УДК 621.39

Гистерезисное управление сигнальной нагрузкой в сети SIP-серверов

П. О. Абаев, Ю. В. Гайдамака, К. Е. Самуйлов

Кафедра систем телекоммуникаций
Российский университет дружбы народов
ул. Миклухо-Маклая, 6, г. Москва, Россия, 117198

В статье, являющейся по сути обзором, исследуются механизмы управления нагрузкой в сетях сигнализации, которые используют три типа порогов для контроля перегрузок. Целью обзора является анализ механизмов и моделей контроля перегрузок SIP-серверов. В основе исследований лежит гистерезисное управление нагрузкой, которое исходно было разработано для общеканальной системы сигнализации №7. Разработаны унифицированные методы описания процедур гистерезисного управления сигнальной нагрузкой. Исследовано современное состояние и проблемы базового механизма контроля перегрузок SIP-серверов, предложенного комитетом IETF. Изложены подходы к построению математических моделей SIP-серверов в виде систем массового обслуживания с гистерезисным управлением.

Ключевые слова: сеть сигнализации, ОКС7, SIP-сервер, гистерезисное управление, механизм контроля перегрузок, пороговое управление.

1. Введение

Пороговое управление нагрузкой является основным инструментом в предотвращении различного рода перегрузок в телекоммуникационных сетях [1–4]. Одним из механизмов является гистерезисное управление [5], которое использует три типа порогов для контроля перегрузок – порог обнаружения перегрузки, порог снижения перегрузки и порог сброса нагрузки. Разновидности этого механизма применяются при обнаружении перегрузок как в сетях общеканальной системы сигнализации №7 (ОКС7) [6–9], так и в сетях, где основой сигнализации является протокол инициации сеансов связи (SIP, Session Initiation Protocol) [9–12].

Основной целью обзора является анализ механизмов контроля перегрузок SIP-серверов. Основные положения этих механизмов определены в стандартах комитета IETF (Internet Engineering Task Force) [3, 4, 10], а в статьях [13–26] показано, что они построены по принципам гистерезисного управления.

Для сетей сигнализации гистерезисное управление было разработано Международным союзом электросвязи (МСЭ) для протоколов ОКС7 (SS7, Signalling System No.7) в стандартах серии Q.700 [1,2]. В рекомендации Q.704 [1] определены три типа перегрузок: перегрузка звена сети, перегрузка маршрута и перегрузка пункта сигнализации. Обработка перегрузки в ОКС7 включает в себя два этапа — обнаружение перегрузки и действия по ее предотвращению. Для обнаружения перегрузки производится контроль числа сообщений в очереди буфера передачи, а действия по предотвращению перегрузки заключаются в ограничении поступающей сигнальной нагрузки. В разделе 1 обзора показано, что в ОКС7 обнаружение перегрузки осуществляется введением в буфере передачи порога обнаружения перегрузки, а действием по предотвращению перегрузки является снижение нагрузки. Кроме этого, для обнаружения перегрузки предусмотрен также порог сброса, при этом действием по предотвращению перегрузки является сброс нагрузки. Во избежание осцилляции очереди при снижении заполненности буфера ниже значения порога обнаружения перегрузки возврат к нормальной нагрузке происходит не сразу, а спустя некоторое время. Для этого в буфере передачи определен порог снижения перегрузки. Соответствующий механизм и называют гистерезисным управлением нагрузкой. Отметим, что для порога сброса нагрузки этот механизм не применяется.

Статья поступила в редакцию 15 августа 2011 г.
Работа выполнена при частичной поддержке РФФИ (грант 10-07-00487-а).

Раздел 2 обзора посвящен исследованию механизмов контроля перегрузок SIP-серверов. Во-первых, мы детально обсуждаем типовые примеры обнаружения перегрузок, проблемы разработки механизмов и требования к механизмам контроля согласно известным на сегодняшний день документам IETF [3,4]. Во-вторых, вводятся типы управления перегрузками и классифицируются механизмы контроля перегрузок SIP-серверов. Заметим, что в документах IETF понятие гистерезисного управления в явном виде не определено, но в ряде статей [13–26] исследованы именно такие механизмы по предотвращению перегрузок SIP-серверов. В этих статьях предложены также различные модели для анализа параметров и индикаторов управления перегрузками. С использованием понятия гистерезисного управления нагрузкой, введенного для ОКС7, мы с единых позиций строим типовые модели управления перегрузками SIP-серверов в виде систем массового обслуживания (СМО). Завершается статья исследованием одной СМО, разработанной авторами статьи для анализа управления перегрузками SIP-серверов с учетом общей длины очереди сигнальных сообщений.

2. Управление перегрузками звена ОКС7

Управление сигнальным трафиком и обнаружение перегрузок звена сети ОКС7 основано на процедуре гистерезисного управления нагрузкой, причем одной из первых публикаций в этой области является статья [27]. Данная процедура реализует контроль за состоянием очереди в буфере передачи и управление сигнальным трафиком на основании данных контроля.

