

## АДАПТИВНЫЙ ОТБОР ОБРАЗЦОВ ПРИ ИЗМЕРЕНИИ ПАРАМЕТРОВ ТРАФИКА КОМПЬЮТЕРНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ НЕЧЕТКОГО РЕГУЛЯТОРА И НЕЙРОННОЙ СЕТИ

**Ключевые слова:** *адаптивный отбор образцов, нечеткий регулятор, нейронная сеть, уровень качества сервиса.*

### ВВЕДЕНИЕ

В области измерения и оценки операционных параметров компьютерных сетей существует много различных инструментов, методов, рекомендаций и стандартов. Наиболее перспективными являются новые стандарты, которые создают рабочие группы IPFIX (IP Flow Information Export) [1] и PSAMP (Packet Sampling) [2]. Речь идет о неинтрузивных методах, которые для оценки операционных параметров используют существующий трафик компьютерной сети. Без генерированной тестирующей нагрузки можно получить более объективные результаты, которые лучше отражают состояние сети в ее реальном применении. Неинтрузивные методы можно использовать в различных областях [3] — для оптимизации работы сети, сбора информации о реализованном сетевом сервисе для пользователей, поддержки таких интерактивных приложений как, например, перенос голоса посредством протокола IP [4], обеспечение качества сервиса (QoS) [5], мобильные коммуникации [6].

Однако неинтрузивные методы требуют обработки большего количества данных, необходимых для отслеживания наблюдаемых параметров. В контексте спецификаций IPFIX и PSAMP можно рассмотреть три главных способа редукции объема обрабатываемых данных на уровне измерительного процесса.

1. Агрегация — это составление записей о потоках, которые передаются на последующую обработку без передачи информации о каждом подхваченном пакете. Записи о потоках, которые содержат их атрибуты, составляются с помощью заранее подготовленных шаблонов. Составить шаблоны можно таким образом, чтобы записи о потоках содержали только специфические атрибуты, которые требуются для измерений, что приведет к уменьшению объема передаваемых данных. Отдельные информационные элементы, используемые для создания шаблонов, рассмотрены в работе [7], а протокол для экспорта данных измерения — в работе [8].

2. Фильтрация — это процесс исключения обработки пакетов, не требующих измерения, что приведет к уменьшению объема передачи записей потоков.

3. Отбор — это процесс выбора наиболее репрезентативного образца из популяции пакетов, в результате чего происходит редукция объема обрабатываемых данных. Однако это может привести к уменьшению точности измерения. Целью отбора является достижение максимально возможного уменьшения количества обрабатываемых данных при соблюдении приемлемой точности измерения.

В работах [2, 9] описано несколько методов отбора пакетов, которые, однако, не отражают характер трафика в сети. Речь идет об унифицированном вероятностном отборе, отборе типа  $n$  из  $N$ , систематическом отборе, основанном на выборе временных интервалов, и систематическом отборе, зависимом от количества пакетов.

Для повышения точности измерения предлагается адаптивный подход к отбору образцов. В [10] предложены метод, основанный на линейной предикции, и метод, основанный на использовании нечеткого регулятора. Эти методы, однако, были разработаны для использования при мониторинге сетей, основанных на прото-

коле SNMP (Simple Network Management Protocol), и их применение к IPFIX без изменения невозможно. Метод, основанный на использовании нечеткого регулятора, в отличие от метода линейной предикции достиг лучших результатов в точности отбора и стал предметом дальнейшего исследования, направленного на оптимизацию точности и возможности его использования в связи с развитием стандартов, разрабатываемых IPFIX.

Цель настоящей статьи — представление метода отбора, с помощью которого можно модифицировать интенсивность отбора в зависимости от характера трафика для максимальной редукции обрабатываемых данных и обеспечения необходимой точности. Разработанный метод включает два взаимосвязанных процесса:

- классификация трафика с использованием нейронной сети;
- управление интервалом отбора с использованием нечеткого регулятора.

Для проверки метода был проведен эксперимент, основанный на мониторинге трафика в существующих инфраструктурах сетей.

