СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Научный журнал Российской академии наук
(издается под руководством Отделения нанотехнологий и информационных технологий РАН)

Издается с 1989 года
Журнал выходит ежеквартально

Учредитель:
Федеральный исследовательский центр «Информатика и управление» Российской академии наук

Журнал включен в базу данных Russian Science Citation Index (RSCI), интегрированную с Web of Science
Журнал входит в систему Российского индекса научного цитирования (РИНЦ)
Журнал включен в базу данных CrossRef (систему DOI — Digital Object Identifier), в базу данных Ulrich's periodicals directory и в информационную систему «Общероссийский математический портал Math-Net.Ru»

Журнал реферируется в «Реферативном журнале» ВИНИТИ и в системе Google Scholar

Журнал включен в сформированный Минобрнауки России Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук

http://www.ipiran.ru/journal/collected

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 28 № 1 Год 2018

СОДЕРЖАНИЕ

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 28 № 1 Год 2018

СОДЕРЖАНИЕ

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 4–19

АНАЛИТИЧЕСКИЙ СИНТЕЗ СУБОПТИМАЛЬНЫХ ФИЛЬТРОВ МЕТОДАМИ МОМЕНТОВ

И. Н. Синицын[1], *В. И. Синицын*[2], *Э. Р. Корепанов*[3]

Аннотация: На основе методов начальных и центральных моментов (МНМ и МЦМ) рассматривается теория аналитического синтеза по критерию минимума средней квадратической (с.к.) ошибки субоптимальных и модифицированных субоптимальных фильтров (СОФ и МСОФ) для нелинейных дифференциальных стохастических систем (СтС), в том числе на гладких многообразиях. Предполагается, что уравнение состояния СтС содержит гауссовские и пуассоновские шумы, а уравнение наблюдения — только гауссовские шумы. Полученные алгоритмы позволяют оценивать влияние на точность и чувствительность инструментальных параметров, а также изучать зависимости от порядка учитываемых вероятностных моментов в разложении апостериорной плотности. Разработано два тестовых примера из области фильтрации угловых процессов в нелинейной с параметрическим шумом информационно-измерительной системе. Алгоритмы положены в основу модуля инструментальной программной системы StS-Filter 2018. Сформулированы возможные обобщения.

Ключевые слова: апостериорное распределение (плотность, характеристическая функция); гауссовский шум; метод начальных моментов (МНМ); метод центральных моментов (МЦМ); модифицированный СОФ (МСОФ); ненормированное апостериорное распределение; нормированное апостериорное распределение; пуассоновский шум; стохастическая система (СтС); субоптимальный фильтр (СОФ); угловая информационно-измерительная система

DOI: 10.14357/08696527180101

1 Введение

В [1] рассмотрены вопросы оценки точности и чувствительности алгоритмов параметрического аналитического моделирования одно- и многомерных распределений в СтС на многообразиях (МСтС) с винеровскими и пуассоновскими шумами на базе МНМ и МЦМ. Выведены уравнения точности и чувствительности методов аналитического моделирования (МАМ). Рассмотрены возможности со-

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sinitsin@dol.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, VSinitsyn@ipiran.ru

[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ekorepanov@ipiran.ru

кращения числа уравнений МНМ и МЦМ для одно- и многомерных распределений. Результаты положены в основу модуля разрабатываемого инструментального символьного программного обеспечения МАМ в среде MATLAB-MAPLE.

Основываясь на [2–4], рассмотрим методы синтеза СОФ для оценивания процессов в нелинейных СтС, в том числе и МСтС, на основе МНМ и МЦМ. При этом, во-первых, будем предполагать, что уравнения состояния СтС содержат гауссовские и пуассоновские шумы; во-вторых, уравнения наблюдения содержат только гауссовские шумы; в-третьих, будем рассматривать СОФ и МСОФ на основе МНМ и МЦМ. Дадим обобщение [5, 6] на случай негауссовских шумов. В [6] на базе гауссовского апостериорного распределения рассмотрены СОФ и МСОФ.

В разд. 2, следуя [5, 6], приведены исходные уравнения дифференциальных МСтС и фильтрационные уравнения для нормированных и ненормированных распределений. Раздел 3 посвящен СОФ и МСОФ на основе МНМ. В разд. 4 рассматриваются СОФ и МСОФ на базе МЦМ. Раздел 5 содержит сведения по оценке чувствительности СОФ и МСОФ на основе МНМ (МЦМ). Тестовые примеры приводятся в разд. 6. Заключение содержит основные выводы и возможные обобщения.

2 Фильтрационные уравнения для нормированного и ненормированного распределений

Следуя [4, 7], будем рассматривать задачу фильтрации состояния систем, моделями которых могут служить стохастические дифференциальные уравнения, понимаемые в смысле Ито. При этом стохастические дифференциальные уравнения модели изучаемой системы могут иметь неизвестные параметры и, как правило, всегда содержат параметры, известные с ограниченной точностью. Поэтому возникает задача непрерывного оценивания неизвестных параметров системы (точнее, ее модели) по результатам непрерывных наблюдений.

Предположим, что правые части уравнений зависят от конечного множества неизвестных параметров, которые будем рассматривать как компоненты вектора параметров Θ. Одним из возможных подходов в таких случаях является следующий прием: неизвестный векторный параметр Θ считают стохастическим процессом $\Theta = \Theta_t$, который определяется дифференциальным уравнением $\dot{\Theta}_t = 0$, и включают компоненты этого векторного процесса в вектор состояния системы («расширяют» вектор состояния путем включения в него неизвестных параметров в качестве дополнительных компонент). Таким образом, задача непрерывного оценивания неизвестных параметров модели системы сводится к задаче непрерывного оценивания состояния системы с расширенным вектором состояния. От неизвестных параметров могут зависеть и уравнения наблюдения. Эти параметры следует включить в вектор Θ и, следовательно, в расширенный вектор состояния.

Итак, пусть векторный стохастический процесс (СтП) $\left[X_t^{\mathrm{T}} Y_t^{\mathrm{T}}\right]^{\mathrm{T}}$ определяется системой векторных стохастических дифференциальных уравнений Ито [4]:

$$dX_t = \varphi\left(X_t, Y_t, \Theta, t\right) dt + \psi'\left(X_t, Y_t, \Theta, t\right) dW_0 + \\ + \int\limits_{R_0^q} \psi''\left(X_t, Y_t, \Theta, t, v\right) P^0(\Theta, dt, dv)\,, \quad X\left(t_0\right) = X_0\,; \quad (1)$$

$$dY_t = \varphi_1\left(X_t, Y_t, \Theta, t\right) dt + \psi_1'\left(X_t, Y_t, \Theta, t\right) dW_0 + \\ + \int\limits_{R_0^q} \psi_1''\left(X_t, Y_t, \Theta, t, v\right) P^0(\Theta, dt, dv), \quad Y\left(t_0\right) = Y_0\,. \quad (2)$$

Здесь $Y_t = Y(t)$ — n_y-мерный наблюдаемый СтП, $Y_t \in \Delta^y$ (Δ^y — гладкое многообразие наблюдений); $X_t = X(t)$ — n_x-мерный ненаблюдаемый СтП (вектор состояния), $X_t \in \Delta^x$ (Δ^x — гладкое многообразие состояний); $W_0 = W_0(\Theta, t)$ — n_w-мерный винеровский СтП ($n_w \geq n_y$) интенсивности $\nu_0 = \nu_0(\Theta, t)$; $P^0(\Theta, \Delta, A) = P(\Theta, \Delta, A) - \mu_P(\Theta, \Delta, A)$, $P(\Theta, \Delta, A)$ представляет собой для любого множества A простой пуассоновский СтП, а $\mu_P(\Theta, \Delta, A)$ — его математическое ожидание, причем

$$\mu_P(\Theta, \Delta, A) = \mathsf{M} P(\Theta, \Delta, A) = \int\limits_{\Delta} \nu_P(\Theta, \tau, A)\, d\tau\,;$$

$\nu_P(\Theta, \Delta, A)$ — интенсивность соответствующего пуассоновского потока событий, $\Delta = (t_1, t_2]$; интегрирование по v распространяется на все пространство R^q с выколотым началом координат; Θ — вектор случайных параметров размерности n_Θ; $\varphi = \varphi(X_t, Y_t, \Theta, t)$, $\varphi_1 = \varphi_1(X_t, Y_t, \Theta, t)$, $\psi' = \psi'(X_t, Y_t, \Theta, t)$, $\psi_1' = \psi_1'(X_t, Y_t, \Theta, t)$ — известные функции, отображающие $R^{n_x} \times R^{n_y} \times R$ соответственно в R^{n_x}, R^{n_y}, $R^{n_x n_w}$ и $R^{n_y n_w}$; $\psi'' = \psi''(X_t, Y_t, \Theta, t, v)$ и $\psi_1''(X_t, Y_t, \Theta, t, v)$ — известные функции, отображающие $R^{n_x} \times R^{n_y} \times R^q$ в R^{n_x} и R^{n_y}. Требуется найти оценку $\hat{X}_t$ СтП X_t в каждый момент времени t по результатам наблюдения СтП $Y(\tau)$ до момента t, $Y_{t_0}^t = \{Y(\tau) : t_0 \leq \tau < t\}$.

Предположим, что выполнены условия [4]:

- уравнение состояния имеет вид (1);
- уравнение наблюдения (2), во-первых, не содержит пуассоновского шума ($\psi_1'' \equiv 0$), а во-вторых, коэффициент при винеровском шуме ψ_1' в уравнениях наблюдения не зависит от состояния ($\psi_1'(X_t, Y_t, \Theta, t) = \psi_1'(Y_t, \Theta, t)$).

В этом случае уравнения задачи нелинейной фильтрации имеют следующий вид:

$$dX_t = \varphi\left(X_t, Y_t, \Theta, t\right) dt + \psi'\left(X_t, Y_t, \Theta, t\right) dW_0 + \\ + \int\limits_{R_0^q} \psi''\left(X_t, Y_t, \Theta, t, v\right) P^0(\Theta, dt, dv)\,, \quad X\left(t_0\right) = X_0\,; \tag{3}$$

$$dY_t = \varphi_1\left(X_t, Y_t, \Theta, t\right) dt + \psi_1\left(Y_t, \Theta, t\right) dW_0\,, \quad Y\left(t_0\right) = Y_0\,. \tag{4}$$

Как известно [4], для любых СтП X_t и Y_t оптимальная оценка $\hat{X}^t$, минимизирующая средний квадрат ошибки в каждый момент времени t, представляет собой апостериорное математическое ожидание СтП X_t: $\hat{X}_t = \mathsf{M}\left[X_t \mid Y_{t_0}^t\right]$. Чтобы найти это условное математическое ожидание, необходимо знать $p_t = p_t(x, \Theta)$ и $g_t = g_t(\lambda, \Theta)$ — апостериорные нормированные одномерную плотность и характеристическую функцию распределения СтП X_t.

Введем ненормированные одномерные апостериорные плотность $\tilde{p}_t(x, \Theta)$ и характеристическую функцию $\tilde{g}_t(\lambda, \Theta)$ согласно формулам [4]:

$$\tilde{p}_t(x, \Theta) = \mu_t p_t(x, \Theta)\,; \quad \tilde{g}_t(\lambda, \Theta) = \mathsf{M}_{\Delta^x}^{p_t}\left[e^{i\lambda^{\mathrm{T}} X_t}\mu_t\right] = \mu_t g_t(\lambda, \Theta)\,.$$

Тогда, обобщая [4] на случай уравнений (3) и (4), получим следующее точное уравнение с.к. оптимальной нелинейной фильтрации на основе нормированного распределения:

$$dg_t(\lambda, \Theta) = \mathsf{M}\Bigg[\Bigg\{i\lambda^{\mathrm{T}}\varphi(X_t, Y_t, \Theta, t) - \frac{1}{2}\,\lambda^{\mathrm{T}}(\psi\nu_0\psi^{\mathrm{T}})\left(X_t, Y_t, \Theta, t\right)\lambda + \\ + \int\limits_{R_0^q}\Big[e^{i\lambda^{\mathrm{T}}\psi''(X_t, Y_t, \Theta, t, v)} - 1 - \\ - i\lambda^{\mathrm{T}}\psi''\left(X_t, Y_t, \Theta, t, v\right)\Big]\,\nu_P(\Theta, t, dv)\Bigg\}e^{i\lambda^{\mathrm{T}} X_t} \mid Y_{t_0}^t\Bigg]dt + \\ + \mathsf{M}\Bigg[\Bigg\{\varphi_1\left(X_t, Y_t, \Theta, t\right)^{\mathrm{T}} - \hat{\varphi}_1^{\mathrm{T}} + \\ + i\lambda^{\mathrm{T}}\Big(\psi\nu_0\psi_1^{\mathrm{T}}\Big)\left(X_t, Y_t, \Theta, t\right)e^{i\lambda^{\mathrm{T}} X_t} \mid Y_{t_0}^t\Bigg\}\Big(\psi_1\nu_0\psi_1^{\mathrm{T}}\Big)^{-1}\left(Y_t, \Theta, t\right)\left(dY_t - \hat{\varphi}_1 dt\right)\Bigg]. \tag{5}$$

Здесь оператор M берется в виде $\mathsf{M} = \mathsf{M}_{\Delta^x}^{p_t}$.

Аналогично, в случае ненормированного распределения, полагая $\mathsf{M} = \mathsf{M}_{\Delta^x}^{\tilde{p}_t}$, имеем:

$$d\tilde{g}_t(\lambda,\Theta)=\mathsf{M}_{\Delta^x}^{\tilde{p}_t}\left\{\left[i\lambda^{\mathrm{T}}\varphi\left(X_t,Y_t,\Theta,t\right)-\frac{1}{2}\left(\psi'\nu_0\psi'^{\mathrm{T}}\right)\left(X_t,Y_t,\Theta,t\right)+\right.\right.$$

$$\left.\left.+\int\limits_{R_0^q}\left[e^{i\lambda^{\mathrm{T}}\psi''(X_t,Y_t,\Theta,t,v)}-1-i\lambda^{\mathrm{T}}\psi''\left(X_t,Y_t,\Theta,t,v\right)\right]\nu_P(\Theta,t,dv)\right]e^{i\lambda^{\mathrm{T}}X_t}\right\}dt+$$

$$+\mathsf{M}_{\Delta^x}^{\tilde{p}_t}\left\{\left[\varphi_1\left(X_t,Y_t,\Theta,t\right)^{\mathrm{T}}+\right.\right.$$

$$\left.\left.+i\lambda^{\mathrm{T}}\left(\psi'\nu_0\psi'^{\mathrm{T}}\right)\left(X_t,Y_t,\Theta,t\right)\right]e^{i\lambda^{\mathrm{T}}X}\right\}\left(\psi'\nu_0\psi'^{\mathrm{T}}\right)^{-1}\left(Y_t,\Theta,t\right)\,dY_t\,. \quad (6)$$

Если функция ψ'' в (3) допускает представление

$$\psi''=\psi'\omega(\Theta,v)\,, \quad (7)$$

где $P^0(\Theta,\Delta,A)=P^0(\Theta,(0,t],dv)$, то уравнения (3), (4) примут следующий вид:

$$\dot{X}_t=\varphi\left(X_t,Y_t,\Theta,t\right)+\psi'\left(X_t,Y_t,\Theta,t\right)V(\Theta,t)\,,\;\; X\left(t_0\right)=X_0\,; \quad (8)$$

$$\dot{Y}_t=\varphi\left(X_t,Y_t,\Theta,t\right)+\psi_1\left(Y_t,\Theta,t\right)V_0(\Theta,t)\,,\qquad Y\left(t_0\right)=Y_0\,. \quad (9)$$

Здесь

$$V_0(\Theta,t)=\dot{W}_0(\Theta,t)\,;\quad V(\Theta,t)=\dot{\bar{W}}(\Theta,t)\,;$$

$$\bar{W}(\Theta,t)=W_0(\Theta,t)+\int\limits_{R_0^q}\omega(\Theta,v)P^0(\Theta,(0,t],dv),$$

где $\nu_P(\Theta,t,v)\,dv=[\partial\mu(\Theta,t,v)/\partial t]\,dv$ — интенсивность пуассоновского потока скачков, равных $\omega(\Theta,t)$. При этом логарифмические производные от одномерных характеристических функций определяются известными формулами:

$$\chi^{\bar{W}}(\rho;\Theta,t)=-\frac{1}{2}\,\rho^{\mathrm{T}}\nu_0(\Theta,t)\rho+\int\limits_{R_0^q}\left[e^{i\rho^{\mathrm{T}}\omega(\Theta,v)}-1-i\rho^{\mathrm{T}}\omega(\Theta,v)\right]\nu_P(\Theta,t,v)\,dv\,.$$

В таком случае интегральный член в (5) и (6) допускает следующую запись:

$$\gamma=\int\limits_{R_0^q}\left[e^{i\lambda^{\mathrm{T}}\psi''(X_t,Y_t,\Theta,t)\omega(\Theta,v)}-1-\right.$$

$$\left.-i\lambda^{\mathrm{T}}\psi''\left(X_t,Y_t,\Theta,t\right)\omega(\Theta,v)\right]\nu_P(\Theta,t,v)\,dv\,. \quad (10)$$

Очевидно, что для гауссовской МСтС $\gamma\equiv 0$. Тогда приходим к известным утверждениям [5–7].

Теорема 2.1. *Пусть для негауссовской MCmC* (3), (4) *выполнены условия существования и единственности. Тогда уравнение с.к. оптимальной нелинейной фильтрации для ненормированной характеристической функции* $\tilde{g}_t(\lambda, \Theta)$ *имеет вид* (6).

Теорема 2.2. *Пусть для гауссовской MCmC* (8), (9) *выполнены условия существования и единственности. Тогда уравнение с.к. оптимальной нелинейной фильтрации для ненормированной характеристической функции* $\tilde{g}_t(\lambda, \Theta)$ *имеет вид* (6) *при условии* (10).

3 Субоптимальные фильтры на основе метода начальных моментов

Как известно [4], необходимость обработки результатов наблюдений в реальном масштабе времени непосредственно в процессе эксперимента привела к появлению ряда приближенных методов оптимальной нелинейной фильтрации, называемых обычно методами субоптимальной фильтрации. В этом случае для приближенного решения уравнения для апостериорной одномерной характеристической функции $g_t(\lambda, \Theta)$ вектора X_t можно использовать МАМ, основанные на параметризации одномерных распределений СтП, определяемого стохастическим дифференциальным уравнением. Эти методы позволяют изучить стохастические дифференциальные уравнения для параметров апостериорного распределения. Простейшим таким методом является МНА апостериорного распределения.

Как известно из [4], фильтрационные уравнения (5) и (6) являются основой аналитического синтеза СОФ и МСОФ. Рассмотрим алгоритмы синтеза сначала на базе МНМ, а затем — МЦМ.

Если аппроксимировать апостериорную плотность $p_t(\Theta, x)$ вектора состояния X_t системы некоторой функцией $p^*(x, \Theta; \vartheta)$, зависящей не только от апостериорных математического ожидания $\hat{X}_t$ и ковариационной матрицы R_t вектора X_t, но и от его апостериорных моментов до порядка N включительно, то к уравнениям для $\hat{X}_t$ и R_t придется добавить уравнения для апостериорных начальных моментов α_r или центральных μ_r ($r_1, \ldots, r_{n_x} = 0, 1, \ldots, N$; $|r| = r_1 + \cdots + r_{n_x} = 3, \ldots, N$). Примем за ϑ вектор начальных моментов α_r ($r = 3, \ldots, N$). Тогда, повторяя вычисления [4] для гауссовской СтС, получим следующие стохастические дифференциальные уравнения для апостериорных начальных моментов вектора X_t:

$$d\alpha_r = A_r^{\alpha} = \beta_r\, dt + \eta_r \left(dY_t - f^{(1)}\, dt \right)$$
$$(r_1, \ldots, r_{n_x} = 0, 1, \ldots, N\,; \quad |r| = r_1 + \cdots + r_{n_x} = 1, \ldots, N)\,, \quad (11)$$

где введены следующие обозначения:

$$\beta_r = \beta_r\left(Y_t, \alpha, \Theta, t\right) =$$
$$= \sum_{s=1}^{n_x} r_s \mathsf{M}_{\Delta^x}^{p_t}\left\{\varphi_s\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_s^{r_s-1} \cdots X_{n_x}^{r_{n_x}}\right\} +$$
$$+ \frac{1}{2}\sum_{s=1}^{n_x} r_s(r_s-1)\mathsf{M}_{\Delta^x}^{p_t}\left\{\sigma_{ss}\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_s^{r_s-1} \cdots X_{n_x}^{r_{n_x}}\right\} +$$
$$+ \sum_{q=2}^{n_x}\sum_{s=1}^{q-1} r_s r_q \mathsf{M}_{\Delta^x}^{p_t}\left\{\sigma_{sq}\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_s^{r_s-1} \cdots X_q^{r_q-1} \cdots X_{n_x}^{r_{n_x}}\right\}; \quad (12)$$

$$\eta_r = \eta_r(Y_t, \alpha, \Theta, t) =$$
$$= \left\{\mathsf{M}_{\Delta^x}^{p_t}\left[\varphi_1\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_{n_x}^{r_{n_x}}\right] - f^{(1)T}\alpha_r +\right.$$
$$\left. + \sum_{s=1}^{n_x} r_s \mathsf{M}_{\Delta^x}^{p_t}\left[\left(\psi\nu\psi_1^{\mathrm{T}}\right)_s\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_s^{r_s-1} \cdots X_{n_x}^{r_{n_x}}\right]\right\} \times$$
$$\times \left(\psi_1\nu_0\psi_1^{\mathrm{T}}\right)^{-1}\left(Y_t, \Theta, t\right), \quad f^{(1)} = f^{(1)}(Y_t, \vartheta, \Theta, t) = \mathsf{M}_{\Delta^x}^{p_t}\left\{\varphi_1(Y_t, X, \Theta, t)\right\}, \quad (13)$$

а $\left(\psi\nu_0\psi_1^{\mathrm{T}}\right)_s$ — s-я строка матрицы $\psi\nu\psi_1^{\mathrm{T}}$. Интегрирование уравнений (11) при начальных значениях моментов α_r, равных соответствующим условным начальным моментам вектора X_0 относительно Y_0, приближенно определяет все апостериорные моменты $\alpha_r, r_1, \ldots, r_{n_x} = 0, 1, \ldots, N;\ |r| = 1, \ldots, N$, составляющие вектор параметров Θ.

При аппроксимации апостериорной плотности $p_t(x, \Theta)$ отрезком ортогонального разложения вида

$$p_t(x, \Theta) = p^*(x, \Theta; \vartheta) = w_1(x, \Theta)\left[1 + \sum_{k=3}^{N}\sum_{|\nu|=k} c_\nu p_\nu(x, \Theta)\right]. \quad (14)$$

Функции β_r, η_r и $f^{(1)}$ представляют собой линейные функции моментов α_r ($|r| = 3, \ldots, N$) с коэффициентами, зависящими от моментов первого и второго порядков:

$$\beta_r = \beta_{0,r} + \sum_{k=3}^{N}\sum_{|\nu|=k}\beta_{\nu,r} q_\nu(\alpha); \quad \eta_r = \eta_{0,r} + \sum_{k=3}^{N}\sum_{|\nu|=k}\eta_{\nu,r} q_\nu(\alpha);$$
$$f^{(1)} = f_0^{(1)} + \sum_{k=3}^{N}\sum_{|\nu|=k} f_\nu^{(1)} q_\nu(\alpha), \quad (15)$$

так как $c_\nu = q_\nu(\alpha)$, а величина $q_\nu(\alpha)$ при $|\nu| \geq 3$ представляет собой линейную функцию моментов α_r ($|r| = 3, \ldots, |\nu|$) с коэффициентами, зависящими от моментов первого и второго порядка.

При аппроксимации плотности $p_t(x, \Theta)$ отрезком ряда Эджуорта с учетом начальных моментов до N-го порядка число слагаемых в сумме по k увеличится до $3N - 6$ и коэффициенты c_ν при $|\nu| > N$ не будут равны $q_\nu(\alpha) = G_\nu(\mu)$, а будут функциями семиинвариантов до N-го порядка, которые надо будет заменить их выражениями через начальные моменты.

Таким образом, в основу аналитического синтеза СОФ может быть положено следующее утверждение.

Теорема 3.1. *Пусть уравнения нелинейной гауссовской дифференциальной СтС* (1) *и* (2) *при* $\psi'' = \psi_1'' = 0$ *допускают применение МНМ. Тогда алгоритм аналитического синтеза СОФ согласно МНМ определяется уравнениями* (11)–(15).

Аналогично, обобщая [4], получим следующие уравнения МСОФ для ненормированных моментов m_r до заданного порядка ($r = 1, \ldots, N$):

$$
\begin{aligned}
dm_r = A_r^m = \mu_t \mathsf{M}_{\Delta^x}^{p_t} \Bigg[& \sum_{l=1}^{n_x} r_l \varphi_l(Y_t, X, \Theta, t) X_1^{r_1} \cdots X_{n_x}^{r_{n_x}} + \\
& + \frac{1}{2} \sum_{l=1}^{n_x} r_l (r_l - 1) \sigma_{oll}\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_l^{r_l - 2} \cdots X_{n_x}^{r_{n_x}} + \\
& + \sum_{l=2}^{n_x} \sum_{h=1}^{n_x - 1} r_l r_h \sigma_{olh}\left(Y_t, X, \Theta, t\right) X_h^{r_h - 1} \cdots X_l^{r_l - 1} \cdots X_{n_x}^{r_{n_x}} \Bigg] dt + \\
& + \mu_t \mathsf{M}_{\Delta^x}^{p_t} \Bigg[\varphi_1\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} X_1^{r_1} \cdots X_{n_x}^{r_{n_x}} + \\
& + \sum_{l=1}^{n_x} r_l \sigma_{1l}\left(Y_t, X, \Theta, t\right) X_1^{r_1} \cdots X_l^{r_l - 1} \cdots X_{n_x}^{r_{n_x}} \Bigg] \sigma_2\left(Y_t, \Theta, t\right)^{-1} dY_t \\
& (r_1, \ldots, r_{n_x} = 0, 1, , \ldots, N, \ \ r = 1, \ldots, N) \, ; \qquad (16)
\end{aligned}
$$

$$
d\mu_t = A^\mu = \mu_t \mathsf{M}_{\Delta^x}^{p_t} \left[\varphi_1\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} \right] \sigma_2\left(Y_t, \Theta, t\right)^{-1} dY_t \, . \qquad (17)
$$

Это замкнутая система уравнений для m_r ($r = 1, \ldots, N$) и μ_t. Она приближенно определяет m_r, μ_t и α_r и, в частности, оптимальную оценку $\hat{X}_t$ и ковариационную матрицу ошибки фильтрации R_t. Начальные значения моментов m_r и $m_r(t_0)$ равны $\alpha_r(t_0)$, где $\alpha_r(t_0)$ — соответствующие условные моменты X_0 вектора X_t относительно Y_0 вектора Y_t, и $\mu(t_0) = 1$, как всегда.

Что касается выбора функции $\tilde{p}_t^*(x,\Theta,\alpha)$, то заранее можно принять конечный отрезок ортогонального разложения $\tilde{p}_t(x,\Theta)$, в частности разложения по полиномам Эрмита, или конечный отрезок ряда Эджуорта [4].

Пользуясь любой ортонормальной системой полиномов [4] $\{p_\nu(x)q_\nu(x)\}$,

$$\int_{-\infty}^{\infty} w(x,\Theta)p_\nu(x)q_\nu(x)\,dx = \delta_{\nu\mu} \tag{18}$$

($w(x,\Theta)$ — некоторая эталонная плотность), можно выбрать в качестве функции $\tilde{p}_t^*(x,\alpha,\Theta)$:

$$\tilde{p}_t^*(x,\Theta,\alpha) = w(x,\Theta)\left[1+\sum_{k=3}^{N}\sum_{|\nu|=k} c_\nu p_\nu(x)\right]. \tag{19}$$

Здесь

$$c_\nu = \mathsf{M}_{\Delta^x}^{p_t}\left[q_\nu(X)\right] = \mathsf{M}_{\Delta^x}^{w} q_\nu(X) = q_\nu(\alpha,\Theta)\,, \tag{20}$$

где $q_\nu(\alpha,\Theta)$ — результат замены в полиноме $q_\nu(x,\Theta)$ всех одночленов $X_1^{r_1}\cdots X_{n_x}^{r_{n_x}}$ соответствующими моментами $\alpha_r = \alpha_{r_1\cdots r_{n_x}}$. В частности, при нормальной плотности $w(x,\Theta)$ с параметрами $\hat{X}_t$ и R_t имеем:

$$p_\nu(x) = \frac{H(x-\hat{X}_t)}{\nu_1!\cdots\nu_{n_x}!}; \quad q_\nu(x) = G_\nu\left(x-\hat{X}_t\right), \tag{21}$$

где $H_\nu(\cdot)$ и $G_\nu(\cdot)$ — многомерные полиномы Эрмита [4].

Таким образом, имеем теорему.

Теорема 3.2. *Пусть уравнения нелинейной гауссовской дифференциальной системы* (1), (2) *при* $\psi'' = \psi_1'' = 0$ *допускают применение модифицированного МЦМ (М*^{2}*ЦМ). Тогда при фиксированном векторе инструментальных параметров* Θ *алгоритм аналитического синтеза МСОФ согласно М*^{2}*ЦМ определяется уравнениями* (16)–(21).

В случае, когда учитываются негауссовость шумов в уравнении состояния и используются уравнения (6)–(8), можно ввести эквивалентный белый шум интенсивности $\nu_э$, согласно условию

$$\frac{1}{2}\,\psi'\nu_э\psi'^{\mathrm{T}} = \frac{1}{2}\,\psi'\nu_0\psi'^{\mathrm{T}} = \gamma\,, \tag{22}$$

где γ определена (8), а затем воспользоваться теоремами 3.1 и 3.2. В результате придем к следующим утверждениям.

Теорема 3.3. *Пусть уравнения нелинейной негауссовской дифференциальной СтС* (6), (7) *допускают применение МНМ. Тогда алгоритм аналитического*

синтеза СОФ согласно МНМ определяется уравнениями теоремы 3.1 *с заменой* ν_0 *на* $\nu_э$ *согласно* (22).

Теорема 3.4. *Пусть уравнения нелинейной негауссовской дифференциальной СтС* (6), (7) *допускают применение МНМ. Тогда алгоритм аналитического синтеза МСОФ согласно МНМ определяется уравнениями теоремы* 3.2 *с заменой* ν_0 *на* $\nu_э$ *согласно* (22).

4 Субоптимальные фильтры на основе метода центральных моментов

Основываясь на [4] и применяя оператор $\mathsf{M}^{p^*}_{\Delta^x}$, придем к следующим уравнениям для СОФ нормированного апостериорного распределения при использовании МЦМ:

$$d\hat{X}_t = A^{\hat{X}_t} = f dt + h\left(dY_t - f^{(1)}dt\right), \tag{23}$$

$$dR_t = A^{R_t} = \left(f^{(2)} - h\psi_1\nu\psi_1^{\mathrm{T}}h^{\mathrm{T}}\right)dt + \sum_{r=1}^{n_y}\rho_r\left(dY_r - f_r^{(1)}dt\right), \tag{24}$$

где

$$f = f(Y_t, \vartheta, \Theta, t) = \mathsf{M}^{p^*}_{\Delta^x}\left[\varphi\left(Y_t, X, \Theta, t\right)\right]; \tag{25}$$

$$f^{(1)} = \left\{f_r^{(1)}\right\} = f^{(1)}\left(Y_t, \vartheta, \Theta, t\right) = \mathsf{M}^{p^*}_{\Delta^x}\left[\varphi_1\left(Y_t, X, \Theta, t\right)\right]; \tag{26}$$

$$f^{(2)} = f^{(2)}\left(Y_t, \vartheta, \Theta, t\right) = \mathsf{M}^{p^*}_{\Delta^x}\Big[(X - \hat{X}_t)\varphi\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} + \\ + \varphi\left(Y_t, X, \Theta, t\right)\left(X^{\mathrm{T}} - \hat{X}_t^{\mathrm{T}}\right) + \left(\psi\nu\psi^{\mathrm{T}}\right)\left(Y_t, X, \Theta, t\right)\Big]; \tag{27}$$

$$h = h\left(Y_t, \vartheta, \Theta, t\right) = \Big\{\mathsf{M}^{p^*}_{\Delta^x}\Big[X\varphi_1\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} + \\ + \left(\psi\nu_0\psi_1^{\mathrm{T}}\right)\left(Y_t, X, \Theta, t\right)\Big] - \hat{X}_t f^{(1)T}\Big\}\left(\psi_1\nu_0\psi_1^{\mathrm{T}}\right)^{-1}\left(Y_t, t\right); \tag{28}$$

$$\rho_r = \rho_r\left(Y_t, \vartheta, \Theta, t\right) = \mathsf{M}^{p^*}_{\Delta^x}\Big[\left(X - \hat{X}_t\right)\left(X^{\mathrm{T}} - \hat{X}_t^{\mathrm{T}}\right)a_r\left(Y_t, X, \Theta, t\right) + \\ + \left(X - \hat{X}_t\right)b_r\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} + b_r\left(Y_t, X, \Theta, t\right)\left(X^{\mathrm{T}} - \hat{X}_t^{\mathrm{T}}\right)\Big] \\ (r = 1, \ldots, n_y), \tag{29}$$

а ϑ — совокупность моментов $\hat{X}_t$, R_t и μ_r ($r_1, \ldots, r_{n_x} = 0, 1, \ldots, N$; $|r| = 3, \ldots, N$),

$$d\mu_r = A_r^{\mu} = \left(\beta_r - \sum_{s=1}^{n_x} r_s f_s \mu_{r-e_s} + \frac{1}{2} \sum_{s=1}^{n_x} r_s \left(r_s - 1\right) h_s \psi_1 \nu \psi_1^{\mathrm{T}} h_s^{\mathrm{T}} \mu_{r-2e_s} + \right.$$
$$\left. + \sum_{q=2}^{n_x} \sum_{s=1}^{q-1} r_s r_q h_s \psi_1 \nu \psi_1^{\mathrm{T}} h_q^{\mathrm{T}} \mu_{r-e_s-e_q} - \frac{1}{2} \sum_{s=1}^{n_x} r_s \eta_{r-e_s} \psi_1 \nu \psi_1^{\mathrm{T}} h_s \right) dt +$$
$$+ \left(\eta_r - \sum_{s=1}^{n} r_s h_s \mu_{r-e_s} \right) \left(dY_t - f^{(1)} dt \right) \quad (30)$$

$$\left(r_1, \ldots, r_{n_x} = 0, 1, \ldots, N; \;\; |r| = 3, \ldots, N\right).$$

Здесь введены следующие обозначения:

$$\beta_r = \beta_r \left(Y_t, \vartheta, \Theta, t\right) = \sum_{s=1}^{n_x} r_s \mathsf{M}_{\Delta^x}^{p^*} \Big[\varphi_s \left(Y_t, X, \Theta, t\right)^{\mathrm{T}} \left(X_1 - \hat{X}_1 \right)^{r_1} \cdots$$
$$\cdots \left(X_s - \hat{X}_s\right)^{r_s - 1} \cdots \left(X_{n_x} - \hat{X}_{n_x} \right)^{r_{n_x}} \Big] +$$
$$+ \frac{1}{2} \sum_{s=1}^{n_x} r_s \left(r_s - 1\right) \mathsf{M}_{\Delta^x}^{p^*} \Big[\sigma_{ss} \left(Y_t, X, \Theta, t\right)^{\mathrm{T}} \left(X_1 - \hat{X}_1 \right)^{r_1} \cdots \left(X_s - \hat{X}_s \right)^{r_s - 2} \cdots$$
$$\cdots \left(X_{n_x} - \hat{X}_{n_x} \right)^{r_{n_x}} \Big] + \sum_{q=2}^{n_x} \sum_{s=1}^{q-1} r_s r_q \mathsf{M}_{\Delta^x}^{p^*} \Big[\sigma_{sq} \left(Y_t, X, \Theta, t\right)^{\mathrm{T}} \left(X_1 - \hat{X}_1 \right)^{r_1} \cdots$$
$$\cdots \left(X_s - \hat{X}_s \right)^{r_s - 1} \cdots \left(X_q - \hat{X}_q \right)^{r_q - 1} \cdots \left(X_{n_x} - \hat{X}_{n_x} \right)^{r_{n_x}} \Big]; \quad (31)$$

$$\eta_r = \eta_r \left(Y_t, \vartheta, \Theta, t\right) = \left\{ \frac{\partial^{|r|} k(Y, \vartheta, \lambda, \Theta, t)}{\partial (i\lambda_1)^{r_1} \ldots \partial (i\lambda_{n_x})^{r_{n_x}}} \right\}_{\lambda = 0} =$$
$$= \Big\{ \mathsf{M}_{\Delta^x}^{p^*} \Big[\varphi_1 \left(Y_t, X, \Theta, t\right)^{\mathrm{T}} \left(X_1 - \hat{X}_1 \right)^{r_1} \cdots \left(X_{n_x} - \hat{X}_{n_x} \right)^{r_{n_x}} \Big] +$$
$$+ \sum_{s=1}^{n_x} r_s \mathsf{M}_{\Delta^x}^{p^*} \Big[(\psi \nu \psi_1^{\mathrm{T}})_s \left(Y_t, X, \Theta, t\right) \left(X_1 - \hat{X}_1 \right)^{r_1} \cdots \left(X_s - \hat{X}_s \right)^{r_s - 1} \cdots$$
$$\cdots \left(x_{n_x} - \hat{X}_{n_x} \right)^{r_{n_x}} \Big] - f^{(1)T} \mu_{r-e_s} \Big\} \left(\psi_1 \nu_0 \psi_1^{\mathrm{T}} \right)^{-1} \left(Y_t, \Theta, t\right), \quad (32)$$

где f_s — s-й элемент матрицы-столбца f, определяемый формулой (25). При этом функция k, входящая в (32), определяется известной формулой [4]:

$$k = k\left(Y_t, \vartheta, \lambda, \Theta, t\right) = \Big\{ \mathsf{M}^{p^*}_{\Delta^x} \Big[\varphi_1\left(Y_t, X, \Theta, t\right)^{\mathrm{T}} + \\ + i\lambda^{\mathrm{T}} \left(\psi \nu \psi_1\right)^{\mathrm{T}} \left(Y_t, X, \Theta, t\right) \Big] e^{-\lambda^{\mathrm{T}}\left(X - \hat{X}_t\right)} - \\ - f^{(1)T} e^{-\lambda^{\mathrm{T}} \hat{X}_t} g_t(\lambda) \Big\} \left(\psi_1 \nu_0 \psi_1^{\mathrm{T}}\right)^{-1} (Y_t, \Theta, t)\,. \quad (33)$$

Уравнения (23), (24) и (30) определяют приближенно все моменты, от которых зависит аппроксимирующая апостериорную плотность $p_t(x, \Theta; \vartheta)$ функция $p^*(x, \Theta; \vartheta)$. В качестве $p^*(x, \Theta; \vartheta)$ обычно берут отрезок ортогонального разложения (15) плотности $p_t(x, \Theta; \vartheta)$, в частности разложения по полиномам Эрмита, или отрезок ряда Эджуорта. В последнем случае можно рассчитывать на более точную аппроксимацию плотности $p_t(x, \Theta; \vartheta)$ при данном наивысшем порядке учитываемых моментов.

Таким образом, получен следующий результат.

Теорема 4.1. *Пусть уравнения нелинейной гауссовской дифференциальной СтС* (1), (2) *при* $\psi'' = \psi_1'' = 0$ *допускают применение МЦМ. Тогда алгоритм аналитического синтеза СОФ согласно МЦМ определяется уравнениями* (23)–(33).

Аналогично получаются алгоритмы аналитического синтеза СОФ и МСОФ для уравнений (6), (7). Получаемые таким образом уравнения СОФ и МСОФ значительно более громоздкие, чем уравнения в теоремах 3.2–3.4, и в статье не приводятся.

5 Чувствительность субоптимальных фильтров по методам моментов

Для оценки чувствительности СОФ и МСОФ на основе МНМ используются алгоритмы [5–7] на базе уравнений для функций чувствительности $\nabla^{\Theta}\alpha_r$, $\nabla^{\Theta} m_r$ и $\nabla^{\Theta}\mu_r$. Последние получаются путем дифференцирования по Θ нелинейных функций A^{α}_r, A^{m}_r и A^{μ}_r, входящих в уравнения теорем 3.1–3.4. Уравнения теоремы 4.1 применяются для получения функций чувствительности $\nabla^{\Theta} X_t$, $\nabla^{\Theta} R_t$ и $\nabla^{\Theta}\mu_t$ путем дифференцирования нелинейных функций $A^{\hat{X}_t}$, A^{R_t} и A^{μ}_r.

6 Тестовые примеры

6.1. В задаче

$$\dot{X}_t = -X_t^3 + X_t V_1\,; \quad \dot{Y}_t = Z_t = X_t + V_2 \quad (34)$$

уравнения СОФ имеют вид (при $\alpha_1 = \hat{X}_t$ и $\alpha_2 = \hat{X}_t^2 + R_t$):

$$\dot{\hat{X}}_t = -\alpha_3 + \left(\alpha_2 - \hat{X}_t^2\right)\left(Z_t - \hat{X}_t\right)\,; \quad (35)$$

$$\dot{\alpha}_2 = -2\alpha_4 + \nu_1\alpha_2 + \nu_2^{-1}\left(\alpha_3 - \hat{X}_t\alpha_2\right)\left(Z_t - \hat{X}_t\right)\,; \quad (36)$$

$$\dot\alpha_3 = 72\hat X_t^5 - 180\hat X_t^3\alpha_2 + 3\left(\nu_1 + 20\hat X_t^2\right)\alpha_3 - 15\hat X_t\alpha_4 + 90\hat X_t\alpha_2^2 - \\ - 30\alpha_2\alpha_3 + \nu_2^{-1}(\alpha_4 - \hat X_t\alpha_3)(Z_t - \hat X_t)\,; \qquad (37)$$

$$\dot\alpha_4 = 256\hat X_t^6 - 480\hat X_t^4\alpha_2 + 160\hat X_t^3\alpha_3 + 6\nu_1\alpha_4 + 120\alpha_2^3 - 60\alpha_2\alpha_4 + \nu_2^{-1}\times \\ \times\left(4\hat X_t\alpha_4 - 20\hat X_t^2\alpha_3 + 60\hat X_t^3\alpha_2 - 30\hat X_t\alpha_2^2 + 10\alpha_2\alpha_3 - 24\hat X_t^5\right)\left(Z_t - \hat X_t\right)\,. \qquad (38)$$

При аппроксимации апостериорной плотности отрезком ряда Эджуорта с учетом моментов до четвертого порядка в правой части последнего уравнения добавится слагаемое $10\mu_3^2 = 10(\alpha_3 - 3\hat X_t\alpha_2 + 2\hat X_t^3)^2$.

Примем за инструментальный параметр $\Theta_1\nu_1$ интенсивность белого шума V_1 в первом уравнении (34). Этот параметр входит в уравнения (35)–(38) линейно, в то время как интенсивность ν_2 белого шума во втором уравнении входит нелинейно, $\Theta_2 = \nu_2^{-1}$. Уравнения (37) и (38) позволяют также оценить точность и чувствительность алгоритма в зависимости от числа начальных моментов третьего и четвертого порядка.

6.2. В задаче (34) при аппроксимации $p_t(x,\Theta)$ отрезком разложения по полиномам Эрмита с учетом моментов до четвертого порядка уравнения СОФ имеют вид:

$$\dot{\hat X}_t = -\hat X_t\left(\hat X_t^2 + 3R_t\right) - \mu_3 + \nu_2^{-1}R_t\left(Z_t - \hat X_t\right)\,; \qquad (39)$$

$$\dot R_t = \left(\nu_1 - 6\hat X_t^2\right)R_t - \nu_2^{-1}R_t^2 - 6\hat X_t\mu_3 + \nu_2^{-1}\mu_3\left(Z_t - \hat X_t\right) - 2\mu_4\,; \qquad (40)$$

$$\dot\mu_3 = 9\hat X_tR_t^2 - 9(\hat X_t^2 + 3R_t)\mu_3 - 9\hat X_t\mu_4 + 3\nu_1(2\hat X_tR_t + \mu_3) - \\ - \frac{3}{2}\,\nu_2^{-1}R_t\mu_3 + \nu_2^{-1}\left(\mu_4 - 3R_t^2\right)\left(Z_t - \hat X_t\right)\,, \qquad (41)$$

$$\dot\mu_4 = 120R_t^3 - 108\hat X_tR_t\mu_3 - \\ - 12\left(\hat X_t^2 + 5R_t\right)\mu_4 + 4\mu_3^2 + 6\nu_1\left(\hat X_t^2R_t + 2\hat X_t\mu_3 + \mu_4\right) - \\ - 2\nu_2^{-1}R_t\left(\mu_4 - 3R_t^2\right) + 6\nu_2^{-1}R_t\mu_3\left(Z_t - \hat X_t\right)\,. \qquad (42)$$

При аппроксимации $p_t(x,\Theta)$ отрезком ряда Эджуорта с учетом моментов до четвертого порядка в правой части последнего уравнения добавится слагаемое $10\mu_3^2$.

Уравнения (39)–(42) позволяют сделать следующие выводы. Во-первых, они позволяют оценить точность и чувствительность алгоритма в зависимости от числа центральных моментов третьего и четвертого порядка, во-вторых, обнаружить линейную зависимость правых частей уравнений от $\Theta_1 = \nu_1$ и нелинейную зависимость $\Theta_2 = \nu_2^{-1}$ от интенсивности ν_2.

6.3. Полученные алгоритмы и тестовые примеры использованы в задачах оценки надежности и безопасности сложных авиационных систем [8].

7 Заключение

Разработана теория аналитического синтеза по критерию минимума средней квадратической ошибки на основе МНМ и МЦМ СОФ и МСОФ для нелинейных дифференциальных СтС, в том числе на гладких многообразиях. Уравнение состояния СтС содержит гауссовские и пуассоновские шумы, а уравнение наблюдения — только гауссовские шумы. Субоптимальные фильтры синтезируются на основе нормированной апостериорной плотности, а МСОФ — ненормированной апостериорной плотности. Алгоритмы на основе МНМ, как правило, проще, чем на основе МЦМ.

Комплекс алгоритмов позволяет оценивать влияние на точность и чувствительность инструментальных параметров, а также изучать зависимости от порядка учитываемых вероятностных моментов в разложении апостериорной плотности. Алгоритмы положены в основу модуля инструментальной программной системы StS-Filter 2018.

Разработаны два тестовых примера из области фильтрации угловых процессов в нелинейной с параметрическим шумом информационно-измерительной системе.

Результаты допускают обобщения на случай дискретных и непрерывно-дискретных МСтС. Большой практический интерес представляет теория аналитического синтеза СОФ и МСОФ в случае МСтС (1) и (2) при пуассоновских шумах в наблюдениях.

Литература

1. *Синицын И. Н.* Методы моментов в задачах аналитического моделирвоания распределений в нелинейных стохастических системах на многообразиях // Системы и средства информатики, 2015. Т. 25. № 3. С. 23–43.
2. *Пугачёв В. С., Синицын И. Н.* Стохастические дифференциальные системы. Анализ и фильтрация. — М.: Наука, 1990. 632 с.
3. *Пугачёв В. С., Синицын И. Н.* Теория стохастических систем. — М.: Логос, 2000; 2004. 1000 с.
4. *Синицын И. Н.* Фильтры Калмана и Пугачёва. — 2-е изд. – М.: Логос, 2007. 776 с.
5. *Синицын И. Н.* Ортогональные субоптимальные фильтры для нелинейных систем на многообразиях // Информатика и её применения, 2016. Т. 10. Вып. 1. С. 34–44.
6. *Синицын И. Н.* Нормальные и ортогональные субоптимальные фильтры для нелинейных стохастических систем на многообразиях // Системы и средства информатики, 2016. Т. 26. № 1. С. 199–226.
7. *Синицын И. Н., Синицын В. И., Корепанов Э. Р.* Модифицированные эллипсоидальные условно-оптимальные фильтры для нелинейных стохастических систем на многообразиях // Информатика и её применения, 2017. Т. 11. Вып. 2. С. 101–111.
8. ГОСТ 23743–88. Изделия авиационной техники. Номенклатура показателей безопасности полета, надежности, контролепригодности, эксплуатационной и ремонтной технологичности.

Поступила в редакцию 21.09.17

ANALYTICAL SYNTHESIS OF SUBOPTIMAL FILTERS BY MOMENTS METHODS

I. N. Sinitsyn, V. I. Sinitsyn, and E. R. Korepanov

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: On the basis of initial and central moments, the theory of analytical synthesis of suboptimal and modificated filters (SOF and MSOF) for differential (on manifolds) nonlinear stochastic systems (StS) is developed. The authors suppose that: (*i*) state equation includes Gaussian and Poisson noises; and (*ii*) observation equation contains Gaussian noises only. For SOF synthesis, the authors use normalized densities and for MOF, unnormalized densities. Questions of instrumental accuracy and sensitivity are discussed. The algorithms are the basis of the software tool StS-Filter 2018. Two test examples for angular information-measurement system are given. Some generalizations are mentioned.

Keywords: *a priori* distribution (density, characteristic function); method of initial moments; method of central moments; modificated method of initial moments; modificated method of central moments; normalized distribution; unnormalized distribution; stochastic system; suboptimal filter; angular information-measurement system; Gaussian noise; Poisson noise

DOI: 10.14357/08696527180101

References

1. Sinitsyn, I. N. 2015. Metody momentov v zadachakh analiticheskogo modelirovaniya raspredeleniy v nelineynykh stokhasticheskikh sistemakh na mnogoobraziyakh [Moments methods in analytical modeling problems in nonlinear stochastic systems]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 25(3):23–43.
2. Pugachev, V. S., and I. N. Sinitsyn. 1987. *Stochastic differential systems. Analysis and filtering*. Chichester, New York, NY: Jonh Wiley. 549 p.
3. Pugachev, V. S., and I. N. Sinitsyn. 2001. *Stochastic systems. Theory and applications*. Singapore: World Scientific. 908 p.
4. Sinitsyn, I. N. 2007. *Fil'try Kalmana i Pugacheva* [Kalman and Pugachev filters]. 2nd ed. Moscow: Logos. 776 p.
5. Sinitsyn, I. N. 2016. Ortogonal'nye suboptimal'nye fil'try dlya nelineynykh sistem na mnogoobraziyakh [Orthogonal suboptimal filters for nonlinear stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 10(1):34–44.
6. Sinitsyn, I. N. 2016. Normal'nye i ortogonal'nye suboptimal'nye fil'try dlya nelineynykh stokhasticheskikh sistem na mnogoobraziyakh [Normal and orthogonal suboptimal filters for nonlinear stochastic systems on manifolds]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(1):199–226.

7. Sinitsyn, I. N., V. I. Sinitsyn, and E. R. Korepanov. 2017. Modifitsirovannye ellipsoidal'nye uslovno-optimal'nye fil'try dlya nelineynykh stokhasticheskikh sistem na mnogoobraziyakh [Modificated ellipsoidal conditionally optimal filters for nonlinear stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 11(2):101–111.
8. GOST 23743–88. Izdeliya aviatsionnoy tekhniki. Nomenklatura pokazateley bezopasnosti poleta, nadezhnosti, kontroleprigodnosti, ekspluatatsionnoy i remontnoy tekhnologichnosti [Nomenclature of security and reliability indexers].

Received September 21, 2017

Contributors

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, principal scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sinitsin@dol.ru

Sinitsyn Vladimir I. (b. 1968) — Doctor of Science in physics and mathematics, associate professor, Head of Department, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; VSinitsyn@ipiran.ru

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Department, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Ekorepanov@ipiran.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 20–34

К ВОПРОСУ О РАСЧЕТЕ НАДЕЖНОСТИ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ: УЧЕТ ХАРАКТЕРИСТИК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

***А. В. Борисов*[1], *А. В. Босов*[2], *А. В. Иванов*[3], *Э. Р. Корепанов*[4]**

Аннотация: Оценка надежности вновь создаваемых и существующих информационно-телекоммуникационных систем (ИТКС), традиционно рутинная и формальная, рассматривается как неотъемлемая часть процесса проектирования. Для расчета надежности системы предлагается целостная методика, в рамках которой, во-первых, выделен набор количественных характеристик надежности, отвечающих качественным показателям технического задания, определяемым на разных стадиях разработки системы. Для аппаратной составляющей — это стандартный коэффициент готовности. Для программного обеспечения (ПО) — это набор вероятностных характеристик, учитывающих специфику разработки и технической поддержки программ. Во-вторых, для выбранных характеристик приводятся соотношения для их расчета на основе базовых методов математической статистики. В-третьих, даны рекомендации по сбору статистики функционирования на разных стадиях жизненного цикла системы. Предложенная методика использовалась авторами в ряде выполненных проектов и доказала свою эффективность.

Ключевые слова: информационно-телекоммуникационная система; проектирование; надежность; вероятностные характеристики надежности

DOI: 10.14357/08696527180102

1 Введение

Информационные технологии (ИТ) давно стали неотъемлемой частью деловых процессов во всех областях человеческой деятельности. Если ИТ применяются не точечно, на отдельных этапах деятельности, а на них возлагается

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ABorisov@ipiran.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, AVBosov@ipiran.ru

[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, AIvanov@ipiran.ru

[4]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, EKorepanov@ipiran.ru

существенный объем функциональности предприятия, то создаются целые системы, называемые информационно-телекоммуникационными системами. Создание ИТКС — это тоже производственный процесс, и к настоящему моменту он достаточно хорошо регламентирован (см., например, ГОСТы серии 34 [1], стандарты ИСО/МЭК 12207 [2], методику Oracle Unified Method [3]). Неотъемлемой частью этого процесса является анализ и расчет надежности создаваемой системы. В последние годы рутинное отношение к этому этапу разработки определенно претерпевает изменение. Так, формальные выкладки с учетом данных, представленных производителями комплектующих и элементов, кардинально влияют на выбор архитектурных решений. Приходится учитывать и факты завышения производителями показателей надежности элементной базы, уточнение которых может вносить существенные коррективы на всех этапах жизни изделия — проектирования, создания и эксплуатации. В рамках этой же тенденции рассматриваются ранее не учитывавшиеся элементы, из которых наиболее сложным является ПО.

Прежде всего ИТКС рассматривается как совокупность оборудования: вычислительного, телекоммуникационного, обеспечивающего. Вопросы надежности для таких элементов ИТКС хорошо проработаны (см., например, ГОСТ 24.701-86 [4]). Программное обеспечение естественным образом также признается составной частью ИТКС, и указанный стандарт регламентирует проведение оценки надежности ИТКС на этапах технического и рабочего проектирования с учетом свойств как технических средств, так и ПО. Известно множество методов, которые могут быть применены для получения оценки надежности [5]. Динамические методы требуют наличия завершенного ПО и используют результаты его исполнения: число отказов, время безотказной работы, результаты тестов. Статистические методы используют в качестве информации спецификации ПО, исходные коды или данные их синтаксического анализа. Архитектурные методы расширяют динамические и статистические методы путем использования знаний об архитектуре ПО: составе и взаимном влиянии компонентов. Эмпирические методы используют метрики процесса проектирования: опыт разработчиков, наличие сертификатов, применяемую методологию проектирования, опыт ранее выполненных проектов.

Каждая из обозначенных категорий методов имеет свои ограничения. Динамические методы чрезвычайно затратны, так как требуют большого объема результатов запуска программ с разными входными данными. В статистических методах на основе метрик сложности существует проблема определения весовых коэффициентов для получения оценки надежности, а статистические методы на основе анализа исходного кода сложны в реализации. В архитектурных методах весьма трудозатратной является декомпозиция системы на компоненты и определение их взаимного влияния на надежность. Наконец, в эмпирических методах сложность представляет получение метрик процесса проектирования.

Продолжающийся рост объемов и сложности современного ПО побуждает искать новые методы оценки надежности. В данной работе предлагается опро-

бованный в ряде опытно-конструкторских проектов метод оценки надежности ИТКС, включающий в числе прочего и оценку надежности ПО. В предположении наличия ошибок в ПО ИТКС и случайного характера их выявления в процессе эксплуатации представлена реализация схемы моделирования работы ИТКС, сбора и накопления статистической информации и обработки для определения вероятностных характеристик ПО, в частности оценки надежности системы в целом. Математические инструменты, используемые в расчетах, ограничиваются традиционными методами теории вероятностей и математической статистики.

2 Методика расчета показателей надежности комплекса технических средств

2.1 Показатели надежности комплекса технических средств и правила их расчета

Расчеты надежности комплекса технических средств (КТС) проводятся согласно ГОСТ 27.002-89 — «Надежность в технике. Основные понятия. Термины и определения».

В общем случае отказом технического средства называется событие, заключающееся в нарушении его работоспособности. Самоустраняющиеся отказы или однократные отказы, требующие незначительного вмешательства обслуживающего персонала, называются сбоями.

При расчете показателей надежности КТС принимаются следующие допущения.

1. Закон распределения времени безотказной работы технических средств на протяжении срока службы — экспоненциальный.
2. Отказы технических средств являются событиями случайными и независимыми.
3. Одновременно возникает не более одного отказа: таким образом, общий поток отказов технических средств является простейшим.
4. Информационно-телекоммуникационная система эксплуатируется в условиях, предусмотренных документацией на нее, и не рассматриваются отказы, вызванные воздействием внешних факторов, не предусмотренных условиями ее применения.
5. Надежность элементов после восстановления равна их надежности в начальный момент.
6. Элементы, находящиеся в ненагруженном резерве, считаются находящимися в состоянии хранения.
7. Служба эксплуатации ИТКС имеет достаточную квалификацию и способна своевременно и безошибочно приступить к устранению отказа.

8. Служба эксплуатации имеет необходимый комплект запасных частей и принадлежностей (ЗИП), предусмотренный рабочей конструкторской и эксплуатационной документацией, и может воспользоваться им немедленно после наступления отказа.
9. Не рассматриваются отказы, вызванные нарушением службой эксплуатации указаний эксплуатационной документации.

Интегральным показателем надежности, количественно характеризующим безотказность и ремонтопригодность КТС, является коэффициент готовности ($K_{\text{г}}$). Аналогичный коэффициент R далее определен в качестве одной из характеристик ПО, так что комплексный показатель надежности ИТКС можно определить как их произведение. Расчет $K_{\text{г}}$ выполняется исходя из следующих положений.

1. Коэффициент готовности нерезервируемых элементов технических средств ($K_{\text{г}i}$):
$$K_{\text{г}i} = (1 + \lambda_i T_{\text{в}i})^{-1} \ , \tag{1}$$
где $\lambda_i = 1/T_{\text{о}i}$ — интенсивность отказов i-го элемента, 1/ч; $T_{\text{о}i}$ — средняя наработка на отказ i-го элемента, ч; $T_{\text{в}i}$ — среднее время восстановления i-го элемента, ч.
Формула (1) может быть представлена в виде:
$$K_{\text{г}i} = \frac{T_{\text{о}i}}{T_{\text{о}i} + T_{\text{в}i}} \, . \tag{2}$$
2. Коэффициент готовности нерезервируемой группы элементов ($K_{\text{г}}$), отказ любого из которых приводит к отказу ИТКС в целом:
$$K_{\text{г}} = \prod_{i=1}^{m} K_{\text{г}i} \, , \tag{3}$$
где $K_{\text{г}i}$ — коэффициент готовности элемента; m — число элементов в группе.
3. Коэффициент готовности резервируемой группы ($K_{\text{грг}}$) из k основных и n резервных элементов ($N = k + n$):
$$K_{\text{грг}} = 1 - \frac{N!}{(n+1)![N-(n+1)]!} \, (\lambda_i T_{\text{в}i})^{n+1} \ . \tag{4}$$

2.2 Критерии отказа комплекса технических средств

Под отказом КТС предлагается понимать нарушение его работоспособности, приводящее к невозможности выполнения ИТКС основных требований назначения, определенных техническим заданием на создание ИТКС. Примерами

могут служить создание документа, загрузка поступающих данных, отображение результатов анализа или мониторинга и т. д. Кроме того, к отказам следует относить факты потери обрабатываемых данных.

С точки зрения состава КТС отказ ИТКС порождается отказом хотя бы одного из элементов:

- комплекса серверного оборудования каждой функциональной подсистемы ИТКС;
- комплекса средств коммуникации между комплексами серверного оборудования (например, аппаратура внутренней коммуникации);
- комплекса вспомогательного оборудования распределения электропитания между компонентами указанных комплексов серверного оборудования.

При необходимости к отказам можно относить и сбои в работе пользовательской аппаратуры, но такие варианты представляются с позиции надежности не очень значимыми. Как правило, автоматизированное рабочее место и периферия не являются критическими элементами ИТКС и их сбои/восстановления составляют часть штатного процесса функционирования.

2.3 Методика определения характеристик надежности комплекса технических средств на этапах разработки и эксплуатации

Оценка показателей надежности для КТС может быть реализована на всех этапах создания ИТКС с использованием расчетных формул (1)–(4). При этом показатели интенсивностей отказов λ_i на начальных этапах получаются на основании данных, представленных производителями, а на последующих, вплоть до этапа эксплуатации, — уточняются при необходимости на основании собираемой статистики отказов элементов КТС.

На этапах технического проектирования (ТП), разработки рабочей конструкторской документации (РКД) и изготовления образца (ИО) для расчетов следует использовать данные по средней наработке на отказ (MTBF, mean time between failures), предоставляемые производителями оборудования, а в случае отсутствия таких данных — использовать имеющиеся в открытом доступе данные по аналогичному оборудованию. В качестве исходных данных по среднему времени восстановления (MTTR, mean time to repair) следует руководствоваться оценками времени автоматического восстановления работоспособности технического средства или оценками времени его восстановления подготовленным специалистом при наличии необходимого комплекта ЗИП.