В ОКС7 для определения статуса перегрузки используют три порога — порог H_1 обнаружения перегрузки (англ., onset), порог L_1 снижения перегрузки (англ., abatement) и порог R_1 сброса нагрузки (англ., discard), а статусы перегрузки определены для международной и национальной версий системы. Всего используются четыре значения статуса перегрузки:

$$h = \begin{cases} 0, & \text{нормальная нагрузка,} \\ i, & \text{уровень } i \text{ перегрузки,} \quad i = 1, 2, 3. \end{cases}$$

В ОКС7 используется также статус сброса нагрузки, определённый только для национальной версии системы, при этом используются три значения статуса сброса нагрузки:

$$r = \begin{cases} 0, & \text{нет сброса,} \\ i, & \text{уровень } i \text{ сброса,} \quad i = 1, 2. \end{cases}$$

На рис. 1 показан процесс изменения статуса перегрузки в зависимости от длины очереди для случая $h \in \{0, 1\}$. В начальном состоянии значение статуса перегрузки $h = 0$, и это значение сохраняется до тех пор, пока заполненность буфера не достигает значения H_1 порога обнаружения перегрузки. Когда длина очереди в буфере достигла значения H_1, значение статуса перегрузки изменяется на $h = 1$. При дальнейшем росте очереди до порога R_1 сброса нагрузки, а также при ее уменьшении до порога L_1 снижения перегрузки значение статуса перегрузки не изменяется, т.е. $h = 1$. При уменьшении заполненности буфера до порога L_1 снижения перегрузки значение статуса перегрузки опять изменится на $h = 0$.

На рис. 2 показана качественная зависимость интенсивности $\lambda(h, r, n)$ сигнальной нагрузки от длины n очереди в буфере передачи при процедуре гистерезисного управления в ОКС7. При обнаружении перегрузки при длине очереди $n = H_1$ нормальное значение λ интенсивности нагрузки снижается до величины λ'. В случае, когда длина очереди достигает значения $n = R_1$, происходит сброс нагрузки, т.е. $\lambda(h, r, n){=}0$ для $n \geqslant R_1$. При последующем уменьшении очереди до значения $n < R_1$ величина интенсивности нагрузки восстановится до значения λ' и сохранится до момента достижения длиной очереди значения $n = L_1$, когда восстанавливается нормальное значение интенсивности нагрузки λ.

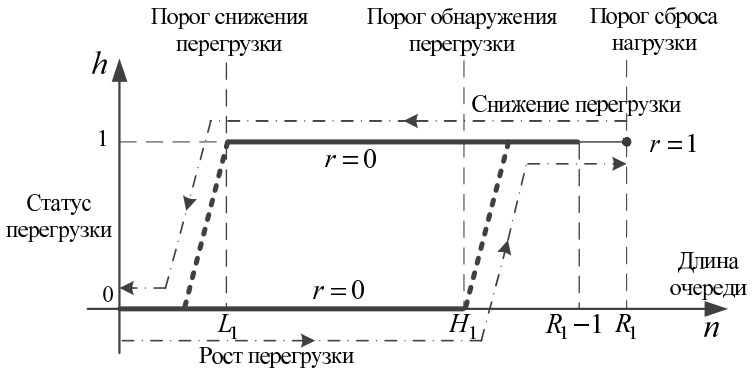

Рис. 1. **Изменение статуса перегрузки**

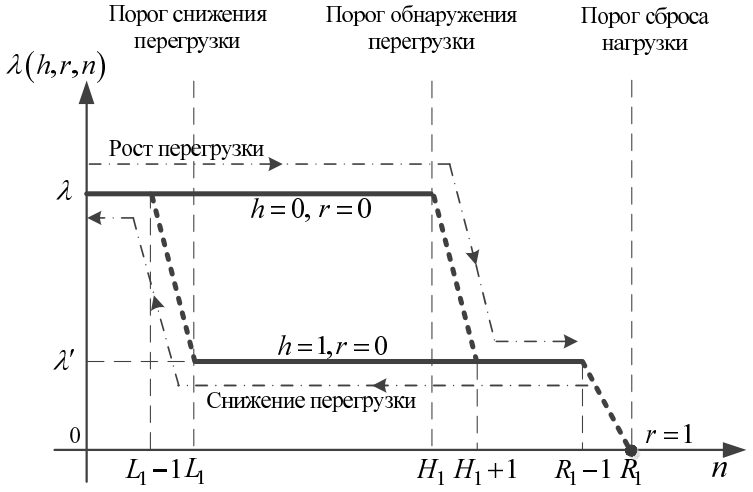

Рис. 2. **Гистерезисное управление сигнальной нагрузкой**

Рекомендациями МСЭ [1,2] предусмотрена возможность реализации нескольких групп порогов, как это показано на рис. 3. Заметим, что на рисунке $R_i < L_{i+1}$, хотя возможен и другой случай ($R_i > L_{i+1}$), который в обзоре не рассматривается для краткости изложения.

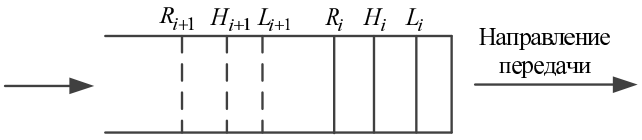

Рис. 3. **Группы порогов в буфере передачи**

На рис. 4 показан процесс изменения статуса перегрузки для двух групп порогов, т.е. $h, r \in \{0, 1, 2\}$. Не приводя детальных описаний процесса изменения статусов h и r, как это было сделано для рис. 1, мы вынесли все необходимые и достаточно очевидные пояснения на график рис. 4. Следует обратить внимание на то, что для $h = 1$ значение статуса сброса изменяется с $r = 0$ на $r = 1$ при достижении длиной очереди значения $n = R_1$. Аналогично, для $h = 2$ значение статуса сброса изменяется с $r = 1$ на $r = 2$ при достижении длиной очереди значения $n = R_2$. Из рис. 4 видно, что в рассматриваемом нами случае для двух

групп порогов при управляющих воздействиях возникают две петли гистерезиса, как показано на рис. 5.