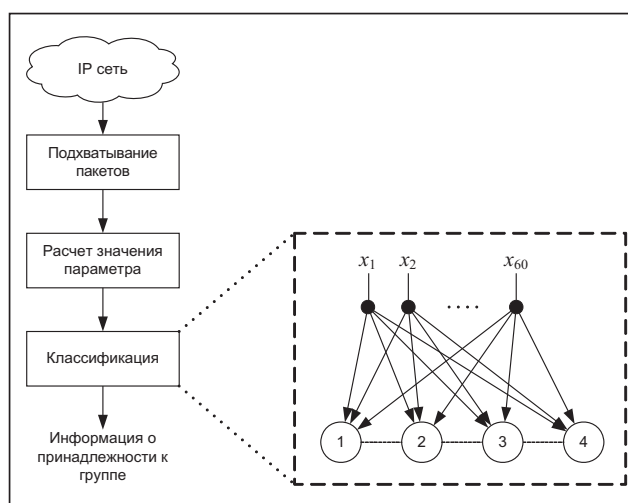
## 1. КЛАССИФИКАЦИЯ ТРАФИКА С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ

Классификация трафика в сети (рис. 1) проводится с целью сортировки его частей по группам, которые с точки зрения адаптивного отбора имеют похожие свойства. Согласно информации о пакетах, которые получил сетевой адаптер, сначала вычисляется значение измеряемого параметра. Известно, что этот параметр изменяется во времени, что фиксируется последовательностью его значений в течение заданного интервала времени. Обычно нас интересует изменение параметра в течение одной секунды (например, использование ширины полосы байт/секунда). Созданная последовательность значений в дальнейшем делится на отрезки одинаковой длины, которые являются входными данными классификатора. Выходом классификатора является отношение отдельных отрезков к каждой группе, которые имеют похожие свойства.

Для проверки метода была использована сеть Когонена [11], которая имеет 60 входов и четыре нейрона, расположенные в топологической решетке размера  $1 \times 4$ . Это значит, что для параметра, представленного значениями, рассчитанными относительно рамки избранного размера (1 с), классификатор свяжет все отрезки соответствующей длины (в данном случае — 60 с) с одной из четырех групп. Структура этой сети изображена на рис. 1, а ее параметры приведены в табл. 1.

**Таблица 1.** Параметры сети Когонена для классификации трафика в сети

Параметр	Значение
Количество входов	60
Размер решетки	$1 \times 4$
Начальный параметр обучения	0,9
Количество итераций адаптации	1000
Параметр обучения для фазы настройки	0,02
Рассеивание адаптирования функции окружающей среды нейрона	1



*Рис. 1.* Основная концепция классификации трафика с использованием сети Когонена

## 2. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕЧЕТКОГО РЕГУЛЯТОРА АДАПТИВНОГО ОТБОРА ОБРАЗЦОВ

Задачей нечеткого регулятора является определение размера интервала отбора образцов в зависимости от характера трафика сети. Если параметр, характеризующий трафик, имеет большие колебания во времени, то интервал отбора образцов должен быть малым. Если параметр не изменяется или изменяется незначительно, то интервал отбора может быть большим. Следовательно, нечеткий регулятор задает на выходе изменение интервала отбора, величина которого определяется формулой

$$\Delta T(i+1) = f[\Delta P(i), T(i)], \quad (1)$$

где  $\Delta P(i)$  — изменение параметра трафика в сети,  $T(i)$  — величина интервала отбора,  $f$  — функция нечеткого регулятора. Изменение параметра трафика в сети представлено отношением

$$\Delta P(i) = |P(i) - P(i-1)|, \quad (2)$$

где  $P(i)$  — значение параметра в итерации  $i$ ,  $P(i-1)$  — значение параметра в итерации  $(i-1)$ . Из отношения (2) следует, что значение  $\Delta P$  представляет собой величину изменения, причем направление этого изменения не учитывается. Величина интервала отбора определяется отношением

$$T(i) = T(i-1) + \Delta T(i), \quad (3)$$

где  $T(i-1)$  — величина интервала отбора в итерации  $(i-1)$ ;  $\Delta T(i)$  — изменение величины интервала отбора, представленное на выходе нечеткого регулятора в предыдущей итерации.

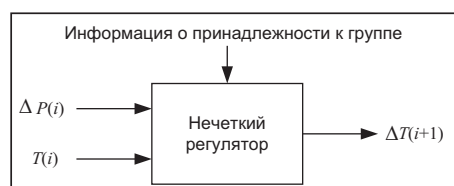


Рис. 2. Модель нечеткого регулятора адаптивного отбора

Модель нечеткого регулятора показана на рис. 2 (его функция описана ниже, а контекст использования представлен в [12]).

**2.1. Нечеткое определение.** Нечеткое определение формирует степень принадлежности к нечетким множествам четких входных значений. Это осуществляется с использованием функций принадлежности для обоих входов нечеткого регулятора.

На рис. 3 изображены функции принадлежности для входа  $\Delta P$ . Четкие значения этого входа принадлежат интервалу  $\langle 0, \pi \rangle$  (параметр  $\pi$  — максимальное ожидаемое значение изменения параметра  $P$ ). Для представления входных значений в нечетком регуляторе используется  $n$  нечетких множеств  $\Delta P_1, \dots, \Delta P_n$ , причем нечеткое определение формирует принадлежность четкого значения входа к каждому определенному нечеткому множеству. Функции принадлежности имеют линейную форму — функция  $\mu_{\Delta P_1}$  имеет форму  $S^-$ , функции  $\mu_{\Delta P_2}, \dots, \mu_{\Delta P_{n-1}}$  имеют форму пилю и функция  $\mu_{\Delta P_n}$  имеет форму  $S^+$  (см. рис. 3).

