

УДК 621.395

УСОВЕРШЕНСТВОВАННЫЙ МЕТОД НЕЙРО-НЕЧЕТКОГО УПРАВЛЕНИЯ ОТБРАСЫВАНИЕМ ПАКЕТОВ В ТРАНЗИТНЫХ МАРШРУТИЗАТОРАХ ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ



К.А. ПОЛЬЩИКОВ

Донбасская государственная
машиностроительная академия



Ю.Н. ЗДОРЕНКО

Военный институт телекоммуникаций и информатизации
Государственного университета телекоммуникаций

Abstract – The need for effective management of queues in the routers of the telecommunications network increases with the increase of network traffic and limits of network resources. An important role in the process of management of queues belongs to a group of methods AQM. One of the most known and used in practice methods is RED. However, practice shows low versatility for using of RED and its modifications. It happens due to the complexity of selecting a large number of parameters in RED during the changing of the network status. This causes inefficient management of advanced discarding of packets. The result is increasing of the average packet delay and packets loss. In this regard, there is need to use an adaptive method that would provide the desired minimization of network packet delay and reduce the amount of packets loss. Such a method is proposed in this article. The proposed method, in contrast to the existing methods, is based on the application of neuro-fuzzy prediction system. Forecast data about the number of dropped packets in the next cycle of service and the number of incoming packets in the current cycle of service are used to make decision about early dropping probability of each incoming packet at the current time interval. In the neuro-fuzzy system of prediction of the amount of dropped packets the value of the bandwidth allocated to the class of packets in the current cycle of service is taken into consideration. The evaluation of the proposed method has shown its great effectiveness in comparison with RED and Tail Drop.

Анотація – Застосування методу раннього випадкового відкидання пакетів RED, а також його модифікацій обмежене складністю їх коректного налаштування. Використання параметрів за замовчуванням призводить до зростання рівня втрат пакетів, а також до збільшення їх середньої затримки. У статті запропоновано удосконалений метод активного управління пакетною чергою. Імовірність відкидання пакетів визначається за допомогою нейро-нечіткої системи з урахуванням зміни пропускної здатності, що виділяється для обслуговування пакетів різних класів.

Аннотация – Применение метода раннего случайного отбрасывания пакетов RED, а также его модификаций ограничено сложностью их корректной настройки. Использование параметров по умолчанию приводит к возростанию уровня потерь пакетов, а также к увеличению их средней задержки. В статье предложен усовершенствованный метод активного управления пакетной очередью. Вероятность отбрасывания пакетов определяется с помощью нейро-нечеткой системы с учетом изменения пропускной способности, выделяемой для обслуживания пакетов различных классов.

Введение

Растущие потребности в обмене информацией, популяризация интернет-приложений, активное использование сетевых информационных ресурсов являются причинами увеличения телекоммуникационного трафика и возникновения перегрузок маршрутизаторов. Негативное влияние сетевых перегрузок заключается в существенном замедлении и блокировке доставки сообщений, ухудшении качества передачи потоков данных.

Для борьбы с перегрузками, возникающими в транзитных маршрутизаторах, традиционно применяются методы управления очередями. Максимальная длина очереди ожидающих обслуживания пакетов определяется размером соответствующего буфера маршрутизатора. Реакцией на поступление пакета при заполненном буфере является его отбрасывание. Этот принцип положен в основу метода Tail Drop («отбрасывание хвоста»), с помощью которого реализуется пассивное управление пакетными очередями [1]. Преимуществом данного метода является его простота. В то же время Tail Drop присущи серьезные недостатки: недопустимые задержки пакетов и джиттер, эффект глобальной синхронизации, несправедливый захват очереди пакетами одного потока.

Для устранения указанных недостатков в современном оборудовании реализуется активное управление очередями (Active Queue Management, AQM), сущность которого состоит в превентивном (заблаговременном) отбрасывании части поступающих в маршрутизатор пакетов до момента заполнения соответствующего буфера. Наиболее распространенным методом активного управления очередями является Random Early Detection (RED) [2]. В соответствии с этим методом решение об отбрасывании того или иного пакета принимается на основе вычисления среднего размера очереди и вероятности отбрасывания пакетов. Недостаток RED заключается в применении эвристического алгоритма со статической настройкой параметров, не позволяющего существенно уменьшить потери пакетов. Для адаптивной настройки параметров RED разработаны его различные модификации [3–6], применение которых, к сожалению, не обеспечивает требуемой минимизации пакетных задержек и потерь в условиях неопределенности и случайности параметров сетевого трафика.

Перспективные решения в области управления процессами обработки и передачи информации связаны с применением гибридных интеллектуальных систем [7–12]. В работе [13] предложен метод управления отбрасыванием пакетов, основанный на нейро-нечетком прогнозировании величин \tilde{Z}_d (нагрузки в текущем цикле) и \tilde{R}_e (потерянной нагрузки в следующем цикле при пассивном управлении пакетной очередью). При этом вероятность отбрасывания пакетов вычисляется по формуле:

$$P = \frac{\tilde{R}_e}{\tilde{Z}_d}. \quad (1)$$

Для получения результатов прогноза используются величины Z_a , Z_b , и Z_c – значения входной нагрузки в трех предыдущих циклах. Применение этого метода позволяет минимизировать количество отброшенных пакетов и уменьшить среднее время пребывания пакета в очереди. Однако анализируемый метод не может быть применен в условиях наличия на входе канала нескольких пакетных очередей, для обслуживания которых выделяемая пропускная способность динамически изменяется [14]. Поэтому актуальной научно-прикладной задачей является усовершенствование метода активного нейро-нечеткого управления пакетными очередями в части учета возможного изменения значений пропускной способности канала, выделяемой для обслуживания пакетов различных классов.

