

## ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004.724

*В.П. Корячко, А.П. Шибанов, Х.Л. Фам*

### НАХОЖДЕНИЕ ХАРАКТЕРИСТИК КАНАЛА ПЕРЕДАЧИ ДАННЫХ ОПТИКО-ЭЛЕКТРОННЫХ СРЕДСТВ

*Рассматривается задача нахождения характеристик канала передачи данных опτικο-электронных средств траекторных измерений летательных аппаратов. Входной поток от каждой измерительной станции является простейшим с заданным уровнем приоритета. Передача кадров через канал производится на основе оконного управления. Находятся показатели качества канала с использованием системы массового обслуживания M/G/1. Определяется степень влияния вероятности ошибки кадра при передаче через канал связи и длины окна протокола передачи на средние значения числа кадров в системе, времени пребывания кадра в системе, времени пребывания кадра в очереди, определенные для приоритетных и неприоритетных кадров.*

**Ключевые слова:** сеть передачи данных, опτικο-электронные средства, измерительная станция, траекторная информация, приоритетное обслуживание, оконное управление, система массового обслуживания M/G/1, кадры с приоритетом без прерывания обслуживания, кадры с приоритетом с прерыванием и дообслуживанием.

**Введение.** Рассматривается решение задачи нахождения вероятностно-временных характеристик канала передачи данных опτικο-электронных средств. Канал состоит из коммутатора и линии связи, соединяющей его с центром приема и обработки траекторной информации [1-4], и является частью корпоративной сети [5-8] полигонного измерительного комплекса. На входной порт коммутатора поступают информационные кадры от измерительных систем, разделяемые по приоритету. Наивысший приоритет имеет траекторная информация реального времени [9-11] или целеуказания, которые могут передаваться в двух режимах: 1) с прерыванием обработки текущего кадра и последующим дообслуживанием (абсолютный приоритет); 2) без прерывания обработки текущего кадра (относительный приоритет) [12-14]. Кадры отложенного режима не являются приоритетными и передаются тогда, когда в очереди канала нет приоритетных кадров. Траекторная информация реального времени и кадры, содержащие целеуказания, порождают пуассоновский поток реального времени с интенсивностью  $\lambda_1$ . Кадры отложен-

ного режима передачи составляют пуассоновский поток с интенсивностью  $\lambda_2$  ( $\lambda_1 + \lambda_2 = \lambda$ ). Выходной порт коммутатора канала передает информацию в центр обработки и принятия решений по управлению объектами летных испытаний. Специализированный канал функционирует по алгоритму «оконной передачи с возвратом на  $n$  шагов назад». Приоритетные кадры и кадры отложенного режима передаются через канал связи по одинаковому алгоритму (с одинаковой средней интенсивностью), то есть в выходном порту коммутатора приоритеты не учитываются. Передача кадров из входного порта в выходной осуществляется коммутационной матрицей за время, пренебрежимо малое по сравнению со временем передачи кадров через линию связи.

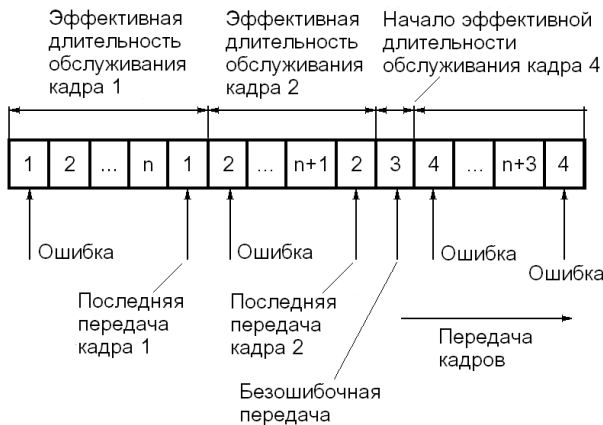
**Целью работы** является нахождение основных показателей качества канала в стационарном режиме работы при передаче приоритетных и неприоритетных данных опτικο-электронных средств траекторных измерений при различных значениях интенсивности входного потока, длины «окна» и вероятности искажения кадра при

передаче. Такими показателями качества при функционировании канала в стационарном режиме являются: среднее время передачи кадра оптоэлектронной информации, дисперсия этого времени, среднее число кадров в системе, среднее время пребывания кадра в системе, среднее число кадров в очереди, среднее время пребывания кадра в очереди, определенные для приоритетных и не приоритетных кадров.

**Теоретическая часть.** Кадры, передаваемые от измерительных систем оптоэлектронной информации, имеют длину, равную одной условной единице. Передатчик имеет право послать подряд без подтверждения  $n$  кадров. Приемник, приняв искаженный кадр, посылает передатчику квитанцию относительно малого размера  $\delta$ . В ней он указывает номер искаженного кадра, который должен быть передан повторно. При этом должны передаваться подряд и кадры, следующие за искаженным (не более  $n$ ), даже если ранее они переданы без ошибок.

Квитанции могут передаваться в этом же канале или добавляться к кадрам обратного потока. Если кадры обратного потока достаточно длинные, то это вносит дополнительную задержку в общее время передачи кадра.

Максимальное время ожидания подтверждения правильности передачи перед началом повторной передачи равно  $n-1$  кадру (рисунок 1).



**Рисунок 1 – Эффективные длительности обслуживания кадров**

Если  $q$  – вероятность безошибочной передачи кадра, то кадр  $i$  может быть отброшен приемником с вероятностью  $1 - q = p$  из-за ошибок. Тогда передатчик будет передавать имеющиеся кадры  $i+1, i+2, \dots, i+n-1$ , а потом повторно будет передан кадр  $i+n$ . Это означает, что интервал времени между началом первой передачи кадра и окончанием передачи этого кадра с учетом единицы времени  $\delta$  на передачу квитанции

равен  $1 + \delta + k(n + \delta)$  единицам времени с вероятностью  $q p^k$  (что соответствует  $k$  повторным передачам). Тогда очередь на передаче имеет распределение времени обслуживания

$$P\{X = 1 + \delta + k(n + \delta)\} = q p^k, \quad k = 0, 1, \dots$$

Первые два момента времени обслуживания  $\bar{X}$  и  $\bar{X}^2$  находятся по формулам:

$$\begin{aligned} \bar{X} &= \sum_{k=0}^{\infty} [1 + \delta + k(n + \delta)] q p^k = \\ &= q \left[ (1 + \delta) \sum_{k=0}^{\infty} p^k + (n + \delta) \sum_{k=0}^{\infty} k p^k \right], \\ \bar{X}^2 &= \sum_{k=0}^{\infty} [1 + \delta + k(n + \delta)]^2 q p^k = \\ &= q \left[ (1 + \delta)^2 \sum_{k=0}^{\infty} p^k + \right. \\ &\quad \left. + 2(1 + \delta)(n + \delta) \sum_{k=0}^{\infty} k p^k + \right. \\ &\quad \left. + (n + \delta)^2 \sum_{k=0}^{\infty} k^2 p^k \right]. \end{aligned}$$

Учитывая, что

$$\begin{aligned} \sum_{k=0}^{\infty} p^k &= \frac{1}{q}, \quad \sum_{k=0}^{\infty} k p^k = \frac{p}{q^2}, \\ \sum_{k=0}^{\infty} k^2 p^k &= \frac{p + p^2}{q^3}, \end{aligned}$$

получаем

$$\bar{X} = 1 + \frac{\delta + np}{q}, \tag{1}$$

$$\begin{aligned} \bar{X}^2 &= (1 + \delta)^2 + 2(1 + \delta)(n + \delta) \frac{p}{q} + \\ &\quad + (n + \delta)^2 \frac{p + p^2}{q^2}. \end{aligned} \tag{2}$$

Величину  $\delta$  надо учитывать в том случае, когда соотношение длины кадра и длины квитанции таково, что параметр  $\delta$  влияет на точность расчетов, либо тогда, когда квитанции передаются во встречном потоке в составе достаточно длинных кадров. Чаще всего величиной  $\delta$  можно пренебречь, так как длина квитанции составляет всего лишь несколько байт и пренебрежимо мала по сравнению с длиной кадра оптоэлектронной информации. Можно пренебречь величиной  $\delta$  и тогда, когда передача квитанции

производится в обратном параллельном потоке в кадрах относительно малой длины.

Величина  $\varepsilon = q + np$  может быть интерпретирована как смесь случайных величин:  $q$  – вероятности успешной передачи кадра с первой попытки и  $np$  – вероятности  $n$  повторных неудачных попыток. Пусть  $\delta = 0$ , тогда

$$\bar{X} = 1 + \frac{np}{q} = \frac{\varepsilon}{q}. \quad (3)$$

$$\overline{X^2} = \frac{\varepsilon^2 + n^2 p}{q^2}. \quad (4)$$

В дальнейших расчетах мы будем использовать формулы (3) и (4). Но при необходимости для расчета моментов времени передачи кадра через канал связи можно применять вместо них выражения (1) и (2).

Рассмотрим два режима приоритетной передачи кадров.

I. Канал передачи данных оптоэлектронных средств с приоритетом без прерывания обслуживания.

Обозначим через  $\bar{N}_Q^k$ ,  $k = 1, 2$  – среднее число требований в очереди  $k$ -го приоритета; через

$$\rho_k = \frac{\lambda_k}{\mu_k} = \lambda_k \bar{X}, \quad k = 1, 2$$

– нагрузку системы для  $k$ -го приоритета. Общая нагрузка системы в стационарном режиме меньше единицы, т. е.  $\rho_1 + \rho_2 < 1$ . Средние задержки  $\bar{W}_1$  – кадров высшего приоритета и  $\bar{W}_2$  – кадров отложенного режима передачи находятся для системы массового обслуживания M/G/1 [15-17] по формулам:

$$\bar{W}_1 = \frac{(\lambda_1 + \lambda_2) \bar{X}^2}{2(1 - \lambda_1 \bar{X})} = \frac{(\lambda_1 + \lambda_2)(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)},$$

$$\begin{aligned} \bar{W}_2 &= \frac{(\lambda_1 + \lambda_2) \bar{X}^2}{2(1 - \lambda_1 \bar{X})(1 - \lambda_1 \bar{X} - \lambda_2 \bar{X})} = \\ &= \frac{\lambda(\varepsilon^2 + n^2 p)}{2(q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)}. \end{aligned}$$

Средняя задержка требования  $k$ -го приоритета  $\bar{T}_k = \bar{X} + \bar{W}_k$ ,  $k = 1, 2$ . Для приоритетных кадров

$$\bar{T}_1 = \frac{\varepsilon}{q} + \frac{(\lambda_1 + \lambda_2)(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)},$$

а для кадров отложенного режима

$$\bar{T}_2 = \frac{\varepsilon}{q} + \frac{\lambda(\varepsilon^2 + n^2 p)}{2(q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)}.$$

По формуле Литтла

$$\bar{N}_Q^k = \lambda_k \bar{W}_k, \quad k = 1, 2$$

получаем выражения  $N_Q^1$  для среднего числа приоритетных кадров в очереди канала и  $N_Q^2$  – среднего числа кадров в очереди, передаваемых в отложенном режиме

$$\bar{N}_Q^1 = \frac{\lambda_1(\lambda_1 + \lambda_2)(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)},$$

$$\bar{N}_Q^2 = \frac{\lambda_2 \lambda(\varepsilon^2 + n^2 p)}{2(q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)}.$$

Выражения для  $N_1$  – среднего числа приоритетных кадров в системе и для  $N_2$  – среднего числа кадров в системе, передаваемых в отложенном режиме, находим с использованием формулы Литтла:

$$\bar{N}_k = \lambda_k \bar{T}_k, \quad k = 1, 2;$$

$$\bar{N}_1 = \frac{\varepsilon \lambda_1}{q} + \frac{\lambda_1(\lambda_1 + \lambda_2)(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)},$$

$$\bar{N}_2 = \frac{\lambda_2 \varepsilon}{q} + \frac{\lambda_2 \lambda(\varepsilon^2 + n^2 p)}{2(q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)}.$$

II. Канал передачи данных оптоэлектронных средств с приоритетом с прерыванием и дообслуживанием.

Среднее значение длительности пребывания в системе приоритетных кадров [9-11] определяется по формуле:

$$\bar{T}_1 = \frac{2\varepsilon(q - \lambda_1 \varepsilon) + \lambda_1(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)}, \quad (5)$$

а среднее значение длительности пребывания в системе кадров, передаваемых в отложенном режиме:

$$\bar{T}_2 = \frac{2\varepsilon(q - \lambda \varepsilon) + \lambda(\varepsilon^2 + n^2 p)}{2(q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)}. \quad (6)$$

Соответствующие значения среднего времени пребывания во входной очереди для приоритетных и неприоритетных кадров канала определяются по выражениям:

$$\bar{W}_1 = \frac{\lambda_1(\varepsilon^2 + n^2 p)}{2q(q - \lambda_1 \varepsilon)}, \quad (7)$$

$$\bar{W}_2 = \frac{\lambda (\varepsilon^2 + n^2 p)}{2 (q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)} + \frac{\lambda_1 \varepsilon^2}{q(q - \lambda_1 \varepsilon)}. \quad (8)$$

Используя формулы Литтла, находим средние значения:

– числа приоритетных кадров в очереди

$$\bar{N}_Q^1 = \lambda_1 \bar{W}_1;$$

– числа кадров в очереди, передаваемых в отложенном режиме,

$$\bar{N}_Q^2 = \lambda_2 \bar{W}_2;$$

– числа приоритетных кадров в системе

$$\bar{N}_1 = \lambda_1 \bar{T}_1;$$

– числа кадров в системе, передаваемых в отложенном режиме,

$$\bar{N}_2 = \lambda_2 \bar{T}_2.$$

Подставляя в эти выражения соответствующие значения из формул (5-8), получаем:

$$\bar{N}_Q^1 = \frac{\lambda_1^2 (\varepsilon^2 + n^2 p)}{2q (q - \lambda_1 \varepsilon)},$$

$$\bar{N}_Q^2 = \lambda_2 \left\{ \frac{\lambda_1 \varepsilon^2}{q (q - \lambda_1 \varepsilon)} + \frac{\lambda (\varepsilon^2 + n^2 p)}{2 (q - \lambda_1 \varepsilon)(q - \lambda \varepsilon)} \right\},$$

$$\bar{N}_1 = \frac{\varepsilon \lambda_1}{q} + \frac{\lambda_1^2 (\varepsilon^2 + n^2 p)}{2q (q - \lambda_1 \varepsilon)},$$

$$\bar{N}_2 = \frac{\lambda_2}{q - \lambda_1 \varepsilon} \left\{ \varepsilon + \frac{\lambda (\varepsilon^2 + n^2 p)}{2 (q - \lambda \varepsilon)} \right\}.$$

**Пример** (система с приоритетом без прерывания обслуживания).

Так как система должна функционировать в стационарном режиме при нагрузке  $\rho = \lambda \bar{X} < 1$ , то принимаем  $\lambda = 0,4$ . Были взяты типовые длины окон  $n = 4, 8$ . Результаты расчетов при  $n = 8$  приведены в таблице 1. По результатам расчетов можно сделать следующие выводы. При вероятности сбоя кадра  $p = 0,01$  уменьшение длины окна с  $n = 8$  до  $n = 4$  приводит к уменьшению среднего времени передачи пакетов с 1,08 до 1,04, то есть примерно на 4 %, что хорошо согласуется с тем фактом, что на практике при перегрузке приемника канала рекомендуется уменьшать размер окна.

Приоритет кадров реального времени определяет более высокое качество их обслуживания по сравнению с качеством обслуживания кадров отложенного режима. Особенно это заметно при относительно больших вероятностях искажения кадра в канале связи. В частности,

при  $p = 0,05$  и  $n = 8$  среднее время ожидания приоритетных пакетов в очереди  $\bar{W}_1 = 1,56$ , а неприоритетных –  $\bar{W}_2 = 3,60$  (отличие в 2,32 раза). Но при уменьшении вероятности сбоя кадра до 0,01 средние значения времени пребывания равны  $\bar{W}_1 = 0,47$  и  $\bar{W}_2 = 0,82$ ; соотношение между этими значениями уменьшается до 1,76. Это объясняется тем, что с уменьшением вероятности ошибки  $p$  приоритетные кадры все меньше задерживаются во входной очереди канала, создавая предпосылки и для уменьшения времени пребывания в очереди кадров отложенного времени.

**Таблица 1 – Результаты расчетов**

Параметр	Для кадров с приоритетом		Для кадров «отложенного времени»	
	$p$		$p$	
	0,05	0,01	0,05	0,01
$\bar{X}$	1,42	1,08	1,42	1,08
$\bar{X}^2$	5,57	1,82	5,56	1,82
$\sigma^2$	3,55	0,65	3,55	0,65
$3\sigma$	5,65	2,42	5,65	2,42
$\bar{W}$	1,56	0,47	3,60	0,82
$\bar{T}$	2,90	1,54	5,02	1,90
$\bar{N}$	0,60	0,31	1,01	0,38

Таблица 2 позволяет сравнить результаты расчетов основных параметров канала передачи данных оптико-электронных средств (траекторной информации) в режиме работы с абсолютным и относительным приоритетами ( $n = 8, p = 0,01, \lambda = 0,4$ ).

Кадры реального времени абсолютного приоритета быстрее передаются и меньше задерживаются в очередях, так как конкурируют только друг с другом. Кадры отложенного режима при относительном приоритете передаются лучше, чем при абсолютном приоритете, так как в этом случае их обработка в канале не прерывается.

**Таблица 2 – Параметры канала**

Параметр	Кадры реального времени		Кадры отложенного режима	
	Абс. приор.	Отн. приор.	Абс. приор.	Отн. приор.
$\bar{T}$	1,31	1,54	2,2	1,9
$\bar{W}$	0,23	0,47	1,12	0,82
$\bar{N}$	0,26	0,31	0,44	0,28

**Выводы.** Предложена методика расчета вероятностно-временных характеристик канала передачи данных с оконным управлением при условии, что на вход канала подается пуассоновский поток данных от нескольких оптоэлектронных средств обработки траекторной информации и канал функционирует в стационарном режиме. Важным в практическом отношении и наиболее сложным является расчет характеристик космического канала связи с передачей через спутник с геостационарной орбитой, в котором задержка передачи кадров и квитанций по тракту «земля-спутник-земля» составляет четверть секунды. Это существенно усложняет выбор длины окна  $n$  и параметров канала при передаче кадров оптоэлектронных средств в реальном масштабе времени. Предложенная методика направлена на уменьшение потерь приоритетных кадров траекторной информации и обеспечение их передачи в реальном масштабе времени.

*Статья подготовлена при поддержке Российского Фонда фундаментальных исследований, грант № 14-07-00106-а.*

#### **Библиографический список**

1. *Корячко В.П., Лукьянов О.В., Шибанов А.П.* Нахождение скрытого параллелизма протоколов для улучшения характеристик сети передачи данных полигонного измерительного комплекса // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 68-75.
2. *Корячко В.П., Лукьянов О.В., Шибанов А.П.* Нахождение периода занятости сети передачи данных полигонного измерительного комплекса // Успехи современной радиоэлектроники. 2014. № 6. С. 8-16.
3. *Корячко В.П., Лукьянов О.В., Шибанов А.П.* Сеть передачи измерительной информации с разделением трафика на профили // Успехи современной радиоэлектроники. 2014. № 10. С. 9-15.
4. *Корячко В.П., Шибанов А.П., Сапрыкин А.Н., Лукьянов О.В.* Планирование потоков в сети передачи данных оптоэлектронных средств // Вестник Рязанского государственного радиотехнического университета. 2014. № 50-1. С. 55-59.
5. *Перепелкин Д.А.* Алгоритм ускоренной адаптивной маршрутизации на базе протокола OSPF при динамическом добавлении элементов корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2010. № 34. С. 65-71.
6. *Корячко В.П., Перепелкин Д.А.* Алгоритм адаптивной ускоренной маршрутизации на базе протокола IGRP при динамическом добавлении элементов корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2012. № 40. С. 47-54.
7. *Перепелкин Д.А.* Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом подключении узлов и линий связи корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2013. № 4-1 (46). С. 67-75.
8. *Перепелкин Д.А.* Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом отказе узлов и линий связи корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 84-91.
9. *Корячко В.П., Шибанов В.А., Ижванов Ю.Л., Шибанов А.П.* Оценка времени передачи файла с учетом старения информации // Информационные технологии. 2010. № 10. С. 40-45.
10. *Корячко В.П., Шибанов В.А., Ижванов Ю.Л., Шибанов А.П.* Нахождение распределения выходной величины GERT-сети со старением информации // Системы управления и информационные технологии, 2009. Т. 38. № 4. С. 39-43.
11. *Корячко В.П., Шибанов В.А., Курдюмов В.В., Шибанов А.П.* Модель агрегированного канала связи со старением информации // Известия Белорусской инженерной академии, 2004. № 1-2. С. 179-182.
12. *Гнеденко Б.В., Даниелян Э.А., Димитров Б.Н. и др.* Приоритетные системы обслуживания. М.: МГУ, 1973. 447 с.
13. *Вишневский В.М., Жарких В.А.* Об одной приоритетной системе с групповым обслуживанием // Изв. АН СССР. Техническая кибернетика. 1979. № 2. С. 96-106.
14. *Вишневский В.М., Дмитриев В.П., Жданов В.С.* Основы передачи информации в вычислительных системах и сетях: учеб. пособие. М.: МГИЭМ, 1998. 162 с.
15. *Клейнрок Л.* Теория массового обслуживания: пер. с англ. М.: Машиностроение, 1979. 432 с.
16. *Бертсекас Д., Галлагер Р.* Сети передачи данных. М.: Мир, 1989. 544 с.
17. *Вишневский В.М.* Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003. 512 с.

УДК 004.72:519.2

*Д.А. Перепелкин***ДИНАМИЧЕСКОЕ ФОРМИРОВАНИЕ ТРАФИКА КОРПОРАТИВНЫХ СЕТЕЙ НА ОСНОВЕ МЕТОДА МАРШРУТИЗАЦИИ ПО ПОДСЕТЯМ**

*Предложен алгоритм динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям, повышающий эффективность процессов маршрутизации в условиях динамических изменений нагрузки на линиях связи.*

**Ключевые слова:** компьютерные сети, корпоративные сети, каналы связи, сетевой трафик, подсети, сегменты, оптимальные структуры, адаптивная маршрутизация.

**Введение.** Развитие новых сетевых технологий требует обеспечения качественного обслуживания сетевого трафика, передаваемого между телекоммуникационными устройствами. Увеличение объемов сетевого трафика в компьютерных сетях и усложнение функций взаимодействия между их удаленными компонентами может приводить к сбоям в работе. Особая роль в этих случаях отводится эффективной маршрутизации пакетов данных в условиях всплесков трафика, локальных перегрузок отдельных элементов сети.