Во время предварительных и государственных испытаний, опытной и штатной эксплуатации необходимо осуществлять сбор статистики об отказах технических средств ИТКС с последующим уточнением расчетных показателей надежности КТС. При этом одинаковые технические средства, входящие в состав ИТКС, следует считать равнонадежными и оценивать показатели их надежности по суммарной статистике.

3 Методика расчета показателей надежности программного обеспечения

3.1 Определение отказа программного обеспечения

Согласно [6], под отказом ПО понимается зарегистрированный факт неспособности выполнить требования назначения ИТКС, заданные в техническом задании на ее разработку, причиной которой являются ошибки в ПО.

С точки зрения негативного результата отказы ПО подразделяются на следующие:

- прекращение функционирования программы, искажения нормального хода ее выполнения, зацикливание на время, превышающее заданный порог;
- прекращение функционирования программы, искажения нормального хода ее выполнения, зацикливание на время, не превышающее заданный порог, но с потерей всех или части обрабатываемых данных;
- прекращение функционирования программы, искажения нормального хода ее выполнения, зацикливание, потребовавшие перезагрузки служб и/или ЭВМ, на которой функционирует ПО.

По имеющемуся опыту разработки пороговым значением времени предварительно следует выбрать 4 ч, а в дальнейшем скорректировать его по результатам опытной эксплуатации. Указанная верхняя граница, как правило, существенно уменьшается. Конкретное значение зависит от многих факторов, в том числе уровня технической подготовки пользователей, характера решаемых задач, принятой в организации исполнительской дисциплины. В этом смысле опытная эксплуатация позволяет довольно точно ответить на вопрос, когда пользователь перестает ждать от программы результата и констатирует ее «зависание» (при этом вполне возможно, что делается это ошибочно, преждевременно).

С точки зрения степени сложности и скорости восстановления функционирования ПО отказы подразделяются на следующие группы:

(1) отказы, устраняемые конечными пользователями ПО (путем перезагрузки служб и/или ЭВМ и пр.);

(2) отказы, устраняемые службой эксплуатации путем дополнительных настроек параметров ПО;

(3) отказы, устраняемые службой эксплуатации путем восстановления ПО отдельных частей или ИТКС в целом;

(4) отказы, устраняемые разработчиками ПО путем модификации программного кода.

3.2 Характеристики надежности программного обеспечения

Программное обеспечение не подвержено старению, его функционирование на каждом фиксированном наборе входных данных (НВД) носит неслучайный

характер. Программное обеспечение предлагается рассматривать как «серый» ящик, как автоматическую систему с частично неизвестными свойствами, процессы выполнения функций в котором порождаются поступающей информацией: входными данными, командами пользователей и т. п.

Далее каждая отдельная совокупность данных, поступающая на обработку ПО и инициализирующая выполнение одной или нескольких задач, называется набором входных данных.

В процессе разработки, тестирования, опытной эксплуатации и различного рода испытаний функционирование ПО проверяется на репрезентативном (с точки зрения всей совокупности функциональных возможностей ИТКС) множестве различных НВД. Результатом этих проверок является регистрация сбоев ПО и устранение порождающих их ошибок ПО. При этом упомянутая коррекция ошибок не гарантирует, что скорректированная версия кода не содержит ошибок, оставшихся или внесенных при корректировке. В то же время проверить функционирование ПО на всех возможных НВД физически невозможно.

Сведения о возможных параметрах надежности ПО в общем случае представлены в [7–11], однако необходимый набор параметров нужно выбрать, руководствуясь их максимальной близостью к характеристикам надежности всей ИТКС, определенным в техническом задании на ее создание. В качестве таких характеристик предлагается использовать следующие параметры, формулы для вычисления которых базируются на сведениях из теории вероятностей [12]:

(1) $n(\alpha, N, p)$ — верхняя ожидаемая граница числа ошибочных строк в коде длиной N строк с вероятностью p ошибки в одной строке и уровнем доверительной вероятности α.

Пусть v — случайное число ошибочных строк в коде длиной N строк, тогда

$$n(\alpha, N, p) = \min\{n \in \mathbb{N} : P\{v \le n\} \ge \alpha\} \,,$$

где через $P\{A\}$ обозначена вероятность случайного события A.

Указанная характеристика вычисляется по формуле:

$$n(\alpha, N, p) = \left[Np + \sqrt{Np(1-p)}\,\Phi^{-1}(\alpha)\right] + 1\,, \tag{5}$$

где $\Phi^{-1}(\alpha)$ — квантиль стандартного нормального распределения уровня α;

(2) T — среднее время наработки ПО на отказ.

Пусть τ — время безотказного функционирования ПО, т. е. временной интервал между моментом запуска ПО и моментом регистрации отказа ПО. Данная величина является случайной, и

$$T = \mathsf{M}[\tau]\,,$$

где через $\mathsf{M}[\tau]$ обозначено математическое ожидание случайной величины τ; при этом предполагается, что распределение τ обеспечивает существование требуемого момента;

(3) S — среднее время восстановления ПО.

Пусть σ — время восстановления функционирования ПО, т. е. временной интервал между моментом начала восстановления функционирования ПО и моментом регистрации восстановления ПО. Данная величина является случайной, и

$$S = \mathsf{M}[\sigma]\,;$$

(4) R — коэффициент готовности ПО.

Данный коэффициент равен вероятности того, что вновь поступивший в ИТКС НВД будет обработан без отказа ПО. Пусть I — случайная величина, индикатор штатной обработки НВД:

$$I = \begin{cases} 1\,, & \text{если отказа ПО нет}\,; \\ 0\,, & \text{если отказ ПО есть}\,, \end{cases}$$

тогда

$$R = \mathsf{M}[I]\,;$$

(5) $K(q, \varepsilon, \beta)$ — минимальная длина серии НВД для оценки коэффициента готовности R с точностью ε при условии, что $R > q$, и доверительной вероятностью β.

Пусть N — длина серии НВД; m — число НВД, обработанных без отказа ПО; ε — параметр точности оценки R; q — нижняя граница коэффициента готовности; p — истинный неизвестный коэффициент готовности. Тогда

$$K(q, \varepsilon, \beta) = \min\left\{ N \in \mathbb{N} : P\left\{ \left| \frac{m}{n} - p \right| < \varepsilon \right\} \geq \beta | p \geq q \right\}.$$

Параметр $K(q, \varepsilon, \beta)$ вычисляется по формуле:

$$K(q, \varepsilon, \beta) = \left[\frac{q(1-q)}{\varepsilon^2} \left(\Phi^{-1}\left(\frac{1-\beta}{2} \right) \right)^2 \right] + 1\,.$$

3.3 Методика определения характеристик надежности программного обеспечения на этапах разработки и эксплуатации

Представленные характеристики надежности вычисляются и уточняются на разных этапах создания и эксплуатации ИТКС и носят как предварительный теоретический, так и изменяющийся статистический характер.

На этапах ТП и РКД вычисление характеристик $n(\alpha, N, p)$, T, S и R невозможно, так как на этих этапах отсутствует информация о количественных характеристиках разрабатываемого ПО. Для вычисления характеристики

$n(\alpha, N, p)$ первоначально отсутствует информация об объеме N разрабатываемого ПО. Характеристики T, S и R вычисляются на основе статистических данных о функционировании разработанного ПО на последующих этапах. Параметр $K(q, \varepsilon, \beta)$ на этих этапах также не вычисляется, так как на данных этапах он бесполезен.

На этапе ИО вычисляются параметры $n(\alpha, N, p)$ и $K(q, \varepsilon, \beta)$. Длина кода N к этому времени уже известна, уровень доверительной вероятности α определяется общими требованиями надежности ИТКС, а параметр p оценивается по опыту предыдущих разработок (своих или других разработчиков).

Например, пусть необходимо оценить верхнюю ожидаемую границу числа ошибочных строк в коде длиной 500 000 *строк с уровнем доверительной вероятности* 0,9999. *Согласно* [13], *объем кода ОС Windows 7 составляет порядка* 50 000 000 *строк, в то время как, согласно* [14], *за все время поддержки ОС Windows 7 выпущено* 448 *обновлений. Таким образом, параметр p оценивается как* 448/50 000 000 $\approx$ 0,00001. *Тогда по формуле* (5)

$$n(0{,}00001; 500\,000; 0{,}00001) = \left[5 + \sqrt{5 \cdot 0{,}99995} \cdot 3{,}72\right] + 1 = 14.$$

На этапе изготовления для оценивания коэффициента готовности ИТКС R должна быть вычислена характеристика $K(q, \varepsilon, \beta)$. Она будет использоваться для определения размера случайной выборки НВД, используемой при проведении испытаний.

Например, если $q = 0{,}99$, $\varepsilon = 0{,}001$ *и* $\beta = 0{,}999$, *то* $K(q, \varepsilon, \beta) = 94\,541$.

На предварительных испытаниях ИТКС необходимо провести оценку параметров T, S и R. Пусть на испытаниях обработана серия НВД длины N, при этом было зафиксировано k сбоев ПО, а также временны́е параметры функционирования/восстановления:

- τ_j, $j = 1, \ldots, k$, — интервалы времени безотказного функционирования ПО до j-го отказа;
- τ_{k+1} — время безотказного функционирования ПО после k-го (последнего) восстановления до окончания проведения испытаний;
- σ_j, $j = 1, \ldots, k$, — время восстановления ПО после j-го отказа.

Тогда начальные оценки параметров T, S и R вычисляются по формулам:

$$\hat{R}_0 = \frac{k}{N}; \qquad \hat{T}_0 = \frac{1}{k+1} \sum_{j=1}^{k+1} \tau_j; \qquad \hat{S}_0 = \frac{1}{k} \sum_{j=1}^{k} \sigma_j. \tag{6}$$

Данные предварительные оценки являются опорными для проведения опытной эксплуатации, государственных испытаний и дальнейшей штатной эксплуатации.

После предварительных испытаний и во время опытной эксплуатации обычно происходит модификация ПО с целью исправления ошибок, поэтому следует ожидать изменения надежностных характеристик T, S и R в сторону их улучшения. В то же время необходимо использовать опорные оценки $\hat{R}_0$, $\hat{T}_0$ и $\hat{S}_0$, представленные выше, поэтому для обновления характеристик рекомендуется использовать следующую процедуру экспоненциального сглаживания.

Пусть между r-м и $(r+1)$-м зарегистрированными отказами ПО за время безотказного функционирования σ_{r+1} было обработано n_{r+1} запросов и для восстановления ПО потребовался временной отрезок, равный σ_{r+1}. Тогда обновленные оценки коэффициента готовности и среднего времени между отказами вычисляются по рекуррентным формулам [15]:

$$\left.\begin{aligned} \hat{R}_{r+1} &= \gamma\hat{R}_r + (1-\gamma)\frac{1}{n_{r+1}}\,; \\ \hat{T}_{r+1} &= \gamma\hat{T}_r + (1-\gamma)\tau_{t+1}\,; \\ \hat{S}_{r+1} &= \gamma\hat{S}_r + (1-\gamma)\sigma_{t+1}\,, \end{aligned}\right\} \qquad (7)$$

где γ $(0 < \gamma < 1)$ — некоторый фиксированный параметр сглаживания. Формулы (7) позволяют отслеживать изменения показателей надежности в реальном масштабе времени по мере получения информации об отказах ПО.

Альтернативным вариантом вычисления оценок T, S и R является использование формул (6) для одновременной обработки всего объема данных по отказам ПО, зарегистрированным за время опытной эксплуатации.

На государственных испытаниях параметры T, S и R оцениваются заново по результатам обработки новой случайной выборки $K(q, \varepsilon, \beta)$ НВД по формулам (6).

Как и в процессе опытной эксплуатации, в процессе штатной эксплуатации оценки параметров T, S и R корректируются с помощью формул экспоненциального сглаживания (7), используя в качестве начальных оценки, полученные по результатам государственных испытаний. Альтернативным вариантом вычисления оценок T, S и R является использование формул (6) для одновременной обработки всего объема данных по отказам ПО, зарегистрированным за какой-либо фиксированный период (месяц, квартал, год).

В результате коррекции ПО и исправления ошибок, ведущих к сбоям, оценки параметров R и T должны расти, а S — уменьшаться. Однако этого может не происходить, если состав и интенсивность поступления реальных данных сильно отличаются от условно-реальных данных, используемых в процессе проведения испытаний и опытной эксплуатации.

3.4 Условия и процедура фиксации отказов программного обеспечения

Все отказы ИТКС, имеющие место в процессе испытаний, опытной и штатной эксплуатации, должны регистрироваться в специальном «Журнале регистрации отказов». Предлагаемая форма журнала приведена на рисунке.

Журнал регистрации отказов Системы

Начат _____

Закончен _____

Число НВД, обработанных к моменту начала журнала, M_0 _____

Дата/время последнего восстановления Системы после аппаратного отказа _____

Дата/время последнего восстановления Системы после отказа ПО, $D_{0,2}$ _____

1	2	3	4	5	6	7	8	9	10	11
№	Дата/время регистрации отказа	Подсистема, комплекс, компонента	Число НВД, обработанных к моменту регистрации отказа	Описание отказа, сведения о повторяемости, критичности, сохранении входных данных, вызвавших отказ	ФИО, подразделение зарегистрировавшего отказ	Действия по восстановлению	Время устранения отказа (общее)	Время отклонения отказа (чистое)	Дата/время восстановления функционирования Системы	Примечание
…	…	…	…	…	…	…	…	…	…	…
L	$D_{L,1}$		M_L					t_L	$D_{L,2}$	
N+1	$D_{L+1,1}$		$M_{L+1,1}$					t_{L+1}	$D_{L+1,2}$	
…	…	…	…	…	…	…	…	…	…	…

Форма «Журнала регистрации отказов»

Для вычисления показателей надежности ПО должна использоваться только статистическая информация относительно отказов ПО, т. е. к рассмотрению принимаются отказы, соответствующие требованиям разд. 2.1.

Входные данные, на которых произошел отказ ПО, должны сохраняться, равно как и системные журналы ПО.

Отказы ПО должны воспроизводиться при повторном выполнении условий. При этом повторное воспроизведение этих условий и появление отказа ПО в журнале повторно не регистрируются.

Тестовый НВД, подготовленный для проведения предварительных и государственных испытаний, должен представлять собой случайную репрезентативную выборку, соответствующую реальным данным. Данные, обрабатываемые в процессе опытной эксплуатации, должны обладать аналогичными свойствами.

4 Заключение

Анализ выполненных в последние годы проектов создания различных ИТКС свидетельствует об изменении отношения к работам по оценке надежности создаваемых систем, отказе от формальных, обладающих сомнительной объективностью кабинетных расчетов. Представленные в работе рассуждения полезны прежде всего потому, что составляют целостный, законченный метод. При этом у авторов уже есть опыт, подтверждающий, что этот метод способен удовлетворить самого придирчивого заказчика. Применение описанного метода позволило не только проанализировать показатели эффективности вновь создаваемых ИТКС, но и добиться повышения их производительности за счет выявления ключевых факторов влияния на комплексный показатель надежности ИТКС.

Однако у предложенного подхода есть существенная сложность — нужен значительный промежуток времени, в течение которого ИТКС уже эксплуатируется, но есть возможность собирать нужную статистику отказов ПО (понятно, что данные выше оценки объемов НВД при любых разумных точностных требованиях будут исчисляться десятками и сотнями тысяч). При этом на практике такой способ расчетов возможен только в рамках этапа разработки — опытной эксплуатации, продолжительность которого обычно невелика. Очевиден также и другой аспект ограниченности предложенной методики — слишком многое зависит от желания и возможностей конкретных пользователей имитировать реальную работу на опытном образце или каком-либо прототипе. Без такого желания оценить характеристики ИТКС и ее ПО в условиях реальных нагрузок, а уж тем более пиковых, не представляется возможным. Данная проблема представляет интерес в плане дальнейшего исследования, так как имеется возможность отказаться от деятельного участия пользователей в описанном методе за счет автоматизированных средств имитации их работы.

Литература

1. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. — М.: Стандартинформ, 2009. 5 с.
2. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. — М.: Стандартинформ, 2011. 100 с.
3. Oracle Unified Method (OUM). http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method1-069271.pdf.

4. ГОСТ 24.701-86. Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. — М.: Стандартинформ, 2009. 11 с.
5. *Моисеев М. Ю.* Формальные методы обеспечения качества ПО. http://kspt.icc.spbstu.ru/media/files/2010/course/softwarequality/lec2.pdf.
6. Dependability: Basic concepts and terminology / Ed. J. C. Laprie. — Vienna: Springer-Verlag, 1992. 266 p.
7. ГОСТ 28806-90. Межгосударственный стандарт. Качество программных средств. — М.: Изд-во стандартов, 2001. 8 с.
8. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. — М.: Изд-во стандартов, 2004. 10 с.
9. ISO/IEC 9126-1:2001. Software engineering — Product quality — Part 1: Quality model. https://www.iso.org/standard/22749.html.
10. ISO/IEC TR 9126-2:2003. Software engineering — Product quality — Part 2: External metrics. https://www.iso.org/standard/22750.html.
11. ISO/IEC TR 9126-3:2003. Software engineering — Product quality — Part 3: Internal metrics. https://www.iso.org/standard/22891.html.
12. *Ширяев А. Н.* Вероятность. — М.: Наука, 1979.
13. How many lines of code in Windows // Knowing.NET, 06.12.2005. http://www.knowing.net/index.php/2005/12/06/how-many-lines-of-code-in-windows.
14. Windows 7 x64 Update and Hotfix List. http://windows-update-checker.com/Lists/Win7x64.htm.
15. *Cadzow J. A.* Foundations of digital signal processing and data analysis. — New York, NY, USA: Macmillan, 1987. 256 p.

Поступила в редакцию 16.11.17

TO THE RELIABILITY OF AN INFORMATION-TELECOMMUNICATION SYSTEM: AN APPROACH TO RECOGNITION OF RELIABLE SOFTWARE CHARACTERISTICS

A. V. Borisov, A. V. Bosov, A. V. Ivanov, and E. R. Korepanov

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: Issues of reliability assessment of newly created and existing information and telecommunication systems (ITS), traditionally routine and formal, are considered as an inalienable part of the design process. To determine the reliability of a system, the holistic methodology is proposed. Within the framework of this methodology, firstly, specific quantitative characteristics of reliability,

which correspond to qualitative indicators determined at different stages of system development, are specified. For the hardware component, this is the availability factor, which is set by the standard indicators of trouble-free operation of the nodes. For software, this is a set of probability characteristics that take into account the specifics of different reliability indicators that characterize the quality of programs. Secondly, for determination of the chosen reliability characteristics, the relations based on the basic methods of mathematical statistics are given. Thirdly, to determine the probability characteristics, recommendations for collecting statistics at different stages of the life cycle of the system are given. The proposed methodology was used by the authors in a set of completed projects and proved to be effective.

Keywords: information and telecommunication systems; system design; reliability; probability reliability characteristics

DOI: 10.14357/08696527180102

References

1. GOST 34.601-90. 2009. Avtomatizirovannye sistemy. Stadii sozdaniya [Information technology. Set of standards for automated systems. Stages of development]. Moscow: Standardinform Publs. 5 p.
2. ISO/IEC 12207:2008. 2011. Informatsionnaya tekhnologiya. Sistemnaya i programmnaya inzheneriya. Protsessy zhiznennogo tsikla programmnykh sredstv [System and software engineering — software life cycle processes (IDT)]. Moscow: Standardinform Publs. 100 p.
3. Oracle Unified Method (OUM). Available at: http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method1-069271.pdf (accessed December 1, 2017).
4. GOST 24.701-86. 2009. Edinaya sistema standartov avtomatizirovannykh sistem upravleniya. Nadezhnost' avtomatizirovannykh sistem upravleniya. Osnovnye polozheniya [Unified system of standards of computer control systems. Dependability of computer control systems. General positions]. Moscow: Standardinform Publs. 11 p.
5. Moiseev, M. Ju. 2010. Formal'nye metody obespecheniya kachestva PO [Formal methods of software quality assurance]. Available at: http://kspt.icc.spbstu.ru/media/files/2010/course/softwarequality/lec2.pdf (accessed December 1, 2017).
6. Laprie, J. C., ed. 1992. *Dependability: Basic concepts and terminology*. Vienna: Springer-Verlag. 266 p.
7. GOST 28806-90. 2001. Mezhgosudarstvennyy standart. Kachestvo programmnykh sredstv [Software quality. Terms and definitions]. Moscow: IPK Izdatel'stvo standartov. 8 p.
8. GOST R ISO/IEC 9126-93. 2004. Informatsionnaya tekhnologiya. Otsenka programmnoy produktsii. Kharakteristiki kachestva i rukovodstva po ikh primeneniyu [Information technology. Software product evaluation. Quality characteristics and guidelines for their use]. Moscow: IPK Izdatel'stvo standartov. 10 p.
9. ISO/IEC 9126-1:2001. Software engineering — Product quality — Part 1: Quality model. Available at: https://www.iso.org/standard/22749.html (accessed December 1, 2017).

10. ISO/IEC TR 9126-2:2003. Software engineering — Product quality — Part 2: External metrics. Available at: https://www.iso.org/standard/22750.html (accessed December 1, 2017).
11. ISO/IEC TR 9126-3:2003. Software engineering — Product quality — Part 3: Internal metrics. Available at: https://www.iso.org/standard/22891.html (accessed December 1, 2017).
12. Shiryaev, A. N. 1984. *Probability*. New York, NY: Springer-Verlag. 578 p.
13. How many lines of code in Windows. December 6, 2005. *Knowing.NET*. Available at: http://www.knowing.net/index.php/2005/12/06/how-many-lines-of-code-in-windows/ (accessed December 1, 2017).
14. Windows 7 x64 Update and Hotfix List. Available at: http://windows-update-checker.com/Lists/Win7x64.htm (accessed December 1, 2017).
15. Cadzow, J. A. 1987. *Foundations of digital signal processing and data analysis*. New York, NY: Macmillan. 256 p.

Received November 16, 2017

Contributors

Borisov Andrey V. (b. 1965) — Doctor of Science in physics and mathematics, principal scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ABorisov@ipiran.ru

Bosov Alexey V. (b. 1969) — Doctor of Science in technology, principal scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44- 2 Vavilov Str., Moscow 119333, Russian Federation; AVBosov@ipiran.ru

Ivanov Alexey V. (b. 1976) — scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; AIvanov@ipiran.ru

Korepanov Eduard R. (b. 1966) — Head of Deparment, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; EKorepanov@ipiran.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 35–52

ВЕРОЯТНОСТНЫЙ ПОДХОД К РЕШЕНИЮ ОБРАТНОЙ ЗАДАЧИ МАГНИТОЭНЦЕФАЛОГРАФИИ

М. Б. Гончаренко[1], *Т. В. Захарова*[2]

Аннотация: Исследование головного мозга является одним из самых активно развивающихся направлений нейронауки, в котором объединяют свои усилия широкие группы специалистов: от психологов до математиков. Одной из важнейших задач, встающих перед исследователями при обработке данных нейрофизиологического эксперимента, является локализация активных зон коры головного мозга. Эта информация критически важна для всех методов нейровизуализации, таких как электроэнцефалография (ЭЭГ), магнитоэнцефалография (МЭГ), функциональная магнитно-резонансная томография (фМРТ) и др. В данной статье рассматриваются данные МЭГ. Магнитоэнцефалография — неинвазивный метод нейровизуализации, позволяющий регистрировать сверхслабые магнитные поля, порожденные нейронами головного мозга. Реконструкция источников по данным МЭГ представляет собой некорректно поставленную обратную задачу. В настоящей работе рассматривается байесовский вывод решения обратной задачи МЭГ. Особое внимание уделено такому преимуществу байесовского подхода, как универсальность, в его рамках получены другие популярные методы, широко используемые в исследованиях. Также рассмотрено обобщение на случай группового эксперимента и описаны возможные дальнейшие пути улучшения методов решения обратной задачи.

Ключевые слова: байесовский подход; магнитоэнцефалография; обратная задача; некорректно поставленная задача; оценка апостериорного максимума; методы оптимизации

DOI: 10.14357/08696527180103

1 Введение

Магнитоэнцефалография — это неинвазивная технология регистрации сверхмалых электромагнитных возмущений, вызванных активностью множества нейронных источников. МЭГ-установка представляет собой шлем с множеством СКВИД-сенсоров (сверхчувствительных квантовых интерферометров), работающих на нестационарном эффекте Джосефсона, они постоянно находятся

[1]Московский государственный университет имени М. В. Ломоносова, факультет вычислительной математики и кибернетики, goncharenko.mir@yandex.ru

[2]Московский государственный университет имени М. В. Ломоносова, факультет вычислительной математики и кибернетики; Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, lsa@cs.msu.ru

в жидком гелии для поддержания состояния сверхпроводимости. Технология МЭГ широко используется как в клинических, так и в исследовательских целях из-за своего высокого пространственного и временно́го разрешения. Ткани и кости человека почти не искажают магнитный сигнал, а современные сенсоры способны работать с достаточно высокой частотой, чтобы регистрировать процессы, длящиеся всего десятки миллисекунд. Подробнее о МЭГ можно прочесть в статьях [1, 2].

Основной интерес исследователей представляет задача локализации активных в эксперименте участков коры по данным МЭГ. Восстановление конфигураций источников и их временно́й динамики требует обращения порождающей модели. Эта обратная задача относится к классу некорректно поставленных задач и требует дополнительных предположений об источниках активности для получения решения.

В данной статье будет рассмотрен байесовский подход к решению обратной задачи МЭГ в распределенной модели источников.

2 Обратная задача магнитоэнцефалографии

Порождающая модель сигналов МЭГ может быть представлена как

$$Y_t = L(J_t) + \varepsilon_t \,, \tag{1}$$

где Y_t — значение магнитной индукции, измеряемое МЭГ-сенсорами; $L(\cdot)$ — оператор Био–Савара–Лапласа; J — неизвестная плотность тока источников; ε_t — шум.

Единицей активации во всех формализациях прямой и обратной задачи являются токовые диполи. Токовый диполь представляет собой абстрактную физическую модель активации $5 \cdot 10^4$ нейронов на площади $\sim 30 = 5{,}5 \times 5{,}5$ мм; более подробное описание токовых диполей можно найти в [1].

Оператор Био–Савара–Лапласа для токового диполя имеет вид:

$$L_r = \frac{\mu_0}{4\pi} \frac{([\vec{Q} \times \vec{r_Q}], \vec{e_r})}{|\vec{r} - \vec{r_Q}|^3} \,, \tag{2}$$

где μ_0 — магнитная проницаемость среды; $\vec{Q}$ — его дипольный момент; $\vec{r_Q}$ — радиус-вектор диполя; $\vec{r}$ — радиус-вектор точки в пространстве, где определяется значение индукции, наведенное диполем; $\vec{e_r} = \vec{r}/\|r\|$ — единичный вектор.

С физической точки зрения обратная задача МЭГ представляет собой поиск конфигурации источников электромагнитной активности по конечному числу измерений. Теоретическая сторона этой задачи была исследована в XIX в. Германом Гельмгольцем, который доказал невозможность ее решения в общем виде.

С математической точки зрения ту же самую задачу можно сформулировать как задачу поиска обратного оператора.

Прямой оператор в (1) может быть представлен как $\mathfrak{L}_1 : (\mathbb{C}(V))^3 \to (\mathbb{C}(D))^3$, где $(\mathbb{C}(V))^3$ — пространство непрерывных функций внутри элемента объема мозга V; $(\mathbb{C}(D))^3$ — пространство непрерывных функций на поверхности D, включающей в себя границу V. В [3] было показано, что ядро $\mathfrak{L}_1$ содержит внутри себя линейное подпространство:

$$M = \{j | j = \Delta m, m \in (\mathbb{C}_0^2(V))^3)\},$$

где $\Delta m = \mathrm{div}(\mathrm{grad}(m))$ — оператор Лапласа, примененный к функции m; $\mathbb{C}_0^2$ — пространство дважды непрерывно дифференцируемых функций с показателем Гельдера, равным нулю.

Это означает, что у обратной задачи с прямым оператором $\mathfrak{L}_1$ нет уникального решения: если j^* — решение, то любое решение вида: $\tilde{j} = j^* + \lambda j'$, где $\lambda \in \mathbb{R}$ и $j' \in M$, также является решением.

Пусть оператор $\mathfrak{L}_2 : (\mathbb{L}^2(V))^3 \to (\mathbb{L}^2(V))^3$ переводит распределение токов в V в магнитное поле внутри того же объема. В этом случае все три компоненты как токов, так и магнитного поля, являются функциями в $\mathbb{L}^2$. В [4] было доказано, что $\mathfrak{L}_2$ — компактный оператор, следовательно, его псевдообратный оператор [5, 6] не будет непрерывным.

Общую модель (1) можно упростить, используя квазистатическую аппроксимацию уравнений Максвелла и уравнения Пуассона: $L(\cdot) \sim L$, где L — линейный оператор (известный под названием lead-field, или gain matrix).

Условия, позволяющие сделать подобный переход:

- топология мозга не изменяется во время эксперимента;
- все потоки (поток крови и т. д.) — стационарны (т. е. обладают одинаковыми характеристиками во время измерения);
- голова испытуемого не двигается и не двигаются лицевые мышцы.

В этом случае задача формулируется в линейной форме (Generalized linear model) как

$$Y = LJ + \varepsilon, \tag{3}$$

где $Y \in \mathbb{R}^{N_s \times N_t}$, $L \in \mathbb{R}^{N_s \times N_d}$, $J \in \mathbb{R}^{N_d \times N_t}$, $\varepsilon \in \mathbb{R}^{N_s \times N_t}$, а N_s, N_d и N_t — число сенсоров, возможных позиций диполей и временны́х отсчетов соответственно.

Подход (3) подразумевает фиксирование N_d позиций токовых диполей (или $3 \times N_d$, с учетом ориентации) и поиск интенсивности источников J в этих точках. Такая формализация задачи называется «имаджинговой». Важное замечание: $N_s \ll N_d$, обычно N_s имеет порядок нескольких десятков или сотен, а N_d — нескольких десятков тысяч. Таким образом, матрица Y далека от квадратной, и задача является некорректно поставленной.

Решением обратной задачи в такой постановке будет псевдообратный к L линейный оператор $M : \hat{J} = MY$.

Существуют и другие подходы к формализации обратной задачи МЭГ. Самой популярной альтернативой можно считать семейство «параметрических» методов (краткий обзор различных моделей прямой и обратной задачи можно найти в [7]). В «параметрической» постановке задача формализуется следующим образом:

$$L_t = \sum_{i=1}^{N_d} G\left(R_t^i\right) Q_t^i + \varepsilon_t \,, \tag{4}$$

где R_t^i и Q_t^i — позиция и дипольный момент i-го источника в момент времени t. Следует отметить, что количество источников активности в «параметрической» постановке подразумевается заранее не известным. Таким образом, решением (4) будет количество, позиция и амплитуда источников [7].

3 Байесовский вывод решения обратной задачи

Существует множество методов решения обратной задачи МЭГ, а также подходов к их получению. Данная секция посвящена применению байесовского подхода к обратной задаче МЭГ, в рамках которого будут получены в качестве частных случаев многие популярные методы.

К достоинствам байесовского подхода можно причислить следующие [8]:

- позволяет избавиться от избыточных переменных путем маргинализации и интегрирования;
- может быть удобно совмещен с техниками стохастического сэмплирования и оптимизации, такими как Markov Chain Monte Carlo (MCMC) и имитация отжига;
- выдает апостериорное распределение источников как результат работы.

В выкладках ниже будет использовано представление плотностей вероятности, принятое в оригинальных статьях. Выражения $p(X)$ и $p(X|Y)$ следует понимать как соответствующие: $p_X(x)$ — многомерная плотность X в точке $x \in \mathbb{R}^m$; $p_{X|Y}(x|y_0)$ — условная плотность X в точке $x \in \mathbb{R}^m$ относительно Y, равного $y_0 \in \mathbb{R}^m$.

Пусть $p(J)$ — априорное распределение источников, а $p(Y|J)$ — правдоподобие, с помощью которого корректируется априорное распределение источников в зависимости от данных.

Таким образом, становится возможным оценить апостериорное распределение активности источников с помощью теоремы Байеса:

$$p(J|Y) = \frac{p(Y|J)p(J)}{p(Y)} \,.$$

Обычно $p(Y)$ считают константой, так как измерения Y фиксированы ($p(Y)$ также называют «обоснованностью», или evidence). Непосредственная оценка амплитуды источников может быть получена как [9]:

$$\hat{J} = \mathbb{E}p(J|Y)\,,$$

где $\mathbb{E}$ — математическое ожидание.

Исходя из принципа максимизации энтропии, а также для упрощения математических выкладок обычно делают следующие предположения:

$$\varepsilon \sim \mathcal{N}(0, \Sigma_\varepsilon)\ ; \qquad J \sim \mathcal{N}(\mu_J, Q)\ ,$$

где $\mathcal{N}(\mu, \Sigma)$ обозначает многомерное нормальное распределение с N-компонентным вектором средних μ и матрицей ковариации Σ, имеющее плотность

$$p_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{N/2}\sqrt{|\Sigma|}} \exp\left\{-\frac{1}{2}\operatorname{tr}\left((x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right)\right\}\ .$$

Логарифмы $p_{\mu,\Sigma}(x)$ и $p(J|Y)$ имеют вид:

$$\log p_{\mu,\Sigma}(x) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma|) - \frac{1}{2}\operatorname{tr}\left((x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right)\ ; \quad (5)$$

$$\begin{aligned}\log p(J|Y) &\sim \log p(Y|J) + \log p(J) \sim \\ &\sim -\frac{1}{2}\operatorname{tr}\Big(J^{\mathrm{T}}(L^{\mathrm{T}}\Sigma_\varepsilon^{-1}L + Q^{-1})J - J^{\mathrm{T}}(L^{\mathrm{T}}\Sigma^{-1}Y + Q^{-1}\mu_J) - \\ &\qquad - \left(Y^{\mathrm{T}}\Sigma_\varepsilon^{-1}J + \mu_J^{\mathrm{T}}\right) + (Y^{\mathrm{T}}\Sigma_\varepsilon^{-1}Y + \mu_J^{\mathrm{T}}Q^{-1}\mu_J)\Big)\ . \quad (6)\end{aligned}$$

Из сравнения (5) и (6) следует, что

$$\operatorname{cov}\left(p(J|Y)\right) = \Sigma_J = \left(L^{\mathrm{T}}\Sigma_\varepsilon^{-1}L + Q^{-1}\right)^{-1}\ ; \quad (7)$$

$$\begin{aligned}\mathbb{E}p(J|Y) = \hat{J} &= \Sigma_J\left(L^{\mathrm{T}}\Sigma_\varepsilon^{-1}Y + Q^{-1}\mu_J\right) = \\ &= \left(L^{\mathrm{T}}\Sigma_\varepsilon^{-1}L + Q^{-1}\right)^{-1}\left(L^{\mathrm{T}}\Sigma_\varepsilon^{-1}Y + Q^{-1}\mu_J\right)\ . \quad (8)\end{aligned}$$

Пусть $\mu_J = 0$ для упрощения выкладок, тогда

$$M = \left(Q^{-1} + L^{\mathrm{T}}\Sigma_\varepsilon^{-1}L\right)^{-1}L^{\mathrm{T}}\Sigma_\varepsilon^{-1} = \{\text{см. [10]}\} = QL^{\mathrm{T}}\left(\Sigma_\varepsilon + LQL^{\mathrm{T}}\right)^{-1}.$$

Таким образом, оценка $\hat{J}$ представима в виде [10]:

$$\hat{J} = MY = QL^{\mathrm{T}} \left(\Sigma_\varepsilon + LQL^{\mathrm{T}} \right)^{-1} Y \,. \tag{9}$$

Оценка ковариационной матрицы (7) может быть использована для построения доверительных интервалов решений.

Существуют и другие подходы, приводящие к точно такой же результирующей формуле (9) [11]: минимизация математического ожидания ошибки; тихоновская регуляризация; обобщенная винеровская фильтрация.

4 Выбор априорных ковариационных матриц

Точность решения (9) сильно зависит от выбора Q и Σ_ε. В этой части статьи будет рассказано о некоторых популярных способах задания Q, а также предложено обобщение, позволяющее комбинировать различные подходы в рамках единой процедуры поиска решения.

Оценку Σ_ε можно получить с помощью записи так называемой «пустой комнаты», т. е. записи естественного шума сенсоров без испытуемого в камере. В противном случае обычно используется следующее представление: $\Sigma_\varepsilon = h_0 I_{N_s}$, где I_{N_s} — единичная матрица размера $N_s \times N_s$ [12, 13].

Далее будут рассмотрены наиболее важные частные случаи, получаемые в рамках байесовского подхода.

4.1 Minimum Norm Estimate

Minimum norm estimate (MNE) — наиболее широко применяемый метод решения обратной задачи МЭГ. Он может быть получен в рамках (9) при $Q = I_{N_d}$, т. е. в предположении, что все диполи имеют одинаковую априорную дисперсию и не коррелированы. Это решение также эквивалентно минимизации $\mathbb{L}^2$-нормы методом наименьших квадратов, т. е. поиску источников с наименьшей энергией. Из недостатков метода MNE можно отметить предпочтение поверхностных источников глубоким и «размазанность» получаемого решения (что во многом характерно для методов, использующих $\mathbb{L}^2$-регуляризацию [14]). Подробное описание метода можно найти в [15].

4.2 Weighted MNE

Weighted MNE (wMNE) борется со смещением решений в область более поверхностных источников, вводя меньшие штрафы глубоким источникам. Этого можно добиться, положив в (9) $Q = \mathrm{diag}\,\{Q_{jj}\}$, $Q_{jj} = ||L_{j\cdot}||^{-p}$, где $||\cdot||$ — евклидова норма; $L_{j\cdot}$ — j-й столбец матрицы L; p — положительный параметр штрафа глубины источника.

4.3 LOw REsolution TomograPhy

Метод LOw REsolution TomograPhy (LORETA), предложенный в [16], оперирует более сложной моделью источников, учитывая зависимость между диполем и некоторым множеством его соседей, предполагая коррелированность их активаций вследствие физической близости.

Решение LORETA может быть получено из (9) при $Q = Q_{\text{wMNE}}(KK^{\text{T}})^{-1}$. Здесь

$$K_{ij} = \begin{cases} 1 & \text{при } i = j\,; \\ -\dfrac{1}{|V_i|} & \text{при } j \in V_i\,; \\ 0 & \text{иначе,} \end{cases}$$

где V_i — множество соседей i-го источника.

4.4 Beamformer

Подход Beamformer подразумевает построение пространственного фильтра, который выделяет активность каждого отдельного диполя. Для построения подобного фильтра в (9) может быть использована следующая ковариационная матрица:

$$Q = B = \operatorname{diag}\{B_{ii}\}\,,\;\; B_{ii} = \frac{1}{\delta_i}\left(L_{\cdot i}^{\text{T}}(YY^{\text{T}})^{-1}L_{\cdot i}\right)^{-1}\,,$$

где $\delta_i = (L_{\cdot i}^{\text{T}}L_{\cdot i})^{-1}$, $i = \overline{1, N_d}$.

Более подробно о методах Beamformer можно прочесть в [17].

4.5 Обобщенный подход к выбору априорной матрицы

Для достижения большей гибкости алгоритма поиска решения в работе [18] была предложена следующая параметризация:

$$Q = \sum_{i=1}^{N_q} h_i D_i\,, \tag{10}$$

где набор $h = \{h_1, \dots h_{N_q}\}$ — гиперпараметры; $D = \{D_1, \dots, D_{N_q}\}$ — набор ковариационных составляющих.

Это представление позволяет использовать преимущества различных методов, комбинируя априорные ковариации D_i в зависимости от данных Y.

5 Сведение к задаче оптимизации

Пусть априорное распределение гиперпараметров (10) имеет следующий вид:

$$p(h) \sim \prod_{i=1}^{N_q} e^{f_i(h_i)}\,, \tag{11}$$

где $f_i(\cdot)$ — некоторые известные функции (предпочтительно выпуклые).

С учетом гиперпараметров распределение источников активности J представимо в виде:

$$p(J) = \int p(J,h)\,dh = \int p(J|h)p(h)\,dh\,.$$

На первом этапе необходимо получить оценку $\hat{h}$ гиперпараметров h, используя данные Y. Это сделает возможным вычисление $p(J|Y, h = \hat{h}) = p(J|Y)$ и дальнейшее использование функционала (8).

Оценка $\hat{h}$ может быть получена из $p(Y,h)$ в рамках подхода Empirical Bayes как оценка максимального правдоподобия [19].

Из формулы Байеса следует, что

$$p(Y,h) = p(Y|h)p(h)\,. \tag{12}$$

С другой стороны,

$$\begin{aligned} p(Y,h) &= \int p(Y,J,h)dJ = \int p(Y|J)p(J|h)p(h)dJ = \\ &= \{\text{так как } p(h) \text{ и } J \text{ независимы }\} = p(h)\int p(Y|J)p(J|h)\,dJ\,. \end{aligned} \tag{13}$$

Из сравнения (12) и (13) следует, что $p(Y|h) = \int p(Y|J)p(J|h)\,dJ$.

В предположении, что

$$p(Y|J) = \mathcal{N}\left(LJ, \Sigma_Y\right)\,; \quad p(J|h) = \mathcal{N}(0,Q), \tag{14}$$

где $\Sigma_Y = LQL^{\mathrm{T}} + \Sigma_\varepsilon$, результирующее распределение имеет вид:

$$p(Y|h) \sim \exp\left\{-\frac{1}{2}\,\mathrm{tr}\left(Y^{\mathrm{T}}\Sigma_Y^{-1}Y\right)\right\}. \tag{15}$$

Таким образом, функционал для оценки $\hat{h}$ не зависит от J и сформулирован только в терминах данных Y. Использование Σ_Y вместо Q существенно снижает

размерность задачи оптимизации, а суммирование с Σ_ε позволяет добавить регуляризационный параметр шума непосредственно в набор гиперпараметров.

Задача оптимизации для поиска $\hat{h}$ будет иметь вид:

$$\hat{h} = \arg\max_h p(Y,h) = \arg\max_h p(Y|h)p(h), \tag{16}$$

что равноценно максимизации $\Theta(h) = \log p(Y|h)p(h)$.

С учетом распределений $p(Y|h)$ (15) и $p(h)$ (11) итоговый вид оптимизируемого функционала может быть записан в явном виде:

$$\Theta(h) = -\frac{N_t}{2}\operatorname{tr}\left(C_Y\Sigma_Y^{-1}\right) - \frac{N_t}{2}\log|\Sigma_Y| - \frac{N_sN_t}{2}\log(2\pi) + \sum_{i=1}^{N_q} f_i\left(h_i\right), \tag{17}$$

где $C_Y = (1/N_t)YY^{\mathrm{T}}$ — выборочная ковариационная матрица.

6 Free Energy

Обычно у исследователей нет никаких представлений об оптимальном наборе гиперпараметров. Также истинное апостериорное распределение h может не быть гауссовым и возможно лишь вычисление некоторой аппроксимации апостериорного распределения.

Пусть $p_0(h)$ — априорное распределение h и $q(h)$ — аппроксимация апостериорного распределения. Следуя принципу максимума энтропии, предположим, что

$$p_0(h) = \mathcal{N}\left(\nu, \Pi^{-1}\right), \quad q(h) = \mathcal{N}\left(\hat{h}, \Sigma_h\right). \tag{18}$$

Тогда

$$\begin{aligned}\log p(Y) &= \int q(h)\log p(Y)\,dh = \int q(h)\log\frac{p(Y,h)}{p(h|Y)}\,dh = \\ &= \int q(h)\log\frac{p(Y,h)q(h)}{q(h)p(h|Y)}\,dh = \int q(h)\log\frac{p(Y,h)}{q(h)}\,dh + \\ &\quad + \int q(h)\log\frac{q(h)}{p(h|Y)}\,dh = F + \underbrace{D_{KL}[q(h)||p(h|Y)]}_{\geq 0},\end{aligned} \tag{19}$$

где $D_{KL}[p(x)||q(x)] = \int p(x)\log(p(x)/q(x))\,dx$ — расстояние Кульбака–Лейблера между плотностями p и q.

Слагаемое F также называют «свободной энергией» (free energy [20]). Таким образом, при $q(h) = p(h|Y)$; справедливы равенства $D_{KL} = 0$ и $F = \log p(Y)$; следовательно, максимизация (19) эквивалентна максимизации F.

В [20] приведен вывод следующей аппроксимации F:

$$\tilde{F} = \underbrace{-\frac{N_t}{2}\operatorname{tr}\left(C_Y\Sigma_Y^{-1}\right) - \frac{N_t}{2}\log|\Sigma_Y| - \frac{N_tN_s}{2}\log(2\pi)}_{\text{accuracy}} - \underbrace{-\frac{1}{2}\operatorname{tr}\left((\hat{h}-\nu)^{\mathrm{T}}\Pi(\hat{h}-\nu)\right) + \frac{1}{2}\log|\Sigma_h\Pi|}_{\text{complexity}}, \quad (20)$$

где $C_Y = (1/N_t)YY^{\mathrm{T}}$ — ковариационная матрица данных.

Первая часть (20) может рассматриваться как штраф точности аппроксимации данных, а вторая — как штраф сложности модели.

Таким образом, функционал (20) является обобщением (17) на случай аппроксимации апостериорного распределения h.

Алгоритм, использующий минимизацию функционала (20), называется Multiple Sparse Priors (MSP) и описан в работе [21].

7 Использование данных группового эксперимента

Агрегирование данных от нескольких испытуемых в рамках одного эксперимента позволяет повысить как устойчивость решения обратной задачи, так и точность локализации источников.

Формулировка прямой задачи МЭГ может быть расширена на случай эксперимента с N испытуемыми следующим образом [22, 23]:

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} L_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L_N \end{pmatrix} \begin{pmatrix} J_1 \\ \vdots \\ J_N \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{pmatrix}.$$

Данный подход увеличивает ранг матриц и делает возможным использование дополнительной информации для более точного выделения интересующей активности.

Все рассмотренные ранее методы естественным образом обобщаются на случай группового эксперимента. Также существуют специализированные алгоритмы, которые используют особенности задачи, выделяя одинаковую для всех испытуемых активность.

Примером такого алгоритма является метод GALA (Group Analysis Leads to Accuracy). Этот метод использует следующую процедуру для построения ковариационной матрицы данных Σ_Y из (14).

Все множество диполей Θ разбивается на подмножество $\Theta_{\text{общ}}$ — диполей со схожими активациями — и его дополнение $\Delta = \Theta \setminus \Theta_{\text{общ}}$. Алгоритм использует

итеративную процедуру для построений этих множеств. Пусть $\Theta^k_{\text{общ}}$ и Δ^k — соответствующие приближения на k-й итерации.

С учетом введенных обозначений декомпозиция матрицы ковариации данных на k-й итерации имеет вид:

$$\Sigma_Y^k = h_1 \Sigma_\varepsilon + L Q^k L^{\mathrm{T}}, \tag{21}$$

где

$$Q^k = h_2^k R^{\Theta^k_{\text{общ}}} + h_3^k D^{\Theta^k_{\text{общ}}} + \sum_{j=2}^{k} h_{2+j}^k D^{\Delta^j}.$$

Формула (21) представляет собой линейную комбинацию ковариации шума (Σ_ε), итеративных приближений матриц ковариации общей компоненты ($R^{\Theta^k_{\text{общ}}}$), индивидуальной ($D^{\Theta^k_{\text{общ}}}$) и остаточной ковариации (D^{Δ^j}), не объясненной предыдущими слагаемыми.

Таким образом, алгоритм разделяет активность на «общую», представляющую основной интерес, «индивидуальную» и «неизвестную», т. е. долю активности, не принадлежащую первым двум типам (которая вносит вклад в итоговое решение, начиная со второй итерации алгоритма).

Детальное описание алгоритма можно найти в работе [23].

8 Перспективные направления решения обратной задачи магнитоэнцефалографии в рамках байесовского подхода

Одним из возможных путей дальнейшего развития алгоритмов локализации источников может стать создание семейства методов, использующих дополнительные предположения о разреженности источников активности.

Как правило, решения, полученные с помощью методов, использующих $\mathbb{L}^2$-регуляризацию или, что эквивалентно, априорное нормальное распределение шума, дают «размазанное» решение с большим числом ненулевых источников малой амплитуды (популярный метод MNE [15] является характерным представителем такого семейства).

Для борьбы с этим недостатком возможен подход с применением априорного распределения Лапласа или $\mathbb{L}^1$-регуляризации. Похожий подход применяется в методе MCE [24]. Для получения разреженного решения может быть использована $\mathbb{L}^1$-регуляризация в задаче оптимизации (16), что эквивалентно применению априорного распределения Лапласа в (18). Руководствуясь схожими идеями, был создан метод MCE [24]. Из его недостатков можно выделить неспособность точно приближать большие участки активации. Возможным решением данной проблемы является применение комбинированной $\mathbb{L}^2\mathbb{L}^1$-регуляризации [7] и использование таких алгоритмов, как ElasticNet [25], RFM [26] и SFM [27]. Авторами статьи ведется работа в данном направлении.

С другой стороны, для достижения разреженности решения возможны адаптации техник compressed sensing по минимизации l^0-нормы (псевдонорма, равная числу ненулевых элементов в векторе). Но в [28] показано, что оператор (2) не удовлетворяет требованиям ограниченной изометрии (RIP, restricted isometry property), что порождает множество проблем применения теории compressed sensing.

Введение зависимости ковариации от времени [29], или поиск пространственно-временны́х кластеров, может быть еще одним источником улучшений, который практически не используется в текущих методах. При таком подходе потребуется введение дополнительного измерения — темпорального, что повышает сложность модели, но может увеличить точность решения за счет использования дополнительной информации.

Еще одним источником дополнительной информации могут быть разного рода сигналы, включающие как сигналы со вспомогательных датчиков во время эксперимента, так и структурные изображения, позволяющие точнее моделировать голову испытуемого. Вот некоторые из возможных источников вспомогательной информации: ЭЭГ — для уточнения конфигурации электромагнитного поля вне головы; фМРТ (fMRI, functional magnetic resonance imaging) — для построения априорного распределения; структурные данные, такие как MEFLASH (FLASH MRI), диффузионная МРТ (DW-MRI, diffusion-weighted MRI), структурная МРТ (sMRI, structural MRI), позволяющие более точно строить модель головы, различая при этом отдельные составляющие: череп, скальп, нервные волокна, серое и белое вещество.

Дополнительное исследование пациентов может сделать возможным персонализацию алгоритма для каждого испытуемого, но потребует построения нового класса комбинированных моделей.

Другим подходом к повышению гибкости модели является использование многоуровневой иерархической схемы гиперпараметров (Hierrarchical Bayes). Такой подход хорошо зарекомендовал себя во многих прикладных задачах машинного обучения. Однако при этом существенно увеличивается вычислительная сложность оптимизационной задачи (16) и могут потребоваться дополнительные эвристики для сокращения пространства поиска оптимальных параметров.

9 Заключение

Вероятностные методы широко и с успехом применяются в различных областях знаний. В представленной работе продемонстрирована возможность вероятностного подхода комбинировать различные популярные методы и гибко регулировать степень их вклада в итоговое решение путем введения гиперпараметров.

В данной статье (впервые на русском языке) приведен полный вывод решения обратной задачи МЭГ в вероятностной постановке, включая процедуру получения оценки гиперпараметров на основе данных эксперимента.

Гибкость байесовского подхода позволяет использовать дополнительную взаимную информацию в обобщении прямой задачи МЭГ на случай группового эксперимента. Модификации приведенного байесовского подхода как в теоретической, так и в алгоритмической частях способны повысить точность локализации активных зон коры мозга, что представляет несомненную ценность не только для нейрофизиологии, но и для других отраслей науки, в особенности биологии и психофизиологии.

Авторы статьи в дальнейшем планируют разработать метод на основе априорного распределения Лапласа, основной особенностью которого является поиск наиболее «разреженного» решения (т. е. такого распределения источников, которое приближало бы зарегистрированную активность наименьшим числом активных диполей).

Литература

1. *Hamalainen M., Hari R., Ilmoniemi R. J., Knuutila J., Lounasmaa O. V.* Magnetoencephalography — theory, instrumentation, and applications to noninvasive studies of the working human brain // Rev. Mod. Phys., 1993. Vol. 65. No. 2. P. 413–497. doi: 10.1103/revmodphys.65.413.
2. *Захарова Т. В., Никифоров С. Ю., Гончаренко М. Б., Драницына М. А., Климов Г. А., Хазиахметов М. Ш., Чаянов Н. В.* Методы обработки сигналов для локализации невосполнимых областей головного мозга // Системы и средства информатики, 2012. Т. 22. № 2. С. 157–175.
3. *Kress R., Kuhn L., Potthast R.* Reconstruction of a current distribution from its magnetic field // Inverse Probl., 2002. Vol. 18. No. 4. P. 1127–1146. doi: 10.1088/0266-5611/18/4/312.
4. *Cantarella J., Turck D. D., Gluck H.* The Biot–Savart operator for application to knot theory, fluid dynamics, and plasma physics // J. Math. Phys., 2001. Vol. 42. No. 2. P. 876–905. doi: 10.1063/1.1329659.
5. *Moore E. H.* On the reciprocal of the general algebraic matrix // B. Am. Math. Soc., 1920. Vol. 26. No. 9. P. 394–395. doi: 10.1090/S0002-9904-1920-03322-7.
6. *Penrose R.* A generalized inverse for matrices // Math. Proc. Cambridge, 1955. Vol. 51. No. 3. P. 406–413. doi: 10.1017/S0305004100030401.
7. *Pascarella A., Sorrentino A.* Statistical approaches to the inverse problem // Magnetoencephalography / Ed. E. W. Pang. — InTech, 2011. P. 93–112. http://www.intechopen.com/books/magnetoencephalography/statistical-approaches-to-the-inverse-problem.
8. *Idier J.* Bayesian approach to inverse problems. — Jersey City, NJ, USA: Wiley, 2008. 392 p. doi: 10.1002/9780470611197.
9. *Lopez J. D.* MEG/EEG brain imaging based on Bayesian algorithms for ill-posed inverse problems. — Manizales: Universidad Nacional de Colombia, 2012. PhD Thesis. 75 p.
10. *Grech R., Cassar T., Muscat J., Camilleri K., Fabri S., Zervakis M., Xanthopoulos P., Sakkalis V., Vantrumste B.* Review on solving the inverse problem in EEG source analysis // J. Neuroeng. Rehabil., 2008. Vol. 5. No. 1. P. 1–25. doi: 10.1186/1743-0003-5-25.

11. *Liu J. S., Chen R.* Sequential Monte Carlo methods for dynamic systems // J. Am. Stat. Assoc., 1998. Vol. 93. No. 443. P. 1032–1044. doi: 10.2307/2669847.
12. *Hansen P. C.* The L-curve and its use in the numerical treatment of inverse problems // Computational inverse problems in electrocardiology. — Southampton, U.K.: WIT Press, 2000. P. 119–142.
13. *Phillips C., Rugg M., Friston K.* Systematic regularization of linear inverse solutions of the EEG source localization problem // NeuroImage, 2002. Vol. 17. No. 1. P. 287–301. doi: 10.1006/nimg.2002.1175.
14. *Bishop C. M.* Pattern recognition and machine learning. — New York, NY, USA: Springer, 2006. 738 p.
15. *Hamalainen M. S., Ilmoniemi R. J.* Interpreting magnetic fields of the brain: Minimum norm estimates // Biomed. Eng., 1994. Vol. 32. No. 1. P. 35–42. doi: 10.1007/BF02512476.
16. *Pascual-Marqui R., Michel C. M., Lehmann D.* Low resolution electromagnetic tomography: A new method for localizing electrical activity in the brain // Int. J. Psychophysiol., 1994. Vol. 18. No. 1. P. 49–65. doi: 10.1016/0167-8760(84)90014-X.
17. *Sekihara K., Nagarajan S. S., Poeppel D., Marantz A., Miyashita Y.* Reconstructing spatio-temporal activities of neural sources using an MEG vector beamformer technique // IEEE T. Bio-Med. Eng., 2001. Vol. 48. No. 7. P. 760–771. doi: 10.1109/10.930901.
18. *Wipf D., Nagarajan S.* A unified Bayesian framework for MEG/EEG source imaging // NeuroImage, 2009. Vol. 44. No. 3. P. 947–966. doi: 10.1016/j.neuroimage.2008.02.059.
19. *Casella G.* An introduction to empirical Bayes data analysis // Am. Stat., 1985. Vol. 39. No. 2. P. 83–87. doi: 10.2307/2682801.
20. *Friston K., Mattout J., Trujillo-Barreto N., Ashburner J., Penny W.* Variational free energy and the laplace approximation // NeuroImage, 2007. Vol. 34. No. 1. P. 220–234. doi: 10.1016/j.neuroimage.2006.08.035.
21. *Friston K., Harrison L., Daunizeau J., Kiebel S., Phillips C., Trujillo-Barreto N., Henson R., Flandin G., Mattoutf J.* Multiple sparse priors for the M/EEG inverse problem // NeuroImage, 2008. Vol. 39. No. 3. P. 1104–1120. doi: 10.1016/j.neuroimage.2007.09.048.
22. *Litvak V., Friston K.* Electromagnetic source reconstruction for group studies // NeuroImage, 2008. Vol. 42. No. 4. P. 1490–1498. doi: 10.1016/j.neuroimage.2008.06.022.
23. *Kozunov V. V., Ossadtchi A.* GALA: Group analysis leads to accuracy, a novel approach for solving the inverse problem in exploratory analysis of group MEG recordings // Front. Neurosci. Switz., 2015. Vol. 9. No. 107. P. 1–20. doi: 10.3389/fnins.2015.00107.
24. *Uutela K., Hamalainen M., Somersalo E.* Visualization of magnetoencephalographic data using minimum current estimates // NeuroImage, 1999. Vol. 10. No. 2. P. 173–180. doi: 10.1006/nimg.1999.0454.
25. *Zou H., Hastie T.* Regularization and variable selection via the elastic net // J. R. Stat. Soc., 2005. Vol. 67. No. 2. P. 301–320. doi: 10.1111/j.1467-9868.2005.00503.x.
26. *Tatarchuk A., Mottl V., Eliseyev A., Windridge D.* Selectivity supervision in combining pattern-recognition modalities by feature- and kernel-selective Support Vector

Machines // 19th Conference (International) on Pattern Recognition Proceedings. — Tampa, FL, USA: IEEE, 2008. P. 324–328. doi: 10.1109/ICPR.2008.4761781.
27. *Tatarchuk A., Urlov E., Mottl V., Windridge D.* A support kernel machine for supervised selective combining of diverse pattern-recognition modalities // 9th Workshop (International) on Multiple Classifier Systems Proceedings. — Cairo, Egypt: Springer, 2010. P. 165–174.
28. *Tellen S.* Sparse reconstruction and realistic head modeling in EEG/MEG. — Munster: University of Munster, 2013. PhD Thesis. 140 p.
29. *Solin A., Jylänki P., Kauramäki J., et al.* Regularizing solutions to the MEG inverse problem using space–time separable covariance functions // Arxiv.org, 2016. 25 p. https://arxiv.org/abs/1604.04931.

Поступила в редакцию 04.10.17

PROBABILISTIC APPROACH TO SOLVING THE MAGNETOENCEPHALOGRAPHY INVERSE PROBLEM

***M. B. Goncharenko*[1] *and T. V. Zakharova*[1,2]**

[1]Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskiye Gory, GSP-1, Moscow 119991, Russian Federation
[2]Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The brain study is the one of the most popular research area in contemporary neuroscience. It accumulates efforts of broad research groups involving different kinds of experts: psychologists, mathematicians, etc. The main problem in this area is how to localize cerebral cortex activity using experimental data. This problem is critical for all neuroimaging methods (functional magnetic resonance imaging, electroencephalography, magnetoencephalography (MEG), etc.). In the paper, MEG data are considered. Magnetoencephalography is a non-invasive neuroimaging technique which allows recording extra weak magnetic fields generated by neurons in human brain. Sources reconstruction using MEG data is an ill-posed inverse problem. The paper considers the Bayesian derivation of the inverse problem solution for MEG data. The main steps are described and necessary calculations are provided. Particular attention was paid to such advantage of the Bayesian approach as universality. It was shown how other popular methods which are widely used in research could be obtained within the unified framework. A generalization to the group-wise experiment is also considered. The paper also provides possible ways of further improvement of the MEG inverse problem solving techniques using the Bayesian approach.