I. Синтез нечеткой нейронной системы прогнозирования количества отброшенных пакетов

При реализации в канале динамического перераспределения пропускной способности, выделяемой для передачи пакетов различных классов, предложенная в работе [13] модель нейро-нечеткого прогнозирования числа отброшенных пакетов требует усовершенствования. Количество отброшенных пакетов заданного класса из числа претендовавших на передачу по каналу зависит, во-первых, от динамики поступления пакетов данного класса для передачи по каналу и, во-вторых, от того, какое текущее значение пропускной способности канала выделяется для передачи этих пакетов.

Исследования показали, что для прогнозирования числа пакетов заданного класса целесообразно на вход нечеткой нейронной системы подавать данные о количестве поступивших пакетов в трех предыдущих циклах (величины Z_a , Z_b и Z_c), а также величину V_d – значение пропускной способности канала, выделяемой в текущем цикле для передачи пакетов этого класса по каналу телекоммуникационной сети. Выходной переменной в такой нечеткой нейронной системе является величина \tilde{R}_e – прогнозируемое значение числа пакетов заданного класса, которые будут отброшены в текущем цикле.

Для каждой входной величины целесообразно использовать по две треугольные функции принадлежности. Осуществление качественного прогноза при таких исходных данных может обеспечить применение алгоритма нечеткого вывода Сугено 1-го порядка [15], в соответствии с которым база нечетких правил имеет следующий вид:

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (2)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (3)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (4)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (5)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (6)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (7)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (8)$$

$$\text{Если } (Z_a = \alpha_1) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (9)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (10)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (11)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (12)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_1) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (13)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (14)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_1) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (15)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_1), \text{ то } (\tilde{R}_e = Y_1), \quad (16)$$

$$\text{Если } (Z_a = \alpha_2) \text{ и } (Z_b = \beta_2) \text{ и } (Z_c = \gamma_2) \text{ и } (V_d = \delta_2), \text{ то } (\tilde{R}_e = Y_1), \quad (17)$$

где α_1 – терм номер 1 входной величины Z_a ; α_2 – терм номер 2 входной величины Z_a ; β_1 – терм номер 1 входной величины Z_b ; β_2 – терм номер 2 входной величины Z_b ; γ_1 – терм номер 1 входной величины Z_c ; γ_2 – терм номер 2 входной величины Z_c ; δ_1 – терм номер 1 входной величины V_d ; δ_2 – терм номер 2 входной величины V_d ; Y_1, Y_2, \dots, Y_{16} – значения индивидуальных выводов нечетких правил.

Вид и параметры функций принадлежности для каждой входной величины показаны на рис. 1 – 4.

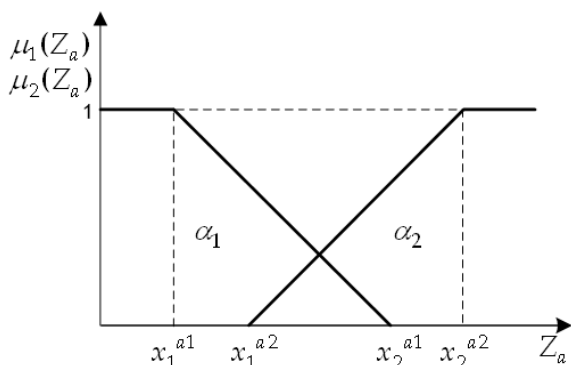


Рис. 1. Вид и параметры функций принадлежности входной величины Z_a

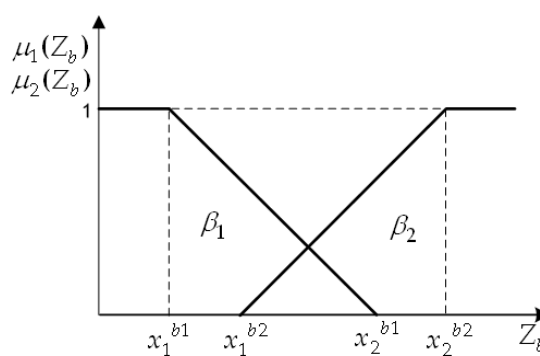


Рис. 2. Вид и параметры функций принадлежности входной величины Z_b

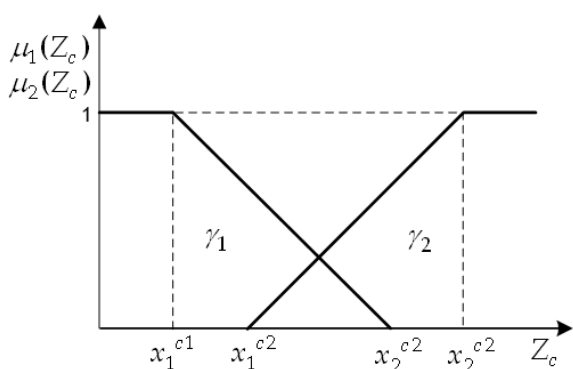


Рис. 3. Вид и параметры функций принадлежности входной величины Z_c

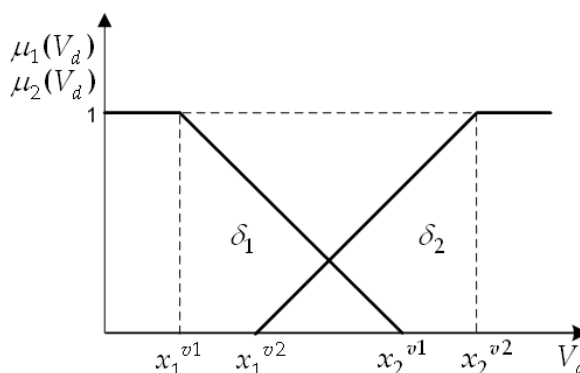


Рис. 4. Вид и параметры функций принадлежности входной величины V_d

В соответствии с заданным алгоритмом значения индивидуальных выводов нечетких правил определяются с помощью выражения:

$$Y_r = y_1^r Z_a + y_2^r Z_b + y_3^r Z_c + y_4^r Z_d + y_5^r, \quad (18)$$

где $y_1^r, y_2^r, \dots, y_5^r$ – коэффициенты для расчета индивидуального вывода нечеткого правила номер r .

