

УДК 65.011.56:004.75

**С.А. Даденков, Е.Л. Кон**Пермский национальный исследовательский политехнический университет,  
Пермь, Россия**АНАЛИТИЧЕСКИЙ РАСЧЕТ И АНАЛИЗ СРЕДНЕГО ВРЕМЕНИ  
РЕАКЦИИ ПРИЛОЖЕНИЯ УЗЛА НА СОБЫТИЕ  
В ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СЕТИ**

Исследован алгоритм циклического обслуживания приоритетных задач (модулей) программы узла информационно-управляющей сети (ИУС). Выполнен анализ факторов, влияющих на производительность приложения узла ИУС. Разработано аналитическое соотношение для определения одного из наиболее важных показателей производительности – времени реакции программы на происходящие события. Выполнены количественная оценка и анализ среднего времени реакции. Сделан вывод об адекватности предложенных соотношений и полученных результатов оценки. Сформулированы некоторые рекомендации к проектированию приложения узла ИУС.

**Ключевые слова:** аналитический расчет, время реакции, информационно-управляющая система, задача, конфигурационные свойства, условие проверки, программа, производительность, событие, узел, циклический алгоритм обслуживания, LonWorks, CSMA.

**S.A. Dadenkov, E.L. Kon**

Perm National Research Polytechnic University, Perm, Russian Federation

**ANALYTICAL CALCULATION AND ANALYSIS OF APPLICATION  
NODE AVERAGE RESPONSE TIME ON EVENTS  
IN DISTRIBUTED CONTROL SYSTEM**

The algorithm of round robin servicing of priority tasks (modules) node ICS (DCS) is researched. Analysis of application node factors influencing performance is made. The analytical ratio is developed for determination of time of response of the program to the occurring events. The quantitative evaluation and analysis of average time response is executed. The conclusion is drawn on adequacy of the proposed relationships and results. Some recommendations to an ICS node application programming are formulated.

**Keywords:** analytical calculation, response time, industrial control system, task, configuration property, when, application, performance, event, node, LonWorks, round-robin, CSMA.

**Введение.** Сегодня большинство создаваемых распределенных информационно-управляющих систем и сетей (РИУС) управления технологическими процессами и объектами критической инфраструктуры

строятся на основе fieldbus-технологий. Это обусловлено высокими техническими показателями fieldbus-систем, в частности, одним из наиболее важных – производительностью. Важность адекватной оценки производительности РИУС заключается в необходимости эффективного проектирования сети требуемой производительности, обеспечиваемой корректным распределением функционала между элементами сети и определением системных параметров.

Анализ публикаций [1–6] иллюстрирует преимущественный интерес ученых к анализу методов временного разделения сетевых каналов. Важность исследований обусловлена характерной, для промышленных ИУС, низкой пропускной способностью сетевых каналов. Влияние метода доступа (в частности, CSMA) на производительность является решающим фактором при высокой загрузке канала. Для наиболее характерной ИУС, малой и средней загруженности канала, важным фактором, влияющим на производительность, становится алгоритм обслуживания задач программы узла сети. Это обусловлено характерной для ИУС малой вычислительной мощностью узлов и большим количеством прикладных задач. Решению задачи количественной оценки производительности приложения узла посвящено малое количество работ [7–10]. В [7] выполнен натурный эксперимент с приложением узла и получены частные результаты оценки производительности. В [8, 9] авторами проведены анализ и систематизация факторов, потенциально влияющих на производительность приложения. В [10] разработана имитационная модель уровня приложения, реализованная в системе моделирования AnyLogic. Анализ работ показывает недостаточность существующих результатов для проектирования ИУС. Поэтому в работе производится разработка аналитического соотношения для расчета одного из наиболее важных показателей производительности ИУС – среднего времени реакции приложения узла на различные события. Исследуемая в работе циклическая дисциплина обслуживания задач программы узла используется во многих ИУС, в частности в распространенной LonWorks-системе [11, 12].