Функции принадлежности для входа  $T$  изображены на рис. 4. Четкие значения этого входа принадлежат интервалу  $\langle 0, \tau \rangle$  (параметр  $\tau$  — максимальное ожидаемое значение величины интервала отбора). Для представления входных значений в нечетком регуляторе используется  $n$  нечетких множеств  $T_1, \dots, T_n$ , причем нечеткое определение формирует принадлежность четкого значения входа каждому определенному нечеткому множеству. Функции принадлежности имеют линейную форму — функция  $\mu_{T_1}$  имеет форму  $S^-$ , функции  $\mu_{T_2}, \dots, \mu_{T_{n-1}}$  имеют форму пилю и функция  $\mu_{T_n}$  имеет форму  $S^+$  (см. рис. 4).

Поскольку функции принадлежности на рис. 3 и 4 накладываются, то значения каждого входа в нечетком регуляторе могут быть представлены степенью принадлежности к одному или двум нечетким множествам. Оба входа имеют одинаковое количество нечетких множеств, что, однако, не является общим требованием для построения нечеткого регулятора, хотя в этом случае происходит упрощение расчета четкого значения выхода.

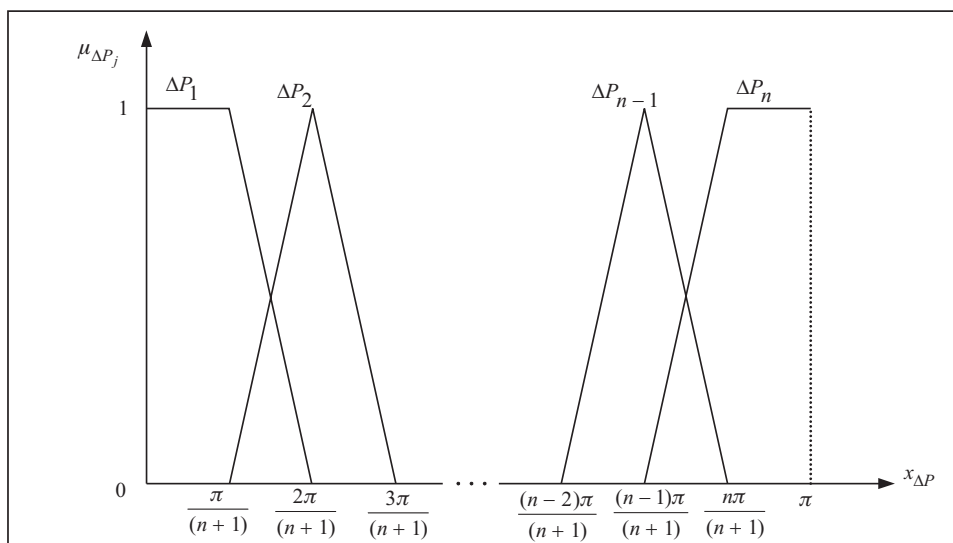


Рис. 3. Функции принадлежности для входа  $\Delta P$

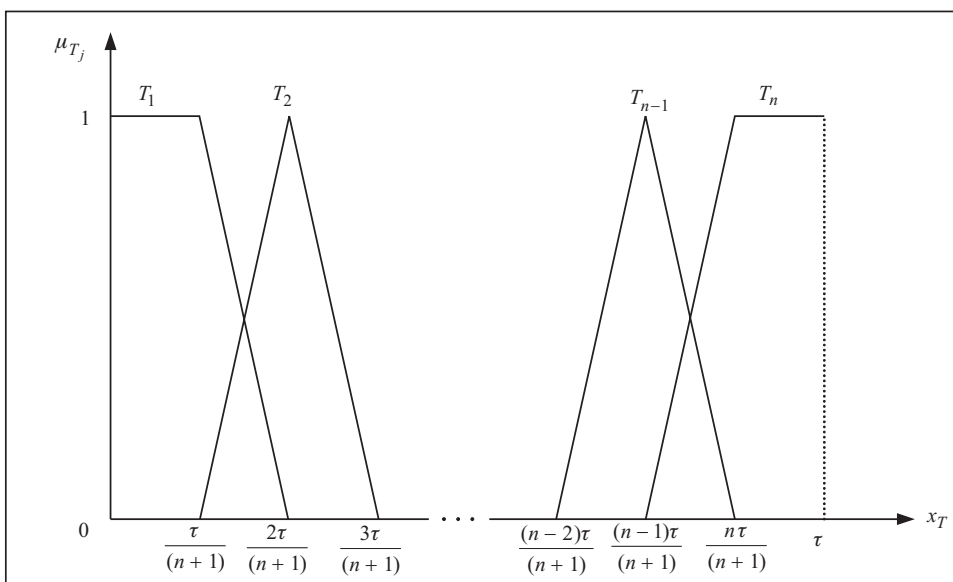


Рис. 4. Функции принадлежности для входа  $T$

### Алгоритм нечеткого определения

#### ВХОД АЛГОРИТМА

$n$  — количество нечетких множеств входов нечеткого регулятора,

$\pi$  — максимальное ожидаемое значение входа  $\Delta P$ ,

$\tau$  — максимальное ожидаемое значение входа  $T$ ,

$x_{\Delta P}$  — четкое значение входа  $\Delta P$ ,

$x_T$  — четкое значение входа  $T$ .

#### ВЫХОД АЛГОРИТМА

$\mu_{\Delta P} = (\mu_{\Delta P_j})$  представляет вектор, где  $\mu_{\Delta P_j}$  — степень принадлежности значения  $x_{\Delta P}$  нечеткому множеству  $\Delta P_j$ ,