Keywords: Bayesian approach; magnetoencephalography; inverse problem; ill-posed problem; *a posteriori* maximum estimation; optimization methods

DOI: 10.14357/08696527180103

References

1. Hamalainen, M., R. Hari, R. J. Ilmoniemi, J. Knuutila, O. V. Lounasmaa. 1993. Magnetoencephalography — theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.* 65(2):413–497. doi: 10.1103/revmodphys.65.413.
2. Zakharova, T. V., S. Yu. Nikiforov, M. B. Goncharenko, M. A. Dranitsyna, G. A. Klimov, M. Sh. Khaziakhmetov, and N. V. Chayanov. 2012. Metody obrabotki signalov dlya lokalizatsii nevospolnimykh oblastey golovnogo mozga [Signal processing methods for localization of nonrenewable brain regions]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 22(2):157–175.
3. Kress, R., L. Kuhn, and R. Potthast. 2002. Reconstruction of a current distribution from its magnetic field. *Inverse Probl.* 18(4):1127–1146. doi: 10.1088/0266-5611/18/4/312.
4. Cantarella, J., D. D. Turck, and H. Gluck. 2001. The Biot–Savart operator for application to knot theory, fluid dynamics, and plasma physics. *J. Math. Phys.* 42(2):876–905. doi: 10.1063/1.1329659.
5. Moore, E. H. 1920. On the reciprocal of the general algebraic matrix. *B. Am. Math. Soc.* 26(9):394–395. doi: 10.1090/S0002-9904-1920-03322-7.
6. Penrose, R. 1955. A generalized inverse for matrices. *Math. Proc. Cambridge* 51(3):406–413. doi: 10.1017/S0305004100030401.
7. Pascarella, A., and A. Sorrentino. 2011. Statistical approaches to the inverse problem. *Magnetoencephalography*. Ed. E. W. Pang. InTech. 93–112. Available at: http://www.intechopen.com/books/magnetoencephalography/statistical-approaches-to-the-inverse-problem (accessed March 26, 2018).
8. Idier, J. 2008. *Bayesian approach to inverse problems*. New Jersey, NJ: Wiley. 392 p. doi: 10.1002/9780470611197.
9. Lopez, J. D. 2012. MEG/EEG brain imaging based on Bayesian algorithms for ill-posed inverse problems. Manizales: Universidad Nacional de Colombia. PhD Thesis. 75 p.
10. Grech, R., T. Cassar, J. Muscat, K. Camilleri, S. Fabri, M. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vantrumste. 2008. Review on solving the inverse problem in EEG source analysis. *J. Neuroeng. Rehabil.* 5(1):1–25. doi: 10.1186/1743-0003-5-25.
11. Liu, J. S., and R. Chen. 1998. Sequential Monte Carlo methods for dynamic systems. *J. Am. Stat. Assoc.* 93(443):1032–1044. doi: 10.2307/2669847.
12. Hansen, P. C. 2000. The L-curve and its use in the numerical treatment of inverse problems. *Computational inverse problems in electrocardiology*. Southampton, U.K.: WIT Press. 119–142.
13. Phillips, C., M. Rugg, and K. Friston. 2002. Systematic regularization of linear inverse solutions of the EEG source localization problem. *NeuroImage* 17(1):287–301. doi: 10.1006/nimg.2002.1175.
14. Bishop, C. M. 2006. *Pattern recognition and machine learning*. New York, NY: Springer. 738 p.
15. Hamalainen, M. S., and R. J. Ilmoniemi. 1994. Interpreting magnetic fields of the brain: Minimum norm estimates. *Biomed. Eng.* 32(1):35–42. doi: 10.1007/BF02512476.

16. Pascual-Marqui, R., C. M. Michel, and D. Lehmann. 1994. Low resolution electromagnetic tomography: A new method for localizing electrical activity in the brain. *Int. J. Psychophysiol.* 18(1):49–65. doi: 10.1016/0167-8760(84)90014-X.

17. Sekihara, K., S. S. Nagarajan, D. Poeppel, A. Marantz, and Y. Miyashita. 2001. Reconstructing spatio-temporal activities of neural sources using an MEG vector beamformer technique. *IEEE T. Bio-Med. Eng.* 48(7):760–771. doi: 10.1109/10.930901.

18. Wipf, D., and S. Nagarajan. 2009. A unified Bayesian framework for MEG/EEG source imaging. *NeuroImage.* 44(3):947–966. doi: 10.1016/j.neuroimage.2008.02.059.

19. Casella, G. 1985. An introduction to empirical Bayes data analysis. *Am. Stat.* 39(2):83–87. doi: 10.2307/2682801.

20. Friston, K., J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny. 2007. Variational free energy and the laplace approximation. *NeuroImage* 34(1):220–234. doi: 10.1016/j.neuroimage.2006.08.035.

21. Friston, K., L. Harrison, J. Daunizeau, S. Kiebel, C. Phillips, N. Trujillo-Barreto, R. Henson, G. Flandin, and J. Mattoutf. 2008. Multiple sparse priors for the M/EEG inverse problem. *NeuroImage* 39(3):1104–1120. doi: 10.1016/j.neuroimage.2007.09.048.

22. Litvak, V., and K. Friston. 2008. Electromagnetic source reconstruction for group studies. *NeuroImage* 42(4):1490–1498. doi: 10.1016/j.neuroimage.2008.06.022.

23. Kozunov, V. V., and A. Ossadtchi. 2015. GALA: Group analysis leads to accuracy, a novel approach for solving the inverse problem in exploratory analysis of group MEG recordings. *Front. Neurosci. Switz.* 9(107):1–20. doi: 10.3389/fnins.2015.00107.

24. Uutela, K., M. Hamalainen, and E. Somersalo. 1999. Visualization of magnetoencephalographic data using minimum current estimates. *NeuroImage* 10(2):173–180. doi: 10.1006/nimg.1999.0454.

25. Zou, H., and T. Hastie. 2005. Regularization and variable selection via the elastic net. *J. R. Stat. Soc.* 67(2):301–320. doi: 10.1111/j.1467-9868.2005.00503.x.

26. Tatarchuk, A., V. Mottl, A. Eliseyev, and D. Windridge. 2008. Selectivity supervision in combining pattern-recognition modalities by feature- and kernel-selective Support Vector Machines. *19th Conference (International) on Pattern Recognition Proceedings.* Tampa, FL: IEEE. 324–328. doi: 10.1109/ICPR.2008.4761781.

27. Tatarchuk, A., E. Urlov, V. Mottl, and D. Windridge. 2010. A support kernel machine for supervised selective combining of diverse pattern-recognition modalities. *9th Workshop (International) on Multiple Classifier Systems Proceedings.* Cairo, Egypt: Springer. 165–174.

28. Tellen, S. 2013. Sparse reconstruction and realistic head modeling in EEG/MEG. Munster: University of Munster. PhD Thesis. 140 p.

29. Solin, A., P. Jylanki, J. Kauramaki, *et al.* 2016. Regularizing solutions to the MEG inverse problem using space–time separable covariance functions. Unpublished. 25 p. Available at: https://arxiv.org/abs/1604.04931 (accessed March 26, 2018)

Received October 4, 2017

Contributors

Goncharenko Miroslav B. (b. 1991) — PhD student, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskiye Gory, GSP-1, Moscow 119991, Russian Federation; goncharenko.mir@yandex.ru

Zakharova Tatiana V. (b. 1962) — Candidate of Science (PhD) in physics and mathematics, associate professor, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskiye Gory, GSP-1, Moscow 119991, Russian Federation; senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; lsa@cs.msu.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 53–64

МЕТОДЫ ИНИЦИАЛИЗАЦИИ ВОКСЕЛЬНОГО ОБЪЕМА В ЗАДАЧЕ ТРЕХМЕРНОЙ РЕКОНСТРУКЦИИ

О. А. Яковлев[1]

Аннотация: Широкий класс алгоритмов трехмерной реконструкции по изображениям основан на представлении сцены с помощью воксельного объема. Под воксельным объемом понимается пространство, ограниченное, как правило, прямоугольным параллелепипедом и разбитое равномерной решеткой на кубы, называемые вокселами. Задача инициализации состоит в поиске ограничивающего параллелепипеда (ОП). Представлены четыре метода инициализации воскельного объема на основе известных проекционных матриц для всех снимков. Для каждого метода приведены экспериментальные оценки как объективных параметров, так и параметров, связанных с влиянием на процесс реконструкции. Последняя группа параметров была получена с помощью специальной процедуры обхода, которая также описана.

Ключевые слова: воксел; воксельный объем; трехмерная реконструкция

DOI: 10.14357/08696527180104

1 Введение

Воксельные модели широко применяются для представления сцены в различных областях трехмерной реконструкции: силуэтной реконструкции [1, 2], многовидовом стерео [3], SLAM (simultaneous localization and mapping) [4, 5]. Под инициализацией воксельного объема понимается поиск параллелепипеда, содержащего сцену, и его дискретизация равномерной решеткой. Размер решетки ограничен доступными вычислительными ресурсами, поэтому разрешение воксельной модели и точность реконструкции зависят только от объема ОП.

Процессу инициализации воксельного объема не уделено достаточного внимания в литературе. К примеру, в работе [3], обобщающей результаты множества исследований в области многовидового стерео, ОП рассматривается в качестве исходных данных. Многие авторы, рассматривающие инициализацию воксельного объема в качестве этапа реконструкции, ссылаются на достаточно раннюю работу [1], в которой изложен вариант поиска ОП при съемке вращаемого объекта неподвижной камерой.

[1]Орловский филиал Федерального исследовательского центра «Информатика и управление» Российской академии наук, maucra@gmail.com

Область применения описанных в настоящей работе методов поиска ОП не ограничивается многовидовым стерео и силуэтной реконструкцией; например, работа [5] показывает, что такая процедура может быть уместна и в задаче SLAM.

2 Оценка центра сцены

Примем, что все дальнейшие рассуждения касаются случая, когда сцена целиком наблюдается на снимках с того или иного ракурса. Для каждого снимка положим безусловно известными проекционную матрицу и прямоугольник, содержащий проекцию сцены. Далее будем обозначать через N число снимков, P_i — проекционную матрицу камеры для снимка i.

Пусть для каждого из N снимков с проекционными матрицами

$$P_i = \begin{bmatrix} \left(p_1^i\right)^{\mathrm{T}} \\ \left(p_2^i\right)^{\mathrm{T}} \\ \left(p_3^i\right)^{\mathrm{T}} \end{bmatrix}$$

известна оценка проекции центра сцены (x_i, y_i), $i = 1, 2, \ldots, N$. В случае отсутствия дополнительной информации такой оценкой может служить центр снимка. Исходя из того, что всякая точка (X, Y, Z) связана со своей проекцией (u, v) соотношением

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P[X\ Y\ Z\ 1]^{\mathrm{T}}, \quad \lambda \neq 0,$$

где P — проекционная матрица 3×4, можно построить переопределенную систему уравнений, решение которой методом наименьших квадратов даст приближение центра сцены на основе исходных оценок. Пусть (X, Y, Z) — искомая точка, а $Q = w[X\ Y\ Z\ 1]^{\mathrm{T}}$ — вектор ее однородных координат. Тогда для снимка i верно:

$$P_i Q = \begin{bmatrix} \lambda_i x_i \\ \lambda_i y_i \\ \lambda_i \end{bmatrix}.$$

Избавляясь от λ_i, получаем:

$$\begin{bmatrix} \left(x_i p_2^i - p_1^i\right)^{\mathrm{T}} \\ \left(y_i p_3^i - p_2^i\right)^{\mathrm{T}} \end{bmatrix} Q = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1}$$

Сформировав из уравнений (1) для каждого снимка матрицу

$$A = \begin{bmatrix} \left(x_1 p_3^1 - p_1^1\right)^{\mathrm{T}} \\ \left(y_1 p_3^1 - p_2^1\right)^{\mathrm{T}} \\ \left(x_2 p_3^2 - p_1^2\right)^{\mathrm{T}} \\ \left(y_2 p_3^2 - p_2^2\right)^{\mathrm{T}} \\ \vdots \\ \left(x_N p_3^N - p_1^N\right)^{\mathrm{T}} \\ \left(y_N p_3^N - p_2^N\right)^{\mathrm{T}} \end{bmatrix},$$

получаем линейную однородную задачу о наименьших квадратах:

$$\min_{Q \in \mathbb{R}^4 : \|Q\| = 1} (AQ)^{\mathrm{T}} (AQ), \tag{2}$$

решение которой может быть найдено с помощью сингулярного разложения матрицы A. Если $Q = [Q_1\ Q_2\ Q_3\ Q_4]^{\mathrm{T}}$ — решение задачи (2), то искомая точка будет иметь координаты $(Q_1/Q_4, Q_2/Q_4, Q_3/Q_4)$.

3 Оптимальная инициализация

Каждый прямоугольник r_i определяет бесконечную пирамиду $F_i \subset \subset \mathbb{R}^3$, всякая точка внутри или на поверхности которой проецируется внутрь или на границу r_i (рис. 1).

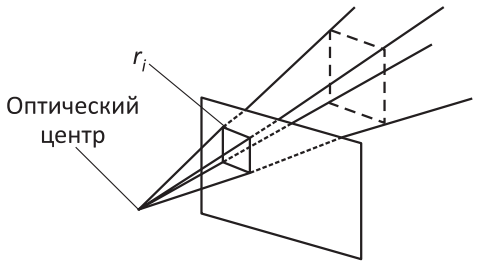


Рис. 1 Обратная проекция прямоугольника

Очевидно, что область пересечения всех пирамид $G = \cap_i F_i$ заведомо содержит в себе сцену. Так как F_i определяет выпуклое множество, то и G является выпуклым множеством. В общем случае область G является неограниченной; следовательно, не существует параллелепипеда конечного объема, содержащего сцену, и возникает необходимость в дополнительных ограничениях (например, максимально возможная удаленность сцены от камер). Если область G ограничена выпуклым многогранником (это возможно даже в случае двух снимков), то минимальный ОП для G будет оптимальным по величине объема ОП для сцены. Заметим, что F_i можно представить в виде пересечения четырех полупространств; следовательно, задача определения G сводится к пересечению полупространств. Эффективные алгоритмы пересечения полупространств, имеющие сложность $O(N \log N)$, описаны в работах [6–8].

Оптимальная инициализация может оказаться неоправданной ввиду потенциальной ресурсоемкости этого процесса, поэтому далее будут рассмотрены методы инициализации, имеющие высокую вычислительную эффективность.

4 Методы покрывающего куба и покрывающей сферы

Используя оценку центра сцены C, найдем наименьший по объему куб с центром в точке C, проекции которого покрывают все r_i. Рассмотрим предикат $H: \ s \in \mathbb{R}_+ \to \{0,1\}$, который принимает значение 1, если проекция куба со стороной s на i-й снимок покрывает r_i для всех $i = 1, 2, \ldots, N$. Предикат $H(s)$ монотонно не убывает с ростом s, что позволяет найти минимальное s, при котором $H(s) = 1$, методом дихотомии.

Заметим, что проекция куба на снимок является выпуклым многоугольником, поэтому проверка того, что проекция куба покрывает прямоугольник r_i, равносильна проверке на пересечение с кубом лучей, проходящих через все вершины прямоугольника r_i, что приводит к меньшим вычислительным затратам.

Проверка на пересечение луча и куба значительно упрощается благодаря тому, что стороны куба параллельны осям координат. Пусть (X_1, Y_1, Z_1) — вершина куба с минимальными значениями координат, а (X_2, Y_2, Z_2) — вершина с максимальными значениями координат, (X_0, Y_0, Z_0) и (a, b, c) — начало и направляющий вектор луча соответственно. Тогда луч и куб имеют общие точки, если система неравенств

$$\left.\begin{array}{c} X_1 \le X_0 + at \le X_2\,; \\ Y_1 \le Y_0 + bt \le Y_2\,; \\ Z_1 \le Z_0 + ct \le Z_2\,; \\ t \ge 0\,, \end{array}\right\} \tag{3}$$

имеет решение.

Асимптотическая сложность метода — $O(N \log(s_{\max}/\varepsilon))$. Здесь $s_{\max}$ — максимально возможная длина стороны куба, ε — требуемая точность или $O(NI)$, где I — число итераций двоичного поиска.

Можно заметить, что метод покрывающего куба определяет наиболее удаленный от центра сцены луч согласно метрике Чебышева. Если s — сторона искомого куба, $L_1, L_2, \ldots, L_{4N}$ — лучи, то

$$s = 2\left(\max_{1 \le i \le 4N} d_\infty(C, L_i)\right), \tag{4}$$

где $d_\infty(C, L_i)$ — расстояние Чебышёва от точки C до луча L_i.

Значение $d_\infty(C, L_i)$ можно определить аналитически — преобразовать систему (3) в параметрическую относительно стороны куба и найти минимальное

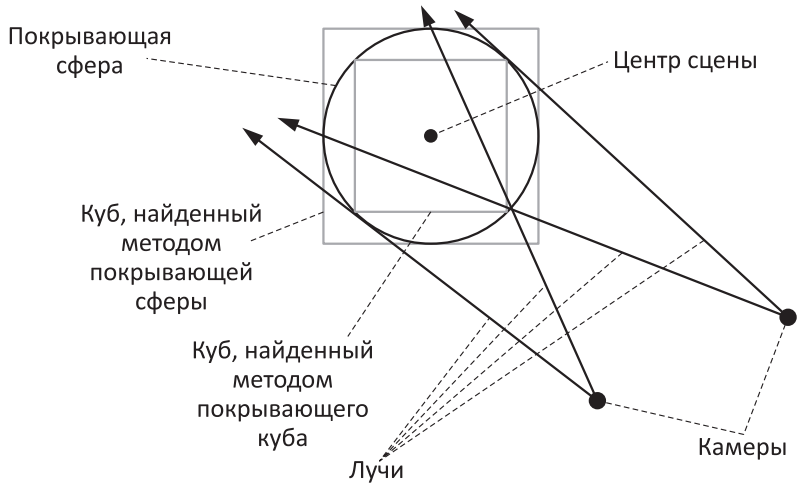


Рис. 2 Худший случай метода покрывающей сферы

значение параметра, при котором система разрешима. Поиск требуемого значения параметра подразумевает пересечение шести полуплоскостей, поэтому, несмотря на линейную вычислительную сложность, аналитический вариант будет иметь преимущество только при достаточно большом числе снимков.

Использование в выражении (4) евклидова расстояния вместо расстояния Чебышева позволит существенно повысить производительность. В этом случае будет определена наименьшая покрывающая сфера, а описанный около нее куб станет результирующим ОП.

Пусть V_c — объем ОП, найденного методом покрывающего куба, а V_s — объем ОП, найденного методом покрывающей сферы. В худшем случае, как показано на рис. 2, эти объемы соотносятся так же, как объемы вписанного и описанного кубов, т. е. $V_s/V_c \leq 3\sqrt{3}$.

Используя параметрическое представление луча $L = ((X_0, Y_0, Z_0), (a, b, c))$, расстояние от точки $p = (X, Y, Z)$ до него можно вычислить максимально эффективно (с точки зрения числа операций):

$$d(p, L) = \begin{cases} \| B \|_2 , & B^{\mathrm{T}} A < 0 , \\ \dfrac{\| A \times B \|_2}{\| A \|_2} , & B^{\mathrm{T}} A \geq 0 , \end{cases}$$

где $A = [a\ b\ c]^{\mathrm{T}}$; $B = [(X - X_0)\ (Y - Y_0)\ (Z - Z_0)]^{\mathrm{T}}$; $A \times B$ — векторное произведение.

5 Метод обратной проекции

Пусть $C = (C_x, C_y, C_z)$ — центр сцены. Построим обратные проекции прямоугольников r_i так, чтобы они лежали в плоскости Π_i, параллельной картинной плоскости снимка i и содержащей точку C.

Рассмотрим систему координат (СК) камеры i. Разобьем проекционную матрицу на компоненты

$$P_i = K_i \left[R_i \; t_i \right] ,$$

тогда координаты точки C в СК i-й камеры:

$$\begin{bmatrix} C'_x \\ C'_y \\ C'_z \end{bmatrix} = R_i \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} + t_i \, .$$

Плоскость Π_i в этой СК описывается уравнением $Z = C'_z$ и обратная проекция любой точки (x, y) на плоскость Π_i может быть вычислена как

$$\begin{bmatrix} X \\ Y \\ C'_z \end{bmatrix} = C'_z \left(K_i^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \right) .$$

Возвращаясь к глобальной СК, получаем точку

$$R_i^{\mathrm{T}} \left(\begin{bmatrix} X \\ Y \\ C'_z \end{bmatrix} - t_i \right) .$$

Результатом инициализации станет минимальный ОП для обратных проекций вершин всех r_i. Вычислительная сложность метода обратной проекции линейна относительно числа снимков.

6 Обход воксельного объема

Для последующей оценки описанных методов рассмотрим процедуру обхода, которая позволит оценить, насколько найденный ОП топологически близок к сцене, которая в него заключена. Рассмотрим задачу обобщенной реконструкции, которая заключается в постепенном отбрасывании из начального объема V вокселов, не принадлежащих сцене. На каждом этапе объем V можно разбить на четыре непересекающихся подмножества:

$$V = S \cup M^+ \cup M^- \cup I \, ,$$

где S — вокселы на поверхности объема; M^+ — вокселы, принадлежащие сцене (модель); M^- — отброшенные вокселы; I — вокселы внутри объема.

Назовем два воксела соседними, если они имеют общую грань, и обозначим $\Omega(v)$ — множество соседей воксела v. Будем считать, что воксел принадлежит поверхности, если существует точка в пространстве, из которой видна хотя бы одна его грань. Это равносильно тому, что у воксела отсутствует хотя бы один сосед из шести возможных.

Изначально

$$S = \{v \in V |\, |\Omega(v)| < 6\}\,;$$
$$I = V \backslash S\,;$$
$$M^+ = \emptyset\,, \quad M^- = \emptyset\,.$$

Построим процедуру обхода воксельного объема, сформулировав правила перемещения вокселов между множествами.

1. Проверке на принадлежность модели подлежат только вокселы из S. Таким образом, очередной воксел из множества S может быть помещен либо в множество M^+, либо в множество M^-.
2. Если некоторый воксел $v \in S$ принадлежит модели, то он просто перемещается из S в M^+:
$$S := S \backslash v\,,$$
$$M^+ := M^+ \cup v\,.$$
3. Если некоторый воксел $v \in S$ отбрасывается, то он делает видимой одну грань каждого соседнего воксела, который до этого момента был внутри объема, т. е. в множестве I:
$$S := (S \backslash v) \cup (I \cap \Omega(v))\,;$$
$$M^- := M^- \cup v\,;$$
$$I := I \backslash \Omega(v)\,.$$

Можно заметить, что максимальная мощность множества S зависит от того, в каком порядке просматриваются его элементы. Если множество S обрабатывать по принципу FIFO (first in, first out), то описанная процедура становится аналогичной обходу в ширину графа, вершинами которого являются вокселы, а ребра существуют между вершинами, соответствующими соседним вокселам. Следовательно, порядок $|S|$ составит $O\left(|V|^{2/3}\right)$, так как на каждом этапе в S будет содержаться не более двух уровней поверхности. Уровень поверхности можно определить так: начальное значение S — поверхность нулевого уровня L_0; множество вокселов, соседних с любым вокселом из L_0 и не содержащихся в L_0 — поверхность первого уровня L_1; вокселы, соседние с любым из L_1 и не содержащиеся в L_1 и L_0, образуют поверхность второго уровня L_2 и т. д.

7 Результаты

Описанные методы инициализации воксельного объема, оценки центра сцены и обхода воксельного объема были реализованы в контексте силуэтной реконструкции. В ходе экспериментов задействовались три набора входных данных: «Икосаэдр» (12 снимков), «Лошадь» (18 снимков) и «Кролик» (46 снимков). Наборы данных были получены методом виртуальной съемки трехмерных моделей (рис. 3) с помощью программы SfmDataGenerator [9], в которую была добавлена поддержка силуэтных снимков.

В качестве оценок центра сцены на снимках использовались центроиды силуэтов, а в качестве r_i — минимальные описанные прямоугольники силуэтов. Пересечение полупространств при оптимальной инициализации осуществлялось рандомизированным алгоритмом [8].

Во всех экспериментах использовался воксельный объем разрешением 300 вокселов по наибольшему измерению.

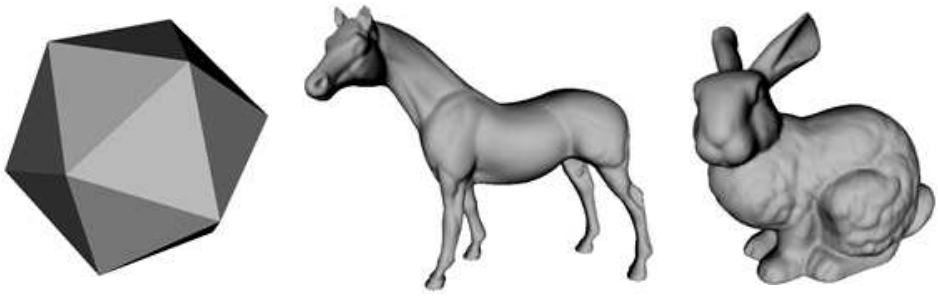


Рис. 3 Эталонные модели

Таблица 1 Результаты экспериментов с набором данных «Икосаэдр»

Метод инициализации объема	Объем найденного параллелепипеда, ед^3	Время инициализации объема, мс	Реконструкция				
			Число вокселов в модели $\lvert M^+\rvert$	Отброшено вокселов $\lvert M^-\rvert$	Не использовано вокселов $\lvert I\rvert$	Максимум вокселов в очереди $\max\lvert S\rvert$	Время, мс
Пересечение пирамид	**1,36**	0,55	191 567	14 225 048	9 162 521	490 356	**1710**
Покрывающий куб	3,33	0,16	114 965	22 851 321	4 304 615	540 002	2318
Покрывающая сфера	3,89	0,04	103 383	23 493 102	3 674 416	540 002	2340
Обратная проекция	3,24	**0,03**	107 114	19 960 492	3 872 128	495 742	2078

Таблица 2 Результаты экспериментов с набором данных «Лошадь»

Метод инициализации объема	Объем найденного параллелепипеда, ед^3	Время инициализации объема, мс	Реконструкция				
			Число вокселов в модели $\lvert M^+\rvert$	Отброшено вокселов $\lvert M^-\rvert$	Не использовано вокселов $\lvert I\rvert$	Максимум вокселов в очереди $\max\lvert S\rvert$	Время, мс
Пересечение пирамид	**0,0035**	0,63	82 785	13 744 191	1 111 224	367 192	**1515**
Покрывающий куб	0,0127	0,23	50 924	26 410 818	538 258	536 408	2613
Покрывающая сфера	0,0221	0,073	34 831	26 661 328	303 841	536 408	2581
Обратная проекция	0,0131	**0,04**	42 398	20 822 285	408 917	458 300	1996

Таблица 3 Результаты экспериментов с набором данных «Кролик»

Метод инициализации объема	Объем найденного параллелепипеда, ед^3	Время инициализации объема, мс	Реконструкция				
			Число вокселов в модели $\lvert M^+\rvert$	Отброшено вокселов $\lvert M^-\rvert$	Не использовано вокселов $\lvert I\rvert$	Максимум вокселов в очереди $\max\lvert S\rvert$	Время, мс
Пересечение пирамид	**0,003**	1,2	180 744	14 989 868	5 233 288	448 330	2379
Покрывающий куб	0,012	0,47	81 234	25 325 748	1 593 018	536 408	2785
Покрывающая сфера	0,019	0,06	60 700	25 908 304	1 030 996	536 408	2737
Обратная проекция	0,012	**0,04**	73 773	21 226 953	1 381 974	477 826	**2355**

Результаты экспериментов (табл. 1–4) показывают, что затраты на оптимальную инициализацию могут быть оправданы не только повышением точности реконструкции, но и ускорением последней за счет более эффективного использования вокселов. Среди остальных методов наилучший результат по производительности и качеству показал метод обратной проекции. Также приведенная оценка на порядок $|S|$ подтверждается экспериментально. Для более наглядного представления результатов введем величину, отражающую эффективность использования воксельного объема в процессе реконструкции. Эта величина может быть представлена процентом неотброшенных вокселов среди всех вокселов исходного объема: $(|M^+| + |I|) \,/\, (|M^+| + |I| + |M^-|) \cdot 100\%$.

На рис. 4 изображены полученные реконструкции в виде полигональных моделей, построенных алгоритмом «Марширующие кубы» [10].

Таблица 4 Эффективность использования вокселов

Метод инициализации	Сцена		
	«Икосаэдр», %	«Лошадь», %	«Кролик», %
Пересечение пирамид	**39,67**	**7,99**	**26,53**
Покрывающий куб	16,2	2,18	6,20
Покрывающая сфера	13,85	1,25	4,04
Обратная проекция	16,62	2,12	6,42

Рис. 4 Результаты реконструкции

8 Выводы

Обобщая свойства описанных методов и результаты экспериментов, можно заключить:

1. Затраты на оптимальную инициализацию могут быть компенсированы в процессе реконструкции при использовании процедуры обхода, не обрабатывающей заведомо невидимые вокселы. Также оптимальная инициализация не требует оценки центра сцены.
2. Инициализация объема методом обратной проекции дает ближайший к оптимальному результат и показывает наилучшее быстродействие.
3. Методы покрывающего куба и покрывающей сферы дают достаточно грубые оценки ОП, в частности потому, что не учитывают топологию сцены, так как найденный ОП всегда является кубом. Таким образом, каких-либо преимуществ от применения этих методов автором выявлено не было.

Литература

1. *Mülayim A. Y., Özün O., Atalay V., Schmitt F.* On the silhouette based 3D reconstruction and initial bounding cube estimation // 5th Fall Workshop (International) on Vision Modeling and Visualization Proceedings. — Berlin: Akademische Verlagsgesellschaft Aka GmbH, 2000. P. 11–18.

2. *Haro G.* Shape from silhouette consensus // Pattern Recogn., 2012. Vol. 45. No. 9. P. 3231–3244.
3. *Furukawa Y., Hernandez C.* Multi-view stereo: A tutorial // Found. Trends Computer Graphics Vision, 2013. Vol. 9. No. 1-2. P. 1–148.
4. *Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., Kohli P., Shotton J., Hodges S., Fitzgibbon A.* KinectFusion: Real-time dense surface mapping and tracking // 10th IEEE Symposium (International) on Mixed and Augmented Reality Proceedings. — Washington, DC, USA: IEEE, 2011. P. 127–136.
5. *Nießner M., Zollhöfer M., Izadi S., Stamminger M.* Real-time 3D reconstruction at scale using voxel hashing // ACM T. Graphic., 2013. Vol. 32. No. 6. Article No. 169.
6. *Brown K. Q.* Fast intersection of half spaces. — Pittsburg, PA, USA: Defense Technical Information Center, 1978. 26 p.
7. *Preparata F. P., Muller D. E.* Finding the intersection of n half-spaces in time $O(n \log n)$ // Theor. Comput. Sci., 1979. Vol. 8. No. 1. P. 45–55.
8. *Motwani R., Raghavan P.* Randomized algorithms. — New York, NY, USA: Cambridge University Press, 1995. 476 p.
9. *Яковлев О. А., Гасилов А. В.* Создание реалистичных наборов данных для алгоритмов трехмерной реконструкции с помощью виртуальной съемки компьютерной модели // Системы и средства информатики, 2016. Т. 26. № 2. С. 98–107.
10. *Lorensen W. E., Harvey C. E.* Marching cubes: A high resolution 3D surface construction algorithm // Comput. Graph., 1987. Vol. 21. No. 4. P. 163–169.

Поступила в редакцию 10.10.17

INITIAL BOUNDING BOX ESTIMATION METHODS FOR VOLUMETRIC THREE-DIMENSIONAL RECONSTRUCTION

O. A. Yakovlev

Orel Brach of the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 137 Moskovskoe Sh., Orel 302025, Russian Federation

Abstract: Volumetric models are widely used in various schemes of three-dimensional (3D) reconstruction: silhouette-based reconstruction, multiview stereo, and SLAM (simultaneous localization and mapping). The volume containing the scene to be reconstructed is divided by a regular grid into small cubes called voxels. This volume is usually bounded by an axis-aligned cube (bounding box). The preliminary to 3D reconstruction is to estimate a bounding box. The paper describes four methods of initial bounding box estimation and provides their experimental evaluation and comparison. The presented methods rely on known camera matrices and do not require any prior information about the scene.

Keywords: voxel; volumetric model; bounding box; 3D reconstruction

DOI: 10.14357/08696527180104

References

1. Mülayim, A. Y., O. Özün, V. Atalay, and F. Schmitt. 2000. On the silhouette based 3D reconstruction and initial bounding cube estimation. *5th Fall Workshop (International) on Vision Modeling and Visualization Proceedings*. Berlin: Akademische Verlagsgesellschaft Aka GmbH. 11–18.
2. Haro, G. 2012. Shape from silhouette consensus. *Pattern Recogn.* 45(9):3231–3244.
3. Furukawa, Y., and C. Hernandez. 2013. Multi-view stereo: A tutorial. *Found. Trends Computer Graphics Vision* 9(1-2):1–148.
4. Newcombe, R. A., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. *10th IEEE Symposium (International) on Mixed and Augmented Reality Proceedings*. Washington, DC: IEEE. 127–136.
5. Nießner, M., M. Zollhöfer, S. Izadi, and M. Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM T. Graphic.* 32(6). Article No. 169.
6. Brown, K. Q. 1978. *Fast intersection of half spaces*. Pittsburg, PA: Defense Technical Information Center. 26 p.
7. Preparata, F. P., and D. E. Muller. 1979. Finding the intersection of n half- spaces in time $O(n \log n)$. *Theor. Comput. Sci.* 8(1):45–55.
8. Motwani, R., and P. Raghavan. 1995. *Randomized algorithms*. New York, NY: Cambridge University Press. 476 p.
9. Yakovlev, O. A., and O. V. Gasilov. 2016. Sozdanie realistichnykh naborov dannykh dlya algoritmov trekhmernoy rekonstruktsii s pomoshch'yu virtual'noy s"emki komp'yuternoy modeli [Generating realistic structure-from-motion datasets through virtual photography]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(2):98–107.
10. Lorensen, W. E., and C. E. Harvey. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph.* 21(4):163–169.

Received October 10, 2017

Contributor

Yakovlev Oleg A. (b. 1992) — junior scientist, Orel Brach of the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 137 Moskovskoe Sh., Orel 302025, Russian Federation; maucra@gmail.com

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 65–76

МОДЕЛЬ ПРОЦЕССА КОРРЕКЦИИ ОШИБОК В СЕМАНТИЧЕСКОЙ СЕТИ

И. М. Адамович[1]***, О. И. Волков***[2]

Аннотация: Статья продолжает серию работ, посвященных моделированию ошибок независимых пользователей при формировании семантической сети, лежащей в основе распределенной технологии поддержки конкретно-исторических исследований. Данная статья посвящена описанию и обоснованию подхода к моделированию организационных мер поиска и исправления этих ошибок. Предложенный подход заключается в развитии модели семантической сети, построенной с опорой на принципы графодинамики и на модель Барабаши–Альберт с поддержкой параллельной фиксации ошибочных и соответствующих им безошибочных действий пользователей, за счет включения в нее механизмов реализации ролей пользователей и различных форм их активности. В рамках этого подхода также был проведен анализ влияния различных классов ошибок на качество сети. С помощью данной модели были проведены экспериментальные проверки эффективности организационных мер поиска и исправления ошибок семантической сети.

Ключевые слова: семантическая сеть; модель; ошибки пользователей; организационные меры; исправление ошибок

DOI: 10.14357/08696527180105

1 Введение

В статье [1] была описана новая распределенная технология поддержки историко-биографических исследований, для которой была обоснована форма организации информации в виде семантической сети. В статье [2] была поставлена и обоснована задача оценки качества этой семантической сети, формируемой одновременно множеством не связанных между собой исследователей. Была описана и обоснована модель, построенная на базе сочетания принципов графодинамики [3] с принципом предпочтительного присоединения [4], позволяющая изучить свойства информации, организованной в форме семантической сети, в динамике. В статье [5] было описано дальнейшее развитие модели семантической сети, предусматривающее включение в нее механизмов параллельной фиксации ошибочных и соответствующих им безошибочных (идеальных) действий

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, Adam@amsd.com

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, Volkov@amsd.com

пользователей, а также механизма имитации поисковых запросов, выполняемых параллельно в искаженной и неискаженной подсетях. Был выработан и обоснован интегральный количественный показатель качества семантической сети — степень деградации. С помощью модифицированной модели были проведены экспериментальные проверки влияния ошибок пользователей на качество семантической сети в динамике. Результаты проверок показали, что в широком диапазоне настраиваемых параметров значения полноты и точности остаются высокими и неограниченный рост степени деградации не происходит. Но при этом отмечается факт взаимовлияния ошибок при их накоплении, из чего вытекает желательность разработки организационных мер поиска и исправления ошибок.

Целью настоящей статьи является описание таких мер и экспериментальная проверка их эффективности.

В данной статье используются понятия и термины в соответствии с [2, 5].

2 Модель пользовательской активности

Исходно в модели модифицирующие воздействия на сеть никак не привязаны к пользователям и их характеристикам. Для изучения действий участников, направленных на исправление накапливаемых в семантической сети ошибок структуры, необходима модификация, включающая модель структуры сообщества пользователей технологии поддержки историко-биографических исследований. Структура сообщества интересна с точки зрения распределения участников по степени активности, где под активностью понимается выполнение следующих действий:

(1) пополнение подсети экземпляров семантической сети и выполнение информационных запросов;

(2) пополнение подсети понятий семантической сети;

(3) поиск и исправление ошибок семантической сети,

а под степенью активности понимается логарифм количества действий пользователя за фиксированный период (например, один месяц) [6].

Эти формы активности обладают свойством вложенности: первый вид активности осуществляется всеми участниками (участников, не ведущих никакой деятельности, можно считать несуществующими). Наиболее активная их часть также осуществляет и второй вид деятельности. И, наконец, самое активное ядро участников осуществляет третий вид деятельности наряду с первыми двумя.

Анализ конкретных примеров общей активности участников сообществ [7, 8] показывает, что распределение участников по степени активности одного вида соответствует усеченному нормальному закону с модой в области слабой активности. Усеченность слева возникает из-за дискретной природы активности, т. е. из-за невозможности фиксации активности ниже 1 действия за период.

Существующие примеры [9] сообществ с разными формами активности, также обладающих свойством вложенности, показывают, что распределение участников

по формам активности также соответствуют усеченному нормальному закону с модой в первой половине всего интервала форм активности. Таким образом, достаточно правдоподобное распределение участников по степени активности с учетом всех ее форм соответствует рисунку.

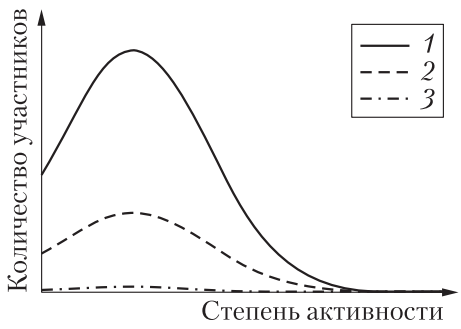


График плотности распределения участников по степени активности: *1* — пополнение подсети экземпляров; *2* — пополнение подсети понятий; *3* — поиск и исправление ошибок сети

При этом в реальном сообществе участники не обладают одной и той же степенью активности на протяжении всего срока существования сообщества. Более того, даже в самых устойчивых больших виртуальных коллективах за год персональный состав этих социумов обновляется как минимум на 50%–60% [10]. Но поскольку число участников на стационарном этапе развития сообщества [11], а также распределение участников по степени активности, как показано выше, остается неизменным, то без ущерба для результата в модели пользовательской активности можно принять постоянными состав участников и степени их активности.

3 Классы ошибок и их влияние на степень деградации

Ошибки пользователей, влияющие на структуру данных (ОСД), подробно описаны в [5]. Их можно разделить на 3 класса:

(1) ошибки подсети понятий: дублирование понятий, ошибки связей типа «значение» и ошибки иерархий;

(2) ошибки связей типа «часть» (Ч-связей) между экземплярами;

(3) ошибки запроса, вызванные обрубанием траектории на перегруженном подпонятиями узле иерархии.

Экспериментальная проверка показывает, что влияние разных классов ошибок на степень деградации сети неодинаково. Максимальный вклад в общий

Таблица 1 Экспериментальные значения удельного влияния ошибок по классам

№	Класс	Доля, %
1	Ошибки подсети понятий	87,11
2	Ошибки Ч-связей	11,97
3	Ошибки запроса	0,92
Итого		100,00

результат вносят ошибки подсети понятий, а минимальный — ошибки запроса. Усредненные значения влияния приведены в табл. 1.

Эти значения будут учтены при формировании организационных мер поиска и исправления ОСД.

4 Возможные организационные подходы

В принципе, возможные подходы к организации работ над неким проектом сводятся к следующим альтернативам [12]:

- инсорсинг — привлечение внутренних профессиональных ресурсов;
- аутсорсинг — привлечение внешних профессиональных ресурсов;
- краудсорсинг — привлечение широкого круга непрофессиональных ресурсов;
- экспертные сети (посткраудсорсинг, ноосорсинг) — возмездное проведение экспертиз со стороны сообщества профессионалов.

С точки зрения рассматриваемых в данной статье задач поиска и исправления ошибок семантической сети, поскольку привлечение методов ноосорсинга к рутинным задачам очевидно неэффективно, а инсорсинг и аутсорсинг в рамках сетевого проекта принципиально не отличаются, все многообразие подходов сводится к двум:

(1) решение задач силами компактной группы оплачиваемых профессионалов-администраторов семантической сети;

(2) решение задач силами широкого круга самих пользователей на безвозмездной основе.

Эффективность второго подхода не вызывает сомнений, с его помощью успешно осуществляется огромное число коммерческих и научных проектов от электронной энциклопедии Википедия до проекта НАСА по анализу марсианской поверхности NASA Clickworkers. На данный момент о краудсорсинге сложилось вполне обоснованное мнение, что «стоит это недорого, а то и вовсе ничего, а качество решений зачастую бывает гораздо выше, чем при заказе у профессионалов» [13]. Таким образом, можно ожидать, что второй подход применительно к задачам поиска и исправления ошибок семантической сети покажет более высокую эффективность, чем первый.

Но данное предположение требует тщательной проверки, поскольку интенсивность труда профессионалов всегда намного выше, чем у добровольца, осуществляющего свою деятельность на общественных началах.

5 Модификация базовых алгоритмов модели

На предыдущих этапах исследования действия над семантической сетью были не персонифицированы, т. е. модель не включала в себя понятия пользователя,

его атрибутов и роли. Базовые алгоритмы модели обеспечивали пополнение семантической сети и выполнение информационных запросов, но не предусматривали возможности корректировки сети. Для моделирования мер поиска и исправления ошибок потребовалось внести следующие дополнения в набор базовых алгоритмов (движок) модели.

5.1 Механизм формирования списка участников

В модифицированной модели поддерживаются две роли участников: пользователь и администратор. Число пользователей и число администраторов задаются как параметры модели. Формирование списка участников осуществляется следующим образом:

- для пользователя случайным образом в соответствии с описанной выше функцией плотности распределения определяется уровень его активности;
- в соответствии с законом распределения, также описанным выше, определяются формы активности, которые осуществляет данный пользователь;
- для каждой формы активности, реализуемой данным пользователем, в соответствии с его уровнем активности вычисляются его периоды осуществления акций. В реальности промежутки между акциями случайны, и вычисленный период является усредненным показателем. Но поскольку моделируется активность за большой промежуток времени, временны́е промежутки без ущерба для результата можно принять строго равными среднему значению;
- для каждой формы активности пользователя в очередь событий устанавливается акция на соответствующий вычисленному интервалу момент времени, как это описано ниже;
- добавление в список участников нового администратора осуществляется аналогично пользователю. Отличие состоит в том, что уровень активности администратора всегда максимален, а форма осуществляемой активности только одна — поиск и исправление ошибок сети.

5.2 Механизм поддержки очереди событий

Очередь событий представляет собой упорядоченный по времени набор описателей событий, каждый из которых содержит момент времени события, идентификатор участника и код формы активности. Для очереди события предусмотрены две процедуры:

(1) установка нового события в очередь;

(2) чтение ближайшего по времени события. При этом модельное время устанавливается равным времени события, данное событие убирается из очереди и в очередь добавляется новое событие, соответствующее тому же участнику, той же форме активности и моменту времени, равному текущему плюс

период, вычисленный для данного участника и данной формы активности при формировании списка участников.

5.3 Признак проверки и механизмы его поддержки

Для поддержки процедуры поиска ошибок в структуру данных семантической сети вводится признак проверки узла. При пополнении сети признаку проверки устанавливается значение «не проверено» в следующих случаях:

- при добавлении нового А- или Н-понятия;
- при добавлении нового экземпляра понятия, допускающего связи типа «часть» для своих экземпляров. При этом значение признака «не проверено» устанавливается и для всех прочих экземпляров этого понятия, поскольку добавление нового экземпляра изменяет структуру Ч-связей между остальными;
- при добавлении нового поддерева к дереву понятий Д-атрибута или А-понятия для узла, к которому осуществляется присоединение поддерева, поскольку количество исходящих связей у него может вырасти сверх объема внимания [14], что может явиться причиной ошибки запроса.

5.4 Механизмы поиска и исправления ошибок

1. Поиск ошибок подсети понятий: выбирается случайное А- или Н-понятие с признаком «не проверено». После проверки и исправления устанавливается признак «проверено». Исправление ошибок производится следующим образом:
   - для А-понятия удаляются все ветви иерархии с $\mathrm{Rank} = \mathrm{R}$. Для всех узлов иерархии с $\mathrm{Rank} = \mathrm{I}$ устанавливается $\mathrm{Rank} = \mathrm{RI}$;
   - для Н-понятия, совпадающего со своим прототипом ($\mathrm{A} = \mathrm{Proto(A)}$), для всех его атрибутов OA_i: $\mathrm{Rank(OA}_i\mathrm{)} = \mathrm{R}$, все их экземпляры заменяются на экземпляры $\mathrm{Proto(OA}_i\mathrm{)}$. При этом если $\mathrm{Type(OA}_i\mathrm{)} \in \{\text{«Д»}, \text{«З»}\}$ и $\exists\, \mathrm{T} = \mathrm{Trace(D(OA}_i\mathrm{))}$, то у экземпляра атрибута вместо T строится новая траектория T' до узла, связанного с экземпляром соответствующей связью ProtoTrace. Все иерархии обрабатываются аналогично А-понятию. Исправленное понятие и все его экземпляры принимают $\mathrm{Rank} = \mathrm{RI}$;
   - для Н-понятия, не совпадающего со своим прототипом ($\mathrm{A} \neq \mathrm{Proto(A)}$), атрибуты обрабатываются аналогично. При этом иерархии не обрабатываются, поскольку Proto(A) не получает признака «проверено». Все экземпляры А заменяются на экземпляры Proto(A). Proto(A) и все его экземпляры принимают $\mathrm{Rank} = \mathrm{RI}$, а понятие А удаляется. Все входящие З-связи на А заменяются на З-связи на Proto(A).

2. Поиск ошибок Ч-связей: выбирается случайный экземпляр с признаком «не проверено». После проверки и исправления Ч-связей данного экземпляра у него устанавливается признак «проверено». После этого осуществляется аналогичная проверка всех экземпляров, связанных с ним Ч-связями, но в количестве не более объема внимания за одну акцию.
3. Поиск перегруженных подпонятиями узлов иерархии: выбирается случайный иерархический узел с признаком «не проверено». За одну акцию проверяется все дерево, которому принадлежит данный узел. При нахождении перегруженного узла вводятся промежуточные узлы, группирующие подпонятия в промежуточные понятия так, чтобы на каждом уровне иерархии количество подпонятий понятия было в пределах объема внимания. У экземпляров соответствующим образом корректируются траектории Trace. После проверки и исправления дерева у всех его узлов устанавливается признак «проверено».

6 Модификация логики модели

1. Принципиальным изменением модели является введение в нее модельного времени за счет описанного выше механизма очереди событий. Период моделирования выбран длительностью в 1 месяц модельного времени, поскольку уровень активности участников в соответствии с описанной выше функцией плотности распределения составляет не менее 1 действия в месяц и, следовательно, данного периода достаточно для того, чтобы абсолютно все участники проявили свою активность.
2. Для сравнения эффективности двух описанных выше организационных подходов в модель вводится поддержка трех сценариев:
   (а) контрольный сценарий (без исправления ошибок сети): в список участников вводятся только участники с ролью «пользователь». При этом пользователи осуществляют только две первые формы активности — «пополнение подсети экземпляров семантической сети и выполнение информационных запросов» и «пополнение подсети понятий семантической сети» в соответствии с описанным выше законом распределения активности;
   (б) краудсорсинговый подход: в список участников вводятся только участники с ролью «пользователь». При этом пользователи поддерживают все три формы активности в соответствии с описанным выше законом распределения активности. Участники с ролью «администратор» в список участников не вводятся;
   (в) профессиональный подход: в список участников вводятся участники с ролями как «пользователь», так и «администратор». При этом пользователи осуществляют только две первые формы активности —

«пополнение подсети экземпляров семантической сети и выполнение информационных запросов» и «пополнение подсети понятий семантической сети» в соответствии с описанным выше законом распределения активности, а администраторы осуществляют исключительно третью форму активности — «поиск и исправление ошибок семантической сети» с максимальной активностью. Число администраторов не превышает 10, что составляет 0,01%–0,1% от числа участников.

Один эксперимент на заданных параметрах модели включает в себя расчеты по каждому из трех сценариев и сравнение их результатов.

3. Поскольку на данном этапе исследования предметом изучения является не динамика степени деградации сети, а сравнение результатов применения двух организационных подходов, серии информационных запросов проводятся не периодически по ходу формирования сети, а однократно по завершению периода моделирования при выполнении каждого из трех сценариев.

4. Для сравнения результатов такой показатель, как степень деградации сети [5], представляется неудобным из-за его нелинейности. Для оценки эффективности определим показатель $\overline{F} = 1 - F$, где F — усредненная F-мера (F-measure) [15], вычисляемая как среднее гармоническое точности и полноты. Показатель $\overline{F}$, как и S [5], представляет собой меру деградации сети, но, в отличие от S, ограничен интервалом $[0; 1]$.

7 Результаты

Были проведены две серии экспериментальных проверок модели:

(1) с реалистичным пределом 10%-ной вероятности ошибки пользователя [5] (серия 1);

(2) с запредельно высоким пределом 50%-ной вероятности ошибки пользователя (серия 2) для большей наглядности проявления эффекта.

Усредненные результаты приведены в табл. 2. Для каждой серии указано значение $\overline{F}$, а для сценариев с исправлением ошибок — их эффективность,

Таблица 2 Усредненные результаты экспериментальных проверок

Сценарий	Серия 1		Серия 2	
	$\overline{F}$	Доля снижения, %	$\overline{F}$	Доля снижения, %
Без исправления	0,06	—	0,24	—
Краудсорсинговое исправление	0,06	0,00	0,24	0,00
Профессиональное исправление	0,00	100,00	0,13	45,83

выраженная в доле снижения $\overline{F}$ по сравнению с контрольным сценарием без исправления ошибок.

Проверка показала, что эффективность краудсорсингового подхода в задачах поиска и исправления ошибок независимых пользователей при формировании семантической сети, лежащей в основе распределенной технологии поддержки конкретно-исторических исследований, крайне низка. Снижение $\overline{F}$ незначительно и находится в пределах точности эксперимента в обеих сериях.

Подход же, основанный на привлечении компактной группы оплачиваемых профессионалов, продемонстрировал высокую эффективность, составившую 100% на реалистическом сценарии и 45,83% на сценарии с запредельно высокой вероятностью ошибки пользователя.

8 Выводы

Предположение о более высокой эффективности подхода, основанного на принципах краудсорсинга, применительно к задачам поиска и исправления ошибок семантической сети по сравнению с подходом, основанным на привлечении компактной группы оплачиваемых профессионалов-администраторов семантической сети, оказалось ложным. Таким образом, при реализации технологии поддержки конкретно-исторических исследований с опорой на семантическую сеть следует принять один из двух вариантов:

(1) полностью отказаться от механизмов поиска и исправления ошибок пользователя, что, как было показано в [5], вполне допустимо, поскольку степень деградации сети по мере ее развития растет незначительно, сделав упор на предотвращении ошибок за счет дружественного интерфейса и т. п.;

(2) обеспечить поиск и исправление ошибок пользователей за счет привлечения профессиональных администраторов, что потребует дополнительного финансирования.

Реализация этой задачи силами самих пользователей без каких-либо дополнительных мер по повышению их мотивации представляется бесперспективной.

Литература

1. *Адамович И. М., Волков О. И.* Технология распределенного автоматизированного анализа исторических текстов // Системы и средства информатики, 2016. Т. 26. № 3. С. 148–161.
2. *Адамович И. М., Волков О. И.* Об одном подходе к моделированию процесса развития семантической сети // Системы и средства информатики, 2017. Т. 27. № 2. С. 143–154.
3. *Юдицкий С. А.* Графодинамическое имитационное моделирование развития сетевых структур // Управление большими системами, 2011. Вып. 33. С. 21–34.

4. *Бадрызлов В. А.* Принципы генерации случайных графов для моделирования сети Интернет // Омский научный вестник, 2014. № 3(133). С. 204–208.
5. *Адамович И. М., Волков О. И.* Влияние ошибок пользователей на динамику качества семантической сети // Системы и средства информатики, 2017. Т. 27. № 4. С. 150–163.
6. *Польдин О. В., Матвеева Н. Н., Стерлигов И. А., Юдкевич М. М.* Публикационная активность вузов: эффект проекта «5–100» // Вопросы образования, 2017. № 2. С. 10–35.
7. *Бреслер М. Г.* Методика комплексного анализа сетевого сообщества // Nauka-Rastudent.ru, 2015. № 09(21). http://nauka-rastudent.ru/21/2919.
8. *Сухарькова М. П.* Онлайн-активность олимпийского волонтерского движения «Сочи-2014» до и после игр // Информационное общество, 2016. № 6. С. 46–56.
9. *Сухарев О. С., Курманов Н. В.* Элементы маркетингового анализа социальной сети // Экономический анализ: теория и практика, 2012. № 35. С. 2–8.
10. *Нестеров В. Ю.* К вопросу о динамике сетевых сообществ // Библиотека учебной и научной литературы, 2002. 8 с. http://sbiblio.com/BIBLIO/archive/nesterov_at_question.
11. *Сазанов В. М.* Сравнительный анализ социально-сетевых проектов // Социально-ориентированные сети и технологии: Исследовательский ресурс социального Интернета, 2009. http://v-school.narod.ru/PAPERS/analiz.doc.
12. *Селиверстова П. О.* Аутсорсинг, краудсорсинг и экспертные сети как альтернатива внутренним ресурсам организации (инсорсингу) // Экономика и менеджмент инновационных технологий, 2014. № 11. С. 39–45.
13. *Хау Дж.* Краудсорсинг: Коллективный разум как инструмент развития бизнеса / Пер. с англ. — М.: Альпина Паблишер, 2012. 288 с. (*Howe J.* Crowdsourcing: Why the power of the crowd is driving the future of business. — 2nd ed. — New York, NY, USA: Three Rivers Press, 2009. 311 p.)
14. *Комарова Т. К.* Психология внимания. — Гродно: ГрГУ, 2002. 124 с.
15. *Агеев М. С., Кураленок И. Е., Некрестьянов И. С.* Официальные метрики РОМИП'2010 // Российский семинар по оценке методов информационного поиска: Тр. РОМИП'2010. — Казань: КФУ, 2010. С. 172–187.

Поступила в редакцию 03.11.17

THE MODEL OF SEMANTIC NET ERROR CORRECTION PROCESS

I. M. Adamovich and O. I. Volkov

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article continues the series of articles on modeling of errors of independent users in forming of semantic net which is the basis for the distributed technology of concrete historical investigation support. The article is devoted to the description and rationale of the approach to modeling of

organizational measures designed to find and correct these errors. The proposed approach is to expand the semantic net model which is based on the concepts of graphodynamics and the Barabasi–Albert model with the mechanism of parallel fixation of mistaken and appropriate nonmistaken user actions by including a new mechanism of user roles and various forms of their activity. In the process of applying this approach, the analysis of influence of different classes of errors on the network quality was made. Experimental studies of effectivity of organizational measures of finding and correction of semantic network errors were undertaken using this model.

Keywords: semantic net; model; user errors; organizational measures; error correction

DOI: 10.14357/08696527180105

References

1. Adamovich, I. M., and O. I. Volkov. I. 2016. Tekhnologiya raspredelennogo avtomatizirovannogo analiza istoricheskikh tekstov [The distributed automated technology of historical texts analysis]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 3(26):148–161.
2. Adamovich, I. M., and O. I. Volkov. 2017. Ob odnom podkhode k modelirovaniyu protsessa razvitiya semanticheskoy seti [An approach to modeling the semantic net evolution]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 2(27):143–154.
3. Yuditskiy, S. A. 2011. Grafodinamicheskoe imitatsionnoe modelirovanie razvitiya setevykh struktur [Graphodynamic simulation modeling of network structures evolution]. *Upravlenie bol'shimi sistemami* [Large-Scale Systems Control] 33:21–34.
4. Badryzlov, V. A. 2014. Printsipy generatsii sluchaynykh grafov dlya modelirovaniya seti Internet [The principles of generation of random graphs for simulation of the Internet]. *Omskiy nauchnyy vestnik* [Omsk Scientific Bull.] 3(133):204–208.
5. Adamovich, I. M., and O. I. Volkov. 2017. Vliyanie oshibok pol'zovateley na dinamiku kachestva semanticheskoy seti [The influence of user errors on the semantic network quality dynamics]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 4(27):150–163.
6. Poldin, O. V., N. N. Matveeva, I. A. Sterligov, and M. M. Yudkevich. 2017. Publikatsionnaya aktivnost' vuzov: effekt proekta "5–100" [Publication activities of Russian universities: The effects of Project 5–100]. *Voprosy obrazovaniya* [Educational Studies] 2:10–35.
7. Bresler, M. G. 2015. Metodika kompleksnogo analiza setevogo soobshchestva [Methods of complex analysis of network community]. *Nauka-Rastudent.ru* 09(21). Available at: http://nauka-rastudent.ru/21/2919/ (accessed October 5, 2017).
8. Sukhar'kova, M. P. 2016. Onlayn aktivnost' olimpiyskogo volonterskogo dvizheniya "Sochi-2014" do i posle igr [Online activity of the Olympic volunteer movement "Sochi-2014" before and after Games]. *Informatsionnoe obshchestvo* [Information Society] 6:46–56.

9. Sukharev, O. S., and N. V. Kurmanov. 2012. Elementy marketingovogo analiza sotsial'noy seti [Elements of marketing analysis of social network]. *Ekonomicheskiy analiz: teoriya i praktika* [Economic Analysis: Theory and Practice] 11(35):2–8.
10. Nesterov, V. U. 2002. K voprosu o dinamike setevykh soobshchestv [On the dynamics of network communities]. *Biblioteka uchebnoy i nauchnoy literatury* [Library of Educational and Scientific Literature]. 8 p. Available at: http://sbiblio.com/BIBLIO/archive/nesterov_at_question/ (accessed October 5, 2017).
11. Sazanov, V. M. 2009. Sravnitel'nyy analiz sotsial'no-setevykh proektov [Comparative analysis of social-network projects]. *Sotsial'no-orientirovannye seti i tekhnologii. Issledovatel'skiy resurs sotsial'nogo Interneta* [Socially-oriented networks and technologies. Research resource of social Internet]. Available at: http://v-school.narod.ru/PAPERS/analiz.doc (accessed October 5, 2017).
12. Seliverstova, P. O. 2014. Autsorsing, kraudsorsing i ekspertnye seti kak al'ternativa vnutrennim resursam organizatsii (insorsingu) [Outsourcing, crowdsourcing and expert networks as alternative to internal resources of the organization]. *Ekonomika i menedzhment innovatsionnykh tekhnologiy* [Economics and Innovations Management] 11(38):39–45.
13. Howe, J. 2009. *Crowdsourcing: Why the power of the crowd is driving the future of business.* New York, NY: Three Rivers Press. 311 p.
14. Komarova, T. K. 2002. *Psikhologiya vnimaniya* [Psychology of attention]. Grodno, Belarus: Yanka Kupala State University of Grodno. 124 p.
15. Ageev, M. S., I. E. Kuralenok, and I. S. Nekrest'yanov. 2010. Ofitsial'nye metriki ROMIP'2010 [ROMIP'2010 official metrics]. *Rossiyskiy seminar po otsenke metodov informatsionnogo poiska: Tr. ROMIP'2010* [ROMIP: Russian Information Retrieval Evaluation Seminar Proceedings]. Kazan: Kazan University Publs. 172–187.

Received November 3, 2017

Contributors

Adamovich Igor M. (b. 1934) — Candidate of Science (PhD) in technology, Head of Department, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119133, Russian Federation; Adam@amsd.com

Volkov Oleg I. (b. 1964) — leading programmer, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119133, Russian Federation; Volkov@amsd.com

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 77–88

ИНФОРМАТИВНОСТЬ КИНЕТИЧЕСКОГО ЭКСПЕРИМЕНТА И ОБЛАСТИ НЕОПРЕДЕЛЕННОСТИ ПАРАМЕТРОВ КИНЕТИЧЕСКИХ МОДЕЛЕЙ*

***С. И. Спивак*[1], *Ф. Т. Зиганшина*[2], *А. С. Исмагилова*[3]**

Аннотация: В решении обратных задач химической кинетики часто сталкиваются с неоднозначностью определения кинетических параметров. Неоднозначность обратной задачи состоит в том, что описание любой математической модели должно отражать некоторые химические реакции в определенной области входных параметров. Ввиду неоднозначности решения используются методы оценки кинетических параметров. Одним из таких методов является метод определения интервалов и областей неопределенности, основанный на идее Л. В. Канторовича.

Ключевые слова: обратная задача; химическая кинетика; информативность эксперимента; интервалы неопределенности

DOI: 10.14357/08696527180106

1 Введение

Предметом исследования настоящей работы являются обратные задачи идентификации механизмов сложных химических реакций на основании кинетических измерений.

Обратная задача состоит в определении констант скоростей элементарных стадий, входящих в механизм сложной химической реакции, на основе экспериментальных данных о концентрациях участвующих в реакции веществ.

Основная трудность — измерению, как правило, доступна только часть участвующих в реакции веществ. Следствием такой недоинформативности является неединственность решения обратной задачи.