**Алгоритм циклического обслуживания задач программы узла.** Назначением работы приложения узла является своевременная реакция (выработка управляющих воздействий на ИМ) на происходящие в контролируемых процессах события (изменения величин параметров, измеряемых датчиками (Д), принимаемых от исполнительных механизмов

(ИМ), принимаемых сообщений от удаленных сетевых узлов (Д и ИМ)). Организация контроля за параметрами процесса производится в рамках программы циклическим обслуживанием модулей, включающих проверку определенных параметров по условию *when* и при успешной проверке условия решения задачи *task*. При этом критерием успешного выполнения проверки и решения задачи является предшествующее проверке событие. Следовательно, событием в системе является не любое изменение величин параметров контролируемого процесса, а только изменение, приводящее к потенциальному (будущему) срабатыванию определенных проверок *when*. Исследуя в работе аспекты производительности алгоритма обслуживания задач программы узла, особое внимание уделяется показателю времени реакции узла на событие. Отметим, что время реакции представляет интервал времени от момента происхождения события и до выработки на него управляющего воздействия определенным модулем программы (решение задачи и обработка ее результата).

Приложение узла представляется циклическим обслуживанием неоднородных по приоритетам задач и образует алгоритм обслуживания (рис. 1). Последовательность обслуживания прикладных модулей выполняется по круговой дисциплине (*round-robin*) с двумя уровнями приоритетов (приоритетные и не приоритетные), образуя соответственно цикл приоритетных задач (ЦПЗ) и цикл не приоритетных задач (ЦНПЗ).

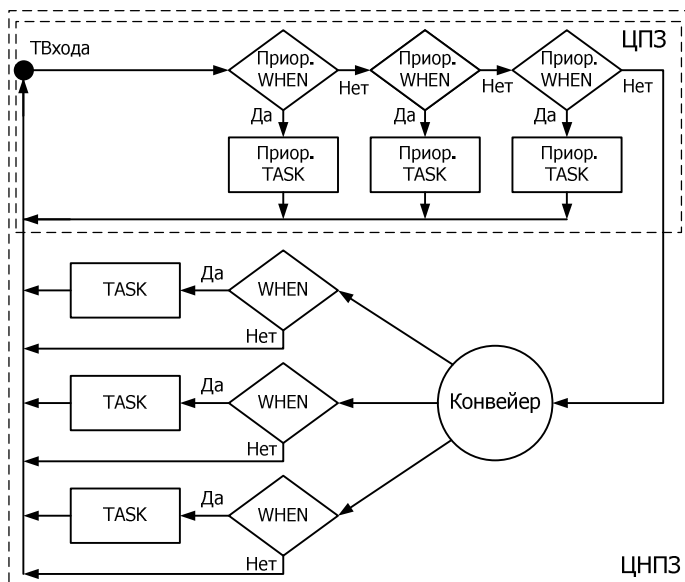


Рис. 1. Циклический алгоритм обслуживания задач приложения

Началом работы программы является точка входа (в литературе встречается термин «точка завершения критической секции»), в рамках которой выполняется обработка входящих (потенциальных событий) и исходящих (результата уже обработанной задачи) сообщений уровня приложения. Затем управление передается циклу приоритетных задач, в рамках которого происходит последовательное (согласно расположению) выполнение приоритетных модулей – проверок *when* и задач *task*. В случае успешного выполнения проверки условия и последующего решения прикладной задачи управление в программе передается в точку входа, совершая при этом фрагмент ЦПЗ, начинается новый цикл приоритетных задач. ЦПЗ завершается, если в ходе последовательной проверки всех модулей не произведено успешных проверок *when*. За время до завершения ЦПЗ может быть выполнено множество задач, при этом в каждой новой итерации (повторении) фрагмента приоритетного цикла количество проверок начальных условий (в порядке расположения) больше либо равно количеству вышестоящих проверок. Таким образом, за счет более частных проверок начальных модулей время реакции на соответствующие им проверки меньше, чем время для вышестоящих модулей, и зависит от числа приоритетных модулей  $n$  программы. При проектировании приложения следует учитывать данный факт и располагать модули в порядке важности задач (начиная с наиболее значимых).