Ввиду сложности структур современных компьютерных сетей задача маршрутизации не решается в полной мере. В большинстве случаев это связано с маршрутизаторами, не справляющимися с поддержанием таблиц маршрутизации и выбором оптимальных маршрутов для определенного класса трафика. Одним из возможных решений задачи повышения качества обслуживания трафика компьютерных сетей является использование нескольких центров управления или принятия решений при расчете и формировании таблиц маршрутизации одновременно на нескольких маршрутизаторах, которые функционируют в интересах той или иной подсети или автономной системы.

Поэтому возникает задача исследования существующих алгоритмов маршрутизации [1, 2] с целью улучшения их характеристик или создания новых алгоритмов маршрутизации.

**Цель работы** – разработка алгоритма динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям, позволяющего уменьшить трудоемкость расчета оптимальных маршрутов в условиях динамических изменений нагрузки на линиях связи и повысить эффективность процессов маршрутизации при распределении трафика.

**Постановка задачи.** В настоящее время распределенные сетевые структуры оптимально

строить на основе базовых (опорных) сетей. Такие задачи актуальны при формировании региональной телекоммуникационной инфраструктуры с учетом особенностей действующих на территории региона операторов связи. Это дает возможность сократить общие затраты на создание высокоскоростных каналов связи, построить устойчивую сетевую структуру, к которой легко подключать новые сегменты с учетом специфики предприятия.

В общем случае корпоративная сеть состоит из множества базовых (коммуникационных) узлов, к которым подключаются пользователи сети, соединенные скоростными каналами связи. Стоимость, или цена, маршрута складывается из стоимости используемых каналов связи и базовых узлов.

Для решения задачи поиска оптимальных маршрутов в корпоративных сетях широко используется алгоритм Дейкстры. Данный алгоритм применяется для построения таблиц маршрутизации в протоколе OSPF (Open Shortest Path First). Характеристики и параметры качества обслуживания протокола OSPF подробно рассмотрены в [3, 4].

Трудоемкость построения таблиц маршрутизации с использованием алгоритма Дейкстры составляет величину  $O(N^2)$ , где  $N$  – число маршрутизаторов или узлов связи в корпоративной сети. В общем случае данный алгоритм основан на присвоении узлам связи временных пометок, определяющих верхнюю границу длины маршрута от начального узла до всех остальных, которые уменьшаются с помощью некоторой итерационной процедуры; после выполнения каждой итерации одна из пометок становится постоянной, что указывает на то, что данная пометка обозначает точную длину маршрута от начального узла связи до узла связи с этой пометкой. Алгоритм заканчивает свою работу, когда просмотрены все узлы или когда найден маршрут от

начального узла связи до искомого.

В [5] установлено, что эффективность процессов маршрутизации [6, 7] и балансировки трафика [8-11] заметно повышается, если корпоративную сеть разбить на несколько подсетей и использовать при формировании и распределении трафика метод маршрутизации по подсетям.

Для решения данной задачи необходимо использовать эффективные алгоритмы сегментации структур корпоративных сетей, которые наиболее полно учитывают информацию о структуре базовой сети и связности каналов связи. В [12] предложен алгоритм формирования оптимальных структур сегментов корпоративной сети по критерию минимальной стоимости. Данный алгоритм использует информацию о базовой структуре корпоративной сети и связности каналов связи для формирования отдельных оптимальных сегментов с целью обеспечения эффективного обмена информацией между взаимодействующими устройствами или отдельными подсетями. В связи с этим целесообразно применить и использовать предложенный алгоритм для решения задачи динамического формирования трафика на основе метода маршрутизации по подсетям.

Разработка новых алгоритмов динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям позволяет уменьшить трудоемкость расчета оптимальных маршрутов в условиях динамических изменений нагрузки на линиях связи и повысить эффективность процессов маршрутизации за счет высокой производительности при распределении трафика между подсетями.

**Разработка алгоритма.** Для повышения эффективности процессов маршрутизации в корпоративных сетях предлагается алгоритм динамического формирования трафика, который позволяет уменьшить трудоемкость построения оптимальных маршрутов в условиях динамических изменений нагрузки на линиях связи. В общем случае для решения данной задачи применяется графовая модель корпоративной сети, в которой множество вершин графа соответствует множеству узлов связи или маршрутизаторов в сети, а множество ребер соответствует возможным каналам связи между этими узлами. Каждое ребро, соответствующее каналу связи, имеет свой вес. На практике весу ребра могут соответствовать стоимость аренды канала связи, затраты на оплату единицы трафика, передаваемого по каналу связи, соответствующему данному ребру, либо более сложная функция, учитывающая большее число параметров корпоративной сети.

Математическую модель корпоративной сети представим в виде неориентированного взвешенного связного графа  $G = (V, E, W)$ , где  $V$  – множество вершин (узлов связи или маршрутизаторов),  $|V| = N$ ,  $E$  – множество ребер (каналов или линий связи),  $|E| = M$ ,  $W$  – множество весов ребер (стоимость каналов связи между узлами).

Задача формирования оптимальных структур сегментов (подсетей) корпоративных сетей решается путем разбиения множества вершин графа на непересекающиеся подмножества вершин с максимально связанными между собой каналами и минимальным суммарным весом ребер, проходящих между подмножествами полученных вершин.

В соответствии с алгоритмом [12] используем следующие понятия. Обозначим через  $G_1$  такой связный подграф графа  $G$ , для которого  $G_1 = (V, E_1, W_1)$ , где  $V$  – множество вершин подграфа (узлов связи или маршрутизаторов),  $|V| = N$ ,  $E_1$  – множество ребер подграфа (каналов или линий связи),  $|E_1| = M_1$ ,  $W_1$  – множество весов ребер подграфа (стоимость каналов связи между узлами). Степень узла связи  $\text{deg}(v_i)$  подграфа  $G_1 \subset G$  – число неповторяющихся каналов связи  $e_{ij} \in E$ , инцидентных узлу  $v_i$ . Узел связи  $v_i$  называется листом дерева (подграфа  $G_1$ ), если его степень  $\text{deg}(v_i) = 1$ . Потомком называется часть минимального покрывающего дерева, имеющая узел-предок. Предком называется узел связи, имеющий одного и более потомков. Величиной связности сегментов  $Q$  называется отношение количества связей внутри сегмента к числу внешних связей у данного сегмента

$$Q = \frac{M_{in}}{M_{out}}.$$

Для разработки алгоритма динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям с учетом приведенных выше положений необходимо ввести временные пометки по каждому узлу связи в подсети и уменьшать их, пока не будет найдена постоянная пометка с минимальной оценкой

$$\min_{j=1..N_i} \{d_j\},$$

где  $d_j$  – пометка вершины с номером  $j$  ( $j = 1 .. N_i$ ),  $N_i$  – число узлов связи в подсети с номером  $i$ ).

Пусть  $d_j$  – пометка узла связи  $v_j$ ,  $v_b$  – начальный узел связи,  $v_t$  – текущий узел связи.

Рассмотрим работу алгоритма динамического формирования трафика корпоративных сетей в условиях динамических изменений нагрузки на линиях связи на основе метода маршрутизации

по подсетям. Укрупненно алгоритм имеет следующий вид.

*Шаг 1.* В графе  $G$  корпоративной сети найти минимальное покрывающее дерево с помощью алгоритма Прима. В качестве критерия оптимальности канала связи выбрать минимальную стоимость его использования.

На данном этапе необходимо найти такой связный подграф  $G_1 \subset G$ , для которого суммарный вес  $w(G_1)$  входящих в него ребер минимален.

*Шаг 2.* Выделить в полученной структуре отдельные сегменты.

Объединение в отдельные сегменты происходит последовательным объединением потомков с их предками. Вначале рассматривается в качестве предка произвольный узел связи  $v_i$  подграфа  $G_1 \subset G$ . Далее оценивается степень данного узла  $\deg(v_i)$ . Если  $\deg(v_i) = 1$ , то данный узел объединяется с узлом, с которым имеет связь, если нет – перейти к следующему узлу (потомку). Данное действие производить до тех пор, пока не будут рассмотрены все узлы связи. Заметим, что в первую очередь рассматриваются связи искомого узла с узлами, которые еще не вошли в какой-либо из сегментов. При отсутствии связи со свободным узлом связи произвести присоединение его к одному из сегментов по следующему принципу. Сначала просмотреть связи данного узла со всеми сегментами  $G_i$ . Определить сегменты  $G_i$ , с которыми имеется связь, принадлежащая минимальному покрывающему дереву  $G_1 \subset G$ . Далее с данными сегментами просмотреть все множество связей. Если у узла с каким-то из сегментов количество связей больше, чем у другого сегмента, то отнести узел в этот сегмент. Если количество связей одинаково для всех сегментов, то принадлежность узла к сегменту определить связью с минимальным весом.

*Шаг 3.* Оценка полученной структуры с помощью величины связности сегментов  $Q$ .

Заметим, что чем больше сегментов получено в результате разбиения, тем меньше величина связности. Минимальная величина связности, при которой полученные сегменты являются оптимальными, имеет значение  $Q = 0,3$ .

*Шаг 4.* Получение сегментов с минимальной или заданной величиной связности.

Для получения сегментов выполняется объединение соседних сегментов с наименьшими весами между собой. Данную операцию целесообразно производить до тех пор, пока величина связности получаемых сегментов не будет больше или равна минимальному или заданному

значению данной величины.

*Шаг 5.* Определить, требуется ли сформировать таблицы маршрутизации в корпоративной сети:

- а) если да – перейти к шагу 7;
- б) иначе – к шагу 6.

*Шаг 6.* Анализируя полученную используемым протоколом маршрутизации информацию, определить, произошло ли изменение маршрутной метрики каналов связи в сформированных подсетях или между ними в корпоративной сети:

- а) если да – перейти к шагу 7;
- б) иначе – к шагу 13.

*Шаг 7.* В корпоративной сети определить узел-источник и узел-получатель пакетов сетевого трафика.

*Шаг 8.* Присвоение начальных значений. Положить  $d^*(v_b) = s$  и считать эту пометку постоянной. Принять  $d(v_j) = \infty$  для всех  $v_j \neq v_b$  и считать эти пометки временными. Присвоить  $t = b$ .

*Шаг 9.* Обновление пометок. Для всех узлов связи  $v_j \in G_i(v_t)$ , инцидентных узлу  $v_t$  и имеющих временные пометки, изменить пометки в соответствии с выражением:

$$d_l = \min_{j=1..N_i} (d_j, \min_{t=1..N_i} \{d_t\} + w_{t,j}),$$

где  $d_l$  – пометка узла связи с номером  $l$  ( $l = 1 .. N_i$ );  $d_t$  – пометка текущего узла связи,  $d_j$  – пометка узла связи, смежного с текущим узлом,  $w_{t,j}$  – вес канала, соединяющего узел  $v_t$  с узлом  $v_j$ .

*Шаг 10.* Превращение пометки в постоянную. Среди всех узлов с временными пометками найти такой узел, для которого

$$d_l^* = \min_{j=1..N_i} \{d_j\},$$

где  $d_l^*$  – постоянная пометка узла связи с номером  $l$  ( $l = 1 .. N_i$ ).

*Шаг 11.* Считать пометку узла  $v_j^*$  постоянной и положить  $v_t = v_j^*$ .

*Шаг 12.* Если все узлы связи имеют постоянные пометки, то перейти к шагу 13. Иначе перейти к шагу 9.

Сам маршрут можно найти, применяя рекурсивно процедуру, в которой реализовано выражение:

$$d_l^* = \min_{j=1..N_i} \{d_j^*\} + w_{j,k},$$



где  $w_{j,k}$  – вес канала, соединяющего узел  $v_j$  с узлом  $v_k$ .

Оно справедливо для последующих и предыдущих узлов, принадлежащих одному маршруту.

*Шаг 13.* а) передать пакеты по доступному оптимальному маршруту;

б) установить флаг передачи.

*Шаг 14.* Переформировать таблицы маршрутизации для каждой изменившейся подсети и список каналов связи между подсетями.

*Шаг 15.* Перейти к шагу 6.

**Пример работы алгоритма.** Представим корпоративную сеть в виде неориентированного взвешенного связанного графа  $G = (V, E, W)$ , показанного на рисунке 1. Далее перейдем к нахождению минимального покрывающего дерева корпоративной сети  $G$ . В качестве начального узла связи выберем узел  $V_1$ . Результат нахождения минимального покрывающего дерева приведен на рисунке 2.

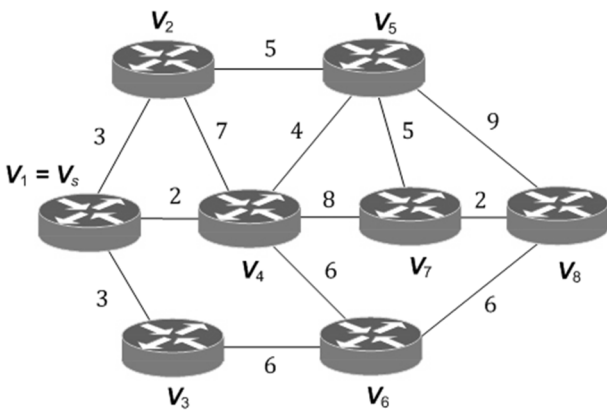


Рисунок 1 – Граф  $G$  корпоративной сети

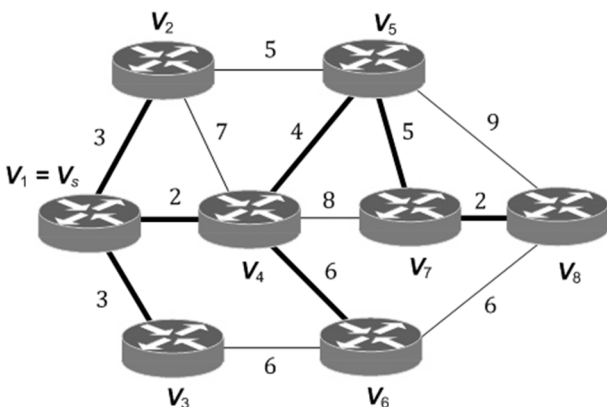


Рисунок 2 – Результат построения минимального покрывающего дерева  $G_1$  для сети  $G$

На основе полученной структуры произведем выделение сегментов  $G_i \subset G_1$  в соответствии с алгоритмом из [12]. Конечный результат сегментации корпоративной сети  $G$  на подсети  $G_2$  и  $G_3$  приведен на рисунке 3.

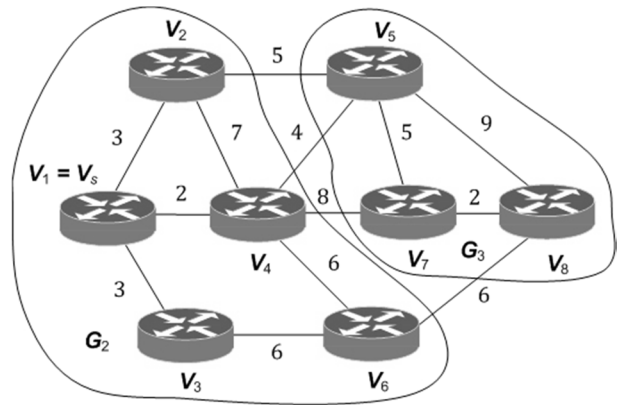


Рисунок 3 – Результат сегментации корпоративной сети  $G$  на подсети  $G_2$  и  $G_3$

В каждой из полученных подсетей  $G_2$  и  $G_3$  выберем узел-источник пакетов сетевого трафика. Например, для подсети  $G_2$  – узел связи  $V_1$ , а для подсети  $G_3$  – узел связи  $V_8$ . Выполним расчет оптимальных маршрутов для каждой подсети  $G_2$  и  $G_3$  по отдельности от узла-источника до каждого узла связи в подсети.

Для подсети  $G_2$  список оптимальных маршрутов от начального узла связи  $V_1$  имеет следующий вид.

**Узел связи  $V_2$**

Оптимальный маршрут  $\pi_2 = (e_{1,2})$  с оценкой  $d_2 = 3$ .

**Узел связи  $V_3$**

Оптимальный маршрут  $\pi_3 = (e_{1,3})$  с оценкой  $d_3 = 3$ .

**Узел связи  $V_4$**

Оптимальный маршрут  $\pi_4 = (e_{1,4})$  с оценкой  $d_4 = 2$ .

**Узел связи  $V_6$**

Оптимальный маршрут  $\pi_6 = (e_{1,4}; e_{4,6})$  с оценкой  $d_6 = 2 + 6 = 8$ .

Для подсети  $G_3$  список оптимальных маршрутов от начального узла связи  $V_8$  имеет следующий вид.

**Узел связи  $V_7$**

Оптимальный маршрут  $\pi_7 = (e_{8,7})$  с оценкой  $d_7 = 2$ .

**Узел связи  $V_5$**

Оптимальный маршрут  $\pi_5 = (e_{8,7}; e_{7,5})$  с оценкой  $d_5 = 2 + 5 = 7$ .

Обмен информацией между подсетями осуществляется через пограничные маршрутизаторы, каналы связи которых одновременно подключены к разным подсетям. Поэтому для обмена информацией между сформированными подсетями в соответствии с предложенным алгоритмом выберем канал связи с минимальной нагрузкой. В приведенном примере – канал связи  $e_{4,5}$  с весом  $w_{4,5} = 4$ . Сформированные деревья оптимальных маршрутов для каждой подсети  $G_2$

и  $G_3$  приведены на рисунке 4.

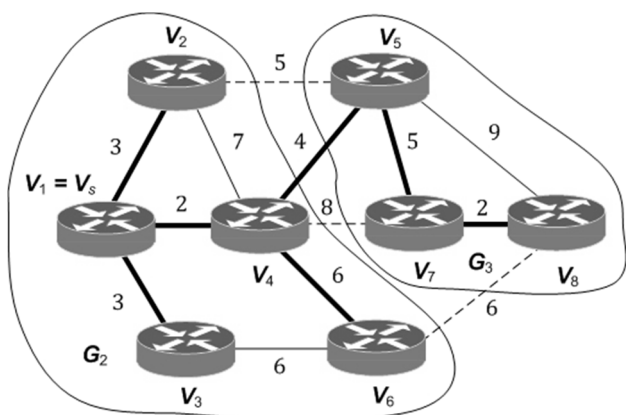


Рисунок 4 – Деревья оптимальных маршрутов для подсетей  $G_2$  и  $G_3$  и между ними

Рассмотрим возможные случаи динамического изменения нагрузки на линиях связи в подсетях  $G_2$  и  $G_3$  корпоративной сети и оценим трудоемкость построения оптимальных маршрутов в соответствии с предложенным алгоритмом.

Пусть увеличилась нагрузка канала связи  $e_{1,2}$  в подсети  $G_2$ , так что  $nw_{1,2} = 10 > w_{1,2} = 3$ . В этом случае дерево оптимальных маршрутов для подсети  $G_2$  будет иметь вид, представленный на рисунке 5. Оценка трудоемкости изменения дерева оптимальных маршрутов составляет величину  $O(N_1^2 / p)$ ,  $N_1$  – число узлов связи в подсети  $G_2$ ,  $p$  – число подсетей в сети  $G$ . В приведенном примере  $N_1 = 5$ ,  $p = 2$ . Таким образом, при увеличении нагрузки канала связи  $e_{1,2}$  в подсети  $G_2$  в соответствии с предложенным алгоритмом трудоемкость расчета дерева оптимальных маршрутов составит 13 элементарных операций. Для сравнения: в классическом алгоритме Дейкстры для изменения дерева оптимальных маршрутов без использования метода маршрутизации по подсетям трудоемкость составит 64 элементарные операции, что позволяет сказать об эффективности предложенного алгоритма.

Пусть уменьшилась нагрузка канала связи  $e_{8,5}$  в подсети  $G_3$ , так что  $nw_{8,5} = 6 < w_{8,5} = 9$ . В этом случае дерево оптимальных маршрутов для подсети  $G_3$  будет иметь вид, представленный на рисунке 6. Оценка трудоемкости изменения дерева оптимальных маршрутов составляет величину  $O(N_2^2 / p)$ ,  $N_2$  – число узлов связи в подсети  $G_3$ ,  $p$  – число подсетей в сети  $G$ . В приведенном примере  $N_2 = 3$ ,  $p = 2$ . Таким образом, при уменьшении нагрузки канала связи  $e_{8,5}$  в подсети  $G_3$  в соответствии с предложенным алгоритмом трудоемкость расчета дерева оптимальных маршрутов составит 5 элементарных операций. Для сравнения: в классическом алгоритме Дейкстры для изменения дерева опти-

мальных маршрутов без использования метода маршрутизации по подсетям трудоемкость составит 64 элементарные операции, что позволяет сказать об эффективности предложенного алгоритма.

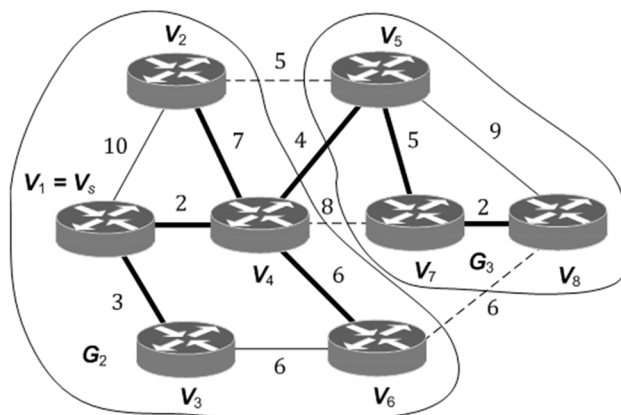


Рисунок 5 – Увеличение нагрузки канала связи  $nw_{1,2} = 10 > w_{1,2} = 3$  в подсети  $G_2$

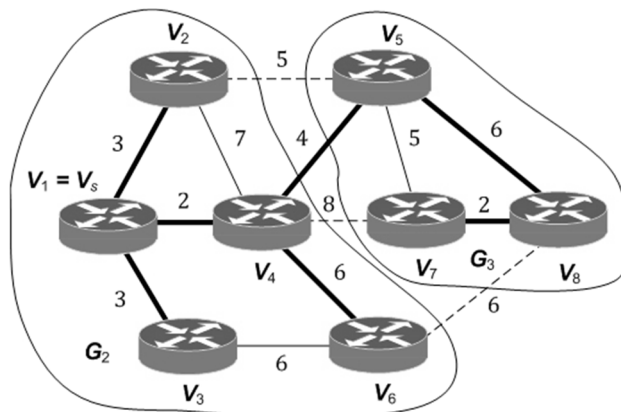


Рисунок 6 – Уменьшение нагрузки канала связи  $nw_{8,5} = 6 < w_{8,5} = 9$  в подсети  $G_3$

Пусть увеличилась нагрузка канала связи  $e_{4,5}$  между подсетями  $G_2$  и  $G_3$ , так что  $nw_{4,5} = 6 > w_{4,5} = 4$ . В этом случае дерево оптимальных маршрутов между подсетями  $G_2$  и  $G_3$  будет иметь вид, представленный на рисунке 7. Оценка трудоемкости изменения списка оптимальных маршрутов между подсетями  $G_2$  и  $G_3$  составляет величину  $O(M_4)$ , где  $M_4$  – число каналов связи между подсетями  $G_2$  и  $G_3$ . Таким образом, при увеличении нагрузки канала связи  $e_{4,7}$  между подсетями  $G_2$  и  $G_3$  в соответствии с предложенным алгоритмом необходимо выполнить четыре элементарные операции для расчета списка оптимальных маршрутов между подсетями. Для сравнения: в классическом алгоритме Дейкстры для изменения дерева оптимальных маршрутов без использования метода маршрутизации по подсетям трудоемкость составит 64 элементарные операции, что позволяет сказать об эффективности предложенного алгоритма.

