негативной формулировке эти условия можно представить как отсутствие взаимных блокировок (тупиков), а также отсутствие последовательностей действий, приводящих к переполнению накопителей.

Следует отметить, что истории развития телекоммуникационных сетей известны случаи, когда наличие незамеченных ошибок в спецификациях протоколов приводило к колоссальным потерям. Стоит вспомнить червь Морриса, использовавший брешь в спецификациях протоколов электронной почты и блокировавший работу сети Интернет почти на неделю.

Таким образом, верификация телекоммуникационных протоколов представляет собой научную проблему, для решения которой необходимы эффективные методы, так как процесс появления новых протоколов становится всё более динамичным. При наличии известных ранее детальных моделей сети Ethernet [3], формальное доказательство инвариантности её протоколов не было выполнено ввиду большой размерности сети Петри.

Целью настоящей работы является доказательство инвариантности протоколов локальной сети Ethernet с архитектурой общей шины для произвольного числа рабочих станций. Применение метода декомпозиции [4] с использованием свойств инвариантов функциональных подсетей [5] позволило решить указанную задачу.

**Протоколы Ethernet.** В сети Ethernet применяются протоколы множественного доступа к среде с прослушиванием несущей и определением коллизий CSMA/CD (Carrier Sense Multiply Access with Collision Detection). Исходные спецификации протоколов регламентируются стандартами IEEE 802.x.

Каждая из рабочих станций прослушивает среду передачи и, в случае отсутствия несущей, может начать передачу данных. Если несколько рабочих станций начнут передачу данных одновременно, то произойдет наложение информации, именуемое коллизией. Рабочая станция располагает средствами обнаружения коллизий. В случае возникновения коллизии передача данных прекращается и возобновляется через некоторый случайный интервал времени.

Спецификациями протоколов регламентируются также форматы передаваемых пакетов (фреймов) данных. В настоящей работе структура пакета данных Ethernet не используется, кроме

того, исследуется сеть с архитектурой общей шины. Рассматриваются начало и завершение передачи пакета, процессы распространения несущей по передающей среде во времени, а также распознавание и обработка коллизий. Заметим, что появление и прекращение несущей распространяется по общей шине в течение определённого времени, поэтому рабочие станции в некоторый фиксированный момент времени располагают различной информацией о состоянии канала.

**Модель Ethernet.** В настоящей работе будет исследована модель Ethernet с архитектурой общей шины и произвольным количеством подключенных рабочих станций. Она представляет собой несколько видоизменённую модель, описанную в работе Маршана [3]. В указанной работе отмечено, что построенная модель является достаточно подробным и точным представлением исходных спецификаций. Но, ввиду большой размерности даже для небольшого количества рабочих станций исследование свойств модели является слишком трудоёмким. Поэтому предложено исследовать упрощённую модель.

Применение методов декомпозиции сетей Петри позволило выполнить верификацию подробной модели, включающей произвольное количество рабочих станций. Исходная модель Маршана модифицирована таким образом, что модель отдельной рабочей станции представляет собой функциональную сеть Петри. Для этого добавлены контактные позиции. Компоновка модели локальной сети выполняется путём совмещения контактных позиций рабочих станций. Таким образом, имеется естественная декомпозиция общей модели на функциональные подсети.

Модель представлена в виде временной сети Петри [6] с ингибиторными дугами. Заметим, что использованы переходы трёх типов: срабатывающие мгновенно, изображаемые узкими прямоугольниками; с детерминированными временами срабатывания, изображаемые закрашенными прямоугольниками; со случайными равномерно распределёнными временами срабатывания, изображаемые широкими не закрашенными прямоугольниками. Ингибиторные дуги применяют для проверки маркировки позиции на ноль. Вместо стрелки такие дуги имеют окружность небольшого диаметра. Контактные позиции изображены линиями, имеющими двойную толщину. В дальнейшем используются как краткие смысловые наименования, так и номера элементов сетей Петри, изображённых на рисунках.

**Модель рабочей станции.** Опишем подробно модель рабочей станции, представленную на рис. 1. Для удобства описания модель можно разбить на две части с помощью пунктирных линий. Верхняя часть моделирует поведение рабочей станции. Нижняя часть описывает взаимодействие рабочей станции с каналом и процесс передачи данных между соседними рабочими станциями.