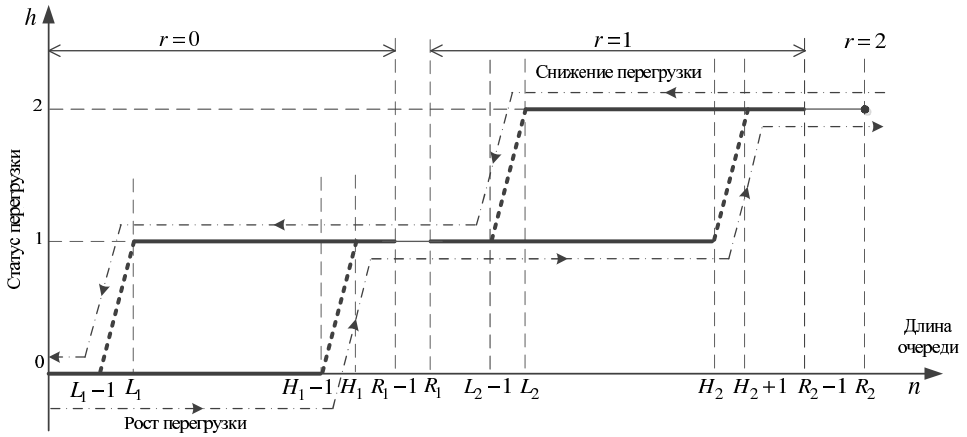

Рис. 4. **Изменение статуса перегрузки для двух групп порогов**

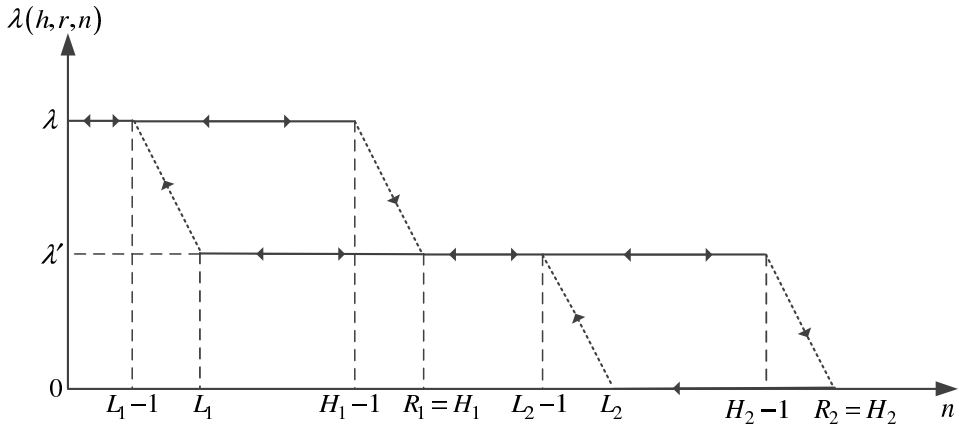

Рис. 5. **Двухуровневое гистерезисное управление нагрузкой для случая** $R_1 = H_1$, $R_2 = H_2$

Качество управления перегрузками может быть оценено по ряду параметров, например, по вероятности пребывания системы в состоянии нормальной нагрузки. Кроме того, оценка может быть дана и в более общем смысле — по некоторым ключевым индикаторам эффективности управления [3, 4], которые не являются предметом анализа в данном обзоре, и мы приведем для них лишь простые примеры. Например, в ОКС7 при возникновении перегрузки на маршруте необходимо перенаправить избыточную нагрузку по альтернативному маршруту с целью предотвращения перегрузки буфера передачи в случае пиковой нагрузки в транзитном пункте сигнализации. В противном случае буфер будет переполнен, а избыточная нагрузка сброшена вместо того, чтобы быть перенаправленной на другой маршрут. Ясно, что пороги в буфере должны быть выбраны так, чтобы управляющее воздействие было реализовано не на пике перегрузки, а в его начале. Аналогично, управляющий механизм должен быстро реагировать на снижение перегрузки и возвращать систему в режим нормальной нагрузки. Ещё одним примером индикатора эффективности гистерезисного управления нагрузкой является наличие осцилляции очереди около пороговых значений в буфере передачи.

Рассмотрим теперь пример вероятностных параметров качества гистерезисного управления нагрузкой [6,27–32], показанного на рис. 5. Под состоянием системы будем понимать тройку $(h, r, n) \in \mathcal{X}$ и представим множество всех состояний $\mathcal{X}$ в виде разбиения $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1 \cup \mathcal{X}_2$, где $\mathcal{X}_0$ — множество состояний нормальной нагрузки, $\mathcal{X}_1$ — множество состояний перегрузки уровня $i = 1$, $\mathcal{X}_2$ — множество состояний перегрузки уровня $i = 2$. Нетрудно убедиться, что

$$\mathcal{X}_0 = \{(h, r, n) : h = 0, r = 0, 0 \leqslant n < H_1\};$$

$$\mathcal{X}_1 = \mathcal{X}_{11} \cup \mathcal{X}_{12} = \{(h, r, n) : h = 1, r = 0, L_1 \leqslant n < R_1\} \cup \cup \{(h, r, n) : h = 1, r = 1, R_1 \leqslant n < H_2\};$$

$$\mathcal{X}_2 = \mathcal{X}_{21} \cup \mathcal{X}_{22} = \{(h, r, n) : h = 2, r = 1, L_2 \leqslant n < R_2\} \cup \cup \{(h, r, n) : h = 2, r = 2, n = R_2\}.$$

Теперь можем определить искомые параметры качества управления — вероятности $P_i = P(\mathcal{X}_i)$, $i = 0, 1, 2$. Ясно, что чем больше вероятность P_0 пребывания системы в множестве состояний нормальной нагрузки, тем выше качество ее функционирования, т.е. одним из условий эффективного функционирования системы является выполнение соотношения $P(\mathcal{X}_0) \gg P(\mathcal{X}_1) + P(\mathcal{X}_2)$. С другой стороны, вероятность сброса всего трафика определяется формулой $\pi = P(r = 2)$ и ясно, что эта вероятность должна быть минимизирована.