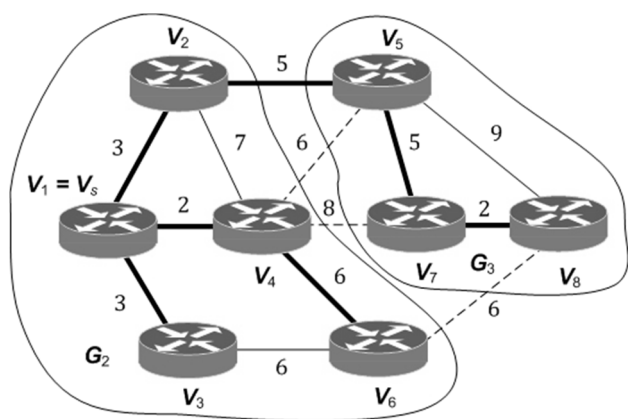


Рисунок 7 – Увеличение нагрузки канала связи  $nw_{4,5} = 6 > w_{4,5} = 4$  между подсетями  $G_2$  и  $G_3$

**Результаты моделирования.** Для подтверждения правильности предложенного алгоритма динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям разработано программное обеспечение моделирования процессов маршрутизации.

При разработке основное внимание уделялось корректности предлагаемого алгоритма и размерности решаемой задачи. Были проведены исследования графовых моделей корпоративных сетей, состоящих из 10, 50 и 100 узлов связи. Исследование разработанного алгоритма динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям показало, что трудоемкость построения оптимальных маршрутов передачи данных значительно меньше, чем у алгоритма Дейкстры и составляет величину порядка  $O(N_i^2 / p)$ ,  $N_i$  – число узлов связи в подсети с номером  $i$ ,  $p$  – число подсетей в корпоративной сети. На основе этого можно сделать вывод, что предложенный алгоритм является эффективным при поиске оптимальных маршрутов в условиях динамических изменений нагрузки на линиях связи.

В дальнейшем предполагается усовершенствование предложенного алгоритма с целью его применения для решения задач адаптивной ускоренной маршрутизации [13-23], многопутевой адаптивной маршрутизации и балансировки трафика [24, 25] в корпоративных сетях одного провайдера связи, а также адаптивной маршрутизации в корпоративных сетях нескольких провайдеров связи [26, 27].

**Заключение.** Предложенный в работе алгоритм динамического формирования трафика корпоративных сетей на основе метода маршрутизации по подсетям позволяет уменьшить трудоемкость расчета оптимальных маршрутов в условиях динамических изменений нагрузки на линиях связи и повысить эффективность процессов маршрутизации за счет высокой производи-

тельности при распределении трафика между подсетями.

*Работа выполнена при финансовой поддержке гранта Президента РФ для молодых ученых - кандидатов наук МК-819.2014.9.*

#### Библиографический список

1. Вишневецкий В.М. Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003. 512 с.
2. Корячко В.П., Перепелкин Д.А. Корпоративные сети: технологии, протоколы, алгоритмы. М.: Горячая линия – Телеком, 2011. 219 с.
3. Олифер В.Г., Олифер Н.А. Основы компьютерных сетей. СПб.: Питер, 2009. 352 с.
4. Корячко В.П., Перепелкин Д.А. Анализ и проектирование маршрутов передачи данных в корпоративных сетях. М.: Горячая линия – Телеком, 2012. 235 с.
5. Лемешко А.В., Вавенко Т.В. Усовершенствование потоковой модели многопутевой маршрутизации на основе балансировки нагрузки // Проблемы телекоммуникаций. 2012. № 1 (6). С. 12-29.
6. Лемешко А.В., Вавенко Т.В. Разработка и исследование потоковой модели адаптивной маршрутизации в программно-конфигурируемых сетях с балансировкой нагрузки // Доклады Томского государственного университета систем управления и радиоэлектроники. 2013. № 3 (29). С. 100-108.
7. Ижванов Ю.Л. Динамическая оценка состояния компьютерных сетей на основе метода сетевой томографии и задачи балансировки трафика // Информатизация образования и науки. 2013. № 3 (19). С. 35-40.
8. Шибанов А.П., Корячко В.П., Ижванов Ю.Л. Моделирование агрегированного телекоммуникационного канала с технологией открытых потоков // Радиотехника. 2012. № 3. С. 109-112.
9. Ижванов Ю.Л., Корячко В.П., Шибанов А.П., Сапрыкин А.Н., Лукьянов О.В. Оптимизация сетей с дозированной балансировкой нагрузки и пиринговыми каналами // Вестник Рязанского государственного радиотехнического университета. 2013. № 1 (43). С. 67-74.
10. Никульчев Е.В., Паяин С.В., Плужник Е.В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 3 (45). С. 54-57.
11. Корячко В.П., Лукьянов О.В., Шибанов А.П. Нахождение скрытого параллелизма протоколов для улучшения характеристик сети передачи данных полигонного измерительного комплекса // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 68-75.
12. Перепелкин Д.А. Алгоритм формирования оптимальных структур сегментов корпоративных сетей с учетом данных о структуре базовой сети и связности каналов связи // Вестник Рязанского государственного радиотехнического университета. 2014. № 50-1. С. 59-64.

13. *Перепелкин Д.А., Перепелкин А.И.* Разработка алгоритмов адаптивной маршрутизации в корпоративных вычислительных сетях // Вестник Рязанского государственного радиотехнического университета. 2006. № 19. С. 114-116.

14. *Перепелкин А.И., Перепелкин Д.А.* Разработка алгоритма динамической маршрутизации на базе протокола OSPF в корпоративных вычислительных сетях // Вестник Рязанского государственного радиотехнического университета. 2009. № 28. С. 68-72.

15. *Перепелкин Д.А.* Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом добавлении элементов корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2010. № 34. С. 65-71.

16. *Перепелкин Д.А.* Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом отказе элементов корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2011. № 37. С. 53-58.

17. *Корячко В.П., Перепелкин Д.А., Перепелкин А.И.* Алгоритм парных перестановок маршрутов в корпоративных сетях // Системы управления и информационные технологии. 2010. Т. 40. № 2. С. 51-56.

18. *Корячко В.П., Перепелкин Д.А., Перепелкин А.И.* Повышение эффективности функционирования корпоративных сетей при динамических изменениях в их структуре и нагрузках на линии связи // Вестник Рязанского государственного радиотехнического университета. 2010. № 33. С. 49-55.

19. *Перепелкин Д.А., Перепелкин А.И.* Повышение качества функционирования корпоративных сетей на базе протокола OSPF // Качество. Инновации. Образование. 2010. № 12. С. 51-56.

20. *Перепелкин Д.А., Перепелкин А.И.* Алгоритм адаптивной ускоренной маршрутизации в условиях динамически изменяющихся нагрузок на линиях связи в корпоративной сети // Информационные технологии. 2011. № 3. С. 2-7.

21. *Перепелкин Д.А.* Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом подключении узлов и линий связи корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2013. № 4-1 (46). С. 67-75.

22. *Перепелкин Д.А.* Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом отказе узлов и линий связи корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 84-91.

23. *Перепелкин Д.А.* Динамическое формирование структуры и параметров линий связи корпоративной сети на основе данных о парных перестановках маршрутов // Информационные технологии. 2014. № 4. С. 52-60.

24. *Корячко В.П., Перепелкин Д.А.* Разработка и исследование математической модели многопутевой адаптивной маршрутизации в сетях связи с балансировкой нагрузки // Электросвязь. 2014. № 12. С. 27-31.

25. *Перепелкин Д.А.* Математическая модель многопутевой адаптивной маршрутизации с балансировкой неоднородной нагрузки в условиях динамических подключений узлов и линий связи в телекоммуникационных сетях // Радиотехника. 2015. № 5. С. 34-40.

26. *Корячко В.П., Перепелкин Д.А., Иванчикова М.А.* Алгоритм адаптивной маршрутизации в корпоративных сетях нескольких провайдеров связи // Вестник Рязанского государственного радиотехнического университета. 2013. № 2 (44). С. 52-56.

27. *Корячко В.П., Перепелкин Д.А., Иванчикова М.А.* Алгоритм парных переходов каналов связи при динамическом изменении нагрузки в корпоративных сетях нескольких провайдеров связи с различными зонами покрытия // Вестник Рязанского государственного радиотехнического университета. 2014. № 48. С. 68-76.

УДК 681.324

*Т.А. Золотарева*

## **АЛГОРИТМИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ИДЕНТИФИКАЦИИ СОСТОЯНИЯ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ**

*Предложены структурные схемы и алгоритмическое описание сигнатурного, статистического, нейросетевого анализаторов с использованием нейросети Хемминга, позволяющие в комплексном применении реализовать систему поддержки принятия решения для идентификации локальной сети.*

**Ключевые слова:** локальная вычислительная сеть, анализатор трафика, интеллектуальная система, искусственный интеллект.

**Введение.** Телекоммуникационные и сетевые технологии являются в настоящее время одной из движущих сил научно-технического прогресса.

Современные информационные технологии активно внедряются во все сферы деятельности человека. Появление локальных и глобальных сетей передачи данных предоставило пользователям компьютеров оригинальные возможности оперативного обмена информацией.

Многочисленные исследования особенностей трафика компьютерных локальных вычислительных сетей (ЛВС) говорят о том, что он имеет особую структуру, сохраняющуюся при различных измерениях. В процессе передачи информации возникают большие модификации при среднем уровне трафика в небольших границах. Подобное явление, значительно влияющее на потери и ошибки пакетов, не учитывается в классической теории электросвязи. Расчет параметров системы распределения информации, предназначенной для обработки сетевого трафика, по классическим формулам часто дает некорректные результаты. *Цель работы* – реализовать интеллектуальную систему поддержки принятия решений (СППР) на основе комплексного подхода (гибридных технологий) к проблеме идентификации состояния ЛВС, включающего эксплуатацию методов сигнатурного, статистического анализа сетевого трафика и искусственного интеллекта для качественного детектирования и идентификации сетевых проблем [1].

**Постановка задачи.** Ситуация, возникшая в современных ЛВС, наличие большого числа сетевых маршрутов, на которых периодически возникают значительные модификации в передаче данных и большие потери пакетов, появление оригинальных свойств сетевого трафика, необходимость обеспечения высокого качества сетевых служб делают актуальным анализ сетевого трафика на основе комплексного подхода.

Главными задачами, которые решаются при управлении компьютерными сетями (КС), являются мониторинг производительности и безопасности, идентификация проблем, контроль и изменение конфигурации ЛВС. Эти задачи выражают главный алгоритм действий администраторов ЛВС.

На первом уровне – уровне мониторинга, администратор обнаруживает отклонения в работе ЛВС. На следующем уровне – уровне идентификации, администратор пытается установить причину этих проблем. На самом высоком уровне, установив источник проблем, администратор принимает решение и вырабатывает последовательность действий по восстановлению нормального режима функционирования

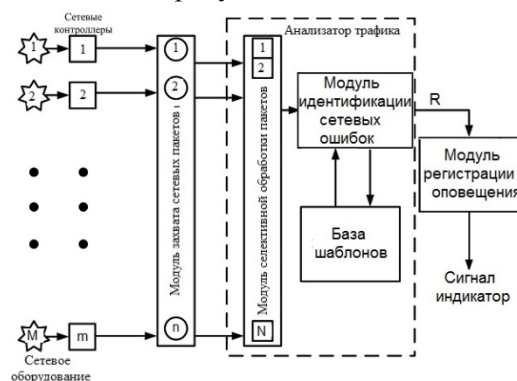
всех сетевых служб.

Поиск эффективных методов идентификации проблемных состояний в работе КС, являющихся следствием технических или технологических сбоев, отказов и простых ошибок, в настоящее время является актуальной научно-технической задачей.

Все методы идентификации сетевых проблем (СП) можно разделить на три группы: сигнатурный анализ, статистический анализ, интеллектуальный анализ. Идентификация работы ЛВС может проводиться в реальном и в отложенном масштабах времени.

**Экспериментальные исследования.** Анализ сигнатур оказался самым первым методом, который использовался для обнаружения проблем в работе КС, связанных с деятельностью злоумышленников. Он базируется на простом понятии совпадения последовательности с эталонным образцом. Во входящем пакете просматривается байт за байтом и сравнивается с сигнатурой – характерной строкой программы, указывающей на наличие вредного трафика.

Структурная схема анализатора трафика представлена на рисунке 1.



**Рисунок 1 – Принципиальная схема анализатора трафика**

Процедура анализа включает два этапа:

- селективный сбор фрагментов пакетов;
- распознавание ошибок по образцам.

Обозначим сетевой трафик как поток пакетов

в виде множества  $X = \{x_i\}_{i=1}^n$ , где  $n$  – общее число пакетов. Основу образцов представим в виде множества  $A$ , объединяющего группы ошибок  $A_j$ ,  $j = \overline{1, g}$ ,

$$A = A_1 \cup A_2 \cup \dots \cup A_g = \bigcup_{j=1}^g A_j,$$

где  $g$  – число групп;  $A_j$  –  $j$ -й кластер, являющийся множеством однотипных ошибок,  $A_j = \{a_{jk}\}_{k=1}^K$ ;  $K$  – общее число ошибок в  $j$ -м кластере.

Ошибка считается найденной, если выпол-

няется условие:  $X \subseteq A$ . На вход модуля регистрации подается сигнал  $R$ , который может принимать два значения ("0" – есть совпадение, "1" – нет совпадения).

В процессе сигнатурного анализа (рисунок 2) данные проходят несколько стадий обработки: декодирование пакетов (идентифицируется протокол, содержащийся в анализируемом пакете, проверяется соответствие данных этому протоколу), препроцессорная обработка (дополнительная обработка пакета данных), процессор обнаружения проблем, модули вывода (происходит регистрация проблемных сигналов).

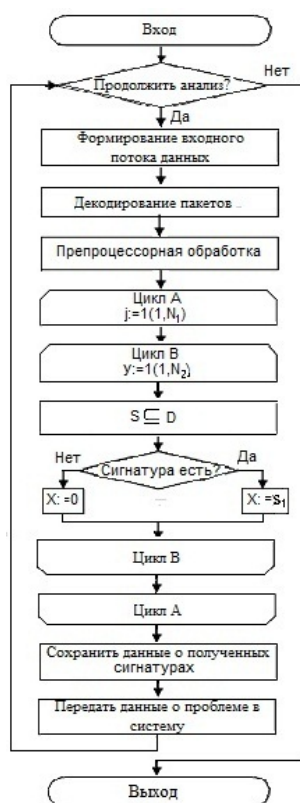


Рисунок 2 – Алгоритмическая реализация блока сигнатурного анализа

При применении *статистических методов* в качестве объектов исследования берутся отдельные сетевые устройства, характеристики трафика, которые являются случайными величинами, изменяющимися во времени и определяющими работу сетевых устройств по протоколам IP, UDP, ICMP, TCP.

Из статистических методов идентификации СП была выбрана модель на основе анализа среднего значения и среднеквадратичного отклонения параметров сетевого трафика. Выбор этого метода объясняется тем, что он наиболее легко реализуем на практике и основан на том обстоятельстве, что в течение определенного интервала времени (на протяжении суток, часов, минут) могут изменяться некоторые статис-

тические характеристики потока пакетов. В этом случае методы обнаружения нарушений базируются на сравнении текущих характеристик потока пакетов с усредненными характеристиками за некоторый промежуток времени. Первые характеристики называются локальными, а вторые – глобальными. Если текущие характеристики сильно отличаются от глобальных характеристик, то делается вывод о проблемном поведении потока пакетов и вполне возможны сбои и отказы в работе сетевого оборудования или программного обеспечения.

Обобщенная схема статистического анализатора, реализующего данный метод идентификации, представлена на рисунке 3.

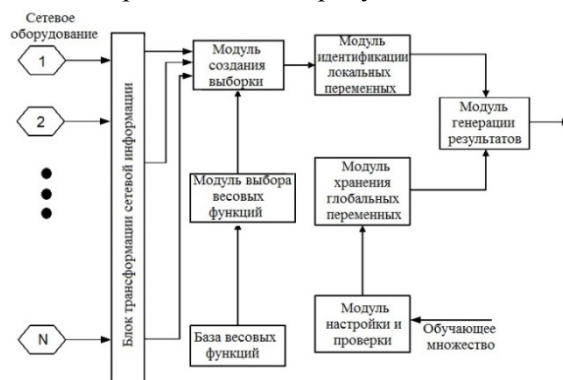


Рисунок 3 – Обобщенная схема статистического анализатора сетевой информации

Алгоритм работы статистического анализатора представлен на рисунке 4.

При поступлении информации в блок создания входных данных по заданной весовой функции  $F(z)$  определяется размер выборки и для нее осуществляется определение локальных статистических характеристик потока пакетов сетевого трафика: выборочного среднего  $\varepsilon$ , выборочной дисперсии  $\Delta^2$  и статистики  $\lambda^2$ .

После этого осуществляется сравнение определенных статистик с глобальными статистиками, полученными на этапе обучения и настройки системы.

*Экспертные информационные системы* (ЭИС) представляют наиболее известную модель для обнаружения СП. ЭИС не используют мониторинг пороговой величины для определения СП в случае его превышения. Они используют подход на основе правил, составляющих основу знаний. Накапливаемая основа знаний создается экспертами в области сетевого управления.

Из всего диапазона существующих моделей представления знаний (логические, фреймовые, сетевые, продукционные и другие модели) наибольшей гибкостью обладают нечеткие модели.

Для формализации знаний в настоящее

время успешно применяются аппараты теории нечетких множеств (ТНМ) и нечеткой логики (НЛ), которые формируются в виде нечетких и лингвистических переменных, а нечеткость отдельных операций в общем процессе принятия решения синтезируется в виде нечетких алгоритмов.

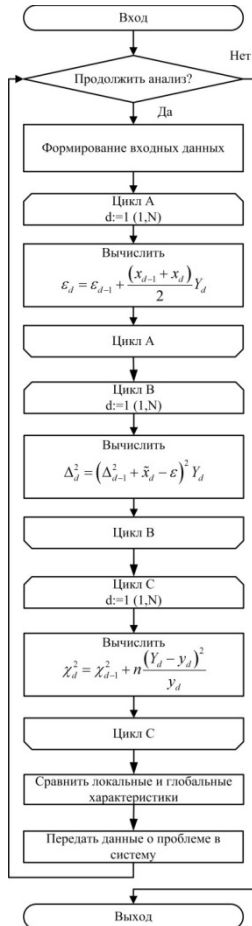


Рисунок 4 – Алгоритм функционирования статистического анализа

Наиболее распространенная ИНС применительно для задачи анализа сетевого трафика (образов типовых сетевых пакетов) является ИНС Хемминга.

ИНС Хемминга предназначена для распознавания класса принадлежности объекта, заданного вектором  $X$  биполярных признаков (возможные значения признаков +1 и -1) размерности  $N$ . Предполагается, что имеются  $M$  классов, каждый из которых характеризуется своим эталонным представителем – объектом  $X_v, v = 1, 2, \dots, V$  [2,3].

Эталонные образы и соответствующие векторы признаков хранятся в основе данных. Они отобраны экспертами для разных типов образов. На рисунке 5 указана схема обработки данных при применении нейросетевого классификатора Хемминга.



Рисунок 5 – Схема обработки данных в нейросетевом анализаторе Хемминга

ИНС Хемминга принимает на  $N$  входов биполярные признаки объекта и после обработки данных активизирует один из  $K$  выходов, который указывает на класс принадлежности предъявленного на входе объекта [4,6].

Для описания алгоритма нейросетевого анализатора сетевого трафика применим следующую обобщенную схему (рисунок 6).



Рисунок 6 – Обобщенный алгоритм работы нейросетевого анализатора

Эксплуатация подготовленной ИНС будет происходить по следующему алгоритму (рисунок 7) [5].

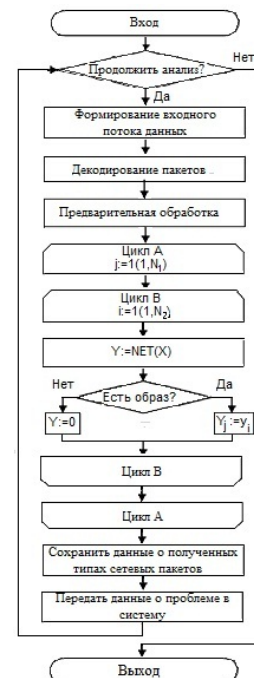
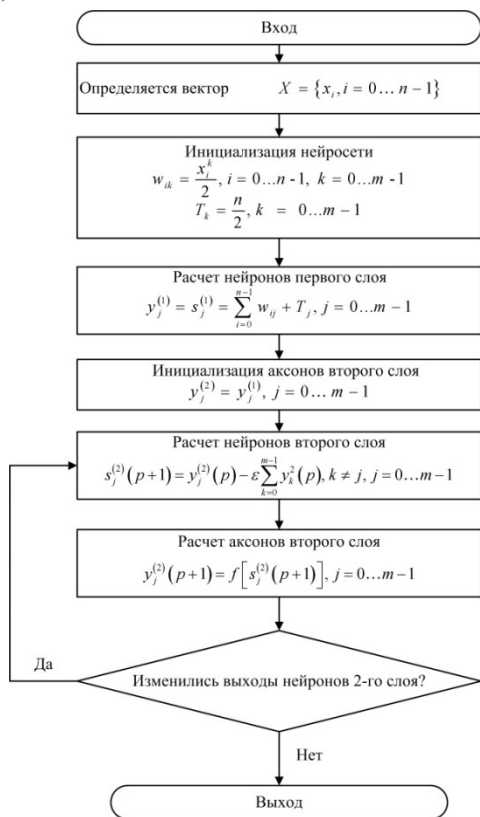


Рисунок 7 – Алгоритм работы нейросетевого анализатора в режиме простой эксплуатации

Более подробно представим особенности функционирования нейросети Хемминга (рисунок 8).



**Рисунок 8 – Подробный алгоритм функционирования нейросети Хемминга**

Описание алгоритма гибридного анализатора сетевого трафика представлено на рисунке 9.