Позиция и переход *think* моделирует внутреннюю активность рабочей станции, не связанную с информационным обменом. В позиции *sense* выполняется прослушивание среды. Если в результате прослушивания установлено, что канал свободен, переход *idle* становится разрешённым. При срабатывании он помещает фишку в позицию *tx* (transmit), указывающую, что станция начинает передачу данных, а также фишку в позицию *ps* (propagation start), моделирующую начало распространения по сети несущей. Наличие фишки в позиции *tx* возбуждает переход *tx*, моделирующий передачу данных в сети. Когда завершается срабатывание перехода *tx*, станция заканчивает передачу данных и возвращается в исходное состояние *think*. Кроме того, фишка помещается в позицию *pe* (propagation end), моделирующую распространение по сети прекращения несущей.

Кроме того, если в любой момент времени при передаче пакета позиция *chstate* (channel state) содержит более двух фишек, срабатывает мгновенный переход *coll* (collision), моделирующий распознавание коллизии в канале. Перемещение фишки с позицию *jam* и срабатывание перехода *jam* моделируют действия по прекращению передачи данных при установлении коллизии. Перемещение фишки в позицию *back* инициирует процесс повторной передачи данных после распознавания коллизии. Случайное время срабатывания перехода *back* моделирует задержку перед повторной передачей.

Если при прослушивании среды в позиции *sense* установлено, что канал занят, то срабатывает переход *busy*, помещая фишку в позицию *pers* (persist). Как только канал освобождается, возбуждается и срабатывает мгновенный переход *pend* (pending). Он перемещает фишку в позицию *start* и после детерминированной временной задержки, представленной переходом *start*, начинается передача данных.