$\mu_T = (\mu_{T_j})$  представляет вектор, где  $\mu_{T_j}$  — степень принадлежности значения  $x_T$  нечеткому множеству  $T_j$ .

## АЛГОРИТМ

1. Построить вспомогательные векторы  $r_{\Delta P} = (r_{\Delta P_j})$  и  $r_T = (r_{T_j})$  ( $j = \overline{1, n-1}$ ) на основании отношений

$$r_{\Delta P_j} = \frac{(n+1)x_{\Delta P}}{\pi} - j, \quad (4)$$

$$r_{T_j} = \frac{(n+1)x_T}{\tau} - j. \quad (5)$$

2. Построить вспомогательные векторы  $k_{\Delta P} = (k_{\Delta P_j})$  и  $k_T = (k_{T_j})$  ( $j = \overline{1, n-1}$ ) на основании отношений

$$k_{\Delta P_j} = 1 - r_{\Delta P_j}, \quad (6)$$

$$k_{T_j} = 1 - r_{T_j}. \quad (7)$$

3. Построить векторы  $\mu_{\Delta P} = (\mu_{\Delta P_j})$  и  $\mu_T = (\mu_{T_j})$  ( $j = \overline{1, n-1}$ ) с использованием одного из следующих шагов:

а) если  $r_{\Delta P_1} < 0 \wedge k_{\Delta P_1} > 1$ , то  $\mu_{\Delta P_1} = 1$ ,  $\mu_{\Delta P_l} = 0$  ( $l = 2, \dots, n$ );

если  $r_{T_1} < 0 \wedge k_{T_1} > 1$ , то  $\mu_{T_1} = 1$ ,  $\mu_{T_l} = 0$  ( $l = 2, \dots, n$ );

б) если  $r_{\Delta P_j} > 0 \wedge r_{\Delta P_j} < 1 \wedge k_{\Delta P_j} > 0 \wedge k_{\Delta P_j} < 1$ , то  $\mu_{\Delta P_l} = 0$  ( $l = 1, \dots, j-1$ ),  $\mu_{\Delta P_j} = k_{\Delta P_j}$ ,  $\mu_{\Delta P_{j+1}} = r_{\Delta P_j}$ ,  $\mu_{\Delta P_l} = 0$  ( $l = j+2, \dots, n$ );

если  $r_{T_j} > 0 \wedge r_{T_j} < 1 \wedge k_{T_j} > 0 \wedge k_{T_j} < 1$ , то  $\mu_{T_l} = 0$  ( $l = 1, \dots, j-1$ ),  $\mu_{T_j} = k_{T_j}$ ,  $\mu_{T_{j+1}} = r_{T_j}$ ,  $\mu_{T_l} = 0$  ( $l = j+2, \dots, n$ );

в) если  $r_{\Delta P_j} = 1 \wedge k_{\Delta P_j} = 0$ , то  $\mu_{\Delta P_l} = 0$  ( $l = 1, \dots, j$ ),  $\mu_{\Delta P_{j+1}} = 1$ ,  $\mu_{\Delta P_l} = 0$  ( $l = j+2, \dots, n$ );

если  $r_{T_j} = 1 \wedge k_{T_j} = 0$ , то  $\mu_{T_l} = 0$  ( $l = 1, \dots, j$ ),  $\mu_{T_{j+1}} = 1$ ,  $\mu_{T_l} = 0$  ( $l = j+2, \dots, n$ );

