УДК 621.391.31

Т. Н. КОВАЛЕНКО

УПРАВЛЕНИЕ НАГРУЗКОЙ И ПРЕДОТВРАЩЕНИЕ ПЕРЕГРУЗОК В ИНТЕЛЛЕКТУАЛЬНОЙ СЕТИ

Ключевым аспектом в обеспечении качества телекоммуникационных услуг является наличие эффективных механизмов управления нагрузкой в сети. Все сети проектируются таким образом, чтобы выполнялись требования по времени ожидания обслуживания для предполагаемых уровней нагрузки. Однако в случае сетевых перегрузок из-за возрастания нагрузки выше ожидаемого уровня время ожидания ответа часто возрастает до неприемлемого уровня. Кроме того, значительно возрастает частота появления ошибок передачи, количество тупиков, потерянных пакетов и пакетов, пришедших в неправильной последовательности. Перегрузки приводят не только к потере прибыли, но и к потере имиджа оператора в глазах потребителей из-за возникающих сбоев в обслуживании. Поэтому по мере развития сетей связи возрастает значимость гибких и устойчивых механизмов управления нагрузкой.

В интеллектуальной сети сетевые перегрузки чаще всего возникают из-за поступления массовых вызовов, стимулируемых масс-медиа. Развертывание и использование услуг интеллектуальных сетей постоянно растет: в Украине уже широко известны услуги бесплатного вызова (Freephone), вызова с повышенной оплатой (Premium Rate), предоставляемые по телефонам 8-800-xxx-xx-xx и 8-900-xxx-xx-xx, а также услуга телеголосования (Televoting). Вызовы именно по этим услугам чаще всего приводят к возникновению перегрузок, поэтому возрастает потребность в эффективных механизмах управления нагрузкой IN.

Постановка задачи

Предотвращение перегрузок в сети представляет собой типичную задачу распределения ресурсов. С одной стороны, сеть должна обслужить все запросы пользователей на обслуживание, которые часто являются непредсказуемыми и имеют пики по времени, скорости и объему передаваемой информации. С другой стороны, любой материальный ресурс в сети имеет конечную емкость, и им необходимо управлять с целью равномерного распределения ресурса между различными запросами. Следовательно, сетевая перегрузка является следствием того, что ресурсы сети не могут обеспечить обслуживание текущих запросов всех пользователей. В IN в качестве таких ресурсов будут рассмотрены процессоры узлов SCP. Главная цель алгоритма управления нагрузкой состоит в том, чтобы предохранить узлы SCP от перегрузок и таким образом обеспечить высокий уровень доступности обслуживания для потребителей даже в случае высокой нагрузки.

Борьба с перегрузками, как правило, выражается в отсеивании вызовов с малой вероятностью успешного завершения. Если этого не делать, то перегрузка приводит к максимальному значению общей занятости и к резкому спаду полезной нагрузки. Этот эффект объясняется переполнением буферов вновь поступающими вызовами и неуспешным их завершением.

Существующие механизмы унаследованы от традиционных сетей с коммутацией каналов и пакетов и ориентированы, как правило, на защиту от перегрузок отдельных узлов сети. Разрабатываемый алгоритм управления нагрузкой должен разрешать сложившуюся ситуацию перегрузки таким образом, чтобы выигрывала вся сеть, а не только отдельные узлы. Кроме того, необходимо обеспечить возможность прогнозирования перегрузки на стадии её формирования и принятия адекватных профилактических мер с целью нормализации ситуации. В данной статье будет представлен новый подход, ориентированный на сетевое представление о нагрузке и оптимизацию прибыли операторов.

Механизмы управления нагрузкой IN

Большая часть механизмов управления нагрузкой IN – это механизмамы управления на уровне сетевых элементов, направленные на защиту от перегрузок отдельных узлов сети. В целом все они могут быть разделены на два типа: согласованные и несогласованные. Согласованные методы подразумевают взаимодействие узлов SSP и SCP для управления трафиком, в то время как в несогласованных механизмах узлы SSP управляют трафиком только на основе данных собственных измерений.