**Рисунок 9 – Алгоритм гибридного анализатора сетевого трафика**

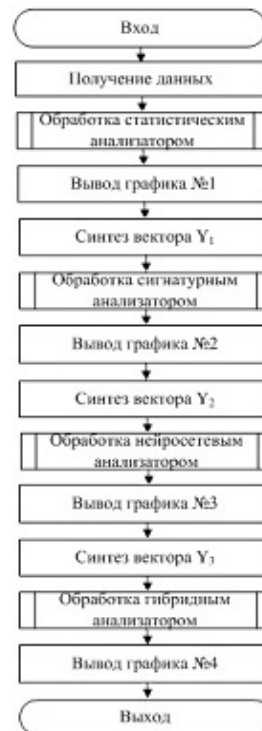
Основа нечеткой логики на представленной схеме представляет собой модифицируемую систему определенных правил.

Для наглядности представим структуру программного продукта в виде менеджера анализаторов, который включает в себя статистический, сигнатурный, нейросетевой и гибридный анализаторы.

С помощью менеджера анализаторов рядо-

вой эксперт может управлять слаженной работой нескольких анализаторов в анализирующем программном модуле (АПМ).

Алгоритм работы АПМ представлен на рисунке 10.



**Рисунок 10 – Алгоритм работы АПМ**

Применив АПМ, эксперт существенно оптимизирует свою работу. При этом предполагается, что алгоритм действий эксперта имеет примерно следующий вид (рисунок 11).



**Рисунок 11 – Алгоритм работы эксперта**

**Заключение.** В статье предложен алгоритм создания современной системы идентификации СП, основанный на комбинированном (гибридном) подходе, который включает в себя хорошо зарекомендовавшие статистические и сигнатур-



ные анализаторы, а также интеллектуальные технологии сетевого анализа.

### **Библиографический список**

1. Таненбаум Э. Компьютерные сети. // пер. с англ.; под ред. А. Гребенькова. – СПб.: Питер, 2012. – 960.
2. Мерков А.Б. Распознавание образов. Введение в методы статистического обучения. – М: URSS, 2011. – 256.
3. Рапопорт Г.Н. Биологический и искусственный разум. Часть 1. Сознание, мышление и эмо-

ции. – М.: Книжный дом "Либроком", 2011. – 184.

4. Хайкин С. Нейронные сети: полный курс. М.: Вильямс, 2006. – 1104.

5. Zolotareva T.A., Ziyautdinov V.S., Malysh V.N. International conference on computer technologies in physical and engineering applications – СПб., 2014. – 221.

6. Макаров И.М., Лохин В.М., Манько С.В., Романов М.П. Искусственный интеллект и интеллектуальные системы управления. – М.: Наука, 2006. – 336.

УДК 621.395.7, 519.854

**Е.С. Скаков, В.Н. Малыш**

## **ЭВОЛЮЦИОННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ БАЗОВЫХ СТАНЦИЙ**

*Предложена модификация эволюционного алгоритма для решения задачи размещения базовых станций при создании беспроводной сети передачи данных. Путем компьютерного моделирования произведено сравнение быстроедействия и точности решений, получаемых предложенным методом и методом полного перебора. Установлено влияние входных параметров задачи на скорость нахождения решения, а также определена лучшая разновидность эволюционного алгоритма применительно к решению поставленной задачи.*

**Ключевые слова:** размещение базовых станций, эволюционный алгоритм, беспроводные сети, оптимизация, SIR.

**Введение.** Важным этапом при проектировании беспроводной сети передачи данных является синтез ее топологической структуры. Он подразумевает планирование территориального размещения базовых передаточных станций и подключение к ним клиентов.

Как известно [1], задача проектирования сети (в т.ч. и беспроводной) может быть сведена к отысканию минимума функционала приведенной стоимости при наличии ограничений на вероятностно-временные и структурные характеристики сети и требований принадлежности множества вариантов архитектуры сети к области технически реализуемых решений.

Среди русскоязычных публикаций по проблеме размещения базовых станций (БС) отметим работу [2], посвященную сетям WiMAX. Подробная классификация алгоритмов размещения базовых станций для сетей UMTS 3G дана в статье [3].

Среди недостатков работ, посвященных рассматриваемой проблеме, отметим следующие:

– решение задачи размещения БС методами, не показывающими высокую скорость расчета

(метод ветвей и границ, эвристический метод Лагранжа и др.);

– отсутствие ограничений, учитывающих уровень затухания сигнала при распространении от БС к клиенту и обратно;

– в большинстве работ не учитывается уровень межсотовых помех;

– большинство моделей не подразумевают использование нескольких типов базовых станций.

Настоящая работа направлена на создание модели, не имеющей вышеперечисленных недостатков. В ней NP-трудная задача размещения БС решается с применением метаэвристического метода оптимизации, а именно эволюционного алгоритма.

**Цель работы** – сформулировать математическую модель задачи оптимального размещения базовых станций; предложить модификацию эволюционного алгоритма для решения данной задачи; исследовать эффективность предложенного алгоритма.

**Постановка задачи.** Задача размещения базовых станций заключается в том, что у нас есть

$N_{tp}$  клиентов, каждого из которых необходимо подключить к базовой станции. Базовая станция может быть установлена на одном из  $N_{ps}$  мест-кандидатов. Имеется  $N_{types}$  типов базовых станций, отличающихся по своим характеристикам. Задача сводится к минимизации общей стоимости установленных базовых станций при выполнении ряда ограничений.

Решение задачи будем представлять в виде пары векторов (одномерных массивов) целых чисел –  $Y$  и  $X$  (пояснение: здесь и в дальнейшем запись вида  $A[j]$  означает обращение к  $j$ -му элементу вектора  $A$ ; элементы массивов нумеруются, начиная с 1):

$Y$  – вектор, содержащий  $N_{ps}$  элементов. Элементы могут принимать целые значения из диапазона  $[0; N_{types}]$ . Если  $Y[i] = 0$ , то на  $i$ -м месте-кандидате не установлена базовая станция. Если  $Y[i] = w$ , то на  $i$ -м месте-кандидате установлена базовая станция  $w$ -го типа;

$X$  – вектор, содержащий  $N_{tp}$  элементов. Элементы могут принимать целые значения из диапазона  $[1; N_{ps}]$ . Если  $X[i] = w$ , то это значит, что  $i$ -й клиент подключен к базовой станции, установленной на  $w$ -м месте-кандидате.

Клиент может быть подключен к месту-кандидату, только если на нем уже установлена базовая станция:

$$Y[X[i]] \neq 0 \quad \forall i \in \{1, 2, \dots, N_{tp}\}. \quad (1)$$

Пусть  $b$  – вектор, элементами которого являются требуемые полосы пропускания для клиентов;  $\beta$  – вектор, элементами которого являются максимальные производительности базовых станций разного типа. Для каждой установленной БС суммарный требуемый трафик не должен превосходить максимально возможную производительность оборудования:

$$\sum_{i=1}^{N_{tp}} p_{is} \cdot b[i] \leq \beta[Y[i]] \quad \forall s \in \{1, 2, \dots, N_{ps}\}, \quad (2)$$

$$p_{is} = \begin{cases} 1, & \text{если } X[i] = s \\ 0 & \text{иначе} \end{cases}.$$

Пусть  $P_{BS}^{\max}$  – вектор, элементами которого являются максимальные мощности базовых станций разного типа;  $P_{TP}^{\max}$  – вектор, элементами которого являются максимальные мощности клиентов;  $P_{BS}^{tar}$  – вектор, элементами которого являются чувствительности базовых станций

разного типа;  $P_{TP}^{tar}$  – вектор, элементами которого являются чувствительности клиентов;  $G$  – двумерный массив (матрица) размерностью  $N_{tp} \times N_{ps}$ , каждый элемент которого  $0 \leq G[i][s] \leq 1$  отражает уровень затухания между клиентом  $i$  и местом установки  $s$ .

Несмотря на затухание сигнала на пути от базовой станции к клиенту, мощность, доходящая от приёмника к передатчику, должна превышать минимальную целевую мощность:

$$\frac{G[i][X[i]] \cdot P_{BS}^{\max}[Y[X[i]]]}{P_{TP}^{tar}[i]} \geq 1 \quad (3)$$

$$\forall i \in \{1, 2, \dots, N_{tp}\}$$

Ограничение (3) составлено для режима downlink. Аналогичное ограничение для режима uplink (несмотря на затухание сигнала на пути от клиента к БС, мощность, доходящая от клиента к БС, должна превышать минимальную целевую мощность) выглядит следующим образом:

$$\frac{G[i][X[i]] \cdot P_{TP}^{\max}[i]}{P_{BS}^{tar}[Y[X[i]]]} \geq 1 \quad (4)$$

$$\forall i \in \{1, 2, \dots, N_{tp}\}$$

Согласно источникам [1] и [3], целевая функция, которую необходимо минимизировать, должна иметь вид

$$\Phi = \sum_{s=1}^{N_{ps}} Cost[Y[s]], \quad (5)$$

где  $Cost$  – вектор стоимостей (включая установку) базовых станций разного типа.

Однако такое представление обладает очевидным недостатком. Задача размещения БС может иметь несколько допустимых решений с одинаковым значением общей стоимости оборудования. В подобном случае непонятно, какое из них считать лучшим. В данной работе предлагается целевую функцию рассчитывать следующим образом:

$$\Phi = \left( \sum_{s=1}^{N_{ps}} Cost[Y[s]] + \sum_{i=1}^{N_{tp}} SIR_{dB}^i \cdot k \right). \quad (6)$$

Первое слагаемое – суммарная стоимость комплекса. Второе слагаемое отвечает за учет уровня  $SIR$  для всех клиентов системы.  $SIR$  – отношение уровня сигнала к уровню помех (signal-to-interference ratio) [4]. В общем случае рассчитывается так:

$$SIR_{dB} = 10 \log_{10} \left( \frac{P_{signal}}{P_{interference}} \right). \quad (7)$$

В нашей задаче мы рассчитываем  $SIR$  от-

дельно для каждого клиента. В числителе фигурирует получаемый сигнал от той БС, к которой подключен наш клиент, в знаменателе – сигналы от остальных БС, которые создают помехи:

$$SIR_{dB}^i = \frac{G[i][X[i]] \cdot P_{BS}^{\max} [Y[X[i]]]}{\sum_{s=1, s \neq X[i]}^{N_{ps}} G[i][s] \cdot P_{BS}^{\max} [Y[s]]}, \quad (8)$$

$k$  – коэффициент, обеспечивающий одновременный учет в целевой функции затрат на создание сети и уровня  $SIR$  клиентов.  $k$  имеет размерность стоимости комплекса базовых станций, тем самым обеспечивая сохранение размерности в формуле (6), т.к.  $SIR$  – безразмерная величина.  $k$  должен обеспечивать «штраф» за низкий уровень  $SIR$  и «награду» за высокий. В данной работе принято  $k = -10$  единиц стоимости ( $k$  имеет отрицательное значение, т.к. мы решаем минимизационную задачу, а значит, хороший высокий уровень  $SIR$  должен уменьшать целевую функцию).

Уместность использования второго слагаемого в формуле (6) продемонстрируем на примере. Пусть у нас есть 2 разных конфигурации беспроводной сети передачи данных (рисунок 1 и рисунок 2, кружками обозначены базовые станции, перечеркнутыми кружками обозначены места-кандидаты без базовых станций, квадратами – клиенты, соединительные линии отражают факт подключения клиента к БС).

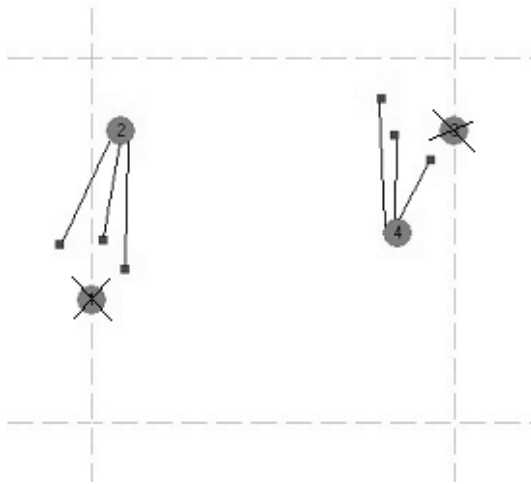


Рисунок 1 – Конфигурация сети № 1

Пусть в обоих случаях стоимость комплекса БС одинаковая, при этом каждое из решений удовлетворяет ограничениям (1) – (4). Тогда, если бы мы воспользовались формулой (5), то значение целевой функции для обоих решений было бы одинаковым. Если же пользоваться формулой (6), то конфигурация № 2 предпочтительней конфигурации № 1. В случае № 2 клиенты расположены ближе к своим базовым станциям, а значит, значения числителей выражений

(8) будут больше. Также во втором случае активные базовые станции удалены друг от друга на большее расстояние, а значит, значения знаменателей выражений (8), отражающие уровень помех от «чужих» БС, будут меньше.

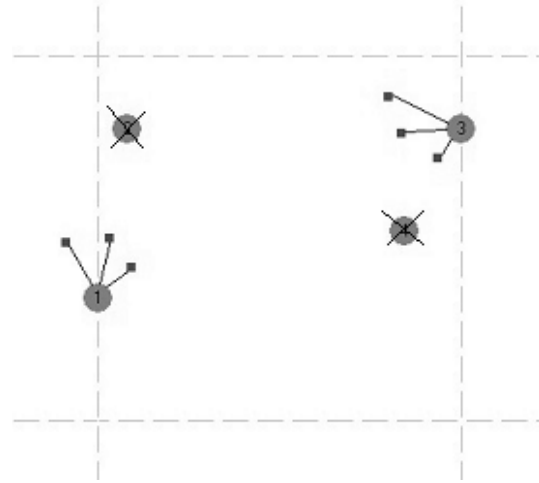


Рисунок 2 – Конфигурация сети № 2

**Эволюционный алгоритм для задачи размещения БС.** В эволюционном алгоритме рассматривается популяция, которая «живет» в течение заранее заданного числа поколений  $N_{gen}$ . Популяция состоит из  $N_{pop}$  особей, каждая из которых в нашем конкретном случае соответствует некоторому решению задачи размещения базовых станций. Каждая особь-решение состоит из двух хромосом  $Y$  и  $X$ , которые были описаны выше.

Основой работы эволюционного алгоритма является операция оценки приспособленности отдельных особей популяции. Согласно принципу естественного отбора, чем выше уровень приспособленности, тем с большей вероятностью особь должна попадать в следующие поколения. В нашей задаче уровень приспособленности особи мы возьмем обратно пропорциональным значению целевой функции решения, которое эта особь представляет. Соответственно чем меньше целевая функция, тем выше приспособленность особи.

В эволюционном алгоритме каждая его итерация (поколение) состоит из операции селекции родительской популяции из исходной и операции размножения (сюда входят скрещивание и мутация). В качестве лучшего решения возвращается лучшая особь в последнем поколении рассматриваемой популяции. На рисунке 3 в виде блок-схемы представлен общий вид эволюционного алгоритма.

Все решения будем проверять на допустимость, т.е. соответствие ограничениям (1), (2), (3) и (4). В новую популяцию будут

включаться только «жизнеспособные» особи, т.е. те, которые удовлетворяют всем ограничениям.

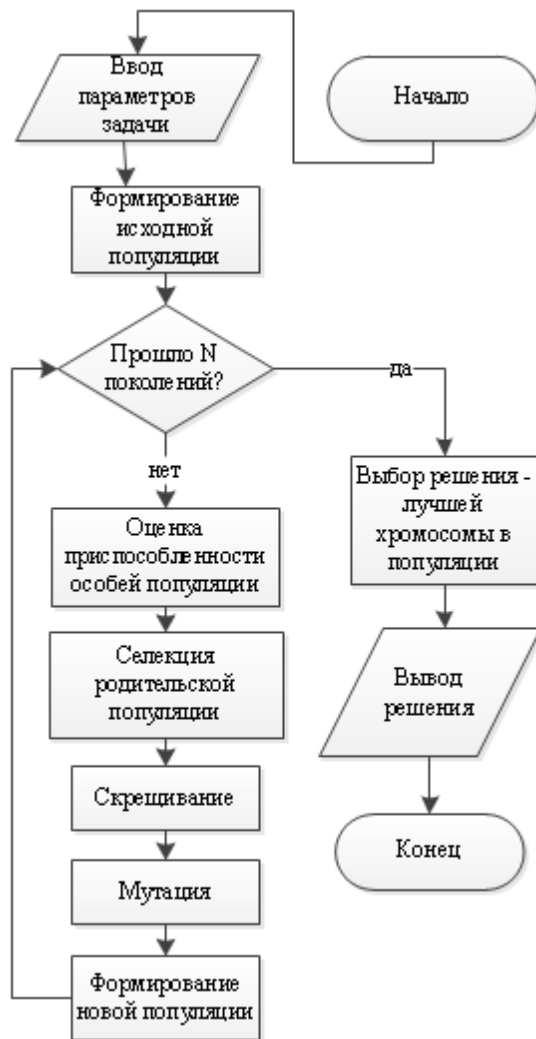


Рисунок 3 – Блок-схема эволюционного алгоритма

**Селекция родительской популяции.** Селекция заключается в выборе (исходя из приспособленности особей) тех хромосом, которые будут участвовать в создании следующего поколения. Наиболее распространёнными являются 3 метода селекции.

1. Метод рулетки. Данный метод для задач минимизации может быть кратко описан следующим образом: «Если значение целевой функции  $i$ -й особи в  $k$  раз меньше, чем значение целевой функции  $j$ -й особи, то вероятность включения  $i$ -й особи в родительский пул должна быть в  $k$  раз больше, чем аналогичная вероятность  $j$ -й особи».

2. Турнирный метод. Реализует последовательность турниров, чтобы выбрать  $N$  особей. Каждый турнир построен на выборке случайным образом  $k$  элементов из популяции и выборе лучшей особи из них. Наиболее распространен турнирный отбор с  $k = 2$ .

3. Ранговый метод. В нем вероятность попа-

дания в родительский пул пропорциональна не значению функции приспособленности, а месту, которое заняла особь в упорядоченной по значению целевой функции популяции.

Пусть у нас  $N$  особей. Тогда ранг  $r_i$   $i$ -й особи будет присвоен из следующих соображений: худшая особь имеет ранг 0, следующая по «качеству» – 1, ..., лучшая особь имеет ранг  $(N - 1)$ . Вероятность попадания  $i$ -й особи в родительскую популяцию равна:

$$p_i = \frac{2r_i}{N(N-1)}. \quad (9)$$

Отдельно стоит сказать о так называемой элитарной стратегии. Она заключается в защите лучших особей при переходе от одного поколения к другому. В классическом эволюционном алгоритме самые приспособленные особи  $i$ -го поколения не всегда переходят в  $(i+1)$ -е поколение. Элитарная стратегия применяется для предотвращения потери лучшей особи, которая гарантированно включается в новую популяцию [5]. В данной работе мы будем использовать элитарную стратегию в дополнение к вышеперечисленным методам селекции.

**Процедура размножения.** Создание потомков осуществляется при помощи генетических операторов. В эволюционном алгоритме, как правило, применяются два генетических оператора: оператор скрещивания (кроссовера) и оператор мутации.

Каждое решение нашей задачи размещения БС кодируется двумя хромосомами, поэтому процедура скрещивания будет отличаться от классической (описание классического генетического алгоритма можно найти в работе [5]).

Можно заметить, что для каждого решения вектор  $X$  полностью определяет, на каких местах в векторе  $Y$  будут стоять не нули. Вектор  $Y$  определяет только то, какой именно тип БС установлен на конкретном месте.

Так как хромосомы  $X$  и  $Y$  зависимы друг от друга, то мы не можем скрещивать их по отдельности (т.к. будет порождаться много недопустимых решений). Поэтому будем скрещивать только векторы  $X_1$  и  $X_2$  родительских особей.

Оператор скрещивания выполняется следующим образом: из родительской популяции случайным образом выбирается пара особей. Далее для каждой пары отобранных таким образом особей разыгрывается позиция гена в хромосоме  $X$ , определяющая так называемую точку скрещивания. Эта точка представляет собой целое число  $l_c$  из интервала  $[1, L-1]$ , где  $L$  – длина

хромосомы  $X$  (в нашем случае  $L = N_{tp}$ ). В результате скрещивания двух родительских хромосом получается потомок, хромосома  $X_{child}$  которого на позициях от 1 до  $l_c$  состоит из генов первого родителя, а на позициях от  $(l_c + 1)$  до  $L$  – из генов второго родителя. Модификацией однотоочечного кроссовера является  $n$ -точечный (т.н. многоточечный) кроссовер. Он аналогичен однотоочечному, однако в нем скрещивание происходит уже по  $n$  точкам.

К получившемуся вектору  $X_{child}$  потомка будет применяться операция мутации. Суть операции мутации заключается в том, что с некоторой небольшой вероятностью  $p_{mut}$  один из элементов вектора  $X_{child}$  меняет свое значение на случайное целое число из диапазона  $[1; N_{ps}]$ .

После получения вектора  $X_{child}$  особи-потомка мы должны сформировать его хромосому  $Y_{child}$ . Хромосома  $X_{child}$  однозначно определяет, какие элементы вектора  $Y_{child}$  являются ненулевыми. Например, у нас есть 5 мест-кандидатов, 4 клиента и 3 типа базовых станций (причем чем меньше порядковый номер типа БС, тем дешевле такие станции). Пусть  $X_1 = [1 \ 5 \ 2 \ 3]$  и  $X_2 = [2 \ 5 \ 5 \ 3]$ . Пусть в результате скрещивания для потомка мы получили  $X_{child} = [1 \ 5 \ 5 \ 3]$ . Это значит, что хромосома  $Y_{child}$  потомка будет иметь вид  $Y_{child} = [V \ 0 \ V \ 0 \ V]$ , где  $V$  – целое число из диапазона  $[1; N_{types}]$ .

Ненулевые элементы вектора  $Y_{child}$  рекомендуется выбирать следующим образом:

- 1) если  $Y_1[i] = w$ , а  $Y_2[i] = 0$ , то  $Y_{child}[i] = w$ ;
- 2) если  $Y_2[i] = w$ , а  $Y_1[i] = 0$ , то  $Y_{child}[i] = w$ ;
- 3) если  $Y_1[i] = Y_2[i] = w$ , то  $Y_{child}[i] = w$ ;

4) если  $Y_1[i] = w$ , а  $Y_2[i] = z$  ( $w \neq z$ ), то у нас несколько методов выбора типа БС, которая будет стоять у потомка на  $i$ -м месте:

- а) брать тот тип, у которого меньше стоимость (чтобы минимизировать целевую функцию);
- б) брать тип БС с большей производительностью;
- в) брать либо тип  $w$ , либо тип  $z$  случайно с вероятностью 0.5.

**Компьютерное моделирование.** Разработанный алгоритм реализован как программное обеспечение (ПО) в среде Embarcadero Delphi

XE5. С помощью данного ПО был проведен ряд вычислительных экспериментов по нахождению оптимального расположения базовых станций и подключению к ним клиентов. Моделирование проводилось на компьютере с процессором Intel Core i5-3470 и оперативной памятью 6 ГБ.