Синтезируемая нейро-нечеткая система состоит из пяти слоев (рис. 5).

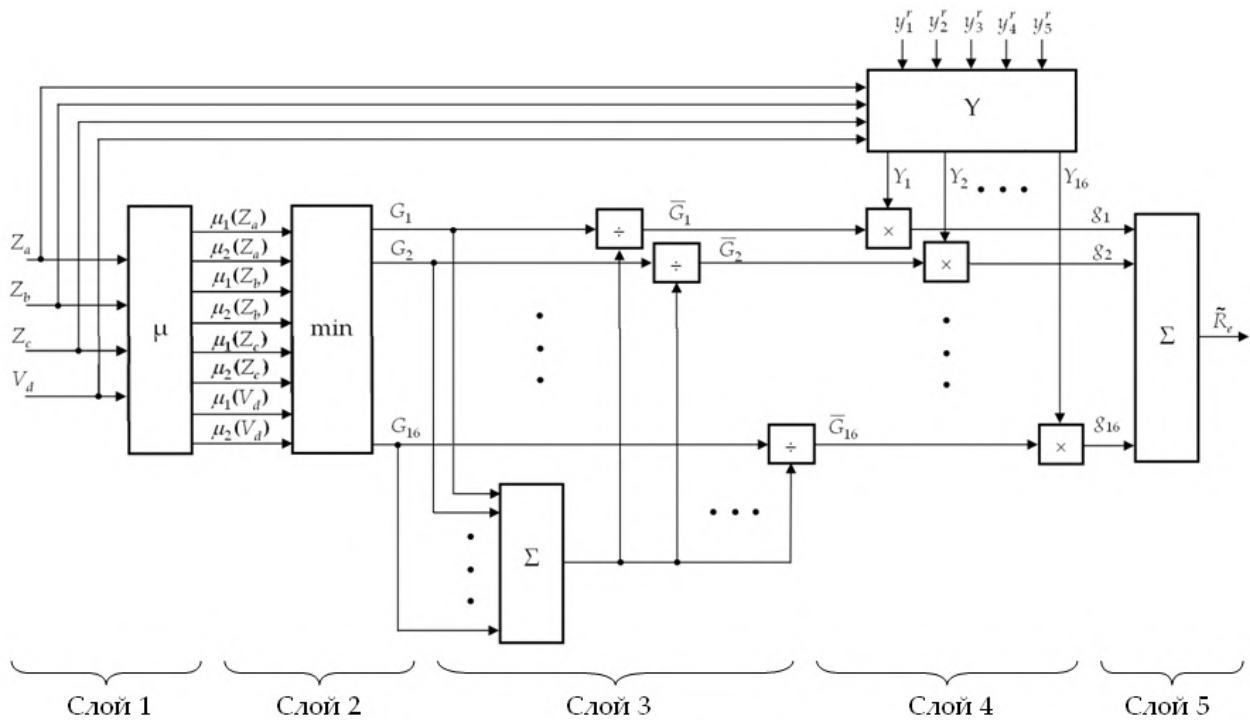


Рис. 5. Структура нейро-нечеткой сети

Первый слой нейронов выполняет процедуру фаззификации, т. е. вычисляет значения функций принадлежности для каждой входной величины:

$$\mu_1(Z_a) = \begin{cases} 1, & Z_a < x_1^{a1}; \\ \frac{x_2^{a1} - Z_a}{x_2^{a1} - x_1^{a1}}, & x_1^{a1} \leq Z_a < x_2^{a1}; \\ 0, & Z_a \geq x_2^{a1}; \end{cases} \quad (19)$$

$$\mu_2(Z_a) = \begin{cases} 0, & Z_a < x_1^{a2}; \\ \frac{Z_a - x_1^{a2}}{x_2^{a2} - x_1^{a2}}, & x_1^{a2} \leq Z_a < x_2^{a2}; \\ 1, & Z_a \geq x_2^{a2}; \end{cases} \quad (20)$$

$$\mu_1(Z_b) = \begin{cases} 1, & Z_b < x_1^{b1}; \\ \frac{x_2^{b1} - Z_b}{x_2^{b1} - x_1^{b1}}, & x_1^{b1} \leq Z_b < x_2^{b1}; \\ 0, & Z_b \geq x_2^{b1}; \end{cases} \quad (21)$$

$$\mu_2(Z_b) = \begin{cases} 0, & Z_b < x_1^{b2}; \\ \frac{Z_b - x_1^{b2}}{x_2^{b2} - x_1^{b2}}, & x_1^{b2} \leq Z_b < x_2^{b2}; \\ 1, & Z_b \geq x_2^{b2}; \end{cases} \quad (22)$$

$$\mu_1(Z_c) = \begin{cases} 1, & Z_c < x_1^{c1}; \\ \frac{x_2^{c1} - Z_c}{x_2^{c1} - x_1^{c1}}, & x_1^{c1} \leq Z_c < x_2^{c1}; \\ 0, & Z_c \geq x_2^{c1}; \end{cases} \quad (23)$$

$$\mu_2(Z_c) = \begin{cases} 0, & Z_c < x_1^{c2}; \\ \frac{Z_c - x_1^{c2}}{x_2^{c2} - x_1^{c2}}, & x_1^{c2} \leq Z_c < x_2^{c2}; \\ 1, & Z_c \geq x_2^{c2}; \end{cases} \quad (24)$$

$$\mu_1(V_d) = \begin{cases} 1, & V_d < x_1^{v1}; \\ \frac{x_2^{v1} - V_d}{x_2^{v1} - x_1^{v1}}, & x_1^{v1} \leq V_d < x_2^{v1}; \\ 0, & V_d \geq x_2^{v1}; \end{cases} \quad (25)$$

$$\mu_2(V_d) = \begin{cases} 0, & V_d < x_1^{v2}; \\ \frac{V_d - x_1^{v2}}{x_2^{v2} - x_1^{v2}}, & x_1^{v2} \leq V_d < x_2^{v2}; \\ 1, & V_d \geq x_2^{v2}. \end{cases} \quad (26)$$