Если известны ограничения на параметры качества функционирования системы, например, $P(\mathcal{X}_i) \leqslant P_i^*, \pi \leqslant \pi^*$, то из этой системы неравенств могут быть найдены конкретные значения порогов перегрузки L_i и H_i и сброса нагрузки R_i, $i = 1, 2$.

Итак, мы кратко изложили общие принципы гистерезисного управления нагрузкой в ОКС7 и ввели все необходимые понятия, которые будут использованы в следующих разделах обзора, посвященных исследованию механизмов контроля перегрузок SIP-серверов.

3. Контроль перегрузок SIP-серверов

Протокол инициирования сеансов связи является основным протоколом установления соединения в сетях следующих поколений. Протокол SIP разработан группой MMUSIC (Multiparty Multimedia Session Control) комитета IETF, а спецификации протокола представлены в документе RFC 3261 [10]. Фрагмент сети SIP на рис. 6 состоит из агентов пользователя (UA, User Agents) и двух прокси-серверов. Перед установлением сессии между вызывающей и вызываемой сторонами агенты пользователей UAc (UA client) и UAs (UA server) должны быть зарегистрированы на своих прокси-серверах с помощью запросов REGISTER. Как показано на рис. 6, установление соединения по протоколу SIP состоит из следующих шагов [9, 11, 12]:

- UAc инициирует соединение отправкой запроса INVITE прокси-серверу 1;
- запрос INVITE передается через прокси-сервер 2 агенту пользователя UAs;
- прокси-сервер 2 и прокси-сервер 1 отвечают UAc сообщением 100 Trying, что запрос INVITE принят к обработке;
- UAs уведомляет прокси-серверы и UAc сообщением 180 Ringing о готовности к установлению соединения;
- после ответа вызываемого абонента UAs сообщением 200 OK уведомляет прокси-серверы и UAc о том, что запрос на установление соединения успешно выполнен;
- UAc сообщением ACK подтверждает прием ответа 200 OK на запрос INVITE и между UAc и UAs устанавливается сессия для обмена медиа потоками;

– после окончания сессии UAc посылает UAs через прокси-серверы запрос BYE на завершение сеанса связи;
– UAs прекращает передачу медиа потока и подтверждает свои действия отправкой UAc сообщения 200 OK.

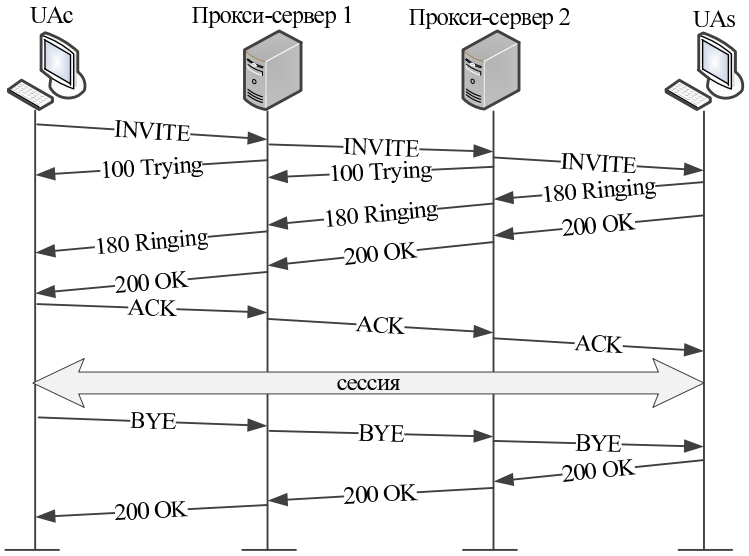

Рис. 6. **Диаграмма установления и разъединения сессии по протоколу SIP**

С ростом числа пользователей услуг, предоставляемых на базе протокола SIP, возникают различного рода перегрузки SIP-серверов из-за отсутствия достаточных ресурсов для установления и завершения сессий между агентами пользователей. Различают два типа перегрузок — перегрузки типа «клиент-сервер» и типа «сервер-сервер» [3]. Перегрузки «клиент-сервер» возникают в серверах-отправителях из-за избыточной нагрузки, создаваемой группами SIP-терминалов, обозначенных на рис. 7 как UAc.