Первая серия вычислительных экспериментов была посвящена исследованию быстродействия метода полного перебора (ПП) для задач малой размерности. Принцип алгоритма полного перебора очень прост: мы должны перебрать все возможные решения задачи, отсеять все недопустимые решения, а среди оставшихся выбрать лучшее (с точки зрения значения целевой функции). В нашей задаче общее количество сочетаний пар хромосом  $X$  и  $Y$  равно  $N_{ps}^{N_{tp}} \cdot ((N_{types} + 1)^{N_{ps}} - 1)$ . Очевидно, что метод полного перебора не позволяет решить задачу размещения базовых станций за полиномиальное время. Результаты полного перебора при  $N_{types} = 2$  и разных значениях  $N_{tp}$  и  $N_{ps}$  приведены в таблице 1.

**Таблица 1 – Время решения задачи методом полного перебора ( $N_{types}=2$ )**

$N_{tp}$	Время решения задачи, с		
	$N_{ps}=5$	$N_{ps}=7$	$N_{ps}=10$
3	0,031	0,710	65,187
5	0,606	31,594	5140,715
7	14,654	1390,257	455798,167

Далее был проведен ряд вычислительных экспериментов с целью сравнения эффективности и быстродействия разновидностей эволюционного алгоритма. 3 метода селекции и 4 разных способа скрещивания дали 12 вариантов эволюционного алгоритма. Во всех случаях используется элитарная стратегия. Зафиксируем следующие параметры:  $N_{pop} = 100$ ,  $N_{gen} = 100$ ,  $N_{tp} = 60$ ,  $N_{ps} = 30$ ,  $N_{types} = 3$ . Результаты приведены в таблицах 2 и 3. Здесь и в дальнейшем время решения задачи приводится как среднее время за 10 запусков, значение целевой функции как лучшее за 10 запусков алгоритма.

**Таблица 2 – Зависимость времени решения задачи 60x30x3 от разновидности эволюционного алгоритма**

Метод селекции	Время решения задачи, с ( $cp$ – число точек скрещивания)			
	$cp=1$	$cp=2$	$cp=3$	$cp=4$
Рулет.	0,370	0,372	0,376	0,378
Турнир.	0,386	0,389	0,403	0,404
Ранг.	0,371	0,372	0,374	0,375

**Таблица 3 – Результат решения задачи 60x30x3 в зависимости от разновидности эволюционного алгоритма**

Метод селекции	Целевая функция, усл. ед. ( $cp$ – число точек скрещивания)			
	$cp=1$	$cp=2$	$cp=3$	$cp=4$
Рулет.	116999	121831	119460	111594
Турнир.	110460	107552	109371	108479
Ранг.	111831	104131	107552	108479

Представленные данные свидетельствуют о том, что в плане быстродействия варианты эволюционного алгоритма почти не отличаются друг от друга. Турнирный метод селекции по скорости незначительно проигрывает двум остальным. Решения с лучшей целевой функцией выдает ранговый метод селекции с 2-точечным скрещиванием. Для дальнейших исследований в нашей работе мы будем использовать именно его.

Затем был проведен ряд вычислительных экспериментов с целью сравнения эффективности предложенного эволюционного алгоритма и метода полного перебора при решении задач малой размерности. Для эволюционного алгоритма зафиксируем следующие параметры:  $N_{pop} = 100$ ,  $N_{gen} = 100$ , метод селекции – ранговый, скрещивание – 2-точечное. Результаты приведены в таблице 4.

Представленные данные свидетельствуют о том, что на задачах малой размерности эволюционный алгоритм обеспечивает получение точных значений целевой функции, как и метод полного перебора, который дает точное решение при каждом запуске алгоритма.

При этом теория генетических алгоритмов дает основание полагать, что при повышении размерности решаемых задач предложенный эволюционный алгоритм будет обеспечивать получение субоптимальных решений, отличающихся от точных. Однако проверить данный факт посредством экспериментов не представляется возможным по причине экспоненциальной зависимости времени работы метода полного перебора от количества входных данных.

**Таблица 4 – Сравнение результатов эволюционного алгоритма (ЭА) и метода полного перебора (ПП)**

Задача ( $N_{tp} \times N_{ps} \times N_{bs\_types}$ )	Целевая функция решения, усл. ед.	
	ЭА	ПП
3x5x2	27382	27382
3x7x2	26784	26784
5x5x2	26776	26776
5x7x2	26570	26570

Далее был проведен ряд вычислительных экспериментов для выявления зависимости времени работы алгоритма и качества решений от числа поколений. Зафиксируем следующие параметры:  $N_{tp} = 60$ ,  $N_{ps} = 30$ ,  $N_{types} = 3$ ,  $N_{pop} = 100$ , метод селекции – ранговый, скрещивание – 2-точечное. Результаты приведены в таблице 5.

**Таблица 5 – Зависимость времени работы и результата ЭА от числа поколений**

Число поколений	10	100	1000	10000
Время работы, с	0,231	0,372	1,686	14,709
Целевая функция, усл. ед.	107152	104131	100214	99898

Был проведен ряд экспериментов для выявления зависимости времени работы алгоритма и качества решений от размера популяции. Зафиксируем следующие параметры:  $N_{tp} = 60$ ,  $N_{ps} = 30$ ,  $N_{types} = 3$ ,  $N_{gen} = 100$ , метод селекции – ранговый, скрещивание – 2-точечное. Результаты приведены в таблице 6.

Если рассматривать влияние параметров настройки эволюционного алгоритма на качество получаемых решений, то можно отметить следующую тенденцию: увеличение числа поколений и размера исходной популяции приводит к повышению качества получаемого решения, но при этом увеличивается время, необходимое для нахождения решения. Таким образом, разработанный эволюционный алгоритм обеспечивает возможность нахождения компромисса между качеством получаемого решения и временем, затраченным на его поиск, в зависимости от выбранных значений параметров настройки.

**Таблица 6 – Зависимость времени работы и результата ЭА от размера популяции**

Размер популяции	10	100	1000	10000
Время работы, с	0,046	0,372	3,556	47,771
Целевая функция, усл. ед.	115612	104131	100162	99012

Также был проведен ряд вычислительных экспериментов для выявления зависимости времени работы алгоритма от размера (количества клиентов и мест-кандидатов) задачи. Зафиксируем следующие параметры:  $N_{types} = 3$ ,  $N_{pop} = 100$ ,  $N_{gen} = 100$ , метод селекции – ранговый, скрещи-

вание – 2-точечное. Результаты приведены в таблице 7. Видно, что даже при решении задач большой размерности предложенный эволюционный алгоритм показывает время работы, не превышающее минуту.

**Таблица 7 – Зависимость времени работы ЭА от размера задачи**

$N_{tp}$	Время решения задачи, с			
	$N_{ps}=50$	$N_{ps}=100$	$N_{ps}=500$	$N_{ps}=1000$
50	0,326	0,511	1,960	3,738
100	0,496	0,787	3,175	5,246
500	2,651	2,738	11,174	20,835
1000	3,204	4,301	21,562	50,962

**Заключение.** Проведенное компьютерное моделирование позволяет сделать следующие выводы:

- предложенная модификация эволюционного алгоритма позволяет решать задачу размещения базовых станций;
- эволюционный алгоритм может находить решение задачи размещения БС за приемлемое время;
- для задач малой и средней размерности результаты эволюционного алгоритма совпадают с результатами метода полного перебора;

– среди всех разновидностей метода наилучшие результаты показывает эволюционный алгоритм, в котором осуществляется ранговая селекция, используется элитарная стратегия и применяется 2-точечное скрещивание особей;

– увеличение числа поколений и размера исходной популяции приводит к повышению качества получаемого решения, но при этом увеличивается время, необходимое для нахождения решения.

#### **Библиографический список**

1. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003. 512 с.
2. Ермолаев С.Ю. Оптимальное размещение базовых станций // Telecommunication Sciences. 2010. Т. 1. № 1. С. 349-355.
3. St-Hilaire M. Efficient solution of the 3G network planning problem // Computers & Industrial Engineering. 2012. Т. 63. № 4. Р. 819-830.
4. Koskie S., Gajic Z. A Nash game algorithm for SIR-based power control in 3G wireless CDMA networks // IEEE/ACM Transactions on Networking. 2005. Т. 13. № 5. Р. 1017-1026.
5. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия-Телеком, 2006. 452 с.

УДК 621.317.75:519.2

**М.М. Гурарий, М.М. Жаров, С.Г. Русаков, С.Л. Ульянов**

## **МЕТОДЫ МОДЕЛИРОВАНИЯ В ЧАСТОТНОЙ ОБЛАСТИ НЕЛИНЕЙНЫХ РАДИОТЕХНИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ**

*Приводятся результаты разработки перспективных методов схемотехнического анализа в частотной области. Обсуждаются способы повышения вычислительной эффективности метода гармонического баланса для моделирования нелинейных периодических процессов в неавтономных схемах и схемах автогенераторов, а также методов моделирования фазового шума автогенераторов.*

**Ключевые слова:** установившийся режим, гармонический баланс, автогенераторы, фазовый шум.

**Введение.** Разработка высокоэффективных методов и алгоритмов автоматизации проектирования радиотехнических схем является в настоящее время одним из наиболее актуальных научно-исследовательских направлений развития САПР СБИС. Наиболее важными с точки зрения выполнения задач схемотехнического анализа и верификации проекта являются средства математического моделирования схем на транзисторном уровне. Традиционные методы и алгоритмы математического моделирования

схем, реализованные в различных применяемых схемных симуляторах, часто не позволяют получать все желаемые характеристики современных радиотехнических ИС, а также не обеспечивают необходимых требований по точности и вычислительной эффективности. В связи с необходимостью расчета ряда специфических характеристик и режимов проектирование радиотехнических блоков современных ИС и систем должно поддерживаться специализированными вычислительными методами, которые обеспечивают

анализ периодического или квазипериодического установившегося режима, расчет гармонических и интермодуляционных искажений, анализ периодических режимов схем автогенераторов, расчет фазового шума и джиттера и т.п.

Переход к технологиям с нанометровыми проектными нормами приводит к дополнительным проблемам, которые не существовали ранее. Шум в полупроводниковых приборах, паразитное взаимодействие элементов, большое число нелинейных элементов требуют применения новых вычислительных процедур при моделировании. Приборный шум при проектных нормах 45 нм и менее становится физическим эффектом первого порядка, влияющим на динамический диапазон. Необходимо учитывать влияние как «белого», так и «фликкер» шума на отдельные радиотехнические блоки и системы в целом. Рост числа нелинейных элементов требует разработки эффективных вычислительных процедур анализа нелинейных искажений. Учет взаимодействия элементов приводит к необходимости моделирования схем очень высокой размерности, содержащих 100000 и более элементов, что требует создания методов решения высоко-размерных задач.

Усилия специалистов направлены на разработку эффективных вычислительных алгоритмов моделирования нелинейных радиотехнических интегральных схем как во временной области, так и в частотной области. Отдельное направление исследований составляет разработка смешанных частотно-временных алгоритмов моделирования. *Цель работы* – анализ основных подходов повышения вычислительной эффективности метода гармонического баланса для моделирования нелинейных периодических процессов в неавтономных схемах и схемах автогенераторов, а также методов моделирования фазового шума автогенераторов.

**Алгоритмы гармонического баланса для частотного анализа установившихся периодических режимов.** Для анализа установившегося периодического режима применяются специальные методы как в частотной, так и во временной области [1]. Частотные методы предпочтительны при анализе радиочастотных и микроволновых схем, которые содержат большое число элементов с распределенными параметрами. Эти методы эффективны для схем с невысокой степенью нелинейности. При анализе сильно нелинейных схем временные методы могут оказаться более эффективными. Применяемые вычислительные методы анализа установившегося периодического режима во временной области основаны, как правило, на методе пристрелки. В

основе частотных методов анализа периодических режимов лежит гармонический баланс [2-5].

Метод гармонического баланса (ГБ) является эффективным методом определения установившегося режима в различных приложениях и, в частности, при моделировании нелинейных электронных схем. Метод ГБ является частотным методом, который обладает рядом преимуществ по сравнению с временными методами анализа установившихся режимов. Метод позволяет эффективно находить установившийся режим в схемах с большим разбросом постоянных времени, в схемах с многочастотным возбуждением и в схемах, содержащих элементы с распределенными параметрами.

Рассмотрим дифференциальное уравнение вида

$$f(x, \dot{x}, u) = 0. \quad (1)$$

Здесь  $u$  – периодическая синусоидальная функция с периодом  $T$ . Гармонический баланс можно рассматривать как метод решения (1) с помощью ряда Фурье

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega t}, \quad (2)$$

где  $\omega = 2\pi/T$ .

Подставляя выражение (2) в (1) и полагая периодичность функции  $f$ , получаем

$$f(x, \dot{x}, u) = \sum_{k=-\infty}^{\infty} F_k(X, U) e^{jk\omega t}.$$

Принимая во внимание линейную независимость экспонент, имеем

$$F_k(X, U) = 0 \quad (3)$$

для каждого индекса гармоник  $k$ .

Принцип гармонического баланса устанавливает, что  $X$  является решением (3) тогда и только тогда, если  $x(t)$  есть решение (1) [2]. Таким образом, ГБ позволяет свести задачу (1) к решению систем нелинейных алгебраических уравнений (3).

В практике схемотехнического моделирования метод ГБ имеет две альтернативные формулировки. В соответствии с первой (подсхемный метод ГБ) схема разбивается на две подсхемы, которые содержат линейные и нелинейные элементы. Вектор переменных содержит только напряжения и токи портов соединения подсхем [6]. Данный подход становится неэффективным при моделировании монолитных ИС, в которых к каждому узлу подключены как линейные, так и нелинейные элементы [7]. Поэтому во многих практических симуляторах применяется формулировка метода ГБ, основанная на модифицированном узловом базисе.



Нелинейная схема может быть описана системой дифференциально-алгебраических уравнений [2, 3]

$$\frac{dq(x(t))}{dt} + i(x(t)) + u(t) = 0. \quad (4)$$

Здесь  $q(\cdot)$ ,  $i(\cdot)$ ,  $u(\cdot)$  – векторы узловых зарядов, токов и входных воздействий,  $x(t)$  – вектор узловых напряжений. Полагая, что  $u(t)$  и  $x(t)$  – периодические функции и ряд Фурье имеет  $K$  гармоник, уравнение (3) примет вид [2, 3]:

$$R_k(X) = jk\omega Q_k(X) + I_k(X) + U_k = 0, \quad (5)$$

где  $Q$ ,  $I$ ,  $U$  – Фурье-образ  $q(\cdot)$ ,  $i(\cdot)$ ,  $u(\cdot)$  и  $k = -K, \dots, 0, \dots, K$ .

Выражение (5) – это система алгебраических уравнений размерности  $(2K+1)N$ , где  $N$  – число схемных переменных. Для учета элементов с распределенными параметрами в уравнение (5) необходимо добавить член  $YX$ , где  $Y$  – матрица адмитансов. При численном решении системы (5) выполняется вычисление векторов зарядов и токов нелинейных элементов. Для этого решение в частотной области  $X$  преобразуется во временную область  $x(t)$ . Затем вычисляются вклады нелинейных элементов  $q(x(t))$ ,  $i(x(t))$ . Эти векторы преобразуются в частотную область  $Q(X)$ ,  $I(X)$  и вычисляются векторы (5). Для выполнения прямого и обратного преобразований Фурье используется алгоритм БПФ.

При возбуждении схемы сигналами с несоизмеримыми частотами  $\omega_1, \omega_2$  установившийся режим является квазипериодическим

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} X_{kl} e^{j(k\omega_1 + l\omega_2)t}. \quad (6)$$

В этом случае система уравнений ГБ имеет вид [2, 3]

$$R_{kl}(X) = j(k\omega_1 + l\omega_2)Q_{kl}(X) + I_{kl}(X) + U_{kl} = 0 \\ k = -K, \dots, 0, \dots, K, \quad l = -L, \dots, 0, \dots, L.$$

Вектор решения содержит частотные компоненты со всевозможными комбинациями  $(k\omega_1 + l\omega_2)$ . Размерность вектора  $(2K+1)(2L+1)N$ . Показано, что в этом случае для выполнения преобразования сигналов можно применить многомерное преобразование Фурье [8, 9].

**Решение систем нелинейных алгебраических уравнений в методе ГБ.** Численное решение систем алгебраических уравнений большой размерности вида (5) представляет значительные трудности, для преодоления которых предложено несколько подходов. В рамках подсхемного варианта ГБ для решения системы уравнений предложено использовать оптимизационные ме-

тоды [6]. Данный подход становится неэффективным при использовании узловой формулировки ГБ ввиду большого числа переменных [2]. Другим способом снижения вычислительных затрат является применение различных релаксационных методов [10, 11]. В работе [2] показано, что релаксационные методы не обеспечивают сходимости к решению и, следовательно, надежные вычислительные алгоритмы должны быть основаны на методе Ньютона.

Применение метода Ньютона для решения (5) приводит к системе линейных алгебраических уравнений (СЛАУ) на каждой итерации [2, 3]

$$J(X^s)\Delta X^s = -R(X^s). \quad (7)$$

Здесь  $R(X) = j\Omega Q(X) + I(X) + U$ ,  $\Omega$  – блочно-диагональная матрица и  $s$  – номер итерации. Матрица якобиана имеет вид

$$J(X) = \frac{\partial R(X)}{\partial X} = \frac{\partial I(X)}{\partial X} + j\Omega \frac{\partial Q(X)}{\partial X}. \quad (8)$$

Матрица  $J$  имеет размерность  $(2K+1)N$  на  $(2K+1)N$  и является блочной. Решение высоко-размерной СЛАУ (7) представляет основную вычислительную трудность в применении ГБ с использованием метода Ньютона.

**Решение систем линейных алгебраических уравнений в методе ГБ.** Применение техники разреженных матриц для решения СЛАУ (7) не позволяет получить эффективный вычислительный метод ГБ. Ряд подходов, предложенных в [2], позволил получить лишь относительно небольшое ускорение вычислительного процесса. Радикальное повышение эффективности вычислений достигается за счет применения итерационных методов решения СЛАУ.

Введем обозначения  $\Gamma$ ,  $\Gamma^{-1}$  для матриц прямого и обратного преобразования Фурье. Тогда система уравнения (5) может быть записана в виде

$$R(X) = j\Omega\Gamma q(\Gamma^{-1}X) + \Gamma i(\Gamma^{-1}X) + \Gamma u = 0. \quad (9)$$

Матрица якобиана примет вид

$$J(X) = j\Omega\Gamma\tilde{C}\Gamma^{-1} + \Gamma\tilde{G}\Gamma^{-1}, \quad (10)$$

где  $\tilde{C}, \tilde{G}$  – диагональные матрицы узловых емкостей и проводимостей, рассчитанные во временной области.

Пусть для решения (7) используется итерационный метод из класса методов крыловских подпространств. Известно, что для реализации метода требуется операция матрично-векторного произведения [12]. В случае решения (7) с матрицей (10) для некоторого вектора  $Z$  необходимо вычислить

$$J(X)Z = j\Omega\tilde{\Gamma}^{-1}Z + \Gamma\tilde{\Gamma}^{-1}Z. \quad (11)$$

Структура данного выражения позволяет применить эффективную процедуру вычислений, в которой выполняется умножение матриц размерности  $N$ , прямое и обратное преобразование Фурье [13, 14].

Для построения вычислительного процесса решения СЛАУ применяются итерационные методы QMR [15] и GMRES [12]. Сравнительные исследования показали, что для реализации метода ГБ более предпочтительным является метод GMRES [16].

Непосредственное применение итерационного метода для решения (7) сталкивается с проблемой медленной сходимости метода. Для улучшения сходимости метода используется предобуславливатель. Во многих практических случаях в качестве предобуславливателя используется блочно-диагональная часть матрицы якобиана. Для моделирования схем с сильно нелинейными характеристиками результирующая эффективность достигается применением более сложных предобуславливателей [17-19].

**Адаптивный гармонический баланс.** По сравнению с узловой формулировкой ГБ достоинством подсхемного варианта ГБ является сокращение числа переменных. Как было отмечено выше, в монолитных ИС затруднительно выделить чисто «линейные» узлы, т. е. узлы, к которым подключены только линейные элементы. Однако можно ввести оценку степени нелинейности узла. На основании этой оценки предлагается [20] динамически изменять число гармоник для каждого узла индивидуально. Тогда число переменных ГБ и порядок системы уравнений имеет вид

$$M = \sum_{i=1}^N (2K_i + 1). \quad (12)$$

Обычно в схеме большая часть узлов характеризуется слабой нелинейностью, для них можно использовать меньшее число гармоник  $K_i \ll K$ . При условии, что число узлов со слабо нелинейным поведением в схеме велико, получаем  $M \ll (2K+1)N$ . Таким образом, подход позволяет значительно снизить размерность решаемых СЛАУ и получить сокращение вычислительных затрат.

Линейная система (7) может быть записана в виде [21, 22]

$$JD^{-1}D\Delta X = -R, \quad (13)$$

где  $D$  обозначает блочно-диагональную часть матрицы  $J$ . Используя подстановку  $W = -R - D\Delta X$ , получаем эквивалентное (13)

уравнение

$$JD^{-1}W = R - JD^{-1}R. \quad (14)$$

Компоненты вектора правой части (14), соответствующие линейным узлам, равны нулю. Следовательно, вектор решения (14) имеет соответствующие нулевые компоненты. Также можно ожидать, что для слабо нелинейных узлов вектор правой части (14) содержит меньшее число значимых гармоник по сравнению с вектором  $R$ . Поэтому степень нелинейности узла предлагается оценивать по норме компонента вектора правой части, соответствующего этому узлу [21, 22]. В зависимости от полученной оценки можно изменять число гармоник в процессе моделирования для каждого узла индивидуально. Оценку предлагается выполнять перед каждой ньютоновской итерацией, что приводит к решению СЛАУ различного порядка на различных итерациях метода Ньютона. Вычислительная схема метода адаптивного ГБ включает процедуру оценки степени нелинейности и определения числа переменных, процедуры формирования и решения редуцированных СЛАУ, проверки критерия снижения размерности [3, 21, 22].

В таблице представлены результаты расчетов для некоторых схем. Таблица содержит исходную размерность задачи ГБ  $(2K+1)N$  и фактор снижения размерности ( $\Phi$ ), который дает усредненное по ньютоновским итерациям отношение исходной размерности к размерности редуцированной системы. Как видно из таблицы, применение адаптивного метода позволяет в несколько раз снижать размерность решаемой линейной системы.