В [1–3] проведено детальное математическое исследование проблемы информативности:

– приведена классификация типов неединственности решения обратных задач химической кинетики в зависимости от типа эксперимента;

*Работа выполнена при финансовой поддержке РФФИ и Правительства Республики Башкортостан (проект 17-47-020068).

[1]Башкирский государственный университет; Институт нефтехимии и катализа, semen.spivak@mail.ru

[2]Уфимский государственный нефтяной технический университет, fairusa85@mail.ru

[3]Башкирский государственный университет, ismagilovaas@rambler.ru

– построена методология анализа информативности кинетических измерений при решении обратных задач, позволяющая выделить число и вид независимых комбинаций констант скоростей реакций, допускающих однозначное оценивание по разным типам кинетического эксперимента;
– доказано, что измеряемые характеристики механизма реакции инвариантны относительно некоторых преобразований кинетических параметров. Доказано, что эти преобразования являются групповыми (непрерывными или дискретными в зависимости от типа эксперимента);
– построена методология редукции систем дифференциальных уравнений химической кинетики к системам меньшей размерности при условии сохранения адекватности реальным массивам измерений.

Неединственность решения обратных задач химической кинетики приводит к существованию областей неопределенности кинетических констант. Под областью неопределенности будем понимать такую область, внутри которой вариация кинетических констант позволяет описывать кинетические измерения в пределах величины их погрешности.

Задачей настоящей работы будет создание методов расчета и анализа таких областей.

2 Методы

Модель описывает измерения в пределах их точности, если выполняется следующая система неравенств:

$$|C_{\text{exp}} - C_{\text{calc}}| \leq \varepsilon\,,$$

где ε — вектор предельно допустимой погрешности измерений при экспериментальном определении C.

Неравенство, приведенное в этой статье, представлено как система неравенств. Эта система неравенств характеризует вариацию экспериментальных данных в пределах величины их максимальной относительной погрешности и имеет вид:

$$\max_{1\leq i\leq N} \frac{|C_i^{\text{exp}} - C_i^{\text{calc}}|}{C_i^{\text{exp}}} \leq \varepsilon\,, \tag{1}$$

где ε — допустимая максимальная относительная погрешность экспериментальных данных C.

Определим по каждой из констант k_j $(j = 1, \ldots, m)$ интервал неопределенности как некоторый отрезок

$$d_j = [\min k_j, \max k_j]\ , \tag{2}$$

вариация k_j внутри которого сохраняет совместность системы (1).

Постановка задач определения интервалов (2) при условии удовлетворения системы ограничений (1) принадлежит создателю линейного программирования Л. В. Канторовичу и изложена в его докладе Сибирскому математическому обществу в 1962 г. [4].

Существенным является тот факт, что сама эта постановка явилась новым словом в теории математической обработки эксперимента. Классическая постановка задач математической обработки эксперимента восходит к работам Лежандра и Гаусса и состоит в поиске таких значений параметров математического описания, при которых минимизируется какой-либо критерий соответствия расчета измерениям, например сумма квадратов отклонений экспериментально измеренных и рассчитанных по модели величин. Такой способ широко известен как метод наименьших квадратов. В основе этого метода лежит и вероятностная составляющая. В случае если погрешность эксперимента распределена по знаменитому нормальному закону, значения параметров, рассчитанные на основе метода наименьших квадратов, являются в некотором смысле наиболее вероятными.

Вопрос состоит в том, насколько обоснованно принимать гипотезу о нормальности распределения погрешности измерений. Еще в начале XX в. знаменитый математик Пуанкаре остроумно пошутил по этому поводу: «Каждый верит в экспоненциальный закон по-своему: математик — потому что считает, что закон подтверждается наблюдениями, экспериментатор считает, что он следует из математических рассуждений». На самом деле установление закона распределения погрешности — очень трудоемкая и совершенно самостоятельная задача. Она сводится к многократному воспроизводству эксперимента в одних и тех же условиях, что практически нереально для достаточно сложных систем. Нам практически неизвестны реальные кинетические каталитические системы, для которых проводились серьезные исследования по установлению закона распределения погрешности кинетических измерений.

В то же время постановка задачи обработки эксперимента, как она сделана Канторовичем, не требует знания информации о статистических свойствах распределения погрешности измерений. Величины ε_j в системе неравенств (1) есть характеристики предельно допустимой погрешности эксперимента. Информация о величине предельно допустимой погрешности, как правило, присутствует у экспериментатора. И тогда выполнение условий (1) означает, что модель описывает измерения в пределах, обусловленных величиной максимально допустимой погрешности измерений, что совершенно естественно.

Метод Канторовича был развит авторами статьи применительно к решению обратных задач химической кинетики [3, 5, 6] и анализу проблемы выбора оптимальной структуры каталитической системы [7]. В работах [8–12] идея Канторовича используется в приложении к хемометрике.

Отметим одну интересную особенность, выявленную в ходе решения обратных задач химической кинетики — идентификации констант скоростей стадий в многостадийных механизмах сложных реакций. Авторы не раз сталкивались

с ситуацией, когда минимум по части констант оказывался равным нулю. Это означает, что заданный массив измерений можно описать без использования соответствующей константы. Иными словами, информация не позволяет идентифицировать данную стадию. Это не означает, что соответствующая стадия идет или не идет. Это означает только то, что она не обеспечена информацией для своего определения. Заданный массив эксперимента не информативен с точки зрения определения соответствующей константы, а значит, и идентификации соответствующей стадии механизма реакции. Сразу возникает вопрос о планировании специальных измерений, которые позволят определить соответствующую константу с заданным уровнем точности.

Принципиальной особенностью метода Канторовича является и тот факт, что, опираясь на идеи математического программирования, он позволяет использовать при анализе информативности измерений решения сопряженной (или двойственной в терминологии линейного программирования) задачи. Решения сопряженной задачи позволяют выделить из большого массива эксперимента точки, определяющие значения минимума и максимума по каждой из констант. И в случае, если интервал d_j по какой-либо из констант оказывается слишком велик, анализ решения сопряженной задачи позволяет построить план измерений (условия новых экспериментов и их точность) с целью уменьшения интервала до величины, определяемой некоторыми дополнительными требованиями.

Таким образом, интервал неопределенности d_j по параметру k_j, задаваемый (2), есть некоторый отрезок, внутри которого выполняется неравенство (1), т. е. в пределах которого кинетическая модель не противоречит измерениям. Вектор $d = (d_1, \ldots, d_m)$ характеризует степень неопределенности каждого из искомых параметров, вызванную погрешностью измерений. С помощью этого вектора можно определить, какая точность измерений и в каких точках необходима, чтобы степень неопределенности в параметрах не превосходила заданной величины.

Под многомерной областью неопределенности будем понимать множество точек — значения кинетических констант, в каждой из которых при численном моделировании протекания реакции выполняется соотношение (1). Таким образом, каждая точка области неопределенности соответствует единственному набору констант для всех реакций. Решив с данным набором констант прямую кинетическую задачу определения концентраций веществ, находим невязку между рассчитанными концентрациями при моделировании и экспериментальными данными. Эта невязка по определению должна быть меньше допустимой максимальной погрешности ε_j.

Таким образом, если кинетическая модель реакции включает m кинетических констант, область неопределенности будет m-мерной. Будем ставить перед собой задачу нахождения интервалов неопределенности и двумерных проекций области неопределенности на плоскость по паре кинетических констант.

3 Алгоритм нахождения интервалов и областей неопределенности

Существует небольшое число методов определения интервалов неопределенности. Например, метод перебора, который является самым простым, но и самым медленным. Его недостаток — необходимость большого объема вычислений значений функции минимизации. В данной работе рассматривается метод, основанный на идее Л. В. Канторовича.

Ставится задача: для каждой константы найти интервал неопределенности (точнее, его границы). Для определения интервала по константе k_j необходимо найти $\min k_j$ и $\max k_j$ при выполнении ограничения (1).

При поиске происходит одновременное решение прямой кинетической задачи с определенным набором констант и проверка удовлетворения найденных значений концентраций неравенству (1). Если найденные значения концентраций удовлетворяют неравенству, то данный набор констант содержится в искомом интервале неопределенности.

Чтобы найти границу искомого интервала d_j, нужно взять определенный набор констант и зафиксировать все константы, кроме одной, например k_j. Набор констант определяется из решения обратной задачи, т. е. определения какой-нибудь одной точки, удовлетворяющей (1).

Рассматривается следующий алгоритм нахождения интервала неопределенности по константе k_j.

В качестве начального приближения рассматривается какое-нибудь значение констант, удовлетворяющее (1). Такое значение может быть найдено путем минимизации какого-либо критерия соответствия расчета измерениям. Пусть начальная точка $(k_1^0, \ldots, k_m^0)$ найдена. Возьмем начальное значение шага h_0. Для нахождения искомого интервала для j-й константы определим $\max k_j$ (алгоритм нахождения $\min k_j$ тот же самый, только шаг следует взять со знаком минус). Добавив к k_j^0 шаг h_0, получим следующий набор констант: $(k_1^0, \ldots, k_j^0 + h_0, \ldots, k_m^0)$. Теперь с имеющимся набором констант решаем прямую задачу и проверяем совместность неравенства (1). В случае удовлетворения неравенству точка $k_j^0 + h_0$ принадлежит искомому интервалу и можно продолжать двигаться в правую сторону (в случае поиска $\max k_j$). Если точка $k_j^0 + h_0$ не удовлетворяет неравенству (1), то уменьшаем шаг в 2 раза: $h_1 = h_0/2$ — и, добавляя его к k_j^0, получаем новый набор констант $(k_1^0, \ldots, k_j^0 + h_1, \ldots, k_m^0)$. С полученным набором констант решаем прямую задачу и проверяем совместность неравенства (1). Данный процесс продолжаем до тех пор, пока не получится шаг с требуемой точностью. Тем самым определяются границы интервала неопределенности. Аналогичная процедура поиска применяется и для остальных констант скоростей.

Алгоритм нахождения областей неопределенности представлен на рис. 1.

Кинетическая реакция может быть описана системой нелинейных дифференциальных уравнений с заданными начальными условиями. Неизвестными в задаче являются константы скоростей элементарных реакций $(k_1, k_2, \ldots, k_z)$.

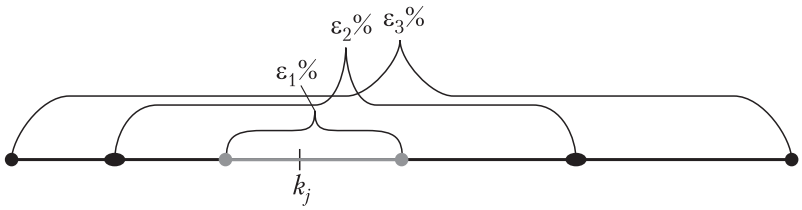


Рис. 1 Поиск интервала неопределенности. Точка k_j соответствует экстремуму целевой функции; ε_1, ε_2 и ε_3 соответствуют относительной погрешности, причем $\varepsilon_1 < \varepsilon_2 < \varepsilon_3$

Задача определения области неопределенности разбивается на две подзадачи.

Первая подзадача. Нахождение хотя бы одного набора констант, удовлетворяющего неравенству (1).

Для этого необходимо решить задачу минимизации критерия соответствия расчетных и экспериментальных данных (1). Результатом решения этой задачи являются значения величин $k_1^0, k_2^0, \ldots, k_z^0$, соответствующие наилучшему (в некотором смысле) приближению к реальным константам. Если же при подстановке этих констант в систему дифференциальных уравнений и нахождении расчетных значений концентраций компонентов они не удовлетворяют системе неравенств, то это может означать, что были взяты слишком маленькие значения погрешности ε_j.

Вторая подзадача. Задача нахождения области была поставлена как задача нахождения проекций многомерной области на двумерную плоскость по паре кинетических констант. Таким образом, область представляет собой множество значений пары констант (k_x, k_y) при фиксированных остальных константах. Для нахождения проекции области неопределенности, каждая точка которой удовлетворяет системе неравенств, используется перебор, организованный следующим образом. После решения первой подзадачи образуется набор констант, удовлетворяющий системе неравенств, а следовательно, принадлежащий искомой области, допустим $(k_1^0, k_2^0, \ldots, k_x^0, \ldots, k_y^0, \ldots, k_z^0)$. Этот набор будет служить начальной точкой поиска. Зафиксируем все константы, кроме k_x и k_y. Теперь зафиксируем одну из этих двух констант, допустим k_x. Далее найдем интервал неопределенности для константы k_y, как это было показано выше. После этого вместо константы k_x возьмем константу $k_x + h$, для которой опять определим интервал неопределенности для k_y. Будем продолжать эту процедуру до тех пор, пока не достигнем правой границы области. Затем для нахождения левой границы повторяем ту же процедуру, только вместо шага h берем $-h$. Таким образом будет получена искомая область неопределенности для пары констант (k_x, k_y).

4 Примеры нахождение интервалов и областей неопределенности

4.1 Поликонденсация аспарагиновой кислоты

В качестве примера можно привести нахождение интервалов и областей неопределенности для реакции поликонденсации аспарагиновой кислоты [13].

Кинетическая схема общей реакции состоит из двух частей, каждая из которых включает три реакции. Реакция поликонденсации проходит независимо в двух зонах матрицы (табл. 1). Здесь A — исходный мономер; B — вода, выделяющаяся во всех происходящих реакциях; C — автокатализирующий промежуточный продукт: димер, тример и т. д.; D — конечный продукт; параметр α отражает долю исходного мономера для зон.

Таблица 1 Кинетическая схема реакции поликонденсации аспарагиновой кислоты

Зона	Химическая реакция	Скорость стадии
Первая	$\alpha A \to B_1 + C_1$	$w_1 = k_1 \alpha A$
	$\alpha A + C_1 \to B_1 + C_1$	$w_2 = k_2 \alpha A C_1$
	$C_1 \to D_1 + B_1$	$w_3 = k_3 C_1$
Вторая	$(1-\alpha)A \to B_2 + C_2$	$w_4 = k_4 (1-\alpha) A$
	$(1-\alpha)A + C_2 \to B_2 + C_2$	$w_5 = k_5 (1-\alpha) A C_2$
	$C_2 \to D_2 + B_2$	$w_6 = k_6 C_2$

Суммарное превращение идет в две стадии. Первая — собственно поликонденсация, которая приводит к росту молекулярной массы — реализуется в виде двух параллельных реакций с выделением воды в качестве побочного продукта: (1) прямого взаимодействия двух молекул мономера и (2) автокаталитического превращения с участием возникающего олигомера. Во второй стадии происходит реакция полимераналогичного превращения получающегося полимера: карбоксильная и имидная группы отщепляют воду с образованием сукцинимидного цикла. Конечным продуктом является полисукцинимид. Его образование обусловлено, таким образом, отщеплением двух молекул воды от молекулы исходного мономера.

Для реакции поликонденсации аспарагиновой кислоты вначале были найдены константы скорости, после чего была выполнена оценка констант. Под оценкой констант скоростей понимается нахождение интервалов неопределенности.

В табл. 2 приведены интервалы неопределенности при температуре 208 °C.

Графическое представление для константы k_3 отражено на рис. 2.

Интервалы неопределенности при температуре опыта 208 °C включают нулевое значение для константы скорости k_3 с относительной погрешностью 2% и 3%.

Таблица 2 Интервалы неопределенности при температуре 208 °C

Константы скорости	Константы $\alpha = 0{,}25$	Интервал неопределенности		
		$\varepsilon = 1\%$	$\varepsilon = 2\%$	$\varepsilon = 3\%$
k_1	0,0045	[0,0037; 0,0055]	[0,00307; 0,0066]	[0,0025; 0,0079]
k_2	0,17	[0,158; 0,185]	[0,144; 0,198]	[0,132; 0,214]
k_3	0,00498	[0,0024; 0,0072]	[0; 0,0098]	[0; 0,0124]
k_4	0,00094	[0,0006; 0,0015]	[0,00038; 0,0019]	[0,0002; 0,0032]
k_5	2,57	[2,5; 2,64]	[2,41; 2,7]	[2,31; 2,77]
k_6	1,69	[1,65; 1,74]	[1,6; 1,81]	[1,56; 1,89]

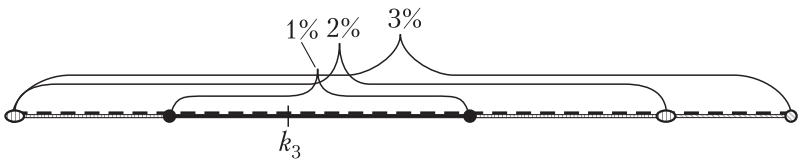


Рис. 2 Интервал неопределенности для константы k_3

Отметим, что из факта равенства некоторых констант нулю не следует отсутствия соответствующей стадии химического превращения. Это свидетельствует о том, что имеющаяся экспериментальная информация не дает возможности математического описания данной стадии. Возникает задача планирования специального эксперимента с целью уменьшения величины интервала неопределенности, выхода минимума на положительное значение. Планирование таких измерений — специальная задача, которая станет предметом дальнейших исследований.

4.2 Гидроалюминирование олефинов

В качестве примера областей неопределенности на рис. 3 приведены области неопределенности для реакции гидроалюминирования олефинов для алюминийорганического соединения типа HAlBu^i_2 (диизобутилалюминийгидрид, сокращенно ДИБАГ) [14]. Кинетическая схема частного случая HAlBu^i_2 (ДИБАГ) имеет следующий вид:

$$A_1 \leftrightarrow 2A_2\,,\ W_1 = k_1X_1 - k_2X_2^2\,;$$
$$A_2 + A_5 \leftrightarrow A_8\,,\quad W_2 = k_6X_2X_5 - k_7X_8\,,$$

где $A_1 = [\mathrm{Cp_2ZrH_2} \cdot \mathrm{ClAlBu}^i_2]_2$, $A_2 = [\mathrm{Cp_2ZrH_2} \cdot \mathrm{ClAlBu}^i_2]$, $A_5 = \mathrm{HAlBu}^i_2$, $A_8 = [\mathrm{Cp_2ZrH_2} \cdot \mathrm{HAlBu}^i_2 \cdot \mathrm{ClAlBu}^i_2]$.

В [7] был сделан вывод о том, что если сравнивать скорости прямой и обратной реакции образования мономера A_2 из комплекса A_1, то равновесие смещено в сто-

рону димерной формы. Самой быстрой оказалась стадия перехода мономера A_2 в неактивный тригидридный комплекс A_8, а равновесие сильно смещено в сторону тригидридного комплекса: было получено соотношение $k_2 > k_1$, $k_6 > k_7$. Этот вывод был сделан на основе численных значений констант, минимизирующих критерий соответствия расчета измерению. Из рис. 3 видно, что область неопределенности находится внутри приведенных соотношений.

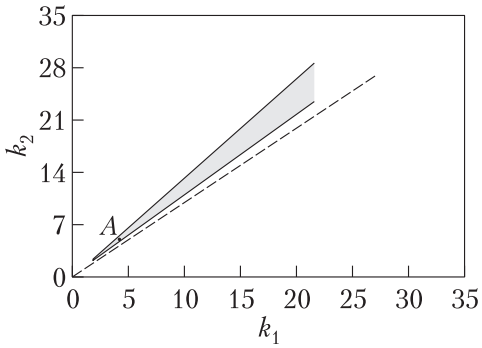


Рис. 3 Область неопределенности для констант k_1 и k_2 в случае ДИБАГ при $t = -40$ °C. Точкой A обозначена начальная точка поиска области

5 Заключение

Метод Канторовича представляет собой принципиально новый подход в задачах математической интерпретации измерений. Показана его перспективность при решении обратных задач химической кинетики.

Создано компьютерное обеспечение расчета интервалов неопределенности и областей неопределенности в двухмерном случае.

Основная проблема при его использовании — расчет многомерных областей неопределенности. Возникающие задачи носят как математический, так и физико-химический характер. В частности, главной становится задача физико-химической интерпретации областей неопределенности. Эти проблемы станут предметом дальнейших исследований в этом направлении.

Литература

1. *Gorskii V. G., Spivak S. I.* Analysis identifiability of parameters in mathematical models of chemical kinetics and thermodynamics // AMSR Trans. Math. Models Tools Chem. Kinetics Ser. A, 1993. Vol. 9. P. 125–148.
2. *Spivak S. I.* Inverse problems of chemical kinetics and thermodynamics // Syst. Anal. Model. Sim., 1995. Vol. 18-19. P. 107–110.
3. *Spivak S. I.* Informativity of experiments and uncertainty regions of model parameters // 15th GAMM–IMACS Symposium (International) on Scientific Computing, Computer Arithmetic and Verified Numerics. — Novosibirsk, 2012. P. 172. http://conf.nsc.ru/files/conferences/scan2012/140000/scan2012Abstracts.pdf.
4. *Канторович Л. В.* О новых подходах в теории обработки наблюдений // Сиб. мат. ж., 1962. Т. 3. С. 701–708.
5. *Spivak S. I., Slinko M. G., Timoshenko V. I., Mashkin V. Yu.* Interval estimation in the determination of parameters of a kinetic model // React. Kinet. Catal. L., 1974. Vol. 3. No. 1. P. 105–113.

6. *Akhunov I. R., Spivak S. I.* Discrimination of the mechanisms of cooxidation of arylalkenes and akkylamines // React. Kinet. Catal. L., 1984. Vol. 24. No. 3-4. P. 305–308.
7. *Спивак С. И.* Информативность кинетических измерений // Химическая промышленность сегодня, 2009. № 9. С. 52–56.
8. *Meinrath G.* Computer-intensive methods for uncertainty estimation in complex situations // Chemometr. Intell. Lab., 2000. Vol. 51. No. 2. P. 175–187.
9. *Rodionova O. Ye., Esbensen K. H., Pomerantsev A. L.* Application of SIC (Simple Interval Calculation) for object status classification and outlier detection — comparison with PLS/PCR // J. Chemometr., 2004. Vol. 18. P. 402–413.
10. *Pomerantsev A. L., Rodionova O. Ye.* Hard and soft methods for prediction of antioxidants' activity based on the DSC measurements // Chemometr. Intell. Lab., 2005. Vol. 79. No. 1-2. P. 73–83.
11. *Rodionova O. Y., Pomerantsev A. L.* Subset selection strategy // J. Chemometr., 2008. Vol. 22. P. 674–685.
12. *Rodionova O. Ye., Pomerantsev A. L.* Simple view on Simple Interval Calculation (SIC) method // Chemometr. Intell. Lab., 2009. Vol. 97. No. 1. P. 64–76.
13. *Badrtdinova F. T., Spivak S. I., Goldberg V. M., Bigaeva L. A.* Kinetic model of the process of asparaginic acid polycondensation according to termogravimetric analysis // Пластические массы, 2011. № 7. С. 37–40.
14. Parfenova L. V., Balaev A. V., Gubaidullin I. M., Pechatkina S. V., Abzalilova L. R., Spivak S. I., Khalilov L. M., Dzhemilev U. M. Kinetic model of olefins hydroalumination by $HAlBu^i_2$ and $AlBu^i_3$ in presence of Cp_2ZrCl_2 catalyst // Int. J. Chem. Kinet., 2007. Vol. 39. No. 6. P. 333–339.

Поступила в редакцию 31.08.17

INFORMATIVITY OF A KINETIC EXPERIMENT AND UNCERTAINTY REGIONS OF KINETIC MODEL

***S. I. Spivak*[1,2], *F. T. Ziganshina*[3], *and A. S. Ismagilova*[1]**

[1]Bashkir State University, 32 Validy Str., Ufa 450076, Russian Federation
[2]Institute of Petrochemistry and Catalysis, Russian Academy of Sciences, 141 Oktyabrya Ave., Ufa 450075, Russian Federation
[3]Ufa State Petroleum Technical University, 1 Kosmonavtov Str., Ufa 450062, Republic of Bashkortostan, Russian Federation

Abstract: In the process of solution of inverse problems of chemical kinetics, one often faces the ambiguity of definition of certain kinetic parameters. The ambiguity of the inverse problem is that the description of any mathematical model must reflect some chemical reactions in a certain area of input parameters. In view of the ambiguity of decision, the methods of estimating kinetic parameters are used. One of these methods is the method for determining the ranges and areas of uncertainty, which is based on the idea of L. V. Kantorovich.

Keywords: inverse problems; chemical kinetics; informativeness of the experiment; region of uncertainty

DOI: 10.14357/08696527180106

Acknowledgments

The work was supported by the Russian Foundation for Basic Research and by the Government of the Republic of Bashkortostan (project 17-47-020068).

References

1. Gorskii, V. G., and S. I. Spivak. 1993. Analysis identifiability of parameters in mathematical models of chemical kinetics and thermodynamics. *AMSR Trans. Math. Models Tools Chem. Kinetics Ser. A* 9:125–148.
2. Spivak, S. I. 1995. Inverse problems of chemical kinetics and thermodynamics. *Syst. Anal. Model. Sim.* 18-19: 107–110.
3. Spivak, S. I. 2012. Informativity of experiments and uncertainty regions of model parameters. *15th GAMM–IMACS Symposium (International) on Scientific Computing, Computer Arithmetics and Verified Numerics*. Novosibirsk. 172. Available at: http://conf.nsc.ru/ files/conferences/scan2012/140000/scan2012Abstracts.pdf (accessed March 30, 2018).
4. Kantorovich, L. V. 1962. O novykh podkhodakh v teorii obrabotki nablyudeniy [On new approaches in the theory of processing observations]. *Siberian Math. J.* 3:701–708.
5. Spivak, S. I., M. G. Slinko, V. I. Timoshenko, and V. Yu. Mashkin. 1974. Interval estimation in the determination of parameters of a kinetic model. *React. Kinet. Catal. L.* 3(1):105–113.
6. Akhunov, I. R., and S. I. Spivak. 1984. Discrimination of the mechanisms of cooxidation of arylalkenes and akkylamines. *React. Kinet. Catal. L.* 24(3-4):305–308.
7. Spivak, S. I. 2009. Informativnost' kineticheskikh izmereniy [Informativity of kinetic measurements]. *Khimicheskaya promyshlennost' segodnya* [Chemical Industry Today] 9:52–56.
8. Meinrath, G. 2000. Computer-intensive methods for uncertainty estimation in complex situations. *Chemometr. Intell. Lab.* 51(2):175–187.
9. Rodionova, O. Ye., K. H. Esbensen, and A. L. Pomerantsev. 2004. Application of SIC (Simple Interval Calculation) for object status classification and outlier detection — comparison with PLS/PCR. *J. Chemometr.* (18):402–413.
10. Pomerantsev, A. L., and O. Ye. Rodionova. 2005. Hard and soft methods for prediction of antioxidants' activity based on the DSC measurements. *Chemometr. Intell. Lab.* 79(1-2):73–83.
11. Rodionova, O. Y., and A. L. Pomerantsev. 2008. Subset selection strategy. *J. Chemometr.* (22):674–685.
12. Rodionova, O. Ye., and A. L. Pomerantsev. 2009. Simple view on Simple Interval Calculation (SIC) method. *Chemometr. Intell. Lab.* 97(1):64–76.
13. Badrtdinova, F. T., S. I. Spivak, V. M. Goldberg, and L. A. Bigaeva. 2011. Kinetic model of the process of asparaginic acid polycondensation according to termogravimetric analysis. *Plasticheskie massy* [Plastic Masses] 7:37–40.

14. Parfenova, L. V., A. V. Balaev, I. M. Gubaidullin, S. V. Pechatkina, L. R. Abzalilova, S. I. Spivak, L. M. Khalilov, and U. M. Dzhemilev. 2007. Kinetic model of olefins hydroalumination by $HAlBu^i_2$ and $AlBu^i_3$ in presence of Cp_2ZrCl_2 catalyst. *Int. J. Chem. Kinet.* 39(6):333–339.

Received August 31, 2017

Contributors

Spivak Semen I. (b. 1945) — Doctor of Science in physics and mathematics, professor, Head of Department, Bashkir State University, 32 Validy Str., Ufa 450076, Russian Federation; Head of Laboratory, Institute of Petrochemistry and Catalysis, Russian Academy of Sciences, 141 Oktyabrya Ave., Ufa 450075, Russian Federation; semen.spivak@mail.ru

Ziganshina Fayruza T. (b. 1985) — Candidate of Science (PhD) in physics and mathematics, associate professor, Ufa State Petroleum Technical University, 1 Kosmonavtov Str., Ufa 450062, Republic of Bashkortostan, Russian Federation; fairusa85@mail.ru

Ismagilova Albina S. (b. 1979) — Doctor of Science in physics and mathematics, assistant professor, Department of Information Security Management, Bashkir State University, 32 Zaki Validi Str., Ufa 450076, Republic of Bashkortostan, Russian Federation; ismagilovaas@rambler.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 89–98

НЕКОТОРЫЕ СИСТЕМОТЕХНИЧЕСКИЕ ВОПРОСЫ ИСПОЛЬЗОВАНИЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ ДЛЯ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ИНФОРМАЦИИ В СИТУАЦИОННЫХ ЦЕНТРАХ*

***В. Е. Гаврилов*[1], *А. А. Зацаринный*[2]**

Аннотация: Статья посвящена вопросам использования интеллектуального анализа данных (ИАД) для обеспечения защиты информации в ситуационных центрах (СЦ). Проведен анализ основных проблем информационной безопасности в облачных вычислительных системах. Рассмотрены основные компоненты системы защиты информации, в которых в настоящее время в той или иной мере используются технологии искусственного интеллекта. Даны предложения по расширению области применения этих технологий в целях защиты информации. Проанализирована существующая нормативная база по информационной безопасности в облачных вычислительных системах.

Ключевые слова: ситуационные центры; автоматизированные системы; облачные вычислительные системы; информационная безопасность; угрозы информационной безопасности; функциональная безопасность; искусственный интеллект; машинное обучение

DOI: 10.14357/08696527180107

1 Введение

Современная система государственного управления, а также управления крупными коммерческими структурами предполагает в процессе принятия решений аналитическую обработку больших объемов неструктурированных данных. Ключевым элементом системы поддержки принятия решений является СЦ, обеспечивающий «автоматизацию процессов мониторинга и ситуационного анализа обстановки в контролируемом информационном пространстве, включая сбор, обобщение, аналитическую обработку, хранение, передачу, визуализацию и защиту информации» [1]. Эффективность информационно-аналитической поддержки системы управления повышается при интеграции ресурсов СЦ федеральных или региональных органов власти, структурных подразделений крупной компании. В сфере государственного управления задача создания взаимоувя-

*Работа выполнена при частичной поддержке РФФИ (проект 15-29-07981 офи-м).

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vegavrilov@yandex.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, AZatsarinny@ipiran.ru

занной системы распределенных СЦ (СРСЦ) на основе объединения ресурсов существующих и создания новых СЦ Указом Президента РФ от 25 июля 2017 г. определена как важнейшая государственная задача стратегической значимости.

Во исполнение поручения Президента РФ межведомственной рабочей группой разработаны Концепция создания СРСЦ и Концепция информационной безопасности СРСЦ. Ряд основополагающих документов, приведенных в [1], определяют организационные и системотехнические решения, рекомендуемые при использовании СРСЦ. При этом взаимодействие разнородных СЦ может обеспечиваться на основе технологии облачных вычислительных систем с помощью типовых комплексов информационного взаимодействия, включающих витрины данных. Этот подход позволяет использовать унаследованные разнородные СЦ в составе Системы и масштабировать ее, не затрагивая функционирующие сегменты.

При реализации такого подхода центральное место занимают проблемы обеспечения информационной безопасности.

2 Основные проблемы обеспечения информационной безопасности в системе распределенных ситуационных центров

Безусловно, эффективное функционирование СЦ возможно только при условии обеспечения надежной защиты информации. В соответствии с [2] автоматизированная система в защищенном исполнении определена как «автоматизированная система, реализующая информационную технологию выполнения установленных функций в соответствии с требованиями стандартов и/или нормативных документов по защите информации». Вместе с тем с точки зрения обеспечения безопасности информации современный СЦ обладает рядом специфических особенностей, затрудняющих построение традиционной системы защиты информации, основанной на взаимодействии поименованных субъектов и объектов в замкнутой доверенной среде в соответствии с установленными правилами и требованиями. Рассмотрим некоторые из этих аспектов.

Одним из ключевых элементов системы защиты информации является подсистема контроля целостности и функциональной замкнутости доверенной программной среды. Однако современный СЦ получает и ведет аналитическую обработку больших объемов информации, полученной с использованием технологий облачных вычислений от внешних, возможно недоверенных, источников, зачастую представленных в форматах, содержащих исполняемый код, что нарушает принцип целостности программной среды.

Использование технологии витрин данных подменяет прямое взаимодействие участников системы СЦ взаимодействием типовых комплексов информационного взаимодействия, построенных по единой технологии и поддерживающих единую политику безопасности. Информационное наполнение витрин данных осуществляется в соответствии с заданным регламентом и включает:

(1) первичное наполнение;

(2) пополнение по времени с заданной периодичностью;

(3) пополнение по событиям из заданного списка;

(4) пополнение по запросам других участников системы СЦ.

Реализация первых трех пунктов не вызывает затруднений и может осуществляться с использованием технологий существующих СЦ. Неформализованный запрос, сформированный в иной программно-технической среде другого участника системы СЦ, поддерживающей иную политику безопасности, порождает новый канал для проведения компьютерных атак и может нести потенциальную угрозу.

Аналитическая обработка информации зачастую производится с использованием технологий ИАД, самостоятельно порождающих исполняемый код, что также нарушает принцип функционирования автоматизированной системы в замкнутой доверенной среде.

Централизованное администрирование системы защиты информации комплексов информационного взаимодействия может вызывать определенные трудности, связанные как с территориальной распределенностью, так и с масштабом системы СЦ.

3 Проактивные методы защиты

С учетом приведенных выше ограничений по применению традиционных средств защиты информации в отдельных сегментах подсистемы защиты информации системы СЦ представляется целесообразным применение проактивных методов [3, 4].

В настоящее время такие методы широко используются в антивирусных средствах и средствах обнаружения компьютерных атак. Признаки компьютерной атаки или наличия угроз вирусного заражения формируются на основе экспертных оценок соответствующих специалистов. При этом проактивные системы антивирусной защиты и защиты от компьютерных атак могут рассматриваться как экспертные системы, что является лишь первым шагом к широкомасштабному использованию технологий ИАД.

Дальнейшее развитие проактивных средств защиты информации видится на пути внедрения методов машинного обучения [5] и их дальнейшего развития — технологий «глубокого самообучения». При этом система на основе анализа больших объемов информации с учетом первоначально сформированного экспертным сообществом набора критериев наличия потенциально опасных конструкций самостоятельно дополняет и уточняет эти критерии. На первом этапе, в частности, такая система могла бы оказать существенную поддержку в деятельности администратора безопасности, сопоставляя и анализируя возможные последствия различных комбинаций большого числа событий, что может вызывать затруднения при экспертном анализе.

Еще одной точкой приложения проактивных методов защиты могут стать межсетевые экраны. В настоящее время эти устройства, как правило, фильтруют трафик по сетевым адресам, протоколам, форматам сообщений и т. п., иногда также осуществляется контентный анализ пропускаемой информации по ключевым словам. Технологии ИАД позволяют повысить эффективность контентного анализа:

- выявляя использование эвфемизмов, что не способен обеспечить поиск по ключевым словам;
- осуществляя структурный анализ как отдельных текстов, так и их совокупности, выявляя попытки выноса защищаемой информации по частям в текстах легальных сообщений;
- проводя статистический анализ трафика с целью выявления использования стеганографических методов передачи информации.

Еще одной задачей, для решения которой могут использоваться технологии ИАД, может стать выявление скрытых логических каналов утечки защищаемой информации, возникающих за счет манипулирования параметрами передачи с использованием недекларированных возможностей программно-аппаратной среды.

Как и в случае с антивирусной защитой и защитой от компьютерных атак, основная проблема создания подобных шлюзов заключается в выработке необходимых критериев потенциально опасных событий.

Заметим, что все перечисленные направления применения технологий ИАД не подменяют традиционных средств и технологий защиты информации при всей их неполноте, а могут и должны использоваться наряду с ними.

4 Функциональная безопасность

Сложившийся в настоящее время подход к обеспечению безопасности информации в автоматизированных системах, как и приведенное выше определение автоматизированной системы в защищенном исполнении, ориентирован на сохранение основных характеристик безопасности информации: целостности, доступности и, в необходимых случаях, конфиденциальности. Вместе с тем для владельца автоматизированной системы в ряде случаев приоритетным является эффективное выполнение заданных функциональных задач. В этом случае говорят о функциональной безопасности автоматизированной системы как способности корректно и эффективно выполнять заданные функциональные задачи, в том числе в условиях возможного противодействия потенциального нарушителя [6]. Чаще всего проблему функциональной безопасности поднимают в отношении автоматизированных систем управления потенциально опасными технологическими процессами в промышленности, на транспорте, в энергетике и т. п. При этом нарушитель путем воздействия на используемые в системе управления информационные технологии пытается создать предпосылки наступления

негативных технологических последствий, а порой и техногенной катастрофы. Атакующие воздействия могут затрагивать первичную информацию, собираемую с различных датчиков контрольных устройств, алгоритмы обработки этой информации, критерии принятия управляющих решений и команды управления исполнительными устройствами. Вопросы физической безопасности автоматизированных систем управления как объектов информатизации являются предметом отдельного рассмотрения и выходят за рамки настоящей статьи. Однако, по мнению авторов, понятие функциональной безопасности актуально и для автоматизированных систем иного назначения, в частности СЦ. При этом нарушитель, как уже отмечалось выше, путем воздействия на используемые информационные технологии пытается снизить эффективность выполнения основной функциональной задачи СЦ — информационной поддержки принятия управленческих решений уполномоченным должностным лицом. Атакующие воздействия потенциального нарушителя могут затрагивать процессы отбора первичной информации, алгоритмы ее аналитической обработки или критерии принятия решений (отбора значимых событий).

Далее рассмотрим некоторые проблемы обеспечения информационной безопасности в широком смысле слова, включающие как традиционные аспекты защиты информации, так и вопросы функциональной безопасности, при выполнении основных функциональных задач системы СЦ и возможности использования для их разрешения методов ИАД.

Функционально система СЦ должна обеспечивать весь цикл управления:

- мониторинг текущей обстановки;
- первичный анализ обстановки;
- поддержку процессов принятия решений;
- доведение и контроль исполнения [1].

Мониторинг текущей обстановки включает сбор, обобщение и систематизацию информации по заданному кругу вопросов.

Сбор информации осуществляется от разнородных источников, зачастую не содержащих каких-либо средств защиты информации, по незащищенным каналам связи. Форматы поступающих данных могут содержать исполняемый код. Минимизация возникающих при этом угроз достигается путем фильтрации входящего трафика с использованием антивирусных средств и средств обнаружения компьютерных атак. Применяемые в настоящее время соответствующие средства все чаще наряду с сигнатурными методами выявления потенциальных угроз используют так называемые проактивные методы, использующие, в свою очередь, технологии ИАД, близкие по сути к экспертным системам. На ранних этапах эффективность проактивных методов была невысока, доля выявляемых вирусных атак, коды которых не содержались в базе данных сигнатур, не превышала 5%–7%. К настоящему времени тестирование антивирусных средств ведущих производителей показывает приближающуюся к 100% эффективность в данных условиях [7].

Весьма актуальной на этапе *обобщения и систематизации информации* является проблема отбраковки «фейковой» информации. Ведущие мировые разработчики поисковых систем прилагают большие усилия к решению этой проблемы, используя технологии ИАД, в том числе методы «глубокого обучения» [8–11]. При этом наряду с отсутствием ясных критериев отбраковки информации возникает еще и проблема создания/отсутствия эталонного массива информации, на котором самообучаемая интеллектуальная система могла бы выработать эти критерии.

На этапе *первичного анализа обстановки* выявляются значимые события и формируется необходимая отчетность соответствующему должностному лицу. С точки зрения информационной безопасности на этом этапе актуальной является традиционная задача контроля целостности программного обеспечения, реализующего типовые алгоритмы первичного реагирования, и их верификация на этапе разработки СЦ на предмет логической непротиворечивости реализуемых сценариев и корректности функционирования, в том числе для любых потоков данных от внешних источников. Указанная задача, учитывая ее сложность, также может решаться с использованием технологии ИАД. Вместе с тем критерии «значимости» событий поддаются формализации лишь в простейших случаях в проблемно-ориентированных СЦ. Для более сложных «универсальных» СЦ при формировании таких критериев также может использоваться технология ИАД.

На этапе *поддержки процессов принятия решений* осуществляются:

- подготовка возможных вариантов развития значимых событий;
- подготовка вариантов решений по типовым значимым ситуациям;
- расчет вариантов реагирования на значимые события;
- подготовка проектов необходимых распорядительных документов;
- подготовка вариантов решений по совершенствованию деятельности в сфере, связанной с выявленными значимыми событиями.

Перечисленные функциональные задачи на этом этапе в той или иной мере связаны с методами моделирования ситуации, в которых уже широко используются технологии ИАД, в том числе и человеко-машинные комплексы. С точки зрения информационной безопасности в части традиционных методов защиты информации актуальны те же задачи, что и на этапе первичного анализа обстановки, а в части функциональной безопасности — корректность классификации возможных значимых событий и, как следствие, выбор адекватного типового сценария развития событий. Именно вмешательство в этот процесс и может стать целью атакующей стороны. Попытки решения этой задачи на основе предметных классификаторов (система практически разрушена или требует серьезной актуализации) вряд ли могут привести к успеху. Вместе с тем использование технологии ИАД, основанной на контекстном анализе всего массива информации, может оказаться более эффективным.

На этапе *доведения и контроля исполнения* задачи защиты информации решаются традиционными средствами с использованием при необходимости каналов шифрованной связи и электронной подписи. С точки зрения функциональной безопасности могут быть полезными методы ИАД, например для выявления недобросовестных отчетов исполнения на основе анализа и сопоставления информации, получаемой из различных источников.

5 Заключение

Отметим некоторые проблемы, которые должны быть преодолены при создании средств защиты информации с использованием технологии ИАД и, в частности, технологий «глубокого обучения». Очевидно, что эффективность самообучаемой системы защиты информации зависит как от точности первоначальных критериев выявления потенциально опасных событий (если они определены), так и от качества обучающей выборки данных. Действительно, если статистические свойства обучающей выборки сильно отличаются от статистических свойств защищаемого информационного пространства, вряд ли можно ожидать эффективной работы системы защиты на основе ИАД. Кроме того, следует иметь в виду, что потенциальный нарушитель также может использовать методы ИАД для подготовки и проведения атак на защищаемую систему. По сути, это уже происходит в широких масштабах в области функциональной безопасности путем размещения тенденциозной, искаженной информации, создания двойников сайтов, создания ботов, формирующих ложные отзывы о качестве товаров и услуг, и т. п. Методы ИАД могут также использоваться нарушителем для формирования искаженных критериев выявления потенциально опасных событий в сети.

Таким образом, в информационном пространстве информационное противоборство выходит на новый уровень с использованием самых современных информационных технологий.

В целом в информационном пространстве назрела необходимость перехода от объектового принципа защиты информации с элементами сетевой защиты при взаимодействии объектов информатизации к комплексной защите всего информационного пространства, основанной на тех же принципах повсеместности и непрерывности на всех этапах жизненного цикла. Информационные технологии настолько глубоко проникли во все сферы деятельности человека, что можно уже говорить о появлении своеобразного цифрового образа окружающего мира — «диджисферы» в составе ноосферы. Система обеспечения безопасности в ней должна строиться на тех же принципах, что и в материальном мире. Объектовая защита — пропускной режим, кадровый отбор персонала, правила внутреннего распорядка и контроль их исполнения службой безопасности и т. п. — должны сочетаться с повсеместной контрразведывательной и оперативно-розыскной работой, деятельностью следственных органов. По сути, первым шагом в этом направлении и стало создание на основе [12] государственной системы обнаруже-

ния и предупреждения компьютерных атак и аналогичных систем в финансовом секторе.

Литература

1. *Зацаринный А. А., Сучков А. П.* Системы ситуационных центров специального назначения. Основные определения, понятия и подходы к созданию // Межотраслевая информационная служба, 2015. № 4. С. 31–41.
2. ГОСТ Р 51624-2000. Защита информации. Автоматизированные системы в защищенном исполнении.
3. *Гаврилов В. Е., Зацаринный А. А.* Некоторые системотехнические и нормативно-методические вопросы обеспечения защиты информации в автоматизированных информационных системах на облачных технологиях с использованием методов искусственного интеллекта // Системы и средства информатики, 2016. Т. 26. № 4. С. 38–50.
4. *Грушо А. А., Забежайло М. И., Зацаринный А. А., Писковский В. О.* О некоторых методах и технологиях искусственного интеллекта, используемых при защите облачных вычислений // Научно-техническая информация. Сер. 2: Информационные процессы и системы, 2017. № 3. С. 1–15.
5. *Осипов Г. С.* Методы искусственного интеллекта. — М.: Физматлит, 2015. 296 с.
6. *Скляр В.* Функциональная безопасность. Ч. 5. Процессы управления и оценивания функциональной безопасности // Securitylab.ru, 22.06.2017. https://www.securitylab.ru/analytics/486866.php.
7. Топ 11 лучших антивирусов 2017 года на компьютер/ноутбук // Akmartis.ru, 18.12.2017. http://akmartis.ru/programmy/rejting-luchshix-antivirusnyx-programm-2017-goda.html.
8. *Yao Y., Viswanath B., Cryan J., Zheng H., Zhao B.* Automated crowdturfing attacks and defenses in online review systems. http://people.cs.uchicago.edu/~ravenben/publications/pdf/crowdturf-ccs17.pdf.
9. 13 онлайн-инструментов для проверки подлинности фотографий. stopfake.org, 29.07.2014. https://www.stopfake.org/13-onlajn-instrumentov-dlya-proverki-podlinnosti-fotografij.
10. *Болецкая К., Брызгалова Е.* Facebook будет использовать искусственный интеллект для выявления фейков // Ведомости, 02.12.2016. https://www.vedomosti.ru/technology/articles/2016/12/02/667876-facebook-iskusstvennii-intellekt.
11. *Смирнов В.* Возможно, вы искали правду: Google ввел глобальную систему проверки новостей // RT на русском, 09.04.2017. https://russian.rt.com/world/article/377045-google-sistema-proverka-fakty.
12. О создании государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации: Указ Президента РФ № 31с от 15 января 2013 г.

Поступила в редакцию 05.12.17

REGARDING SYSTEMIC AND TECHNICAL PROBLEMS OF APPLYING INTELLECTUAL DATA ANALYSIS FOR PROVIDING INFORMATION PROTECTION IN SITUATIONAL CENTERS

V. E. Gavrilov and A. A. Zatsarinny

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article is dedicated to problems of applying intellectual data analysis for providing information protection in situational centers. The authors present the analysis of general problems of information security in cloud computational systems. The authors consider general components of information protection systems which, at present, apply artificial intelligence technologies to some extent. The authors suggest propositions for extending the application area of the technologies to protect the information. The current regulatory base of information security in cloud computational systems is analyzed.

Keywords: situational center; automated systems; cloud computational systems; informational security; data security threats; functional security; artificial intelligence; machine learning

DOI: 10.14357/08696527180107

Acknowledgments

The work was partially supported by the Russian Foundation for Basic Research (project 15-29-07981 ofi-m).

References

1. Zatsarinnyy, A. A., and A. P. Suchkov. 2015. Sistemy situatsionnykh tsentrov spetsial'nogo naznacheniya. Osnovnye opredeleniya, ponyatiya i podkhody k sozdaniyu [Special purposes situational center systems. General definitions, concepts and creation approaches]. *Mezhotraslevaya informatsionnaya sluzhba* [Intersectoral Informational Office] 4:31–41.
2. GOST R 51624-2000. Zashchita informatsii. Avtomatizirovannye sistemy v zashchishchennom ispolnenii [Data security. Secured automated systems].
3. Gavrilov, V. E., and A. A. Zatsarinny. 2016. Nekotorye sistemotekhnicheskie i normativno-metodicheskie voprosy obespecheniya zashchity informatsii v avtomatizirovannykh informatsionnykh sistemakh na oblachnykh tekhnologiyakh s ispol'zovaniem metodov iskusstvennogo intellekta [On system-technical and regulatory-methodological problems of data security in cloud automated information systems using artificial intelligence technologies]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(4):38–50.
4. Grusho, A. A., M. I. Zabezhaylo, A. A. Zatsarinnyy, and V. O. Piskovskiy. 2017. O nekotorykh metodakh i tekhnologiyakh iskusstvennogo intellekta, ispol'zuemykh pri zashchite oblachnykh vychisleniy [Regarding some methods and technologies of artificial

intelligence, which are used in cloud computation security]. *Nauchno-tekhnicheskaya informatsiya. Ser. 2: Informatsionnye protsessy i sistemy* [Science and Technical Information. Vol. 2: Information Processes and Systems] 3:1–15.

5. Osipov, G. S. 2015. *Metody iskusstvennogo intellekta* [Artificial intelligence methods]. Moscow: Fizmatlit. 296 p.
6. Sklyar, V. 2017. Funktsional'naya bezopasnost' [Functional security]. Available at: http://www.securitylab.ru/analytics/486866.php (accessed December 18, 2017).
7. Top 11 luchshikh antivirusov 2017 goda na komp'yuter/noutbuk. 2017. [Top 11 of best antivirus soft of 2017 for PC/laptops]. Available at: http://akmartis.ru/programmy/rejting-luchshix-antivirusnyx-programm-2017-goda.html (accessed November 29, 2017).
8. Yao, Y., B. Viswanath, J. Cryan, H. Zheng, and B. Zhao. 2017. Automated crowdturfing attacks and defenses in online review systems. Available at: http://people.cs. uchicago.edu/~ravenben/publications/pdf/crowdturf-ccs17.pdf (accessed December 2, 2017).
9. Stopfake.org. 2014. 13 onlayn-instrumentov dlya proverki podlinnosti fotografiy [The 13 online tools for authenticating photos]. Available at: https://www.stopfake.org/13-onlajn-instrumentov-dlya-proverki-podlinnosti-fotografij/ (accessed December 2, 2017).
10. Boletskaya, K., and E. Bryzgalova. 02.12.2016. Facebook budet ispol'zovat' iskusstvennyy intellekt dlya vyyavleniya feykov [Facebook will use artificial intelligence for fake identification]. *Vedomosti*. Available at: https://www.vedomosti.ru/technology/articles/2016/12/02/667876-facebook-iskusstvennii-intellekt (accessed December 2, 2017).
11. Smirnov, V. 09.04.2017. Vozmozhno, vy iskali pravdu: Google vvel global'nuyu sistemu proverki novostey [Probably, you were looking for a truth. Google applies global system of news verification].*RT in Russian*. Available at: https://russian.rt.com/world/article/377045-google-sistema-proverka-fakty (accessed December 2, 2017).
12. Ukaz Prezidenta No. 31s ot 15 yanvarya 2013 g. "O sozdanii gosudarstvennoy sistemy obnaruzheniya, preduprezhdeniya i likvidatsii posledstviy komp'yuternykh atak na informatsionnye resursy Rossiyskoy Federatsii" [Decree of the President of Russia #31s dated January 15, 2013 "On the establishment of a state system for detecting, preventing, and eliminating the consequences of computer attacks on the information resources of the Russian Federation"].

Received December 5, 2017

Contributors

Gavrilov Victor E. (b. 1950) — senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation, Moscow 119333, Russian Federation; vegavrilov@yandex.ru

Zatsarinny Alexander A. (b. 1951) — Doctor of Science in technology, professor, Deputy Director, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; azatsarinny@ipiran.ru

ОБ АНАЛИЗЕ ОШИБОЧНЫХ СОСТОЯНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ*

***А. А. Грушо*[1], *М. И. Забежайло*[2], *А. А. Зацаринный*[3], *А. В. Николаев*[4], *В. О. Писковский*[5], *В. В. Сенчило*[6], *И. В. Судариков*[7], *Е. Е. Тимонина*[8]**

Аннотация: Приведен анализ ошибочных состояний распределенных вычислительных систем на примере OpenStack, вызванных отсутствием синхронизации между компонентами облачной вычислительной среды (ОВС), а также рассмотрены вероятные источники возникновения рассинхронизации. Представлен разбор журнальных записей при анализе одного из перечисленных ошибочных состояний программной платформы OpenStack. Показан метод анализа подобных ситуаций. Предложено построение и поддержание в актуальном состоянии информационной модели управляемой ОВС. Обоснована необходимость использования такой модели для выполнения требований надежного управления ОВС, предоставления качественных облачных услуг типа сервис, платформа, инфраструктура, сетевые функции, цепочки сетевых функций.

Ключевые слова: распределенные вычислительные системы; облачные вычислительные среды; OpenStack; RabbitMQ; Nova; Neutron; анализ ошибочных состояний; информационные облачные сетевые инфраструктуры

DOI: 10.14357/08696527180108

1 Введение

Некоторые свойства инфраструктуры распределенных ОВС порождают встречающиеся при их эксплуатации ошибочные состояния [1].

*Работа выполнена при частичной поддержке РФФИ (проект 15-29-07981 офи-м).

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, grusho@yandex.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, m.zabezhailo@yandex.ru

[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, AZatsarinny@ipiran.ru

[4]Институт химической физики им. Н. Н. Семёнова Российской академии наук, gentoorion@mail.ru

[5]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vpvp80@yandex.ru

[6]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, volodias@mail.ru

[7]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, 4seev3@gmail.com

[8]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, eltimon@yandex.ru

В [1] обосновано, что одним из факторов, инициирующих сетевые сбои и ошибки, является неизбежная рассинхронизация компонентов сети. Рассмотрим, как влияет физическое удаление инфраструктурных компонентов друг от друга на теоретическую возможность существования уязвимости в инфраструктуре, вызванную естественной разницей системных часов, принадлежащих разным сетевым компонентам. Для простоты будем считать тактовую частоту обобщенного устройства равной 3 ГГц, что соответствует для оптоволокна характерному расстоянию $l = 7$ см (показатель преломления принят равным 1,5). Для уверенной синхронизации взаимодействующие устройства должны находиться друг от друга на расстояниях, сравнимых с l. В противном случае при реконфигурации сети необходимо предусматривать дополнительные средства, обеспечивающие целостность и непротиворечивость работы сети. Например, можно использовать методы, описанные в работе [1].

Возможно использование результатов работы [2] и деление сети на так называемые слои виртуальных подсетей. В подсетях возможно независимое, полное и непротиворечивое конфигурирование и функционирование виртуальных машин (ВМ) и сетевых компонентов, принадлежащих слою. Отметим, что пока удается найти только достаточные условия непротиворечивости работы сети.

Противоречия конфигураций приводят к появлению уязвимостей в течение временно́го промежутка, который определяется временны́ми интервалами прохождения пакета между сетевыми компонентами. Можно ввести понятие поверхности атаки как отношение времени рассинхронизации двух серверов ко времени прохождения IP-пакета между ними. Чем больше поверхность атаки, тем выше вероятность потенциального нарушителя внедрить свой код в атакуемый узел сети. Отметим, что возможности такой атаки возрастают, если атакующее программное обеспечение (ПО) находится физически ближе к цели, чем контролирующий конфигурацию сети сервер (по аналогии с компьютерными играми и задержкой реакции игры на воздействие игрока).

Для контролируемой защищенной работы инфраструктуры необходимо, чтобы корректирующее воздействие на узлы и компоненты проходило быстрее воздействия атакующей стороны на те же компоненты. А это значит, что управление распределенной ОВС, включая принятые меры противодействия вирусам, должно физически находиться как можно ближе к критическим компонентам инфраструктуры.

2 Ошибки синхронизации в распределенных облачных вычислитедьных средах на примере OpenStack

Рассмотрим помимо описанного ранее «нештатного» поведения компонент OpenStack из-за нехватки ресурсов RabbitMQ [3] другие подобные ошибки в комплексе ПО OpenStack, которые представлены в таблице.

Ошибки в комплексе ПО OpenStack

Ошибка	Описание
1. Нехватка ресурсов RabbitMQ приводит к потере сетевого соединения, распад облачной структуры на «подоблака» [3]	Нехватка числа файловых дескрипторов и последующей перезагрузки системы доставки сообщений RabbitMQ
2. Обрыв соединения со службой RabbitMQ во время миграции ВМ приводит к сбою в машине состояний Nova [4]	Обрыв соединения с упомянутой службой RabbitMQ во время миграции ВМ приводит к сбою в машине состояний Nova. При этом статусом в БД Nova остается «миграция» и указание на узел, с которого мигрировали ВМ, хотя ВМ работают на целевом узле. Эта ошибка, скорее всего, имеет своими корнями некорректную обработку сложного дерева состояний Nova, возникающего при обрыве связи со службой доставки сообщений. Скорее всего, вместо корректной обработки всех возможных состояний в коде Nova используется просто набор тайм-аутов, не обновляя и не синхронизируя актуальные состояния
3. При перезапуске ВМ сетевое подключение не формируется заново и ВМ остается без доступа к сети [5]	Если первая попытка запуска ВМ по каким-то причинам не увенчалась успехом, но при этом сетевое подключение на запущенной ВМ не было уничтожено, то при перезапуске ВМ сетевое подключение не формируется заново и ВМ остается без доступа к сети. Данная ошибка, скорее всего, обусловлена некорректной обработкой сбоя при создании ВМ в сетевой компоненте Neutron. При этом если у ВМ имеются два сетевых интерфейса, то из альтернативной сети успешный запуск возможен
4. Команда безусловного уничтожения ВМ, отданная на этапе создания блочного устройства, приводит к сбою [6]	В распределенных системах, компоненты которых разрабатываются без жестко зафиксированного единого проекта, фактически невозможно полностью корректно обработать (и даже выписать) все возможные промежуточные состояния, в которых может оказаться рассматриваемая промежуточная система. В данной ошибке команда безусловного уничтожения ВМ, отданная на этапе создания блочного устройства, приводит к сбою
5. Ошибка при удалении текущего IP-адреса [7]	Ошибка может возникнуть при нарушении целостности актуальной информации об объектах тенанта при конкурентном выполнении команд «удалить ВМ» и «удалить текущий IP-адрес»
6. Ошибка миграции при наличии у ВМ дополнительного диска [8]	Происходит при «живой» миграции ВМ под управлением Nova (libvirt), имеющей присоединенное блочное устройство на cinder на основе Ceph. Вероятнее всего, данная ошибка возникает из-за несогласованности API Nova с поддержкой libvirt операций с Ceph remote block device (rbd)

Окончание таблицы на с. 102

Ошибки в комплексе ПО OpenStack (*окончание*)

Ошибка	Описание
7. Удаленные пользователи OpenStack могут обходить защиту от подмены IP, вклиниваясь в процедуру запуска ВМ тенанта [9]	Уязвимость CVE-2015-5240 (Race Conditions vulnerability in Openstack Neutron) обусловлена тем, что число состояний алгоритмов, использующих распределенные компоненты, очень велико и корректная обработка разнообразных нештатных ситуаций в промежуточных состояниях является скорее исключением, чем правилом
8. Обращение в БД к записи, которая еще не завершена [10] (Аномалия 2. P1.Dirty Read)	Имеет место при первом запуске ВМ на сервисе Nova, когда в БД планировщика еще нет полной информации о дисковых ресурсах этого сервиса. По-видимому, это связано с рассогласованностью формата БД планировщика и инсталляционных и инициализационных процедур сервиса Nova: планировщик пытается прочитать в БД запись, которая еще полностью не завершена
9. Задержки при работе с БД [11]	В некоторых таблицах, используемых компонентом Nova, продублированы индексы, что приводит к неоправданным задержкам при работе с такими таблицами
10. Попытка присоединить дополнительный том для хранения данных к экземпляру ВМ игнорируется [12]	Невозможно присоединить и использовать дополнительный заново созданный том для хранения данных, причем этот том отображается как доступный для работы. Причина, по-видимому, в неполноте информации, доступной для работы Nova, которая, в свою очередь, не может выполнить команду QEMU по причине дублирования
11. Невозможно удалить образ ВМ, с которой запущен экземпляр ВМ [13]	В БД хранится свойство образа, которое указывает на факт работающей ВМ, созданной из образа. При попытке удаления образа информация об этом указателе не сбрасывается и не дает удалить образ вопреки заявленной функциональности

Указанные выше уязвимости имеют несколько общих черт:

– не завершена разработка и не соблюдается протокол взаимодействия компонентов;

– не стабилизирована работа программных интерфейсов и не соблюдаются требования к их использованию.

При работе компонентов не соблюдаются принятые нормы и правила при обработке исключений и ошибок. Возможно, это недостатки проектирования.

Традиционно обнаружение подобных ошибок осуществляется во время эксплуатации. Как правило, при обнаружении какой-либо аномалии в функционировании системы осуществляется детальное изучение журналов событий и сопоставление ошибок и предупреждений. Фактически первичный анализ сводится к установлению корреляций между сообщениями в журналах различных подсистем. Затем осуществляется поиск подобных ошибок в Интернете и при

нахождении таковых проводится анализ причин их возникновения. При возможности, осуществляется «увязка» по схеме «причина–следствие» пар сообщений об ошибках. Далее формируется первичная «модель» последовательности событий, которая проверяется на тестовых примерах. Тестовые примеры затем используются как «шаги воспроизведения» в описании уже новой ошибки в базе данных (БД) системы контроля ошибок и управления циклом жизни ПО.