По завершении ЦПЗ управление программой передается к ЦНПЗ. Цикл неприоритетных задач выполняется за одну итерацию, обрабатывая фрагмент цикла с модулем, следующим в порядке расположения. После обработки управление передается в точку входа. Таким образом, время реакции на неприоритетное событие в программе может включать время обслуживания всех фрагментов и полных циклов приоритетных и предшествующих неприоритетных циклов.

Как отмечено, значимым фактором, влияющими на время реакции, является количество циклов обслуживания, предшествующих работе с целевым (для которого наступило событие) модулем. В свою очередь, количество циклов определяется частотой (интенсивностью) происхождения событий или, что то же самое, интенсивностью срабатывания проверок модулей.

Решающим фактором, влияющим на интенсивность срабатывания проверок являются моменты происхождения событий и, в частности,

интервалы времени между событиями. Учет данного фактора особо важен ввиду малой вычислительной мощности узла, обуславливающей большое время обработки точки входа  $T_V$ , время проверки  $T_W$ , время задачи  $T_T$ .

Критерием «успешности» завершения проверки является конфигурационное свойство (*configuration property*), используемое в условии *when* и описывающее тип и граничные значения параметров проверки. Наиболее распространены пять видов конфигурационных свойств [11, 12]: 1) тактовые импульсы передачи (максимальное время между передачами, *Heartbeat, maxSndTime*); 2) тактовые импульсы приема (*maxRcvTime*); 3) дроссель передачи (минимальное время между передачами, *Throttle, minSndTime*); 4) дельта передачи (минимальное изменение контролируемого параметра, *sndDelta, minDeltaLevel*); 5) уровень передачи (изменение параметра, связанное с выходом за допустимые границы его значения, *minLevel, maxLevel*). Учет представленных факторов и их анализ для оценки производительности в известных источниках не проводились. Поэтому в рамках настоящей работы авторами предлагается следующий принцип учета данного фактора. Каждая проверка, вне зависимости от типа, характеризуется различными интервалами времени между наступлением событий, которые могут быть описаны различными законами распределения времени. Тактовые импульсы передачи/приема можно описать равномерным распределением, поскольку для них задаются и соблюдаются практически фиксированные интервалы между событиями. Остальные проверки ввиду неопределенности моментов времени их срабатывания могут быть описаны экспоненциальным законом распределения времени.

**Анализ обслуживания цикла приоритетных задач.** Выполним анализ приложения с  $n$  приоритетными модулями и без неприоритетных модулей. Тогда цикл обслуживания задач может быть представлен следующим образом (рис. 2).

Рассмотрим наихудший по времени реакции на событие сценарий работы приложения. Предположим, что событие модуля  $x$  наступает в момент времени неуспешного завершения  $x$ -й проверки *when*. Тогда среднее время реакции  $T_{Rx}$  на данное событие представляет собой сумму времени завершения текущего цикла приоритетных задач  $T_{Ex}$ , времени от начала цикла и до начала  $x$ -й проверки  $T_{Bx}$ , а также полного времени обработки задачи модуля  $x$ :

$$T_{Rx} = T_{Ex} + T_{Bx} + [T_V + T_{Wx} + T_{Tx}] \quad (1)$$

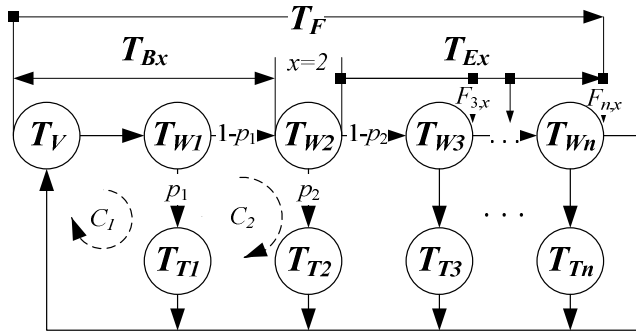


Рис. 2. Цикл приоритетных задач

Типовое время обработки  $T_V$ ,  $T_W$ ,  $T_T$  известно из литературы [7, 11].