г) если  $r_{\Delta P_j} = 0 \wedge k_{\Delta P_j} = 1$ , то  $\mu_{\Delta P_l} = 0$  ( $l = 1, \dots, j-1$ ),  $\mu_{\Delta P_j} = 1$ ,  $\mu_{\Delta P_l} = 0$  ( $l = j+1, \dots, n$ );

если  $r_{T_j} = 0 \wedge k_{T_j} = 1$ , то  $\mu_{T_l} = 0$  ( $l = 1, \dots, j-1$ ),  $\mu_{T_j} = 1$ ,  $\mu_{T_l} = 0$  ( $l = j+1, \dots, n$ );

д) если  $r_{\Delta P_{n-1}} > 1 \wedge k_{\Delta P_{n-1}} < 0$ , то  $\mu_{\Delta P_n} = 1$ ,  $\mu_{\Delta P_l} = 0$  ( $l = 1, \dots, n-1$ );

если  $r_{T_{n-1}} > 1 \wedge k_{T_{n-1}} < 0$ , то  $\mu_{T_n} = 1$ ,  $\mu_{T_l} = 0$  ( $l = 1, \dots, n-1$ ).

**2.2. Инференция.** Для адаптивного отбора используется нечеткий регулятор типа Сагены, преимущество которого — простота реализации. Базис знаний нечеткого регулятора определяется правилом типа

$$p_{jk}: \text{ЕСЛИ } \mu_{\Delta P}(x_{\Delta P}) = \mu_{\Delta P_j} \wedge \mu_T(x_T) = \mu_{T_k}, \text{ ТО } x_{\Delta T} = x_{\Delta T_{jk}}, \quad (8)$$

где  $x_{\Delta T}$  — четкое значение выхода  $\Delta T$ ,  $x_{\Delta T_{jk}}$  — четкое значение выхода  $\Delta T$  для правила  $p_{jk}$ ,  $j = 1, \dots, n$ ,  $k = 1, \dots, n$ . Очевидно, что базис знаний будет содержать  $n^2$  правил.

Задача инференции состоит в определении веса выхода для каждого правила. Для этого в разработанном алгоритме используется метод  $\min$ , с помощью которого вес выхода определен наименьшей степенью принадлежности входов.

### Алгоритм инференции

#### ВХОД АЛГОРИТМА

$\mu_{\Delta P} = (\mu_{\Delta P_j})$  — вектор, где  $\mu_{\Delta P_j}$  определяет степень принадлежности значения  $x_{\Delta P}$  нечеткому множеству  $\Delta P_j$ ;

$\mu_T = (\mu_{T_j})$  — вектор, где  $\mu_{T_j}$  определяет степень принадлежности значения  $x_T$  нечеткому множеству  $T_j$ .

**ВЫХОД АЛГОРИТМА**

$W = (w_{jk})$  — матрица, где  $w_{jk}$  определяет вес выхода для правила  $P_{jk}$ .

**АЛГОРИТМ**

1. Построить вспомогательную матрицу  $A$  с помощью отношения

$$A = \mu_{\Delta P}^T \cdot B, \quad (9)$$

где  $B = (b_j)$  — вектор-строка, в которой  $b_j = 1$  ( $j = 1, \dots, n$ ).

2. Построить вспомогательную матрицу  $C$  с помощью отношения

$$C = D \cdot \mu_T, \quad (10)$$

где  $D = (d_j)$  — вектор-столбец, в котором  $d_j = 1$  ( $j = 1, \dots, n$ ).

3. Построить матрицу  $W = (w_{jk})$  с помощью отношения

$$w_{jk} = \min(a_{jk}, c_{jk}), \quad (11)$$

где  $j, k = \overline{1, n}$ .

**2.3. Определение выхода нечеткого регулятора.** Базис знаний нечеткого регулятора типа Сагана определен на основании правил, которые присваивают выходу регулятора четкие значения из множества реальных чисел. Поскольку четкие значения входов нечеткого регулятора могут принадлежать более чем одному нечеткому множеству, то можно применить более чем одно правило. Следовательно, общий выход формируется далее из значений выходов каждого примененного правила. Для этого целесообразно использовать взвешенное среднее значение весов выходов отдельных правил. Этот подход является более простым по сравнению с определением выхода нечеткого регулятора типа Мамдани. Четкое значение выхода определяется отношением

$$x_{\Delta T} = \frac{\sum_{j=1}^n \sum_{k=1}^n w_{jk} x_{\Delta T_{jk}}}{\sum_{j=1}^n \sum_{k=1}^n w_{jk}}. \quad (12)$$