В то же время, например, ошибка 3 (см. таблицу) [5], связанная с неправильным сетевым подключением перезапущенной ВМ в относительно современном окружении OpenStack release Mitaka, не может быть проанализирована стандартными средствами. Суть ошибки заключается в том, что ВМ пересоздается на Nova, отличной от первоначальной, а сетевой порт остается на агенте компонента Neutron на том же хосте, что и первоначальная Nova. При этом стандартные средства Neutron позволяют обнаружить «привязку» порта к физическому хосту только по идентификатору ВМ и хосту ее расположения, который после перезапуска ВМ уже изменился. Соответственно, чтобы просто увязать факты, что агент поддерживает старый сбойной порт с присутствием на хосте, на котором он реализован, надо анализировать либо журналы Nova и Neutron, собранные со всех Nova-хостов облака, либо конфигурационные базы OpenStack вручную. Процесс это не быстрый, а в интенсивно загруженной среде и без централизованного упорядоченного журнала событий, возможно, еще и дающий неоднозначный результат.

Далее, обнаружив хост локализации сбойного порта, придется увязать записи из журнала агента Neutron с журналом утилиты, которая проводила развертывание ВМ и виртуальной сети, например Heat. Убедившись в соответствии временны́х меток и идентификаторов запросов, порта, ВМ и анализируемой подсети, можно будет приступить к попытке моделировать подобную ситуацию. Например, можно выделить отдельную зону с неправильно сконфигурированной Nova, которая будет неспособна развернуть ВМ и пытаться воспроизвести ситуацию с перезапуском ВМ на другой Nova.

Очевидно, что стандартными средствами такую задачу также решить невозможно и придется прибегать к некоторым ухищрениям, тщательно следя за тем, чтобы сами эти приемы не внесли своего вклада в особенности генерируемой ошибки. В описании данной ошибки для ее воспроизведения предлагается модифицировать код Nova на тестовом узле, что сложно признать хорошим методом тестирования по любым канонам.

Разберем детально журнальные записи для ситуации, при которой возникает ошибка 5 при удалении текущего IP-адреса (см. таблицу) [7]. В описании указано, что при манипуляции с ВМ и созданием, остановкой и удалением возникает необходимость удаления ассоциированного с ВМ IP-адреса, которое иногда приводит к возникновению ошибки, связанной с тем, что собственно «текущий IP-адрес» к моменту исполнения команды может уже не существовать в результате выполнения конкурентно запущенной команды по удалению ВМ. Ошибка относится к так называемому «мерцающему» типу, когда факт ее возникновения зависит

от массы характеристик слабо контролируемого окружения. Такой тип ошибок является, пожалуй, одним из самых сложных и неприятных из всех возможных для разработчика. Чтобы определить, что же послужило причиной ошибки, часто приходится проводить массу экспериментов, обрабатывать огромный объем генерируемых журнальных записей и данных.

Ручной разбор журнальных записей достаточно трудоемкий в силу того, что приходится кроме упорядочивания событий по времени восстанавливать скрытые взаимосвязи и ассоциации между объектами. Отсутствие актуальной модели ОВС, объединяющей описание как отдельных объектов инфраструктуры, так и связей между ними, неизбежно приводит не только к нарушению целостности конфигурации, но и к появлению ошибок управления ОВС, в том числе «мерцающих». Методологическая причина этого кроется в нарушении порядка и синхронизации процессов управления [1, 3].

Выход из ситуации видится в решении задачи создания и поддержания в актуальном состоянии информационной модели аппаратно-программного комплекса, включая конфигурацию виртуальной среды ОВС. В традиционном варианте реализации аппаратно-программных комплексов эта задача успешно решается средствами класса IBM Rational / Telelogic Tau [14, 15]. В традиционном подходе применение инструментов подобного класса (IBM Rational / Telelogic DOORS / Tau / Synergy / TauTester / DocExpress) позволяет значительно оптимизировать выполнения ряда ITIL (Information Technology Infrastructure Library) процессов, в частности Change и Release Management. В случае же ОВС, когда характерные времена уменьшаются до секунд, наличие подобного отлаженного инструментария является одним из необходимых условий проведения успешной реконфигурации и защиты от нежелательных вторжений.

Применение подобных средств даст возможность:

- упорядочить процессы управления ОВС, сохраняя целостность конфигурации ОВС в практически любой момент времени;
- определять временны́е периоды и управлять ими, когда ОВС не находится в состоянии, отвечающем требованиям SLA (Service Level Agreement) и политик безопасности, и поэтому уязвима и ограниченно работоспособна;
- декомпозировать ОВС на составляющие компоненты, тем самым понижая размерность и сложность как каждого компонента, так и ОВС в целом, а значит, переводя задачу верификации процессов реконфигурации ОВС в разряд разрешимых.

3 Выводы

Подводя итог приведенных выше рассуждений, можно заключить, что для стабилизации работы продукта необходимо:

– предусмотреть центр регистрации ошибок и мониторинга;
– конфигурацию ОВС проводить в условиях обеспечения мультивариантности, процесс реконфигурации должен быть двухфазным [1];
– план реконфигурации должен быть по возможности правильным и непротиворечивым.

С этой целью граф модели ОВС должен состоять из подмножеств плоскостей [2], оптимально обеспечивающих правильность, непротиворечивость и упорядоченность плана реконфигурации [1]. Конфигурации должны регистрироваться в БД с поддержкой транзакций и двухфазного коммита. Уровень изоляции транзакций зависит от необходимого уровня обслуживания и критичности работы компонентов. Регистрация событий должна вестись в централизованном иерархическом порядке в соответствии с планом реконфигурации ресурса. Для регистрации и ситуационного анализа работы ОВС, а также для возможности определить время, узел, характер и вектор атаки вредоносного ПО, если таковая будет иметь место. Помимо смысловых меняющихся полей необходимо регистрировать:

– отметки времени узлов, занятых в транзакциях, включая БД и серверы регистрации событий (журналирование). В случае иерархии таких серверов, при анализе данных необходимы отметки времени серверов, участвующих в регистрации;
– идентификатор версии конфигурации;
– если ОВС выполнена с использованием SDN (Software-Defined Networking), идентификатор «плоскости», slice, в которой проводилась реконфигурация;
– идентификаторы проекта, тенанта, объекта конфигурации и их взаимосвязи.

Под объектом конфигурации здесь понимается сущность, влияющая на работу инфраструктуры, как-то: узел (физический и виртуальный), IP-адрес, порт, протокол, приложение, сервис, сетевая функция, сформированные цепочки сетевых функций, учетная запись, права доступа и т. д.

К сожалению, на настоящий момент не существует ни теоретически обоснованных подходов к построению целостных непротиворечивых конфигураций в распределенных ОВС, ни целостных методов автоматической проверки, способных адекватно тестировать все состояния распределенных систем. Соответственно, ошибки в них выявляются по большей части на этапе эксплуатации. Естественно ожидать, что даже после длительного цикла эксплуатации и исправления ошибок все равно в системе будут оставаться проблемы, способные привести к ошибочным состояниям при маловероятных стечениях обстоятельств. Наличие поддерживаемой в актуальном состоянии информационной модели аппаратно-программного комплекса ОВС позволит существенно сократить путь к решению задачи управления и поддержки целостных непротиворечивых конфигураций в распределенных ОВС.

Литература

1. *Грушо А. А., Забежайло М. И., Зацаринный А. А., Писковский В. О.* Безопасная автоматическая реконфигурация облачных вычислительных сред // Системы и средства информатики, 2016. Т. 26. № 3. С. 83–92.
2. *McGeer R.* Declarative verifiable SDI specifications // 37th IEEE Symposium on Security and Privacy Proceedings. — IEEE, 2016. P. 198–203. http://spw16.langsec.org/papers/mcgeer- verifiable-sdi-specs.pdf.
3. Грушо А. А., Забежайло М. И., Зацаринный А. А., Николаев А. В., Писковский В. О., Тимонина Е. Е. Классификация ошибочных состояний в распределенных вычислительных системах и источники их возникновения // Системы и средства информатики, 2017. Т. 27. № 2. С. 29–40.
4. OpenStack Compute (nova): Inconsistent state when connection to conductor is lost during live migration, 2016. https://bugs.launchpad.net/nova/+bug/1536589.
5. OpenStack Compute (nova): Network not always cleaned up when spawning VMs, 2016. https://bugs.launchpad.net/nova/+bug/1597596.
6. OpenStack Compute (nova): Nova force-delete can't delete instance in vm_state block_device_mapping, 2017. https://bugs.launchpad.net/nova/+bug/1704945.
7. OpenStack Compute (nova): Race conditions between compute and schedule disk report. https://bugs.launchpad.net/nova/+bug/1704975.
8. OpenStack Compute (nova): Live migration fails with an attached non-bootable Cinder volume (Pike), 2017. https://bugs.launchpad.net/nova/+bug/1715569.
9. Neutron: Race Conditions vulnerability in Openstack Neutron, 2015. https://cyber.vumetric.com/vulns/CVE-2015-5240/race-conditions-vulnerability-openstack-neutron.
10. OpenStack Compute (nova): Race conditions between compute and schedule disk report, 2016. https://bugs.launchpad.net/nova/+bug/1610679.
11. OpenStack Compute (nova): Duplicate indexes in nova-db, 2016. https://bugs.launchpad.net/nova/+bug/1641185.
12. OpenStack Compute (nova): Can't attach volume to volume-backed instance, 2017. https://bugs.launchpad.net/nova/+bug/1678694, 2017.
13. OpenStack Compute (nova): Can not delete the image has been launched instance when use rbd. https://bugs.launchpad.net/nova/+bug/1697391.
14. *De Wet N., Kritzinger P.* Towards model-based communication protocol performability analysis with UML 2.0 // Southern African Telecommunication Networks and Applications Conference (SATNAC) Proceedings. — Spier Wine Estate, 2004. 6 p. http://www.satnac.org.za/proceedings/2004/Software/No%20105%20-%20de%20Wet.pdf.
15. Standards-based, model-driven development solution for complex systems, 2016. https://www-03.ibm.com/software/products/en/ratitau.

Поступила в редакцию 04.02.18

ABOUT THE ANALYSIS OF ERRATIC STATUSES IN THE DISTRIBUTED COMPUTING SYSTEMS

A. A. Grusho[1], *M. I. Zabezhailo*[1], *A. A. Zatsarinny*[1], *A. V. Nikolaev*[2], *V. O. Piskovski*[1], *V. V. Senchilo*[1], *I. V. Sudarikov*[1], *and E. E. Timonina*[1]

[1]Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation

[2]N. N. Semenov Institute of Chemical Physics, Russian Academy of Sciences; 4-1 Kosygina Str., Moscow 119991, Russian Federation

Abstract: The analysis of erratic statuses of the distributed computing systems on the example of OpenStack, caused by the absence of synchronization between components of a cloudy computing environment and, also, probable sources of their origin are considered. Detail analysis of journal records is provided in the analysis of one of the listed erratic statuses of software platform of OpenStack. The method of the analysis of similar situations is shown. Creation and maintenance in current state of an information model of a managed cloudy computing environment is suggested. The necessity to use such model for execution of requirements of positive control of a cloudy computing environment and provision of high-quality cloudy services of such types as service, platform, infrastructure, network functions, chains of network functions is substantiated.

Keywords: distributed computing systems; cloudy computing environments; OpenStack; RabbitMQ; Nova; Neutron; analysis of erratic statuses; information cloudy network infrastructures

DOI: 10.14357/08696527180108

Acknowledgments

The paper was partially supported by the Russian Foundation for Basic Research (project 15-29-07981 ofi-m).

References

1. Grusho, A. A., M. I. Zabezhailo, A. A. Zatsarinnyy, and V. O. Piskovski. 2016. Bezopasnaya avtomaticheskaya rekonfiguratsiya oblachnykh vychislitel'nykh sred [Secure automatic reconfiguration of cloudy computing]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(3):83–92.
2. McGeer, R. 2016. Declarative verifiable SDI specifications. *37th IEEE Symposium on Security and Privacy Proceedings*. IEEE. 198–203. Available at: http://spw16.langsec.org/papers/mcgeer-verifiable-sdi-specs.pdf (accessed February 3, 2018).

3. Grusho, A. A., M. I. Zabezhailo, A. A. Zatsarinnyy, A. V. Nikolaev, V. O. Piskovski, and E. E. Timonina. 2017. Klassifikatsiya oshibochnykh sostoyaniy v raspredelennykh vychislitel'nykh sistemakh i istochniki ikh vozniknoveniya [Erroneous states classification in distributed computing systems and sources of their occurrences]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 27(2):29–40.
4. OpenStack Compute (nova): Inconsistent state when connection to conductor is lost during live migration. 2016. Available at: https://bugs.launchpad.net/nova/+bug/1536589 (accessed Febrary 3, 2018).
5. OpenStack Compute (nova): Network not always cleaned up when spawning VMs. 2016. Available at: https://bugs.launchpad.net/nova/+bug/1597596 (accessed February 3, 2018).
6. OpenStack Compute (nova): Nova force-delete can't delete instance in vm_state block_device_mapping. 2017. Available at: https://bugs.launchpad.net/nova/+bug/1704945 (accessed February 3, 2018).
7. OpenStack Compute (nova): Race conditions between compute and schedule disk report. Available at: https://bugs.launchpad.net/nova/+bug/1704975. 2017 (accessed February 3, 2018).
8. OpenStack Compute (nova): Live migration fails with an attached non-bootable Cinder volume (Pike). 2017. Available at: https://bugs.launchpad.net/nova/+bug/1715569 (accessed February 3, 2018).
9. Neutron: Race Conditions vulnerability in Openstack Neutron. 2015. Available at: https://cyber.vumetric.com/vulns/CVE-2015-5240/race-conditions-vulnerability-openstack-neutron/ (accessed February 3, 2018).
10. OpenStack Compute (nova): Race conditions between compute and schedule disk report. 2016. Available at: https://bugs.launchpad.net/nova/+bug/1610679 (accessed February 3, 2018).
11. OpenStack Compute (nova): Duplicate indexes in nova-db. 2016. Available at: https://bugs.launchpad.net/nova/+bug/1641185 (accessed February 3, 2018).
12. OpenStack Compute (nova): Can't attach volume to volume-backed instance. 2017. Available at: https://bugs.launchpad.net/nova/+bug/1678694 (accessed February 3, 2018).
13. OpenStack Compute (nova): Can not delete the image has been launched instance when use rbd. 2017. Available at: https://bugs.launchpad.net/nova/+bug/1697391 (accessed February 3, 2018).
14. De Wet, N. and P. Kritzinger. 2004. Towards model-based communication protocol performability analysis with UML 2.0. *Southern African Telecommunication Networks and Applications Conference (SATNAC) Proceedings*. Spier Wine Estate. 6 p. Available at: http://www.satnac.org.za/proceedings/2004/Software/No%20105%20-%20de%20Wet.pdf (accessed Febrary 3, 2018).
15. Standards-based, model-driven development solution for complex systems. 2015. Available at: https://www-03.ibm.com/software/products/en/ratitau (accessed February 3, 2018).

Received February 4, 2018

Contributors

Grusho Alexander A. (b. 1946) — Doctor of Science in physics and mathematics, professor, Head of Laboratory, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; grusho@yandex.ru

Zabezhailo Michael I. (b. 1956) — Doctor of Science in physics and mathematics, Head of Laboratory, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; m.zabezhailo@yandex.ru

Zatsarinny Alexander A. (b. 1951) — Doctor of Science in technology, professor, Deputy Director, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; AZatsarinny@ipiran.ru

Nikolaev Andrei V. (b. 1973) — Candidate of Science (PhD) in physics and mathematics, senior scientist, N. N. Semenov Institute of Chemical Physics, Russian Academy of Sciences; 4-1 Kosygina Str., Moscow 119991, Russian Federation; gentoorion@mail.ru

Piskovski Viktor O. (b. 1963) — Candidate of Science (PhD) in physics and mathematics, senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; vpvp80@yandex.ru

Senchilo Vladimir V. (b. 1963) — scientist, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; volodias@mail.ru

Sudarikov Igor V. (b. 1989) — scientist, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; 4seev3@gmail.com

Timonina Elena E. (b. 1952) — Doctor of Science in technology, professor, leading scientist, Institute of Informatics Problems, Federal Research Center "Computer Sciences and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; eltimon@yandex.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 110–122

МОДЕЛИРОВАНИЕ БЕЗОПАСНЫХ АРХИТЕКТУР РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ КОМПЛЕКСНОЙ ВИРТУАЛИЗАЦИИ*

***Н. А. Грушо*[1], *В. В. Сенчило*[2]**

Аннотация: Безопасные архитектуры распределенных информационных систем (ИС) можно строить на основе виртуальных машин. Каждая виртуальная машина выполняет строго определенный функционал. Это позволяет выделить виртуальные машины, решающие исключительно задачи информационной безопасности (ИБ). Рассматривается влияние архитектуры, использующей виртуальные машины для обеспечения ИБ, на скорость приема и передачи данных между элементами ИС. Исследования потери скорости при передаче и приеме подтвердили возможности построения безопасных архитектур с помощью изоляции виртуальных доменов виртуальными серверами безопасности.

Ключевые слова: информационная безопасность; распределенные информационно-вычислительные системы; виртуализация

DOI: 10.14357/08696527180109

1 Введение

Архитектурные решения безопасных ИС с использованием виртуальных машин [1, 2] требуют не только теоретического изучения, но и моделирования с целью получения достоверных данных о применимости таких решений. Основой безопасных архитектур являются виртуальные машины, связанные между собой в единую ИС. Каждый элемент такой ИС выполняет строго определенный функционал, что позволяет применить в защищающих изоляцию доменов виртуальных машинах методы интеллектуального анализа данных (ИАД) [3] для обеспечения ИБ. В данной работе рассматривается влияние архитектуры виртуальных механизмов ИБ ИС на ее производительность, в частности на скорость приема и передачи данных между элементами ИС.

*Работа выполнена при частичной поддержке РФФИ (проект 15-07-02053).

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, info@itake.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, volodias@mail.ru

2 Описание архитектуры

Первым шагом к построению безопасных ИС с использованием виртуальных машин является определение перечня требований по ИБ для задач, решаемых ИС. Серверы отвечают за предоставление конкретных сервисов, а клиенты обращаются к сервисам, например к файловым хранилищам, базам данных, веб-интерфейсам и т. д. Отдельно необходимо вместе с этими элементами выделить элементы ИС, отвечающие за ИБ. Также выделяется группа инфраструктурных элементов ИС, например маршрутизаторы, межсетевые экраны (брандмауэры) и т. д.

Следующий шаг — это виртуализация выделенных элементов (серверов, маршрутизаторов и т. п.). Полученные виртуальные машины объединяются в единую ИС с наиболее выгодной в плане производительности и безопасности топологией.

Средства виртуализации, такие как программное обеспечение Oracle Virtual Box, позволяют объединять виртуальные машины в локальные вычислительные сети (ЛВС) с помощью таких механизмов, как NAT (Native Address Translation), сетевой мост (Network Bridge) или виртуальная внутренняя сеть, когда виртуальные машины могут обмениваться данными только между собой, не имея доступа к внешней сети хост-машины (гипервизора).

Система трансляции сетевых адресов (NAT) используется для подмены адреса отправителя в пакетах, передаваемых от клиента к серверу. Например, узел А в ЛВС через NAT (узел Б) отправляет пакет серверу, а NAT подменяет адрес отправителя на свой адрес. Сервер полагает, что общается с узлом Б, хотя на самом деле этот пакет был отправлен узлом А. При получении ответа от сервера NAT проводит обратную подмену адреса получателя (на адрес узла А) и передает этот пакет узлу А. Данная система используется для того, чтобы предоставить множеству клиентов ЛВС доступ во внешнюю сеть, имея всего один внешний адрес. Клиенты ЛВС имеют только внутренние сетевые адреса.

Система подключения с помощью сетевого моста (Network Bridge) аналогична подключению клиентов с помощью сетевого коммутатора. Пакеты от клиентов передаются сразу во внешнюю сеть. А все пакеты, полученные из внешней сети, передаются напрямую клиентам.

Виртуальная сеть — это объединение виртуальных машин в ЛВС. Данная ЛВС не имеет доступа к внешней сети гипервизора. Виртуальная сеть работает аналогично объединению персональных компьютеров (ПК) ЛВС с помощью коммутатора.

Изоляция элементов ИС с помощью виртуализации обеспечивает безопасность всей ИС и возможность контроля каждого из них, независимо от состояния других элементов. Такое разделение приводит к увеличению нагрузки на хост-машины и, как следствие, к снижению скорости обмена данными в ИС. В связи с этим возникает необходимость исследования влияния архитектуры виртуализации на скорость обмена данными в ней.

3 Моделирование информационной системы на базе виртуальных машин

Рассмотрим упрощенную архитектуру безопасной ИС с использованием виртуальных машин. Далее перечислены основные элементы упрощенной ИС. Элемент, предоставляющий сервис, — сервер. Элемент, обращающийся к предоставляемому сервису, — клиент. Элемент, осуществляющий связь между клиентом и сервером, — хост-машина. Элемент, обеспечивающий управление потоком данных (маршрутизация или фильтрация трафика), — маршрутизатор или промежуточная виртуальная машина.

В корпоративных сетях для виртуализации используются высокопроизводительные серверы с несколькими многоядерными процессорами и быстродействующей дисковой системой. Это позволяет запускать одновременно десятки виртуальных машин без снижения производительности каждой из них в отдельности. Для моделирования упрощенной архитектуры ИС можно использовать обычные ПК, способные обеспечить работу хотя бы двух виртуальных машин одновременно. Для снижения нагрузки на этот ПК роль сервера можно вынести на отдельный ПК, подключив его по локальной сети к хост-машине.

В результате упрощения ИС состоит из:

- ПК-сервера;
- ПК-хост-машины;
- ЛВС между хост-машиной и сервером;
- виртуальной машины-клиента;
- виртуальной машины — промежуточного сетевого узла, выполняющего функцию защиты информации.

Будем считать, что элементы: сервер, ЛВС, хост-машина–сервер, хост-машина — это неизменяемые элементы ИС. В таком случае на скорость обмена данными в ИС между клиентом и сервером будут влиять промежуточные сетевые узлы, т. е. их конфигурация, настройка сетевых интерфейсов. Также на скорость обмена данными будут влиять уровень виртуализации клиента и настройка его подключения к ЛВС. Это может быть виртуализация на хост-машине, а может быть паравиртуализация на другой виртуальной машине. Подключение к ЛВС может быть реализовано любым способом, предоставляемым программным обеспечением (ПО) гипервизора.

Элементами, осуществляющими контроль сетевого трафика (функция по обеспечению безопасности, рассмотренная ранее), являются хост-машина и промежуточный сетевой узел.

Реализация описанной выше упрощенной модели безопасной ИС в виде макета может показать возможность создания такой ИС, а также позволит оценить скорость взаимодействия элементов ИС между собой.

4 Макеты информационной системы с использованием виртуальных машин

4.1 Макет 1

Рассмотрим архитектуру ИС, в которой одна из виртуальных машин является гипервизором для другой виртуальной машины (вложенные виртуальные машины). В качестве средства виртуализации используем ПО Oracle VirtualBox.

На физическом гипервизоре Клиент 1 с помощью ПО VirtualBox создается виртуальная машина Клиент 2, выполняющая функцию гипервизора, на котором запускается еще одна виртуальная машина Клиент 3. Эта «вложенная» виртуальная машина зависима от ресурсов и конфигурации хоста — виртуальной машины Клиент 2. Подключение к ИС «вложенной» виртуальной машины может быть осуществлено с помощью конфигурирования ЛВС внутри виртуальной машины Клиент 2. Рассматриваются две конфигурации подключения виртуальных машин Клиент 2 и Клиент 3 к ЛВС: режим NAT и режим сетевого моста.

Основной процесс взаимодействия клиентов и сервера — это обмен данными между собой. Для измерения скорости такого взаимодействия требуется специальное ПО LanBench. Данное ПО производит измерение скорости приема,

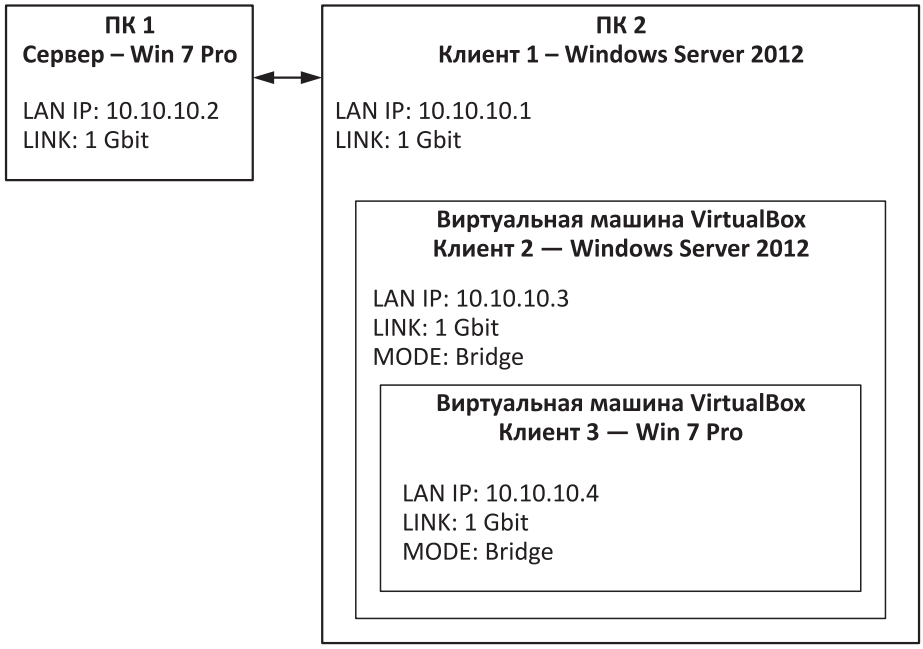


Рис. 1 Макет для тестирования сети вложенной виртуальной системы в режиме «сетевой мост»

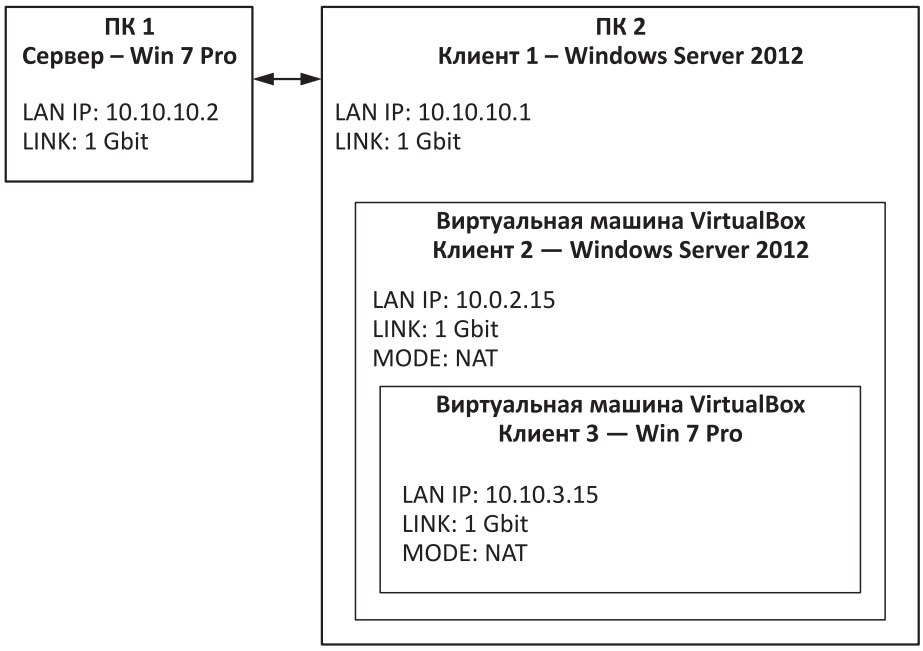


Рис. 2 Макет для тестирования сети вложенной виртуальной системы в режиме NAT

передачи, одновременного приема-передачи между клиентом и сервером в ЛВС.

Описанная выше архитектура реализуется в виде макетов со структурой, указанной на рис. 1 и 2.

Для проведения экспериментов необходимо следующее ПО и оборудование:

– ПК «Сервер» (далее по тексту — Сервер) с сетевым адаптером, поддерживающим скорость 1 Гбит/с, операционной системой (далее по тексту — ОС) Microsoft Windows 7 Professional и ПО LAN BENCH. Процессор Intel Core-i7, 8 ядер, 8 ГБ оперативной памяти;

– ПК «Клиент 1» (далее по тексту — Клиент 1) с сетевым адаптером, поддерживающим скорость 1 Гбит/с, ОС Microsoft Windows Server 2012 R2, ПО LAN BENCH, ПО Virtual Box. Процессор Intel Core-i5, 4 ядра, 8 ГБ оперативной памяти;

– сетевой кабель для соединения Сервера и Клиента 1.

После включения Сервера необходимо произвести настройку сетевого адаптера, задав ему фиксированный адрес IPv4: 10.10.10.2, маску сети 255.255.255.0. Шлюз не указывается. После этого необходимо запустить программу LAN BENCH. После запуска программы необходимо нажать «Listen». На этом подготовка Сервера завершена.

После включения Клиента 1 необходимо произвести настройку сетевого адаптера, задав ему фиксированный адрес IPv4: 10.10.10.1, маску сети 255.255.255.0. Шлюз не указывается. После этого необходимо запустить программу LAN BENCH.

Настройки LAN BENCH для Клиента 1 и виртуальных машин Клиент 2 и Клиент 3 одинаковы:

- в меню выбрать File → Configure;
- задать адрес Сервера 10.10.10.2;
- задать Test duration 10 с;
- задать Packet Size 5 КБ;
- задать Connections 20;
- для измерения скорости передачи выбрать «Send only»;
- для измерения скорости приема выбрать «Receive only».

На Клиенте 1 необходимо скачать и установить ПО VirtualBox. После установки ПО необходимо создать виртуальную машину. Исходя из имеющихся ресурсов, в свойствах виртуальной машины необходимо настроить:

- количество используемых процессоров — 2;
- объем выделенной оперативной памяти — 4 ГБ;
- сетевой адаптер — 1 в режиме «сетевой мост».

На эту виртуальную машину необходимо установить ОС Windows Server 2012 и запустить ее. Запущенная виртуальная машина — виртуальная машина Клиент 2.

В виртуальной машине Клиент 2 необходимо произвести настройку сетевого адаптера, задав ему фиксированный адрес IPv4: 10.10.10.3, маску сети 255.255.255.0. Шлюз не указывается. После этого необходимо запустить программу LAN BENCH.

В виртуальной машине Клиент 2 необходимо установить ПО VirtualBox. После установки необходимо создать виртуальную машину. Исходя из задействованных ресурсов, в свойствах виртуальной машины необходимо настроить:

- количество используемых процессоров — 1;
- объем выделенной оперативной памяти — 2 ГБ;
- сетевой адаптер — 1 в режиме «сетевой мост».

На эту виртуальную машину необходимо установить ОС Windows 7 Pro и запустить ее. Запущенная виртуальная машина — виртуальная машина Клиент 3.

В виртуальной машине Клиент 3 необходимо произвести настройку сетевого адаптера, задав ему фиксированный адрес IPv4: 10.10.10.4, маску сети 255.255.255.0. Шлюз не указывается. После этого необходимо запустить программу LAN BENCH.

4.2 Макет 2

Второй макет реализует архитектуру с параллельно запущенными виртуальными машинами, одна из которых начинает выполнять роль маршрутизатора (Клиент 2). Подключение остальных элементов ИС (Клиент 3) к ЛВС возможно только через маршрутизатор — виртуальную машину Клиент 2. Хост-машина Клиент 1 физически подключена к ЛВС и имеет доступ к Серверу.

Для моделирования такой архитектуры был собран макет, соответствующий структуре, показанной на рис. 3.

Требования к ПО и оборудованию соответствуют макету 1. Настройка Сервера соответствует настройке на макете 1. Выбрана следующая настройка виртуальных систем Клиент 2 и Клиент 3.

На хост-машине Клиент 1 необходимо установить ПО VirtualBox. После установки ПО необходимо создать виртуальную машину. В свойствах виртуальной машины выбраны настройки:

– количество используемых процессоров — 2;

– объем выделенной оперативной памяти — 4 ГБ;

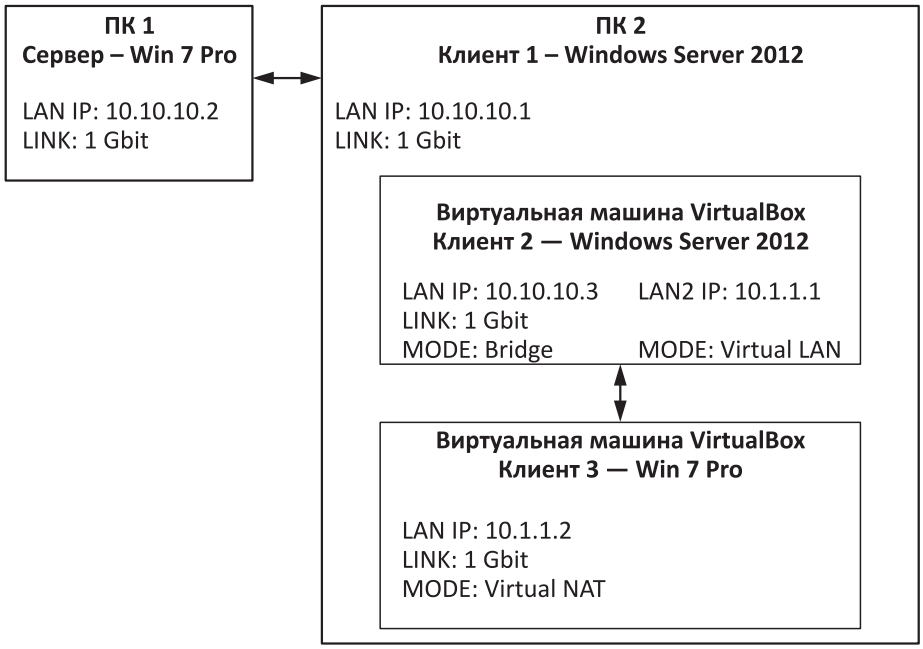


Рис. 3 Макет для тестирования сети с параллельными виртуальными системами и маршрутизацией

– сетевой адаптер 1 — в режиме «сетевой мост»;

– сетевой адаптер 2 — в режиме «внутренняя сеть».

На эту виртуальную машину необходимо установить ОС Windows Server 2012 и запустить ее. Запущенная виртуальная машина — виртуальная машина Клиент 2.

В виртуальной машине Клиент 2 необходимо произвести настройку сетевого адаптера 1, задав ему фиксированный адрес IPv4: 10.10.10.3, маску сети 255.255.255.0. Шлюз не указывается. В свойствах сетевого адаптера 2 необходимо указать IPv4 адрес 10.1.1.1 с маской 255.255.255.0. Затем необходимо установить компонент «маршрутизация» в Windows Server и настроить NAT между сетевым адаптером 1 и сетевым адаптером 2. Также необходимо добавить статический маршрут для сетевого адаптера 2: сеть назначения 10.1.1.0, маска 255.255.255.0, шлюз 10.1.1.1. После этого необходимо запустить программу LAN BENCH.

В хост-машине Клиент 1 необходимо создать новую виртуальную машину. В свойствах виртуальной машины необходимо настроить:

– количество используемых процессоров — 2;

– объем выделенной оперативной памяти — 2 ГБ;

– сетевой адаптер — 1 в режиме «внутренняя сеть».

На эту виртуальную машину необходимо установить ОС Windows 7 Pro и запустить ее. Запущенная виртуальная машина — виртуальная машина Клиент 3.

В виртуальной машине Клиент 3 необходимо произвести настройку сетевого адаптера, задав ему фиксированный адрес IPv4: 10.1.1.2, маску сети 255.255.255.0. Шлюз 10.1.1.1.

После этого необходимо запустить программу LAN BENCH.

5 Моделирование информационных потоков в информационной системе

5.1 Эксперименты для макета 1

Для проведения эксперимента на хост-машине Клиент 1 и виртуальных машинах Клиент 2 и Клиент 3 по очереди запускается программа LAN BENCH и проводится измерение скорости обмена данными между ними и Сервером. Сначала проводится измерение скорости передачи, а затем — скорости приема. Результаты измерений помещаются в таблицу. Затем макет настраивается в соответствии с рис. 2, и эксперимент повторяется. Для сопоставления результаты записываются в ту же таблицу. После проведения всех экспериментов необходимо сравнить скорости работы вложенных виртуальных систем в локальной сети.

5.2 Эксперименты для макета 2

Для проведения эксперимента на хост-машине Клиент 1 и виртуальной машине Клиент 2 по очереди запускается программа LAN BENCH и проводится измерение скорости приема и передачи данных между ними и Сервером. При этом служба маршрутизации на виртуальной машине Клиент 2 должна быть отключена. Результаты измерений записываются в таблицу. Затем на виртуальной машине Клиент 2 включается служба маршрутизации, и измерения проводятся для виртуальных машин Клиент 2 и Клиент 3. Результаты также записываются в таблицу. После проведения всех экспериментов появляется возможность сравнить скорости обмена данными в локальной сети.

6 Результаты эксперимента

Результаты экспериментов разбиты на группы «Передача» и «Прием», а также отображены в виде гистограмм на рис. 4. Значения на гистограмме выражены в Мбит/с.

Анализ полученных результатов позволяет объяснить данные о скорости **передачи** данных:

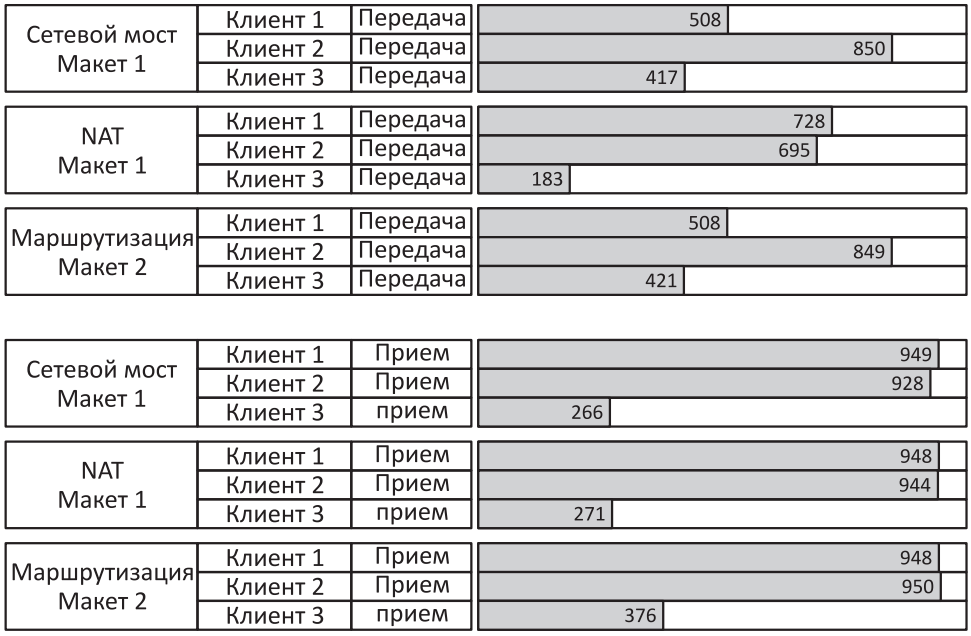


Рис. 4 Результаты экспериментов

1. Для хост-машины **Клиент 1**: в режиме NAT скорость передачи данных выше, чем в режиме «сетевой мост» для макета 1 или «маршрутизация» для макета 2. В режиме NAT сетевой трафик хост-машины Клиент 1, не предназначенный для виртуальных машин, отсекается, т. е. виртуальные машины его не обрабатывают, а следовательно, нагрузка на хост-машину Клиент 1 ниже.
2. Для виртуальной машины **Клиент 2**: в режиме «сетевой мост» скорость передачи выше, чем в режиме NAT. Обработчик сетевого трафика VirtualBox принимает пакеты, проводит «трансляцию» адресов и пересылает их на сетевой адаптер, подключенный к локальной сети. Трансляция адресов вызывает замедление передачи данных.
3. Для виртуальной машины **Клиент 2**: на макете 1 в режиме «сетевой мост» и макете 2 (маршрутизация) скорость передачи одинакова, так как сетевой трафик поступает напрямую на сетевой адаптер хост-машины Клиента 1, не требуя дополнительных системных ресурсов для обработки, в отличие от режима NAT.
4. Для виртуальной машины **Клиент 3**: аналогично виртуальной машине Клиент 2 скорость передачи данных в режиме NAT существенно ниже, чем при работе в режиме «сетевой мост» или же в режиме маршрутизации на макете 2.

Анализ полученных результатов позволяет объяснить информацию о скорости **приема** данных:

1. Скорость приема данных хост-машиной Клиент 1 и виртуальными машинами Клиент 2 и Клиент 3 не зависит от выбранного режима работы сетевого адаптера для макетов 1 и 2.
2. Скорость приема данных виртуальной машиной Клиент 3 (на макетах 1 и 2) значительно ниже, чем скорость приема хост-машиной Клиент 1 и виртуальной машиной Клиент 2, так как подключение к ЛВС виртуальной машины Клиент 3 — это «виртуальное» подключение, иначе — подключение через виртуальные адаптеры. Чем меньше выделено (осталось) системных ресурсов на виртуализацию сетевого адаптера, тем ниже скорость приема.
3. Скорость приема «вложенной» виртуальной машины Клиент 3 (макет 1) всего на 100 Мбит/с ниже скорости приема «параллельно» запущенной виртуальной машины Клиент 3 (макет 2). Как и в предыдущем выводе, скорость приема данных зависит от выделенных системных ресурсов виртуального сетевого адаптера. Если на макете 1 виртуальная машина Клиент 3 работает в режиме паравиртуализации, то на макете 2 виртуальная машина Клиент 3 работает в режиме обычной виртуализации и имеет больше выделенных системных ресурсов, что объясняет бóльшую скорость приема.

Таким образом, из полученных результатов можно сделать следующие выводы.

Конфигурация сетевых адаптеров, топология ЛВС в моделируемой ИС с использованием виртуальных машин влияет на скорость обмена данными между элементами ИС.

Наибольшее снижение скорости передачи данных происходит при использовании сложных сетевых механизмов (NAT) вместе с паравиртуализацией. Это происходит потому, что вычисления производятся внутри виртуальной машины. Как следствие, вырастает нагрузка на хост-машину.

Также стоит обратить внимание на производительность различных ОС. Например, для оптимальной работы промежуточных сетевых узлов предпочтительнее использовать серверные ОС.

7 Заключение

Рассмотренная упрощенная модель и созданные макеты демонстрируют возможность создания безопасной архитектуры ИС с использованием виртуальных машин. Полученные результаты показывают, что изоляция элементов ИС приводит к повышению нагрузки на хост-машину и, как следствие, к снижению скорости обмена данными между элементами в 2–3 раза на простых ПК. Однако изоляция доменов с помощью выделения специальных виртуальных машин, берущих на себя все функции безопасности обмена информации, решает задачи контроля информационных потоков в ИС. Отсюда следует возможность эффективного синтеза [4] и анализа ИБ в ИС [5].

Требуются дальнейшие исследования в области применения средств виртуализации для построения безопасных ИС. Необходимо провести моделирование на гипервизорах Microsoft Hyper-V, VMware, QEMU и др. с использованием систем обеспечения ИБ, встроенных в виртуальные машины защиты информации, например брандмауэрами с DPI (Deep Packet Inspection) и системами ИАД для анализа данных мониторинга [6–8].

Литература

1. *Грушо А. А., Грушо Н. А., Левыкин М. В., Тимонина Е. Е.* Безопасные архитектуры распределенных информационно-вычислительных систем на основе комплексной виртуализации // Проблемы информационной безопасности. Компьютерные системы, 2016. № 4. С. 32–35.
2. *Грушо А. А., Грушо Н. А., Тимонина Е. Е.* Оценка защищенности в безопасных архитектурах распределенных информационных систем // Системы и средства информатики, 2016. Т. 26. № 4. С. 31–37.
3. *Грушо А. А., Грушо Н. А., Забежайло М. И., Тимонина Е. Е.* Интеллектуальный анализ данных в обеспечении информационной безопасности // Проблемы информационной безопасности. Компьютерные системы, 2016. № 3. С. 55–60.
4. *Грушо А. А., Грушо Н. А., Тимонина Е. Е.* Синтез архитектуры информационной безопасности в распределенных информационно-вычислительных системах // Проблемы информационной безопасности. Компьютерные системы, 2017. № 2. С. 23–30.

5. *Grusho A., Grusho N., Levykin M., Timonina E.* Analysis of information security of distributed information systems // 9th Congress (International) on Ultra Modern Telecommunications and Control Systems and Workshops Proceedings. — Piscataway, NJ, USA: IEEE, 2017. P. 96–100.
6. *Грушо А., Забежайло М., Зацаринный А.* Контроль и управление информационными потоками в облачной среде // Информатика и её применения, 2015. Т. 9. Вып. 4. С. 95–101.
7. *Grusho A., Timonina E., Shorgin S.* Modelling for ensuring information security of the distributed information systems // 31th European Conference on Modelling and Simulation Proceedings. — Dudweiler, Germany: Digitaldruck Pirrot GmbHP, 2017. P. 656–660.
8. *Grusho A., Grusho N., Zabezhailo M., Zatsarinny A., Timonina E.* Information security of SDN on the basis of meta data // Computer network security / Eds. J. Rak, J. Bay, I. V. Kotenko, *et al.* — Lecture notes in computer science ser. — Springer, 2017. Vol. 10446. P. 339–347.

Поступила в редакцию 26.11.17

MODELING OF SECURE ARCHITECTURE OF DISTRIBUTED INFORMATION SYSTEMS ON THE BASIS OF INTEGRATED VIRTUALIZATION

N. A. Grusho and V. V. Senchilo

Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: Secure architecture of distributed information systems can be built on the basis of virtual machines. Each virtual machine carries out strictly particular functional. It allows defining virtual machines which are solving problems of information security only. In the paper, the influence of the architecture using virtual machines for ensuring information security on collecting speed and data transmission between elements of the information system is considered. Research of loss of speed by transfer and reception confirmed possibilities of creation of secure architecture by means of isolation of virtual domains with virtual servers of security.

Keywords: distributed information security; distributed information system; virtualization

DOI: 10.14357/08696527180109

Acknowledgments

The paper was partially supported by the Russian Foundation for Basic Research (project 15-07-02053).

References

1. Grusho, A. A., N. A. Grusho, M. V. Levykin, and E. E. Timonina. 2016. Bezopasnye arkhitektury raspredelennykh informatsionno-vychislitel'nykh sistem na osnove kompleksnoy virtualizatsii [Secure architecture of distributed information systems on the basis of integrated virtualization]. *Problemy informatsionnoy bezopasnosti. Komp'yuternye sistemy* [Problems of Information Security. Computer Systems] 4:32–35.
2. Grusho, A. A., N. A. Grusho, and E. E. Timonina. 2016. Otsenka zashchishchennosti v bezopasnykh arkhitekturakh raspredelennykh informatsionnykh sistem [Security evaluation in secure architecture of distributed information systems]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(4):31–37.
3. Grusho, A. A., N. A. Grusho, M. I. Zabezhailo, and E. E. Timonina. 2016. Data mining in ensuring information security. *Autom. Control Comp. S.* 50(8):722–725.
4. Grusho, A. A., N. A. Grusho, and E. E. Timonina. 2017. Information security architecture synthesis in distributed information computation systems. *Autom. Control Comp. S.* 51(8):799–804.
5. Grusho, A., N. Grusho, M. Levykin, and E. Timonina. 2017. Analysis of information security of distributed information systems. *9th Congress (International) on Ultra Modern Telecommunications and Control Systems and Workshops Proceedings*. Piscataway, NJ. 96–100.
6. Grusho, A., M. Zabezhailo, and A. Zatsarinnyy. 2015. Kontrol' i upravlenie informatsionnymi potokami v oblachnoy srede [Information flow monitoring and control in cloud computing environment]. *Informatika i ee Primeneniya — Inform. Appl.* 9(4):95–101.
7. Grusho, A., E. Timonina, and S. Shorgin. 2017. Modelling for ensuring information security of the distributed information systems. *31th European Conference on Modelling and Simulation Proceedings*. Dudweiler, Germany: Digitaldruck Pirrot GmbHP. 656–660.
8. Grusho, A., N. Grusho, M. Zabezhailo, A. Zatsarinny, and E. Timonina. 2017. Information security of SDN on the basis of meta data. *Computer network security*. Eds. J. Rak, J. Bay, I. V. Kotenko, *et al*. Lecture notes in computer science ser. 10446:339–347.

Received November 26, 2017

Contributors

Grusho Nikolai A. (b. 1982) — Candidate of Science (PhD) in physics and mathematics, senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; info@itake.ru

Senchilo Vladimir V. (b. 1963) — scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; volodias@mail.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 123–138

БАЛАНСИРОВКА НАГРУЗКИ В ЗАЩИЩЕННЫХ СЕТЯХ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ SDN

***О. Ю. Гузев*[1], *И. В. Чижов*[2]**

Аннотация: Технология SDN (Software-Defined Networking, программно-конфигурируемые сети) по сравнению с традиционными IP-сетями позволяет программировать поведение сети при помощи централизованного контроллера. Передающие трафик устройства в этом случае занимаются только пересылкой фреймов на основании таблиц потоков, загружаемых в них контроллером. Таблица потоков строится на контроллере в ходе обработки информации о приходящих на передающие устройства потоках трафика. Перечисленные свойства технологии были использованы при создании SDN-балансировщика нагрузки для устройств защищенных сетей. В статье рассматривается архитектура и программное обеспечение балансировщика. Приводятся описания схем и результаты экспериментов по балансировке нагрузки на такие устройства, как L3-криптошлюз, TLS (Transport Layer Security, защита транспортного уровня) шлюз, IDS (Intrusion Detection System, система обнаружения вторжений).

Ключевые слова: SDN; ПКС; программно-конфигурируемые сети; контроллер; OpenFlow; криптошлюз; TLS; система обнаружения вторжений; IDS; балансировка нагрузки; DPDK; Open vSwitch; Beacon

DOI: 10.14357/08696527180110

1 Введение

Традиционные IP-сети в большинстве своем комплексны, статичны и сложны в обслуживании [1–4]. При этом с каждым годом бизнес предъявляет все большие требования к их функционалу и производительности [5]. Такое противоречивое состояние развития IP-сетей требует применения новых подходов, позволяющих упростить, стандартизировать, виртуализировать, централизовать сетевые устройства (СУ), процессы и управление ими. Одним из таких подходов является технология SDN [4, 6].

В данной статье рассматривается использование технологии SDN для создания высокопроизводительного, конфигурируемого балансировщика нагрузки

[1]Центр научных исследований и перспективных разработок, ОАО «ИнфоТеКС», oleg.guzev@infotecs.ru

[2]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики; Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ichizhov@cs.msu.ru

и применение данного балансировщика для масштабирования таких устройств защищенных сетей, как L3-криптошлюз, TLS-шлюз, IDS.

Данная задача является актуальной, поскольку к балансировщику нагрузки, используемому в указанной сфере, помимо поддержки базовых алгоритмов балансировки и производительности предъявляются дополнительные специфические требования с точки зрения функциональности. К таким требованиям можно отнести:

– возможность изменения содержимого полей служебных заголовков Ethernet-фрейма, IP-пакета, TCP/UDP (Transmission Control Protocol / User Datagram Protocol) сегмента;

– поддержку протоколов туннелирования GRE (Generic Routing Encapsulation), VXLAN (Virtual eXtensible Local Area Network);

– согласованную работу с балансировщиком нагрузки на другом конце масштабируемого защищенного канала связи для обслуживания случаев отказа отдельных линков;

– использование любого набора полей служебных заголовков фрейма/пакета/сегмента для идентификации балансируемого потока трафика.

Кроме того, с точки зрения пути прохождения полезного трафика можно выделить несколько типов СУ, для которых может требоваться балансировка нагрузки (рис. 1).

Большинство балансировщиков нагрузки традиционных IP-сетей не отвечают всем указанным требованиям. Кроме того, в них отсутствует возможность программируемости, что затрудняет использование одного устройства в разных сценариях балансировки. В то же время технология SDN и протокол OpenFlow легко позволяют реализовать все перечисленные требования.

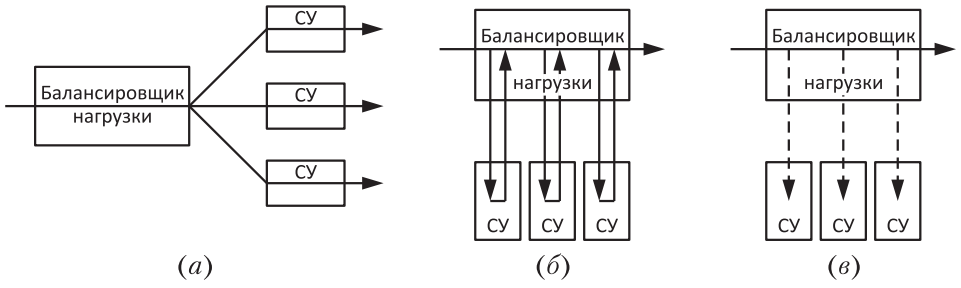


Рис. 1 Типы СУ, для которых выполняется балансировка нагрузки: (*а*) СУ устанавливаются «в разрыв» пути прохождения полезного трафика; (*б*) балансировщик нагрузки «заворачивает» полезный трафик на СУ, после обработки на которых трафик возвращается на балансировщик для дальнейшей коммутации; (*в*) полезный трафик «зеркалируется» балансировщиком на СУ для анализа

2 Программно-аппаратное обеспечение SDN-балансировщика

SDN-балансировщик нагрузки (далее балансировщик) включает два основных компонента: передающее устройство (OpenFlow-коммутатор) и SDN-контроллер. Далее отдельно рассмотрим каждый из них.

2.1 Передающее устройство

Задачей передающего устройства является коммутация приходящих Ethernet-фреймов на основании таблицы потоков, полученной от контроллера. Самостоятельными интеллектуальными функциями обработки трафика данное устройство не обладает. Взаимодействие с контроллером осуществляется по протоколу OpenFlow или схожим. Программно-аппаратная реализация передающего устройства может различаться в зависимости от требуемой пропускной способности.

Если **объем балансируемого трафика не превышает 10 ГБ/с Full-Duplex**, то в качестве передающего устройства можно использовать сервер общего назначения с двумя портами 10G. В экспериментах, описанных в данной статье, использовалось передающее устройство со следующими основными характеристиками:

- процессор: 1 процессор Intel Xeon E5-2620 v3 (2,40 ГГц, 6 ядер);
- оперативная память: 8 ГБ;
- сетевые адаптеры 10G (полезный трафик): 2 адаптера Intel 82599ES;
- сетевые адаптеры 1G (OpenFlow, управление): 2 адаптера Intel I350-AM2.

Обязательное требование к сетевым адаптерам 10G — поддержка технологии Intel DPDK (Data Plane Development Kit, далее DPDK). Тип и объем жесткого диска критического значения не имеют, так как в ходе работы OpenFlow-коммутатора регулярных операций чтения-записи не происходит. Следует добавить, что приведенная выше аппаратная конфигурация не является минимальной для SDN-балансировки трафика 10 ГБ/с Full-Duplex. В задачи данной работы не входила оптимизация аппаратных ресурсов с целью максимального снижения стоимости передающего устройства.

Основное программное обеспечение экспериментального передающего устройства включало Debian Server 8.7.1, Open vSwitch 2.7.0 и Intel DPDK 16.11.

Во всех экспериментах, описанных в данной статье, на передающем устройстве использовалась технология DPDK для предоставления программному коммутатору Open vSwitch (далее OVS), функционирующему в пользовательском окружении Linux, эксклюзивного доступа к ресурсам. Благодаря этому была получена производительность коммутатора OVS 10 ГБ/с Full-Duplex для IP-пакетов с MTU (maximum transmission unit) в диапазоне 256–1500 Б. Для IP-пакетов размером 64 Б максимальная производительность OVS составила 6990 МБ/с Half-Duplex. Нагрузочное тестирование проводилось при помощи инструмента

Pktgen 3.1.2. В ходе настройки OVS/DPDK для каждого порта 10G создавались четыре RX-очереди. Каждая очередь ассоциировалась с одним логическим ядром процессора. Для портов 10G использовался драйвер vfio-pci. Также для работы OVS/DPDK на этапе загрузчика Linux резервировались 2048 2M-страниц оперативной памяти.

Если **объем балансируемого трафика превышает 10 ГБ/с Full-Duplex**, то в качестве передающего устройства можно использовать аппаратный коммутатор с требуемым числом портов 10/40/100G и поддержкой протокола OpenFlow нужной версии.

2.2 SDN-контроллер

Аппаратное обеспечение SDN-контроллера может значительно различаться в зависимости от максимально возможной нагрузки на контроллер. Нагрузка на контроллер зависит от количества новых потоков в секунду (OpenFlow-сообщения PACKET_IN от передающих устройств), которые контроллер должен обработать, и от сложности логики обработки каждого потока. Ключевой аппаратной характеристикой для работы контроллера является производительность процессора. В экспериментах на сервере контроллера использовались 2 процессора Intel Xeon E5-2609 v3 (1,90 ГГц, 6 ядер), что было избыточным для нагрузки, создаваемой на контроллере в ходе тестирования. Для управляющей связи с передающими устройствами использовался порт 1G. Программное обеспечение контроллера включало: Debian Server 8.7.1 и модифицированный контроллер Beacon.

Открытый контроллер Beacon написан на языке Java и обладает следующими преимуществами по сравнению со многими другими SDN-контроллерами [7, 8]: многопоточность, высокая производительность, легкость разработки собственных приложений. Основные недостатки контроллера Beacon: старшая поддерживаемая версия протокола OpenFlow-1.0, нестабильность работы, в настоящее время не развивается и не поддерживается. Модификация исходного контроллера Beacon заключалась в минимизации его функционала до самого необходимого для повышения стабильности работы, а также в разработке собственных приложений для балансировки нагрузки.

3 Отказоустойчивость SDN-балансировщика нагрузки

Отказоустойчивость SDN-балансировщика нагрузки зависит от следующих факторов:

- отказоустойчивость при обрыве канала связи, по которому происходит балансировка трафика;
- отказоустойчивость контроллера;
- отказоустойчивость передающих устройств.

Далее рассмотрим варианты реализации первых двух позиций.

3.1 Отказоустойчивость при обрыве канала связи

Алгоритм обеспечения отказоустойчивости при обрыве канала связи, по которому происходит балансировка трафика, рассмотрен на примере схемы масштабируемой защищенной связи двух географически распределенных корпоративных сетей / ЦОД. На рис. 2 представлены физическая и логическая топологии распределенной сети. Далее приводится описание алгоритма отказоустойчивости.

На рис. 2 показана схема соединения двух сегментов корпоративной сети через публичную сеть при помощи масштабируемого защищенного канала связи. Шифрование и расшифрование корпоративного трафика выполняются при помощи шести L3-криптошлюзов, установленных по три на границе каждого сегмента корпоративной сети. Балансировка открытого корпоративного трафика на криптошлюзы осуществляется на передающих устройствах (ПУ 1 и 2), управляемых контроллером по протоколу OpenFlow. Контроллер располагается в первом сегменте корпоративной сети и связывается с передающим устройством второго сегмента при помощи отдельного защищенного канала связи. Между передающими устройствами созданы GRE-туннели в количестве, равном числу каналов, по которым происходит балансировка. В примере используются три канала. Каждый туннель проходит через «свою» пару криптошлюзов. GRE-туннели необходимы для создания L2-сегментов между передающими устройствами для передачи сообщений LLDP (Link Layer Discovery Protocol). Каждое передающее устройство периодически (в эксперименте — каждые 5 с) посылает во

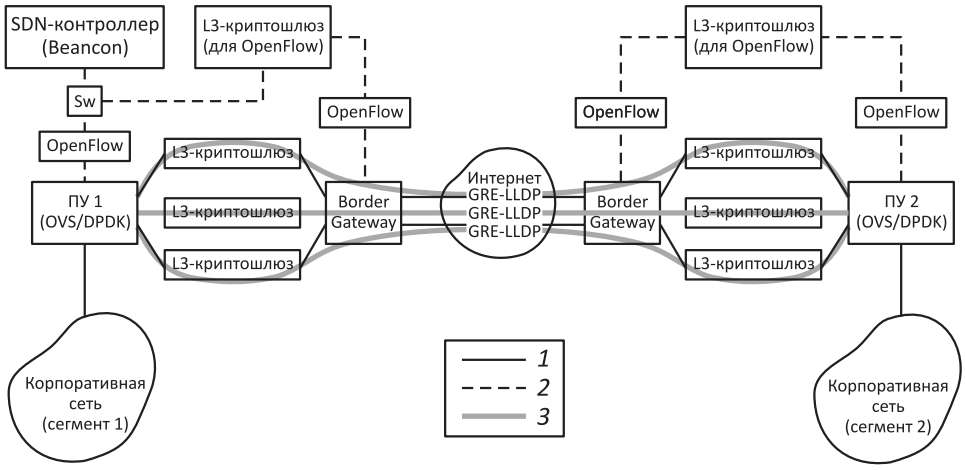


Рис. 2 Физическая и логическая топологии для алгоритма отказоустойчивости при обрыве канала связи: *1* — балансируемый корпоративный трафик; *2* — управляющий OpenFlow-трафик между контроллером и передающим устройствами; *3* — GRE-тоннели между передающими устройствами для передачи LLDP-сообщений

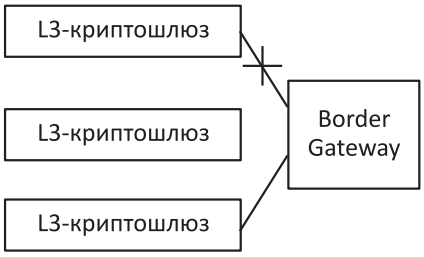


Рис. 3 Обрыв канала связи

все свои GRE-туннели LLDP-сообщения, в которых помимо прочей служебной информации содержатся идентификатор устройства и идентификатор порта, с которого было отправлено сообщение. Все полученные от «соседа» LLDP-сообщения передающее устройство пересылает на контроллер. Таким образом, контроллер в режиме реального времени «знает» состояние всех каналов связи между передающими устройствами.