Время окончания цикла  $T_{Ex}$  может быть определено суммой времени обработки каждого  $i$ -го модуля ( $i > x$ ,  $i = x + 1..n$ ), равного времени осуществления проверки и решения задачи с вероятностью наступления события  $p_i$ , и обязательным совместным условием  $F_{i,x}$  того, что все предшествующие (начиная с  $x + 1$ ) модулю  $i$  проверки не успешны:

$$T_{Ex} = \begin{cases} 0, & \text{при } x = n, \\ \sum_{i=x+1}^n [F_{i,x}(T_{Wi} + p_i \cdot T_{Ti})], & \text{при } x < n. \end{cases} \quad (2)$$

Вероятность успешного результата проверки при обработке  $i$ -го модуля определяется временем  $t_i$ , прошедшим с момента последней проверки модуля, и средним временем  $T_i$  между событиями этого модуля. В зависимости от типа конфигурационного свойства условия проверки вероятность может быть описана экспоненциальным (3) и равномерным (4) распределением.

$$p_i = 1 - e^{-t_i/T_i}, \quad (3)$$

$$p_i = \begin{cases} t_i/T_i, & \text{при } t_i \leq T_i, \\ 1, & \text{при } t_i > T_i. \end{cases} \quad (4)$$

Функция вероятности  $F_{i,x}$  того, что все модули после  $x$  и до  $i$  были не успешны, может быть определена совместными вероятностями неуспешных проверок *when*:

$$F_{i,x} = \begin{cases} 1, & \text{при } i = x + 1, \\ \prod_{k=x+1}^{i-1} (1 - p_k), & \text{при } i > x + 1. \end{cases} \quad (5)$$

Определим время  $T_{Bx}$  с начала цикла и до начала обработки модуля  $x$ . Это время равняется времени обработки точки входа при  $x = 1$ . При  $x > 1$  время определяется суммой времени обработки каждого модуля программы до  $x$ , определяемого произведением числа фрагментов цикла ПЗ, и соответствующей продолжительности фрагмента цикла:

$$T_{Bx} = \begin{cases} T_V, & \text{при } x = 1, \\ T_V + \sum_{i=1}^{x-1} \left[ \sum_{o=1}^i (T_V + T_{W_o} + T_{T_o}) \cdot \prod_{k=1}^o C_k + T_{W_o} \cdot M_{o,x} \right], & \text{при } x < n, \end{cases} \quad (6)$$

где  $C_k$  – среднее количество подряд следующих циклов обработки модуля  $k$  при первом обращении (и срабатывании) к нему (или частота выполнения задачи модуля  $k$ ),  $M_{o,x}$  – частота выполнения неуспешных проверок *when* в рамках фрагментов цикла до модуля  $x$ :

$$C_i = \sum_{k=1}^{10} k \cdot p_i^k, \quad (7)$$

$$M_{o,x} = \begin{cases} 1, & \text{при } o + 1 = x, \\ 1 + C(o + 1), & \text{при } o + 1 > x. \end{cases} \quad (8)$$

Приведем пример расчета времени доступа к модулю  $x$  с номером 2 по формуле (6):

$$T_{B2} = T_V + (T_V + T_{W_1} + T_{T_1})C_1 + T_{W_1}.$$

Таким образом, время от начала цикла до модуля 2 составляет сумму времени точки входа и одной проверки, предшествующей целевому модулю, а также времени выполнения  $C_1$  циклов обслуживания модуля  $x$  с номером 1.