Значения  $x_{\Delta T_{jk}}$  для отдельных правил из базиса знаний необходимо определить эмпирически. Очевидно, что если изменение значений наблюдаемого параметра будет большим ( $x_{\Delta P} \rightarrow \pi$ ) и соответственно большим будет интервал отбора ( $x_T \rightarrow \tau$ ), то необходимо интервал отбора уменьшить, а значит принять  $x_{\Delta T} < 0$ . Наоборот, если изменение значений наблюдаемого параметра будет малым ( $x_{\Delta P} \rightarrow 0$ ) и соответственно малым будет интервал отбора ( $x_T \rightarrow 0$ ), то необходимо интервал отбора увеличить, а значит принять  $x_{\Delta T} > 0$ . Явное определение этой функциональной зависимости не является обязательным, но иногда характер решаемой проблемы требует ее определения. В предложенном нечетком регуляторе целесообразно использовать отношение

$$x_{\Delta T_{jk}} = \frac{\delta(n-j-k+1)^3}{(n-1)^3}, \quad (13)$$

где  $\delta$  — максимальное значение изменения интервала отбора,  $j = 1, \dots, n$  и  $k = 1, \dots, n$ . Это значит, что интервал отбора в каждой итерации действия нечеткого регулятора будет иметь поправку на значение из интервала  $< -\delta, \delta >$ .

Выходная функция нечеткого регулятора определена отношением (13) и изображена на рис. 5.

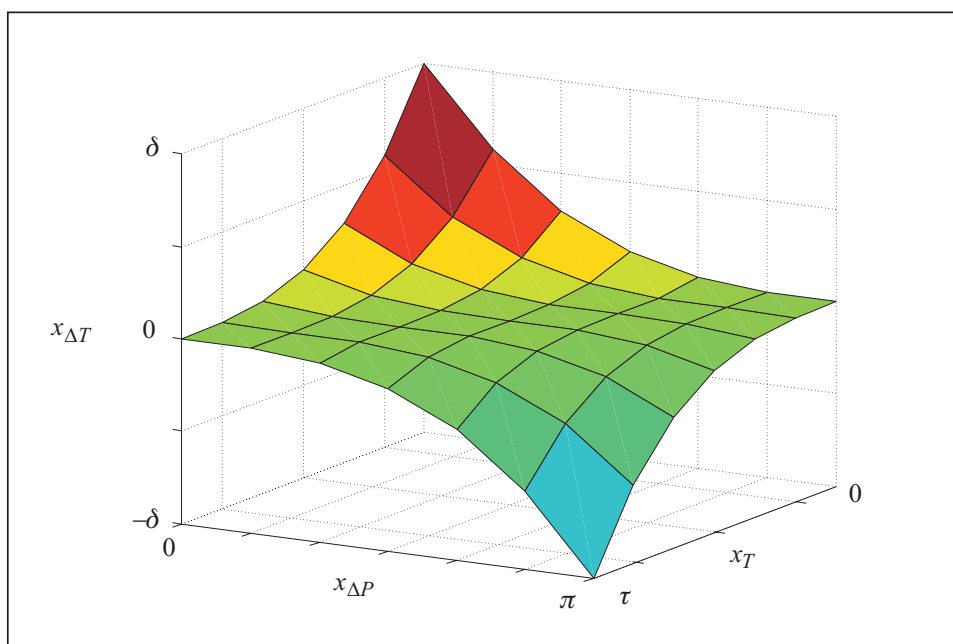


Рис. 5. Выходная функция нечеткого регулятора

### 3. ЭКСПЕРИМЕНТЫ

**3.1. Методика реализации экспериментов.** Для выполнения экспериментов использовалась программа MATLAB, для которой были разработаны «скрипты», реализующие обработку данных, модель нечеткого регулятора и симуляционная модель адаптивного отбора. Отметим, что созданные модели оказались в несколько раз эффективнее моделей, которые используют редактор FIS и модуль SIMULINK. Для создания модели сети Когонена, используемой для классификации трафика, применялись инструменты NN программы MATLAB. Можно предполагать, что после реализации метода в виде измерительного инструмента эффективность последнего проявится еще ярче.

Данные, использованные в эксперименте, получены из реального TCP/IP трафика сети. Эти данные представлены последовательностью пар

$$(time, size), \quad (14)$$

где *time* — время захвата пакета, измеренное с точностью  $10^{-6}$  с, *size* — величина пакета (в байтах). Для представления разработанных методов в настоящей статье использована популяция, состоящая из 12934 пакетов, записанных в течение 31 мин. Параметры нечеткого регулятора определены эмпирически на основе экспериментов с разными типами трафика в компьютерных сетях. Конкретное нахождение параметров в статье не представлено, поскольку при каждом применении метода параметры должны определяться индивидуально. Более эффективная реализация нечеткого регулятора с использованием упомянутого выше метода является предметом дальнейшего исследования авторов.