В согласованных стратегиях управления на уровне сетевых элементов алгоритм обнаружения перегрузки расположен в SCP и работает в сочетании с находящимся в SSP механизмом отбрасывания части вызовов. Когда обнаружена перегрузка SCP, по сети SS7 к SSP посылается управляющее сообщение, содержащее информацию о серьезности перегрузки, там оно интерпретируется и запускается механизм отбрасывания соответствующего количества вызовов, чтобы снизить входящий трафик IN. Из этих механизмов наиболее широко используется автоматическое прореживание вызовов (ACG – Automatic Call Gapping), именно он определен в стандартах IN [1].

При автоматическом прореживании вызовов SCP исходя из своей производительности и поступающей нагрузки определяет уровень трафика, который должен принять каждый SSP. Эта информация посылается к SSP в сообщении, содержащем два параметра: интервал прореживания g_i и продолжительность τ. В течение следующих τ секунд SSP примет только один запрос на обслуживание каждые g_i секунд. Эти два параметра динамически корректируются в соответствии с наблюдаемым уровнем трафика. Механизм прореживания представлен на рис. 1.

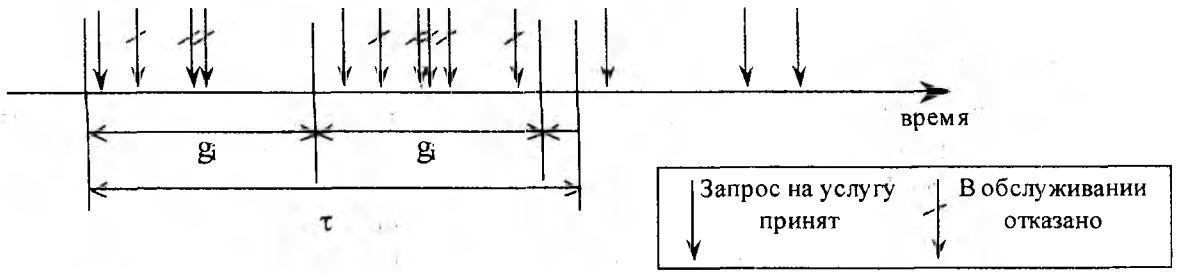

Рис. 1

Для определения уровня перегрузок, согласно которому устанавливаются значения g_i и τ, передаваемые в SSP, в SCP могут использоваться следующие три типа измеряемых параметров поступающих сообщений IN:

1. Среднее время ответа: среднее время, необходимое SCP для обработки запроса на услугу.
2. Подсчет необслуженных сообщений: под необслуженными сообщениями понимают те сообщения, которым отказано в обслуживании в SCP из-за истечения времени ожидания или переполнения буферов.
3. Интенсивность поступления сообщений: относится к полной интенсивности поступления запросов на обслуживание, предложенных SCP.

SCP в течение установленного интервала измерения определяет все три данные величины. Одним из способов отслеживания уровня перегрузки состоит в назначении целочисленного значения согласно таблице порогов исходя из всех трех критериев. В качестве значения уровня перегрузки SCP принимается максимальное из трех целых чисел. Наконец, есть два способа передачи информации об уровне нагрузки от SCP к SSP: в качестве составляющей ответного сообщения TCAP SS7, которые SCP передает узлу SSP, или в качестве самостоятельных запросов ACG, не являющихся частью ответных сообщений.

В несогласованных механизмах управления нагрузкой как обнаружение перегрузки, так и отбрасывание запросов, происходит в SSP – SCP не передает узлу SSP никаких уведомлений о перегрузке. В SSP заложен алгоритм, который контролирует время ответа SCP на запросы, поступающие от данного SSP. Если время задержки ответа на запрос превышает заданное значение, полагают, что произошла перегрузка, и входящие вызовы отбрасываются. Наиболее широко используемым алгоритмом такого типа является управление на основе механизма окна.