Сложность численного расчета среднего времени реакции узла на событие в программе по предложенным соотношениям заключается в сложности определения времени  $t_i$ , прошедшего с момента последней проверки модуля  $i$ . Данное время влияет на вероятность (3, 4) успешной проверки модуля. Вероятность составляет основу расчета количества циклов обслуживания модулей и проверок (7, 8), тем самым влияет на время начала (6) и окончания (2) цикла и соответственно на время  $t_i$ . В условиях обозначенной взаимозависимости параметров для выполнения количественных расчетов необходимо задаться определенным значением указанного времени. В рамках данной работы мы предлагаем считать время  $t_i$ , прошедшее с последней проверки модуля  $i$ ,

равным времени полного приоритетного цикла  $T_F$  задач с последней решаемой задачей модуля  $n$ :

$$t_i = T_F = T_V + \sum_{i=1}^n T_{W_i} + T_{T_n}. \quad (9)$$

Введенное допущение справедливо и описывает верхнюю границу возможного времени  $t_i$  для малой нагрузки на программу, когда выполнение задач производится редко. Для средней и высокой нагрузки в приложении реальное время  $t_i$  для малых  $i$  меньше принятого  $T_F$ , что будет приводить к появлению дополнительных циклов и увеличению времени реакции, а для модулей программы с большими номерами  $i$  время  $t_i > T_F$ , что будет немного занижать расчетное время реакции для соответствующих задач ввиду меньшей вероятности  $p_i$  (3, 4) успешного завершения проверки.

**Количественный анализ среднего времени реакции.** Выполним анализ среднего времени реакции программы узла на события. Целью анализа являются исследование адекватности результатов, получаемых с помощью соотношений (1–9), а также выработка рекомендаций к проектированию приложения узла ИУС. Критерием адекватности результатов примем подтверждение ожидаемого (известного из анализа) характера изменения времени реакции на различные события в программе:

- **Среднее время реакции на события начальных (в порядке расположения в программе) модулей** всегда меньше либо равно времени реакции последующих модулей, т.е.  $T_{R_i} \leq T_{R_x}$  при  $i < x$ . Это справедливо, поскольку каждой последующей проверке предшествуют циклы обслуживания ранних модулей. Исключением являются прикладные сценарии, характеризующиеся перегруженностью событиями одного или нескольких модулей (как правило, начальных), что приводит к уменьшению времени реакции на них и увеличению времени для вышестоящих задач.

- Изменение среднего времени реакции по модулям 1.. $n$  программы при малой интенсивности событий, должно носить линейный возрастающий характер, поскольку задачи выполняются редко. С увеличением интенсивности событий (уменьшением времени между событиями) кривая изменения времени должна приобретать форму экспоненты, поскольку время реакции на каждую последующую задачу включает в себя время всех предшествующих циклов.



Особый интерес представляет анализ эффективности перехода от приложения с  $n$  модулями к  $j$  совместным модулям ( $j < n$ ), объединяющим несколько проверок и задач, при условии неизменной суммарной интенсивности событий  $\lambda_{\Sigma} = T_{\Sigma}^{-1}$ . При этом время совместной проверки увеличивается кратно числу объединенных условий *when*. Предполагается, что снижение количества проверок *when* должно привести к уменьшению времени реакции узла для каждого модуля ввиду сокращения числа предшествующих фрагментов цикла ПЗ.

Для анализа указанных положений количественный анализ среднего времени реакции программы узла на события выполняется для следующих исходных данных (таблица).

Исходные данные

№	$n$	$T_i(T_{1..n})$ (мс)	$T_V$ (мс)	$T_W$ (мс)	$T_T$ (мс)
1	10	1000	1	0,3	1
2		50			
3		20			
4	5	500		0,6	
5		25			
6		10			
7	10	500 ( $T_3 = 10$ )		0,3	
8		50 ( $T_4 = 5$ )			

Оценка времени реакции для исходных данных № 1–6 выполняется с целью установления общего соответствия результатов аналитических соотношений (1–9) представленным ранее положениям. Результаты представлены на рис. 3.