#### Результаты моделирования схем

№	Схема	$N$	$(2K+1)N$	$\Phi$
1	Источник втор. питания	6	486	6,1
2	Усилитель	8	488	3,8
3	Дифф. пара	12	492	5,0
4	Усилитель класса С	12	612	5,9
5	МДП усилитель	24	1464	3,2
6	ОУ uA741	29	4089	5,6
7	Фильтр 5 порядка	183	7503	4,2
8	ОУ mosAMP	159	38319	6,3

**Метод гармонического баланса для анализа автогенераторов.** Для автономных схем входное воздействие отсутствует ( $U=0$ ), а частота  $\omega$  включается в вектор неизвестных. Для того чтобы число переменных соответствовало числу уравнений, вводится дополнительное условие нулевой фазы первой гармоники напряжения в заданном  $p$ -м узле. Это условие записывается

как равенство нулю мнимой части соответствующей комплексной гармоники и обеспечивает выбор единственного из бесконечного множества решений, различающихся фазовым сдвигом. Система уравнений ГБ принимает в этом случае вид [2, 3]:

$$\begin{aligned} R(X, \omega) = I(X) + j\Omega Q(X) = 0 \\ \text{Im}(X_{1,p}) = 0 \end{aligned} \quad (15)$$

Для решения системы (15) применяется, как правило, метод Ньютона, на каждом шаге которого решается система линейных алгебраических уравнений

$$J\Delta X + \Lambda\Delta\omega = -R(X, \omega) \quad \text{Im}(\Delta X_{1,p}) = 0. \quad (16)$$

Здесь  $\Delta X$ ,  $\Delta\omega$  – это коррекции решения на шаге Ньютона,  $J = \partial R(X, \omega) / \partial X$  – матрица якобиана системы (15),  $\Lambda = \partial R(X, \omega) / \partial \omega$ . Проблема применения метода Ньютона (16) для решения (15) связана с необходимым учётом вырожденного (тривиального) решения, которое соответствует статической рабочей точке (нулевым амплитудам всех гармоник напряжений и произвольному значению частоты). Для исключения сходимости алгоритма Ньютона к вырожденному решению необходимо задавать начальное приближение, достаточно близкое к требуемому периодическому решению [3, 23]. Однако при разработке генераторной схемы для пользователя практически невозможно оценить желательный высокоразмерный вектор начальных гармоник. Поэтому для улучшения сходимости при моделировании генераторных схем введена концепция искусственного зонда [24], который представляет собой соединение источника напряжения или тока с идеальным фильтром первой гармоники. Введение зонда в схему автогенератора позволяет свести автономную систему уравнений к неавтономной системе уравнений с дополнительными условиями [24]

$$\text{Re}\{I_{prb}(V_{prb}, \omega)\} = 0, \quad \text{Im}\{I_{prb}(V_{prb}, \omega)\} = 0, \quad (17)$$

где  $I_{prb}$ ,  $V_{prb}$  – ток и напряжение искусственного зонда. Вычислительная схема в этом случае состоит из внешнего и внутреннего блоков. Во внешнем блоке решается система двух уравнений (17) относительно амплитуды и частоты искусственного зонда. Во внутреннем блоке решается неавтономная система уравнений гармонического баланса (5) с найденными значениями амплитуды и частоты искусственного зонда. Полученное решение для схемы с зондом является решением для исходной схемы (15), так как выполнение условий нулевого тока (17) эквивалентно отсутствию зонда. Применение такого подхода позволяет задавать начальные значения

только для двух неизвестных:  $V_{prb}$  и  $\omega$ . Оценка начального значения частоты  $\omega$  может выполняться с помощью алгоритма анализа полюсов передаточной функции линеаризованной схемы в статической рабочей точке [25], и тогда задачей пользователя остается лишь задание корректного значения амплитуды  $V_{prb}$ .

Дальнейшее совершенствование метода основано на полной автоматизации оценки начального приближения. Для этой оценки было предложено использовать значение  $V_{prb}$ , при котором достигается минимум зависимости  $|I_{prb}(V_{prb})|$  [24]. Предложено несколько модификаций этого метода. В [26] для улучшения свойств сходимости внешнего итерационного процесса предлагается оптимизационная процедура. В [27] внешний итерационный процесс строится на основе метода гомотопии. В [28] используется полигармонический искусственный зонд, для которого требуются начальные значения высших гармоник. В [29] предложена процедура автоматического выбора узла подключения искусственного зонда и алгоритм решения (17) на основе анализа изменения активной и реактивной мощности при вариации амплитуды зонда.

Основное ограничение подхода [24] состоит в его недостаточной надежности. Как показано в работах [30, 31], для высокооборотных автогенераторов зависимость  $|I_{prb}(V_{prb})|$  часто может не иметь локального минимума, а в тех случаях, когда он присутствует, полученное приближение не всегда обеспечивает сходимость итераций.

Проведённый анализ [30, 31] показал, что источник недостатков метода связан с постоянством частоты при построении зависимости  $|I_{prb}(V_{prb})|$ . Для его устранения предлагается новый тип искусственно встраиваемого зонда с фиксацией не частоты зонда, а фазы его тока, которая принимается нулевой относительно сигнала зонда. При изменении амплитуды (процесс продолжения по параметру) частота зонда меняется так, чтобы обеспечивалась нулевая фаза тока зонда. Этот метод был назван методом продолжения решения по параметру с применением «зонда с подстройкой частоты» (ПЧ) [30, 31]. Вычислительная схема метода включает процедуру трассировки траектории кривой  $\text{Re}(I_{prb}(V_{prb}, \omega(V_{prb})))$  при определении зависимости  $\omega(V_{prb})$  из решения уравнения  $\text{Im}(I_{prb}(V_{prb}, \omega(V_{prb}))) = 0$ . В процедуре продолжения осуществляется управление величиной шага, а также автоматическое изменение параметра продолжения для вычисления траектории

в случае её неоднозначности [32, 33]. Процесс завершается, когда величина отклика схемы (действительная часть тока зонда) становится равной нулю.

Наиболее важная особенность предложенного метода с ПЧ – существенное улучшение сходимости по сравнению с методом без ПЧ. Сравнительный анализ методов [32], проведенный для набора из 11 тестовых генераторных схем, экспериментально подтвердил этот вывод. Метод позволяет находить решение в случае сложных траекторий и при условии существования множества решений [32].

**Метод анализа фазового шума автогенераторов.** Идеальный автогенератор обеспечивает колебания на фундаментальной частоте. В реальных схемах наличие источников шума в полупроводниковых приборах приводит к искажению выходного сигнала и появлению колебаний на частотах около фундаментальной частоты. Эти колебания обусловлены воздействием шума, который изменяет как амплитуду, так и фазу выходного сигнала. В соответствии с нелинейной природой автогенератора амплитудные отклонения со временем затухают, а фазовые отклонения сохраняются и могут неограниченно накапливаться с течением времени.

Таким образом, доминирующим фактором, определяющим спектр колебания генератора, являются флуктуации фазы колебания. Случайные воздействия приводят к размыванию спектра сигнала вблизи гармоник частоты колебаний. На практике интересуются шириной и формой спектра вблизи первой гармоники или частоты колебаний. Для характеристики фазового шума автогенератора используются спектральная плотность мощности (СПМ) сигнала  $x$  ( $S_x$  или  $L$ ) или фазы сигнала  $\phi$  ( $S_\phi$ ) [34]. Спектральная плотность мощности  $S_x$  содержит как амплитудные, так и фазовые компоненты шума, а  $S_\phi$  – только фазовые.

Для выполнения шумового анализа и расчета СПМ шума предложено несколько методов, различающихся способом представления математической модели схемы и принятой системой допущений. Допущение о малости шумового воздействия по сравнению с полезным сигналом позволяет строить методы шумового анализа на основе теории возмущений и линеаризации нелинейных систем.

Теория линейных стационарных систем (ЛТС) позволяет на качественном уровне предсказывать шумовые характеристики. Полученные на основе этой теории выражения для расчета шума приводят к значительным отклонениям

от экспериментальных результатов. С целью уменьшения различий была получена полуэмпирическая модель, которая позволяет рассчитывать СПМ шума для LC генератора [35].

Теория фазового шума в автогенераторах, основанная на теории возмущений и декомпозиции возмущенного отклика на составляющие, обуславливающие фазовый и амплитудный шум, разработана в работах [36-39]. Возмущенное решение представляется в виде:

$$x(t) = x_s(t + \theta(t)) + \Delta x(t + \theta(t)), \quad (18)$$

где  $x_s(t + \theta(t))$  – временной сдвиг вдоль предельного цикла, и случайная переменная  $\theta(t)$  отвечает за возмущения по касательной к предельному циклу,  $\Delta x(t + \theta(t))$  – амплитудные отклонения от предельного цикла.

Используя результаты теории Флоке при дополнительном условии на вектор возмущения, в работе [37] получено скалярное уравнение

$$\frac{d\theta(t)}{dt} = v^T(t + \theta(t))b(t). \quad (19)$$

Здесь  $v(t)$  – вектор проекции возмущений (PPV),  $b(t)$  – внешнее воздействие,  $\theta(t)$  – неизвестная фазовая переменная.

Рассматривая  $\theta(t)$  как стохастическую функцию и применяя корреляционную теорию, в работах [38, 39] получены выражения для СПМ фазового шума. Показано, что спектр автогенератора имеет характер лоренциана около каждой гармоники.

Несмотря на то, что эта теория позволяет находить спектр автогенератора вследствие фазового шума в виде лоренциана и определять дисперсию фазового шума, вычислительные методы, построенные на базе этой теории, имеют ряд ограничений. Во-первых, получение векторов Флоке, необходимых для анализа шума, сталкивается с серьезными вычислительными трудностями. Во-вторых, методы позволяют определять компоненту шума, обусловленную фазовыми отклонениями, которая является доминирующей при приближении к частоте колебаний автогенератора. В случае если необходимо определять шум для достаточно больших смещений от частоты колебаний, методы будут давать погрешность, так как влияние амплитудных отклонений не учитывается.

Линейная периодически нестационарная модель (LPTV) широко используется для анализа [34]. Существенным достоинством этого подхода является простота и удобство его внедрения в программы схемотехнического моделирования в рамках периодического малосигнального анализа. Шум на выходе схемы с периодически изме-

няющейся рабочей точкой имеет характер циклоstationарного процесса вне зависимости от типа источника шума [34]. Для stationарных источников, зависящих от режима, циклоstationарность обусловлена модуляцией периодически изменяющейся величиной смещения, для независимых от режима stationарных источников - периодической передаточной функцией от источника на выход схемы.

При малом периодическом возбуждении поведение автогенератора описывается линейной системой с периодическими коэффициентами. В частотной области линейная периодически неstationарная модель для автогенераторов записывается аналогично случаю неавтономных систем и имеет вид [3]

$$J(\Delta\omega)\Delta X = B. \quad (20)$$

Здесь  $B$  – это вектор гармоник возбуждения, где компоненты  $B_{kl}$  – гармонические сигналы частотой  $k\omega_0 + \Delta\omega$ , приложенные к узлу  $l$ ;  $\Delta X$  – вектор малосигнального решения,  $J(\Delta\omega)$  – матрица якобиана при заданном отклонении  $\Delta\omega$  частоты возбуждения от собственной частоты генератора  $\omega_0$ . Матрица  $J(\Delta\omega)$  имеет вид [3]

$$J(\Delta\omega) = G + j(\omega_0\Lambda + \Delta\omega E)C = J_0 + j\Delta\omega C, \quad (21)$$

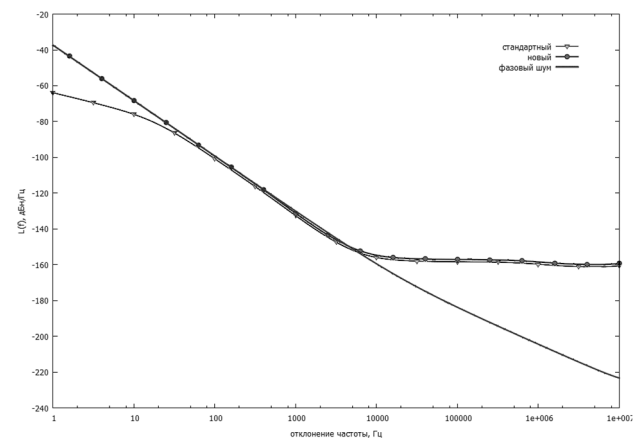
где  $\Lambda = \text{diag}(-k, \dots, 0, \dots, k)$  – диагональная матрица индексов гармоник,  $J_0 = G + j\omega_0\Lambda C$  – матрица гармонического якобиана в точке периодического решения,  $E$  – единичная матрица. Матрица  $J(\Delta\omega)$  при  $\Delta\omega = 0$  совпадает с матрицей  $J_0$ ,  $J(0) = J_0$ .

Матрица  $J_0$  является вырожденной, поэтому вблизи гармоник собственной частоты генератора матрица системы  $J(\Delta\omega)$  плохо обусловлена. В некоторых случаях эта вырожденность вызывает рост численной погрешности и приводит к неустойчивости оценки СПМ в области малых отклонений от собственной частоты. Во многих других случаях матрица якобиана, полученная численными методами, отличается от теоретического значения, и не является вырожденной при нулевом отклонении от собственной частоты. Это приводит к появлению плоской части на графике СПМ автогенератора без фликкер-шума и к неверному наклону графика СПМ автогенератора с фликкер-шумом. Поэтому на практике такой подход не позволяет охватить весь диапазон частот.

Численный метод расчета СПМ шума, свободный от указанных недостатков, предложен в работе [40]. Метод основан на эквивалентном преобразовании системы уравнений периодиче-

ского малосигнального анализа. Показано, что предложенное эквивалентное преобразование системы позволяет исключить сингулярность матрицы при нулевых смещениях частоты и соответственно устранить вычислительные трудности при расчетах вблизи частоты колебаний.

В качестве иллюстрации на рисунке представлены результаты расчетов СПМ шума схемы автогенератора, выполненного по КМОП технологии. Схема содержит 91 МДП транзистор и имеет частоту колебаний 15,1 МГц. На рисунке представлены зависимости СПМ от частоты смещения, полученные различными методами: исходный метод расчета на основе линейной периодически неstationарной модели (стандартный), предложенный метод расчета (новый), метод расчета на основе фазового уравнения (фазовый шум). Можно отметить хорошее совпадение результатов стандартного и предложенного методов расчета вплоть до частоты 50 Гц, ниже которой стандартный метод дает существенную погрешность. Отметим также хорошее совпадение СПМ шума, полученного с помощью предложенного метода, с фазовым шумом в области малых смещений. Метод расчета фазового шума дает погрешность при больших смещениях. Таким образом, предложенный метод обеспечивает вычисление СПМ во всем частотном диапазоне отклонений.



**Зависимости СПМ шума от частоты смещения, полученные с помощью различных методов**

**Заключение.** В статье проведен анализ способов повышения эффективности частотных методов схемотехнического анализа, включая развитие метода гармонического баланса для моделирования нелинейных периодических процессов и фазового шума автогенераторов.

#### Библиографический список

1. Русаков С.Г. Моделирование нелинейных радиочастотных схем в системах автоматизации схемотехнического проектирования // Автоматизация проектирования. -1997. - № 2. - С. 2-8.

2. Kundert K.S., White J., Sangiovanni-Vincentelli A. Steady-State Methods for Simulating Analog and Microwave Circuits. – Boston: Kluwer Academic Publishers, 1990. – P. 247.

3. Актуальные проблемы моделирования в системах автоматизации схемотехнического проектирования/ под ред. А.Л. Стемповского – М.: Наука, 2003. – 430 с.

4. Алексеев О.В., Асович П.Л., Соловьев А.А. Спектральные методы анализа нелинейных радиоустройств с помощью ЭВМ. – М.: Радио и связь, 1985. – 152 с.

5. Ильин В.Н., Жигалов И.Е., Ланцов В.Н. Методы автоматизированного схемотехнического проектирования нелинейных радиотехнических цепей // Известия Вузов. Радиоэлектроника. – 1985. – №6. – С. 7-17.

6. Nakla M.S., Vlach J. A piecewise harmonic balance technique for determination of the periodic response of nonlinear systems // IEEE Trans. on Circuits and Systems. – 1976. – Vol. CAS-23. – No 2. – P. 85-91.

7. Kundert K.S., Sangiovanni-Vincentelli A. Simulation of nonlinear circuits in the frequency domain // IEEE Trans. on Computer-Aided Design. – 1986. – Vol. CAD-5. – No 4. – P. 521-535.

8. Rizzoli V. General-purpose harmonic balance analysis of nonlinear microwave circuits under multitone excitation / V. Rizzoli, A. Cecchetti, A. Lipparini, F. Mastri // IEEE Trans. on Microwave Theory and Techniques. – 1988. – Vol. MTT-36. – P. 1650-1659.

9. Brachtendorf H.G. Numerical steady state analysis of electronic circuits driven by multi-tone signals / H.G. Brachtendorf, G. Welsch, R. Laur, A. Bunse-Gerstner // Electrical Engineering. – 1996. – No 79. – P. 103-112.

10. Kerr A.R. A technique for determining the local oscillator waveforms in microwave mixer // IEEE Trans. on Microwave Theory and Techniques. – 1975. – Vol. MTT-23. – No 10. – P. 828-831.

11. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. – М.: Мир, 1975.

12. Saad Y. Iterative Methods for Sparse Linear Systems. – PWS Publishing Company, 1996. – P. 447.

13. Feldmann P., Melville B., Long D. Efficient frequency domain analysis of large nonlinear analog circuits // in Proc. IEEE Custom Integrated Circuits Conf., San Diego. – 1996. – P. 461-464.

14. Telichevesky R., Kundert K., White J. Efficient Steady-State Analysis based on Matrix-Free Krylov-Subspace Methods // in Proc. of 32 IEEE Design Automation Conf., San Francisco. – 1995. – P. 480-484.

15. Freund R., Golub G.H., Nachtigal N.M. Iterative solution of linear systems // Acta Numerica. – 1991. – P. 57-100.

16. Gourary M. Iterative Solution of Linear Systems in Harmonic Balance Analysis / M. Gourary, S. Rusakov, S. Ulyanov, M. Zharov, et al. // IEEE MTT-S Int. Microwave Symposium Digest, Denver. – 1997. – P. 1507-1510.

17. Gourary M. M. Adaptive Preconditioners for the Simulation of Extremely Nonlinear Circuits Using Har-

monic Balance / M. M. Gourary, S. G. Rusakov, S. L. Ulyanov, M. M. Zharov, et al. // IEEE MTT-S Int. Microwave Symposium Digest, Anaheim. – 1999. – Vol. 2. – P. 779-782.

18. Gourary M. M. The Enhancing of Efficiency of the Harmonic Balance Analysis by Adaptation of Preconditioner to Circuit Nonlinearity / M. M. Gourary, S. G. Rusakov, S. L. Ulyanov, M. M. Zharov et al // Proc. Asia and South Pacific Design Automation Conf., Yokohama. – 2000. – P. 537-540.

19. Veerse F. Efficient iterative time preconditioners for harmonic balance RF circuit simulation // in Proc. Int. Conf. Computer-Aided Design, San Jose. – 2003. – P. 251-254.

20. Gourary M. M. New methods for speeding up computation of Newton updates in harmonic balance / M.M. Gourary, S.G. Rusakov, S.L. Ulyanov, M.M. Zharov, et al. // Proc. IEEE Int. Conf. on Computer-Aided Design, San Jose. – 1999. – P. 61-64.

21. Гурарий М.М. Адаптивный метод гармонического баланса / М.М. Гурарий, М.М. Жаров, С.Г. Русаков, С.Л. Ульянов // Проблемы разработки перспективных микроэлектронных систем. Сб. трудов/под ред. акад. РАН А.Л. Стемповского. – М.: ИПИМ РАН, 2005. – С. 13-18.

22. Gourary M.M. Numerical Aspects of Adaptive Harmonic Balance Method in Circuit Simulation / M.M. Gourary, S.G. Rusakov, A.L. Stempkovsky, S.L. Ulyanov, M.M. Zharov // Zbornik radova konferencije MIT 2009. – Novosibirsk: Institute of Computational Technologies. – 2010. – P. 155-162.

23. Rizzoli V., Costanzo A., Neri A. Harmonic-balance analysis of microwave oscillators with automatic suppression of degenerate solution // Electronics Letters. – 1992. – Vol. 28. – No 3. – P. 256-257.

24. Ngoya E. Steady-state analysis of free or forced oscillators by harmonic balance and stability investigation of periodic and quasi-periodic regimes / E. Ngoya, A. Suarez, R. Sommet, R. Quer // Int. J. of Microwave and Millimeter-Wave Comp.-Aided Eng. – 1995. – Vol. 5. – No 3. – P. 210-223.

25. Chang C-R, Steer M. B. Computer-Aided analysis of free-running microwave oscillators // IEEE Trans. on Microwave Theory and Techniques. – 1991. – Vol. 39. – No 10. – P. 1735-1745.

26. Lampe S., Laur R. Global optimization applied to the oscillator problems // in Proc. of Design Automation and Test in Europe Conf., Paris. – 2002. – P. 322-326.

27. Duan X., Mayaram K. Frequency domain simulation of high-Q oscillators with homotopy methods // in Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design, San Jose. – 2004. – P. 683-686.

28. Boianapally K., Mei T., Roychowdhury J. A multi-harmonic probe technique for computing oscillator steady states // in Proc. IEEE/ICM Int. Conf. Comput.-Aided Design, San Jose. – 2005. – P. 609-612.

29. Brambilla A., Gruosso G., Gajani G. Robust harmonic-probe method for the simulation of oscillators // IEEE Trans. on Circuits and Systems I: Regular Papers. – 2010. – Vol. 57. – No. 9. – P. 2531-2541.

30. Gourary M. Simulation of high-Q oscillators /

M. Gourary, S. Ulyanov, M. Zharov, S. Rusakov, et al. // in Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design, San Jose. – 1998. - P. 162-169.

31. *Gourary M.* A robust and efficient oscillator analysis technique using harmonic balance / M. Gourary, S. Ulyanov, M. Zharov, S. Rusakov, et al. // Computer Methods in Applied Mechanics and Engineering. – 2000. - Vol. 181. - P. 451-466.

32. *Gourary M.* Frequency adjusting numerical technique for oscillator simulation / M. Gourary, S. Rusakov, S. Ulyanov, M. Zharov, et al. // Proc. European Conf. on Circuit Theory and Design, Cork, Ireland. – 2005. - Vol. 3. - P. 121-124.

33. *Gourary M.M.* Frequency Domain Techniques for Simulation of Oscillators / M.M. Gourary, S.G. Rusakov, A.L. Stempkovsky, S.L. Ulyanov, M.M. Zharov // Proc. of IEEE East-West Design & Test Symposium (EWDTS'09). Moscow, Russia. – 2009. - P. 63-66.

34. *Kundert K.S.* Introduction to RF Simulation and Its Application // J. of Solid-State Circuits. - 1999. - Vol. 34. - No 9. - P. 1298-1319.

35. *Leeson D.* A simple model of feedback oscillator

noise spectrum // Proc. IEEE. - 1966. - Vol. 54. - P. 329-330.

36. *Kaertner F.X.* Determination of the correlation spectrum of oscillators with low noise // IEEE Trans. Microwave Theory Tech. - 1989. - Vol. 37. - P. 90-101.