В SSP для определения возможности передачи запроса на узел SCP используются два параметра. Размер окна W показывает число запросов, на которые в любой момент времени могут быть не получены ответы, счетчик C – фактическое число этих необслуженных запросов. Механизм работает следующим образом:

– если $C = W$, окно передачи закрывается, и следующий запрос блокируется, иначе SSP сможет передавать новые запросы, увеличивая с каждой передачей значение счетчика C;

– значение W контролируется таймерами – каждый раз, когда к SCP передается сообщение, в SSP запускается таймер, и если время ожидания ответа истекает, SSP интерпретирует это как наличие перегрузки SCP. В этом случае уменьшается размер окна W;

– когда получен положительный ответ (прежде чем истекает соответствующий таймер), значение счетчика $C0$ увеличивается и размер окна W увеливается на единицу, когда $C0$ достигает порогового значения C_{max};

– на стороне SCP любой новый поступающий запрос отбрасывается в случае переполнения очереди поступающих сообщений.

Работа механизма показана на рис. 2:

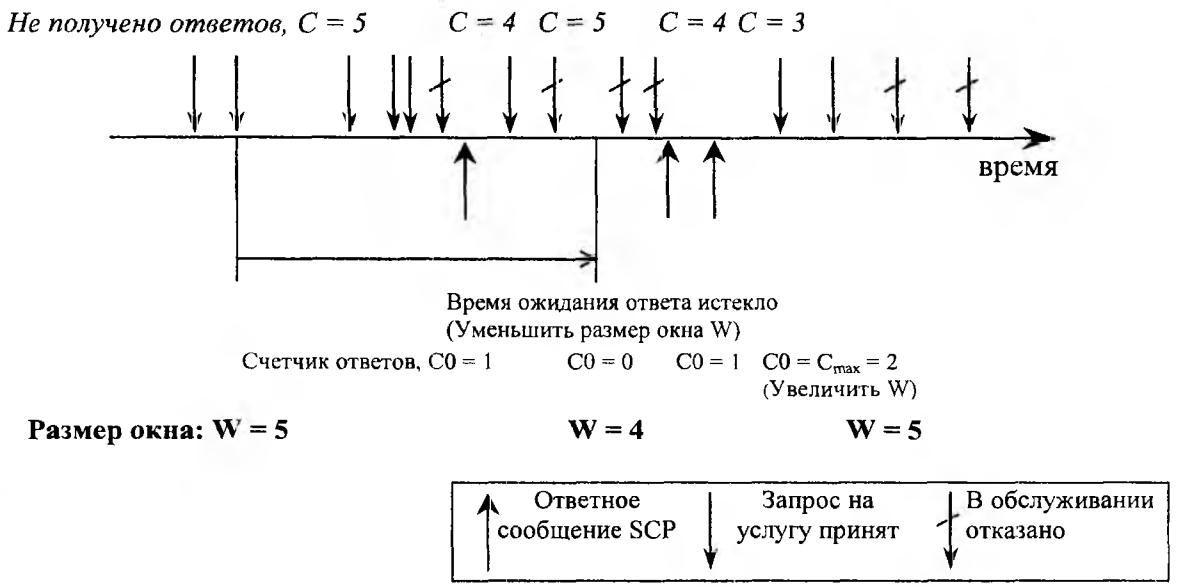

Рис. 2

Сетевой подход к управлению нагрузкой IN

Модель распределенной системы принятия решений состоит из составляющих, возможности которых и способ их организации в целостную структуру определяют поведение системы в целом [2]. Так как это говорит о коллективном поведении заинтересованных лиц, принимающих решение, в литературе [2, 3] в качестве альтернативной методологии описания этих систем были предложены рыночные модели, основанные на понятиях цены и конкуренции. При таком подходе составляющие систему «агенты» рассматриваются как потребители и производители в искусственной экономической системе, а их действия представлены в терминах производства и потребления продукции.

В данной статье рыночные модели представляли интерес с точки зрения их приложения для решения задач распределения ресурсов IN, при этом максимизировал некоторый общий критерий (интенсивность обслуживания, качество обслуживания, доходность сети и т.д.). В работе [3] рассмотрена аналогичная задача: авторы представили способ применения рыночного подхода к распределению качества обслуживания (QoS) в мультимедийных приложениях, при котором QoS динамически распределяется между заказчиками. Динамические изменения потребностей агентов и состояния сети приводят к тому, что агенты постоянно меняют свои решения. Целью описанного в [3] метода является обеспечение механизма эффективного распределения полосы пропускания, который будет реагировать на динамические изменения запросов потребителя.