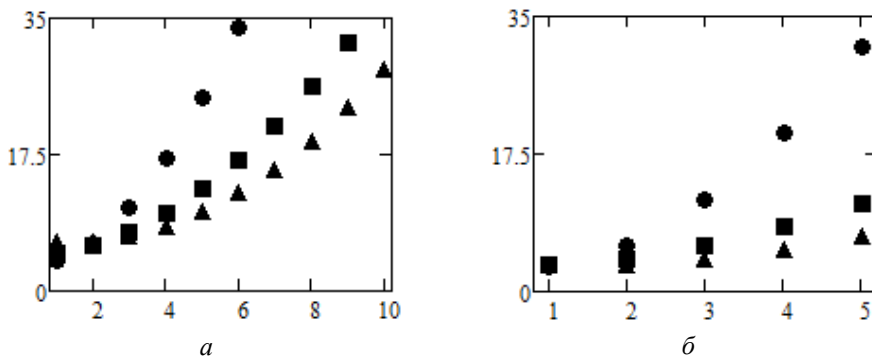


Рис. 3. Зависимость времени реакции на событие от номера модуля  $x$  при различном времени  $T$  между событиями:  $a - x (n = 10)$ ; № 1–3 /  $T_{1:10} = 1000 \blacktriangle, 50 \blacksquare, 20 \bullet$ ;  
 $b - x (n = 5)$ ; № 4–6 /  $T_{1:5} = 500 \blacktriangle, 25 \blacksquare, 10 \bullet$

Анализ характера изменения среднего времени реакции модуля на событие подтверждает обозначенные рациональные положения. Сравнивая попарно результаты оценок № 1–3 и № 4–6, подтверждается гипотеза эффективности перехода к программе с меньшим числом модулей (в данном случае от  $n$  равного 10 к 5) при неизменной суммарной интенсивности нагрузки, создаваемой модулями программы. При этом, среднее время реакции модулей сокращается более чем в два раза. Данное положение ранее не подтверждалось в известной литературе и может быть использовано на этапе проектирования для построения эффективных прикладных программ узлов ИУС.

Анализ приложений с исходными данными № 7–8 выполняется с целью подтверждения реакции программы на появление большой нагрузки, создаваемой определенным модулем. Согласно исходным данным исследуется влияние увеличения интенсивностей событий модулей 3 и 4 на время реакции модулями программы. Результаты представлены на рис. 4.

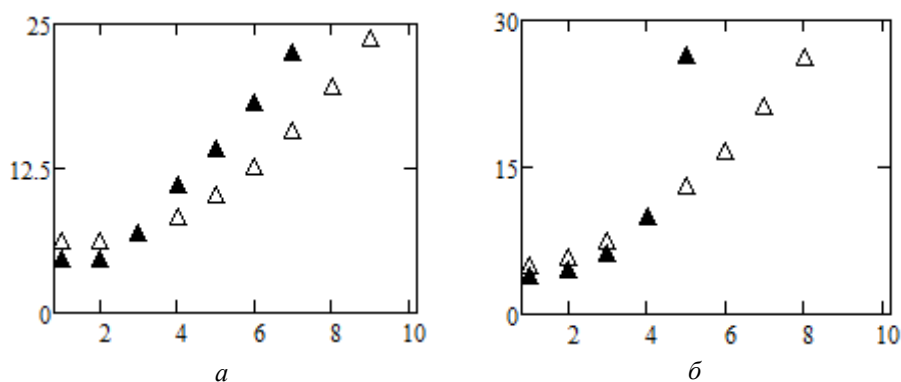


Рис. 4. Влияние номера модуля-источника интенсивных событий на время реакции модулей  $x$  программы:  $a - x (n = 10)$ ; № 7 /  $T_{1:10} = 500\Delta$ , ( $T_3=10$ ) ▲;  
 $b - x (n = 5)$ ; №7 /  $T_{1:10} = 50\Delta$ , ( $T_4=5$ ) ▲