Предположим, на одном из каналов произошла авария (рис. 3). В этом случае по одному из GRE-туннелей прекращается передача LLDP-сообщений. В результате контроллер не получает никакой новой информации об LLDP-трафике, ассоциированном с данным туннелем. После неполучения нескольких последовательных LLDP-сообщений (в эксперименте — двух) контроллер на всех передающих устройствах удаляет из таблицы потоков все правила, отправляющие трафик через порты, терминирующие отказавший канал связи. Далее потоки трафика, которые ранее коммутировались через отказавший канал, становятся «новыми» для передающего устройства и согласно стандартной логике SDN отправляются на контроллер для анализа. В итоге контроллер перенаправляет их через другие работоспособные каналы.

Описанное GRE-туннелирование выполняется только для LLDP-трафика. Полезный трафик корпоративной сети передается балансировщиком на криптошлюзы без какой-либо дополнительной инкапсуляции.

3.2 Отказоустойчивость контроллера

Контроллер является важнейшим и критичным звеном SDN-инфраструктуры, так как полностью определяет поведение передающих устройств. Контроллер использует L3-канал для управляющей связи с передающими устройствами. Следовательно, для резервирования контроллера в режиме Active-Passive имеет смысл использовать технологию с общим для всех серверов контроллера виртуальным IP-адресом, который будут «знать» все передающие устройства. На рис. 4 показана схема резервирования контроллера в режиме Active-Passive. Далее приведено описание варианта алгоритма отказоустойчивости.

В схеме, показанной на рис. 4, для резервирования контроллера в режиме Active-Passive используется VRRP (Virtual Router Redundancy Protocol, Linux Keepalived). Активный и пассивный контроллеры, передающее устройство и L3-криптошлюз, предназначенный для защиты управляющего OpenFlow-канала к другим географическим локациям, подключены к одной локальной сети 172.16.1.0/24. На интерфейсах активного и пассивного контроллера настраивается VRRP-группа, в которой помимо прочего определяются: приоритеты

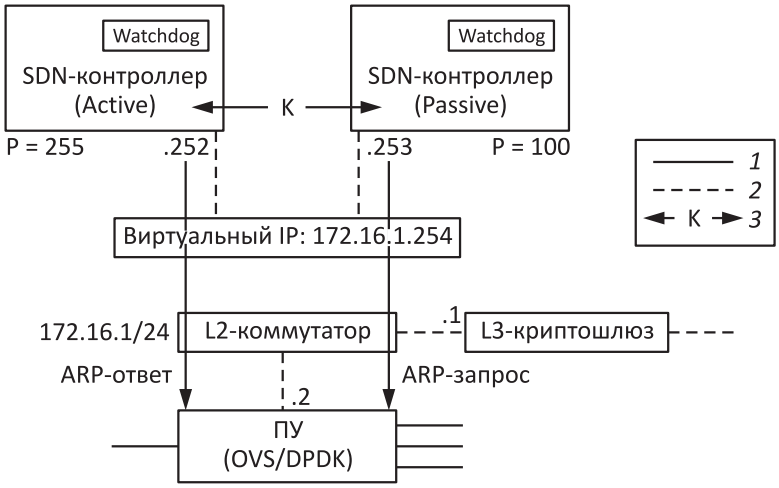


Рис. 4 Схема резервирования контроллера в режиме Active-Passive: *1* — корпоративный трафик; *2* — OpenFlow; *3* — VRRP Keepalived

контроллеров, общий виртуальный IP-адрес, таймеры. Активным (MASTER) становится контроллер с наибольшим приоритетом (на рис. 4 P = 255). Все передающие устройства для связи с контроллером используют виртуальный IP-адрес 172.16.1.254, а отвечает на ARP (Address Resolution Protocol) и OpenFlow запросы от передающих устройств именно активный контроллер. Также активный контроллер регулярно (по умолчанию каждую секунду) посылает пассивному сообщение Keepalive. В схеме на рис. 4 данное сообщение передается через те же интерфейсы, что и сообщения протокола OpenFlow. Это исключает ситуацию, когда управляющий OpenFlow-канал в результате аварии недоступен, а выделенный канал для сообщений Keepalive функционирует нормально, в результате чего аварийного переключения на пассивный контроллер не происходит. При неполучении пассивным контроллером нескольких последовательных сообщений Keepalive (по умолчанию в течение 3–4 с) последний становится активным, обновляет ARP-кэш передающих устройств и начинает отвечать на OpenFlow-сообщения от них. Для реализации отказоустойчивости помимо протокола VRRP на каждом сервере-контроллере используется специальный процесс (на схеме Watchdog), который регулярно отслеживает наличие в памяти и состояние процесса SDN-контроллера и при необходимости перезапускает его.

4 Алгоритм балансировки и результаты экспериментов

В данном разделе кратко рассмотрен алгоритм балансировки нагрузки, использованный во всех экспериментах, а также представлены результаты балан-

сировки на такие узлы защищенных сетей, как L3-криптошлюз, TLS-шлюз, IDS, в качестве которых использовались продукты и новые разработки компании ИнфоТеКС (ViPNet Coordinator HW, ViPNet TLS-Gateway и ViPNet IDS). Однако предложенные алгоритмы могут быть с тем же успехом применены и для оборудования других производителей.

4.1 Алгоритм балансировки нагрузки

Алгоритм балансировки нагрузки был реализован на языке программирования Java в виде модуля контроллера Beacon. Работа алгоритма начинается с чтения конфигурационного файла, в котором содержатся: IP-адреса устройств, на которые происходит балансировка; IP-адрес доступа самого балансировщика; метрики направлений балансировки. Данная операция выполняется один раз при инициализации модуля. Направлением балансировки может быть исходящий порт балансировщика или IP-адрес подключенного к нему устройства, требующего балансировки. Далее формируется однострочный массив, в котором каждое направление балансировки встречается столько раз, сколько указывает его метрика (по умолчанию 100). Чтобы распределение потоков трафика по направлениям балансировки было равномерным, среднее число новых потоков в секунду должно превышать сумму метрик всех направлений балансировки. В качестве идентификатора потока использовались четыре параметра: IP-адрес отправителя, IP-адрес получателя, транспортный порт отправителя, транспортный порт получателя.

При инициализации передающего устройства контроллер загружает в его таблицу потоков правило по умолчанию — все новые потоки отправлять на контроллер. Далее для каждого нового потока (при получении контроллером сообщения PACKET_IN) контроллер случайным образом выбирает из массива направление балансировки и инсталлирует новое правило в таблицу потоков передающего устройства. Каждое правило имеет таймер неактивности (Idle Timeout, в экспериментах — 60 с), благодаря которому правило автоматически удаляется из таблицы потоков при отсутствии трафика данного потока в течение 60 с. Дополнительно для каждого потока могут выполняться операции изменения содержимого полей служебных заголовков фрейма/пакета/сегмента, отправка потока в туннель GRE/VXLAN и др. В рамках алгоритма также был реализован механизм ARP-Responder, позволяющий генерировать ответы на ARP-запросы, приходящие на передающее устройство.

4.2 Балансировка нагрузки на L3-криптошлюзы

Рассмотрим результаты тестирования балансировки нагрузки на L3-криптошлюзы. На рис. 5 представлена схема стенда.

При проведении нагрузочного тестирования для генерации тестового трафика использовался инструмент Pktgen 3.1.2. Передающие устройства (ПУ 1 и 2)

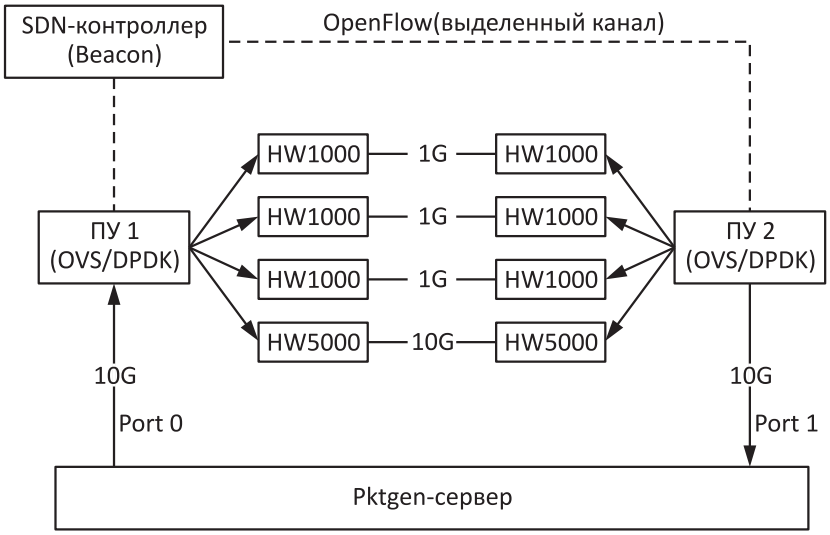


Рис. 5 Схема стенда для тестирования балансировки нагрузки на L3-криптошлюзы

подключены к Pktgen-серверу при помощи оптических 10G сетевых адаптеров Intel 82599ES, на которых активирована технология DPDK.

Обозначения HW1000 и HW5000 соответствуют L3-криптошлюзам ViPNet Coordinator HW1000 и ViPNet Coordinator HW5000 соответственно. Пропускная способность шифрованного канала HW1000, использованного в экспериментах, составляет 900–1000 МБ/с, HW5000 — 6–6,5 ГБ/с. Используемый алгоритм шифрования — ГОСТ28147-89 в режиме CBC (Cipher Block Chaining). Поскольку исследование производительности L3-криптошлюзов и прочих использованных в работе устройств защищенных сетей не входило в задачи данной статьи, подробная информация об их программно-аппаратном обеспечении и функционале здесь не приводится и частично может быть найдена в [9]. Тестовый UDP-трафик передавался в режиме Half-Duplex (порт-отправитель Pktgen-сервера — Port 0, порт-получатель — Port 1) и включал 2000 потоков в секунду, полученных путем перебора транспортного порта отправителя в диапазоне 10001–12000. Данного количества потоков было достаточно для тестирования равномерности балансировки нагрузки. Задача исследования максимальной производительности контроллера в данной работе не стояла и уже была решена в [8, 9]. Для алгоритма балансировки были выбраны следующие метрики направлений балансировки: HW1000 — метрика 10, HW5000 — метрика 60.

В таблице представлены результаты тестирования балансировки нагрузки на L3-криптошлюзы. Под пропускной способностью понимается объем тестового трафика в секунду, приходящего на порт 1 Pktgen-сервера после отправки

Результаты тестирования балансировки нагрузки на L3-криптошлюзы

L3-криптошлюзы, на которые направлена балансировка нагрузки (одна сторона защищенного канала)	Пропускная способность, МБ/с		
	UDP, MTU = 1432 Б	UDP, MTU = 256 Б	UDP, MTU = 64 Б
1 HW1000	933	461	180
1 HW5000	6241	1016	288
3 HW1000 (балансировка)	2806	1398	552
3 HW1000 + 1 HW5000 (балансировка)	**9041**	**2419**	**858**

с порта 0, балансировки на передающем устройстве ПУ 1, шифрования и расшифрования на криптошлюзах HW1000/5000 и сборки на передающем устройстве ПУ 2. В ходе всех тестов нагрузка на L3-криптошлюзы была максимально возможной. Загрузка процессоров криптошлюзов составляла 98%–100%.

В первой колонке таблицы перечислены L3-криптошлюзы, подключенные к одному передающему устройству. Передающее устройство на противоположном конце защищенного канала балансирует/принимает трафик на/от аналогичного набора L3-криптошлюзов (см. рис. 5). Как видно из таблицы, пропускная способность шифрованного канала при балансировке трафика на три криптошлюза HW1000 и один криптошлюз HW5000 практически равна сумме пропускных способностей отдельных криптошлюзов. Небольшие колебания значений вызваны спецификой работы инструмента Pktgen.

4.3 Балансировка нагрузки на TLS-шлюзы

На рис. 6 представлена схема стенда для тестирования балансировки нагрузки на TLS-шлюзы.

Схема на рис. 6 состоит из следующих основных компонентов:

– TLS-Meter — генератор тестового трафика, представляющего собой HTTPS-запросы, идущие на виртуальный адрес передающего устройства

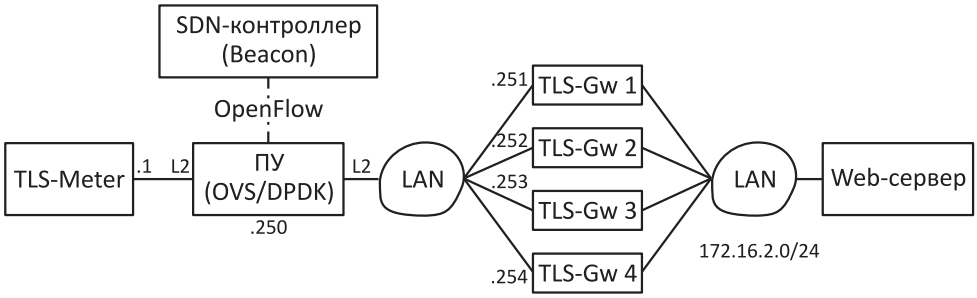


Рис. 6 Схема стенда для тестирования балансировки нагрузки на TLS-шлюзы

172.16.1.250. Данный инструмент разработан в компании ИнфоТеКС, в описываемых экспериментах использует алгоритм ГОСТ Р 34.10-2012;

- SDN-балансировщик, включающий передающее устройство и контроллер;
- TLS-шлюзы (TLS-Gw 1–4), на которые происходит балансировка HTTPS-запросов. Все четыре шлюза были представлены одной тестовой моделью, работающей в режиме HTTPS-Proxy с использованием алгоритма шифрования ГОСТ Р 34.10-2012;
- Web-сервер — веб-сервер, которому TLS-шлюзы адресуют расшифрованные HTTP-запросы.

Генерация тестового трафика выполнялась с 240 IP-адресов. Длительность теста — 3 мин, версия TLS-1.2. В ходе тестов измерялось количество установленных HTTPS-соединений в секунду в зависимости от числа TLS-шлюзов. Полученные результаты не являются предельными характеристиками, так как нагрузочное тестирование TLS-шлюза в задачи данной работы не входило. Замеры нагрузки производились на L2-коммутаторе, соединяющем передающее устройство и TLS-шлюзы, путем SPAN (Switched Port ANalyzer) зеркалирования всего проходящего через коммутатор трафика на выделенный персональный компьютер, сбора трафика при помощи tcpdump и анализа при помощи Wireshark. На рис. 7 показаны графики зависимости количества установленных HTTPS-соединений от времени теста для каждого TLS-шлюза, а также суммарная нагрузка.

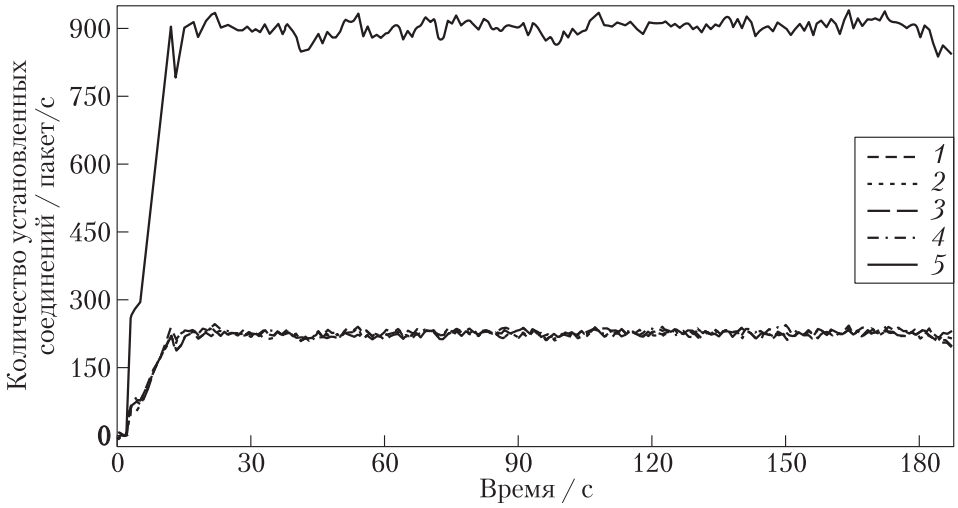


Рис. 7 График зависимости количества установленных HTTPS-соединений от времени теста для каждого TLS-шлюза (*1–4* — TLS-Gw 1–TLS-Gw 4) и суммарная балансируемая нагрузка (*5*)

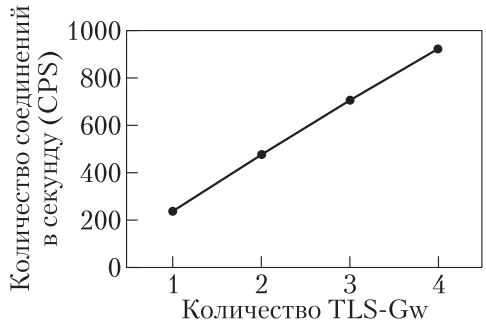


Рис. 8 График зависимости количества установленных HTTPS-соединений в секунду от числа TLS-шлюзов

Как видно на рис. 7, производительность каждого отдельного TLS-шлюза в ходе всего теста составляла приблизительно 235 соединений в секунду. Для всех четырех TLS-шлюзов наблюдались равномерность и одинаковость загрузки, что говорит о корректной балансировке HTTPS-трафика. Суммарная производительность системы составила четырехкратное увеличение производительности одного TLS-шлюза.

На рис. 8 показан график зависимости количества установленных HTTPS-соединений в секунду от числа TLS-шлюзов, нагрузка на которые проходила балансировку. График, представленный на рис. 8, имеет почти линейный характер, что говорит о том, что количество направлений балансировки практически не влияет на производительность балансировщика. Поскольку в тестах использовались TLS-шлюзы одинаковой производительности, метрики направлений балансировки имели одинаковые значения — 100. В данном эксперименте для реализации отказоустойчивости каналов связи, по которым происходила балансировка, использовался следующий алгоритм. Каждые 2 с контроллер генерирует с передающего устройства ARP-запросы на IP-адреса всех TLS-шлюзов (172.16.1.251–172.16.1.254). В случае неполучения двух последовательных ARP-ответов от одного TLS-шлюза данное направление балансировки считается нефункциональным и исключается из алгоритма балансировки. При этом отправка ARP-запросов на IP-адрес «выбывшего» TLS-шлюза продолжается, и в случае получения ARP-ответа данное направление балансировки снова начинает использоваться для передачи полезного трафика.

4.4 Балансировка нагрузки на системы обнаружения вторжений

На рис. 9 представлена схема стенда для тестирования балансировки нагрузки на масштабируемую систему обнаружения вторжений (IDS).

В схеме на рис. 9 полезный трафик зеркалируется на L2-коммутаторе и отправляется на передающее устройство SDN-балансировщика, где равномерно балансируется на три системы обнаружения вторжений, максимальная производительность каждой из которых составляет 6 ГБ/с (использовалась модель ViPNet IDS2000). Особенностью алгоритма балансировки для данного случая является то, что при обработке каждого нового потока контроллер инсталлирует в таблицу потоков передающего устройства два правила коммутации: прямое и обратное. В прямом правиле IP-адреса и транспортные порты отправите-

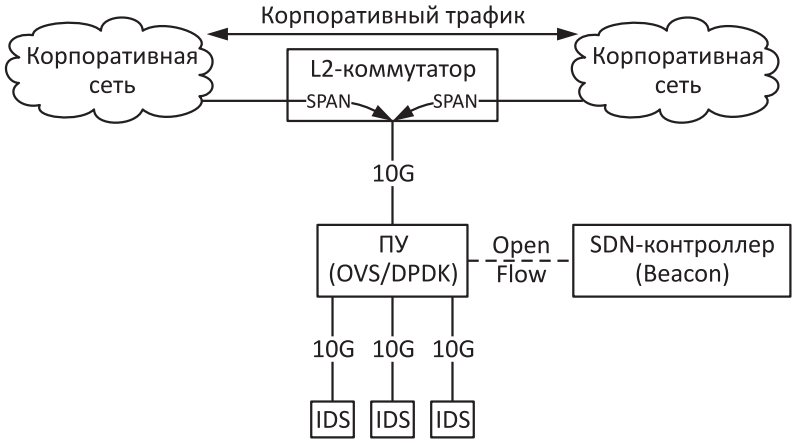


Рис. 9 Схема стенда для тестирования балансировки нагрузки на масштабируемую систему обнаружения вторжений

ля и получателя соответствуют полученным из первого пакета нового потока. В обратном правиле идентификаторы отправителя и получателя меняются местами. Оба правила используют одно и то же направление балансировки. Таким образом гарантируется, что весь трафик одной TCP/UDP-сессии будет анализироваться одной IDS.

Кратко рассмотрим результаты эксперимента. В ходе теста загрузка сетевых интерфейсов всех IDS была практически одинаковой и синхронно менялась при изменении объема подаваемого на передающее устройство трафика. Анализ трафика, приходящего на каждую IDS после балансировки, показал отсутствие «разорванных» TCP-сессий, что говорит о том, что контроллер корректно распределяет потоки трафика по направлениям балансировки.

5 Заключение

Технология SDN благодаря централизованному управлению сетью и программируемости хорошо подходит для решения задач балансировки трафика при необходимости согласованной работы балансировщиков, отказоустойчивости и нестандартных алгоритмов балансировки. Open vSwitch при использовании технологии Intel DPDK успешно коммутирует не менее 10 ГБ/с Full-Duplex-трафика. Предложены и частично протестированы механизмы отказоустойчивости каналов связи, по которым происходит балансировка (режим Active-Active), и SDN-контроллера (режим Active-Passive). Успешно проведены экспериментальные исследования балансировки нагрузки на такие устройства защищенных сетей, как L3-криптошлюз, TLS-шлюз, IDS. Во всех экспериментах был ис-

пользован унифицированный алгоритм балансировки, использующий весовые коэффициенты каналов.

Литература

1. *Feamster N., Balakrishnan H.* Detecting BGP configuration faults with static analysis // 2nd Conference on Symposium on Networked Systems Design and Implementation Proceedings, 2005. Vol. 2. P. 43–56.
2. *Sherry J., Ratnasamy S.* A survey of enterprise middlebox deployments. — Berkeley, CA, USA: Univ. California, Electr. Eng. Comput. Sci. Dept., 2012. Technical Report UCB/EECS-2012-24.
3. *Kim H., Feamster N.* Improving network management with software defined networking // IEEE Commun. Mag., 2013. Vol. 51. No. 2. P. 114–119.
4. *Kreutz D., Ramos F. M. V., Verissimo P., Rothenberg C. E., Azodolmolky S., Uhlig S.* Software-defined networking: A comprehensive survey // P. IEEE, 2015. Vol. 103. No. 1. P. 14–76.
5. *Назаров М. А.* Определение, основные понятия и архитектуры программно-конфигурируемых сетей — SDN (Software Defined Networking) // Информатизация и связь, 2015. № 4. С. 82–87.
6. *Greene K.* TR10: Software-defined networking // 10 Breakthrough Technologies: MIT Technology Review, 2009. http://www2.technologyreview.com/article/412194/tr10-software-defined-networking.
7. *Erickson D.* The Beacon OpenFlow controller // 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking Proceedings, 2013. P. 13–18.
8. *Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R.* Advanced study of SDN/OpenFlow controllers // 9th Central & Eastern European Software Engineering Conference in Russia Proceedings, 2013. Article No. 1. https://www.researchgate.net/publication/262155093_Advanced_study_of_SDNOpenFlow_controllers/overview.
9. Infotecs: Продукты. http://infotecs.ru/product/all/?line=vipnet-network-security.

Поступила в редакцию 26.11.17

SDN LOAD BALANCING FOR SECURE NETWORKS

***O. Yu. Guzev*[1] *and I. V. Chizhov*[2,3]**

[1]Research and Development Center, JSC "InfoTeCS," 1/23, b. 1 Staryy Petrovsko-Razumovskiy Pr., Moscow 127287, Russian Federation
[2]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 2nd Education Building, Faculty CMC, GSP-1, Leninskie Gory, Moscow 119991, Russian Federation
[3]Institute of informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The software-defined networking (SDN) technology in comparison with traditional IP networks allows programming the network's behavior using a centralized controller. In this case, forwarding devices deal only with forwarding frames based on flow tables loaded into them by the controller. Flow tables are built on the controller during the processing of information about traffic flows arriving at forwarding devices. The above properties of the technology were used to create the SDN load balancer for devices of secure networks. The article discusses the architecture and software of the balancer. Descriptions of schemes and results of experiments on load balancing for such devices as L3-VPN (Level 3 Virtual Private Network) gateway, TLS (Transport Layer Security) gateway, and IDS (Intrusion Detection System) are given.

Keywords: software-defined networking (SDN); controller; OpenFlow; VPN gateway; TLS; intrusion detection system; IDS; load balancing; DPDK; Open vSwitch; Beacon

DOI: 10.14357/08696527180101O

References

1. Feamster, N., and H. Balakrishnan. 2005. Detecting BGP configuration faults with static analysis. *2nd Conference on Symposium on Networked Systems Design and Implementation Proceedings*. 2:43–56.
2. Sherry, J., and S. Ratnasamy. 2012. A survey of enterprise middlebox deployments: Berkeley, CA: Univ. California, Electr. Eng. Comput. Sci. Dept. Technical Report UCB/EECS-2012-24.
3. Kim, H., and N. Feamster. 2013. Improving network management with software defined networking. *IEEE Commun. Mag.* 51(2):114–119.
4. Kreutz, D., F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. 2015. Software-defined networking: A comprehensive survey. *P. IEEE* 103(1):14–76.
5. Nazarov, M. A. 2015. Opredelenie, osnovnye ponyatiya i arkhitektury programmno-konfiguriruemykh setey — SDN (Software Defined Networking) [Determination, the

basic concepts, and architecture of the program defined networks]. *Informatizatsiya i svyaz'* [Informatization and Communication] 4:82–87.

6. Greene, K. 2009. TR10: Software-defined networking. *10 Breakthrough Technologies: MIT Technology Review*. Available at: http://www2.technologyreview.com/article/412194/tr10-software-defined-networking (accessed February 24, 2018).
7. Erickson, D. 2013. The Beacon OpenFlow controller. *2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking Proceedings*. 13–18.
8. Shalimov, A., D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky. 2013. Advanced study of SDN/OpenFlow controllers. *9th Central & Eastern European Software Engineering Conference in Russia Proceedings*. Article No. 1. Available at: https://www.researchgate.net/publication/262155093_Advanced_study_of_SDNOpenFlow_controllers/overview (accessed February 24, 2018).
9. Infotecs. Produkty [Products]. Available at: http://infotecs.ru/product/all/?line=vipnet-network-security (accessed February 24, 2018).

Received November 26, 2017

Contributors

Guzev Oleg Yu. (b. 1980) — Candidate of Science (PhD) in technology, researcher, Research and Development Center, JSC "InfoTeCS," 1/23, b. 1, Staryy Petrovsko-Razumovskiy Pr., Moscow 127287, Russian Federation; oleg.guzev@infotecs.ru

Chizhov Ivan V. (b. 1984) — Candidate of Science (PhD) in physics and mathematics, associate professor, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 2nd Education Building, Faculty CMC, GSP-1, Leninskie Gory, Moscow 119991, Russian Federation; senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ichizhov@cs.msu.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 139–155

SDN-БАЛАНСИРОВКА НАГРУЗКИ НА КРИПТОГРАФИЧЕСКИЕ МАРШРУТИЗАТОРЫ ПРИ ОБЪЕДИНЕНИИ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

***О. Ю. Гузев*[1], *И. В. Чижов*[2]**

Аннотация: Работа посвящена решению задачи балансировки нагрузки на криптографические маршрутизаторы, предназначенные для защиты каналов связи между центрами обработки данных (ЦОД). Предлагается метод балансировки, основанный на использовании технологии программно-конфигурируемых сетей (SDN, software-defined networking). Рассмотрены основные составные компоненты метода, начиная от базовой обобщенной модели криптографического маршрутизатора и заканчивая полным детальным описанием алгоритма балансировки. Также в завершение работы представлен алгоритм обеспечения отказоустойчивости в режиме Active-Active масштабируемого канала связи между ЦОД.

Ключевые слова: криптографический маршрутизатор; криптомаршрутизатор; балансировка нагрузки; эластичный центр обработки данных; программно-конфигурируемая сеть; ПКС; SDN; OpenFlow

DOI: 10.14357/08696527180111

1 Введение

В ряде случаев для защиты каналов связи между ЦОД используются средства шифрования сетевого трафика. При этом для обеспечения прохождения данных между сегментами глобальной открытой сети применяются шифраторы, работающие на сетевом (L3) уровне эталонной модели ISO/OSI (International Standard Organization/Open Systems Interconnection) и имеющие возможность выполнять маршрутизацию шифрованного трафика. В данной работе такие устройства будем называть криптографическими маршрутизаторами, криптомаршрутизаторами или просто криптографическими устройствами. В силу того что такие устройства теряют часть пропускной способности за счет накладных расходов на шифрование и расшифрование трафика, применяются технологии масштабирования защищенных каналов связи, во-первых, для увеличения пропускной

[1]Центр научных исследований и перспективных разработок, ОАО «ИнфоТеКС», oleg.guzev@infotecs.ru

[2]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики; Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ichizhov@cs.msu.ru

способности канала связи и, во-вторых, для повышения его отказоустойчивости. Также для объединения ЦОД часто применяют различные технологии L2-туннелирования трафика канального уровня (ARP (Address Resolution Protocol), LLDP (Link Layer Discovery Protocol) и др.), например такие, как GRE (Generic Routing Encapsulation) [1–5] и VXLAN (Virtual eXtensible Local Area Network) [6].

В открытой литературе практически отсутствуют работы, посвященные вопросам балансировки нагрузки на произвольные криптографические маршрутизаторы. При этом некоторые производители шифровального оборудования предлагают проприетарные балансировщики нагрузки, решающие задачу для конкретного типа криптографических устройств. Также встречаются работы, например [7], посвященные вопросам создания кластеров криптомаршрутизаторов, работающих на основе протокола IPsec. Однако существуют криптографические устройства, использующие другие защищенные протоколы, в том числе и не имеющие открытого описания. В работе ставится задача организации балансировки нагрузки на произвольные криптографические маршрутизаторы независимо от используемых технологий защиты информации.

2 Базовая модель обобщенного криптографического маршрутизатора

Опишем базовую модель криптографического маршрутизатора как сетевого устройства, которой удовлетворяет подавляющее большинство подобного рода устройств, в том числе и использующих протокол IPsec.

1. Криптографический маршрутизатор обрабатывает (шифрование, маршрутизация) только трафик сетевого уровня. Никакой трафик канального уровня не может преодолеть криптографический маршрутизатор.
2. Криптомаршрутизаторы со стороны локальной сети «внутренними» интерфейсами физически подключаются к портам коммутатора уровня распределения (балансировщика нагрузки).
3. Между криптомаршрутизаторами можно создавать виртуальные защищенные сети. Между двумя виртуальными сетями прохождение трафика полностью исключается. Внутри одной виртуальной сети трафик в зашифрованном виде может свободно передаваться от одного криптографического устройства к другому.
4. Пусть на входном интерфейсе криптомаршрутизатора со стороны локальной сети получен сетевой пакет с IP-адресом назначения IP_dst. Криптомаршрутизатор по таблице маршрутизации проверяет, за каким другим криптомаршрутизатором его виртуальной сети находится хост с адресом IP_dst. Если такого криптомаршрутизатора нет, то пакет отбрасывается. В противном случае пакет шифруется на соответствующем ключе и отправляется нужному криптомаршрутизатору.

5. При получении зашифрованного пакета криптомаршрутизатор делает попытку его расшифровать. Если это сделать не удается, то пакет отбрасывается. После расшифрования определяется IP-адрес назначения пакета. Если этот адрес доступен со стороны локальной сети, то пакет отправляется получателю, если нет — отбрасывается.
6. На криптомаршрутизаторе можно явно настроить список сетевых адресов, которые находятся в локальной сети «за этим устройством» и к которым криптомаршрутизатор обеспечивает доступ для других криптомаршрутизаторов. В рамках одной виртуальной сети на всех криптомаршрутизаторах эти адреса не должны пересекаться для обеспечения однозначности маршрутизации шифрованного трафика.

Для оптимизации работы алгоритма балансировки и использования сетевых адресов необходимо унифицировать параметры настройки всех криптомаршрутизаторов одного ЦОД, одной виртуальной сети, включая адреса сетевых интерфейсов. Единственное различие, которое допускается (но не требуется) между ними, заключается в наборе криптографических ключей связи (ключей шифрования) с устройствами других ЦОД. Однако строго потребуем, чтобы ключи связи любых взаимодействующих друг с другом криптографических устройств были различны.

С учетом описанной модели криптографического маршрутизатора можно сформулировать следующие требования к методу балансировки нагрузки.

1. Необходимо обеспечить прохождение трафика из одного ЦОД в другой с учетом того, что взаимодействующие хосты могут мигрировать между ЦОД. Другими словами, требуется поддержка так называемых эластичных ЦОД.
2. Необходимо реализовать механизм отказоустойчивости каналов связи между ЦОД.
3. Необходимо выполнять балансировку открытого трафика ЦОД до туннелирования GRE/VXLAN и шифрования. Это гарантирует максимальную равномерность балансировки и целостность пользовательских сессий.

3 Подход к решению задачи балансировки нагрузки

Опишем основную идею предлагаемого подхода. Уже отмечалось, что распределять инкапсулированный трафик не имеет большого смысла, поэтому необходимо:

(1) привязать каждый GRE/VXLAN-туннель к паре последовательно связанных криптографических маршрутизаторов;

(2) распределять трафик, идущий из локальной сети, по соответствующим туннелям, подменяя IP-адреса получателя и отправителя.

PRIORITY	IN_PORT	ETH_TYPE	IP_PROTO	IP_DST	IP_SRC	actions
0	any	any	any	any	any	controller

Рис. 1 Пример OF-таблицы

Исходя из этих положений, необходимо обеспечить выполнение всех требований к балансировщику нагрузки. Предлагается для проработки технических деталей использовать технологию SDN. Условно SDN состоит из двух центральных компонентов [8]: SDN-контроллера и простейшего передающего SDN-коммутатора. При этом SDN-контроллер играет роль основного «мозга» сети. Он управляет SDN-коммутаторами (далее — коммутатор, передающее устройство). Взаимодействие между контроллером и коммутатором может происходить по достаточно широкому набору протоколов. Остановимся на одном из самых популярных — протоколе OpenFlow (OF) [9]. Коммутаторы, понимающие язык OF, в дальнейшем будем называть OF-коммутаторами или просто коммутаторами, если из контекста ясно, что речь идет именно об OF-коммутаторе. В каждом OF-коммутаторе имеются OF-таблицы вида, представленного на рис. 1.

Таблица, показанная на рис. 1, включает OF-записи (OF-правила). Каждая запись имеет приоритет (в дальнейшем столбец «приоритет» будет опущен и приоритет записей будет определяться по правилу: чем ниже запись в таблице, тем ее приоритет выше). Все столбцы, кроме последнего «actions», являются идентификаторами пакетов в одном потоке. Последний столбец определяет действие, которое должен выполнить OF-коммутатор над пакетами данного потока. Из действий особо отметим два: controller — обратиться за указаниями на контроллер, а также число N — отправить в порт с номером N. Остальные действия будут рассмотрены в ходе изложения деталей алгоритма. Будем предполагать, что таблицы нумеруются от 0, в таблице с номером 0 имеется запись с нулевым приоритетом, которая отправляет весь трафик на контроллер (см. рис. 1).

Перед описанием алгоритмов работы балансировщика нагрузки рассмотрим частный пример объединения нескольких ЦОД в единый сетевой сегмент.

4 Объединение центров обработки данных в единый сетевой сегмент

Рассмотрим классическую задачу объединения нескольких ЦОД. Пусть имеются три ЦОД. Необходимо объединить их таким образом, чтобы они находились в едином сегменте локальной сети. Для решения этой задачи используются технологии инкапсуляции сетевого трафика уровня L2. При такой инкапсуляции Ethernet-фреймы прозрачно для сетевых устройств преодолевают маршрутизаторы сети Интернет и попадают в другой ЦОД в неизменном виде. Таким образом создается впечатление, что сетевые устройства, расположенные в разных ЦОД, находятся в одном сегменте локальной сети.

В качестве протоколов инкапсуляции обычно используются либо протокол GRE, либо протокол VXLAN. В этом случае иногда говорят, что между гра-

ницами ЦОД создаются GRE- или VXLAN-туннели. Рассмотрим возможную реализацию такого объединения средствами технологии SDN и программных OF-коммутаторов. Для построения схемы используются:

(1) OF-коммутаторы с поддержкой технологии туннелирования по протоколу GRE или VXLAN, а также с поддержкой протокола OF версии 1.3;

(2) SDN-контроллер с поддержкой протокола OF версии 1.3.

Точкой создания туннеля со стороны ЦОД является выходной интерфейс OF-коммутатора, поэтому через граничные маршрутизаторы каждого ЦОД проходит только GRE/VXLAN-трафик. На граничном OF-коммутаторе выделяется один порт, к которому подключается внутренняя локальная сеть. Для простоты будем считать, что этот порт имеет номер 01, и в дальнейшем будем называть такой порт портом доступа и обозначать как eth01. Предполагается, что все конечные устройства локальной сети сначала подключены к некоторому коммутатору уровня доступа, использующему для коммутации Ethernet-фреймов MAC-таблицу, а этот коммутатор, в свою очередь, уже подключен к порту eth01 граничного OF-коммутатора, упомянутого выше. Для простоты коммутатор уровня доступа на всех рисунках не показан. Также в граничном OF-коммутаторе выделяется специальный порт, который выполняет инкапсуляцию/декапсуляцию Ethernet-фреймов, идущих из/в локальную сеть соответственно. Этот порт будем называть туннельным и считать, что туннельный порт имеет номер 02. Для создания туннелей потребуется также IP-интерфейс на OF-коммутаторе. Договоримся, что этот интерфейс будет иметь номер 03, называть такой интерфейс будем сетевым. Порт OF-коммутатора, к которому подключается граничный маршрутизатор ЦОД, будем называть внешним (по умолчанию номер 04).

Также на каждом OF-коммутаторе предполагается наличие виртуальных портов veth04.1 и veth04.2. Данные виртуальные порты реально не существуют в конфигурациях OF-коммутаторов, однако они будут использоваться в OF-таблицах. Каждый виртуальный порт отвечает за связь между парой ЦОД. Так, будем считать, что между собой связаны следующие виртуальные порты:

– veth04.1 OF-коммутатора ЦОД 1 и veth04.1 OF-коммутатора ЦОД 2;

– veth04.2 OF-коммутатора ЦОД 1 и veth04.2 OF-коммутатора ЦОД 3;

– veth04.2 OF-коммутатора ЦОД 2 и veth04.1 OF-коммутатора ЦОД 3.

Основные перечисленные порты/интерфейсы OF-коммутатора ЦОД 1 показаны на рис. 2.

Вся логика организации связи полностью ложится на SDN-контроллер, который, устанавливая OF-правила в таблицы OF-коммутаторов, управляет потоками данных для обеспечения корректной связности между ЦОД. В описании схемы коммутации будем предполагать, что связь между каждыми двумя ЦОД осуществляется напрямую, т. е. обрыв канала между парой ЦОД не предполагает изменения пути прохождения трафика через третий ЦОД. Другими словами, не

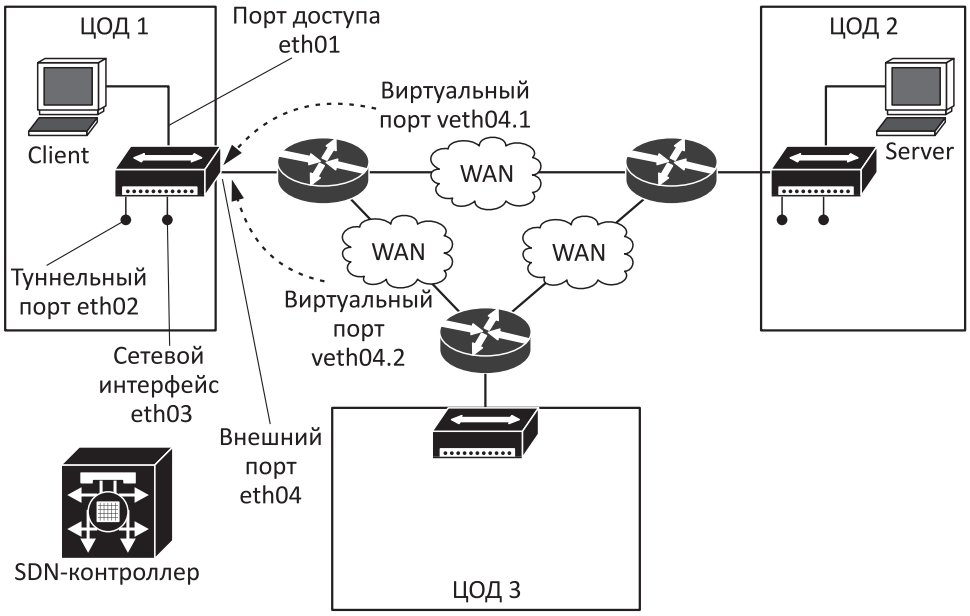


Рис. 2 Физические и виртуальные порты/интерфейсы OF-коммутатора ЦОД 1

будем ставить целью обеспечение отказоустойчивости связи между двумя ЦОД за счет связей с третьим ЦОД. Это допущение позволит упростить дальнейшее описание, а для изложения технологии балансировки оно не является существенным.

GRE-туннель будет создаваться между сетевыми интерфейсами OF-коммутаторов разных ЦОД. Одна граница GRE-туннеля будет иметь сетевой адрес 172.16.101.2/24 (ЦОД 1), а вторая — 172.16.102.2/24 (ЦОД 2). Адреса криптомаршрутизаторов (шлюзов) в этом случае будут 172.16.101.1/24 и 172.16.102.1/24 соответственно.

Рассмотрим базовый алгоритм работы SDN-контроллера на примере прохождения ICMP (Internet Control Message Protocol, протокол межсетевых управляющих сообщений) трафика между рабочей станцией client в ЦОД 1 и сервером server в ЦОД 2. В каждом OF-коммутаторе будут использоваться три таблицы потоков: 0, 1 и 2. Таблица 0 будет служить для классификации трафика и его распределения между криптомаршрутизаторами, таблицы 1 и 2 — для обработки трафика, соответственно исходящего и входящего в локальную сеть. Алгоритм заполнения ARP-таблиц L3-устройств здесь не приводится и подразумевается, что все необходимые ARP-записи сформированы.

1. В начальный момент времени на рабочей станции client пользователь вводит команду ping server.

2. Предполагаем, что раньше связи между двумя узлами не было и рабочая станция обращается к серверу впервые. В этом случае рабочая станция для формирования Ethernet-фрейма должна узнать MAC-адрес сервера по его IP-адресу. Следовательно, с рабочей станции отправляется широковещательный ARP-запрос, который попадает на OF-коммутатор ЦОД 1.

3. В начальный момент времени в таблице 0 OF-коммутатора стоит правило, отправляющее первый пакет каждого нового потока на контроллер. Коммутатор воспринимает ARP-запрос от рабочей станции как новый поток и поэтому обращается к контроллеру для получения указаний о том, как обработать данный новый поток.

4. Контроллер, получив информацию о новом потоке, определяет, что адрес получателя ему неизвестен, поэтому на коммутатор он должен передать команду отправить этот поток во все порты, кроме входящего, т. е. запустить процесс изучения топологии. Однако в случае использования туннелей недостаточно отправить трафик во все порты при помощи таблицы 0. Контроллер указывает OF-коммутатору ЦОД 1 с помощью правила resubmit запустить процедуру обработки трафика по таблице с номером 1. Таким образом, контроллер отсылает команду обработать поток по таблице 1, считая, что он пришел последовательно из всех виртуальных портов: eth04.1 и eth04.2. В результате пакет будет отправлен по всем возможным направлениям: в ЦОД 2 и в ЦОД 3.

5. Получив команду от контроллера, OF-коммутатор ЦОД 1 отправляет трафик в таблицу 1. Для каждого порта с номером N в этой таблице существует запись, которая указывает, что если поток пришел в порт с номером N (напомним, что в таблице 1 это виртуальные порты), то этот поток должен быть отправлен в GRE/VXLAN-порт eth02 с заданными параметрами туннеля. Пакет после попадания в туннельный порт будет упакован, и новый туннельный пакет должен быть отправлен через сетевой интерфейс eth03. Однако для формирования Ethernet-фрейма недостаточно информации, а именно: нет указания, какой использовать аппаратный адрес получателя. В итоге от имени адреса 172.16.101.2 формируется ARP-запрос к 172.16.102.2 (либо к 172.16.103.2). Этот запрос с подменой целевого адреса поступает на граничный маршрутизатор, который, в свою очередь, посылает ARP-ответ, обрабатываемый на коммутаторе при помощи таблицы 0. В итоге на коммутаторе ЦОД 1 формируется Ethernet-фрейм и отправляется в выходной порт eth04.

6. Туннельный трафик через глобальную сеть попадает на граничные маршрутизаторы ЦОД 2 и ЦОД 3 и затем на туннельный порт eth02 граничного OF-коммутатора каждого ЦОД. Далее будем рассматривать только ЦОД 2, так как в ЦОД 3 события будут происходить аналогично за тем лишь исключением, что широковещательный ARP-запрос к server «умрет» в локальной

сети ЦОД 3 в силу того, что на него никто не ответит. OpenFlow-коммутатор ЦОД 2 в соответствии с таблицей 0 отправляет декапсулированный GRE-трафик на обработку в таблицу с номером 2.

7. Первоначально в таблице 2 стоит единственная запись, указывающая считать каждый приходящий пакет началом нового потока и отправлять его на анализ контроллеру. Таким образом, после попадания из туннеля в соответствующий порт таблицы 2 пакет будет отправлен для анализа на контроллер.
8. Контроллер, получив пакет, определяет MAC-адрес отправителя и фиксирует в своей MAC-таблице, что этот адрес доступен за таким-то виртуальным портом (veth04.1 или veth04.2) OF-коммутатора ЦОД 2. Порт определяется на основании информации, полученной от OF-коммутатора, в которой указывается, в какой конкретно порт таблицы 2 пришел пакет нового потока. В примере пакет с MAC-адресом client придет в порт 4.1 таблицы 2, поэтому контроллер зафиксирует запись client:4.1.
9. Контроллер передает на OF-коммутатор ЦОД 2 следующие команды: отправить полученный пакет в выходной порт eth01, в таблице 0 установить запись, предписывающую отправлять весь трафик с MAC-адресом получателя client в порт 4.1 таблицы 1, которая связана с внешним портом eth04, отвечающим за связь с ЦОД 1, а в таблице 2 установить запись, на основании которой OF-коммутатор должен отправлять весь трафик, приходящий с аппаратного адреса client на адрес server, в порт eth01 локальной сети. Таким образом, последнее правило предотвратит обращения к контроллеру для всех последующих пакетов данного потока. Отметим, что правило должно выставляться с таймером неактивности для отслеживания изменения топологии сети.
10. В результате ARP-запрос от клиента попадет в локальную сеть ЦОД 2 и потом серверу.
11. Сервер отправит клиенту ARP-ответ, который вернется на OF-коммутатор ЦОД 2.
12. Полученный ARP-ответ OF-коммутатор может обработать, так как в таблице 0 уже имеется запись, соответствующая ARP-ответу: MAC-адрес получателя в ответе равен client. Эта запись диктует отправить пакет на обработку в порт 4.1 таблицы 1. И далее, согласно правилам таблицы 1, пакет будет инкапсулирован и отправлен в ЦОД 1.
13. В конечном счете ARP-ответ поступит на OF-коммутатор ЦОД 1. Для этого ARP-ответа повторится ровно та же последовательность событий, что и для ARP-запроса на OF-коммутаторе ЦОД 2.
14. ICMP-запрос от клиента в сторону сервера, как и ICMP-ответ от сервера, уже не вызовут обращений к контроллеру, так как на каждом OF-коммутаторе уже появятся правила обработки данных пакетов.

Далее приступим к описанию технических деталей алгоритма балансировки трафика на криптографические маршрутизаторы.

5 Алгоритм балансировки трафика, направляемого на криптомаршрутизаторы

Описание технических деталей будем вести на основании примера, приведенного на рис. 2, а также рассмотренной выше схемы IP-адресации. Конфигурация OF-коммутаторов будет мало отличаться от описанной ранее. Добавится дополнительный порт для подключения криптомаршрутизатора. Сетевые настройки и имена интерфейсов показаны на рис. 3.

Таблица с номером 0 в части обработки GRE-трафика после инициализации OF-коммутатора ЦОД 1 имеет вид, представленный на рис. 4.

Частично поясним записи таблицы. В строках 2–5 в адресах IP_DST вида 172.16.M.P буква M означает 100+номер ЦОДа, P — номер порта, в который подключен криптомаршрутизатор. При отправке GRE-трафика в глобальную сеть подменяется IP-адрес назначения GRE-туннеля. Это сделано для упрощения настройки криптомаршрутизаторов. Последние две строки таблицы предназначены для обработки поступающего на коммутатор ЦОДа 1 туннельного трафика.

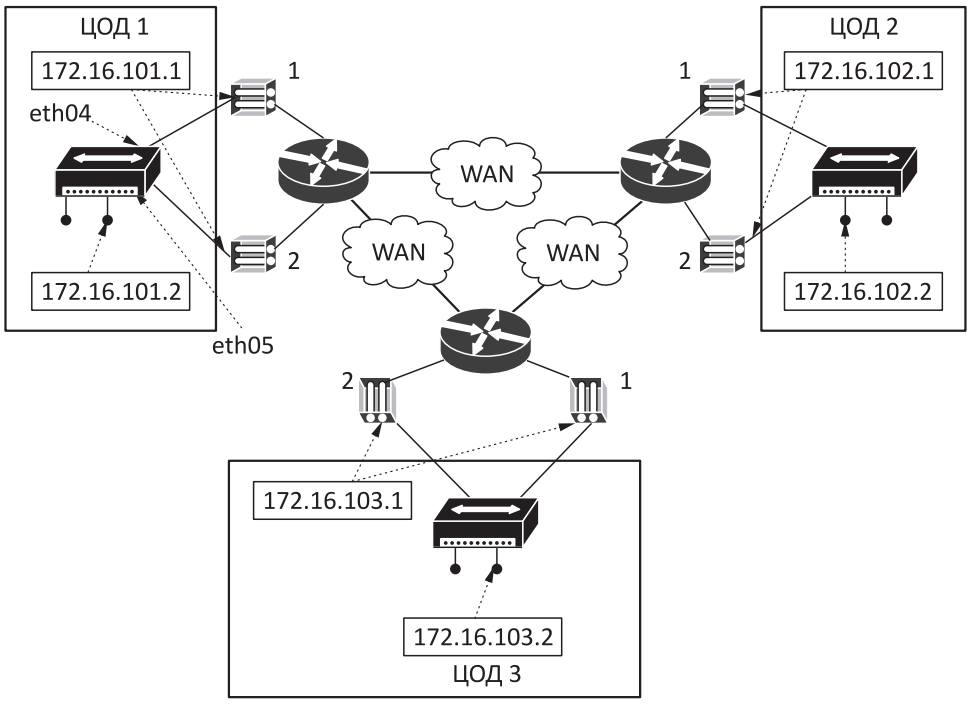


Рис. 3 IP-адресация в схеме балансировки трафика, направляемого на криптографические маршрутизаторы

IN_PORT	ETH_TYPE	IP_PROTO	IP_DST	IP_SRC	actions
any	any	any	any	any	controller
eth03	IP	GRE	172.16.102.4	any	set_field: 172.16.102.2 → ip_dst eth04
eth03	IP	GRE	172.16.102.5	any	set_field: 172.16.102.2 → ip_dst eth05
eth03	IP	GRE	172.16.103.4	any	set_field: 172.16.103.2 → ip_dst eth04
eth03	IP	GRE	172.16.103.5	any	set_field: 172.16.103.2 → ip_dst eth05
eth04	IP	GRE	any	any	eth02
eth02	IP	GRE	any	172.16.102.2	resubmit(2, 4.1)
eth02	IP	GRE	any	172.16.103.2	resubmit(2, 4.2)

Рис. 4 Часть таблицы потоков 0, отвечающая за обработку GRE-трафика

IN_PORT	ETH_TYPE	ARP_OP	ARP_SPA	ARP_TPA	actions
eth03	ARP	REP	172.16.101.2	172.16.101.1	eth04
eth04	ARP	REQ	172.16.101.1	172.16.101.2	eth03

Рис. 5 Часть таблицы потоков 0, отвечающая за обработку ARP-трафика

Часть таблицы 0, отвечающая за обработку ARP-трафика (рис. 5), обеспечивает передачу ARP-запросов от криптомаршрутизаторов к сетевому интерфейсу eth03 OF-коммутатора, а также обратную передачу ARP-ответов.

Таблица с номером 1 будет иметь вид, представленный на рис. 6.

Дадим некоторые пояснения. Для управления прохождением трафика через криптомаршрутизатор будем использовать различные адреса удаленного интерфейса GRE-туннеля. Таким образом, IP-адрес отправителя однозначно определяет ЦОД-отправитель, а IP-адрес получателя определяет ЦОД-полу-

IM_PORT	ETH_TYPE	IP_PROTO	actions
any	any	any	conroller
2.4	any	any	set_field: 172.16.101.2 → tun_src set_field: 172.16.102.4 → tun_dst out: 2
2.5	any	any	set_field: 172.16.101.2 → tun_src set_field: 172.16.102.5 → tun_dst out: 2
3.4	any	any	set_field: 172.16.101.2 → tun_src set_field: 172.16.103.4 → tun_dst out: 2
3.5	any	any	set_field: 172.16.101.2 → tun_src set_field: 172.16.103.5 → tun_dst out: 2

Рис. 6 Таблица потоков 1

чатель и через какой криптомаршрутизатор проходил трафик. Так, например, GRE-туннель вида 172.16.102.2–172.16.103.4 означает, что трафик поступает от ЦОД 2, идет к ЦОД 3 и проходит через криптомаршрутизатор, который подключен к порту eth04 OF-коммутатора ЦОД 2. В приведенной таблице для каждого удаленного ЦОД и каждого криптомаршрутизатора, ведущего к нему, заданы свои адреса конечных точек GRE-туннеля. Входные виртуальные порты OF-коммутатора (первая колонка таблицы на рис. 6) формируются по схеме: первая цифра — номер ЦОД, вторая цифра — номер порта, в который подключен криптомаршрутизатор.

Далее опишем базовый алгоритм работы контроллера при балансировке ICMP-трафика.

1. В начальный момент времени на рабочей станции пользователь вводит команду ping server.
2. Дальше повторяются шаги 2–4 описанной ранее схемы прохождения трафика без балансировки. Как и ранее, рассмотрим только трафик, идущий в сторону ЦОД 2.
3. В начальный момент времени в таблице 1 (ЦОД 1) пакет, приходящий в порты 4.1 и 4.2, является первым пакетом нового потока и OF-коммутатор обращается к контроллеру за инструкциями.
4. Контроллер, получив пакет из порта 4.1 таблицы 1 (в сторону ЦОД 2), выбирает случайным образом (либо в соответствии с заданным алгоритмом балансировки) криптомаршрутизатор, через который должен пройти трафик. Пусть был выбран криптомаршрутизатор с номером 2. В этом случае контроллер отсылает OF-коммутатору команду установить правило: отправлять весь данный поток с помощью команды resubmit из таблицы 0 в таблицу 1, но в порт с номером 5.1.
5. Получив команду от контроллера, OF-коммутатор отправляет трафик в туннельный порт с указанием, какие параметры туннеля должны быть установлены. Однако для создания пакетного трафика недостаточно информации, а именно: нет указания, какой использовать аппаратный адрес получателя. В итоге от имени адреса 172.16.101.2 формируется ARP-запрос к 172.16.102.5 и далее с помощью правил таблицы 0 он попадет на контроллер, так как к нему будет применима только запись с самым низким приоритетом.
6. Контроллер, получив ARP-запрос, отправит на коммутатор два OF-правила: первое указывает, что необходимо подменить адрес источника и адрес назначения в ARP-запросе и отправить его в порт eth05, за которым находится граничный маршрутизатор, а второе указывает сделать обратную подмену и отправить в порт eth03. Правила приведены на рис. 7.

   Опишем правило формирования адресов для замены. Пусть к контроллеру поступает ARP-запрос со следующими параметрами:

IN_PORT	ETH_TYPE	ARP_OP	ARP_SPA	ARP_TPA	actions
eth03	ARP	REQ	172.16.101.2	172.16.102.5	set_field: 172.16.101.1 → arp_tpa set_field: 172.16.101.3 → arp_spa eth04
eth04	ARP	REP	172.16.101.1	172.16.101.3	set_field: 172.16.101.2 → arp_tpa set_field: 172.16.102.5 → arp_spa eth03

Рис. 7 Таблица потоков 0 после обработки ARP-запроса

$$\text{arp\_spa} = 172.16.10\text{N}.2, \quad \text{arp\_tpa} = 172.16.10\text{M}.\text{P},$$

где N — номер ЦОД отправителя; M — номер ЦОД получателя; P — номер порта OF-коммутатора ЦОДа отправителя, к которому подключен выбранный при балансировке криптомаршрутизатор. Тогда запрос трансформируется в следующий:

$$\text{arp\_spa} = 172.16.10\text{N}.(\text{M}+1), \quad \text{arp\_tpa} = 172.16.10\text{N}.1$$

и отправляется в порт номер eth0P.

Обратная трансформация выполняется по правилу: ARP-ответ с параметрами

$$\text{arp\_spa} = 172.16.10\text{N}.1, \quad \text{arp\_tpa} = 172.16.10\text{N}.(\text{M}+1),$$

пришедший в порт eth0P, преобразуется в ARP-ответ

$$\text{arp\_spa} = 172.16.10\text{M}.\text{P}, \quad \text{arp\_tpa} = 172.16.10\text{N}.2$$

и отправляется в порт eth03.

7. После указанной подмены в сетевой порт eth03 придет ARP-ответ с аппаратным адресом выбранного граничного маршрутизатора. Теперь Ethernet-фрейм, содержащий туннельный пакет, будет сформирован и отправлен в выходной интерфейс eth05.
8. Туннельный трафик через глобальную сеть попадает на граничный маршрутизатор ЦОД 2 и ЦОД 3 и далее на туннельный порт граничного OF-коммутатора каждого ЦОД. Это обеспечивается начальными правилами обработки ARP-запросов в OF-таблице с номером 0. Далее будем рассматривать только ЦОД 2, так как в ЦОД 3 события будут происходить аналогично за тем лишь исключением, что широковещательный ARP-запрос к server «умрет» в локальной сети ЦОД 3 в силу того, что на него никто не ответит. OpenFlow-коммутатор в соответствии с таблицей 0 отправляет распакованный GRE-трафик на обработку в таблицу с номером 2.
9. Дальнейшие шаги полностью совпадают с шагами 7–11 схемы без балансировки нагрузки.

10. В результате на OF-коммутатор ЦОД 1 придет ARP-ответ от сервера server.
11. Дальнейшие шаги почти полностью повторяют шаги 12–14 схемы без балансировки нагрузки. Отличие заключается только в том, что ARP-ответ после попадания в таблицу 1 коммутатора ЦОД 2 также проходит описанную процедуру балансировки.

Резюмируем процесс прохождения трафика через таблицы OF-коммутаторов. На рис. 8 показан процесс прохождения ARP-запроса от клиента к серверу. На рис. 9 показан процесс прохождения ARP-ответа от сервера к клиенту.

Осталось описать настройки каждого криптографического маршрутизатора. В каждом ЦОД сетевые настройки криптографических маршрутизаторов со стороны локальной сети абсолютно идентичны. Список адресов доступа на криптографических маршрутизаторах также абсолютно идентичен. Фактически в настройках указывается только один адрес доступа — локальная точка создания GRE-туннеля. Так, для криптомаршрутизаторов ЦОД 1 этим адресом будет 172.16.101.2.

6 Поддержка эластичных центров обработки данных

При объединении нескольких ЦОД, как уже говорилось ранее, требуется обеспечить миграцию хоста из одного ЦОД в другой с сохранением возможности установления с ним соединения других хостов. Описанная схема балансировки учитывает такую миграцию, что обеспечивается использованием таймера неактивности в OF-правилах.

Так, в таблицах 0 и 2 правила выставляются с указанием тайм-аута неактивности (idle_timeout). Если хост мигрирует в другой ЦОД, то трафик перестает приходить на локальный OF-коммутатор ЦОД, из которого мигрировал узел. В итоге на коммутаторе в другом ЦОД также не будет входящего трафика в сторону мигрировавшего узла, поэтому неактуальные правила будут удалены как на локальном OF-коммутаторе, так и на OF-коммутаторах других ЦОД.

После описанного выше удаления правил в таблицах при поступлении на какой-либо OF-коммутатор нового трафика, идущего в сторону мигрировавшего узла, коммутатор обратится к контроллеру, который даст указания рассылать трафик широковещательно во все туннели. В конечном счете трафик дойдет до мигрировавшего узла, и на всех OF-коммутаторах будут установлены новые правила обработки входящих пакетов согласно описанным выше алгоритмам.