В представленном в данной статье подходе управление нагрузкой осуществляется посредством маркеров. Маркеры «продаются» агентами производителей (узлы SCP) и «покупаются» агентами потребителей (узлы SSP). Количество маркеров, проданных SCP, определяет поступающую к нему нагрузку, а количество маркеров, купленных SSP, определяет, сколько запросов IN он может принять. «Торговля» маркерами осуществляется таким образом, чтобы максимизировать общую выгоду.

Для описания алгоритма вводится следующая система обозначений. Рассмотрим систему, состоящую из I SCP, J классов услуг и K SSP. Кроме того, обозначим произвольный SCP, класс услуги и SSP i, j, и k, соответственно, а прибыль, приносимую предоставлением услуги класса j, обозначим $r(j)$.

Все SSP содержат IJ пулов маркеров – по одному для каждой пары SCP и класса услуги. При поступлении в i-й SCP запроса класса j из соответствующего пула изымается один маркер. Пустой пул означает, что данный SCP не может больше принимать запросы соответствующего класса. Пулы заполняются заново на «аукционах», которые проводятся каждые T единиц времени (обычно T составляет величину порядка десятков секунд). Перед каждым аукционом рассматриваются последние предложения о цене, поступившие от всех агентов.

Обозначим заявки, поступающие от агента i-го SCP, как m_i, $p_i(1)$, ... , $p_i(J)$, где m_i – доступный в течение следующих T секунд ресурс процессора i-го SCP, $p_i(j)$ – требуемый для предоставления услуги класса j ресурс i-го SCP.

Заявки, поступающие от агента k-го SSP, обозначим как $q_k(1)$, ... , $q_k(J)$, где $q_k(j)$ – количество запросов на услугу IN класса j, ожидаемого от k-го SSP в течение следующего периода в T секунд. Самый простой способ формирования заявки – взять количество запросов, которые поступили в течение T последних секунд, и предположить, что в течение следующих T секунд поступит такое же их число.

На аукционе максимизируется общая ожидаемая выгода, измеряемая как полная прибыль. Метод состоит в предоставлении маркеров по одному таким образом, чтобы максимизировать ожидаемую предельную выгоду на каждом шаге итерации, измеряемую как отношение дохода к стоимости для каждого выданного маркера. В течение аукциона агенты SCP хранят запись об общем количестве остающихся ресурсов, о количестве проданных ресурсов (в терминах выданных маркеров) и стоимостью, соответствующей проведению транзакций, обозначаемых s_i, $n_{i,k}(j)$ и $c_{i,k}(j)$, соответственно. Пусть m_i – количество проданных агентом i-го SCP маркеров, которое обозначает количество ресурсов, предоставленных этим SCP. Значение s_i регулирует поступающую на i-й SCP нагрузку, $n_{i,k}(j)$ определяет, сколько запросов на услугу класса j может принять k-й SSP, для передачи его на обслуживание в i-й SCP.

Ожидаемое предельное значение дохода $u_k(j)$, связанное с предоставлением дополнительного маркера класса j агенту k-го SSP определяется как доход, связанный с предостав-

лением данной услуги, умноженный на вероятность того, что запрос на эту услугу поступит в течение аукциона. Последняя может быть получена, если принять ожидаемое количество запросов на услугу $q_k(j)$ равным среднему значению распределения Пуассона, и тогда получим

$$u_k(j) = r(j) \sum_{w=n_k(j)}^{\infty} \frac{q_k(j)^w}{w! e^{-q_k(j)}}. \tag{1}$$

Ожидаемое предельное значение стоимости $v_i(j)$, связанной с предоставлением дополнительного маркера класса j агентом i-го SCP, определяется как ожидаемый доход при альтернативном распределении тех же ресурсов. Оно вычисляется как общий доход, ожидаемый при всех альтернативных распределениях такого же количества ресурсов. Таким образом,

$$v_i(j) = \sum_{j'=1}^{J} \sum_{k'=1}^{K} \frac{p_i(j)}{p_i(j')} u_{k'}(j'). \tag{2}$$