Анализ полученных результатов позволил установить влияние модуля-источника интенсивной нагрузки на среднее время реакции для модулей программы. Ввод в программу высокоинтенсивного модуля  $x$  (модули 3 и 4, см. рис. 4,  $a$ ,  $b$  соответственно) приводит к уменьшению времени реакции для предшествующих его модулей, что обусловлено увеличением частоты проверок начальных модулей. Другим важным последствием введения интенсивного модуля является увеличение

времени реакции всех последующих за ним модулей, поскольку им начинает предшествовать большее количество фрагментов цикла ПЗ.

Таким образом, проведенное в рамках работы исследование циклического обслуживания задач в программе узла ИУС и выполненный анализ результатов оценки времени реакции на события в приложении подтверждают адекватность разработанных аналитических соотношений. Выполненная количественная оценка времени реакции и анализ позволили сформулировать рекомендации к проектированию приложения узла.

### Библиографический список

1. Miśkowitz M. Analysis of Mean Access Delay in Variable-Window CSMA // *Sensors. Schweiz: Molecular Diversity Preservation International*. – 2007. – Vol. 7. – P. 3535–3559.
2. Moshe Kam. Collision Resolution Simulation for Distributed Control Architectures using LonWorks // *IEEE International Conference on Automation Science and Engineering*. – Edmonton, Canada: IEEE, 2005.
3. Buchholz P., Plonnigs J. Analytical analysis of access-schemes of the CSMA type // *IEEE International Workshop on Factory Communication Systems*. – Wien, 2004. – P. 127–136.
4. Даденков С.А., Кон Е.Л. Исследование производительности алгоритма доступа к среде predictive p-persistent CSMA протокола // *Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления*. – 2012. – № 6. – С. 217–230.
5. Даденков С.А., Кон Е.Л. Оценка и анализ производительности LonWorks сети на основе predictive p-persistent CSMA протокола // *Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления*. – 2012. – № 6. – С. 206–216.
6. Даденков С.А., Кон Е.Л. Подход к построению аналитической модели информационно-управляющей сети LonWorks на основе нейрочипов // *Нейрокомпьютеры: разработка, применение*. – М.: Радиотехника, 2013. – № 11. – С. 64–69.
7. Miśkowitz M. Latency Characteristics of Event-Driven Task Scheduler Embedded in Neuron Chip // *IJCSNS International Journal of Computer Science and Network Security*. – 2007. – Vol. 7, No. 12. – P. 132–149.

8. Даденков С.А., Чмыков В.В. К проблеме анализа производительности промышленных сетей (на примере LonWorks) // Sworld: сб. науч. тр. – 2014. – Т.10, № 3. – С. 81–87.

9. Даденков С.А., Чмыков В.В. К вопросу анализа конфигурационных свойств узлов как фактора, влияющего на производительность сети LonWorks // Sworld: сб. науч. тр. – 2014. – Т. 5, № 1. – С. 84–90.

10. Даденков С.А., Чмыков В.В. Разработка имитационной модели уровня приложений узла LonWorks сети // Автоматизированные системы управления и информационные технологии: матер. краев. науч.-техн. конф. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2014. – С. 124–130.

11. Дитрих Д., Лой Д., Швайнциер Г.Ю. LON-технология, построение распределенных приложений: пер. с нем. / под ред. О.Б. Низамутдинова. – Пермь: Звезда, 1999. – 242 с.

12. LonTalk protocol specification, ANSI/CEA-709.1-B. United States: ISO/IEC JTC 1 SC 25, 2006.

## References

1. Miśkiewicz M. Analysis of Mean Access Delay in Variable-Window CSMA. Sensors. Schweiz: Molecular Diversity Preservation International, 2007, vol. 7, pp. 3535-3559.