7 Детектирование обрыва защищенной связи

Для детектирования обрыва соединения предлагается использовать специальный протокол канального уровня LLDP [10]. Предлагается настроить SDN-контроллер таким образом, чтобы через некоторые промежутки времени во все GRE/VXLAN-туннели с помощью правила resubmit(1,) отправлялся

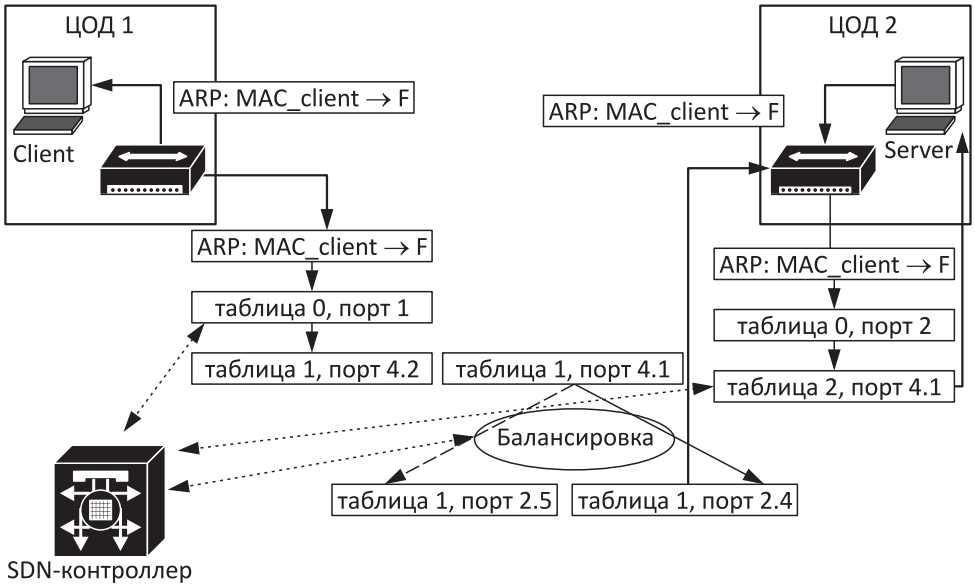


Рис. 8 Процесс прохождения ARP-запроса от клиента к серверу через таблицы OF-коммутаторов ЦОД 1 и ЦОД 2

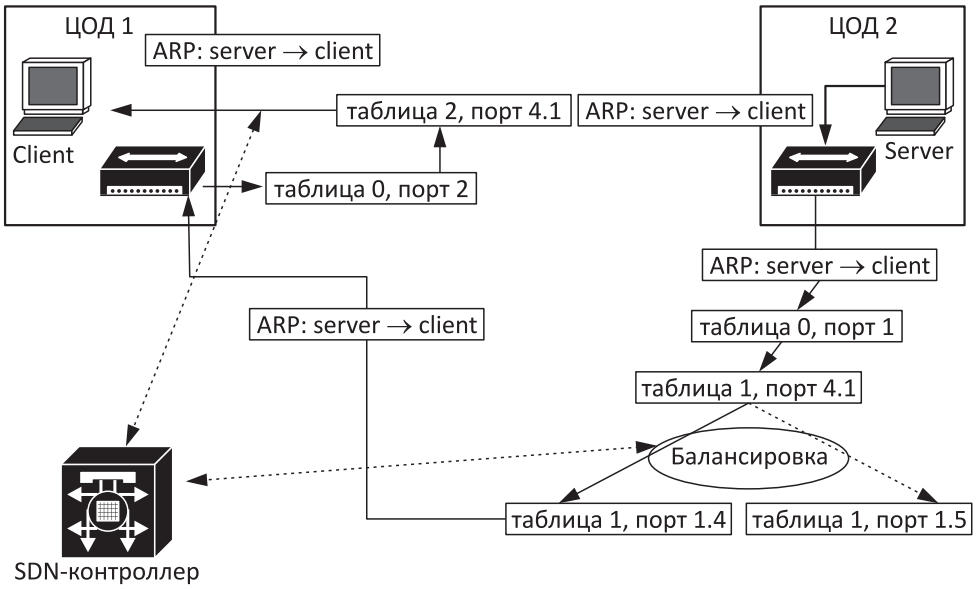


Рис. 9 Процесс прохождения ARP-ответа через таблицы ЦОД 2 и ЦОД 1

LLDP-трафик. Контроллер, получив LLDP-пакет, устанавливает на коммутаторе правило блокировать LLDP-трафик от данного узла. Это требуется, чтобы предотвратить обращение коммутатора к контроллеру в случае получения LLDP-трафика, ранее уже приходившего на коммутатор. Правило ставится с таймером неактивности, что гарантирует удаление правила в случае потери соединения. Когда правило будет удалено, коммутатор сообщит об этом контроллеру, и тот исключит связь из алгоритма балансировки. Восстановление связи приведет к появлению на коммутаторах смежных ЦОД LLDP-трафика. Об этом будет сообщено контроллеру, и связь будет вновь добавлена в список связей для балансировки нагрузки.

Немаловажно, что предложенный алгоритм балансировки позволяет отказаться от единого SDN-контроллера, который часто является узким местом в сетях такого типа. При выходе контроллера из строя будет нарушена работоспособность всей распределенной сети. Описанный подход не требует наличия единого контроллера, так как касательно балансировки каждый ЦОД может работать независимо от других ЦОД. В итоге возможно разместить в каждом ЦОД свой независимый SDN-контроллер с программой, реализующей предложенный алгоритм.

8 Заключение

В работе описан алгоритм балансировки нагрузки на криптографические маршрутизаторы при объединении ЦОД в единый эластичный ЦОД. Главными достоинствами подхода являются: существенное упрощение настройки криптографического оборудования, отказоустойчивость связи при потере криптографического узла в одном ЦОД, независимость от используемого протокола криптографической защиты сетевого уровня.

Литература

1. *Hanks S., Li T., Farinacci D., Traina P.* Generic routing encapsulation (GRE) (informational). RFC 1701, 1994. https://tools.ietf.org/html/rfc1701.
2. *Hanks S., Li T., Farinacci D., Traina P.* Generic routing encapsulation over IPv4 networks. RFC 1702, 1994. https://tools.ietf.org/html/rfc1702.
3. *Hamzeh K., Pall G., Verthein W., Taarud J., Little W., Zorn G.* Point to point tunneling protocol (informational). RFC 2637, 1999. http://www.faqs.org/rfcs/rfc2637.html.
4. *Li T., Hanks S., Meyer D., Traina P.* Generic routing encapsulation (GRE) (proposed standard). RFC 2784, 2000. http://www.faqs.org/rfcs/rfc2784.html.
5. *Dommety G.* Key and sequence number extensions to GRE (proposed standard). RFC 2890, 2000. http://www.faqs.org/rfcs/rfc2890.html.
6. IEEE 802.1ad — provider bridges, 2005. http://www.ieee802.org/1/pages/802.1ad.html.

7. *Nuopponen A., Vaarala S., Virtanen T.* IPsec clustering // Security and protection in information processing systems / Eds. Y. Deswarte, F. Cuppens, S. Jajodia, L. Wang. — IFIP advances in information and communication technology ser. — Springer US, 2004. Vol. 147. P. 367–379.
8. *Kreutz D., Ramos F. M. V., Verissimo P., Rothenberg C. E., Azodolmolky S., Uhlig S.* Software-defined networking: A comprehensive survey // P. IEEE, 2015. Vol. 103. No. 1. P. 14–76.
9. OpenFlow Switch Specification. Version 1.3.2 (Wire Protocol 0x04). April 25, 2013. ONF TS-009. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf.
10. IEEE 802.1AB — station and media access control connectivity discovery, 2005. http://www.ieee802.org/1/pages/802.1ab.html.

Поступила в редакцию 26.11.17

SDN LOAD BALANCING ON L3-VPN GATEWAYS IN DATA CENTERS INTERCONNECTION

***O. Yu. Guzev*[1] *and I. V. Chizhov*[2,3]**

[1]Research and Development Center, JSC "InfoTeCS," 1/23, b. 1 Staryy Petrovsko-Razumovskiy Pr., Moscow 127287, Russian Federation
[2]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 2nd Education Building, Faculty CMC, GSP-1, Leninskie Gory, Moscow 119991, Russian Federation
[3]Institute of informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: This work presents the method of load balancing on level 3 virtual private network (L3-VPN) gateways to scale secure communication channels between data centers. The method is based on the software-defined networking (SDN) technology. The load balancing and channel fault tolerance (Active-Active) algorithms are described in detail.

Keywords: VPN-gateway; load balancing; scaling; elastic data center; software-defined networking (SDN); OpenFlow

DOI: 10.14357/08696527180101

References

1. Hanks, S., T. Li, D. Farinacci, and P. Traina. 1994. Generic routing encapsulation (GRE) (informational). RFC 1701. Available at: https://tools.ietf.org/html/rfc1701 (assessed February 24, 2018).

2. Hanks, S., T. Li, D. Farinacci, and P. Traina. 1994. Generic routing encapsulation over IPv4 networks. RFC 1702. Available at: https://tools.ietf.org/html/rfc1702 (assessed February 24, 2018).
3. Hamzeh, K., G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. 1999. Point to point tunneling protocol (informational). RFC 2637. Available at: http://www.faqs.org/rfcs/rfc2637.html (assessed February 24, 2018).
4. Li, T., S. Hanks, D. Meyer, and P. Traina. 2000. Generic routing encapsulation (GRE) (proposed standard). RFC 2784. Available at: http://www.faqs.org/rfcs/rfc2784.html (assessed February 24, 2018).
5. Dommety, G. 2000. Key and sequence number extensions to GRE (proposed standard). RFC 2890. Available at: http://www.faqs.org/rfcs/rfc2890.html (assessed February 24, 2018).
6. IEEE 802.1ad — provider bridges. 2005. Available at: http://www.ieee802.org/1/pages/802.1ad.html (accessed February 22, 2018).
7. Nuopponen, A., S. Vaarala, and T. Virtanen. 2004. IPsec clustering. *Security and protection in information processing systems*. Eds. Y. Deswarte, F. Cuppens, S. Jajodia, and L. Wang. IFIP advances in information and communication technology ser. Springer US. 147:367–379.
8. Kreutz, D., F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. 2015. Software-defined networking: A comprehensive survey. *P. IEEE* 103(1):14–76.
9. OpenFlow Switch Specification. Version 1.3.2 (Wire Protocol 0x04). 2013. ONF TS-009, Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf (accessed February 22, 2018).
10. IEEE 802.1AB — station and media access control connectivity discovery. 2005. Available at: http://www.ieee802.org/1/pages/802.1ab.html (accessed February 22, 2018).

Received November 26, 2017

Contributors

Guzev Oleg Yu. (b. 1980) — Candidate of Science (PhD) in technology, researcher, Research and Development Center, JSC "InfoTeCS," 1/23, b. 1, Staryy Petrovsko-Razumovskiy Pr., Moscow 127287, Russian Federation; oleg.guzev@infotecs.ru

Chizhov Ivan V. (b. 1984) — Candidate of Science (PhD) in physics and mathematics, associate professor, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 2nd Education Building, Faculty CMC, GSP-1, Leninskie Gory, Moscow 119991, Russian Federation; senior scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ichizhov@cs.msu.ru

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2018. Т. 28. № 1. С. 156–176

ПРИМЕНЕНИЕ НЕЧЕТКОГО ЗАЩИЩЕННОГО ХРАНИЛИЩА ДЛЯ ИСПРАВЛЕНИЯ НЕТОЧНОСТЕЙ В АУТЕНТИФИКАЦИОННЫХ ДАННЫХ

Д. Е. Гордиенко[1], *Ю. В. Косолапов*[2], *А. С. Мышко*[3]

Аннотация: В ряде методов аутентификации возникают неточности в силу невозможности или сложности воспроизведения аутентификационных данных, абсолютно совпадающих с данными, указанными пользователем при первичной регистрации в информационной системе. К таким методам относятся, в частности, биометрические методы, а также некоторые методы на основе графических паролей. Исследуется возможность применения схемы нечеткого защищенного хранилища (fuzzy vault) для исправления таких неточностей и обеспечения высокой стойкости к подбору аутентификационных данных.

Ключевые слова: нечеткое защищенное хранилище; аутентификация; клавиатурный почерк; графический пароль

DOI: 10.14357/08696527180112

Введение

Аутентификация пользователей при проверке их прав и полномочий является неотъемлемым механизмом защиты от несанкционированного доступа. К наиболее часто используемым методам аутентификации относятся парольная аутентификация, биометрическая аутентификация, аутентификация по смарт-картам и аутентификация на основе цифровых сертификатов [1]. Последние два метода, с одной стороны, обладают высокой стойкостью, но, с другой стороны, к недостаткам этих способов стоит отнести то, что пользователю необходимо иметь некоторый носитель информации: смарт-карту или токен — для прохождения процедуры аутентификации, а проверяющей стороне необходимо иметь соответствующие специальные аппаратные интерфейсы для считывания аутентификационной информации с носителя. Частично этот недостаток снимается в системах биометрической аутентификации, так как пользователю нет необходимости для прохождения аутентификации иметь при себе носитель информации, однако проверяющая сторона должна иметь аппаратные интерфейсы для считывания биометрических данных. Таким образом, для биометрической ау-

[1]Институт математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета, dimedved@rambler.ru

[2]Институт математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета, itaim@mail.ru

[3]Институт математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета, me.metida@gmail.com

тентификации, аутентификации по смарт-картам и аутентификации на основе цифровых сертификатов непросто добиться переносимости этих методов, при которой пользователь смог бы проходить аутентификацию в информационной системе с различных типов устройств. Наиболее универсальной (с точки зрения переносимости) является парольная аутентификация, так как не задействуются носители информации и специфические аппаратные интерфейсы для считывания данных (необходима только клавиатура). В случае применения сложных паролей такая аутентификация обеспечивает высокую стойкость к подбору пароля. Однако сложные пароли трудны для запоминания (особенно при редком обращении к процедуре аутентификации) [2], а ошибка даже в одном символе сложного пароля приводит к отказу в доступе. Это часто провоцирует пользователей записывать пароли, что в свою очередь ослабляет защиту на основе символьных паролей.

В настоящей работе ставится задача исследования возможности построения методов аутентификации, которые обладали бы переносимостью, сравнимой с переносимостью метода парольной аутентификации, обеспечивали бы высокую стойкость к подбору аутентификационной информации и при этом не требовали бы на этапе аутентификации абсолютной точности аутентификационных данных, предъявляемых на этапе регистрации. Последнее требование к методам аутентификации по аналогии с [3] будем называть требованием *нечеткой аутентификации* (fuzzy authentication). В работе модифицируются и исследуются два метода аутентификации: на основе клавиатурного почерка и на основе графических паролей. Выбор этих методов аутентификации в качестве основы для модификации и исследования обусловлен тем, что для прохождения аутентификации не требуется специальных интерфейсов ввода, кроме клавиатуры. Кроме того, эти методы представляются наиболее удобными для пользователя, так как применение графических паролей существенно облегчает пользователям задачу запоминания придуманного пароля [4–7], а при биометрической аутентификации от пользователя не требуется запоминания аутентификационной информации вовсе [1]. Для обеспечения возможности нечеткой аутентификации и одновременно обеспечения высокой стойкости в настоящей работе предлагается использовать схему *нечеткого защищенного хранилища* (fuzzy vault) из [8], основанную на применении методов помехоустойчивого кодирования. Отметим, что применение помехоустойчивого кодирования для исправления ошибок, допускаемых пользователем (оператором), не нова, некоторые аспекты такого применения рассмотрены в [9]. В [10] для исправления ошибок биометрической аутентификации также используются помехоустойчивые коды, однако применяемая в [10] схема *нечеткого вручения бит* (fuzzy bit commitment) из работы [11] не устойчива к перестановке символов в аутентификационной информации, что нежелательно, например, для аутентификации по графическому паролю, где главным требованием является выбор (в произвольном порядке) правильного *множества* картинок.

Работа имеет следующую структуру. В первом разделе приводятся сведения о схеме нечеткого защищенного хранилища и строится математическая

модель применения этой схемы для аутентификации. Эта модель уточняется для биометрической аутентификации на основе клавиатурного почерка и для аутентификации на основе графических паролей во втором и третьем разделах соответственно. Там же приводится описание проведенных экспериментов и анализируются их результаты. В заключении обсуждаются полученные в работе результаты и делаются выводы о возможности применения схемы нечеткого защищенного хранилища в построенных методах аутентификации.

1 Математическая модель аутентификации на основе схемы нечеткого защищенного хранилища

Приведем необходимые сведения о нечетком защищенном хранилище в соответствии с [12]. Пусть $\mathbf{s}$ — секрет, представляющий собой k-мерный вектор со значениями из конечного поля $\mathbb{F}_q$ (q — степень простого числа), $\mathrm{md} : \mathbb{F}_q^k \to \to \mathbb{F}_q^c$ — отображение, ставящее в соответствие вектору из $\mathbb{F}_q^k$ вектор из $\mathbb{F}_q^c$, $c \in \mathbb{N}$ (в настоящей работе отображение md играет роль контрольной суммы для вектора). В качестве секрета может быть криптографический ключ, секретное сообщение, пароль и т. п. На отображение md наложим следующее ограничение: если X — случайная величина, принимающая значения из $\mathbb{F}_q^k$ равновероятно, то случайная величина $\mathrm{md}(X)$ принимает значения из $\mathbb{F}_q^c$ также равновероятно (с вероятностью $1/q^c$). Рассмотрим биективное отображение $\pi : \mathbb{F}_q^k \to \mathbb{F}_q[x]$ из $\mathbb{F}_q^k$ в кольцо полиномов $\mathbb{F}_q[x]$ степени не выше $k + c - 1$, действующее на вектор $\mathbf{s} = (s_0, \ldots, s_{k-1})$ по правилу:

$$\pi(\mathbf{s}) = s(x) = s_0 x^0 + s_1 x^1 + \cdots + s_{k-1} x^{k-1} + \cdots + s_{k+c-1} x^{k+c-1} \, (\in \mathbb{F}_q[x]) \ , \quad (1)$$

где

$$(s_k, \ldots, s_{k+c-1}) = \mathrm{md}(\mathbf{s}) \, . \quad (2)$$

Пусть $A = \{\alpha_1, \ldots, \alpha_t\}$ ($\subseteq \mathbb{F}_q$) — секретное множество мощности t ($\geqslant k + c$). Алгоритм LOCK на основе секрета $\mathbf{s}$, секретного множества A и набора из r случайных пар из $\mathbb{F}_q \times \mathbb{F}_q$ строит список пар чисел V мощности $t + r$, который представляет собой нечеткое защищенное хранилище для секрета $\mathbf{s}$.

Поясним алгоритм LOCK. Сначала по секрету $\mathbf{s}$ в соответствии с (1) строится полином $\pi(\mathbf{s})$ (метка 1 в алгоритме). Затем (шаги с метками 2–4) по множеству аргументов A и секретному полиному $\pi(\mathbf{s})$ строится список Q пар вида (*значение аргумента*, *значение полинома*). Такие пары будем называть точками, а первый и второй элементы пары будем соответственно называть абсциссой и ординатой точки. На шагах алгоритма с метками 5–9 строится список R шумовых пар точек. При этом множество абсцисс шумовых точек и множество ординат шумовых точек не должны пересекаться с соответствующими множествами ранее выбранных значений (см. шаги с метками 6 и 7). Такой способ построения

Алгоритм 1: LOCK

Исходные параметры: Секрет $\mathbf{s}$ ($\in \mathbb{F}_q^k$), множество $A = \{\alpha_i\}_{i=1}^t$, r — число добавляемых случайных пар чисел.

Результат: $V(\subset \mathbb{F}_q \times \mathbb{F}_q)$, $|V| = t + r$, V — хранилище секрета $\mathbf{s}$ (список пар чисел).

1 $s(x) = \pi(\mathbf{s})$; $i = 1$; $V = \varnothing$; $Q = \varnothing$; $R = \varnothing$;
2 **до тех пор, пока** $i \leqslant t$, **выполнять**
3 $\quad Q = Q \cup \{(\alpha_i, s(\alpha_i))\}$
4 $\quad i = i + 1$
конец цикла
5 **до тех пор, пока** $i \leqslant t + r$, **выполнять**
6 $\quad$ Выбрать случайно α_i из $\mathbb{F}_q \setminus \{\alpha_j\}_{j=1}^i$
7 $\quad$ Выбрать случайно y_i из $\mathbb{F}_q \setminus \{s(\alpha_j)\}_{j=1}^i$
8 $\quad R = R \cup \{(\alpha_i, y_i)\}$
9 $\quad i = i + 1$
конец цикла
10 $V = R \cup Q$
11 Перемешать случайным образом список V
12 **возвратить** *перемешанный список* V

не позволяет наблюдателю защищенного хранилища выделять наборы точек, из которых только одна точка принадлежит списку Q, а другие — случайные (принадлежат списку R). На шаге с меткой 10 списки Q и R объединяются в список V, а на шаге с меткой 11 полученный список точек случайным образом перемешивается (меняется порядок следования точек в списке) и возвращается на шаге с меткой 12. Перемешивание выполняется для того, чтобы нельзя было по первым t элементам списка $Q \cup R$ восстановить A и, как следствие, секрет $\mathbf{s}$.

Набор V в общем случае может быть размещен в общедоступном (незащищенном) месте. В дальнейшем множество A будем называть множеством, *закрывающим* секрет $\mathbf{s}$ в хранилище V. Отметим, что если задать на множестве A линейный порядок, то набор $(s(a))_{a \in A}$ представляет кодовое слово помехоустойчивого кода Рида–Соломона, соответствующее вектору $(s_0, s_1, \ldots, s_{k-1}, s_k, \ldots, s_{k+c-1})$ [8]. Объем (в битах) нечеткого защищенного хранилища составляет $2(t + r)\lceil \log_2(q) \rceil$ бит.

В [12] показано, что для *открытия* хранилища V (т. е. для извлечения секретного вектора $\mathbf{s}$ из V) с высокой вероятностью (порядка $1 - 1/q^c$) может использоваться такое множество $A' = \{\alpha'_i\}_{i=1}^{t'}(\subset \mathbb{F}_q)$, что

$$\left|A \setminus \left(A \cap A'\right)\right| \leqslant \lfloor t - (k + c) \rfloor\,, \;\; t' \leqslant t\,. \tag{3}$$

В этом заключается «нечеткость» защищенного хранилища V, так как «открыть» его можно также и с помощью множества A', несколько отличающегося

от множества A, с помощью которого хранилище было закрыто. Отметим, что для декодирования истинного секрета вместо известных декодеров для кода Рида–Соломона в настоящей работе по аналогии с [12] используется алгоритм интерполяции Лагранжа. Такой выбор обусловлен простотой реализации метода декодирования, несмотря на то что при этом возможно замедление процесса декодирования и ненулевая (порядка $1/q^c$) вероятность ошибочного декодирования в пределах допустимого числа ошибок. Пусть $\mathrm{LG}_{k+c}(\widetilde{W})$ — алгоритм, который на основании множества пар точек $\widetilde{W}$ ($|\widetilde{W}| = k+c$) из $\mathbb{F}_q \times \mathbb{F}_q$ с помощью интерполяционного многочлена Лагранжа строит полином степени не выше $k+c-1$. Для открытия хранилища V с помощью множества A' используется построенный в настоящей работе алгоритм UNLOCK. В алгоритме на шагах с метками 1–5

Алгоритм 2: UNLOCK

Исходные параметры: Хранилище $V = \{(x_i, y_i)\}_{i=1}^{r+t}$, $A' = \{\alpha'_i\}_{i=1}^{t'}$, $t' \leqslant t$.

Результат: $\mathbf{s}' \in \mathbb{F}_q^k$ или сообщение об ошибке открытия хранилища.

1 $W = \varnothing$; $i = 1$; found = FALSE
2 **до тех пор, пока** $i \leqslant t$, **выполнять**
3 | Найти в V такую пару (x, y), что $\alpha'_i = x$
4 | Если такая пара найдена, то $W = W \cup \{(x, y)\}$
5 | $i = i + 1$
конец цикла
6 **если** $|W| \geqslant k + c + 1$, **тогда**
7 | По множеству W построить мультимножество $\mathcal{W}$ вида (4)
8 | $i = 1$
9 | **до тех пор, пока** found = FALSE u $i \leqslant C_{|W|}^{k+c}$, **выполнять**
10 | | $s'(x) = \mathrm{LG}_{k+c}(W_i)$ // $W_i \in \mathcal{W}$
11 | | // $s'(x) = s'_0x^0 + s'_1x^1 + \cdots + s'_{k-1}x^{k-1} + s'_kx^k + \cdots + s'_{k+c}x^{k+c}$
12 | | **если** $\mathrm{md}(s'_0, \ldots, s'_{k-1}) = (s'_k, \ldots, s'_{k+c})$ **тогда**
13 | | | found = TRUE
| | **конец условия**
14 | | $i = i + 1$
| **конец цикла**
конец условия
15 **если** found = FALSE, **тогда**
16 | **возвратить** *Ошибка открытия хранилища*
конец условия
иначе
17 | **возвратить** $\mathbf{s}' := (s'_0, \ldots, s'_{k-1})$
конец условия

сначала по множеству A' строится множество W пар чисел. Именно из защищенного хранилища V выбираются те точки, абсциссы которых принадлежат A'. Далее, при условии $|W| \geqslant k + c$ (проверка выполняется на шаге с меткой 6), по множеству W на шаге с меткой 7 строится мультимножество $\mathcal{W}$ (выполнение условия $|W| \geqslant k+c$ необходимо для того, чтобы имелась возможность применить алгоритм LG_{k+c}):

$$\mathcal{W} = \{W_i\}_{i=1}^{C_{|W|}^{k+c}}, \;\; W_i \subseteq W, \;\; |W_i| = k + c. \tag{4}$$

Отметим, что, поскольку $|W_i| = k + c$, к набору точек W_i может быть применим алгоритм интерполяции LG_{k+c}. На шаге с меткой 1 вводится логическая переменная found, которая принимает значение TRUE в случае, когда найден кандидат на секрет (изначально этой переменной присвоено значение FALSE). По каждому W_i с помощью алгоритма LG_{k+c} строится полином $s'(x)$ (см. шаг с меткой 10) и соответствующий кандидат $\mathbf{s}' \in \mathbb{F}_q^k$ на истинный секрет. Истинность секрета проверяется на шаге с меткой 12 посредством вычисления контрольной суммы $\mathrm{md}(\mathbf{s}')$ и сравнения результата с последними c коэффициентами полинома $s'(x)$. Весь цикл с шагами, обозначенными метками 9–14, завершается либо в случае, когда найден кандидат на секретный полином, для которого истинным является сравнение, проверяемое на шаге 12 (в этом случае на шаге 17 возвращается кандидат на секрет), либо когда просмотрены все элементы мультимножества $\mathcal{W}$, но при этом кандидат не найден. В последнем случае условие, проверяемое на шаге с меткой 15, будет истинным, и на шаге с меткой 16 будет возвращено сообщение об ошибке открытия хранилища; эта ситуация возможна в случае, когда условие (3) не выполнено, в частности, когда множество A' существенно отличается от A.

Построенную модель закрытия и открытия нечеткого защищенного хранилища с алгоритмами LOCK и UNLOCK назовем $\mathrm{FV}(q, k, t, r, c)$-схемой. Модель регистрации и аутентификации пользователя в соответствии с построенной математической моделью изображена на рис. 1, где цифрами в кружках обозначена последовательность выполняемый действий.

Оценим защищенность $\mathrm{FV}(q, k, t, r, c)$-схемы. Пусть p_V — вероятность открыть хранилище V, закрытое с помощью алгоритма LOCK для $\mathrm{FV}(q, k, t, r, c)$-схемы, когда для открытия хранилища случайным образом выбираются $k + c$ точек из V. Другими словами, p_V — это вероятность успешного открытия нечеткого защищенного хранилища случайно выбранным набором A'. Пусть также $\mathcal{S}_V$ — множество полиномов степени не выше $k + c - 1$, которые могут быть успешно открыты по V с помощью алгоритма UNLOCK:

$$\mathcal{S}_V = \left\{ s'(x) \in \mathbb{F}_q[x] : \begin{array}{l} \deg(s'(x)) < k + c \wedge \\ \exists Q \subset V | Q = \{(x_i, s'(x_i))\}_{i=1}^{t} \wedge \\ \mathrm{md}\left(s'_0, \ldots, s'_{k-1}\right) = \left(s'_k, \ldots, s'_{k+c-1}\right) \end{array} \right\}.$$

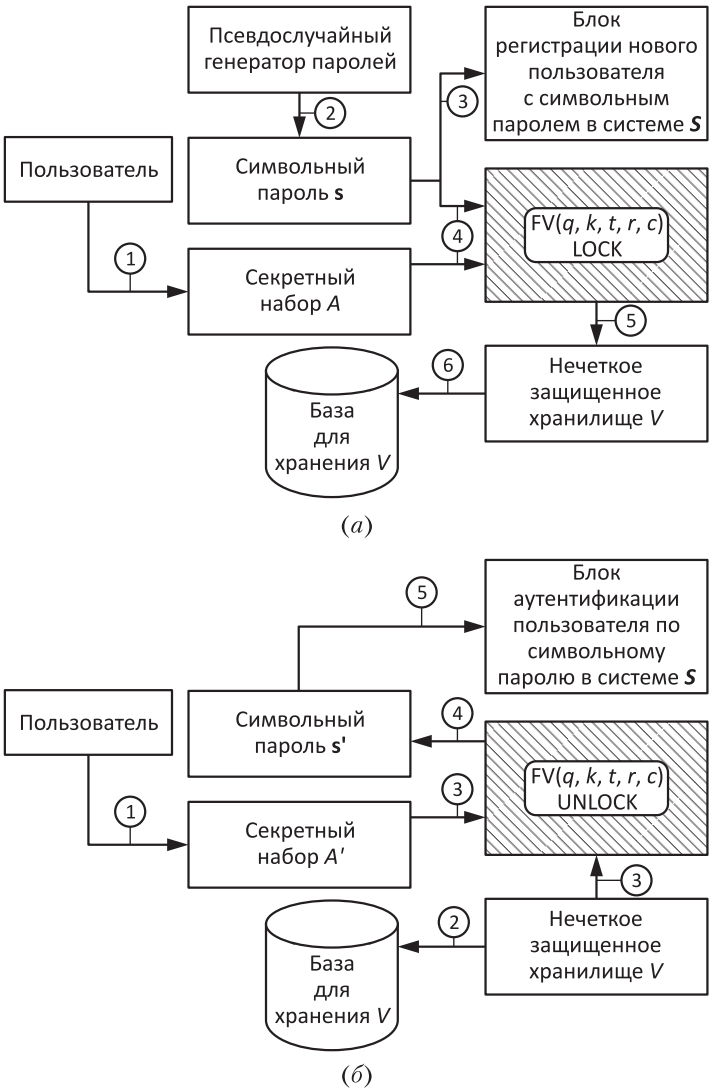


Рис. 1 Модель применения $\mathrm{FV}(q, k, t, r, c)$-схемы в системах аутентификации пользователей: (a) регистрация пользователя в системе $\boldsymbol{S}$; (b) аутентификация пользователя в системе $\boldsymbol{S}$

Отметим, что чем больше $|\mathcal{S}_V|$, тем меньше вероятность найти *истинный* полином $s(x)$, соответствующий секрету $\mathbf{s}$ (каждый из полиномов множества $\mathcal{S}_V$ может рассматриваться как полином, соответствующий некоторому секрету, при этом нет оснований отдать предпочтение какому-либо из этих полиномов).

Замечание 1. Вычислительная сложность построения множества $\mathcal{S}_V$ по хранилищу V имеет порядок $\min\{\mathcal{O}(q^{k+c}), \mathcal{O}(C_{t+r}^t)\}$, где $\mathcal{O}(q^{k+c})$ — сложность перебора всех полиномов степени ниже $k+c$, а $\mathcal{O}(C_{t+r}^t)$ — сложность перебора всех наборов по t точек в множестве из $t+r$ элементов.

Таким образом, из замечания 1 следует, что небольшая мощность множества $\mathcal{S}_V$ не обязательно означает незащищенность $\mathrm{FV}(q,k,t,r,c)$-схемы. Другими словами, схема может быть нестойкой в теоретико-информационном смысле, но при этом стойкой в вычислительном.

Утверждение 1. *Пусть V — нечеткое защищенное хранилище, соответствующее $\mathrm{FV}(q,k,t,r,c)$-схеме, $|V| = t+r$, $\mu > 0$ $(\in \mathbb{R})$. Тогда*

(1) *$\mathrm{FV}(q,k,t,r,c)$-схема позволяет исправить до $e = t-(k+c)$ ошибок в аутентификационных данных с вероятностью порядка $1 - 1/q^c$;*

(2) $p_V = C_t^{k+c}/C_{t+r}^{k+c} \leqslant (1-(k+c)/(t+r))^r$;

(3) $|\mathcal{S}_V| \geqslant (\mu/3)q^{k-t}(r/t)^t$ *с вероятностью не менее* $1-\mu$.

Доказательство. Число допустимых ошибок следует из (3), а вероятность ошибочного декодирования следует из того, что случайная величина $\mathrm{md}(X)$ имеет равномерное распределение, когда X — случайная равномерно распределенная величина со значениями из $\mathbb{F}_q^k$.

Докажем оценку для p_V. Пусть V — защищенное хранилище, в котором закрыт секрет $\mathbf{s}(\in \mathbb{F}^k)$. Так как $|V| = t+r$, а для нахождения полинома степени не выше $k+c-1$ требуется $k+c$ точек, то всего имеется C_{t+r}^{k+c} возможных наборов таких точек. При этом существует C_t^{k+c} подходящих наборов, открывающих секрет $\mathbf{s}$. Поэтому

$$p_V = \frac{C_t^{k+c}}{C_{t+r}^{k+c}} = \frac{t!(t+r-(k+c))!}{(t+r)!(t-(k+c))!} = \\ = \frac{(t-(k+c)+1)\cdots(t-(k+c)+r)}{(t+1)\cdots(t+r)} = \\ = \left(1-\frac{k+c}{t+1}\right)\cdots\left(1-\frac{k+c}{t+r}\right) \leqslant \left(1-\frac{k+c}{t+r}\right)^r.$$

В соответствии с [8, лемма 4], в множестве V с вероятностью по меньшей мере $1-\mu$ найдется не менее $(\mu/3)q^{k+-t}(r/t)^t$ наборов мощности t, соответствующих полиномам степени не выше $k+c-1$. Однако не все эти полиномы могут удовлетворять условию вида (2). Так как вероятность выполнения условия (2) для случайного вектора из $\mathbb{F}_q^{k+c}$ равна $1/q^c$, ожидаемая мощность множества $\mathcal{S}_V$ удовлетворяет неравенству $|\mathcal{S}_V| \geqslant (\mu/3)q^{k+c-t}(r/t)^t q^{-c}$. □

В разд. 2 и 3 строятся схемы аутентификации на основе клавиатурного почерка и на основе графических паролей. В обоих случаях предполагается, что пользователь регистрируется в системе $\mathbf{S}$ с помощью *символьного* пароля $\mathbf{s}$, который генерируется псевдослучайным образом (в общем случае без участия пользователя). Таким образом, пароль $\mathbf{s}$ является аутентификационным фактором. Для защиты доступа к этому паролю используется $\mathrm{FV}(q, k, t, r, c)$-схема. В случае использования клавиатурного почерка пароль $\mathbf{s}$ защищается с помощью аутентификационных данных A, полученных по клавиатурному почерку пользователя; в случае применения графических паролей в качестве набора A используется секретный набор изображений, выбранных пользователем.

2 Схема биометрической аутентификации по клавиатурному почерку

2.1 Математическая модель

Клавиатурный почерк пользователя можно представить в виде последовательности длительностей $\tau_x^{(p)}$ удержания клавиш (где «x» — символ удерживаемой клавиши) и интервалов $\tau_{x,y}^{(n)}$ между отпусканием предыдущей (с символом «x») и нажатием следующей клавиши (с символом «y»). Считается, что клавиатурный почерк может рассматриваться как уникальная особенность индивида [13, 14]. В связи с этим имеется ряд работ, в которых исследуется возможность применения клавиатурного почерка для аутентификации (см., например, [10, 13–15]. Заметим, что биометрическая аутентификация на основе клавиатурного почерка является самым дешевым из методов биометрической аутентификации. В то же время низкая точность этого метода не позволяет применять его самостоятельно [13]. Поэтому биометрическая аутентификация на основе клавиатурного почерка обычно представляет собой двухфакторную аутентификацию: с одной стороны, пользователь вводит секретный пароль (возможно, несложный и легко запоминающийся), а с другой стороны, он дополнительно аутентифицируется по особенностям клавиатурного набора [15]. В этом случае длина секретной фразы должна быть не менее 21–42 символов, а в качестве особенностей клавиатурного почерка используются как интервалы $\tau_{x,y}^{(n)}$, так и длительности $\tau_x^{(p)}$. В настоящей работе исследуется упрощенный случай, когда вводимая при регистрации/аутентификации пользователя фраза является *короткой* — 11 символов, а в качестве особенности клавиатурного почерка пользователя используется только последовательность интервалов $\tau_{x,y}^{(n)}$ для символов из вводимой фразы. Для исправления возможных ошибок при аутентификации используется схема нечеткого защищенного хранилища $\mathrm{FV}(q, k, t, r, c)$. В предлагаемой схеме, как и в других схемах на основе клавиатурного почерка, предполагаются три этапа: обучения, регистрации и аутентификации. В качестве отображения md применяется алгоритм CRC32.

На *этапе обучения* для каждого пользователя собирается информация о нажатых клавишах и интервалах между последовательными нажатиями. После сбора статистики информация о пользователе представляется в виде множества пар: $\{([x,y],\overline{\tau}^n_{x,y}) \in (K\times K)\times\mathbb{N}\}$, где K — множество символов всех клавиш, пара $[x,y](\in K\times K)$ означает нажатие клавиши y после x, а $\overline{\tau}^n_{x,y}$ — среднее время в миллисекундах между их нажатиями, $\overline{\tau}^n_{x,y} \in [1, 10\ 000]$. Также для пользователя сохраняется среднеквадратичное отклонение ϵ для нажатий. Стандартная клавиатура имеет 101 клавишу, поэтому для представления пары $([x,y],\overline{\tau}^n_{x,y})$ выбрано конечное поле Галуа $\mathbb{F}_{2^{28}}$ ($|\mathbb{F}_{2^{28}}| = 2^{28} \geqslant 101^2\cdot 10\ 000$). Зафиксируем биективное отображение, обозначим $\varphi : \mathbb{F}_{2^{28}} \to \{0,1\}^{28}$ и каждому элементу f поля $\mathbb{F}_{2^{28}}$ поставим в соответствие вектор $\mathbf{f} = (f_1,\ldots,f_{28}) = \varphi(f)$. Координаты с номерами от 1 до 14 вектора $\mathbf{f}$ используются для представления пары $[x,y]$, а координаты с номерами от 15 до 28 — для представления длительности $\overline{\tau}^n_{x,y}$ (в миллисекундах):

$$\varphi(f) = \mathbf{f} = (\underbrace{f_1,\ldots,f_7}_{x},\underbrace{f_8,\ldots,f_{14}}_{y},\underbrace{f_{15},\ldots,f_{28}}_{\overline{\tau}^n_{x,y}}) \in \{0,1\}^{28}.$$

Вектор множества $\{0,1\}^{28}$, сопоставляемый тройке $([x,y],\overline{\tau}^n_{x,y})$, обозначим $\mathbf{f}_{(x,y)}$, а соответствующий элемент поля Галуа $\mathbb{F}_{2^{28}}$ обозначим символом $f_{(x,y)}$, $\varphi(f_{(x,y)}) = \mathbf{f}_{(x,y)}$.

На этапе обучения статистика накапливается путем набора пользователем текста длиной не менее T символов. После накопления статистики пользователю *на этапе регистрации* предлагается ввести парольную фразу $\mathbf{x} = (x_1,\ldots,x_l)$ (возможно, легко запоминающуюся), по которой строится *множество* $A = \{f_{(x_{i+1},x_i)}\}_{i=1}^{\widetilde{l}}$, $\widetilde{l} \leqslant l-1$. Заметим, что $\widetilde{l}$ может быть меньше $l-1$, так как для фразы $\mathbf{x}$ могут найтись такие $i \neq j$, что $(x_{i+1},x_i) = (x_{j+1},x_j)$. В этом случае в множестве следует оставить только один из элементов: $f_{(x_{i+1},x_i)}$ или $f_{(x_{j+1},x_j)}$. Далее для пользователя псевдослучайным образом генерируется пароль $\mathbf{s}$, который с помощью множества A и алгоритма LOCK закрывается в нечетком защищенном хранилище V (см. рис. 1, *а*).

На этапе аутентификации пользователю предлагается ввести ту же фразу $\mathbf{x}$, и после ее ввода строится множество $A' = \{f'_{(x_{i+1},x_i)}\}_{i=1}^{\widetilde{l}}$. По хранилищу V и множеству A' строится набор пар W следующим образом. Для каждого $f'_{(x_{i+1},x_i)}$ из A' проверяется, имеется ли в V пара (a,b) такая, что первые (слева) 14 координат вектора $\mathbf{f}'_{(x_{i+1},x_i)} = \varphi(f'_{(x_{i+1},x_i)})$ совпадают в первыми 14 координатами вектора $\varphi(a)$, а модуль разности десятичного представления последних 14 координат вектора $\mathbf{f}'_{(x_{i+1},x_i)}$ и десятичного представления последних 14 координат вектора $\varphi(a)$ не более ϵ. Если обе проверки проходят, то пара (a,b) добавляется в множество W. Открытие секрета $\mathbf{s}'$ выполняется с помощью алгоритма

UNLOCK и множества W (см. рис. 1, *б*). С помощью $\mathbf{s}'$ выполняется попытка аутентификации пользователя в системе $\mathbf{S}$.

Отметим, что с точки зрения защищенности для предлагаемой схемы аутентификации в алгоритме LOCK генерировать случайным образом абсциссы шумовых точек нежелательно, так как, исследовав защищенное хранилище V, недоброжелатель сможет отбросить случайные точки (не соответствующие никаким парам клавиш), проанализировав первые 14 координат векторного представления абсциссы каждой точки из V. В алгоритме LOCK набор шумовых точек предлагается формировать следующим образом: первые 14 координат векторного представления абсциссы генерируемой шумовой точки должны соответствовать какой-нибудь паре клавиш, а десятичное представление последних 14 координат векторного представления абсциссы шумовой точки должно быть в пределах от 1 до 10 000.

2.2 Результаты экспериментов

Для проведения экспериментов выбрана схема нечеткого защищенного хранилища $\mathrm{FV}(2^{28}, 5, 11, 85\ 000, 1)$. Длина защищаемого секрета составляет 5 символов поля $\mathbb{F}_{2^{28}}$, что соответствует секрету длиной не менее 17 символов над полем мощности 256. В соответствии с утверждением 1, эта схема позволяет исправлять до $11 - (5 + 1) = 5$ ошибок при длине пароля (множества A) в 11 символов. Такое число исправляемых ошибок выбрано экспериментально для обеспечения высокой вероятности успешной аутентификации легального пользователя под своим именем. При выбранных параметрах $-\log_2(p_V) \approx 79$ и $\log_2 |\mathcal{S}_V| \geqslant 77$ ($\mu = 2^{-5}$). Размер хранилища составляет 581 КБ. Отметим, что выбранные параметры обеспечивают стойкость к подбору, сравнимую со стойкостью обычной парольной аутентификации при длине пароля 11 символов (когда используются строчные и заглавные буквы латинского алфавита, цифры и специальные символы). Если такой стойкости достаточно, то защищенное хранилище V может быть размещено на общедоступном хранилище. Усилить стойкость можно путем, например, увеличения параметра r при остальных фиксированных значениях.

Для исследования были выбраны 10 пользователей $U_1, \ldots, U_{10}$, каждому из которых на этапе обучения было предложено набрать один и тот же текст длиной $T = 10\,836$. Информация о нажатиях клавиш фиксировалась с помощью программного обеспечения Basic Key Logger [16]. После этого пользователи регистрировались в модельной информационной системе с одним и тем же паролем длины 11 символов. Один пароль для всех пользователей использован для того, чтобы исследовать вероятность аутентификации от имени другого пользователя. Каждый пользователь выполнял по 10 попыток аутентификации от своего имени и 10 попыток от имени каждого из других 9 пользователей (всего каждый пользователь проходил 100 попыток аутентификации). Первоначально для всех пользователей было выбрано общее значение ϵ так, чтобы все пользователи могли с высокой вероятностью аутентифицироваться от своего имени. В результате

Таблица 1 Результаты аутентификации по клавиатурному почерку ($\epsilon_i = 1100$)

Пользователь	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}
U_1	0,9	0,3	0,8	0,7	0,9	0,7	0,8	0,5	0,8	0,7
U_2	0,4	0,9	0,2	0,7	0,7	0,5	0,6	0,6	0,5	0,6
U_3	0,6	0,2	0,9	0,2	0,3	0,3	0,2	0,3	0,2	0,2
U_4	0,4	0,5	0,0	1,0	0,6	0,2	0,9	0,7	0,6	0,2
U_5	0,8	0,7	0,4	0,8	0,8	0,5	0,7	0,9	0,6	0,7
U_6	0,4	0,5	0,3	0,1	0,5	1,0	0,0	0,2	0,4	0,6
U_7	0,3	0,7	0,0	1,0	0,8	0,1	1,0	1,0	0,8	0,3
U_8	0,4	0,5	0,2	0,8	0,9	0,2	0,9	1,0	0,7	0,8
U_9	0,6	0,6	0,0	0,7	0,7	0,2	0,7	0,7	0,9	0,8
U_{10}	0,7	0,4	0,1	0,2	0,7	0,6	0,4	0,8	0,7	1,0

собранной на этапе обучения статистики было выбрано значение $\epsilon = 1100$. Результаты экспериментов обобщены в табл. 1. Пользователи, указанные в первом столбце табл. 1, пытались пройти аутентификацию от имени пользователей, указанных в первой строке этой таблицы; в ячейках указана частота успеха прохождения аутентификации.

Например, из табл. 1 видно, что в большинстве случаев каждый пользователь со значительной вероятностью может пройти успешно аутентификацию от имени другого пользователя, что подтверждает известную слабость аутентификации по клавиатурному почерку. Уменьшить вероятность успешной аутентификации от имени другого пользователя можно путем увеличения числа биометрических характеристик, по которым собирается статистика пользователя. Например, кроме интервалов между нажатиями учитывать также и длительности нажатий клавиш, а также использовать для каждого пользователя U_i свое значение ϵ_i, так как эта характеристика также может рассматриваться как особенность клавиатурного почерка. В частности, в рамках проведенных экспериментов установлено, что величина ϵ_i может варьироваться для разных пользователей (табл. 2). Для пользователя U_7 проведен эксперимент, в котором учитывается значение ϵ_i. Результаты

Таблица 2 Значения ϵ_i для пользователей

Пользователь	ϵ_i, мс
U_1	1000
U_2	800
U_3	1200
U_4	850
U_5	1000
U_6	1200
U_7	1100
U_8	950
U_9	800
U_{10}	1200

Таблица 3 Результаты аутентификации по клавиатурному почерку для пользователя U_7 (ϵ_i зависит от пользователя)

Пользователь	U_7
U_1	0,3
U_2	0,8
U_3	0,1
U_4	0,3
U_5	0,7
U_6	0,2
U_7	0,9
U_8	0,2
U_9	0,7
U_{10}	0,3

попыток успешной аутентификации этого пользователя приведены в табл. 3. Сравнивая значения из табл. 3 со значениями из табл. 1 (строка, соответствующая пользователю U_7), можно сделать вывод: использование для каждого пользователя соответствующей величины ϵ_i позволяет снизить вероятность успешной аутентификации от имени другого пользователя. Например, если при $\epsilon_i = 1100$ пользователь U_7 мог успешно с вероятностью 1 аутентифицироваться от имени пользователей U_4 и U_8, то использование для каждого пользователя своего значения ϵ_i позволило снизить эту вероятность до 0,3 и 0,2 соответственно.

3 Схема аутентификации на основе графических паролей

3.1 Математическая модель

Аутентификация на основе графических паролей, как показывают исследования, позволяет упростить запоминание пароля [5, 7]. К различным способам аутентификации по графическому паролю, в частности, относятся:

- соединение точек решетки в определенной последовательности;
- выбор подмножества изображений из заданного набора изображений;
- выбор множества элементов на фиксированном изображении.

Одной из уязвимостей таких систем является слабость против атаки подсматривания (shoulder surfing attack), когда атакующий имеет возможность наблюдать вводимый графический пароль [7]. Кроме того, пользователь обязан в точности воспроизвести графический пароль для успешной аутентификации. В настоящей работе предлагается схема аутентификации по графическому паролю, которая, с одной стороны, существенно затрудняет атаку подсматривания, а с другой стороны, позволяет пользователю допускать небольшое число ошибок при аутентификации. Опишем предлагаемый способ аутентификации.

В основе способа лежит использование $\mathrm{FV}(89, 6, 9, 79, 2)$-схемы. При этом размер нечеткого защищенного хранилища составляет 144 байта. В качестве отображения md используется алгоритм CRC16 (так как поле $\mathbb{F}_{89}$, то в качестве контрольной суммы берутся 14 бит значения, вычисленного с помощью CRC16), таким образом длина секрета увеличивается на 2 символа поля $\mathbb{F}_{89}$ и число исправляемых (допускаемых пользователем) ошибок равно $t-(k+c) = 9-(6+2) = 1$. В системе $\mathbf{S}$ фиксируется набор $\mathcal{P} = \{p_1, \ldots, p_{88}\}$ изображений (изображений именно 88, а не 89, чтобы было удобно разместить на экране все изображения в прямоугольной таблице размера 8×11). В качестве такого набора $\mathcal{P}$ в работе выбрано множество идеограмм, используемых в электронных сообщениях и вебстраницах. Каждому изображению ставится однозначно в соответствие элемент поля $\mathbb{F}_{89}$; пусть $\xi : \mathcal{P} \to \mathbb{F}_{89}$ — такое соответствие. Набор $\mathcal{P}$ и соответствие ξ не являются секретными. Также пусть $\mathcal{U} = \{u_1, \ldots, u_{88}\}$ — набор символов (групп символов), которые могут быть набраны на клавиатуре, S_{88} — симметрическая группа подстановок на множестве $\{1, \ldots, 88\}$.

На этапе регистрации случайным образом генерируется подстановка $\sigma \in S_{88}$ и по каждому изображению $p_i \in \mathcal{P}$ формируется новое изображение p'_i, состоящее из изображения p_i и символа $u_{\sigma(i)}$; для удобства изображение p'_i обозначим парой $(p_i, u_{\sigma(i)})$. На рис. 2 показан пример сформированного изображения p'_i в случае, когда p_i — смайлик, а $u_{\sigma(i)}$ — символ «%». Набор $\mathcal{P}'$, сформированный таким образом, отображается на экране. Пользователь последовательно выбирает подмножество $P \subset \mathcal{P}$ мощности 9, $P = \{p_{i_1}, \ldots, p_{i_9}\}$, вводя на клавиатуре соответствующие выбранным изображениям символы $u_{\sigma(i_1)}, \ldots, u_{\sigma(i_9)}$ (набор P здесь играет роль множества A, использованного на рис. 1, *а*, с помощью которого закрывается секрет $\mathbf{s}$). Порядок набора символов, соответствующих выбранным изображениям, не имеет значения. По набору P формируется набор $A = \{\xi(p_{i_1}), \ldots, \xi(p_{i_9})\}$, с помощью которого и с помощью алгоритма LOCK случайно сгенерированный пароль $\mathbf{s} \in \mathbb{F}_{89}^6$ закрывается в защищенном хранилище V (см. рис. 1, *а*). Отметим, что вводимые пользователем символы не отображаются на экране. Таким образом, для проведения атаки подсматривания наблюдателю необходимо одновременно следить, какие клавиши нажимает пользователь, и сопоставлять их с картинками p'_i, отображенными на экране, что затрудняет проведение такой атаки.

Рис. 2 Пример формируемого изображения

На этапе аутентификации генерируется случайная перестановка $\widetilde{\sigma}$ и отображается набор $\widetilde{\mathcal{P}}' = \{(p_1, u_{\widetilde{\sigma}(1)}), \ldots, (p_{88}, u_{\widetilde{\sigma}(88)})\}$. Заметим, что перестановка $\widetilde{\sigma}$ выбирается независимо от перестановки σ, использованной на этапе регистрации. Пользователь формирует множество A', набирая на клавиатуре символы, соответствующие изображениям, выбранным им на этапе регистрации, и применяя к отмеченным изображениям соответствие ξ. При этом порядок набора не имеет значения, т. е. этот порядок может не совпадать в точности с порядком набора на этапе регистрации. Заметим, способность не учитывать неточности в порядке набора приводит к некоторому снижению стойкости по сравнению с системами, где учитывается порядок на вводимых аутентификационных данных. Отметим также, что множество набираемых на этапе аутентификации символов может отличаться от символов, набираемых на этапе регистрации, так как подстановки σ и $\widetilde{\sigma}$ генерируются независимо. С помощью набора A', хранилища V и алгоритма UNLOCK выполняется попытка открытия V для извлечения секрета $\mathbf{s}'$ (см. рис. 1, *б*).

В соответствии с утверждением 1, при выбранных параметрах пользователь может допустить одну ошибку, $-\log_2(p_V) \approx 32{,}7$ и $\log_2 |\mathcal{S}_V| = 2{,}19$ (при $\mu = 2^{-5}$). Значение $\log_2 |\mathcal{S}_V|$ мало, т. е. мощность множества кандидатов на информационное сообщение также маленькая. Однако вычислительная сложность построения множества $\mathcal{S}_V$, в силу замечания 1, достаточно большая — $\mathcal{O}(2^{39})$.

Усилить стойкость (увеличить значение $\mathcal{S}_V$) можно за счет увеличения мощности поля $\mathbb{F}_q$. Например, при увеличении поля до $\mathbb{F}_{1024}$ получим: $-\log_2(p_V) \approx 61$ и $\log_2|\mathcal{S}_V| = 24{,}77$. Однако это может затруднить отображение всего набора картинок (количество отображаемых различных картинок практически равно мощности поля). В частности, необходимо либо уменьшать размеры картинок (для размещения их всех на одном экране), либо разбивать множество картинок на последовательно отображаемые страницы с изображениями. Например, при $q = 1024$ возможно отображение 9 страниц, на каждой из которых может быть размещено до $11 \times 11 = 121$ картинок. Во всех этих случаях представляется, что удобство использования такого метода аутентификации будет нарушено. Таким образом, неограниченно увеличивать мощность поля не представляется целесообразным.

Заметим также, что при увеличении длины истинного пароля $\mathbf{s}$ с 6 до 8 символов с сохранением одной допустимой ошибки потребуется увеличить мощность множества изображений t с 9 до 11 изображений, что будет соответствовать схеме нечеткого хранилища $\mathrm{FV}(89, 8, 11, 79, 1)$. Увеличение числа допустимых ошибок также потребует увеличения t на соответствующее число изображений.

3.2 Результаты экспериментов

В работе проведены эксперименты с целью выявления особенностей аутентификации на основе графических паролей. Для этого такой способ аутентификации сравнивался с классической аутентификацией на основе символьного пароля. В случае аутентификации по символьному паролю пользователю генерировался псевдослучайным образом пароль $\mathbf{p}$ длиной 6 символов, состоящий из арабских цифр, строчных и заглавных букв латинского алфавита (всего 62 символа). Во втором случае для пользователя псевдослучайно генерировался пароль $\mathbf{s}$ длиной 6 символов над полем $\mathbb{F}_{89}$, который закрывался в защищенном хранилище с помощью 9 выбранных пользователем изображений (изображения выбираются из 88 изображений, размещенных в прямоугольной таблице размера 8×11 ячеек). Отметим, что энтропия символьного пароля $\mathbf{p}$ равна $6\log_2(62) \approx 35{,}72$, а энтропия графического пароля A равна $-\log_2(p_V) \approx 32{,}7$. Другими словами, параметры эксперимента выбраны так, чтобы оба пароля обеспечивали приблизительно одинаковую стойкость к подбору.

Целью эксперимента является определение того, может ли графический пароль быть более удобной альтернативой символьному паролю (когда оба пароля обеспечивают одинаковую стойкость). После регистрации пользователь проходит аутентификацию двумя способами: с помощью графического и символьного пароля. Для успешной аутентификации графическим паролем необходимо правильно указать любые 8 изображений из выбранных (в любом порядке). Символьный пароль необходимо воспроизвести без ошибок. В качестве результата фиксируется число попыток, которое потребовалось пользователю для аутентификации.

Эксперимент регулярно повторялся с целью выявления статистически значимого различия между выбранными способами аутентификации. Сравнение средних уровней в группах проводилось с помощью критерия Вилкоксона для связанных выборок, а именно: проверялась нулевая гипотеза об отсутствии различий в методах аутентификации; альтернативная (рабочая) гипотеза — различия существуют. Различия признавались статистически значимыми на уровне $p < 0{,}05$. Было проведено 40 наблюдений, включающих 10 пользователей. Сравнение средних уровней числа попыток, потребовавшихся пользователям для аутентификации, позволило выявить статистически значимые различия между выбранными способами (см. табл. 4, где представлены средние значения числа попыток аутентификации $\pm$ $\pm$ среднеквадратичное отклонение). Значение p в таблице соответствует вероятности совершения статистической ошибки первого рода: ошибочного отклонения нулевой гипотезы. Расчеты производились с помощью сервиса Statzilla [17]. Полученное значение $p = 0{,}0008$ значительно ниже выбранного уровня значимости и позволяет отклонить гипотезу об отсутствии различий в группах в пользу рабочей гипотезы. Среднее число попыток символьной аутентификации оказалось выше. На рис. 3 приведена диаграмма размаха для числа попыток аутентификации обоими способами. Как видно из диаграммы, максимальное число попыток символьной аутентификации составляет 6, графической — 3. Меди-

Таблица 4 Сравнение средних уровней показателя числа попыток для символьного и графического способов аутентификации

Пароль	Число попыток
Символьный	$1{,}85 \pm 1{,}42$
Графический	$1{,}2 \pm 0{,}52$
	$p = 0{,}0008$

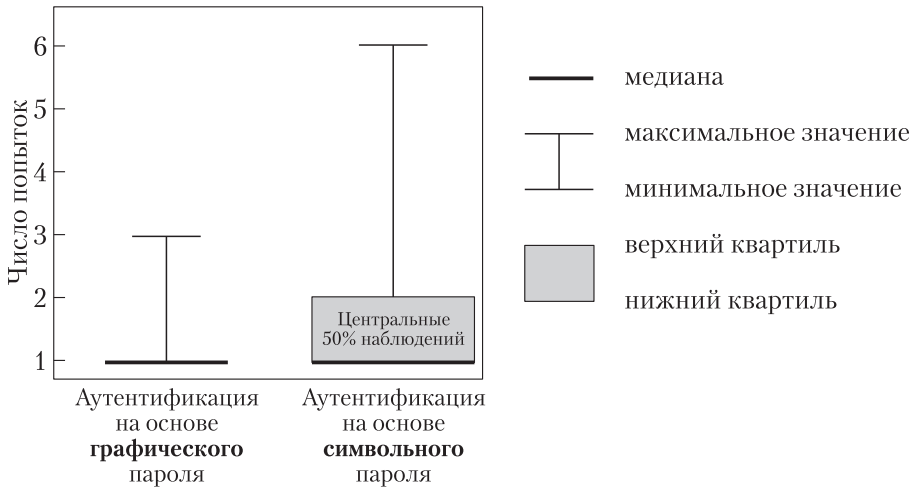


Рис. 3 Диаграмма размаха для числа попыток авторизации

анное значение числа попыток аутентификации для обоих способов совпадает и составляет 1 попытку. При этом для символьного способа аутентификации значения, попавшие в интерквартильный размах (т. е. центральные 50% наблюдений выборки), составляют от 1 до 2 попыток, что выше соответствующих значений для графического способа, совпадающих с медианой и составляющих 1 попытку.

На основе полученного результата можно сделать вывод, что графический способ аутентификации вызывает меньшую нагрузку на память пользователя. Стоит отметить, однако, что попытка графической аутентификации занимает значительно больше времени (порядка 1 мин), чем попытка аутентификации символьным паролем (порядка 10–20 с), что не было отражено в дизайне эксперимента.

4 Заключение

В предлагаемой модели на основе нечеткого защищенного хранилища вместо применяемого алгоритма интерполяции может применяться какой-либо эффективный алгоритм декодирования кодов Рида–Соломона. Это, в частности, позволит хранить секреты большей длины при тех же параметрах q, t и r. Например, если $c = 1$, то максимальная длина секрета при использовании отображения md (в процессе декодирования) может быть $t - 3$, когда допускается одна ошибка. Если же используется декодер кода Рида–Соломона, то максимальная длина секрета может быть на один символ больше — $(t - 2)$ — и при этом также может быть исправлена одна ошибка. При использовании известных полиномиальных по сложности декодеров может уменьшиться время аутентификации, так как перебор по подмножествам набора (4) имеет неполиномиальную от k сложность.