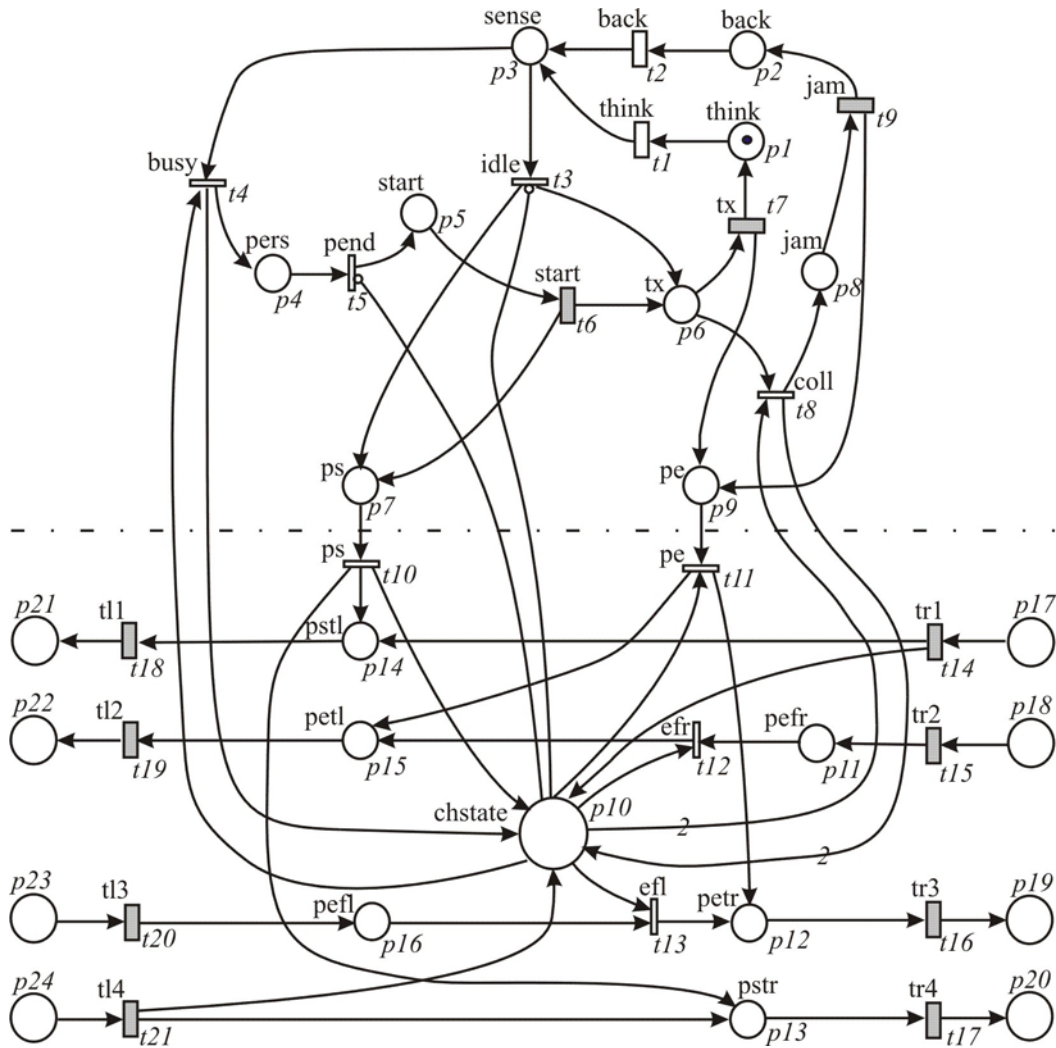


Рисунок 1 – Модель рабочей станции Ethernet

В нижней части сети основным элементом является позиция *chstate*, представляющая состояние канала, распознаваемое рабочей станцией. Поскольку, появление и прекращение несущей представляют собой процессы, время реализации которых учитывается в модели, каждая из рабочих станций может распознавать различное состояние канала. Если позиция *chstate* пуста – канал свободен, наличие ровно одной фишки свидетельствует о занятости канала, если же позиция содержит, по крайней мере, две фишки, то имеет место коллизия.

Переходы *t11*, *t12*, *t13*, *t14* моделируют задержки распространения (прекращения) несущей для левой соседней рабочей станции. Аналогичную роль играют переходы *tr1*, *tr2*, *tr3*, *tr4* для правой соседней станции. Заметим, что, так как предполагается компоновать общую модель путём совмещения позиций, времена срабатывания перечисленных переходов должны быть выбраны равными половине фактической задержки.

Когда станция начинает передачу данных и помещает фишку в позицию *ps*, срабатывает мгновенный переход *ps* и помещает по фишке в позиции *chstate*, *pstl*, *pstr*. Наличие фишек в позициях *pstl*, *pstr* моделирует распространение сигнала влево и вправо по общей шине.

Когда станция прекращает передачу данных и помещает фишку в позицию *pe*, срабатывает мгновенный переход *pe*, забирает одну фишку из позиции *chstate* и помещает по одной фишке в позиции *petl*, *petr*. Наличие фишек в позициях *petl* и *petr* моделирует прекращение сигнала в общей шине.

Сигнал начала передачи данных по общей шине может также поступать от соседней слева станции в позицию *p24* и от соседней справа станции в позицию *pstl*. Прекращение передачи данных по общей шине моделируется поступлением фишки от соседней слева рабочей станции в позицию *pefl* и от соседней справа рабочей станции в позицию *pefr*.

Заметим, что при поступлении сигнала начала передачи данных слева либо справа  $tl4$ ,  $tr1$  в позицию  $chstate$  поступает фишка, моделируя занятость канала, известную рабочей станции. Прекращение передачи данных рабочей станцией, расположенной слева либо справа на общей шине, приводит к срабатыванию переходов  $efl$ ,  $efr$  соответственно. Таким образом моделируется освобождение канала.

Итак, построенная сеть имеет входные позиции  $X = \{p_{17}, p_{18}, p_{23}, p_{24}\}$  и выходные позиции  $Y = \{p_{19}, p_{20}, p_{21}, p_{22}\}$ , и, в соответствии с определением [4], является функциональной сетью Петри. Матрица инцидентности сети имеет вид:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1	-1						1															
2		-1							1													
3	1	1	-1	-1																		
4				1	-1																	
5					1	-1																
6			1			1	-1	-1														
7			1			1				-1												
8								1	-1													
9								1	1		-1											
10			0	0	0			0		1	-1	-1	-1	1								1
11											1	-1			1							
12												1		1		-1						
13										1							-1					1
14										1				1				-1				
15											1	1							-1			
16													-1							1		
17														-1								
18															-1							
19																1						
20																	1					
21																		1				
22																			1			
23																				-1		
24																						-1

**Модель сети.** Рассмотрим правила построения модели Ethernet с архитектурой общей шины, насчитывающей  $k$  рабочих станций. Общая схема такой модели представлена на рис. 2.