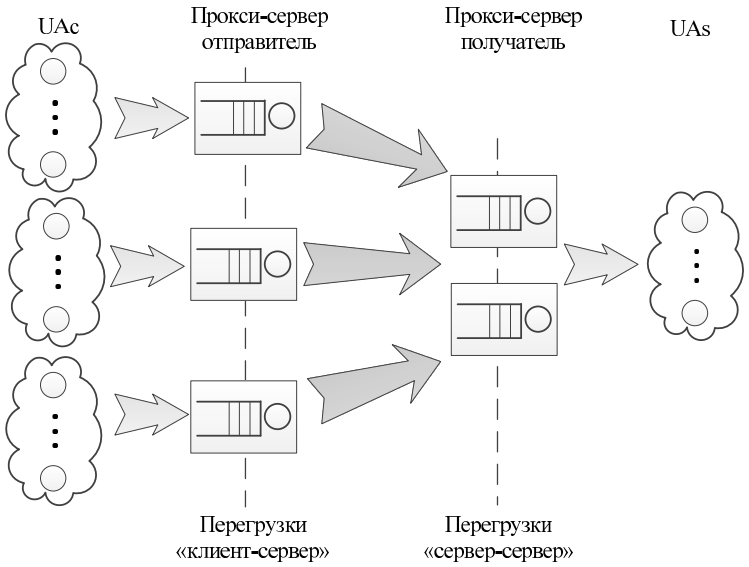

Рис. 7. **Типы перегрузок по топологии соединений**

Примером перегрузки «клиент-сервер» служит так называемый лавинный перезапуск, который происходит, когда большое число UAc пытаются зарегистрироваться на серверах-отправителях. Одним из примеров является сценарий «Манхэттенского перезапуска» (англ., «Manhattan Reboots» scenario), когда в результате аварии произошло отключение электричества в этом крупнейшем районе города, и после восстановления электроснабжения все SIP-терминалы одновременно пытались зарегистрироваться на серверах, создав тем самым большой поток сообщений REGISTER.

Отметим, что проблема перегрузки «клиент-сервер» может быть решена оператором сети связи простым увеличением числа SIP-серверов, но кроме того, в 2009 году в RFC 5626 [33] в протокол SIP были внесены изменения в части механизма предотвращения лавинных подключений UAc к серверу, которые во многом решили проблему. Поэтому далее мы рассматриваем только перегрузки типа «сервер-сервер», которые могут возникать в ситуации, когда большое число вызовов начинает одновременно поступать на один номер (UAs). Примером такой ситуации является участие пользователей в телеголосовании или их реакция на рекламный ролик, сообщающий о том, что первые дозвонившиеся пользователи получат ценный подарок.

В протоколе SIP в части механизма контроля перегрузок до сих пор имеются существенные недоработки. Далее мы приведем описание этого механизма, после чего в соответствии с RFC 5390 [3] кратко сформулируем проблемы, которые этот механизм не позволяет разрешить.

Базовый механизм из RFC 3261 [10] (далее механизм 503) в случае перегрузки прокси-получателя предусматривает отправку прокси-отправителю сообщения 503 Service Unavailable. В случае, когда сервер не может обработать запрос из-за временной перегрузки, ему следует отклонить этот запрос с кодом ошибки 503. Отправителю, получившему сообщение 503, следует действовать так, как в случае получения сообщения 500 Server Internal Error, и не следует перенаправлять сообщение с кодом 503 далее вверх по цепочке SIP-серверов, уведомляя их о перегрузке на одном из ниже лежащих серверов, до тех пор, пока прокси-отправитель не сможет определить, что прокси-получатель будет отвечать на каждое его сообщение сообщением с кодом 503. Прокси-отправитель должен повторить исходное сообщение прокси-получателю или направить его на другой сервер. Перегруженный прокси-получатель также может добавить заголовок Retry-After в сообщение с кодом 503, указав время в секундах, в течение которого он не хочет получать каких-либо сообщений от отправителя. Отправитель на этот период времени прекращает направлять сообщения получателю, вместо этого сообщения направляются на альтернативные сервера. Отправка сообщений на сервер, сообщивший о перегрузке, возобновляется по истечении интервала времени, указанного в заголовке Retry-After сообщения 503. Отметим, что RFC 3261 предусматривает в случае перегрузки возможность получателю сбрасывать поступающие сообщения без уведомления отправителя.

Перейдем к описанию проблем, возникающих в результате применения механизма 503 контроля перегрузок SIP-серверов [3]. Заметим, что в документах IETF эти проблемы до сих пор не решены.

Проблема усугубления перегрузки (англ., load amplification) заключается в тенденции значительно увеличивать нагрузку в периоды перегрузок, тем самым вызывая дальнейшее усугубление проблемы и приближая момент обвала сети. На рис. 8 изображен пример сети [4], состоящий из сервера C1 балансировки нагрузки (англ., load balancer) и трех серверов получателей — C2, C3, C4.

C1 получает запросы от большого числа SIP-клиентов (прокси-серверов и/или UAc) и, функционируя в режиме балансировки нагрузки, распределяет поступающие сообщения на C2, C3 и C4. Рассмотрим ситуацию, когда все три сервера находятся в состоянии перегрузки. Получив от C1 очередное сообщение, C2 должен ответить сообщением 503 Service Unavailable. Поскольку C2 перегружен, то на генерацию и отправку сообщения с кодом 503 ему требуется некоторое время. В случае, когда протокол SIP функционирует поверх протокола UDP, следует ожидать ретрансляций сообщений, которые увеличат нагрузку на C2. Даже в

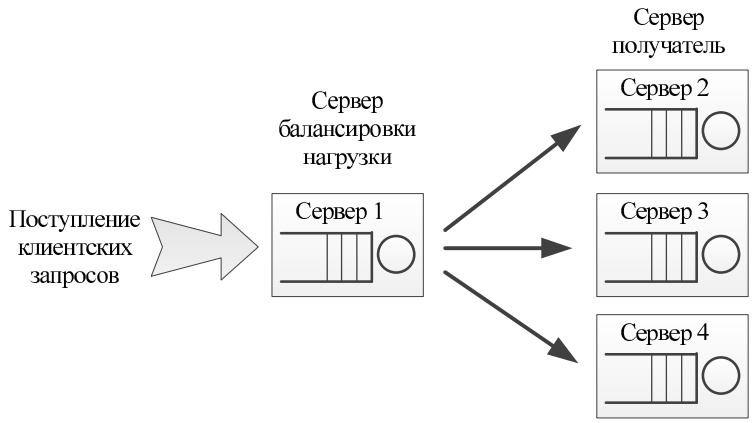

Рис. 8. **Фрагмент сети с сервером балансировки нагрузки**

случае функционирования поверх протокола TCP перегруженный сервер может не успеть отправить подтверждение на TCP-сообщение, тем самым усугубив перегрузки из-за ретрансляций сообщений. C1, получив сообщение с кодом 503 от C2, должен перенаправить сообщение на C3 или C4, которые также находятся в состоянии перегрузки и могут вызвать нежелательные дополнительные ретрансляции сообщений по такому же сценарию, как в случае с C2. Итак, обработка одного запроса от SIP-клиента, который поступил на C1 и будет направлен на один из серверов C2, C3,C4, влечет за собой четыре транзакции (SIP-клиент-C1, C1-C2, C1-C3, C1-C4), каждая из которых в случае протокола UDP может вызвать до семи ретрансляций исходного запроса. В случае нормальной загрузки сети один запрос от SIP-клиента может сгенерировать один запрос (C1-C2, или C1-C3, или C1-C3) и два ответа (C2-C1, или C3-C1, или C4-C1, и C1-SIP-клиент). Но, когда сеть перегружена, один запрос от SIP-клиента до истечения таймера ожидания ответа на этот запрос может сгенерировать до 21 запроса и 21 ответа в случае протокола UDP. Ситуация лучше, когда используется протокол TCP, но даже при отсутствии ретрансляции TCP-сегмента один запрос от SIP-клиента может сгенерировать 3 запроса и 4 ответа. Как в случае протокола UDP, так и в случае протокола TCP, итак перегруженным серверам приходится выполнять дополнительную работу. Следовательно, механизм 503 эффективен, когда перегружен отдельный сервер, но в случае перегрузки сети серверов механизм приводит к увеличению нагрузки и может вызвать полный отказ в обслуживании сообщений.

Проблема неполного использования кластера SIP-серверов (англ., underutilization). В RFC 3261 не прописано, как получатель сообщения с кодом 503 должен на него реагировать, более того, существуют конфигурации сети, в которых отсутствует возможность четкой идентификации отправителя этого сообщения. В некоторых реализациях идентификация отправителя производится не по IP-адресу сервера, а по его доменному имени. Для рассмотренного выше примера (рис. 8) предположим, что C2, C3, C4 образуют кластер, и при обращении по URI сервер DNS возвращает адрес одного из них. Если сервер из кластера при обращении к нему ответит сообщением с кодом 503, то для C1 это означает, что весь кластер C2-C4 находится в состоянии перегрузки. Следовательно, C1 перестанет направлять сообщения на весь кластер, хотя в нем есть не перегруженные серверы.

Проблема использования сообщения 503 с заголовком Retry-After (англ., Retry-After Problem). Механизм 503 определяет для сервера-отправителя только два состояния — состояние отправки сообщений, когда сервер-отправитель передает всю нагрузку без ограничений, и состояние ожидания отправки, когда сервер-отправитель не передает ничего до истечения таймера (англ., all-or-nothing

technique). В сети с сервером балансировки нагрузки даже в случае незначительной перегрузки одного из серверов-получателей могут возникнуть значительные осцилляции. Рассмотрим сеть на рис. 3, но без C4, в предположении, что C2 перегружен. Тогда C1, отправляя любое сообщение C2, в ответ получает сообщение 503 с заголовком Retry-After. Согласно механизму 503 C1 прекращает отправлять сообщения C2 и перенаправляет всю нагрузку, которую он ранее делил между C2 и C3, серверу C3. Это приводит к перегрузке C3, который также отвечает сообщением 503 с заголовком Retry-After. C1 отклоняет поступающие на него сообщения до тех пор, пока не истечет время запрета на отправку сообщений C2. По истечении этого времени C1 направит все запросы, которые он ранее делил между C2 и C3, серверу C2, который снова окажется в состоянии перегрузки. Этот процесс может продолжаться циклически. Считается [4], что механизм 503 эффективен, когда за балансировщиком стоит большое число серверов-получателей, тогда он может работать эффективно и нагружать сервера равномерно, не вызывая перегрузок.

Проблема неоднозначного использования сообщения 503 (англ., Ambiguous Usages). В стандарте четко не прописаны случаи, когда сервер получатель должен отвечать сообщением с кодом 503. В различных реализациях сообщение 503 используется для индикации разных состояний. Например, в RFC 3398 [34] определено, что шлюз сигнализации отправляет сообщение 503 в ответ на сообщения о невозможности обработать запрос, которые не обязательно означают, что шлюз перегружен. Такая неоднозначность создает дополнительные трудности и негативно влияет на производительность узлов сети, вынуждает вводить в базовый механизм дополнительное разделение сообщения 503 на типы в зависимости от ситуации. Также остается неясным вопрос о необходимости ретрансляции запросов, на которые получено сообщение 503.

Сформулируем теперь основные требования [3] к механизмам, отвечающие сформулированным выше проблемам.

- Механизмы контроля перегрузок должны поддерживать производительность серверов на приемлемом уровне с учетом требований к заданному показателю качества обслуживания. Минимальное значение производительности сервера является критическим значением для оценки эффективности механизма контроля перегрузок. Существенно, что в случае ретрансляции сообщений по причине перегрузки получателя сервер-отправитель должен направлять сообщения только на серверы, работающие в нормальном режиме.
- В случае сбоя, приводящего к перегрузке сервера, механизм должен сглаживать негативное влияние на работу смежных серверов. Это поможет избежать полного или частичного простоя участка сети и восстановить нормальную работу.
- Сообщения, уведомляющие о перегрузке сервера, должны позволять четко определить причину отправки сообщения, например, перегрузка смежного сервера или сбой, произошедший не по причине перегрузки.
- Механизм должен обеспечивать возможность ограничения поступающей на сервер нагрузки, сброс нагрузки должен производиться не сразу, а в несколько приемов.
- Механизм должен обеспечивать возможность взаимодействия серверов, находящихся в различных доменах.
- Механизм не должен определять приоритеты на обработку сообщений соседних серверов, а должен назначать их в соответствии с локальной политикой в зависимости от типов поступающих сообщений, например, экстренные вызовы.
- Механизм должен однозначно идентифицировать ситуации, в которых отправители будут проводить ретрансляцию сообщений, количество управляющих сообщений должно быть минимально.
- Механизм должен обеспечивать возможность однозначно определять отправителя сообщения о перегрузке.
- Механизм должен обеспечивать возможность работы сети SIP-серверов с сервером балансировки нагрузки.

Заметим, что в зависимости от метода, на основании которого сервер-отправитель определяет состояние сервера-получателя и управляет нагрузкой, механизмы контроля перегрузок делятся на явные (англ., explicit) и неявные (англ., implicit).

Различают [4] три типа управления, схематично показанные на рис. 9.

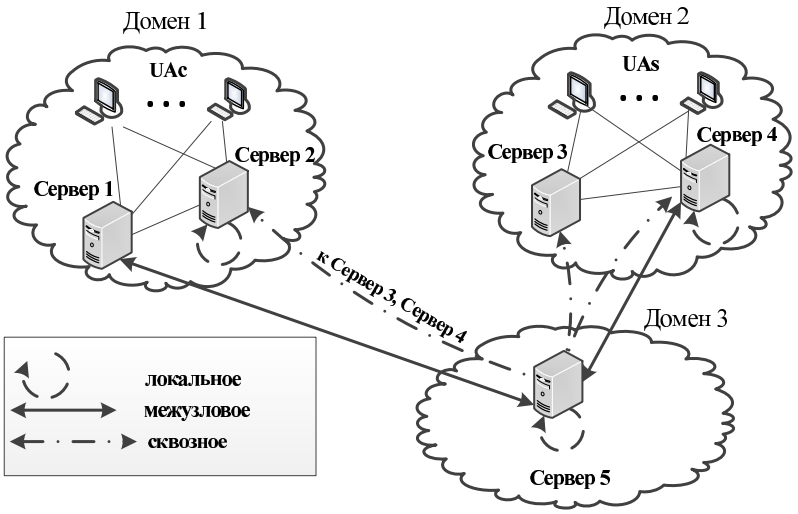

Рис. 9. **Межузловое, сквозное и локальное управление перегрузками**

Локальное управление (англ., local). Механизм локального контроля перегрузок предполагает, что сервер на основе мониторинга текущего уровня использования своих ресурсов принимает решение о сбросе части нагрузки. Идея этого механизма заключается в том, что серверу требуется меньше ресурсов на сброс сообщения, чем на его обработку. Однако сброс сообщений приведет к их ретрансляции, и, следовательно, сервер, продолжая сбрасывать сообщения, будет затрачивать ресурсы на обработку этих сообщений и не решит проблему перегрузки. Поэтому локальное управление не может самостоятельно справиться с возрастающей нагрузкой, а, следовательно, не может использоваться в качестве единственного механизма управления. Рекомендуется применять для контроля поступающей нагрузки другие механизмы управления, а механизм локального управления должен использоваться как мера последней необходимости.

Межузловое управление (англ., hop-by-hop). Идея состоит в создании отдельного контура управления для каждой пары смежных серверов, причем контуры управления каждой пары независимы друг от друга. На рис. 9 показаны два контура С1–С5, С4–С5, причем каждый контур управляет одним транзитным участком (англ., hop). Управляющие сообщения, полученные от сервера-получателя, не передаются вверх по всей цепочке серверов. Вместо этого сервер выполняет управляющее действие на основе полученного сообщения, например, полностью или частично сбрасывает нагрузку. Если сервер отправитель определяет состояние перегрузки, то он направляет управляющее сообщение вверх по цепочке и так далее. Таким образом, механизм межузлового управления является простым и масштабируемым решением, предусматривает контроль перегрузок только между соседними узлами, а также позволяет уменьшить влияние перегрузки на другие узлы сети и избежать сбоев в функционировании оборудования. К преимуществам механизма можно отнести отсутствие необходимости передавать управляющие сообщения между узлами, находящимися на расстоянии более одного переприема, и собирать статистику о статусах перегрузок серверов, с которыми узел не связан напрямую.