Вторым слоем нейронов осуществляется процедура агрегирования, в результате которой определяются степени истинности каждого нечеткого правила:

$$G_1 = \mu_1(Z_a) \wedge \mu_1(Z_b) \wedge \mu_1(Z_c) \wedge \mu_1(V_d); \quad (27)$$

$$G_2 = \mu_1(Z_a) \wedge \mu_1(Z_b) \wedge \mu_1(Z_c) \wedge \mu_2(V_d); \quad (28)$$

$$G_3 = \mu_1(Z_a) \wedge \mu_1(Z_b) \wedge \mu_2(Z_c) \wedge \mu_1(V_d); \quad (29)$$

$$G_4 = \mu_1(Z_a) \wedge \mu_1(Z_b) \wedge \mu_2(Z_c) \wedge \mu_2(V_d); \quad (30)$$

$$G_5 = \mu_1(Z_a) \wedge \mu_2(Z_b) \wedge \mu_1(Z_c) \wedge \mu_1(V_d); \quad (31)$$

$$G_6 = \mu_1(Z_a) \wedge \mu_2(Z_b) \wedge \mu_1(Z_c) \wedge \mu_2(V_d); \quad (32)$$

$$G_7 = \mu_1(Z_a) \wedge \mu_2(Z_b) \wedge \mu_2(Z_c) \wedge \mu_1(V_d); \quad (33)$$

$$G_8 = \mu_1(Z_a) \wedge \mu_2(Z_b) \wedge \mu_2(Z_c) \wedge \mu_2(V_d); \quad (34)$$

$$G_9 = \mu_2(Z_a) \wedge \mu_1(Z_b) \wedge \mu_1(Z_c) \wedge \mu_1(V_d); \quad (35)$$

$$G_{10} = \mu_2(Z_a) \wedge \mu_1(Z_b) \wedge \mu_1(Z_c) \wedge \mu_2(V_d); \quad (36)$$

$$G_{11} = \mu_2(Z_a) \wedge \mu_1(Z_b) \wedge \mu_2(Z_c) \wedge \mu_1(V_d); \quad (37)$$

$$G_{12} = \mu_2(Z_a) \wedge \mu_1(Z_b) \wedge \mu_2(Z_c) \wedge \mu_2(V_d); \quad (38)$$

$$G_{13} = \mu_2(Z_a) \wedge \mu_2(Z_b) \wedge \mu_1(Z_c) \wedge \mu_1(V_d); \quad (39)$$

$$G_{14} = \mu_2(Z_a) \wedge \mu_2(Z_b) \wedge \mu_1(Z_c) \wedge \mu_2(V_d); \quad (40)$$

$$G_{15} = \mu_2(Z_a) \wedge \mu_2(Z_b) \wedge \mu_2(Z_c) \wedge \mu_1(V_d); \quad (41)$$

$$G_{16} = \mu_2(Z_a) \wedge \mu_2(Z_b) \wedge \mu_2(Z_c) \wedge \mu_2(V_d). \quad (42)$$

С помощью третьего слоя нейронов осуществляется нормализация результатов агрегирования:

$$\bar{G}_r = \frac{G_r}{\sum_{r=1}^{16} G_r}. \quad (43)$$

Четвертый слой осуществляет активизацию, в результате которой по формуле (18) определяются индивидуальные выводы каждого нечеткого правила. Кроме того, нейронами этого слоя вычисляется произведение результатов нормализации и активизации:

$$g_r = \bar{G}_r Y_r. \quad (44)$$

Значение прогнозируемой величины определяется в результате дефаззификации, выполняемой пятым слоем нейронов. При этом складываются результаты функционирования четвертого слоя системы:

$$\tilde{R}_e = \sum_{r=1}^{16} g_r. \quad (45)$$

Для получения значений коэффициентов $x_1^{a1}, x_2^{a1}, x_1^{a2}, x_2^{a2}, x_1^{b1}, x_2^{b1}, x_1^{b2}, x_2^{b2}, x_1^{c1}, x_2^{c1}, x_1^{c2}, x_2^{c2}, x_1^{v1}, x_2^{v1}, x_1^{v2}$ и x_2^{v2} необходима настройка весов нейронов первого слоя, а для получения значений $y_1^r, y_2^r, \dots, y_5^r$ требуется настройка весов нейронов четвертого слоя. Настройку нечеткой нейронной системы предлагается осуществлять на основе применения алгоритма обратного распространения ошибки [16]. Для обучения нейронов сформированы данные, представленные в виде следующей матрицы:

$$\begin{bmatrix} Z_1 & Z_2 & Z_3 & V_4 & R_5 \\ Z_2 & Z_3 & Z_4 & V_5 & R_6 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ Z_i & Z_{i+1} & Z_{i+2} & V_{i+3} & R_{i+4} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ Z_{I-4} & Z_{I-3} & Z_{I-2} & V_{I-1} & R_I \end{bmatrix}, \quad (46)$$

где Z_i – количество пакетов, поступивших для передачи по каналу в цикле номер i ; V_i – значение пропускной способности канала, выделяемой для передачи пакетов заданного класса в цикле номер i ; R_i – количество пакетов заданного класса, отброшенных в цикле номер i при пассивном управлении соответствующей очередью, $1 \leq i \leq I$.