37. *Kaertner F.X.* Analysis of white and f-a noise in electrical oscillators // Int. J. Circ. Theory Appl. - 1990. - Vol. 18. - P. 485-519.

38. *Demir A., Mehrotra A., Roychowdhury J.* Phase Noise in Oscillators: A Unifying Theory and Numerical Methods for Characterization // IEEE Trans. on Circuits and Systems - I. - 2000. - Vol. 47. - P. 655-674.

39. *Demir A.* Phase Noise and Timing Jitter in Oscillators with Colored-Noise Sources. // IEEE Trans. on Circuits and Systems - I. - 2002. - Vol. 49. - No 12. - P. 1782-1791.

40. *Gourary M. M.* New Numerical Technique for Cyclostationary Noise Analysis of Oscillators / M.M. Gourary, S.G. Rusakov, S.L. Ulyanov, M.M. Zharov et al // Proc. of the 37th European Microwave Conf., Munich. – 2007. - P. 1173-1176.

УДК 004.021

*С.Н. Кириллов, Л.С. Крупнов*

## ЭНТРОПИЯ ПАРОЛЕЙ КАК МЕРА ОЦЕНКИ СТОЙКОСТИ К МАШИННОМУ ПЕРЕБОРУ

*Рассматривается задача оценки стойкости паролей к машинному перебору. В качестве меры оценки стойкости обосновано применение энтропии. Приводится описание алгоритма оценки стойкости паролей, основанного на вычислении суммарной энтропии типовых составных последовательностей, позволяющего дать количественную оценку стойкости пароля. На основе анализа работы полученного алгоритма даны рекомендации по правилам выбора наиболее стойких паролей.*

**Ключевые слова:** алгоритм оценки стойкости пароля, энтропия, машинный перебор.

**Введение.** Широкое распространение многопарольных паролей в качестве единственного способа аутентификации для большинства информационных сервисов вызывает интерес со стороны злоумышленников к атакам машинного перебора, направленным на получение несанкционированного доступа. Уязвимость данного способа аутентификации подтверждается последними исследованиями взломанных баз данных с паролями от крупных международных компаний, среди которых Yahoo, LinkedIn, eHarmony, и Last.fm [1]. Проведенные исследования говорят о том, что пользователи зачастую выбирают слишком простые и легко угадываемые пароли. Отсюда вытекает задача генерирования стойких к машинному перебору паролей,

обладающих при этом хорошей запоминаемостью для человека.

Обзор существующих алгоритмов оценки стойкости пароля (АОСП) дает противоречивые качественные результаты (таблица 1). В данном исследовании сравнивается работа АОСП от крупнейших телекоммуникационных сервисов на ноябрь 2014 года. Пароли при этом выбирались на основе следующих правил [2].

1. qwer43@! – представляет собой простую клавиатурную последовательность (порядок символов определяется расположением клавиш на клавиатуре), слегка видоизмененную верхним регистром последних двух символов.

2. SUpErM4n\$78 – представляет собой словарное слово с добавлением трех случайных

символов в конце пароля. Некоторые символы при этом заменены стандартными подстановками ( $u \rightarrow U, A \rightarrow 4$ ).

3. starresistmermaidwar – набор из четырех случайных словарных слов без пробелов.

**Таблица 1** — Результаты работы различных АОСП

Пароль \ Сервис	qwer43@!	SUper4n\$78	starresistmermaidwar
Gmail	Надежный	Надежный	Надежный
Dropbox	Слабый	Посредственный	Отличный
Twitter	Отличный	Отличный	Хороший
Yandex	Надежный	Надежный	Надежный
Mail.ru	Сильный	Сильный	Сильный
Qip.ru	Средний	Нормальный	Слабый

Как следует из анализа таблицы 1, рассматриваемые сервисы по-разному качественно оценивают стойкость одних и тех же паролей. При этом отсутствует единая шкала измерений, а оценка стойкости носит условный характер. Это говорит о необходимости обоснования единой количественной меры стойкости пароля, обеспечивающей адекватную ее оценку.

*Цель работы* – обосновать эффективный АОСП к машинному перебору и разработать на его основе рекомендации по выбору наиболее стойких паролей.

**Теоретическая часть.** Формально процесс подбора пароля это угадывание случайной последовательности символов методом перебора всех возможных вариантов. В терминах теории вероятности каждый отдельный символ (или группа символов) в пароле будет являться случайным событием с определенной вероятностью появления  $p(i)$ , где  $i$  – порядковый номер символа. При этом весь пароль в целом можно рассматривать как единый исход, состоящий из определенного количества случайных событий. Число всевозможных исходов и будет определять количество различных вариантов паролей, подлежащих перебору в пространстве событий (ПС)  $\Omega$  [3].

В качестве меры стойкости пароля примем

среднее количество итераций перебора, при котором злоумышленник с вероятностью  $P$  угадает взламываемый пароль. При фиксированном времени проверки одного пароля  $t_0$  данная величина легко преобразуется в среднее время  $T_p$ , необходимое злоумышленнику на взлом с вероятностью успеха, равной  $P$ .

В терминах теории информации ПС  $\Omega$  может быть охарактеризовано такой мерой неопределенности, как энтропия. В этом случае энтропия будет описывать количество информации, содержащейся в подбираемом пароле, в случае равномерного распределения вероятностей исходов [4]:

$$H_0 = \log_2 \Omega. \quad (1)$$

При этом можно легко оценить стойкость пароля, а следовательно, и среднее время, необходимое для его подбора:

$$T_p = P \cdot t_0 \cdot 2^{H_0}. \quad (2)$$

Формулы (1), (2) справедливы только в случае равновероятного появления каждой комбинации пароля из множества  $\Omega$ , что зачастую не имеет места на практике. Энтропию такого распределения будем называть максимальной и обозначать  $H_0$ .

При неравномерном распределении вероятностей энтропия снижается и изменяется сам алгоритм перебора паролей. В данном случае злоумышленник перебирает пароли в порядке убывания вероятности их появления, тем самым увеличивая свои шансы на успех и уменьшая среднее время подбора пароля. Для обоснования энтропии в качестве меры стойкости пароля необходимо оценить ее влияние на вероятность  $P$  и среднее время  $T_p$  успешного перебора.

Произведено имитационное моделирование процесса подбора пароля для различных ПС  $\Omega$ . В каждом случае вычислялась энтропия  $H$  ПС и производился последовательный перебор исходов в порядке уменьшения вероятности их появления. Количество итераций перебора при этом фиксировано и равно  $2^H$ . В качестве выходных данных выступала сумма всех перебранных исходов, которая определяла вероятность успешного подбора пароля.

В исследовании использовались ПС с энтропией от 4 до 32 бит и с различным соотношением  $H/H_0$ . Для каждого значения энтропии  $H$  была сгенерирована выборка из не менее 100 ПС. При этом каждое ПС имело собственный закон распределения вероятностей исходов. Суммарно в исследовании принимало участие 128 выборок ПС. Для примера на рисунке 1 приведены результаты моделирования для одной выборки ПС, обладающих значениями энтропии  $H=8$ , где  $i$  –



номер отдельного ПС в выборке.

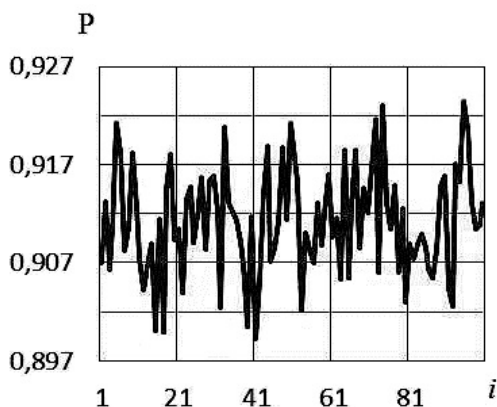


Рисунок 1 – Вероятность успешного подбора пароля при  $H=8$

Среднеквадратическое отклонение полученной величины составляет 0,0054, что говорит об устойчивости итоговой вероятности успешного подбора пароля  $P$  для ПС с близкими значениями энтропии  $H$ . Подобное поведение прослеживалось на всех исследованных выборках, при этом средние значения вероятности  $P$  локализованы в узком интервале от 0,86 до 0,95.

Проведенные исследования подтверждают прямую зависимость среднего времени взлома пароля  $T$  от его энтропии  $H$ , что позволяет использовать энтропию в качестве меры стойкости пароля к машинному перебору.

**Оценка значения энтропии пароля.** Для вычисления энтропии разобьем пароль на непересекающиеся последовательности, обладающие минимальной энтропией. В качестве такой последовательности могут выступать отдельные символы или их комбинации, слова, числа, даты и т.п. Чем больше последовательностей мы сможем выделить в пароле, тем точнее удастся оценить его энтропию. Каждую отдельную последовательность будем рассматривать как отдельное случайное событие  $x_k$  с собственным распределением вероятностей и энтропией  $H(x_k)$ .

Для получения общей энтропии пароля необходимо учитывать зависимость вероятности появления одних последовательностей при наличии других. Согласно [5] для двух зависимых событий  $x_1$  и  $x_2$  можно представить вероятность составного события в виде  $p(i,j) = p_i p_i(j)$ , где  $p_i(j)$  – условная вероятность появления события  $x_2=j$  при уже совершенном событии  $x_1=i$ ,  $p_i$  – априорная вероятность появления события  $x_1=i$ . Формула суммарной энтропии  $H(x_1, x_2)$  при этом будет иметь вид:

$$\begin{aligned} H(x_1, x_2) &= H(x_1) + H(x_2) - H(x_1 x_2) = \\ &= H(x_1) + H(x_2 | x_1) \end{aligned} \quad (3)$$

здесь  $H(x_1)$  и  $H(x_2)$  – энтропии событий  $x_1$  и  $x_2$

соответственно,  $H(x_2|x_1)$  – энтропия события  $x_2$  при состоявшемся событии  $x_1$ . Энтропия  $H(x_1 x_2)$  при этом определяет среднюю неопределенность события  $x_2$  после наблюдения исхода события  $x_1$ . Физический смысл энтропии  $H(x_1 x_2)$  можно определить как оценку взаимосвязи между событиями  $x_1$  и  $x_2$ . Таким образом, чем больше взаимосвязей между составными случайными событиями можно выявить, тем меньше будет итоговая энтропия составного события.

В более развернутом виде, согласно формуле Шеннона, дифференциальная энтропия может быть выражена следующим образом [4]:

$$H(x_1, x_2) = -\sum_{i,j} p(i,j) [\log_2 p_i + \log_2 p_i(j)]. \quad (4)$$

Данную формулу можно расширить на неограниченное число непересекающихся последовательностей  $x_1 \dots x_n$  в виде:

$$\begin{aligned} H(x_1, x_2, \dots, x_n) &= H(x_1) + H(x_2 | x_1) + \\ &+ \dots + H(x_n | x_1 \dots x_{n-1}). \end{aligned} \quad (5)$$

Необходимо заметить, что на практике достаточно сложно получить условные вероятности для всех последовательностей пароля. Однако доля взаимной информации для ансамблей из трех и более событий, как правило, пренебрежительно мала, что позволяет не учитывать ее без серьезной потери в точности оценки энтропии. Представив общую формулу для расчета энтропии, перейдем к частным формулам для отдельных классов последовательностей.

**Пароли в виде символьной последовательности.** Если рассматривать каждый символ в пароле как отдельное случайное событие, то алфавит будет представлять собой ПС с вероятностью появления каждого символа, равной  $p(i)$ . В общем случае энтропия каждого символа рассчитывается с помощью классической формулы Шеннона [4]:

$$H = -\sum_{i=1}^N p_i \log_2 p_i. \quad (6)$$

Последовательность  $L$  символов из алфавита емкостью  $n$ , полученная с помощью машинного генератора случайных чисел, будет обладать при этом максимально возможной энтропией  $H_0$ , ввиду равной вероятности появления для каждого события (символа), и полной независимости событий друг от друга:

$$p_i = 1/n^L, \quad a \quad H = H_0 = L \cdot \log_2 n.$$

На месте каждого символа может быть один из  $n$  различных вариантов. При этом  $n$  – это емкость алфавита, в который могут входить как символы различных языков, так и цифры, знаки препинания и т.п.

По вышеприведенной формуле можно рассчитать энтропию для тех последовательностей, которые не были отнесены к какому-либо другому классу.

Однако, если последовательность из того же самого алфавита сгенерирована человеком, то вероятность появления каждого из символов будет в общем случае отличаться от  $1/n$ , поскольку часто в паролях используются одни и те же символы и цифры. Результирующая энтропия, рассчитанная по формуле (6), будет меньше верхней границы энтропии  $H_0$ , так как на выбор события наложено дополнительное условие в виде его априорной вероятности появления. Данную ситуацию можно проиллюстрировать на примере пароля, состоящего из символов английского алфавита. В таблице 2 приведены априорные вероятности появления каждого из символов [6].

Таблица 2 – Вероятность появления каждого символа английского алфавита

Символ	Вероятность	Символ	Вероятность
Пробел	0,2	L	0,029
E	0,105	C	0,023
T	0,072	F,U	0,0225
O	0,0654	M	0,021
A	0,063	P	0,0175
N	0,059	Y,W	0,012
I	0,055	G	0,011
R	0,054	B	0,0105
S	0,052	V	0,008
H	0,047	K	0,003
D	0,035	X	0,002
		J,Q,Z	0,001

В этом случае для пароля, состоящего из одного символа, значения средней и максимальной энтропии будут равны:

$$H = 4,03 \text{ бит}, H_0 = 4,76 \text{ бит}.$$

При количестве символов в пароле больше одного, средняя энтропия  $H$  на каждый символ будет неизбежно снижаться за счет влияния условной вероятности. Чтобы оценить предельные значения  $H$ , в [4] были проведены исследования, связанные с оценкой избыточности английского языка. В ходе эксперимента испытуемым требовалось угадать зашифрованное слово из  $l$  букв, имея возможность делать предположения по каждому символу и получать утвердительный или отрицательный ответ. На основе общего количества попыток вычислялась энтропия слова, а затем и значение энтропии на один символ. Результаты исследований приведены на рисунке 2 [4].

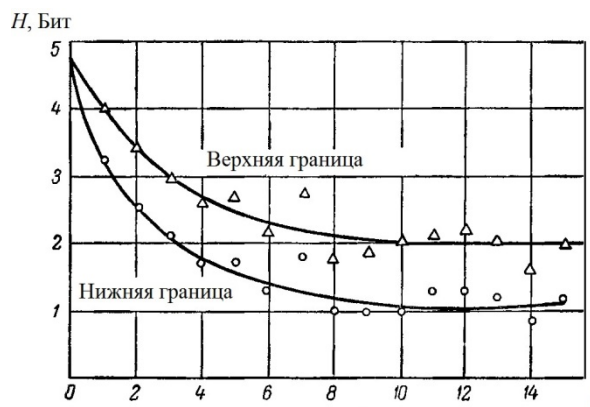


Рисунок 2 – Количество бит энтропии на букву от числа букв

Данные зависимости позволяют оценить взаимное влияние условных вероятностей на энтропию пароля, состоящего из английских букв. Основное снижение энтропии (2-3 раза) приходится на первые 3 символа, в то время как дальнейший учет условных вероятностей существенного влияния на энтропию не оказывает.

#### Пароли в виде словарных слов и фраз.

Если частью пароля является слово, то оно целиком рассматривается как отдельное случайное событие. При этом полное ПС  $\Omega$  определяется словарем  $V$  [7]. С точки зрения злоумышленника, перебор слов необходимо производить в порядке уменьшения вероятности их появления. Для вычисления энтропии в данном случае важно правильно определить размер словаря. Поскольку перебор заканчивается на проверяемом слове, то его позиция  $R_i$  будет определять размер словаря. Итоговое значение пространства событий  $V$  будет равняться максимальной позиции  $R_i$  слова, обладающего такой же вероятностью появления  $p_i$ , что и проверяемое слово. Энтропия при этом вычисляется следующим образом:

$$H = -\sum_{i=1}^V p_i' \log_2 p_i', \quad V = \max(R_j | \forall j, p_j \geq p_i),$$

$$p_i' = p_i / \sum_{i=1}^V p_i.$$

При наличии дополнительной априорной информации о предполагаемом слове можно использовать специфические словари технических терминов, имен, фамилий, календарных дат и т.п.

Согласно формулам (5), (6), фразы, составленные из отдельных слов, накладывают дополнительные условия на состав пароля. В данном случае это можно учесть, рассматривая вероятности появления одних частей речи после других, как, например прилагательное после существительного (таблица 3):

Таблица 3 – Энтропия слов и словосочетаний с учетом условных вероятностей появления отдельных частей речи

Пароль	Энтропия, бит
зеленый	5,044
стол	9,573
зеленыйстол	12,718
столзеленый	13,907

Из анализа таблицы видно, что пароль «зеленыйстол» обладает меньшим значением энтропии, нежели «столзеленый», так как шаблон «прилагательное-существительное» в речи встречается чаще, чем «существительное-прилагательное». При этом энтропия каждого пароля меньше суммы энтропий, входящих в его состав слов, ввиду влияния условных вероятностей.

**Пароли в виде клавиатурных последовательностей** характеризуются набором символов, набранных подряд на клавиатуре. Для учета всех возможных комбинаций потребуются следующие параметры последовательности:

$L$  – длина пароля,  $L \geq 2$ ;

$s$  – число возможных начальных символов;

$f$  – число смен направлений,  $f \leq L-1$ ;

$q$  – средний «порядок» каждой клавиши, равный числу смежных с ней клавиш.

В этом случае полное ПС будет описываться формулой [2]

$$\Omega = \left( \sum_{i=2}^L \sum_{j=1}^{\min f, i-1} \binom{i-1}{j-1} s q^j \right), \text{ а энтропия определяется следующим выражением:}$$

$$H = \log_2 \left( \sum_{i=2}^L \sum_{j=1}^{\min f, i-1} \binom{i-1}{j-1} s q^j \right).$$

Подсчет всех возможных конфигураций смены направлений для клавиатурной комбинации длины  $i$ , имеющей  $j$  смен направления, производится с помощью функции «биномиальный коэффициент из  $(i-1)$  по  $(j-1)$ ». Поскольку первая смена направления всегда происходит на первой букве, к обоим элементам добавляется значение «-1». При каждой из  $j$  смен направлений существует  $q$  возможных вариантов продолжения последовательности, т. е. всего  $q^j$  возможностей на каждую конфигурацию. Злоумышленник также должен перебрать все начальные символы, число которых равно  $s$ . Данная формула учитывает ситуацию, когда последовательность приближается к краю клавиатуры. При этом подразумевается, что следующий символ будет набран с противоположного края клавиатуры.

Проанализировав различные варианты кла-

виатурных последовательностей, можно сделать вывод, что длинные участки без поворотов увеличивают энтропию пароля в среднем на 0,3 бита за каждый последующий символ, в то время как каждый символ, изменяющий направления последовательности, добавляет в среднем 2-3 бита энтропии.

**Пароли с модификаторами последовательностей.** Каждая последовательность может быть видоизменена с помощью следующих простых модификаторов (таблица 4).

Таблица 4 – Добавочная энтропия для простых модификаторов пароля

Модификатор	Величина добавочной энтропии
Первая буква слова – заглавная	1 бит
Перевод раскладки клавиатуры на противоположную при наборе пароля (в случае существования 2х основных раскладок)	1 бит
Транслитерация (при условии наличия однозначного соответствия исходного алфавита алфавиту транслитерации)	1 бит
Ввод слова задом наперед	1 бит
Инверсия строчных и прописных букв (режим CapsLock)	1 бит

Поскольку для проверки наличия отдельно каждого модификатора требуется одна дополнительная итерация, вся энтропия пароля увеличивается всего на 1 бит.

Существуют и более сложные модификаторы, как, например изменение определенных символов на мнемонические подстановки (например, 0→O, A→4), либо смена регистра нескольких символов на противоположный. При числе заглавных букв (замененных символов), равном  $U$ , и числе строчных букв (незамененных символов), равном  $u$ , нам потребуется перебрать все возможные комбинации расположений указанных замен в пароле. Формула добавочной энтропии при этом будет иметь вид [2]:

$$H = \log_2 \left( \sum_{i=1}^{\min(U,u)} \binom{U+u}{i} \right).$$

В качестве примера был рассмотрен пароль из 10 символов, измененный с помощью сложного модификатора (рисунок 3).

Количество добавочной энтропии при этом максимально, если замене подлежит около половины символов пароля. Это связано с тем, что если замене подверглось больше половины символов, с точки зрения минимума энтропии весь

пароль нужно рассматривать как подвергнутый сразу двум модификаторам:

- 1) инверсия строчных и прописных букв;
- 2) сложный модификатор замены прописных букв на строчные.

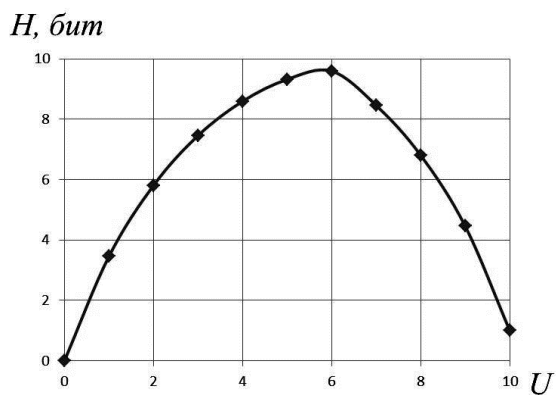


Рисунок 3 – Зависимость добавочной энтропии от количества заглавных букв в 10-символьном пароле

**Экспериментальная часть.** Согласно приведенным выше формулам, была рассчитана энтропия трех типовых паролей, предоставленных ранее (таблица 1). Результаты приведены в таблице 5.

Таблица 5 – Энтропия типовых паролей

Пароль	Энтропия, бит
qwer43@!	23,155
SUperm4n\$78	29,96
starresistmermaidwar	39,237

Полученные значения позволяют более точно оценить стойкость рассматриваемых паролей по сравнению с данными таблицы 1. При этом стойкость паролей оценивается количественной, а не качественной характеристикой.

С точки зрения запоминаемости были рассмотрены наиболее популярные шаблоны генерации пароля и их модификации, а также получены их значения энтропии (таблица 6).

На основе данных исследований предложена следующая процедура генерации паролей, стойких к машинному перебору и обладающих хорошей запоминаемостью.

1. В качестве основы выбирается последовательность, изначально обладающая высокой энтропией. Случайно сгенерированные последовательности подойдут при ограничениях на длину пароля, в то время как комбинации из 3х и более независимых словарных слов будут обладать хорошей запоминаемостью и высокой энтропией.

2. Последовательность подвергается воздействию модификаторов. Рекомендуется использовать сложные модификаторы вместо комбинации простых. В случае замены регистра букв

наиболее эффективно будет заменить около половины символов пароля, используя для запоминания измененных позиций численный или визуальный шаблон.