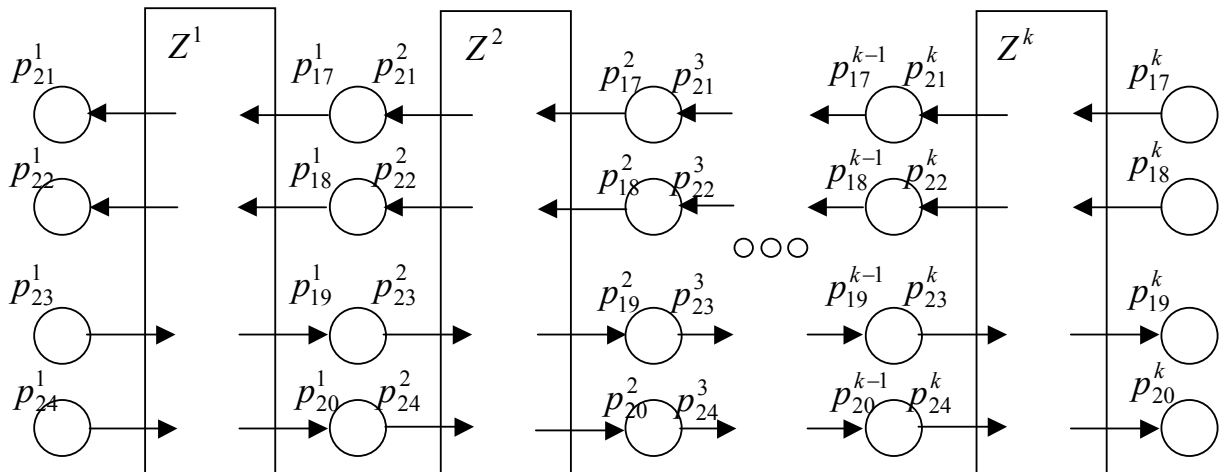


Рисунок 2 – Модель сети Ethernet с общей шиной

Модели рабочих станций  $Z^i$  объединяются путём совмещения контактных позиций. Причём каждая из станций взаимодействует ровно с двумя соседними станциями. Рассмотрим, например, подсеть  $Z^2$ . Контактные позиции  $p_{21}, p_{22}, p_{23}, p_{24}$  подсети  $Z^2$  объединяются с контактными позициями  $p_{17}, p_{18}, p_{19}, p_{20}$  соответственно подсети  $Z^1$ . Кроме того, контактные позиции

$P_{17}, P_{18}, P_{19}, P_{20}$  подсети  $Z^2$  объединяются с контактными позициями  $P_{21}, P_{22}, P_{23}, P_{24}$  соответственно подсети  $Z^3$ . Принадлежность элементов к той либо иной подсети будем отображать с помощью верхнего индекса, равного номеру подсети.

Заметим, что построенная модель имеет естественную декомпозицию на функциональные подсети  $\{Z^i\}$ . Этот факт будет использован далее при анализе свойств модели.

**Исследование свойств модели.** Как отмечено в работах [1, 2, 3], идеальная модель корректного телекоммуникационного протокола должна обладать такими свойствами, как ограниченность, консервативность, живость.

Для исследования перечисленных свойств применяют сетевые инварианты [1]. В этом случае модель корректного протокола должна быть инвариантной. Поскольку известные методы вычисления инвариантов [7,8] имеют экспоненциальную сложность, их применение становится практически невозможным при исследовании сетей большой размерности.

В работе [5] представлен метод вычисления инвариантов сети Петри на основе инвариантов её функциональных подсетей. При исследовании произвольной заданной сети Петри предварительно выполняется её декомпозиция на функциональные подсети. Метод декомпозиции описан в работе [4].