Рассматриваемую методику реализации экспериментов с целью ее сравнения с конвенционными методами отбора можно описать посредством следующих шагов.

1. Использовать данный метод отбора для создания представительного набора данных.
2. Сформировать значения наблюдаемого параметра на основе значений, полученных при отборе.
3. Обработать данные, полученные методом отбора, и данные, сравниваемые методом трансформации данных Мирила-Еге, который описан в [13].

4. Вычислить отклонение данных, полученных методом отбора, от сравниваемых данных, используя отношение

$$\text{err} = \sum_{j=1}^n [P_r(T_j) - P_s(T_j)]^2, \quad (15)$$

где  $n$  — длина последовательности данных, представляющих наблюдаемый параметр, последовательность  $P_r$  — сравниваемый параметр и последовательность  $P_s$  — параметр, полученный отбором (обе последовательности обработаны в предыдущих шагах).

Представительный набор данных, полученный в шаге 1, в реальной системе передается для дальнейшей обработки другим частям измерительной системы — на шаг 2 этого процесса. Для экспериментов все шаги были выполнены в среде программы MATLAB. Шаг 3 не был бы задействован, если бы сравнивались только методы отбора, основанные на выборе временных интервалов. Однако если для сравнения используются методы, основанные на выборе пакетов, то результатом будут данные с другим средним значением, а рассчитанное отклонение от сравниваемых данных невозможно сравнить с результатами, полученными другими методами. Преимущество такого преобразования заключается в том, что сравниваться будет форма временной зависимости данного параметра, а не его значение. Однако в результате сравнения мы не получаем абсолютной ошибки измерения, что является недостатком. Шаги 3 и 4 этого процесса нужны только для сравнения точности различных методов, но в реальной измерительной системе не используются.

**3.2. Сравнение методов отбора.** Метод адаптивного отбора, основой которого является нечеткий регулятор, сравнивался с методом стандартного вероятностного отбора, отбором типа  $n$  из  $N$ , систематическим отбором, основанном на времени, и систематическим отбором, основанном на количестве пакетов. Приведенные методы описаны в [9, 2].

Точность отдельных методов отбора можно сравнивать только в случае, когда образцы приблизительно одинаковых размеров. График отклонения данных от сравниваемых для всех рассматриваемых методов изображен на рис. 6. Как видим, адаптивный отбор, основанный на нечетком регуляторе и нейронной сети, дает отклонение данных на одну степень меньше относительно сравниваемых данных, чем конвенционные методы. При выборке, большей, чем 9% от полной популяции пакетов, полученной ошибкой можно пренебречь.

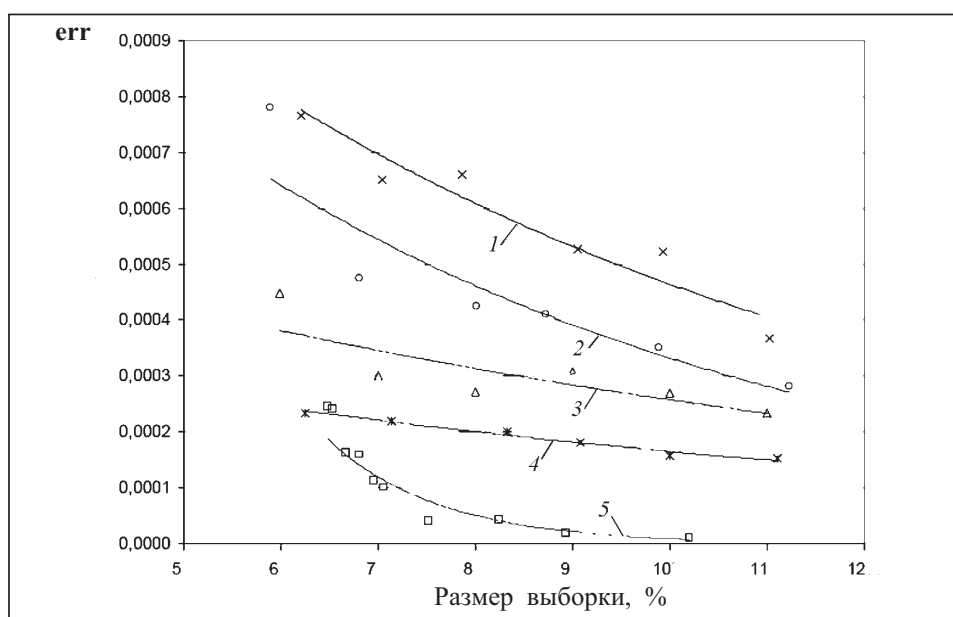


Рис. 6. График сравнения методов отбора: 1 — стандартный вероятностный отбор; 2 — систематический отбор, основанный на выборе временных интервалов; 3 — отбор типа  $n$  из  $N$ ; 4 — систематический отбор, основанный на количестве пакетов; 5 — адаптивный отбор, основанный на нечетком регуляторе и нейронной сети