Для удобства проведения расчетов выражение (2) лучше представить следующим образом:

$$v_i(j) = p_i(j) w_i, \tag{3}$$

где

$$w_i = \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{u_k(j)}{p_i(j)}. \tag{4}$$

Пусть распределение (i, j, k) соответствует предоставлению маркера на обслуживание заявки на услугу класса j агенту k-го SSP агентом i-го SCP. Предельная выгода такого действия – это отношение дохода к стоимости:

$$\delta_{i,k}(j) = \frac{u_i(j)}{v_i(j)}. \tag{5}$$

Во время аукциона полная прибыль максимизируется распределением ресурсов таким образом, чтобы выдача каждого маркера приводила к максимальному увеличению общей выгоды. Следовательно, оптимальным распределением ресурсов на каждой итерации являются значения i, j и k, максимизирующие значение $\delta_{i,k}(j)$.

Теперь можно представить формальное описание алгоритма:

Шаг 1. Инициализация.

Для всех SSP $k = 1, \ldots, K$: для всех SCP $i = 1, \ldots, I$ и всех видов услуг $j = 1, \ldots, J$ установить начальные значения $n_{i,k}(j) = 0$ и $c_{i,k}(j) = 0$, $\pi_k(j) = e^{-q_i(j)}$, $\Pi_k(j) = 1 - \pi_k(j)$, $u_k(j) = r_k(j)\Pi_k(j)$.

Для всех SCP $i = 1, \ldots, I$: установить начальные значения $s_i = c_i$, $w_i = \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{u_k(j)}{p_i(j)}$, для всех видов услуг $j = 1, \ldots, J$ $v_i(j) = p_i(j) w_i$.

Шаг 2. Определение вариантов оптимального распределения ресурсов.

Для всех SCP $i = 1, \ldots, I$, для которых $s_i \geq p_i(j)$, всех видов услуг $j = 1, \ldots, J$ и всех SSP $k = 1, \ldots, K$ составляем список вариантов распределения ресурсов (i^*, j^*, k^*), которые максимизируют $\delta_{i,k}(j)$.

Шаг 3. Выбор одного оптимального варианта распределения ресурсов из нескольких.

Если оптимальными являются несколько вариантов распределения ресурсов, выбрать один (i', j', k') согласно правилам, описанным ниже.

Шаг 4. Выполнение оптимального распределения:

$$n_{i'}(j') = n_{i'}(j') + 1,$$

$$c_{i',k'}(j') = c_{i',k'}(j') + u_{k'}(j'),$$

$$s_{i'} = s_{i'} - p_{i'}(j').$$

Шаг 5. Обновление внутренних переменных:

$$\pi_{k'}(j') := \pi_{k'}(j') q_{k'}(j') / n_{k'}(j').$$

$$\Pi_{k'}(j') := \Pi_{k'}(j') - \pi_{k'}(j').$$

$$u_{k'}(j') = r_{k'}(j') \Pi_{k'}(j').$$

Для всех SCP $i = 1,\ldots,I$: $w_i := w_i - r_{k'}(j')\pi_{k'}(j') / p_i(j')$, для всех услуг $j = 1,\ldots,J$ $v_i(j) = p_i(j) w_i$.

Шаг 6. Цикл.

Если имеется хотя бы один SCP, для которого $s_i \geq \min\{p_i(j)\}$, то перейти к шагу 2, в противном случае производится выход из цикла.

Для определения одного варианта оптимального распределения из совокупности полученных на 2-м шаге можно использовать выбор случайным образом – на значение общей прибыли это никак не повлияет, но на практике предпочтительнее применить один или одновременно несколько из следующих критериев:

1. С целью обеспечения относительно равномерной нагрузки на все SCP из всех полученных оптимальных вариантов распределения ресурсов предпочтение отдается тому, для которого значение s_i максимально, то есть услуга передается на обслуживание наименее загруженному SCP.

2. С целью максимизации общей выгоды с точки зрения рационального использования ресурса из всех полученных оптимальных вариантов распределения отдается предпочтение ресурсов тому, для которого значение $p_i(j)$ максимально, так как по мере уменьшения значения s_i может сложиться ситуация, когда для единственного оптимального варианта распределения ресурса оставшегося ресурса SCP будет недостаточно, то есть $s_i < p_i(j)$.

3. С целью максимально справедливого для данного метода распределения ресурсов между запросами на различные услуги, независимо от того, является ли количество запросов на данную услугу очень большим или относительно небольшим по сравнению с количеством доступных ресурсов, из всех полученных оптимальных вариантов распределения ресурсов предпочтение отдается тому, для которого отношение $n_k(j) / q_k(j)$ является минимальным.

Результаты сравнительного анализа методов управления нагрузкой IN

Для проведения сравнительного анализа механизмов управления нагрузкой на уровне сетевых элементов и представленного сетевого подхода была рассмотрена сеть, состоящая из пяти узлов SSP и двух узлов SCP, соединенных между собой по принципу «каждый с каждым». Конфигурация сети показана на рис. 3.

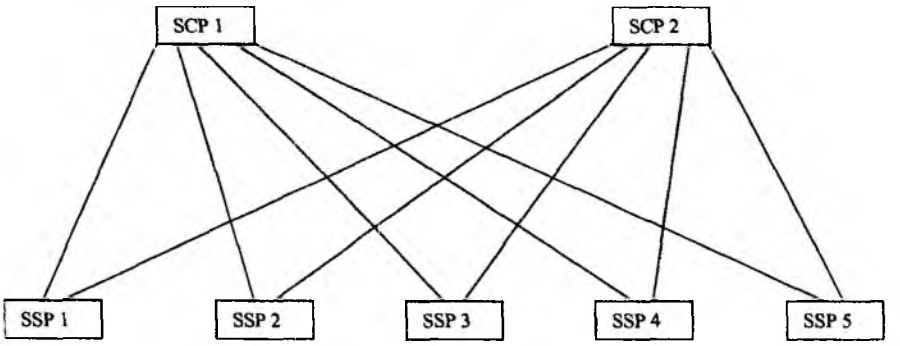

Рис. 3

В качестве распределяемого ресурса рассматривалась производительность процессоров SCP, измеряемая количеством транзакций, которые данный SCP может обработать за время аукциона $T = 10$ с, причем для обоих SCP она составляла $m_i = 250$ транзакций. Полагалось, что узлы SSP обслуживают 300, 100, 400, 200 и 500 тысяч пользователей, соответственно, предоставляя им пять видов услуг: FPH, PRM, ACC, CCC, VOT [1]. Среднее число поступающих запросов на каждую услугу было определено по формуле

$$q_k(j) = \alpha(j) N_k q_0(j) T, \qquad (6)$$

где $\alpha(j)$ – доля обслуживаемых пользователей, заинтересованных в услуге типа j, принятая равной 0,8; 0,3; 0,6; 0,4 и 0,5 для каждого типа услуги, соответственно; N_k – количество пользователей, обслуживаемых k-м узлом SSP; $q_0(j)$ – интенсивность поступления запросов на услугу типа j от одного пользователя [4,5].

В результате проведения расчетов была получена следующая матрица средних значений числа поступающих на узлы SSP запросов на каждый тип услуги (номер строки – j, номер столбца – k):

$$q = \begin{bmatrix} 21{,}6 & 7{,}2 & 28{,}8 & 14{,}4 & 36 \\ 6{,}75 & 2{,}25 & 9 & 4{,}5 & 11{,}25 \\ 9 & 3 & 12 & 6 & 15 \\ 6 & 2 & 8 & 4 & 10 \\ 6{,}3 & 2{,}1 & 8{,}4 & 4{,}2 & 10{,}5 \end{bmatrix}$$

Доход, получаемый от предоставления каждого типа услуги, рассчитывался как произведение средней продолжительности разговора пользователя [4, 5] на стоимость одной минуты разговора. Для услуги FPH он составил 157,3 ед. (10% вызовов городские, 90% – междугородние), для услуги PRM – 350 ед. (стоимость одной минуты – 150 ед. для всех вызовов), для услуг ACC и CCC – 52 и 144,5 ед., соответственно (городских вызовов – 80 и 30%, междугородних – 20 и 70%, соответственно), для услуги VOT – 20,37 ед. (стоимость одной минуты – 300 ед. для 50% вызовов, 10% вызовов – городские, остальные 40% – междугородние). Таким образом, $r(j) = [157{,}3 \quad 350 \quad 52 \quad 144{,}5 \quad 20{,}37]$.

Как было показано в [5], узел SCP должен обработать одну транзакцию при предоставлении услуг FPH или PRM, три транзакции при предоставлении услуг ACC или CCC и две транзакции при предоставлении услуги VOT, т.е. для обоих SCP $p_i(j) = [1 \quad 1 \quad 3 \quad 3 \quad 2]$.

Для решения задачи распределения ресурсов и ограничения поступающей нагрузки были использованы два подхода: сетевой и метод ACG. В последнем случае применялся тот же алгоритм определения оптимального на данной итерации варианта распределения ресурсов, с той лишь разницей, что на втором шаге в качестве оптимального варианта распределения

ресурсов выбирался тот, для которого значение $\Pi_{k'}(j')$ оказывалось максимальным, то есть маркер предоставлялся на обслуживание той услуги и тому SSP, которые имеют самую высокую вероятность поступления запроса. Остальные запросы были отброшены.

В результате решения поставленной задачи для сетевого подхода были получены следующие матрицы распределения производительности SCP:

Матрица распределения производительности первого и второго SCP (номер строки – тип услуги j, номер столбца – узел SSP k)

$$n_1 = \begin{bmatrix} 14 & 8 & 19 & 10 & 22 \\ 5 & 5 & 5 & 3 & 11 \\ 8 & 0 & 7 & 2 & 5 \\ 2 & 1 & 6 & 2 & 9 \\ 2 & 1 & 2 & 1 & 5 \end{bmatrix} \qquad n_2 = \begin{bmatrix} 15 & 4 & 19 & 11 & 24 \\ 7 & 1 & 10 & 6 & 7 \\ 1 & 3 & 5 & 4 & 10 \\ 6 & 2 & 5 & 4 & 4 \\ 1 & 0 & 3 & 1 & 2 \end{bmatrix}$$

Матрица распределения общей производительности между услугами и SSP

$$n = \begin{bmatrix} 29 & 12 & 38 & 21 & 46 \\ 12 & 6 & 15 & 9 & 18 \\ 9 & 3 & 12 & 6 & 15 \\ 8 & 3 & 11 & 6 & 13 \\ 3 & 1 & 5 & 2 & 7 \end{bmatrix}$$

Для метода ACG были получены следующие матрицы распределения производительности SCP.

Матрица распределения производительности первого и второго SCP (номер строки – тип услуги j, номер столбца – узел SSP k)

$$n_1 = \begin{bmatrix} 14 & 5 & 15 & 11 & 22 \\ 4 & 2 & 5 & 5 & 5 \\ 8 & 0 & 4 & 1 & 9 \\ 6 & 1 & 4 & 3 & 6 \\ 5 & 1 & 5 & 3 & 4 \end{bmatrix} \qquad n_2 = \begin{bmatrix} 10 & 4 & 17 & 5 & 18 \\ 4 & 1 & 6 & 1 & 8 \\ 2 & 4 & 10 & 6 & 8 \\ 1 & 2 & 5 & 2 & 6 \\ 2 & 2 & 5 & 2 & 8 \end{bmatrix}$$

Матрица распределения общей производительности между услугами и SSP

$$n = \begin{bmatrix} 24 & 9 & 32 & 16 & 40 \\ 8 & 3 & 11 & 6 & 13 \\ 10 & 4 & 14 & 7 & 17 \\ 7 & 3 & 9 & 5 & 12 \\ 7 & 3 & 10 & 5 & 12 \end{bmatrix}$$

Как видно из полученных результатов, в результате распределения ресурсов в соответствии с сетевым подходом услугам FPH, PRM и CCC по сравнению с механизмом ACG отдается некоторый приоритет, в то время как нагрузка, вызванная поступлением заявок на услуги ACC и VOT, ограничивается более жестко. В первом случае общая относительная прибыль оператора δ_Σ составит 22,783, а при реализации алгоритма, рекомендованного в [1], она составила бы 14,545. На рис. 4 представлены графики, показывающие рост относительной прибыли оператора на каждой итерации.