Несмотря на то что построенная схема аутентификации по клавиатурному почерку не может применяться самостоятельно, этот метод может найти применение в качестве дополнительного средства аутентификации, которое может применяться, например при удаленной аутентификации пользователя для снижения риска несанкционированного доступа [18]. Аутентификация по клавиатурному почерку может выполняться периодически в течение сеанса работы пользователя для проверки того, что в системе работает авторизованный пользователь. При этом неудачная аутентификация не обязательно должна приводить к блокированию сеанса пользователя, однако это может быть сигналом для администратора. Например, удаленное выполнение самостоятельной работы студентами может дополнительно контролироваться клавиатурным почерком, статистика для которого собиралась в течение, например, семестра. Неудачная попытка прохождения аутентификации по клавиатурному почерку не может служить препятствием к выполнению самостоятельной работы студентом, однако для преподавателя такая попытка может быть причиной дополнительного опроса студента, например при встрече в аудитории. Стоит отметить, что предложенный в качестве дополнительного средства аутентификации способ обладает рядом преимуществ: он прост в реализации, скрыт от пользователя и не требует дополнительных затрат на

оборудование. Представляется, что для снижения вероятности ложной аутентификации возможно использование двух защищенных хранилищ — по каждому на одну из характеристик клавиатурного почерка (для интервалов между нажатиями и для длительностей удержания клавиш). Паролем для системы **S** в этом случае может служить посимвольная сумма секретов, закрытых в каждом из этих хранилищ.

Метод аутентификации на основе графического пароля, наоборот, может использоваться как самостоятельное средство аутентификации, например для аутентификации на мобильных устройствах или в приложениях, где от пользователя нечасто требуется проверка подлинности (в силу длительности процедуры аутентификации). За счет генерации случайной подстановки $\widetilde{\sigma}$ на этапе аутентификации клавиатурный ввод (набираемые символы, соответствующие выбираемым картинкам) пользователя будет каждый раз новый, что затрудняет подсматривание пароля. Отображаемые изображения также могут выводиться в произвольном (случайном) порядке. Даже если недоброжелатель имеет возможность вести видеорегистрацию клавиатурного набора пользователя при аутентификации, то в случае предлагаемой схемы проведение этой атаки затрудняется тем, что каждый раз ввод пользователя будет разным и наблюдателю необходимо, чтобы при видеорегистрации фиксировался как клавиатурный набор, так и отображаемый на экране набор изображений $\widetilde{\mathcal{P}}'$.

Построенная математическая модель аутентификации с использованием нечеткого защищенного хранилища может применяться для реализации как самодостаточных механизмов аутентификации, так и дополнительных методов аутентификации, в которых возможны неточности в аутентификационных данных. Защищенность таких методов может быть оценена с помощью утверждения 1 и замечания 1. При этом ограничения в применении построенной модели обусловлены в большей степени особенностями выбираемых типов аутентификации, а не особенностями нечеткого защищенного хранилища.

Литература

1. *Lal N. A., Prasad S., Farik M.* A review of authentication methods // Int. J. Sci. Technology Res., 2016. Vol. 5. No. 11. P. 246–249.
2. *Syed Idrus S. Z., Cherrier E. , Rosenberger C., Schwartzmann J.-J.* A review of authentication methods // Aust. J. Basic Appl. Sci., 2013. Vol. 7. No. 5. P. 95–107.
3. *Ur-Rehman O., Zivic N.* Fuzzy authentication algorithm with applications to error localization and correction of images // WSEAS Trans. Syst., 2013. Vol. 12. No. 7. P. 371–383.
4. *Olson G. M., Olson J.,S.* Human–computer interaction: Psychological aspects of the human use of computing // Annu. Rev. Psychol., 2003. Vol. 54. P. 491–516.
5. *Angeli A. D., Coventry L., Johnsona G., Renaud K.* Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems // Int. J. Hum.–Comput. St., 2006. Vol. 63. P. 128–152. doi: 10.1016/j.ijhcs.2005.04.020.

6. *Gao H.* A new graphical password scheme resistant to shoulder-surfing // Conference (International) on Cyberworlds. — Singapore: IEEE, 2010. P. 194–199. doi: 10.1109/CW.2010.34.
7. *Rittenhouse R. G., Chaudry J. A., Lee M.* Security in graphical authentication // Int. J. Security Appl., 2013. Vol. 7. No. 3. P. 347–356.
8. *Juels A., Sudan M.* A fuzzy vault scheme // Design. Code. Cryptogr., 2006. Vol. 38. No. 2. P. 237–257. doi: 10.1007/s10623-005-6343-z.
9. 59 *Бояринов И. М., Давыдов А. А., Мамедли Э. М., Смеркис Ю. Б.* Использование помехоустойчивого кодирования для защиты информации от ошибок оператора // Автоматика и телемеханика, 1983. № 2. С. 5–49.
10. *Hoang T., Choi D.* Secure and privacy enhanced gait authentication on smart phone // Sci. World J., 2014. Vol. 2014. Article ID 438254. 8 p. doi: 10.1155/2014/438254. https://www.hindawi.com/journals/tswj/2014/438254/.
11. *Juels A., Wattenberg M.* A fuzzy commitment scheme // 6th ACM Conference on Computer and Communications Security Proceedings. — New York, NY, USA: ACM, 1999. P. 28–36. doi: 10.1145/319709.319714.
12. *Uludag U., Pankanti S., Jain A. K.* Fuzzy vault for fingerprints // Audio- and video-based biometric person authentication / Eds. T. Kanade, A. K. Jain, N. K. Ratha. — Lecture notes in computer science ser. — Springer, 2005. Vol. 3546. P. 310–319. doi: 10.1007/11527923_32.
13. *Брюхомицкий Ю. А.* Статистические методы распознавания клавиатурного почерка // Известия Южного федерального университета, 2009. № 11(100). С. 139–147.
14. *Khitsenko V. E., Krutohvostov D. S.* Increasing the reliability of authentication keyboard handwriting // 12th Conference (International) on Actual Problems of Electronics Instrument Engineering Proceedings. — IEEE, 2014. P. 262–265. doi: 10.1109/APEIE.2014.7040894.
15. *Akhmetov B. S., Ivanov A. I., Kartbaev T. S., Malygin A. U., Mukapil K.* Biometric dynamic personality authentication in open information space // Int. J. Computer Technology Appl., 2013. Vol. 4. No. 5. P. 846–855.
16. Basic Key Logger. https://sites.google.com/site/basiclabbook/keyloggerbasiclabbook/.
17. Statzilla. Статистика онлайн. https://online.statzilla.ru.
18. *Grassi P. A., Garcia M. E., Fenton J. L.* Digital identity guidelines // NIST Special Publication 800-63-3. https://pages.nist.gov/800-63-3/sp800-63-3.html.

Поступила в редакцию 26.05.17

USING THE FUZZY VAULT TO CORRECT INACCURACIES IN AUTHENTICATION DATA

D. E. Gordienko, Yu. V. Kosolapov, and A. S. Mishko

Institute of Mathematics, Mechanics, and Computer Science named after I. I. Vorovich, Southern Federal University, 105/42 Bol'shaya Sadovaya Str., Rostov-on-Don 344006, Russian Federation

Abstract: In a number of authentication methods, there are inaccuracies due to the impossibility or complexity of reproducing the authentication data exactly coinciding with the data specified by the user during initial registration in the information system. Such methods include, in particular, biometric methods as well as some methods based on graphic passwords. The paper explores the possibility of using the fuzzy vault scheme to correct such inaccuracies and provide high resistance to the selection of authentication data in a keyboard authentication system and in a graphical password authentication system.

Keywords: fuzzy vault; authentication; keyboard handwriting; graphic password

DOI: 10.14357/08696527180112

References

1. Lal, N. A., S. Prasad, and M. Farik. 2016. A review of authentication methods. *Int. J. Sci. Technology Res.* 5(11):246–249.
2. Syed Idrus, S. Z., E. Cherrier, C. Rosenberger, and J.-J. Schwartzmann. 2013. A review of authentication methods. *Aust. J. Basic Appl. Sci.* 7(5):95–107.
3. Ur-Rehman, O., and N. Zivic. 2013. Fuzzy authentication algorithm with applications to error localization and correction of images. *WSEAS Trans. Syst.* 12(7):371–383.
4. Olson, G. M., and J. S. Olson. 2003. Human-computer interaction: Psychological aspects of the human use of computing. *Annu. Rev. Psychol.* 54:491–516.
5. Angeli, A. D., L. Coventry, G. Johnsona, and K. Renaud. 2006. Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *Int. J. Hum.–Comput. St.* 63:128–152. doi: 10.1016/j.ijhcs.2005.04.020.
6. Gao, H. 2010. A new graphical password scheme resistant to shoulder-surfing. *Conference (International) on Cyberworlds*. Singapore: IEEE. 94–199. doi: 10.1109/CW.2010.34.
7. Rittenhouse, R. G., J. A. Chaudry, and M. Lee. 2013. Security in graphical authentication. *Int. J. Security Appl.* 7(3):347–356.
8. Juels, A., and M. Sudan. 2006. A fuzzy vault scheme. *Design. Code. Cryptogr.* 38(2):237–257. doi: 10.1007/s10623-005-6343-z.
9. Boyarinov, I. M., A. A. Davydov, E. M. Mamedli, and Yu. B. Smerkis. 1983. Ispol'zovanie pomekhoustoychivogo kodirovania dlya zashchity informatsii ot oshibok operatora [Use of noise-tolerant encoding for protection of data from operator error]. *Automat. Rem. Contr.* 2:5–49.

10. Hoang, T., and D. Choi. 2014. Secure and privacy enhanced gait authentication on smart phone. *Sci. World J.* Vol. 2014. Article ID 438254. 8 p. doi: 10.1155/2014/438254. Available at: https://www.hindawi.com/journals/tswj/2014/438254/ (accessed May 11, 2017).
11. Juels, A., and M. Wattenberg. 1999. Fuzzy commitment scheme. *6th ACM Conference on Computer and Communications Security Proceedings*. 28–36. doi: 10.1145/319709.319714.
12. Uludag, U., S. Pankanti, and A. K. Jain. 2005. Fuzzy vault for fingerprints. *Audio- and video-based biometric person authentication*. Eds. T. Kanade, A. K. Jain, and N. K. Ratha. Lecture notes in computer science ser. Springer. 3546:310–319. doi: 10.1007/11527923_32.
13. Bryukhomitsky, Yu. A. 2009. Statisticheskie metody raspoznavaniya klaviaturnogo pocherka [Statistical methods of keystroke dynamics ecognition]. *Izvestiya Yuzhnogo federal'nogo universiteta* [Bull. SFedU] 11(100):13–147.
14. Khitsenko, V. E., and D. S. Krutohvostov. 2014. Increasing the reliability of authentication keyboard handwriting. *12th Conference (International) on Actual Problems of Electronics Instrument Engineering Proceedings*. 262–265. doi: 10.1109/APEIE.2014.7040894.
15. Akhmetov, B. S., A. I. Ivanov, T. S. Kartbaev, A. U. Malygin, and K. Mukapil. 2013. Biometric dynamic personality authentication in open information space. *Int. J. Computer Technology Appl.* 4(5):846–855.
16. Basic Key Logger. Available at: https://sites.google.com/site/basiclabbook/keyloggerbasiclabbook/ (accessed May 17, 2017).
17. Statzilla. Statistika onlayn [Statistics online]. Available at: https://online.statzilla.ru (accessed May 13, 2017).
18. Grassi, P. A., M. E. Garcia, and J. L. Fenton. 2017. Digital identity guidelines. NIST Special Publication 800-63-3. Available at: https://pages.nist.gov/800-63-3/sp800-63-3.html (accessed May 11, 2017).

Received May 26, 2017

Contributors

Kosolapov Jury V. (b. 1982) — Candidate of Science (PhD) in technology, associate professor, Institute of Mathematics, Mechanics, and Computer Science named after I. I. Vorovich, Southern Federal University, 105/42 Bol'shaya Sadovaya Str., Rostov-on-Don 344006, Russian Federation; itaim@mail.ru

Gordienko Dmitry E. (b. 1993) — student, Institute of Mathematics, Mechanics, and Computer Science named after I. I. Vorovich, Southern Federal University, 105/42 Bol'shaya Sadovaya Str., Rostov-on-Don 344006, Russian Federation; dmgordienko@gmail.com

Myshko Anastasia S. (b. 1993) — bachelor, Institute of Mathematics, Mechanics, and Computer Science named after I. I. Vorovich, Southern Federal University, 105/42 Bol'shaya Sadovaya Str., Rostov-on-Don 344006, Russian Federation; me.metida@gmail.com

SYSTEMS AND MEANS OF INFORMATICS 2018. Vol. 28. No. 1. P. 177–191

SITUATIONAL ONLINE RESOURCE PLANNING IN ACCORDANCE WITH MANDATORY AND ORIENTING RULES

A. V. Ilyin[1] *and V. D. Ilyin*[2]

Abstract: The technology of situational resource planning based on online services is proposed. A key role is assigned to the expert planner, who interacts with online services for forming the situation portraits and resource allocation, assesses the situations, forms systems of mandatory and orienting rules, makes requests for the calculation of plans, and estimates their feasibility and effectiveness. The foundations of methodologies for situational online resource allocation and budgeting taking into account the priorities of expense items are outlined. Formulation of the linear problem of resource allocation based on the system of mandatory and orienting rules and the method of target displacement of solution executed by the resource planning service are presented. The statement and the principle of solving the budget planning problem are given, taking into account the uncertainty of the data represented by numeric segments. The online service "Cost planning" used to solve the problems of planning budgets (state, corporate, etc.) is briefly described.

Keywords: situational resource planning; portrait of situation; online service; mandatory and orienting rules; linear problem of resource allocation; target displacement of solution; budget planning; priorities of expense items; interval cost planning

DOI: 10.14357/08696527180113

1 Introduction

In our days of intensive development of cloud computing and online services for various purposes (navigation, education, etc.) [1–3], the attention of researchers and information technology developers is attracted to the idea of integrated service-based automation of various activities. Since the mid-1990s, the importance of successful implementation of this idea (called "digital economy") has been steadily growing and associated with the competitiveness of corporations and countries [4–8].

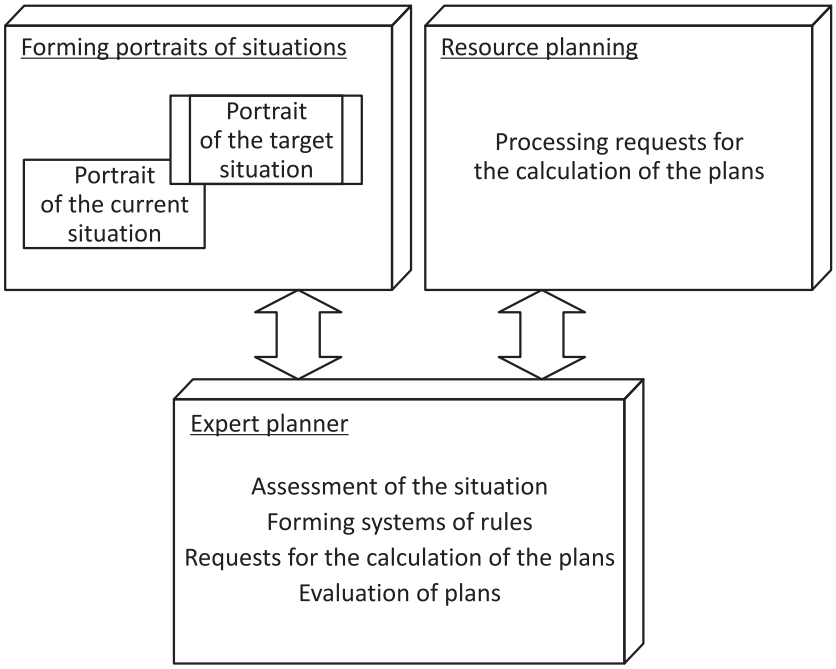
The article serves as a thematic continuation of the articles [9, 10] dedicated to situational informatization of the population activities considered as an aggregate

[1]State Research Institute of Aviation Systems, 7 Viktorenko Str., Moscow 125319, Russian Federation

[2]Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

of education, production, and other kinds of activity. Situational informatization is studied as a means of stage-by-stage organizational and technological improvement of the object through planned transitions from the starting situation to the target one. The situations are presented by formalized descriptions of the predefined set of states spaces (defence, economics, etc.), which characterize the country's potential [11]. The object is monitored on the basis of portraits of the current situations. Information technologies based on the situational informatization methodology are to be implemented in the human–machine environment for problem solving ("s-environment") [12] which serves as an infrastructural base of online services (banking, logistics, etc.).

The authors participate in the scientific research "Creating the methodology of informatization of normalized economic mechanism and software implementation of expert resource planning based on e-services" at the Institute of Informatics Problems of the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences. The first phase of software implementation includes a set of Resource Planning Online Services (www.res-plan.com) for solving problems of budgeting and resource allocation in accordance with mandatory and orienting



The main components of the situational resource planning technology

rules [13, 14]. Computational methods take into account the actual incompleteness of information for planning and experience in creating and implementing resource planning technologies [15–18].

The technology of situational resource planning is based on the interaction of the expert planner with the online services for forming portraits of situations and resource planning (see the figure).

A problem is considered as the aggregate $\{Formul, Rulsys, Alg, Prog\}$, where $Formul$ is the problem statement (which includes the concepts and relations between them); $Rulsys$ — a set of systems of mandatory and orienting rules for solving the problem; Alg is the union of sets of algorithms where each set corresponds to one system from $Rulsys$; $Prog$ is the union of sets of programs, where each set corresponds to one algorithm. A description of applicability is given for each element of $Rulsys$, Alg, and $Prog$. Descriptions of applicability of the $Rulsys$ elements include the specification of the problem solver type (stand-alone computer, network computer cooperation, human–computer cooperation, etc.); the requirements for information security, etc. Descriptions of applicability of the Alg elements include data on the permissible modes of the problem solver work (automatic local, automatic distributed, interactive local, etc.), requirements for the result, etc. Descriptions of the programs applicability include data on implementation languages, operating systems, etc. [12].

2 Methodology for Situational Online Resource Allocation in Accordance with Mandatory and Orienting Rules

Linear problems of resource allocation are traditionally solved by the methods of linear programming (LP). Applicability of LP-methods is based on assumption that the data values used in the calculation are the same as the actual values at the time of decision application. It is also known that optimal plans for the practical data, as a rule, do not exist due to the incompatibility of constraints. The input data can vary quite considerably between the time of plan calculation and the time of its application. In extreme cases, the set of main variables and the system of resource constraints may also change. This requires an operational data correction.

Solution of the LP problem is typically calculated by means of the simplex method [19] or the interior point method [20]. The solution can be found only if the constraints are compatible. In case of incompatibility, the Chebyshev point is often calculated as the compromise solution. The search for this point is also performed using the simplex method.

It is known that in practice, use of LP methods is quite problematic. One problem is the mathematical incorrectness of the LP problem due to the instability of solution for small changes of the input data [21]. Another problem is the nonapplicability of solution found as a Chebyshev point in case of constraints incompatibility. Such

solutions cannot be applied because they violate constraints that must be met: it is impossible to allocate more resource than available.

Expert planner, using a program that can solve only the LP problem and the problem of search for Chebyshev point, is too limited in the choice of means to obtain the desired results. Traditional LP software does not allow an expert intervention in the search for solution. If given system of constraints is incompatible, programs propose to adjust the input data.

To make up for these shortcomings, professor V. D. Ilyin has proposed idea of the informal statement and solving a linear problem of resource allocation [22]. A. V. Ilyin has implemented this idea and developed *the method of target displacement of solution*. This method is implemented in the *technology of interactive resource allocation in accordance with the customizable system of rules*. This technology allows an expert to search for plans in accordance with his/her knowledge of the applicability and efficiency of the plans. Software implementation of the technology has been developed and tested in a number of applications [23].

In practice, an efficient solution of linear resource allocation problem is usually not the result of solving an LP problem. A concretization of the efficiency concept can vary and depends on many factors: resource stocks, the fulfillment of contractual deliveries to customers, corporate performance indicators, etc. Typically, a concretization is defined by experts on resource planning (experts planners). In each situation, the planning result depends on the skills and awareness of the expert planner. When awareness is changed, an expert can change his/her view on efficiency. Breadth of choice of feasible solutions, which are analyzed by the expert, has a significant impact on his/her final decision. In this sense, the methodical fullness of search for feasible solutions is crucial.

2.1 Rules of resource allocation

Expert planner defines rules of resource allocation in the form of requirements on the values of resource functions $F_i(\boldsymbol{x})$ — linear forms, whose values depend on vector $\boldsymbol{x}$ of allocation and numerical coefficients (vectors and matrix are written in bold).

In general case, a simple rule can be written in one of three forms:

$$F_i(\boldsymbol{x}) = c_i[p_i]\,; \;\; F_i(\boldsymbol{x}) \le c_i[p_i]\,; \;\; F_i(\boldsymbol{x}) \ge c_i[p_i]$$

where c_i is the constant; p_i is the priority of the rule ($0 < p_i \le \infty$); and square brackets denote optionality of priority.

A composite rule is a logical combination of simple rules. In terms of Boolean algebra, a simple rule is an elementary formula and a composite rule, in general case, is formed from simple rules by means of logical operations [conjunction, disjunction, negation ($\wedge$, $\vee$, $\neg$)].

Based on the analysis of the situation portrait [9, 10], expert planner performs step-by-step search for solution. At each step, he/she customizes rules that

determine the change of solution. (Any rule may remain unchanged during the search.)

The rules can be *mandatory* or *orienting*. Mandatory rules have an absolute priority ($p_i = \infty$), that is, they cannot be violated. Orienting rules specify the desired values of resource functions, setting the direction for displacement of solution.

Let $\boldsymbol{x}^0$ be a given vector of resource allocation and $\{F_i(\boldsymbol{x}) = F_i(\boldsymbol{x}^0) + h_i[p_i], h_i \neq 0\}$ a given composite rule: the simple orienting rules, related by conjunction, are enclosed in curly brackets.

Let say that the vector of allocation $\boldsymbol{x}$ satisfies the given orienting rules ($\boldsymbol{x}$ is more efficient than $\boldsymbol{x}^0$), if $F_i(\boldsymbol{x}^0) < F_i(\boldsymbol{x}) \leq F_i(\boldsymbol{x}^0) + h_i$ is true for $\forall\, h_i > 0$ and $F_i(\boldsymbol{x}^0) + h_i \leq F_i(\boldsymbol{x}) < F_i(\boldsymbol{x}^0)$ is true for $\forall\, h_i < 0$.

For example, implementation of the composite orienting rule "the supply of fuel should be increased by 700 t for the consumer K and increased by 1000 t for the consumer N" means that the supply is increased for both consumers, but not necessarily by the specified amounts exactly.

An optimization rule is defined as the special type of a composite rule. It can be written as $Q_{\min}(\boldsymbol{x}) = F_i(\boldsymbol{x})$: $P_1 \wedge \cdots \wedge P_k$ or $Q_{\max}(\boldsymbol{x}) = F_i(\boldsymbol{x})$: $P_1 \wedge \cdots \wedge P_k$, where $P_1, \ldots, P_k$ are the simple mandatory rules (constraints).

This expresses a standard formulation of the LP problem. Note that it includes the mandatory rules only.

The developed technology allows formulating and trying to solve LP problem at any step during the search for solution: an expert can set an optimization rule to any resource function and select a subsystem of rules to define a system of constraints. For example, (in case of constraints compatibility) an optimization solution can be considered as the starting point for target displacement of the solution.

2.2 General linear problem of resource allocation

Let a_{ij} $(i = 1, \ldots, m,\ j = 1, \ldots, n)$ be a consumption of the ith resource for unit of intensity of the jth activity; b_i $(i = 1, \ldots, m)$ a stock of the ith resource; and x_j $(j = 1, \ldots, n)$ intensity of the jth activity to be found. Total consumption of the ith resource is expressed by a linear form $a_{i1}x_1 + \cdots + a_{in}x_n$. A composite rule defining the system of resource constraints is $\{a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i\}$ $(i = 1, \ldots, m)$. Further, a set of efficiency indicators may be defined as $\{c_{i1}x_1 + \cdots + c_{in}x_n\}$ $(i = 1, \ldots, k)$ where c_{ij} is the ith specific efficiency indicator for unit of intensity of the jth activity. A simple rules may be defined for any efficiency indicator also. Any rule may have a priority p_i $(0 < p_i \leq \infty,\ 1 \leq i \leq m + k)$.

In general case, a two-sided constraint (conjunction of two simple rules) may be defined for each resource function. Therefore, let rewrite the overall system as $\{[b_i \leq]a_{i1}x_1 + \cdots + a_{in}x_n[\leq B_i][p_i];\ x_j \geq 0\}$ $(i = 1, \ldots, m + k,\ j = 1, \ldots, n)$ (all the coefficients from resource constraints and efficiency indicators are denoted as a_{ij};

square brackets denote optionality of constraints and priorities; and the variables x_j are nonnegative in accordance with resource allocation problem).

The general problem is the search for the vector of allocation $\boldsymbol{x} = (x_1, \ldots, x_n)$, providing the values of resource functions, which are estimated by the expert planner as the most efficient and realizable for a given situation.

The informality of the problem statement is stipulated by orientation to the computational experiment mode, which involves the possibilities of changing the input data and system of rules, governing the search for solutions. In general case, the expert planner solves a set of particular problems, having the formal statements and algorithms, and performs comparative analysis of solutions.

2.3 Target displacement of solution

The informal method of target displacement of solution is designed for expert planner who forms the system of rules and analyzes solutions in step-by-step dialogue with specialized software. On the first step, the expert can choose the initial solution (starting point) arbitrarily. By default, the software proposes a compromise solution — Chebyshev point. If the system of constraints is compatible, such solution ensures equal reserves for resource constraints; otherwise, it ensures the minimization of the maximum deficit:

$$\min_x \max_i (a_{i1}x_i + \cdots + a_{in}x_n - b_i),\ x_j \geq 0\ (i = 1, \ldots, s,\ j = 1, \ldots, n)$$

where s is the number of constraints in the system reduced to the form $\boldsymbol{Ax} \leq \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$.

The expert can also specify an optimization rule and try to solve the LP problem. In case of constraints compatibility, the solution can be considered as the initial one. Then, the expert analyzes the received values of resource functions and estimates realizability and efficiency of a solution. If the values satisfy, the solution is the final one and the target displacement is not needed. If not, the expert planner specifies the requirements for changes of some values, that is, modifies the system of rules for resource allocation. The rules define the direction and magnitude for displacement to the next point. [When the next point satisfies the expert, this is the final solution.]

Solutions can be entered into a database of possible plans for future analysis. Thus, the trajectory of the solution is stored, enabling rollback.

A step of target displacement of solution is calculated as follows.

Let $\boldsymbol{x}^0 = (x_1^0, \ldots, x_n^0)$ $(x_j^0 \geq 0,\ j = 1, \ldots, n)$ be a current point (received on the previous step), and an expert planner has defined a composite rule for displacement from $\boldsymbol{x}^0$ to a target point $\boldsymbol{x} = (x_1, \ldots, x_n)$ $(x_j \geq 0,\ j = 1, \ldots, n)$: $\{F_i(\boldsymbol{x}) = F_i(\boldsymbol{x}^0) + h_i[p_i]\}$ where $F_i(\boldsymbol{x}) = a_{i1}x_1 + \cdots + a_{in}x_n,\ i = 1, \ldots, l$; $0 < p_i < \infty$ for $h_i \neq 0$, $p_i = \infty$ for $h_i = 0$.

Formally, the system of rules for displacement can be inconsistent. Therefore, the simple rules with $h_i \neq 0$ are treated as orienting rules and the value h_i is called

the desired step (which is often different from *the actual step* of the function that can be obtained for the given set of rules). If actual and desired steps have the same sign for all the simple rules, the new point anyway increases the efficiency of the solution.

The point $\boldsymbol{x}$ is searched as follows. First, the projection to the hyperplane $a_{i1}x_1 + \cdots + a_{in}x_n = F_i(\boldsymbol{x}^0) + h_i$ for $\forall\, h_i \neq 0$ is calculated. The direction vector of the normal to this hyperplane is $(a_{i1}, \ldots, a_{in})$. So, one should change variables by $ha_{i1}, \ldots, ha_{in}$ where h is to be calculated, to find the projection. A displacement along the normal gives the function increment $h(a_{i1}^2 + \cdots + a_{in}^2)$, which is to be equal h_i; so, $h = h_i/(a_{i1}^2 + \cdots + a_{in}^2)$ (naturally, $a_{i1}^2 + \cdots + a_{in}^2 \neq 0$). Thus, the projection is $\left(x_1^0 + a_{i1}h_1/(a_{i1}^2 + \cdots + a_{in}^2)\right) \cdots \left(x_n^0 + a_{in}h_n/(a_{i1}^2 + \cdots + a_{in}^2)\right)$.

When projections are found for $\forall\, h_i \neq 0$, one gets the desired increment for each variable: $\Delta x_{ji} = a_{ij}h_i/(a_{i1}^2 + \cdots + a_{in}^2)$ $(j = 1, \ldots, n$; let indices of the functions go from 1 to s, $1 \leq s \leq l)$. Now, let us compute the *average head of normals*, using formulas

$$x_j = x_j^0 + \frac{p_1\Delta x_{j1} + \cdots + p_s\Delta x_{js}}{p_1 + \cdots + p_s} \quad (j = 1, \ldots, n).$$

The average head of normals is "closer" to the hyperplanes which define the rules with higher priorities. If the priorities are not set, they are considered to be equal to 1, and

$$x_j = x_j^0 + \frac{\Delta x_{j1} + \cdots + \Delta x_{js}}{s} \quad (j = 1, \ldots, n).$$

Then, if $\exists\, k : h_k = 0$ $(1 \leq k \leq l)$, the average head of normals is projected to the hyperplane $a_{k1}x_1 + \cdots + a_{kn}x_n = F_k(\boldsymbol{x}^0)$; otherwise, the average head itself pretends to be the result of the step.

Next, the condition $x_j \geq 0$, $j = 1, \ldots, n$, is verified and potential negative variables are changed to zero.

Finally, let us compare actual and desired steps. If they have the same sign for all the functions in the scope, the calculated point is the target $x_1, \ldots, x_n$. If not, an expert should correct the system of rules.

3 Methodology for Situational Online Budgeting Taking into Account the Priorities of Expense Items

The planned revenues of states, corporations, and most of individuals depend on sales volumes, market prices, exchange rates, and other factors that determine the changes in economic situation. The earlier the forecast is made, the less reason to represent the result as a point, i. e., exact number. However, even for the state, budgets expenditures are planned on the basis of point assumptions about the amount of income. The requests of expense items and the planning results also have a point representation. Therefore, the budget plans are inevitably inaccurate and need to be

remade in the course of implementation. Furthermore, sometimes a set of income sources and expense items changes at different levels of hierarchical details of the planned budget. All these factors should be considered when developing the budget planning methodology [13, 23].

3.1 Approach to budget planning

The problem of budget planning is considered as a specialization of the more general problem of interval planning the costs of an arbitrary resource. The problem has the informal statement containing the mandatory and orienting rules. The mandatory rules include restrictions on the consumption of the resource to ensure the feasibility of solution and limitations that define nonredundant satisfaction of the requests for resource. The orienting rules define the direction of the search for solution. A solution always satisfies the mandatory and orienting rules in the extent defined by the interval specifics of the problem. If fulfillment of the orienting rules is possible, the solution corresponding to them is treated as more efficient than any other. A set and form of the rules can be changed by an expert during the search for acceptable plan.

3.2 Principle of interval cost planning taking into account the priorities of expense items

The resource amount and the requests of expense items are specified as numeric segments. The values of the planned costs are computed as numeric segments also. First, the resource allocation problem is solved for the top-level expense items. Then, if any expense item has the detailing items, part of the resource allocated to the item is considered as the resource amount to be allocated between the detailing items, and the separate resource allocation problem is solved, etc. For example, part of money allocated to the item "Communications" can be allocated between the items "Electricity," "Internet," "Mobile phones," and "Landline phone;" after that, part of money allocated to the item "Mobile phones" can be allocated between the items representing the concrete mobile users. The number of the detailing levels is not limited.

The priorities of expense items can be specified and used in solving each particular resource allocation problem in the hierarchy. The problem has the following informal statement.

For a numeric segment $[a, A]$ $(a \geq 0,\ A > 0)$, which expresses the expected resource amount, segments $[b_i, B_i]$ $(b_i \geq 0,\ B_i > 0,\ i = 1, \ldots, n)$, which specify the requests of expense items, and weighting coefficients (priorities) of the expense items $p_i > 0$ $(i = 1, \ldots, n)$, it is required to find a cost plan $[x_i, X_i]$: $\{0 \leq x_i \leq b_i,\ X_i \leq B_i,\ i = 1, \ldots, n\}$. Depending on presence of the resource shortage for sum of the left bounds and sum of the right bounds of the requests, one of the following situations takes place:

(1) $b_1 + \cdots + b_n > a$, $B_1 + \cdots + B_n > a$. In this case, the problem for the left bounds is to be solved and then, the *problem for the right bounds* (see below);

(2) $b_1 + \cdots + b_n \le a$, $B_1 + \cdots + B_n > a$. In this case, the left bounds are set equal to the minimum requests ($x_i = b_i$) and the problem for the right bounds is to be solved;

(3) $b_1 + \cdots + b_n > a$, $B_1 + \cdots + B_n \le a$. In this case, the problem for the left bounds is to be solved and the right bounds are set equal to the maximum requests ($X_i = B_i$); or

(4) $b_1 + \cdots + b_n \le a$, $B_1 + \cdots + B_n > a$. In this case, there is no problem: the left bounds are set equal to the minimum requests ($x_i = b_i$) and the right bounds are set equal to the maximum requests ($X_i = B_i$).

The mandatory rule for solving the problem for the left bounds: $x_1 + \cdots + x_n = a$.

The orienting rules for solving the problem for the left bounds are the proportions $x_i/x_j = p_i b_i/(p_j b_j)$ for each $1 \le i \le n$, $1 \le j \le n$, where $b_j > 0$ (for $b_j = 0$, obviously, $x_j = 0$).

The mandatory rule for solving the problem for the right bounds:

$$X_1 + \cdots + X_n = A\,.$$

The orienting rules for solving the problem for the right bounds are the proportions:

$$\frac{X_i - x_i}{X_j - x_j} = \frac{p_i(B_i - b_i)}{p_j(B_j - b_j)} \text{ for each } 1 \le i \le n,\ 1 \le j \le n\,,$$

where $\{B_i > b_i, B_j > b_j\}$, and

$$\frac{X_i}{x_j} = \frac{p_i B_i}{p_j B_j} \text{ for each } 1 \le i \le n,\ 1 \le j \le n\,,$$

where $\{B_i = b_i,\ B_j = b_j\}$.

The iterative algorithms for solving the problems for the left and right bounds are described in [13].

3.3 Online service for budgeting taking into account the priorities of expense items

The interval cost planning method taking into account the priorities of expense items is implemented in the working online service which drastically enhances efficiency and flexibility of budget planning.

In a client application of the "Cost Planning" service, user specifies minimum amount of the resource as the sum of opening balance and expected income in the worst case scenario. The maximum amount of the resource should be specified as the sum of opening balance and expected income in the optimal scenario.

User also specifies a table of expense items and for each row, the lowest and the highest expected costs (or exact value) can be entered — the requests of the expense items. A separate table of details can be created for any expense item. The number

of detail levels is not limited. The priorities (the weighting coefficients) can be specified for any table. Some requests can be marked as obligatory (e. g., wages or rents can rarely be reduced). Different applied precision (minimal significant value) can be set for data and results for any table.

Then, when user commands 'Allocate' from client application, it connects to the service via Internet and sends it a query for resource planning. The service (program which works on reliable server in 24/7 mode) receives the query, performs computations, and immediately sends the results back to the client application. The results are the values 'Allocate min.' and 'Allocate max.' for each expense item — the plan for the worst and the best scenarios. Sum of 'Allocate min.' values complies the specified minimum amount of the resource and sum of 'Allocate max.' values complies the specified maximum. The application also displays values 'Allocate avg.' (so, user can see an approximate resource allocation).

Afterwards, in the course of the plan implementation, when a part of the resource is received or spent, or more precise information on expected income or costs is obtained, user inputs the corresponding data in client application, executes the command 'Allocate' again, and gets the refined results. If the exact resource amount is specified (i. e., minimum = maximum), then the received values 'Allocate max.' can be treated as exact decision of the cost planning task.

There are no restrictions on scale of the budgeting tasks. The samples for enterprise and family budgets are delivered within client application package [14].

4 Concluding Remarks

1. The methodology for online resource allocation in accordance with mandatory and orienting rules significantly extends the traditional arsenal of facilities for solving linear problems of resource allocation. The most important new feature is the ability to perform step-by-step search for the most efficient and realizable solution of the general linear problem of resource allocation. At any step, an expert planner can analyze the values of resource functions and customize the system of orienting and mandatory rules, governing the search. If the value of some "objective" function is estimated as the most efficient, an expert can set the mandatory rule of fixing the function value.
2. The advantages of the methodology for variational online budgeting taking into account the priorities of expense items:
   - if user specifies the bounds for resource and requests cautiously and follows the plan prepared with the Service, then the probability of going beyond the budget is drastically reduced;
   - for each expense item, user beforehand sees the bounds for possible costs and narrows them in the course of the plan implementation;

- if upper bound is less than the minimum request for some item, then user can timely attract investments, or exclude the item, or correct other costs;
- if the planning results are too "tight," user can temporarily exclude any expense item from consideration: it can be done by setting a "tick" in the corresponding cell of the table;
- user can simulate any real cost: set minimum request equal to maximum, mark it as obligatory, execute the command 'Allocate,' and see the changes of bounds for the rest of expense items; and
- user can manually adjust the planning results.

The methodology of the variational interval budgeting in a system with hierarchical structure of expense items, where priorities may be set at any level of hierarchy, and the online service "Cost Planning," which implements this methodology, do not have the known analogues.

References

1. Trumba Corp. 2007. Five benefits of Software as a Service. Available at: http://www.trumba.com/connect/knowledgecenter/pdf/Saas_paper_WP-001.pdf (accessed February 4, 2018).
2. Kavakli, E., C. Kalloniatis, H. Mouratidis, and G. Gritzalis. 2015. Privacy as an integral part of the implementation of cloud solutions. *Comput. J.* 58(10):2213–2224. doi: 10.1093/comjnl/bxu118.
3. Jede, A., and F. Teuteberg. 2016. Understanding socio-technical impacts arising from software as-a-service usage in companies. *Business Inform. Syst. Eng.* 58(3):161–176. doi: 10.1007/s12599-016-0429-1.
4. Tapscott, D. 1996. *The digital economy: Promise and peril in the age of networked intelligence*. New York, NY: McGraw-Hill. 342 p.
5. Christensen, C. M. 1997. *The innovator's dilemma: When new technologies cause great firms to fail*. Boston: Harvard Business School Press. Available at: http://www.hbs.edu/faculty/Pages/item.aspx?num=46 (accessed February 4, 2018).
6. Oxford Economics. 2015. The new digital economy: How it will transform business. Available at: http://www.pwc.com/mt/en/publications/assets/the-new-digital-economy.pdf (accessed February 4, 2018).
7. G20 Summit. 2016. G20 digital economy development and cooperation initiative. Available at: http://en.kremlin.ru/supplement/5111 (accessed February 4, 2018).
8. Government of the Russian Federation. 2017. Tsifrovaya ekonomika Rossiyskoy Federatsii: Programma, utverzhdennaya rasporyazheniem Pravitel'stva Rossiyskoy Federatsii ot 28 iyulya 2017 g. No. 1632-r [The program "Digital Economy of the Russian Federation" approved by Government Order No. 1632-r dated July 28, 2017]. 87 p. Available at: http://d-russia.ru/wp-content/uploads/2017/07/programma-tsifrov-econ.pdf (accessed February 4, 2018).

9. Ilyin, A. V., and V. D. Ilyin. 2017. Osnovy kontseptsii situatsionnoy informatizatsii zhiznedeyatel'nosti [Basics of the concept of situational informatization of population activities]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 27(3):188–201. doi: 0.14357/08696527170315.
10. Ilyin, A. V., and V. D. Ilyin. 2017. Situatsionnaya informatizatsiya zhiznedeyatel'nosti: model' ob"ekta [Situational informatization of population activities: The model of the object]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 27(4):187–196. doi: 10.14357/08696527170415.
11. Ilyin, V. D. 1996. *Osnovaniya situatsionnoy informatizatsii* [Fundamentals of situational informatization]. Moscow: Nauka, Fizmatlit. 180 p.
12. Ilyin, A. V., and V. D. Ilyin. 2016. Sozdanie cheloveko-mashinnoy sredy resheniya zadach [Creation of a human–machine environment for problem solving]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 26(4):149–161. doi: 10.14357/08696527160413.
13. Ilyin, A. V. 2015. Internet-servis planirovaniya raskhodov [The online service for cost planning]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 25(2):111–122. doi: 10.14357/08696527150207.
14. Ilyin, A. V. 2018. Online resource planning services. Available at: https://www.resplan.com (accessed February 4, 2018).
15. Al-Mashari, M. 2002. Enterprise resource planning (ERP) systems: A research agenda. *Ind. Manage. Data Syst.* 103(1):22–27. doi: 10.1108/02635570310456869.
16. Umble, E. J., R. R. Haft, and M. M. Umble. 2003. Enterprise resource planning: Implementation procedures and critical success factors. *Eur. J. Oper. Res.* 146(2):241–257. doi: 10.1016/S0377-2217(02)00547-7.
17. Marnewick, C., and L. Labuschagne. 2005. A conceptual model for enterprise resource planning (ERP). *Inform. Manage. Computer Security* 13(2):144–155. doi: 10.1108/09685220510589325.
18. Amoako-Gyampah, K. 2007. Perceived usefulness, user involvement and behavioral intention: An empirical study of ERP implementation. *Comput. Hum. Behav.* 23(3):1232–1248. doi: 10.1016/j.chb.2004.12.002.
19. Dantzig, G. B. 1963. *Linear programming and extensions*. Princeton, NJ: Princeton University Press and the RAND Co. 656 p.
20. Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* 4(4):373–395.
21. Tikhonov, A. N., and V. Y. Arsenin. 1977. *Solutions of ill-posed problems*. New York, NY: Winston. 258 p.
22. Ilyin, A. V., and V. D. Ilyin. 1995. *Arkhitektura vychislitel'nogo yadra kompleksa programmnykh sredstv resursnogo obosnovaniya resheniy* [Architecture of computational kernel of software designed to search for resource management decisions]. Moscow: Institute of Informatics Problems of the Russian Academy of Sciences. 23 p.
23. Ilyin, A. V. 2013. *Ekspertnoe planirovanie resursov* [Expert resource planning]. Moscow: Institute of Informatics Problems of the Russian Academy of Sciences. 58 p.

Received February 10, 2018

Contributors

Ilyin Alexander V. (b. 1975) — Candidate of Science (PhD) in technology, leading engineer, State Research Institute of Aviation Systems, 7 Viktorenko Str., Moscow 125319, Russian Federation; ilyin@res-plan.com

Ilyin Vladimir D. (b. 1937) — Doctor of Science in technology, professor, Head of Laboratory, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; vdilyin@ipiran.ru

СИТУАЦИОННОЕ ОНЛАЙН-ПЛАНИРОВАНИЕ РЕСУРСОВ ПО ОБЯЗАТЕЛЬНЫМ И ОРИЕНТИРУЮЩИМ ПРАВИЛАМ

***А. В. Ильин*[1], *В. Д. Ильин*[2]**

[1]Государственный научно-исследовательский институт авиационных систем, ilyin@res-plan.com

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vdilyin@ipiran.ru

Аннотация: Предложена технология ситуационного планирования ресурсов на основе онлайн-сервисов. Ключевая роль отведена эксперту-планировщику, который, взаимодействуя с онлайн-сервисами формирования портретов ситуаций и планирования ресурсов, оценивает ситуации, формирует системы обязательных и ориентирующих правил, делает запросы на расчеты планов и оценивает их реализуемость и эффективность. Изложены основы методологий ситуационного онлайн-распределения ресурсов и планирования бюджетов с учетом приоритетов расходных статей. Приведены постановка задачи линейного распределения ресурсов на основе системы обязательных и ориентирующих правил и метод целевого перемещения решения, выполняемого сервисом планирования ресурсов. Приведены постановка и принцип решения задачи планирования бюджета, учитывающие неопределенность данных, представленных вещественными отрезками. Дано краткое описание работы онлайн-сервиса планирования расходов, используемого для решения задач планирования бюджетов (государственного, корпоративного и др.).

Ключевые слова: ситуационное планирование ресурсов; портрет ситуации; онлайн-сервис; обязательные и ориентирующие правила; линейная задача распределения ресурсов; целевое перемещение решения; планирование бюджета; приоритеты расходных статей; интервальное планирование расходов

DOI: 10.14357/08696527180113

Литература

1. Trumba Corp. Five benefits of Software as a Service. 2017. http://www.trumba.com/connect/knowledgecenter/pdf/Saas_paper_WP-001.pdf.
2. *Kavakli E., Kalloniatis C., Mouratidis H., Gritzalis G.* Privacy as an integral part of the implementation of cloud solutions // Comput. J., 2015. Vol. 58. No. 10. P. 2213–2224. doi: 10.1093/comjnl/bxu118.
3. *Jede A., Teuteberg F.* Understanding socio-technical impacts arising from software as-a-service usage in companies // Business Inform. Syst. Eng., 2016. Vol. 58. No. 3. P. 161–176. doi: 10.1007/s12599-016-0429-1.
4. *Tapscott D.* The digital economy: Promise and peril in the age of networked intelligence. — New York, NY, USA: McGraw-Hill, 1996. 342 p.
5. *Christensen C. M.* The innovator's dilemma: When new technologies cause great firms to fail. — Boston: Harvard Business School Press, 1997. http://www.hbs.edu/faculty/Pages/item.aspx?num=46.
6. Oxford Economics. The new digital economy: How it will transform business. 2015. http://www.pwc.com/mt/en/publications/assets/the-new-digital-economy.pdf.
7. G20 digital economy development and cooperation initiative. G20 Summit, September 5, 2016. http://en.kremlin.ru/supplement/5111.
8. Цифровая экономика Российской Федерации: Программа, утвержденная распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р. 87 с. http://d-russia.ru/wp-content/uploads/2017/07/programma-tsifrov-econ.pdf.
9. *Ильин А. В., Ильин В. Д.* Основы концепции ситуационной информатизации жизнедеятельности // Системы и средства информатики, 2017. Т. 27. № 3. С. 188–201. doi: 0.14357/08696527170315.
10. *Ильин А. В., Ильин В. Д.* Ситуационная информатизация жизнедеятельности: модель объекта // Системы и средства информатики, 2017. Т. 27. № 4. С. 187–196. doi: 10.14357/08696527170415.
11. *Ильин В. Д.* Основания ситуационной информатизации. — М.: Наука, Физматлит, 1996. 180 с.
12. *Ильин А. В., Ильин В. Д.* Создание человеко-машинной среды решения задач // Системы и средства информатики, 2016. Т. 26. № 4. С. 149–161. doi: 10.14357/08696527160413.
13. *Ильин А. В.* Интернет-сервис планирования расходов // Системы и средства информатики, 2015. Т. 25. № 2. С. 111–122. doi: 10.14357/08696527150207.
14. *Ильин А. В.* Интернет-сервисы планирования ресурсов, 2016. https://www.resplan.com.
15. *Al-Mashari M.* Enterprise resource planning (ERP) systems: A research agenda // Ind. Manage. Data Syst., 2002. Vol. 103. No. 1. P. 22–27. doi: 10.1108/02635570310456869.
16. *Umble E. J., Haft R. R., Umble M. M.* Enterprise resource planning: Implementation procedures and critical success factors // Eur. J. Oper. Res., 2003. Vol. 146. No. 2. P. 241–257. doi: 10.1016/S0377-2217(02)00547-7.
17. *Marnewick C., Labuschagne L.* A conceptual model for enterprise resource planning (ERP) // Inform. Manage. Computer Security, 2005. Vol. 13. No. 2. P. 144–155. doi: 10.1108/09685220510589325.

18. *Amoako-Gyampah K.* Perceived usefulness, user involvement and behavioral intention: An empirical study of ERP implementation // Comput. Hum. Behav., 2007. Vol. 23. No. 3. P. 1232–1248. doi: 10.1016/j.chb.2004.12.002.
19. *Dantzig G. B.* Linear programming and extensions. — Princeton, NJ, USA: Princeton University Press and the RAND Co., 1963. 656 p.
20. *Karmarkar N.* A new polynomial-time algorithm for linear programming // Combinatorica, 1984. Vol. 4. No. 4. P. 373–395.
21. *Tikhonov A. N., Arsenin V. Y.* Solutions of ill-posed problems. — New York, NY, USA: Winston, 1977. 258 p.
22. *Ильин А. В., Ильин В. Д.* Архитектура вычислительного ядра комплекса программных средств ресурсного обоснования решений. — М.: ИПИ РАН, 1995. 23 с.
23. *Ильин А. В.* Экспертное планирование ресурсов. — М.: ИПИ РАН, 2013. 58 с.

Поступила в редакцию 10.02.2018

ОБ АВТОРАХ

Адамович Игорь Михайлович (р. 1934) — кандидат технических наук, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Борисов Андрей Владимирович (р. 1965) — доктор физико-математических наук, главный научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Босов Алексей Вячеславович (р. 1969) — доктор технических наук, главный научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Волков Олег Игоревич (р. 1964) — ведущий программист Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Гаврилов Виктор Евдокимович (р. 1950) — старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Гончаренко Мирослав Богданович (р. 1991) — аспирант кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

Гордиенко Дмитрий Евгеньевич (р. 1993) — студент Института математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета

Грушо Александр Александрович (р. 1946) — доктор физико-математических наук, профессор, заведующий лабораторией Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Грушо Николай Александрович (р. 1982) — кандидат физико-математических наук, старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Гузев Олег Юрьевич (р. 1980) — кандидат технических наук, исследователь Центра научных исследований и перспективных разработок, ОАО «ИнфоТеКС»

Забежайло Михаил Иванович (р. 1956) — доктор физико-математических наук, доцент, заведующий лабораторией Института проблем информатики Фе-

дерального исследовательского центра «Информатика и управление» Российской академии наук

Захарова Татьяна Валерьевна (р. 1962) — кандидат физико-математических наук, доцент кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Зацаринный Александр Алексеевич (р. 1951) — доктор технических наук, профессор, заместитель директора Федерального исследовательского центра «Информатика и управление» Российской академии наук

Зиганшина Файруза Тахваловна (р. 1985) — кандидат физико-математических наук, доцент кафедры инженерной графики Уфимского государственного нефтяного технического университета

Иванов Алексей Владимирович (р. 1976) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Ильин Александр Владимирович (р. 1975) — кандидат технических наук, ведущий инженер Государственного научно-исследовательского института авиационных систем

Ильин Владимир Дмитриевич (р. 1937) — доктор технических наук, профессор, заведующий лабораторией Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Исмагилова Альбина Сабирьяновна (р. 1979) — доктор физико-математических наук, доцент, профессор кафедры управления информационной безопасностью Башкирского государственного университета

Корепанов Эдуард Рудольфович (р. 1966) — кандидат технических наук, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Косолапов Юрий Владимирович (р. 1982) — кандидат технических наук, доцент Института математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета

Мышко Анастасия Сергеевна (р. 1993) — бакалавр Института математики, механики и компьютерных наук им. И. И. Воровича Южного федерального университета

Николаев Андрей Владимирович (р. 1973) — кандидат физико-математических наук, старший научный сотрудник Института химической физики им. Н. Н. Семёнова Российской академии наук

Писковский Виктор Олегович (р. 1963) — кандидат физико-математических наук, старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Сенчило Владимир Викторович (р. 1963) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Синицын Владимир Игоревич (р. 1968) — доктор физико-математических наук, доцент, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, главный научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Спивак Семён Израилевич (р. 1945) — доктор физико-математических наук, профессор, заведующий кафедрой Башкирского государственного университета; заведующий лабораторией Института нефтехимии и катализа Российской академии наук

Судариков Игорь Валерьевич (р. 1989) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Тимонина Елена Евгеньевна (р. 1952) — доктор технических наук, профессор, ведущий научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Чижов Иван Владимирович (р. 1984) — кандидат физико-математических наук, доцент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Яковлев Олег Альбертович (р. 1992) — младший научный сотрудник Орловского филиала Федерального исследовательского центра «Информатика и управление» Российской академии наук

Правила подготовки рукописей статей для публикации в журнале «Системы и средства информатики»

Журнал «Системы и средства информатики» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информационных технологий.

Журнал издается на русском языке. По специальному решению редколлегии отдельные статьи могут печататься на английском языке.

Тематика журнала охватывает следующие направления:

- информационно-телекоммуникационные системы и средства их построения;
- архитектура и программное обеспечение вычислительных машин, комплексов и сетей;
- методы и средства защиты информации.

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

   Публикация предоставленной автором(ами) рукописи не должна нарушать положений глав 69, 70 раздела VII части IV Гражданского кодекса, которые определяют права на результаты интеллектуальной деятельности и средства индивидуализации, в том числе авторские права, в РФ.

   Ответственность за нарушение авторских прав, в случае предъявления претензий к редакции журнала, несут авторы статей.

   Направляя рукопись в редакцию, авторы сохраняют свои права на данную рукопись и при этом передают учредителям и редколлегии журнала неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на перевод ее на английский язык, а также на ее распространение в России и за рубежом. Каждый автор должен представить в редакцию подписанный с его стороны «Лицензионный договор о передаче неисключительных прав на использование произведения», текст которого размещен по адресу http://www.ipiran.ru/publications/licence.doc. Этот договор может быть представлен в бумажном (в 2-х экз.) или в электронном виде (отсканированная копия заполненного и подписанного документа).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.

2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.

3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

   Возвращение рукописи на доработку не означает ее принятия к печати.

   Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам.

   Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.

5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.
6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или LaTeX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно.
7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.

   Параметры страницы: формат — A4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху и снизу — 2, от края до нижнего колонтитула — 1,3.

   Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

   Рекомендуемый объем рукописи — не свыше 15 страниц указанного формата. При превышении указанного объема редколлегия вправе потребовать от автора сокращения объема рукописи.

   Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

   Все страницы рукописи нумеруются.

   Шаблоны примеров оформления представлены в Интернете:

   http://www.ipiran.ru/publications/collected/template.doc
8. Статья должна содержать следующую информацию на ***русском и английском языках***:
   - название статьи;
   - Ф.И.О. авторов, на английском можно только имя и фамилию;
   - место работы, с указанием города и страны и электронного адреса каждого автора;
   - сведения об авторах, в соответствии с форматом, образцы которого представлены на страницах:
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp и
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
   - аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных. Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком. В аннотации не должно быть ссылок на литературу и, по возможности, формул;
   - ключевые слова — желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.
   - источники финансирования работы (ссылкм на гранты, проекты, поддерживающие организации и т. п.
9. Требования к спискам литературы.
   Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в каждом из списков литературы в порядке первых упоминаний.

Списки литературы представляются в двух вариантах:

(1) **Список литературы к русскоязычной части**. Русские и английские работы — на языке и в алфавите оригинала.

(2) **References**. Русские работы и работы на других языках — в латинской транслитерации с переводом на английский язык; английские работы и работы на других языках — на языке оригинала.

Необходимо для составления списка "References" пользоваться размещенной на сайте http://www.translit.net/ru/bgn/ бесплатной программой транслитерации русского текста в латиницу.

Список литературы "References" приводится полностью отдельным блоком, повторяя все позиции из списка литературы к русскоязычной части, независимо от того, имеются или нет в нем иностранные источники. Если в списке литературы к русскоязычной части есть ссылки на иностранные публикации, набранные латиницей, они полностью повторяются в списке "References".

Примеры ссылок на различные виды публикаций в списке "References":

Описание статьи из журнала:

Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

Описание статьи из электронного журнала:

Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Описание материалов конференций:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Описание книги (монографии, сборники):

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Описание переводной книги (в списке литературы к русскоязычной части необходимо указать: / Пер. с англ. — после названия книги, а в конце ссылки указать оригинал книги в круглых скобках):

1. В русскоязычной части:
   Тимошенко С. П., Янг Д. Х., Уивер У. Колебания в инженерном деле / Пер. с англ. — М.: Машиностроение, 1985. 472 с. (*Timoshenko S. P., Young D. H., Weaver W.* Vibration problems in engineering. — 4th ed. — N.Y.: Wiley, 1974. 521 p.)

2. В англоязычной части:
   Timoshenko, S. P., D. H. Young, and W. Weaver. 1974. *Vibration problems in engineering*. 4th ed. N.Y.: Wiley. 521 p.

Описание неопубликованного документа:

Latypov, A. R., M. M. Khasanov, and V. A. Baikov. 2004. Geology and production (NGT GiD). Certificate on official registration of the computer program No. 2004611198. (In Russian, unpubl.)

Описание интернет-ресурса:

Pravila tsitirovaniya istochnikov [Rules for the citing of sources]. Available at: http://www.scribd.com/doc/1034528/ (accessed February 7, 2011).

Описание диссертации или автореферата диссертации:

Semenov, V. I. 2003. Matematicheskoe modelirovanie plazmy v sisteme kompaktnyy tor [Mathematical modeling of the plasma in the compact torus]. D.Sc. Diss. Moscow. 272 p.

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

Описание ГОСТа:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. Moscow: Standardinform Publs. 10 p.

Описание патента:

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

10. Присланные в редакцию материалы авторам не возвращаются.
11. При отправке файлов по электронной почте просим придерживаться следующих правил:
    - указывать в поле subject (тема) название журнала и фамилию автора;
    - использовать attach (присоединение);
    - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
12. Журнал «Системы и средства информатики» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

Адрес редакции журнала «Системы и средства информатики»:

Москва 119333, ул. Вавилова, д. 44, корп. 2, ФИЦ ИУ РАН

Тел.: +7 (499)135-86-92 Факс: +7 (495)930-45-05

e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич)

http://www.ipiran.ru/journal/collected

Requirements for manuscripts submitted to Journal "Systems and Means of Informatics"

Journal "Systems and Means of Informatics" publishes theoretical, review, and discussion articles on the research and development in the field of information technology.

The journal is published in Russian. By a special decision of the editorial board, some articles can be published in English.

Topics covered include the following areas:

- information and communication systems and tools of their design;
- architecture and software of computational complexes and networks; and
- methods and tools of information protection.

1. The Journal publishes original articles which have not been published before and are not intended for simultaneous publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editorial Board, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

   Agreement on the transfer of rights to publish:

   "*We, the undersigned authors of the manuscript ". . . ," pass to the Founder and the Editorial Board of the Journal "Systems and Means of Informatics" the nonexclusive right to publish the manuscript of the article in Russian (or in English) in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.*

   Author(s) signature(s): (name(s), address(es), date)."

   This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

   The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article has no classified data prohibited for publication.
2. A submitted article should be attached with **the data on the author(s)** (see item 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board and other authors about revisions and final approval of the proofs.
3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing, it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.
4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.
5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.
6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or LaTeX (.tex) formats, and additionally in the .pdf format. All documents

may be sent by e-mail or provided on a CD or diskette. A hard copy submission is not necessary.

7. The recommended typesetting instructions for manuscript.

   Pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer 1.3.

   Text: font —Times New Roman, font size — 14, paragraph indent — 0.5, line spacing — 1.5, justified alignment.

   The recommended manuscript size: not more than 15 pages of the specified format. If the specified size exceeded, the editorial board is entitled to require the author to reduce the manuscript.

   Use only standard abbreviations. Avoid abbreviations in the title and abstract. The full term for which an abbreviation stands should precede its first use in the text unless it is a standard unit of measurement.

   All pages of the manuscript should be numbered.

   The templates for the manuscript typesetting are presented on site:

   http://www.ipiran.ru/publication/collected/template.doc

8. Articles should enclose data both in **Russian and English**:

   - title;
   - author's name and surname;
   - affiliation — organization, its address with ZIP code, city, country, and official e-mail address;
   - data on authors according to the format (see site):
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp and
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
   - abstract (not less than 100 words) both in Russian and in English. Abstract is a short summary of the article that can be published separately. The abstract is the main source of information on the article and it could be included in leading information systems and data bases. The abstract in English has to be an original text and should not be an exact translation of the Russian one. Good English is required. In abstracts, avoid references and formulae.
   - Indexing is performed on the basis of keywords. The use of keywords from the internationally accepted thematic Thesauri is recommended.

     Important! Keywords must not be sentences.
   - Acknowledgments.

9. References. Russian references have to be presented both in English translation and in Latin transliteration (refer http://www.translit.net/ru/bgn/).

   Please take into account the following examples of Russian references appearance:

   Article in journal:

   Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

   Journal article in electronic format:

   Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic

exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Conference proceedings:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Books and other monographs:

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Dissertation and Thesis:

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

State standards and patents:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. M.: Standardinform Publs. 10 p.

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

References in Latin transcription are presented in the original language.

References in the text are numbered according to the order of their first appearance; the number is placed in square brackets. All items from the reference list should be cited.

10. Manuscripts and additional materials are not returned to Authors by the Editorial Board.
11. Submissions of files by e-mail must include:
    - the journal title and author's name in the "Subject" field;
    - an article and additional materials have to be attached using the "attach" function;
    - an electronic version of the article should contain the file with the text and a separate file with figures.
12. "System and Means of Informatics" journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

Editorial Board address:

FRC CSC RAS, 44, block 2, Vavilov Str., Moscow 119333, Russia

Ph.: +7 (499) 135 86 92, Fax: +7 (495) 930 45 05

e-mail: rust@ipiran.ru (to Prof. Rustem Seyful-Mulyukov)

http://www.ipiran.ru/english/journal_systems.asp

SYSTEMS AND MEANS OF INFORMATICS (SISTEMY I SREDSTVA INFORMATIKI)

SCIENTIFIC JOURNAL

Volume 28 No.1 Year 2018

I N T H I S I S S U E: