

УДК 621.391.31

# ОЦЕНКА ПАРАМЕТРОВ SIP-СЕТИ С ИСПОЛЬЗОВАНИЕМ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ



[Е.Д. КУЗЬМИНЫХ](#)

Харьковский национальный  
университет радиоэлектроники

*Розроблена модель Петрі сценарію встановлення з'єднання в мережах SIP, яка дозволяє оцінити ймовірно-часові характеристики сигнального трафіка та динаміку використання буферів SIP серверів.*

*The Petri model of call setup and disconnect in SIP network is worked out. This model allows to evaluate the timed and probability characteristics of signaling traffic and the dynamics of server buffer utilization.*

*Разработана модель Петри сценария установления соединения в сетях SIP, которая позволяет оценить вероятностно-временные характеристики сигнального трафика и динамику использования буферов SIP-сервера.*

## Введение

Рост числа пользователей сетей IP-телефонии вызвал проблему, связанную с контролем перегрузок SIP-серверов, возникающих из-за отсутствия достаточных ресурсов для установления и завершения сессий пользователей. При перегрузке пропускная способность SIP-сервера может значительно снижаться и даже привести к выходу сервера и всей сети SIP из строя. Перегрузка в сети SIP может быть вызвана различными причинами [1]. Это может быть результатом активности пользователей, пытающихся установить соединения примерно в одно и то же время, например, при голосовании на ТВ-шоу или в чрезвычайной ситуации. Другая распространенная причина перегрузки – отказ одного из элементов в кластере SIP серверов, что снижает общую производительность и требует распределения входящей нагрузки между остальными серверами кластера. Еще одна причина перегрузки, когда оконечные терминалы после сбоя в сети пытаются одновременно зарегистрироваться, например, после отключения питания или смены настроек поставщиком услуг.

Измерение характеристик элементов SIP-сети в условиях перегрузки и оценка этих характеристик как для веб-серверов, так и для существующих телефонных сетей является экономически необоснованной, а иногда нереализуемой задачей. Поэтому возникает задача моделирования поведения элементов сети SIP для исследования параметров сети в режиме перегрузки.

Задачам борьбы с перегрузкой в сетях SIP посвящено много работ, которые условно можно разделить на три группы: исследование таймеров ретрансляции сообщений, заложенных в протоколе SIP [2, 3]; исследование локальных методов борьбы с перегрузкой [4, 5]; исследование методов борьбы с перегрузкой с обратной свя-

зью [6 - 8]. В большинстве работ для моделирования поведения SIP-сети использовались такие средства моделирования, как Matlab, OPNET [3] или ns-2 [4]. Но эти средства имеют встроенные элементы сети с определенным набором характеристик, и поэтому некоторые параметры элементов сложно менять, т. к. требуется знание языков программирования (например, C++) для внесения изменений в настройки сети. При работе с моделью в перечисленных средствах моделирования графическое представление модели отделено от его описательного содержания.

В работе предложено использовать иерархические раскрашенные сети Петри [9], которые уже успешно используются для исследования моделей телекоммуникационных систем. В первой части статьи рассмотрены основные особенности установления соединения в сетях SIP. Вторая часть посвящена моделированию SIP-сети с подробным описанием всех элементов. Модель отражает такие особенности архитектуры «клиент-сервер» в сети SIP, как трехфазный механизм установления соединения, разное время обслуживания запросов и ответов на сервере, а также обеспечивает лёгкую масштабируемость сетей Петри при изменении параметров SIP-сети. В третьей части проведена оценка параметров сети SIP на примере измерения задержки времени установления соединения, задержки в буфере сервера, текущей длины очереди в буфере, коэффициента использования ресурсов SIP-сервера.

## I. Анализ архитектуры сети SIP

Сеть, построенная на базе протокола сигнализации SIP [9], становится всё более распространённой и выполняет функции управления вызовами при предоставлении сервисов телефонии, видеоконференций и обмена данными. Архитектура сети SIP представлена на рис. 1 и состоит из таких элементов сети, как прокси-сервер (Proxy Server), сервер определения местоположения (Location Server), сервер переадресации (Redirect Server), абонентские SIP терминалы.

Основным элементом, реализующим функции управления соединением, является *терминал*. Остальные элементы сети отвечают за маршрутизацию вызовов и, при необходимости, – за предоставление дополнительных услуг.

*Прокси-сервер* принимает запросы, обрабатывает их и в зависимости от типа запроса выполняет определенные операции. Это может быть поиск и вызов пользователя, маршрутизация запроса, предоставление услуг видеоконференции и т.д. Прокси-сервер состоит из клиентской и серверной частей, поэтому может принимать вызовы, инициировать собственные запросы и возвращать ответы. Прокси-сервер может быть физически совмещен с *сервером определения местоположения* (в этом случае он называется registrar) [10].

*Сервер переадресации* предназначен для определения текущего адреса вызываемого пользователя. Вызывающий пользователь передает на сервер сообщение с известным ему адресом вызываемого пользователя, а сервер обеспечивает переадресацию вызова на текущий адрес этого пользователя.

Для хранения текущего адреса пользователя служит *сервер определения местоположения* пользователей, представляющий собой базу данных адресной информации. Кроме постоянного адреса пользователя, в этой базе данных может храниться один или несколько текущих адресов.

Протоколом SIP предусмотрены 3 основные сценария установления соединения: с участием прокси-сервера, с участием сервера переадресации и непосредственно между пользователями. Различие между перечисленными сценариями заключается в том, что по-разному осуществляется поиск и приглашение вызываемого пользователя. В нашей работе прокси-сервер выполняет функции всех вышеперечисленных серверов, как показано на рис.1, а.

Исходной информацией для построения модели является схема сценария соединения по протоколу SIP с участием прокси-сервера (рис. 1, б). Сеть SIP имеет клиент-серверную архитектуру, унаследованную от протокола HTTP. Клиент выдает запросы, в которых указывает, что он желает получить от сервера. Сервер принимает запрос, обрабатывает его и выдает ответ, который может содержать уведомление об успешном выполнении запроса, уведомление об ошибке или информацию, затребованную клиентом.

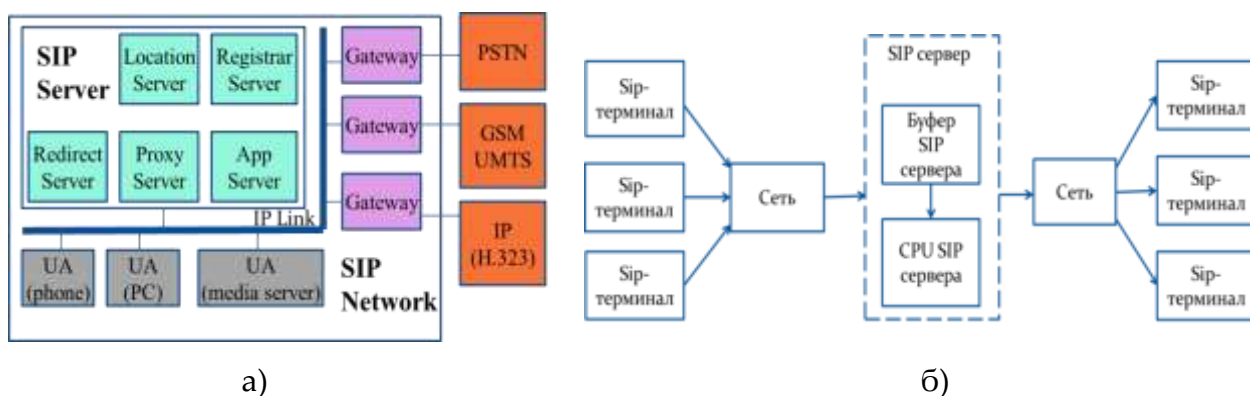


Рис. 1. Архитектура SIP-сети (а) и схема сценария установления соединения по протоколу SIP (б)

На рис. 2 показан обмен сообщениями между абонентом и сервером, необходимыми для установки соединения, а затем и разъединения. Абонент посылает прокси-серверу запрос на соединение, отправляя сообщение INVITE. Прокси-сервер возвращает сообщение 100 Trying и передает сообщение INVITE вызываемому абоненту. Вызываемая сторона отвечает сообщением 180 Ringing, которое прокси-сервер пересылает вызывающей стороне. После того как вызываемый абонент снимет трубку, вызывающей стороне отправляется сообщение 200 OK, которое транслируется прокси-сервером. Вызываемому абоненту возвращается подтверждающее сообщение ACK.

С этого момента соединение считается установленным и начинается обмен медиа-трафиком по протоколам RTP/RTCP. Сторона, желающая завершить соединение, посылает сообщение BYE, и после получения подтверждающего 200 OK соединение разрывается.

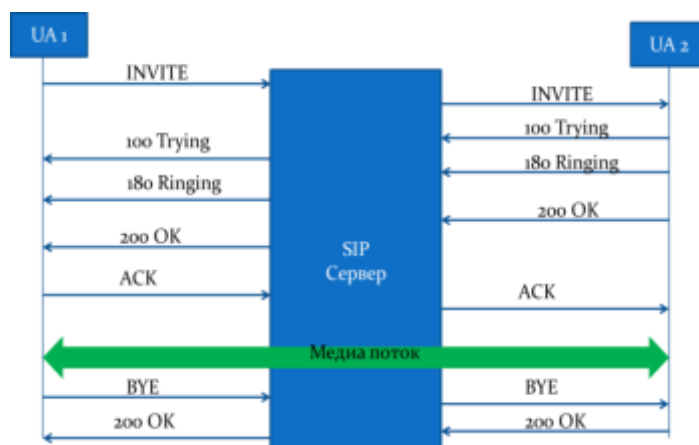


Рис. 2. Сценарий установления соединения по протоколу SIP

## II. Модель SIP-сети

Для имитационного моделирования SIP-сети были выбраны раскрашенные сети Петри [11] и среда CPN Tools [12], которая свободно распространяется для использования в некоммерческих целях и представляет собой интегрированную среду для графического ввода моделей, имитации их динамики и определения основных свойств. Кроме средств рисования сетей, система содержит встроенный язык программирования ML, предназначенный для описания атрибутов графических объектов. Алгоритмы обработки, зависящие от параметров трафика (адреса, размера, приоритета обслуживания и др.), описываются посредством задания параметров маркеров. Иерархичность позволяет строить сложные многокомпонентные модели. Временные параметры маркера обеспечивают возможность моделирования динамических характеристик объекта (например, задержку, время обслуживания). Эти сети успешно применены для моделирования компьютерных сетей, механизмов управления трафиком и распределения нагрузки [12 - 14].

Модель SIP-сети, структурная схема которой изображена на рис. 1, б, представляет собой иерархическую сеть Петри, состоящую из пяти отдельных листов. На рис. 3 изображён верхний уровень иерархии модели, на остальных листах представлены блок генератора трафика *Source*; участок сети *Network*, вносящий задержку; блок буфера сервера *Buffer*, в котором происходит распределение сообщений по очередям; блок *Server*, в котором происходит обслуживание сообщений и генерация ответа, если это необходимо.

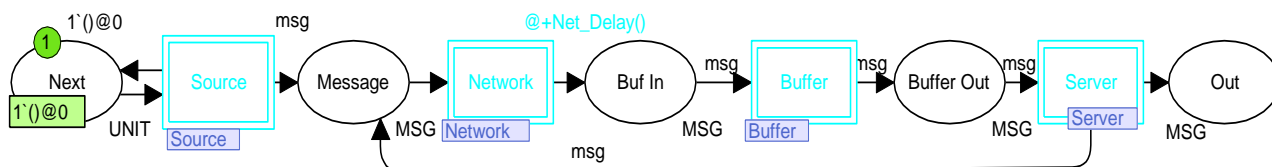


Рис. 3. Модель SIP-сети

Для описания цветов, переменных и функций используется язык CPN ML [11]:

- *INT* – стандартный цвет, представляющий целое число;

- *colset SIZE= int with 300..1000* - цвет, представляющий размер сообщения в байтах;
- *colset TYPE= with INV | nonINV1 | nonINV2 | nonINV3 | nonINV4 | nonINV5 | nonINV6* – цвет, представляющий тип сообщения;
- *colset BOOL= bool* – цвет, представляющий режим работы сервера (нормальный или в режиме перегрузки);
- *colset PRIO = int with 1..7* - цвет, представляющий приоритет сообщения;
- *colset MSG = product INT\*TYPE\*PRIO\*SIZE\*INT\*INT\*INT\*INT timed* – сложный цвет, представляющий содержание сообщения;
- *colset SetUpDelay= product INT\*INT* - цвет, представляющий задержку в установлении соединения;
- *colset Buffer = list MSG* - цвет, который представляет собой список маркеров цвета MSG;
- *colset Message = union f:MSG + avail* - сложный цвет, представляющий наличие или отсутствие свободных ресурсов на сервере.

Переменные, используемые в модели:

- *var R: SIZE* - переменная цвета SIZE, содержащая размер сообщения;
  - *var state:BOOL* - переменная цвета BOOL, содержащая состояние сервера;
  - *var buf: Buffer* - переменная цвета Buffer, которая содержит список маркеров цвета MSG, находящихся в очереди;
  - *var msg, newMsg: MSG* - переменная цвета MSG, содержащая сообщения, используемая для работы со списком маркеров цвета MSG;
  - *var i,ii: INT* - служебные переменные цвета INT;
  - *var ts, at, wt, pt: INT* - переменные цвета INT, содержащие время создания сообщения, время прибытия сообщения на сервер, время ожидания обработки сервером, время обработки сервером соответственно;
  - *var r: INT* - переменная цвета INT, которая содержит размер информации, находящейся в буфере очереди.
- Функции, используемые в модели:
- *fun SetType() = TYPE.ran()* - функция, выбирающая случайное число из диапазона, описанного цветом TYPE. Используется в генераторе трафика для задания типа сообщения при создании сообщения;
  - *fun SetPriority () = PRIO.ran()* - функция, выбирающая случайное число из диапазона, описанного цветом PRIO;
  - *fun Net\_Delay() = round (gamma (1.0,5.0))* - функция, определяющая задержку передачи сообщения по сети;
  - *fun SetSize(tp) = if tp = INV then 930 else 670* - функция, которая выбирает размер сообщения в зависимости от его типа. Используется в генераторе трафика для задания значения поля R при создании сообщения;
  - *fun cT() = IntInf.toInt(time())* - функция для получения текущего модельного времени.

– *fun startProc (MSG)* - функция, определяющая скорость обработки сообщений ресурсами сервера (листинг 1).

```

fun startProc ((i,tp,pri,R,ts,at,wt,pt):MSG) =
  let
    val new_at = ts
    val proc_time =
      if tp = INV andalso i <> 7777777
      then round(exponential(85.9/1000.0))
      else round(exponential(384.6/1000.0))
    val time_stamp =
      ModelTime.add(time(), ModelTime.fromInt(proc_time))
    val new_ts = IntInf.toInt(time_stamp)
    val new_wt = wt + (cT() - ts)
    val new_pt = pt + proc_time
  in (i,tp,pri,R,new_ts,new_at,new_wt,new_pt),proc_time)
end
    
```

Листинг 1. Тело функции *startProc (MSG)*

Трафик моделируется последовательностью маркеров, поступающих на блок генерации сообщений, цвета которых отвечают за тип сообщения *tp*, приоритет сообщения *pri*, размер сообщения *R*. Также вводится метка времени создания сообщения *ts*, которое используется для сбора статистики и анализа работы модели. Позиция *Count* служит для фиксации времени создания сообщения, для того чтобы в дальнейшем рассчитать время установления соединения.

Позиция *Next* задаёт закон распределения поступления сообщений в сеть. Трафиком можно управлять, задавая ему различные законы распределения, варьируя интенсивность и процентное соотношение запросов-ответов. На рис. 4 представлена модель блока генерации сообщений.

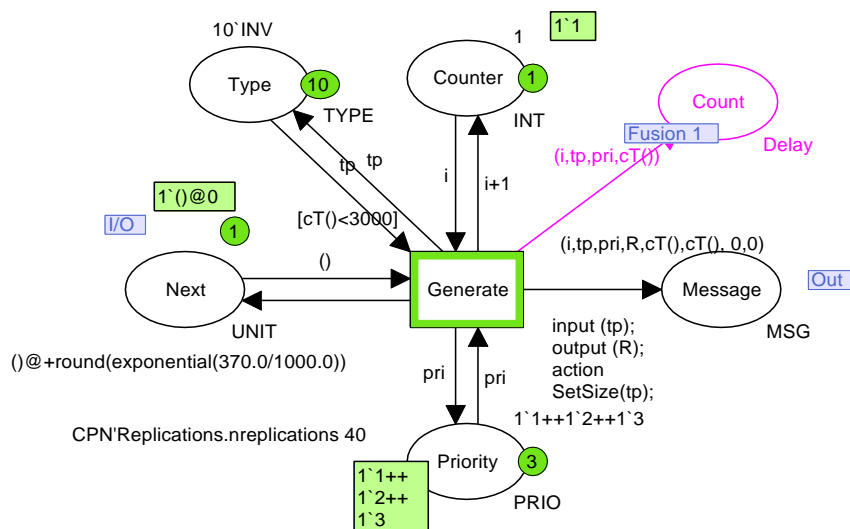


Рис. 4. Модель блока генерации сообщений

После того как сообщение сгенерировано, оно попадает в блок *Network*. В этом блоке можно с помощью элементов сетей Петри описать более подробно прохождение через такие сетевые элементы, как концентраторы и маршрутизаторы. В нашем случае этот блок просто вносит задержку, описанную функцией *Net\_Delay()*.

После прохождения по сети сообщение попадает на сервер SIP, где помещается в блок буфера *Buffer*, модель которого показана на рис. 5. Буфер необходим для минимизации потерь сообщений в периоды временных перегрузок, когда процессор сервера не справляется с обработкой сообщений. Какие сообщения будут обслужены, зависит от конкретного алгоритма и от приоритета, к которому принадлежит сообщение.

Перед тем, как поместить сообщение в очередь, производится проверка состояния сервера (позиция *Overload state*), находится ли он в нормальном состоянии работы или в состоянии перегрузки. Для того чтобы обнаружить перегрузку на сервере, вводим два пороговых значения *L* и *H*. Когда текущая загрузка буфера становится больше *H*, прокси-сервер переходит в состояние перегрузки и находится в этом состоянии до тех пор, пока текущая загрузка становится меньше порога *L*, что означает, что SIP-сервер выходит из режима перегрузки.

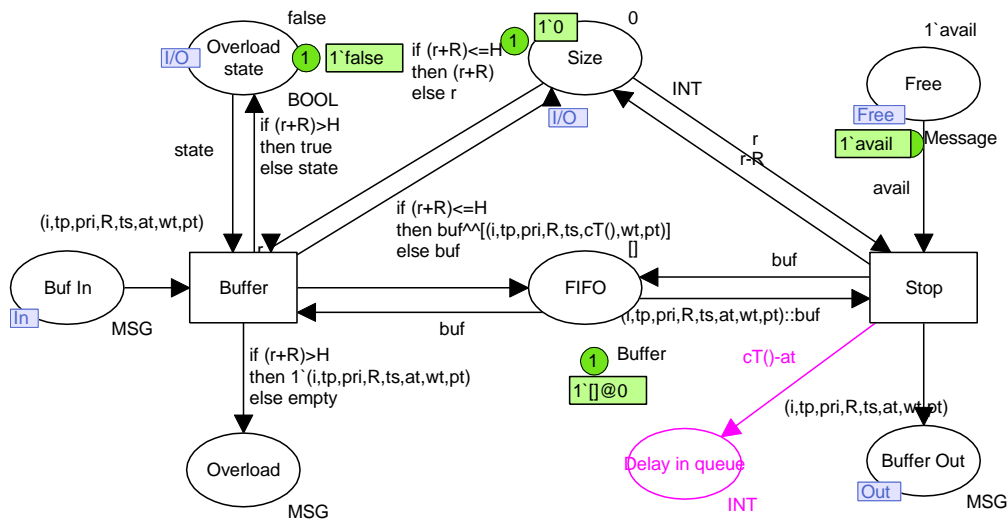


Рис.5. Модель буфера сервера

Если перегрузки нет, то сообщение помещается в очередь типа FIFO (переход в позицию *FIFO*). В позиции *FIFO* находится маркер с цветом *list MSG*, с помощью которого формируется список маркеров, подходящий для реализации модели очереди типа FIFO. В позиции *Size* находится маркер, в котором хранится суммарная загрузка буфера на текущий момент времени. После перехода сообщения в позицию *FIFO* его размер (значение поля *R*) прибавляется к значению маркера в позиции *Size*, также фиксируется время попадания сообщения в очередь, и поле *at* изменяется на текущее время. Сообщение может покинуть очередь при условии, что сервер в состоянии его обработать. Наличие свободного ресурса моделируется позицией *Free*, которая наполняется маркерами *avail* после обработки сообщений на сервере. Если сообщение освобождает очередь, его размер вычитается из размера позиции *Size*. В

служебной позиции *Delay in queue* записывается время нахождения сообщения в очереди. Можно менять этот блок буфера на другие алгоритмы управления очередями и формирования выходного трафика, которые являются основными механизмами борьбы с перегрузками в сетях.

После выхода из буфера сообщение обрабатывается в блоке *Server* (рис. 6). Время обработки для разных типов сообщений отличается и моделируется функцией *startProc(msg)*. Здесь же, в этом блоке, происходит формирование ответа на запрос в соответствии со сценарием установления соединения по протоколу SIP, описанным на рис.2. Для простоты описания все запросы и ответы, кроме запросов типа INVITE, относятся к сообщениям типа non-INVITE и помечаются переменной *nonINV*, принадлежащей цвету *TYPE*.

Для подсчета обработанных сообщений на сервере вводится позиция *Out*, для подсчета времени установления соединения вводится позиция *SetUpDelay*. Каждый запрос нумеруется уникальным числом, содержащимся в позиции *i*. Время отправления запроса хранится в позиции *Count*. Функция *cT()* определяет текущее модельное время. Позиция *Count* запоминает номер запроса *i* и время отправления запроса в сеть *at*.

Позиция *Delay* запоминает временные штампы всех сообщений, которые прошли все стадии установления соединения. В качестве времени установления соединения рассмотрен интервал времени между посылкой запроса и получением подтверждения ACK. Это значение сохраняется в позиции *SetUpDelay* для каждого обработанного запроса. Переход *Count* распознаёт принадлежность подтверждения на запрос к нужному запросу. Описатель дуги, соединяющей переход *Count* с позицией *SetUpDelay*, вычисляет время установления соединения ( $ts-at$ ).

### III. Оценка параметров модели

Построенная модель отлажена и протестирована в пошаговом режиме имитации динамики сети Петри. Для этих целей сообщения, генерируемые генератором трафика, трассировались через блоки сети SIP по пути к серверу, обрабатывались там и при необходимости формировался ответ в соответствии с рис. 3. Также проводилось наблюдение за поведением модели в процессе автоматической имитации с визуальным отображением динамики сети в режиме так называемой игры фишек и в режиме имитации без отображения промежуточных маркировок, целью которого является накопление статистической информации. В модели предусмотрены измерительные фрагменты, обеспечивающие накопление статистики о размере текущей длины очереди, задержки в очереди, задержек при обработке сообщений, среднего времени установления соединения.



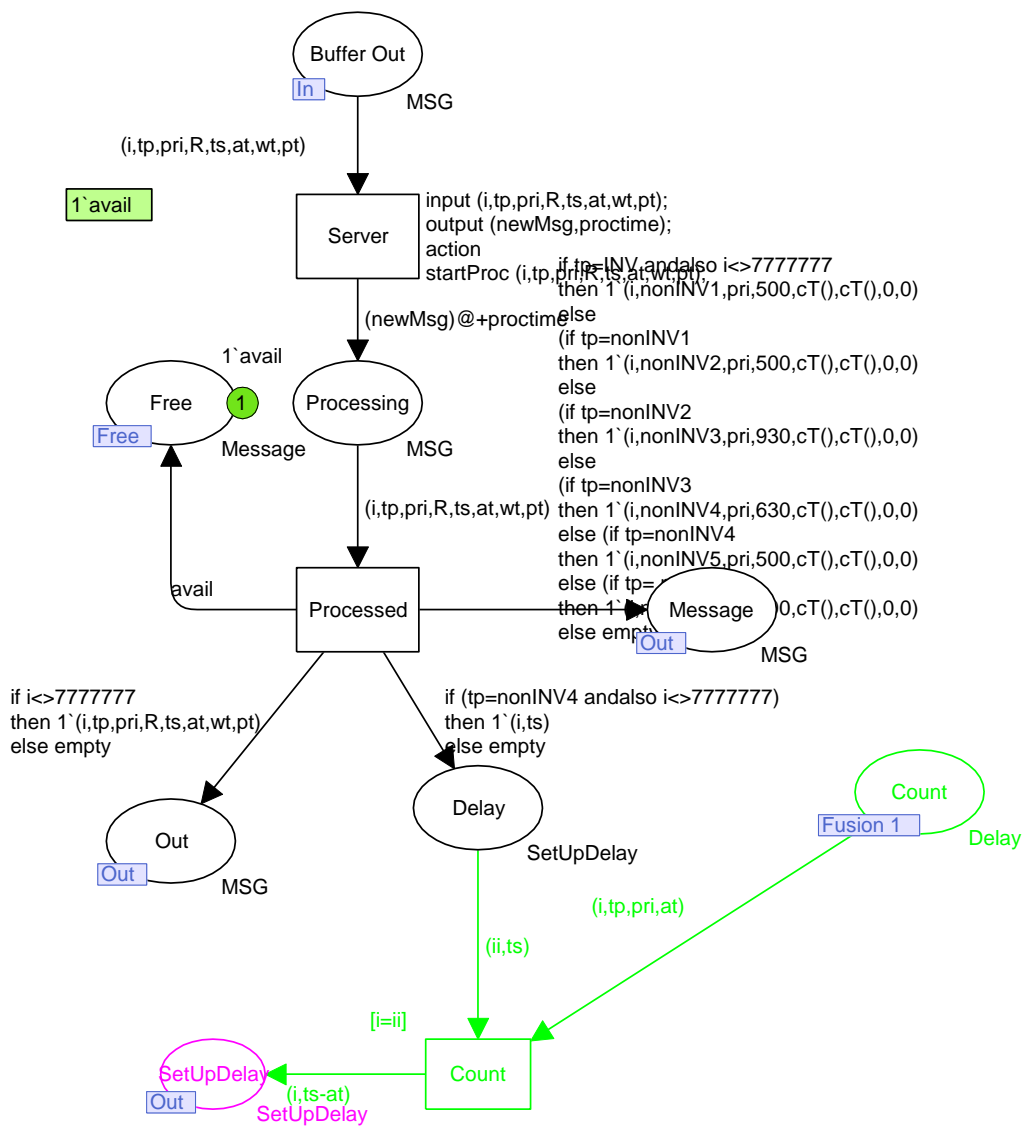


Рис. 6. Модель блока обработки сообщений

В табл. 1 представлены параметры описанной ранее модели. Данные были собраны на реальной сети крупного провайдера IP-телефонии в Украине. Значение единицы модельного времени соответствует 1 мс.

Таблица 1. Входные параметры модели сети SIP

| Параметр   | Значение                  |
|--|---------------------------|
| Среднее время обработки запроса INVITE сервером, мс          | 11,64                     |
| Среднее время обработки сообщения non- INVITE, мс            | 2,6                       |
| Размер буфера, байт  | 13000 и 130000            |
| Верхний порог обнаружения перегрузки, %                      | 80                        |
| Нижний порог обнаружения перегрузки, %                       | 60                        |
| Средние длины сообщений SIP(зависят от типа сообщения), байт | 500..950                  |
| Задержка в сети, мс  | Gamma(1.5, 5.0)           |
| Интенсивность входного потока, сообщ/с                       | Exponential (10.0..100.0) |

Результаты моделирования при разных входных интенсивностях поступления сообщений помещены в табл. 2. Измерение параметров сети SIP проводилось для разных размеров буферов серверов для сообщений, ожидающих обработки. Средняя пропускная способность сервера - 37 сообщ/с, т.е. значение интенсивности входного потока в 30 сообщ/с вынуждает сервер работать в режиме, близком к перегрузке, а интенсивность 40 сообщ/с и более переводит сервер в режим перегрузки. В этом режиме сервер не успевает обслужить все входящие сообщения, и растет число повторных передач.

Таблица 2. Параметры SIP-сети

| Параметры                                | Интенсивность входного потока, сообщ/с |                |              |              |              |              |              |              |
|--|--|----------------|--------------|--------------|--------------|--------------|--------------|--------------|
|  | 20<br>(мал.)*                          | 20<br>(бол.)** | 30<br>(мал.) | 30<br>(бол.) | 40<br>(мал.) | 40<br>(бол.) | 50<br>(мал.) | 50<br>(бол.) |
| Время установления соединения, мс        | 73,2                                   | 69,4           | 140          | 175          | 1084         | 2447         | 1367         | 4157         |
| Время обработки сообщений на сервере, мс | 3,9                                    | 3,9            | 3,88         | 3,89         | 3,69         | 3,84         | 3,5          | 3,55         |
| Задержка в очереди буфера, мс            | 6,2                                    | 5,7            | 31           | 25,2         | 56           | 358          | 71           | 642          |
| Коэффициент загрузки сервера             | 0,54                                   | 0,5            | 0,8          | 0,8          | 0,97         | 0,99         | 0,998        | 0,999        |
| Повторно переданные сообщения            | 0                                      | 0              | 32           | 0            | 617          | 180          | 1144         | 1020         |

\* - для малых размеров буфера (13000 байт);

\*\* - для больших размеров буфера (130000 байт).

Применение буферов больших размеров позволяет на некоторое время задержать рост повторных передач, но при этом значительно возрастает задержка установления соединения. При относительной нагрузке в 1,5 раза число повторно переданных сообщений при применении буферов малых и больших размеров почти одинаково, в то время как задержка установления соединения возрастает в 3 раза и более при использовании буфера большого размера, поэтому использование больших размеров буферов на SIP-серверах является нецелесообразным.

## Выводы

В статье предложена математическая модель процесса установления соединения в сети SIP в форме раскрашенных сетей Петри в среде моделирующей системы CPN Tools. Модель позволяет учитывать размер буфера сервера, пороговые значения на обнаружение перегрузки, количество процессоров для обработки сообщений и реализует дисциплину FIFO при работе с очередями. Исследованы зависимости времени установления соединения, задержки в очереди буфера, коэффициента загрузки SIP-сервера от интенсивности входного потока и от размеров буфера.

Анализ результатов имитационного моделирования показал, что применение буферов больших размеров позволяет на некоторое время задержать рост повтор-

ных передач, но это приводит к увеличению задержки установления соединения. При нагрузке  $\rho = 1,5$  число повторно переданных сообщений слабо зависит от размера буферов, в то время как задержка установления соединения при использовании буфера большого размера возрастает в 3 раза, поэтому использование больших размеров буферов на SIP-серверах является нецелесообразным.

Разработанные блоки модели, такие как генератор SIP-сообщений, блок буфера сервера, блок обработки сообщений, можно использовать в качестве компонентов для моделирования сетей передачи данных со сложной топологией, а также при исследовании и проектировании различного телекоммуникационного оборудования.

### Список литературы:

1. Rosenberg J. Requirements for Management of Overload in the Session Initiation Protocol, IETF RFC 5390, 2008. – 14 p.
2. Иевлева Т. В., Журавлев С. В. Обнаружение и предотвращение перегрузки оборудования SoftSwitch при регистрациях SIP-телефонов // Электросвязь. – 2007. – № 17. – С. 16 – 19.
3. Hong Y. Controlling Retransmission Rate for Mitigating SIP Overload // Proceedings of IEEE International Conference on Communications (ICC 2011), 2011. – P. 1 – 5.
4. Ohta M. Overload Control in a SIP Signaling Network // Proceedings of World Academy of Science, Engineering and Technology. – 2006. – Vol. 12. – P. 205 – 210.
5. Noel E., Johnson C. Novel Overload Controls for SIP Networks // 21st International Teletraffic Congress, 2009. – P. 1 – 8.
6. Homaouni M., Nemati H. Controlling Overload in SIP Proxies: An Adaptive Window Based Approach Using No Explicit Feedback // Proceedings of IEEE Globecom, 2010. – P. 1 – 5.
7. Hilt V., Widjaja I. Controlling overload in networks of SIP servers // Proceedings of IEEE International Conference on Network Protocols, 2008. – P. 83 – 93.
8. Кашин М.М. Методы борьбы с перегрузками в сети SIP // Инфокоммуникационные технологии. – 2011. – Том 9, № 1. – С. 67 – 70.
9. Rosenberg J., Schulzrinne H. SIP: Session Initiation Protocol, IETF RFC 3261, June 2002. – 269 p.
10. Гольдштейн В.С., Пинчук А.В., Суховицкий А.Л. IP-телефония. – М.: Радио и связь, 2001. – 336 с.
11. Jensen K. Coloured Petri nets: basic concepts, analysis methods and practical use. – Berlin: Heidelberg, New York: Springer-Verlag, 1996. – 234 p.
12. Зайцев Д.А., Шмелёва Т.Р. Моделирование телекоммуникационных систем в CPN Tools: Учебное пособие. – Одесса:ОНАС им. А. С. Попова, 2008. – 68 с.
13. Кизилов Е.А., Пушкарев В.А. Моделирование самоподобного трафика цветными сетями Петри // Новые информационные технологии и системы: труды IX Международной научно-технической конференции в 2 ч. – Пенза: Изд-во ПГУ, 2010. – Ч. 2. – С. 73 – 77.
14. Лысяк Т.Н., Кузьминых Е.Д. Исследование методов профилирования трафика в условиях нестационарной нагрузки // Радиотехника: Всеукр. науч.-техн. сб. – 2007. – Вып.148. – С. 67 – 79.