Так как множество  $\{Z^i\}$  для построенной модели Ethernet с общей шиной определяет разбиение сети Петри на функциональные подсети, можно применить метод, описанный в работе [5], для вычисления инвариантов. Напомним, что инвариантом сети называют целое неотрицательное решение (вектор-строка)  $\bar{x}$  уравнения

$$\bar{x} \cdot A = 0,$$

где  $A$  – матрица инцидентности сети Петри для инвариантов позиций либо транспонированная матрица инцидентности для инвариантов переходов.

Отметим, что в построенной модели Ethernet все функциональные подсети имеют одинаковую структуру и различаются лишь верхним индексом своих элементов. При вычислении инвариантов подсети  $Z^i$  не будем учитывать ингибиторные и тестирующие дуги, поскольку они не изменяют маркировку сети. Кроме того, дуги  $(t_3, p_6)$ ,  $(t_6, p_6)$ ,  $(t_{10}, p_{13})$ ,  $(t_{10}, p_{14})$ ,  $(t_{11}, p_{12})$ ,  $(t_{11}, p_{15})$  опущены, а дуга  $(t_{11}, p_6)$  добавлена, так как исходная модель протокола не является инвариантной.

Применение метода Тудика [7] даёт следующую матрицу базисных решений для инвариантов позиций:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	1	1	1	1	1	1	1		1		1			1		1		1			1		1
2												1				1			1					1
3										1					1			1				1		
4						1		1	1															
5														1			1					1		
6													1							1				1

Строка с номером  $l$  представленной матрицы является базисным инвариантом, который при построении общего решения умножается на некоторый произвольный целый неотрицательный коэффициент  $z_l$ . Столбцы матрицы соответствуют позициям сети  $Z^i$ .

В соответствии с [5] после вычисления инвариантов функциональных подсетей необходимо найти общие инварианты для контактных позиций. Так как в композиции двух соседних подсетей  $Z^i$  и  $Z^{i+1}$  участвуют четыре контактных позиции левой подсети  $\{P_{17}^i, P_{18}^i, P_{19}^i, P_{20}^i\}$  и четыре контактных позиции правой подсети  $\{P_{21}^{i+1}, P_{22}^{i+1}, P_{23}^{i+1}, P_{24}^{i+1}\}$ , система уравнений для контактных позиций имеет следующую форму:

$$\begin{cases} z_1^i + z_5^i = z_5^{i+1}, \\ z_3^i = z_1^{i+1} + z_3^{i+1}, \\ z_1^i + z_2^i = z_2^{i+1}, \\ z_6^i = z_1^{i+1} + z_6^{i+1}. \end{cases}$$

Эта система имеет общее решение, которое комбинирует инварианты соседних подсетей:

$$\begin{cases} z_1^{i,i+1} \cdot (z_3^i, z_6^i, z_1^{i+1}), \\ z_2^{i,i+1} \cdot (z_1^i, z_2^{i+1}, z_5^{i+1}), \\ z_3^{i,i+1} \cdot (z_6^i, z_6^{i+1}), \\ z_4^{i,i+1} \cdot (z_5^i, z_5^{i+1}), \\ z_5^{i,i+1} \cdot (z_3^i, z_3^{i+1}), \\ z_6^{i,i+1} \cdot (z_2^i, z_2^{i+1}), \\ z_7^{i,i+1} \cdot (z_4^{i+1}), \\ z_8^{i,i+1} \cdot (z_4^i), \end{cases}$$

где  $z_l^{i,i+1}, i = \overline{1, k-1}, l = \overline{1, 8}$  – произвольные неотрицательные целые числа.