2. Moshe Kam. Collision Resolution Simulation for Distributed Control Architectures using LonWorks. *IEEE International Conference on Automation Science and Engineering*. Edmonton, Canada: IEEE, 2005.

3. Buchholz P., Plonnigs J. Analytical analysis of access-schemes of the CSMA type. *IEEE International Workshop on Factory Communication Systems*. Wien, 2004, pp. 127-136.

4. Dadenkov S.A., Kon E.L. Issledovanie proizvoditel'nosti algoritma dostupa k srede predictive p-persistent CSMA protokola [Research of productivity of algorithm of media access of predictive p-persistent CSMA of the protocol]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2012, no. 6, pp. 217-230.

5. Dadenkov S.A., Kon E.L. Otsenka i analiz proizvoditel'nosti LonWorks seti na osnove predictive p-persistent CSMA protokola [Assessment and performance review of LonWorks of a network on the

basis of protocol predictive p-persistent CSMA]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2012, no. 6, pp. 206-216.

6. Dadenkov S.A., Kon E.L. Podkhod k postroeniiu analiticheskoi modeli informatsionno-upravliaiushchei seti LonWorks na osnove neurochipov [Approach to creation of analytical model of the management information LonWorks network on the basis of neurochips]. *Neirokomp'iutery: razrabotka, primeneniye*. Moscow: Radiotekhnika, 2013, no. 11, pp. 64-69.

7. Miśkowitz M. Latency Characteristics of Event-Driven Task Scheduler Embedded in Neuron Chip. *IJCSNS International Journal of Computer Science and Network Security*, 2007, vol. 7, no.12, pp. 132-149.

8. Dadenkov S.A., Chmykov V.V. K probleme analiza proizvoditel'nosti promyshlennykh setei (na primere LonWorks) [To a problem of performance review of industrial networks (on the example of LonWorks)]. Sworld: *sbornik nauchnykh trudov*, 2014, vol. 10, no. 3, pp. 81-87.

9. Dadenkov S.A., Chmykov V.V. K voprosu analiza konfiguratsionnykh svoistv uzlov, kak faktora, vliiaiushchego na proizvoditel'nost' seti LonWorks [To a question of the analysis of configuration properties of nodes, as the factor influencing LonWorks network productivity]. Sworld: *sbornik nauchnykh trudov*, 2014, vol. 5, no. 1, pp. 84-90.

10. Dadenkov S.A., Chmykov V.V. Razrabotka imitatsionnoi modeli urovnia prilozhenii uzla LonWorks seti [Development of a simulation model of level of applications of the LonWorks node of a network]. *Materialy kraevoi nauchno-tekhnicheskoi konferentsii Avtomatizirovannye sistemy upravleniia i informatsionnye tekhnologii*. Permskii natsional'nyi issledovatel'skii politekhnicheskii universitet, 2014, pp. 124-130.

11. Ditrikh D., Loi D., Shvaintser G.Iu., Nizamutdinov O.B. LON-tekhnologiia, postroenie raspredelennykh prilozhenii: perevod s nemetskogo [LON technology, creation of the distributed applications: translation from German]. Perm': Zvezda, 1999. 242 p.

12. LonTalk protocol specification, ANSI/CEA-709.1-B. United States: ISO/IEC JTC 1 SC 25, 2006.

### **Сведения об авторах**

**Даденков Сергей Александрович** (Пермь, Россия) – ассистент кафедры автоматике и телемеханики Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: dadenkov@rambler.ru).

**Кон Ефим Львович** (Пермь, Россия) – кандидат технических наук, профессор кафедры автоматике и телемеханики Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: kel-40@yandex.ru).

### **About the authors**

**Dadenkov Sergey Alexandrovich** (Perm, Russian Federation) is an Assistant at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky prospect, e-mail: dadenkov@rambler.ru).

**Kon Efim Lvovich** (Perm, Russian Federation) is Ph.D. of Technical Sciences, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky prospect, e-mail: kel-40@yandex.ru).

Получено 12.12.2014