3. Вычисляется энтропия полученного пароля с помощью машинной реализации рассматриваемого АОСП. При необходимости пароль подвергается дополнительной модификации (при недостатке 2-5 бит энтропии) или изменяется его основа (при недостатке 6 и более бит энтропии).

Таблица 6 – Оценка сложности запоминания распространенных шаблонов паролей

Описание шаблона	Энтропия, бит	Эмпирическая оценка запоминаемости
Словарные слова	3-11	Простая
Слова с простыми модификаторами	4-12	Простая
Слова со сложными модификаторами	5-17	Сложная
Комбинации слов (2-4 слова)	21-38	Средняя
Комбинации слов с простыми модификаторами (2-4 слова)	22-39	Сложная
Комбинации слов со сложными модификаторами (2-4 слова)	28-42	Очень сложная
Случайные символьные последовательности (5-10 символов)	25-55	Очень сложная
Клавиатурные последовательности (5-10 символов)	18-25	Простая

**Заключение.** Обоснован АОСП на базе оценки энтропии, позволяющий определить стойкость пароля к машинному перебору с точки зрения вероятности угадывания целевого пароля. Это позволяет, с одной стороны, абстрагироваться от содержания пароля (отдельный символ, слово, дата и т.п.) и рассматривать только распределения вероятностей составляющих его случайных величин, что позволяет расширить применение АОСП на любое представление пароля. С другой стороны, АОСП позволяет оценить стойкость пароля с наиболее наглядной характеристикой – средним временем подбора (при условии использования злоумышленником максимально эффективного алгоритма перебора). На основе анализа работы АОСП предложена процедура получения паролей, обладающих

необходимой стойкостью к машинному перебору. Даны рекомендации по выбору наиболее простых для запоминания составных частей пароля, отвечающих при этом необходимым требованиям защищенности.

#### **Библиографический список**

1. Kevin Doel, Scary Logins: Worst Passwords of 2012 — and How to Fix Them. - Los Gatos: PRWeb, 2012г.

2. Dan Wheeler, Realistic password strength estimation [Электронный ресурс] // Технический блог Dropbox [Официальный сайт]. URL: <https://tech.dropbox.com/2012/04/zxcvbn-realistic-password-strength-estimation/> (дата обращения: 15.01.2015)

15.01.2015)

3. Севастьянов Б. А. Курс теории вероятностей и математической статистики. - М.: Наука, 1982. - 256 с.

4. Бриллюэн Л. Наука и теория информации. - М.: Физматгиз, 1960. - 392 с.

5. Феллер В. Введение в теорию вероятностей и ее приложения. - М.: Мир, 1964. - 498 с.

6. Галлагер Р. Теория информации и надёжная связь. - М.: Советское радио, 1974. - 720 с.

7. Ляшевская О.Н., Шаров С.А. Частотный словарь современного русского языка (на материалах Национального корпуса русского языка). - М.: Азбуковник, 2009. - 1087 с.

УДК 004.655.3

**А.М. Гостин, С.С. Косоруков**

## **ИСПОЛЬЗОВАНИЕ МЕТОДА СВЯЗЫВАНИЯ ДАННЫХ НА ОСНОВЕ УПРАВЛЯЮЩИХ ТЕГОВ ПРИ РАЗРАБОТКЕ СИСТЕМЫ СБОРА ПОКАЗАТЕЛЕЙ ДЕЯТЕЛЬНОСТИ ПРЕПОДАВАТЕЛЕЙ И СТУДЕНТОВ**

*Рассматривается использование метода связывания данных на основе управляющих тегов при разработке системы сбора показателей деятельности преподавателей и студентов. Метод позволяет гибко подходить к формированию необходимой выходной структуры данных, используя запросы к различным источникам, включая таблицы баз данных и XML файлы.*

**Ключевые слова:** управляющие теги, метод связывания данных.

**Введение.** Для функционирования системы сбора показателей деятельности преподавателей и студентов используются различные источники данных — таблицы баз данных, XML файлы, веб-сервисы. В условиях частого изменения правил расчета показателей требуется гибкое изменение структуры данных. Стандартом современных информационных систем является использование модели разделения данных и представлений, образуя таким образом промежуточный продуктовый слой. Это придает информационным системам необходимую гибкость и масштабируемость. Особенно это важно для приложений, обрабатывающих XML данные и использующих стандарт XML для хранения данных в файлах. Вместе с тем актуальным является получение и обработка данных, хранящихся в реляционных СУБД, доступ к которым осуществляется с помощью SQL запросов.

Основное преимущество стандарта XML — эффективность представления сложноструктурированных данных. Это свойство используется в различных прикладных стандартах, например,

в открытом формате документов (ODF), в формате электронных учебных курсов (SCORM), в формате представления контрольных измерительных материалов (IMS QTI). Недостаток XML формата — неэффективное использование системных ресурсов для адресации большого количества записей при организации их совместной обработки. Например, размер оперативной памяти при обработке XML документа в соответствии со стандартом XML DOM (объектная модель документа) до 10 раз превышает размер исходного XML файла с данными [1]. При возрастании нагрузки на сервер это становится критичным. Применение стандарта последовательного чтения XML файлов (Simple API for XML, SAX) позволяет избежать данных проблем, но в отличие от XML DOM при его использовании невозможно сохранять изменения в исходных документах [2].

Для формирования запросов к XML документам обычно используется стандарт XPath или более мощный стандарт запросов XQuery, позволяющий генерировать XML контент. Однако

проблема производительности при этом по-прежнему остается актуальной, также форма построения FLWOR запроса в соответствии со стандартом XQuery семантически сложна, поскольку смешивает управляющие конструкции For, Let, Where, Order by и Return, характерные для процедурных языков, с набором возвращаемых XML тегов [3].

Поскольку результат запроса представляет собой XML структуру, для вывода его содержимого целесообразно использовать XSLT преобразование, позволяющее отобразить результат в HTML формате для браузера пользователя.

Для формирования запросов к реляционным СУБД традиционно используется стандарт SQL. Основное преимущество реляционных СУБД — скорость и эффективность обработки данных, достигаемые за счет фиксированного размера логических записей [4]. Недостаток — сложность формирования запросов к связанным данным. Для запросов к связанным и иерархическим данным в стандарте SQL обычно используются левые объединения, группировки и индексирование. Синтаксис таких запросов становится громоздким, сложно читаемым, связанные данные и SQL запросы к ним трудно модифицировать. Кроме того, поскольку данные в реляционных СУБД обычно хранятся в плоских таблицах, представление сложных иерархических структур данных в них затруднено [5].

В настоящее время имеется ряд решений, позволяющих совмещать скорость и эффективность обработки SQL запросов с гибкостью обработки, присущей XML структурам, например, функциональный стандарт SQL/XML и сочетание его с XQuery [6]. Данное решение позволяет делать SQL запросы к полям таблиц, содержащим XML структуры на основе специальных функций типа XMLElement, расширяющих семантику языка SQL. Но при этом синтаксическая сложность полученных запросов заметно возрастает.

Нужно отметить, что распространенность XSLT стандарта и XSLT шаблонов в реальных проектах напрямую связано с использованием XML данных. Также XSLT стандарт не позволяет напрямую использовать в шаблонах пользовательские переменные и функции. Они должны быть предварительно инкапсулированы в XML структуре или определены в тегах отдельного пространства имен.

Таким образом, в ходе программной реализации системы сбора показателей деятельности преподавателей и студентов выявлены следующие задачи:

- необходимость сочетания гибкости XML

представления и скорости выполнения SQL запросов;

- необходимость интеграции данных, полученных из различных источников: XML файлов, таблиц СУБД, веб-сервисов и локальных переменных;

- возможность использования XSLT преобразования над результатами запросов для представления браузером содержимого в HTML формате.

**Целью работы** является разработка метода связывания данных на основе управляющих XML тегов (запрос queryXML) для решения вышеуказанных проблем.

**Теоретическая часть.** В запросах queryXML в рамках стандарта XML описывается не только структура результирующего XML документа (например, как в XML Schema), но могут определяться источники данных, а также сами данные с применением расширения стандарта XPath [7].

Запрос к данным представляет собой XML структуру queryXML, содержащую набор управляющих тегов — команд. Результатом запроса является сгенерированная XML структура, над которой впоследствии может быть применено XSLT преобразование.

Управляющий тег представляет собой XML узел со списком атрибутов, генерирующий XML содержимое. Каждый управляющий тег задает один из возможных источников данных: XML документ, результат удаленного запроса к веб-сервису, результат SQL запроса в виде XML-представления или XML переменную, что определяется расширенным XPath запросом.

В зависимости от генерируемого результата, возвращающего отдельный тег, набор тегов, атрибут или связывающую переменную, необходимо описать как минимум четыре вида управляющих тегов: element, elements, attribute и variable. Команды управления выполняются по функциональному принципу в порядке своего размещения в составе XML структуры запроса query. Если у команды имеются потомки, то результат их выполнения встраивается в общую структуру генерируемого содержимого. Переменные в атрибутах и содержании тегов интерполируются.

Структура запроса queryXML в расширенной форме Бэкуса-Наура имеет вид:

```
queryXML ::= EmptyQueryTag |
           SQueryTag content EQueryTag
EmptyQueryTag ::= SQueryTag '>'
SQueryTag ::= '<query' S AttrName
           (S AttrRoot)? (S AttrSelect)?
```

```

content ::= (element | elements | attribute | variable)*
EQueryTag ::= '</query>'
AttrName ::= 'name=' QQName
AttrRoot ::= 'root=' QQName
AttrSelect ::= 'select=' QExtXPath
QQName ::= "" QName ""
QName ::= [A-Za-z0-9]*
QExtXPath ::= "" ExtXPath ""

```

Управляющий тег `query` генерирует содержимое, заданное расширенным XPath выражением `ExtXPath`. Атрибут `root` задает имя корневого узла возвращаемой XML структуры. Если атрибут `root` не задан, корневой узел возвращаемой XML структуры получает имя `name`. Если атрибут `select` не задан, то генерируется пустой корневой узел.

Затем вложенные команды `content` генерируют свое содержимое, которое совмещается с результатом запроса `query`.

Расширенные запросы `ExtXPath` адресуют внешние источники данных: XML документы, внешние URL, внутренние переменные `Var`, а также SQL запросы, возвращающие данные в виде XML-представления:

```

ExtXPath ::= (Document | Url | Var)? XPath | SQLExpr
Document ::= 'document(" FileName "')
Url ::= 'url(" URLString "')
Var ::= '$' QName
SQLExpr ::= 'sql(" DBName "') SQLSelect

```

Лексема `FileName` определяет имя XML файла, `URLString` — URL адрес веб-сервиса, `Var` — имя глобальной XML переменной, `SQLSelect` — строка SQL запроса к базе данных `DBName`.

Функция `sql()` расширенного запроса `ExtXPath` возвращает результат в виде XML-представления `SQLRes`, соответствующего выбранным записям таблицы `TableName` базы данных:

```

SQLRes ::= '<' TableName '/>' | '<' TableName '>' S ('<row' (FieldsList)* '/>')* S '<' TableName '>'
FieldsList ::= (S QName '=' Value)*

```

Команды запроса, определяемые лексемой `content`, выполняются в процедурном стиле в порядке размещения в структуре запроса. Если у XML узла команды имеются потомки, то результат их выполнения встраивается в общую структуру генерируемого содержимого. Внутри команд можно использовать относительную адресацию XPath (по аналогии с XSLT шаблонами). Переменные, встречаемые в атрибутах и содержании тегов, интерполируются.

Команда `element` обеспечивает генерацию или переопределение элемента XML:

```

element ::= EmptyElementTag | SElementTag content EElementTag
EmptyElementTag ::= SElementTag '/>'
SElementTag ::= '<element' S AttrName (S AttrSelect)? (S AttrValue)? (S AttrFields)?
EElementTag ::= '</element>'
AttrName ::= 'name=' QQName
AttrSelect ::= 'select=' QExtXPath
AttrValue ::= 'value=' QQName
AttrFields ::= 'fields=' QFields
QFields ::= "" Field (',' Field)* ""
Field ::= QName | '@' QName

```

Список атрибутов тега `element`: `name` — имя элемента (обязательный атрибут), `select` — источник запроса, `value` — значение XML узла, `fields` — список полей (фильтр).

Команда генерирует или переопределяет XML элемент, заданный выражением `select` или атрибутом `value`. Возвращаемый корневой узел XML структуры получает имя, указанное атрибутом `name`. Если узел с таким именем уже имеется на данном уровне сгенерированной XML структуры, то он переопределяется (перегружается). Если такого узла нет, то он создается. Если на данном уровне сгенерированной структуры имеется несколько указанных элементов, то для переопределения выбирается первый из них. Переменные в атрибутах команды интерполируются.

Если задан атрибут `value`, то XML элемент инициализируется вычисляемым или текстовым значением `value`. Если атрибуты `select` и `value` не заданы и у данного тега отсутствуют потомки, то элемент с таким именем удаляется из сгенерированной XML структуры.

Если задан атрибут `fields`, то XML структура инициализируется только списком указанных полей (потомков) или атрибутов данного элемента. Элементы полей или атрибутов в списке разделяются запятой. Если атрибут `fields` не задан, фильтр имеет значение `*` (все поля по умолчанию).

Команда `elements` обеспечивает генерацию или переопределение массива элементов XML:

```

elements ::= EmptyElementsTag | SElementsTag content EElementsTag
EmptyElementsTag ::= SElementsTag '/>'
SElementsTag ::= '<elements' S AttrName (S AttrSelect)? (S AttrFields)? (S AttrFilter)? (S AttrLimit)? (S AttrPage)?
EElementsTag ::= '</elements>'
AttrName ::= 'name=' QQName
AttrSelect ::= 'select=' QExtXPath

```

```

AttrFilter ::= 'filter=' QInVar
AttrLimit ::= 'limit=' QNum
AttrPage ::= 'page=' QNum
AttrFields ::= 'fields=' QFields
QFields ::= "" Field (',' Field)* ""
Field ::= QName | '@' QName
QInVar ::= "" InVar (',' InVar)* ""
InVar ::= '$in{' QName '}'
QNum ::= Var | InVar | [0-9]*

```

Список атрибутов тега `elements`: `name` — имя элементов (обязательный атрибут), `select` — источник запроса, `filter` — список входных переменных (фильтр), `limit` — максимальное количество возвращаемых элементов, `page` — номер страницы списка элементов, `fields` — фильтр полей.

Команда генерирует или переопределяет XML элементы, заданные выражением `select`. Возвращаемые узлы XML структуры получают имя, заданное атрибутом `name`.

Если узлы с таким именем уже имеются на данном уровне сгенерированной XML структуры, то они переопределяются (перегружаются). Если таких узлов нет, то они создаются. Если атрибут `select` не задан, а также у данного тега отсутствуют потомки, то элементы с таким именем удаляются из сгенерированной XML структуры. Переменные в атрибутах команды интерполируются.

Если задан атрибут `fields`, то XML структура инициализируется только списком указанных полей (потомков) и атрибутов данных элементов. Элементы полей и атрибутов в списке разделяются запятой. Если атрибут `fields` не задан, фильтр имеет значение `*` (все поля по умолчанию).

Если задан атрибут `limit`, то XML структура инициализируется указанным количеством элементов, полученных в ходе запроса `ExtXPath`. Если задан атрибут `page`, то отсчет элементов начинается с порядкового номера, вычисляемого по формуле  $(page - 1) * limit$ .

Атрибут `filter` используется совместно с выражениями конкретизации в расширенном запросе `ExtXPath` и функцией `sql()`. Он содержит список `http` переменных `QInVar`, разделенных запятой. Если задан атрибут `filter`, то при интерполяции строки расширенного запроса `ExtXPath` в атрибуте `select` все неопределенные входные переменные `InVar` заменяются на символ `*` (любое значение).

Команда `attribute` обеспечивает генерацию или переопределение атрибута узла XML:

```

attribute ::= EmptyAttrTag |
           SAttrTag Cont EAttrTag

```

```

EmptyAttrTag ::= SAttrTag '/>'
SAttrTag ::= '<attribute' S AttrName
           (S AttrSelect)? (S AttrValue)?
EAttrTag ::= '</attribute>'
Cont ::= [^<>]*

```

Список атрибутов управляющего тега `attribute`: `name` — имя атрибута (обязательный атрибут), `select` — источник запроса, `value` — значение атрибута, `content` — текстовое содержание атрибута.

Команда генерирует или переопределяет атрибут XML элемента, заданный выражением `select` или атрибутом `value`. Возвращаемый атрибут получает имя `name`.

Если атрибут с таким именем уже имеется у родительского элемента `element` сгенерированной XML структуры, то он переопределяется (перегружается). Если такого атрибута нет, то он создается. Если задан атрибут `value`, то атрибут инициализируется вычисляемым или строковым значением `value`. Переменные в атрибутах команды интерполируются.

Если атрибуты `select` и `value` не заданы, то атрибут с таким именем удаляется из сгенерированного XML узла.

Команда `variable` обеспечивает инициализацию XML переменной:

```

variable ::= EmptyVarTag |
           SVarTag Cont EVarTag
EmptyVarTag ::= SVarTag '/>'
SVarTag ::= '<variable' S AttrName
           (S AttrSelect)? (S AttrValue)?
EVarTag ::= '</variable>'

```

Список атрибутов управляющего тега `variable`: `name` — имя переменной, `select` — источник запроса, `value` — значение переменной, `content` — текстовое содержание переменной.

Команда инициализирует переменную, заданную выражением `select`, атрибутом `value` или его текстовым содержанием. Возвращаемая переменная получает имя `name`. Переменные могут использоваться для конкретизации вложенных запросов.

Если задан атрибут `value`, то переменная инициализируется вычисляемым значением `value`. Переменные в атрибутах команды интерполируются.

**Экспериментальные исследования.** Метод связывания данных на основе управляющих тегов был успешно апробирован при разработке системы сбора показателей деятельности преподавателей в РГРТУ (имена изменены).

Например, в таблице `users` базы данных `dblib` указаны данные преподавателей:



uid	surname	name	patronymic
118	Иванов	Иван	Иванович
...	...	...	...
503	Петров	Петр	Петрович

В XML файле publications.xml размещен список публикаций:

```

<publications>
  <publication id="123">
    <title>Координатная привязка изображений от геостационарных спутников по контурным точкам диска Земли</title>
    <authors>
      <author uid="118"/>
      <author uid="503"/>
    </authors>
    <labels>
      <label name="wscience">0</label>
      <label name="scopus">0</label>
      <label name="rinc">1</label>
      <label name="vak">1</label>
    </labels>
  </publication>
  .....
</publications>

```

Тогда запрос queryXML на выборку списка публикаций и публикационной активности в журналах, индексируемых в системах Web of science, Scopus, РИНЦ и входящих в список ВАК, для преподавателя Иванова Ивана Ивановича может иметь следующий вид:

```

<query name="list">
  <variable name="pub"
    select="document('publications.xml')"/>
  <element name="user"
    select="database('dblib')/users[@uid='118']">
    <variable name="uid" select="@uid"/>
    <element name="publications">
      <elements name="publication"
        select="$pub/publication[//author[@uid='$uid']]
          /title"/>
    </element>
    <element name="pub_activity">
      <attribute name="wscience"
        select="sum($pub/publication[//author
          [@uid='$uid']]//label[@name='wscience'])"/>
      <attribute name="scopus"
        select="sum($pub/publication[//author
          [@uid='$uid']]//label[@name='scopus'])"/>
      <attribute name="rinc"
        select="sum($pub/publication[//author
          [@uid='$uid']]//label[@name='rinc'])"/>
      <attribute name="vak"
        select="sum($pub/publication[//author

```

```

          [@uid='$uid']]//label[@name='vak'])"/>
    </element>
  </element>
</query>

```

Соответственно результат запроса будет иметь следующий вид:

```

<list>
  <user uid="118" surname="Иванов"
    name="Иван" patronymic="Иванович">
    <publications>
      <publication>Координатная привязка изображений от геостационарных спутников по контурным точкам диска Земли</publication>
      .....
    </publications>
    <pub_activity wscience="0" scopus="0"
      rinc="1" vak="1"/>
  </user>
</list>

```

Результат запроса содержит данные преподавателя, список его научных трудов и публикационную активность. В ходе выполнения запроса документ publications.xml загружается в переменную \$pub и кэшируется в память, что позволяет эффективно использовать системные ресурсы сервера при расчете показателей публикационной активности. Для вывода результата запроса выполняется XSLT преобразование по заданному XSLT шаблону на стороне сервера, после чего ответ выводится в браузер пользователя в виде HTML страницы.

**Выводы.** Предложенный в статье метод связывания данных на основе управляющих тегов позволяет существенно упростить получение информации из различных источников данных, а главное, отделить запросы от программной реализации – все запросы размещены в отдельных редактируемых файлах и не связаны с программным кодом приложения [8].

Метод показал свою эффективность при создании следующих информационных систем и сервисов:

- система дистанционного тестирования «Академия»;
- страница «Вопросы-ответы» на Интернет портале РГРТУ;
- сайт службы трудоустройства выпускников РГРТУ;
- электронная библиотека РГРТУ;
- база данных студентов «Контингент»;
- система учета документов;
- система сбора показателей деятельности преподавателей и студентов «Личный кабинет преподавателя».

Все вышеуказанные системы написаны на интерпретируемом языке Perl 5.6 с использованием авторской библиотеки ESOP 2.0 и представляют собой веб-приложения, функционирующие под управлением веб-сервера Apache 2.2 и модуля mod\_perl 2.0. Для хранения данных используется СУБД MySQL 5.1 и хранилище XML файлов.

#### **Библиографический список**

1. Календарев А. XML-native DB. XQuery // Системный администратор — Вып. № 5(66). 2008.
2. Рэй Э., Дж. Макинтош. Perl & XML. Библиотека программиста. СПб.: Питер, 2003 – 218 с.
3. Плешачков П.О. Методы управления транзакциями в XML-ориентированных СУБД: дис. канд. физ.-мат. наук: Защищена 2006. – 166 с.
4. Баранчиков А.И., Баранчиков П.А. Организация доступа к записям таблиц в базах данных // Вестник Рязанского государственного

радиотехнического университета. 2007. № 20. С. 78 – 81.

5. Баранчиков А.И., Баранчиков П.А. Практическая реализация дискреционно-ролевого доступа на чтение к записям БД // Вестник Рязанского государственного радиотехнического университета. 2009. № 29. С. 65 – 69.

6. Matthias Nicola, Fatma Ozcan. PureXML in DB2 9: Which way to query your XML data? // IBM developerWorks. – August 2007.

7. Антипов В.А., Гузенко Р.Е. Трехуровневая метамодель отображения семантики предметной области на структуру XML сообщений // Вестник Рязанского государственного радиотехнического университета. 2010. № 31. С. 43 – 48.

8. Гостин А.М., Косоруков С.С. Использование программной библиотеки ESOP под управлением модуля mod\_perl 2.0: межвуз. сб. науч. трудов «Информационные технологии в образовании». Рязань: РГРТУ, 2012. – С. 41–43.