Для получения матрицы (46) в течение $I = 500$ циклов наблюдения проводились измерения требуемых параметров одного из выходных портов маршрутизатора реальной телекоммуникационной сети. Настройка нейро-нечеткой системы осуществлялась с помощью восьми циклов обучения. Результаты обучения нейронов первого слоя представлены в таблице 1, а четвертого слоя – в таблице 2.

Таблица 1. Результаты обучения нейронов первого слоя

Параметр	x_1^{a1}	x_2^{a1}	x_1^{a2}	x_2^{a2}	x_1^{b1}	x_2^{b1}	x_1^{b2}	x_2^{b2}
Значение	0,012	12,02	0,021	12,02	0,011	12,04	0	12,04
Параметр	$x_1^{\bar{m}1}$	$x_2^{\bar{m}1}$	$x_1^{\bar{m}2}$	$x_2^{\bar{m}2}$	x_1^{v1}	x_2^{v1}	x_1^{v2}	x_2^{v2}
Значение	0,022	12,06	0,021	11,89	2	18	2	18

Таблица 2. Результаты обучения нейронов четвертого слоя

r	Значения коэффициентов				
	y_1^r	y_2^r	y_3^r	y_4^r	y_5^r
1	0	0	0,1843	0,5265	0,0263
2	0	0,1801	-0,0316	0	0
3	0	-0,02457	0,6134	0,1773	0,0886
4	-0,053	0,2965	0,027	-0,002	-0,0005
5	0,0011	0,0014	0,0003	0,0003	0,0002
6	-0,025	0,1876	0,2702	0,1634	0,0408
7	0,0003	-0,0856	-0,1105	-0,0245	-0,0122
8	-0,1062	0,3229	0,3492	0,2753	0,0688
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	-0,0664	-0,0796	-0,1194	-0,0531	-0,0133
13	0,0037	0,0047	0,0016	0,0011	0,0005
14	-0,05	-0,075	0	-0,0333	0,0083
15	0,0011	0,0014	0,0003	0,0003	0,0001
16	-0,1327	-0,1592	-0,2389	-0,1062	-0,0265

В программной среде Matlab выполнен ряд имитационных экспериментов для оценки точности прогнозирования нечеткой нейронной системой количества отброшенных пакетов. На рис. 6–9 представлены результаты одного из экспериментов. На рис. 6 показано, что в циклах $i = 1...7$ по каналу передавалось по 4 пакета, а в циклах $i = 8...12$ значение V_i уменьшилось до 2 пакетов, передаваемых в течение каждого цикла.