Таким образом, базис инвариантов всей сети  $Z$  состоит из  $8 \cdot (k-1)$  инвариантов, которые могут быть представлены в обобщённой форме как:

$$\begin{cases} z_1^{i,i+1} \cdot (p_{11}^i, p_{13}^i, p_{15}^i, p_{18}^i, p_{20}^i, p_{22}^i, p_{24}^i, p_{21}^{i+1}, p_2^{i+1}, p_3^{i+1}, p_4^{i+1}, p_5^{i+1}, p_6^{i+1}, p_7^{i+1}, p_8^{i+1}, p_{10}^{i+1}, p_{12}^{i+1}, p_{15}^{i+1}, p_{17}^{i+1}, p_{19}^{i+1}, p_{22}^{i+1}, p_{24}^{i+1}), \\ z_2^{i,i+1} \cdot (p_1^i, p_2^i, p_3^i, p_4^i, p_5^i, p_6^i, p_7^i, p_8^i, p_{10}^i, p_{12}^i, p_{15}^i, p_{17}^i, p_{19}^i, p_{22}^i, p_{24}^i, p_{12}^{i+1}, p_{14}^{i+1}, p_{16}^{i+1}, p_{17}^{i+1}, p_{19}^{i+1}, p_{21}^{i+1}, p_{23}^{i+1}), \\ z_3^{i,i+1} \cdot (p_{13}^i, p_{20}^i, p_{24}^i, p_{13}^{i+1}, p_{20}^{i+1}, p_{24}^{i+1}), \\ z_4^{i,i+1} \cdot (p_{14}^i, p_{17}^i, p_{21}^i, p_{14}^{i+1}, p_{17}^{i+1}, p_{21}^{i+1}), \\ z_5^{i,i+1} \cdot (p_{11}^i, p_{15}^i, p_{18}^i, p_{22}^i, p_{11}^{i+1}, p_{15}^{i+1}, p_{18}^{i+1}, p_{22}^{i+1}), \\ z_6^{i,i+1} \cdot (p_{12}^i, p_{16}^i, p_{19}^i, p_{23}^i, p_{12}^{i+1}, p_{16}^{i+1}, p_{19}^{i+1}, p_{23}^{i+1}), \\ z_7^{i,i+1} \cdot (p_6^{i+1}, p_8^{i+1}, p_9^{i+1}), \\ z_8^{i,i+1} \cdot (p_6^i, p_8^i, p_9^i), \quad i = \overline{1, k-1}. \end{cases}$$

где  $z_l^{i,i+1}$  – произвольные целые неотрицательные коэффициенты.

Аналогичным образом могут быть получены инварианты переходов сети.

Так как, например, при выборе всех  $z_l^{i,i+1}$  равными единице, получаем положительный инвариант всей сети, то построенная модель Ethernet с архитектурой общей шины и произвольным количеством  $k$  рабочих станций является инвариантной сетью Петри.

**Выводы.** Таким образом, в настоящей работе доказана инвариантность модели Петри протоколов локальной сети Ethernet. Модель построена для сети Ethernet с общей шиной и представляет собой композицию функциональных подсетей, моделирующих рабочие станции. Применение композиции подсетей позволило доказать инвариантность протоколов для произвольного количества рабочих станций локальной сети. Инвариантность модели подтверждает корректность протоколов Ethernet.

### Литература

1. Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE, April 1989.- Vol. 77. – P. 541-580.
2. Diaz M. Modelling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Model // Computer Networks. – № 6. – 1982. – P. 419-441.
3. Ajmone Marsan M., Chiola G., Fumagalli A. An Accurate Performance Model of CSMA/CD Bus LAN // Advances in Petri Nets. – LNCS. – Vol. 266, 1987. – P. 146-161.
4. Zaitsev D.A. Subnets with input and output places // Petri Net Newsletter. – April 2003. – Vol. 64. – P. 3-6. Cover Picture Story.
5. Зайцев Д.А. Инварианты функциональных подсетей // Научные труды ОНАС им. А.С. Попова. – №4, 2003. – С. 57-63.
6. Зайцев Д.А., Слепцов А.И. Уравнения состояний и эквивалентные преобразования временных сетей Петри // Кибернетика и системный анализ. – 1997. – № 5. – с. 59-76.
7. Toudic J.M. Linear Algebra Algorithms for the Structural Analysis of Petri Nets // Rev. Tech. Thomson CSF, 1982. – № 1. – Vol. 14. – P. 136-156.
8. Zaitsev D.A. Formal Grounding of Toudic Method // Proc. of 10th Workshop Algorithms and Tools for Petri Nets. – September 26-27, 2003. – Eichstaett, Germany. – P. 184-190.