Из данного графика можно сделать вывод, что представленная в статье методика действительно обеспечивает оператору бо́льшую прибыль по сравнению с рекомендованным ранее механизмом ACG, не допуская при этом перегрузки узлов SCP. Максимальная прибыль обеспечивается за счет оптимального распределения разрешений на обработку заявок потребителей на каждой итерации алгоритма.

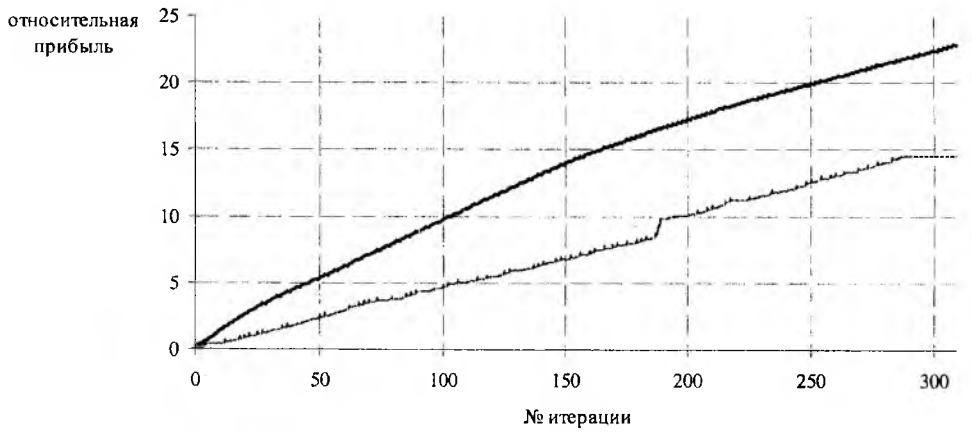

Рис. 4

Выводы

В статье представлен новый подход к управлению нагрузкой в интеллектуальной сети, ориентированный на сетевое представление о нагрузке и оптимизацию прибыли операторов. Сравнительный анализ механизма ACG, описанного в рекомендациях ITU-T, и разработанного алгоритма показал, что последний не только защищает от перегрузки все узлы SCP, но и обеспечивает оператору более высокую прибыль. Кроме того, представленный метод предотвращения перегрузок позволяет прогнозировать перегрузку на стадии её формирования и может быть адаптирован к различным скоростям изменения поступающей нагрузки путем увеличения или уменьшения значения T.

В сети IN зачастую под угрозой перегрузки находятся не только центральные процессоры SCP, но и другие ресурсы сети, такие как устройство управления диалогом TCAP, также расположенное в SCP, процессоры SDP, речевые платы IP. В таком случае даже если ресурсы процессоров SCP контролируются и защищены от перегрузок, перегрузка других типов ресурсов может привести к снижению QoS до неприемлемого уровня. Чтобы предотвратить возникновение таких ситуаций, алгоритм управления нагрузкой должен учитывать конечный объем всех ресурсов, требуемых для предоставления услуги, прежде чем принять запрос на эту услугу для обслуживания. Это может быть реализовано путем введения в алгоритм отдельных типов маркеров для каждого из ресурсов, которые требуются для обработки запроса, что является предметом дельнейших исследований.

Список литературы: 1. ITU-T. Recommendations Q.1200 – Q.1218. Helsinki, 1993. 2. *Wellman M.P.* A market oriented programming environment and its application to distributed multicommodity flow problems // Journal of Artificial Intelligence Research. 1993. № 1. С. 1 – 23. 3. *Yamaki E.A.* A Market-Based Approach to Allocating QoS for Multimedia Applications // Proceedings 2nd International Conference on Multi-Agent Systems. 1996. № 3. С. 86 – 88. 4. *Кучерявый А.Е., Миков А.С., Ревелова З.Б., Парамонов А.И.* Характеристики нагрузки интеллектуальной сети // Электросвязь. 2000. № 11. С. 7 – 9. 5. *Коваленко Т.Н.* Расчет нагрузки в интеллектуальной сети // Радиотехника. 2001: Всеукр. межвед. научн-техн. сб.

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 27.04.2003