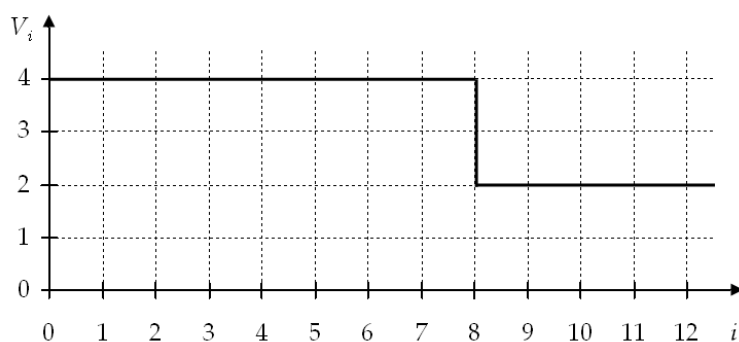


Рис. 6. Значения V_i , использованные в имитационном эксперименте

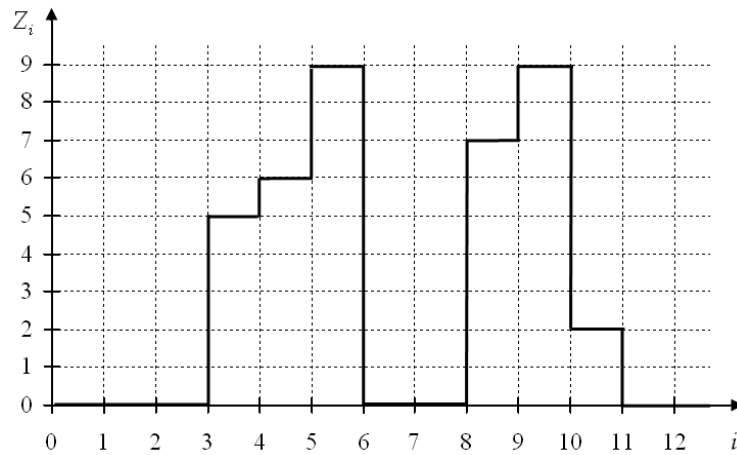


Рис. 7. Значения Z_i , использованные в имитационном эксперименте

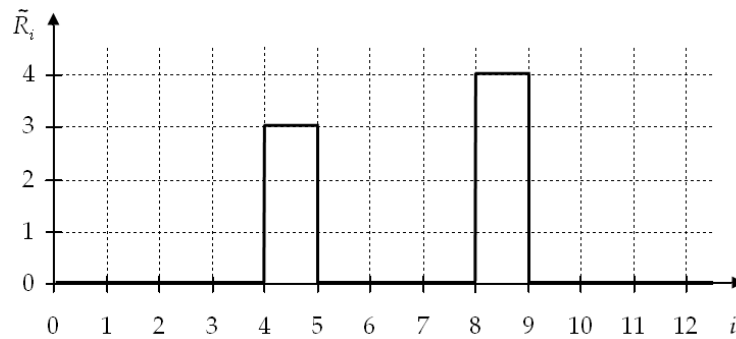


Рис. 8. Значения \tilde{R}_i , полученные в имитационном эксперименте

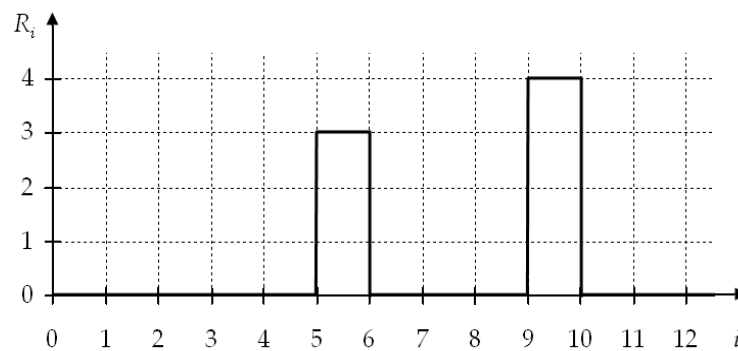


Рис. 9. Значения R_i , наблюдаемые в имитационном эксперименте

Указанное уменьшение выделяемой пропускной способности привело к тому, что на фоне наблюдаемого в циклах 8 и 9 возрастания количества поступивших пакетов Z_i (см. рис. 7) количество отброшенных пакетов в цикле 9 увеличилось по сравнению с циклом 4 ($R_9 > R_4$). Анализ представленных на рис. 8 и рис. 9 результатов показывает, что прогнозируемое значение \tilde{R}_i , которое вычисляется с помощью нейро-нечеткой системы в каждом текущем цикле i , практически совпадает с реальными значениями числа отброшенных пакетов R_i , зафиксированными в следующем цикле $(i+1)$. Точность прогнозирования, установленная в результате многочисленных имитационных экспериментов, составляет 95–97 %.

II. Оценка эффективности применения предложенного метода

Пусть в цикле i для передачи по каналу поступило Z_i пакетов. При этом из числа находившихся в очереди пакетов в предыдущем цикле $(i-1)$ не удалось передать q_{i-1}^E пакетов. Тогда общее число пакетов, которое требовалось передать в цикле i , определяется по формуле:

$$A_i = Z_i + q_{i-1}^E. \quad (47)$$

Если в цикле i пропускная способность канала V_i превысила значение A_i , то в этом цикле было передано $Y_i = A_i$ пакетов. В противном случае в цикле i было передано $Y_i = V_i$ пакетов. Поэтому формула для вычисления количества переданных в цикле i пакетов имеет вид:

$$Y_i = \begin{cases} A_i, & 0 \leq A_i < V_i; \\ V_i, & A_i \geq V_i. \end{cases} \quad (48)$$

Из числа пакетов, претендовавших на передачу в цикле i , одна часть пакетов R_i^{AQM} , возможно, была отброшена за счет реализации борьбы с перегрузками средствами AQM, вторая часть пакетов q_i^B могла быть буферизована, а третья часть R_i^{TD} могла быть отброшена за счет ограниченного объема Q буферной памяти. Условием отсутствия дефицита буферной памяти в цикле i является выполнение следующего неравенства:

$$A_i - 1 - R_i^{AQM} \leq Q. \quad (49)$$

Наличие слагаемого «-1» в неравенстве (49) объясняется тем, что в число пакетов, находящихся в очереди в цикле i , не мог попасть пакет, который первым передавался по каналу в этом цикле. Если неравенство (49) в цикле i было выполнено, то $R_i^{TD} = 0$. В противном случае по причине ограниченного объема буферной памяти в цикле i будет отброшено $R_i^{TD} = A_i - 1 - R_i^{AQM} - Q$ пакетов. Формула для вычисления числа отброшенных пакетов из-за дефицита буферной памяти имеет вид:

$$R_i^{TD} = \begin{cases} 0, & A_i - 1 - R_i^{AQM} \leq Q; \\ A_i - 1 - R_i^{AQM} - Q, & A_i - 1 - R_i^{AQM} > Q. \end{cases} \quad (50)$$

Общее количество пакетов, одновременно находившихся в очереди в цикле i , не может превышать размер буфера Q :

$$q_i^B = \begin{cases} 0, & A_i - 1 - R_i^{AQM} < 0; \\ A_i - 1 - R_i^{AQM}, & 0 \leq A_i - 1 - R_i^{AQM} < Q; \\ Q, & A_i - 1 - R_i^{AQM} \geq Q. \end{cases} \quad (51)$$

Как было указано выше, первый переданный в цикле i пакет не мог находиться в очереди в этом цикле. Поэтому, если $q_i^B > 0$, то в результате передачи по каналу Y_i количество находившихся в очереди пакетов в течение цикла i было уменьшено на величину $(Y_i - 1)$. Таким образом, число находившихся в очереди пакетов, которые в цикле i не удалось передать по каналу, можно вычислить с помощью выражения:

$$q_i^E = \begin{cases} 0, & q_i^B = 0; \\ q_i^B - (Y_i - 1), & q_i^B > 0. \end{cases} \quad (52)$$

Если $q_{i-1}^E > 0$, то один из числа q_{i-1}^E пакетов в цикле i передавался первым и в очереди в этом цикле не находился. Поэтому число пакетов, которые находились в очереди и в цикле $(i-1)$ и в цикле i , можно определить по формуле:

$$q_i^{BO} = \begin{cases} 0, & q_{i-1}^E = 0; \\ q_{i-1}^E - 1, & q_{i-1}^E > 0. \end{cases} \quad (53)$$

Количество пакетов, попавших в очередь в цикле i из числа поступивших в этом цикле пакетов, можно найти по формуле:

$$q_i^{BN} = q_i^B - q_i^{BO}. \quad (54)$$

Пакеты из числа q_i^B в цикле i находились в очереди в течение следующего суммарного времени:

$$t_i^B = \sum_{j=1}^{q_i^B} \frac{j}{V_i}. \quad (55)$$

Пакетам из числа q_i^{BO} в цикле i пришлось находиться в очереди в течение следующего суммарного времени:

$$t_i^{BO} = \sum_{j=1}^{q_i^{BO}} \frac{j}{V_i}. \quad (56)$$

Пакеты из числа вновь поступивших и попавших в очередь в цикле i (из числа q_i^{BN}) находились в очереди в этом цикле в течение такого суммарного времени:

$$t_i^{BN} = t_i^B - t_i^{BO}. \quad (57)$$

Итак, в первом цикле в очереди появились q_1^{BN} пакетов. Суммарное время ожидания этими пакетами обслуживания в цикле 1 составило t_1^{BN} . Во втором цикле в очереди появилось q_2^{BN} новых пакетов, которым в цикле 2 пришлось находиться в очереди в течение суммарного времени t_2^{BN} и т. д.

На основе имеющихся данных о значениях указанных величин, полученных для каждого i -го цикла ($i = 1, 2, \dots, I$), можно вычислить среднюю задержку пакета в очереди по формуле:

$$\bar{t} = \frac{\sum_{i=1}^I t_i^{BN}}{\sum_{i=1}^I q_i^{BN}}. \quad (58)$$

В качестве показателей эффективности применения разработанного метода будем использовать величину \bar{t} и величину суммарных потерь пакетов, вычисляемую по формуле:

$$R_{\Sigma} = \sum_{i=1}^I (R_i^{TD} + R_i^{AQM}). \quad (59)$$

Результаты расчета указанных выше величин по формулам (47)–(59), соответствующие процессу пассивного управления очередью, представлены в табл. 3.

Таблица 3. Значения параметров пассивного управления очередью

Параметр	Номер цикла										
	1	2	3	4	5	6	7	8	9	10	11
Z_i	0	0	5	6	9	0	0	7	9	2	0
V_i	4	4	4	4	4	4	4	2	2	2	2
A_i	0	0	5	7	12	5	1	7	14	9	7
Y_i	0	0	4	4	4	4	1	2	2	2	2
R_i^{AQM}	0	0	0	0	0	0	0	0	0	0	0
R_i^{TD}	0	0	0	0	3	0	0	0	5	0	0
q_i^B	0	0	4	6	8	4	0	6	8	8	6
q_i^E	0	0	1	3	5	1	0	5	7	7	5
q_i^{BO}	0	0	0	0	2	4	0	0	4	6	6
q_i^{BN}	0	0	4	6	6	0	0	5	4	2	0
t_i^B	0	0	2,5	5,25	9	2,5	0	7,5	18	18	10,5
t_i^{BO}	0	0	0	0	0,75	2,5	0	0	5	10,5	10,5
t_i^{BN}	0	0	2,5	5,25	8,25	0	0	7,5	13	7,5	0
\bar{t}	1,6296										
R_{Σ}	8										

Изменение величины V_i и количество пакетов, поступивших в каждом цикле i , соответствует данным имитационного эксперимента, результаты которого представлены на рис. 6 и 7. Объем буферной памяти составил 8 пакетов. Результаты, полученные на основе применения метода RED, представлены в табл. 4. При этом расчет вероятности отбрасывания пакетов был выполнен с помощью выражения:

$$P_i^{DEL} = \frac{P_{max} \cdot (avg_i - q_{min})}{q_{max} - q_{min}}, \quad (60)$$

где

$$avg_i = (1 - w_q) \cdot avg_{i-1} + w_q q_i^B. \quad (61)$$

Вычисления осуществлялись при следующих исходных данных: $w_q = 0,002$, $avg_0 = 4$, $q_{max} = 8$, $q_{min} = 2$ и $P_{max} = 0,5$.

Таблица 4. Значения параметров управления очередью на основе RED

Параметр	Номер цикла										
	1	2	3	4	5	6	7	8	9	10	11
Z_i	0	0	5	6	9	0	0	7	9	2	0
V_i	4	4	4	4	4	4	4	2	2	2	2
A_i	0	0	5	6	10	4	0	7	13	9	7
Y_i	0	0	4	4	4	4	0	2	2	2	2
R_i^{RED}	0	0	1	1	2	0	0	1	2	0	0
R_i^{TD}	0	0	0	0	0	0	0	0	2	0	0
q_i^B	0	0	3	4	7	3	0	5	8	8	6
q_i^E	0	0	0	1	4	0	0	4	7	7	5
q_i^{BO}	0	0	0	0	0	3	0	0	3	6	6
q_i^{BN}	0	0	3	4	7	0	0	5	5	2	0
t_i^B	0	0	1,5	2,5	7	1,5	0	5	18	18	10,5
t_i^{BO}	0	0	0	0	0	1,5	0	0	3	10,5	10,5
t_i^{BN}	0	0	1,5	2,5	7	0	0	5	15	7,5	0
\bar{t}	1,481										
R_Σ	9										

Результаты активного управления очередью, полученные на основе реализации предложенного метода, представлены в табл. 5.

Таблица 5. Значения параметров, полученных в результате управления очередью на основе применения предложенного метода

Параметр	Номер цикла										
	1	2	3	4	5	6	7	8	9	10	11
i	1	2	3	4	5	6	7	8	9	10	11
Z_i	0	0	5	6	9	0	0	7	9	2	0
V_i	4	4	4	4	4	4	4	2	2	2	2
A_i	0	0	5	7	9	5	1	7	9	9	7
Y_i	0	0	4	4	4	4	1	2	2	2	2
R_i^{AQM}	0	0	0	3	0	0	0	5	0	0	0
R_i^{TD}	0	0	0	0	0	0	0	0	0	0	0
q_i^B	0	0	4	3	8	4	0	1	8	8	6
q_i^E	0	0	1	0	5	1	0	0	7	7	5
q_i^{BO}	0	0	0	0	0	4	0	0	0	6	6
q_i^{BN}	0	0	4	3	8	0	0	1	8	2	0
t_i^B	0	0	2,5	1,5	9	2,5	0	0,5	18	18	10,5
t_i^{BO}	0	0	0	0	0	2,5	0	0	0	10,5	10,5
t_i^{BN}	0	0	2,5	1,5	9	0	0	0,5	18	7,5	0
\bar{t}	1,5										
R_Σ	8										

Анализ результатов, представленных в табл. 3–5, показывает, что применение предложенного усовершенствованного метода позволяет снизить среднюю задержку пакетов на 8–9 % по сравнению с использованием метода Tail Drop и сократить потери пакетов на 11% по сравнению с методом RED. Таким образом, на основе применения метода, представленного в статье, можно существенно повысить эффективность отбрасывания пакетов в транзитных маршрутизаторах телекоммуникационной сети в условиях изменения пропускной способности каналов, выделяемой для обслуживания пакетов различных классов.

Выводы

1. Применение известных методов управления отбрасыванием пакетов (управления очередями) для борьбы с перегрузками не обеспечивает требуемой минимизации пакетных задержек и потерь в условиях неопределенности и случайности параметров сетевого трафика. В статье предложен усовершенствованный метод, в котором, в отличие от известных методов, определение вероятности отбрасывания пакетов основано на применении нейро-нечеткой системы с учетом изменения пропускной способности, выделяемой для обслуживания пакетов различных классов.

2. Синтезированная нейро-нечеткая система функционирует в соответствии с алгоритмом Сутено 1-го порядка, имеет пять нейронных слоев, выполняющих процедуры нечеткого вывода (фаззификацию, агрегирование, нормализацию, активизацию и дефаззификацию). В результате оценки эффективности предложенного метода установлено, что его применение позволяет на 8–9 % уменьшить среднюю задержку

пакетов и на 11% сократить потери пакетов по сравнению с использованием известных методов управления очередями.

Список литературы:

1. Braden B, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet [Электронный ресурс] // RFC 2309. – Режим доступа: <http://tools.ietf.org/html/rfc2309>.
2. Floyd S., Jacobson V. Random early detection gateways for congestion avoidance // IEEE/ACM Transactions on networking. – 1993. – 1 (4). – P. 397–413.
3. Lin D., Morris R. Dynamics of Random Early Detection // Proc. of ACM SIGCOMM. – Sept. 1997. – P. 127–137.
4. Floyd S., Ramakrishnan G., Shenker S. Adaptive RED: An Algorithm For Increasing the Robustness of RED's Active Queue Management [Электронный ресурс] // Technical report, ICSI, August 2001. – Режим доступа: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>
5. Abdeljaber H., Mahafzeh M., Thabtah F., Woodward M. Fuzzy Logic Controller of Random Early Detection based on Average Queue Length and Packet Loss Rate // Performance Evaluation of Computer and Telecommunication Systems (SPECTS'2008). – Edinburgh, UK. – P. 428-432.
6. Ott T. J., Lakshman T. V., Wong L. H. SRED: Stabilized RED // Proc. of 18th Annual Joint Conf. INFOCOM'99. – Vol. 3. – P. 1346–1355.
7. Polshchykov K. O. Synthesis of neuro-fuzzy systems of data flows intensity control in mobile ad-hoc network // Microwave and Telecommunication Technology (CriMiCo), 23rd International Crimean Conference. – Sevastopol, 2013. – P. 517–518.
8. Tarasov O. F., Polshchykov K. O., Yeryomin N. V. Neuro Fuzzy Predicting Mathematic Model of Computer Network Load // International Conference «Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET'2014)», 2014. – P. 406–408.
9. Польщиков К. А. Обобщенные модели нейро-нечетких систем управления интенсивностью потоков данных в мобильной радиосети // Science and Education a New Dimension. – Budapest, 2013. – Vol. 8. – P. 133–137.
10. Польщиков К.А. Метод нейро-нечеткого управления интенсивностью отправки данных узлами-источниками в мобильной радиосети специального назначения // Наука і техніка повітряних сил Збройних Сил України. – Харків: ХУПС, 2012. – № 3 (9). – С. 118 – 122.
11. Польщиков К.А., Здоренко Ю.Н., Сокол Г.В. Методика нейро-нечеткого прогнозирования потерь пакетов при перегрузке компьютерной сети // Научный вестник ДГМА. – 2011. – №2 (8Е). – С. 77 – 86.
12. Польщиков К. А., Кубракова Е. Н., Краснобаев В.А. Модель нейро-нечеткого прогнозирования средней интенсивности поступления запросов на передачу потоков реального времени по каналу телекоммуникационной сети // Системи обробки інформації. – 2014. – Вип. 2 (118). – С. 193–197.
13. Польщиков К.О. Метод нейро-нечеткого активного управління пакетними чергами в мобільній радіомережі спеціального призначення // Збірник наукових праць ВІПІ НТУУ «КПІ». – 2010. – Вип. 1. – С. 52 – 59.
14. Польщиков К. О., Масесов М. О., Здоренко Ю. М. Метод прийняття рішення про розподіл пропускну здатності телекомунікаційного каналу на основі лінійного програмування // Сучасні інформаційні технології у сфері безпеки та оборони. – 2013. – № 3 (18). – С. 83 – 89.
15. Takagi T., Sugeno M. Fuzzy Identification of Systems and Its Applications to Modeling and Control // IEEE Transactions on Systems, Man and Cybernetics. – 1985. – Vol. 15, N. 1. – P. 116–132.
16. Rumelhart D.E., Hilton G.E., Williams R.J. Learning Internal Representations by Error Propagation // In Parallel Distributed Processing, t. 1. – Cambridge: M.I.T. Press, 1986. – P. 318-362.