## ЗАКЛЮЧЕНИЕ

Настоящая статья подготовлена в рамках решения оптимизации измерения и оценки параметров трафика в компьютерных сетях. Цель статьи — представление результатов исследования оптимизации процессов измерения, обусловленной необходимостью уменьшить объем обрабатываемых данных при измерении в сверхскоростных сетях. Особое внимание уделено разработке метода адаптивного отбора, который изменяет размер интервала отбора в зависимости от характера трафика. Суть адаптивного отбора состоит в использовании нечеткого регулятора для модификации интервала отбора, причем параметры нечеткого регулятора оптимизированы для разных классов трафика в сетях. Для классификации трафика в сети используется нейронная сеть Когонена.

Проверка разработанных методов и их сравнение с существующими методами отбора реализована экспериментально с использованием архивных данных трафика реальной компьютерной сети. Сравнение основано на оценке точности методов, т.е. на определении отклонения данных, полученных отбором, от сравниваемых данных. Исходя из этого можно утверждать, что с помощью предложенного метода достигается более высокая степень точности результата по сравнению с конвенционными методами.

Дальнейшая работа в этой области будет направлена на исследование возможностей реализации классификации трафика в нейронной сети с рекуррентным обучением, а также использование генетического алгоритма оптимизации параметров нечеткого регулятора для каждого класса трафика сети. Исходя из этого, можно избежать трудоемкой инициализации нечеткого регулятора, что повысит эффективность разработанного метода.

Результаты, полученные в данной статье, являются частью проектов «Оценка параметров трафика широкополосных коммуникационных инфраструктур: разработка аппликационных платформ» (VEGA No. 1/2175/05) и «Сеть виртуальных лабораторий для управления реальными системами» (KEGA No. 3/3121/05).

## СПИСОК ЛИТЕРАТУРЫ

1. Sadasivan G., Brownlee N., Claise B., Quittek J. Architecture for IP flow information export. — Internet Draft, IPFIX Working Group, September 2006. — 32 p.
2. Zseby T., Molina M., Duffield N., Niccolini S., Raspall F. Sampling and filtering techniques for IP packet selection. — Internet Draft, PSAMP Working Group, June 2007. — 42 p.
3. Zseby T., Boschi E., Brownlee N., Claise B. IPFIX Applicability. — Internet Draft, IPFIX Working Group, June 2007. — 32 p.
4. Čáky P., Klimo M., Palúch P., Skvarek O. End-to-end VoIP quality measurement // Acta Electrotechnica et Informatica. — 2006. — 6, N 1. — P. 47–51.
5. Uramová J. Impact of network state information on QoS // Acta Electrotechnica et Informatica. — 2006. — 6, N 1. — P. 43–46.
6. Mostafa H., Čičák P. Performance simulation of a Mobile-IP extension for optimized roaming service // Acta Electrotechnica et Informatica. — 2006. — 6, N 1. — P. 11–19.
7. Quittek J., Bryant S., Claise B., Aitken P., Meyer J. Information model for IP flow information export. — Internet Draft, IPFIX Working Group, February 2007. — 170 p.
8. Claise B. Specification of the IPFIX protocol for the exchange of IP traffic flow information. — Internet Draft, IPFIX Working Group, November 2006. — 64 p.
9. Hronský M., Jakab F., Potocký M., Jakab R., Giertl J. Sampling algorithms for nonintrusive measurement in network oriented educational systems // 4th International Conference on Emerging e-learning Technologies and Applications (ICETA 2005), Košice, Slovak Republic, 13–14 September 2005, elfa, s.r.o., 2005. — P. 165–171.
10. Hernandez E., Chidester M., George A. Adaptive sampling for network management // Journal of Network and Systems Management. — 2001. — 9, N 4. — P. 409–434.
11. Kohonen T. Self-organizing maps. Springer Series in Information Sciences. — Berlin: Springer, 1995. — 378 p.
12. Giertl J. Optimization of measurement and evaluation of operational parameters in computer networks (in Slovak), Dissertation thesis, Technická univerzita v Košiciach, Košice, 2006. — 93 p.
13. Giertl J., Jakab F., Bača J., Andoga R., Mirilovič M. Contribution to adaptive sampling of QoS parameters in computer networks // Acta Electrotechnica et Informatica. — 2006 — 6, N 1. — P. 52–59.

Поступила 12.09.2007