Сквозное управление (англ., end-to-end) предусматривает наличие управляющего контура, объединяющего все узлы сети по маршруту следования сообщений от UAc в направлении UAs. Механизм должен собирать информацию по маршруту о состоянии всех узлов, включая серверы и UA-клиенты, и ограничить поступающую по маршруту нагрузку как можно ближе к источнику. Сбор информации должен проводиться регулярно, по всем потенциально возможным маршрутам следования сообщений, и результат должен отправляться источнику в управляющих сообщениях. UAc или сервер должны уменьшать число запросов только в направлении конкретного перегруженного сервера. Предположим, что сквозное управление реализовано, как показано на рис. 9, по контурам C2–C5–C3 и C2–C5–C4. Пусть только C3 находится в состоянии перегрузки, тогда C2 должен сократить число запросов только в направлении C3. Во многих случаях задача определения узла, которому будет доставлено в итоге сообщение, представляется сложной и трудно предсказуемой, так как зависит от применяемых политик маршрутизации вызовов, балансировки нагрузки, запрашиваемых услуг и пр. Если предыдущее сообщение, относящееся к процедуре инициации сессии, было направлено на перегруженный сервер, то не обязательно следующее сообщение будет направлено на этот же сервер.

Существенная проблема сквозного механизма контроля перегрузок заключается в необходимости мониторинга всех потенциально возможных маршрутов и определения, какие сообщения-запросы следует ограничить, а какие могут быть отправлены. Даже в случае наличия этой информации крайне сложно четко определить, по какому маршруту будет следовать запрос.

Основные идеи пяти явных механизмов управления, которые работают на основе уведомления, сформулированы нами на основании [4, 35] и заключаются в следующем.

- *Механизм снижения скорости передачи* (англ., Rate-based Overload Control) заключается в том, что сервер-получатель, исходя из текущего значения своей производительности, имеет возможность регулировать ограничения на скорости передачи соседних серверов-отправителей.
- *Механизм просеивания потока сообщений* (англ., Loss-based Overload Control) дает возможность серверу-получателю запросить отправителей снизить нагрузку на заданное число процентов, которое вычисляется получателем с учетом его текущей загрузки.
- *Механизм управления размером окна* (англ., Window-based Overload Control) определяется счетчиком числа отправленных сообщений, на которые ещё не получены положительные подтверждения. При отправке сообщения значение счетчика увеличивается, при получении подтверждения — уменьшается. Сервер приостанавливает отправку сообщений, если значение счетчика станет равным размеру окна перегрузки (англ., overload window).
- *Механизм снижения нагрузки по сигналу* (англ., Signal-based Overload Control) заключается в использовании сообщения 503 Service Unavailable в качестве ответа, который сигнализирует об обнаружении перегрузки сервера-получателя. Получив это сообщение, сервер-отправитель снижает нагрузку на сервер-получатель, чтобы избежать дальнейшего получения сообщений с кодом 503. Сервер-отправитель сохраняет передачу с текущим значением интенсивности нагрузки до тех пор, пока он продолжает получать сообщения с кодом 503. После того, как такие сообщения перестанут поступать, север отправитель может постепенно увеличивать нагрузку на сервер-получатель.
- *Механизм временного запрета передачи* (англ., On-/Off Overload Control) осуществляет запрет по команде от сервера-получателя. В качестве управляющего сообщения предлагается использовать сообщение 503 Service Unavailable с заголовком Retry-After.

В отличие от явных механизмов неявные механизмы управления перегрузкой гарантируют, что процесс передачи сообщений является самоограничивающимся (англ., self-limiting) и снижает скорость передачи сообщений на сервере-отправителе, когда появляются признаки перегрузки сервера-получателя. Таким

признаком может служить отсутствие ответов со стороны получателя в течение заданного интервала времени. Идея неявного управления заключается в том, что сервер-отправитель должен определить возможную перегрузку на сервере-получателе, даже если для этого не предусмотрены сообщения уведомления от сервера-получателя. Неявные механизмы гарантируют, что перегруженный сервер, не имеющий достаточных ресурсов на генерацию уведомления о перегрузке, не будет завален запросами. Из рассмотренных выше явных механизмов управления только механизм управления размером окна является самоограничивающимся, так как отправитель не сможет продолжить отправку сообщений в случае размера окна, равного нулю, до тех пор, пока не получит соответствующего уведомления. Остальные механизмы не обладают таким свойством и должны применяться совместно с неявным механизмом для того, чтобы ограничить передачу сообщений в случае, когда получатель не имеет достаточных ресурсов для отправки уведомления, сигнализирующего о перегрузке.

Во всех явных и неявных механизмах могут быть также реализованы приоритеты сообщений (англ., Message Prioritization), согласно которым сервер принимает решение о сбросе или перенаправлении сообщений. Принятие решения происходит с учетом локальных политик, архитектуры сети, а также услуг, которые предоставляет сервер. Документ RFC 4412 [36] определяет метод маркировки сообщений при помощи поля заголовка Resource-Priority header field. В зависимости от особенностей сети и предоставляемых услуг стандарт позволяет разделить сообщения на три группы:

- высокоприоритетные запросы, которые должны по возможности быть сохранены в случае перегрузки;
- низкоприоритетные запросы, которые должны быть сброшены в случае перегрузки;
- метки, которые не влияют на политику обслуживания сообщений по приоритетам в данной подсети.

Заметим, что эффективнее сбрасывать сообщения-запросы, а не сообщения-ответы, поскольку сброс ответа вызовет ретрансляции и увеличит нагрузку на сервер. Кроме того, механизмы управления перегрузкой не меняют процедуру ретрансляции сообщений, описанную в RFC 3261 [10].

Как говорилось выше, в известных авторам обзора источниках отсутствуют математические модели контроля перегрузок SIP-серверов. Поэтому в следующем разделе обзора на основании принципов гистерезисного управления нагрузкой ОКС7 мы впервые строим модель СМО с гистерезисным управлением перегрузками SIP-сервера.

4. Математические модели SIP-сервера с гистерезисным управлением нагрузкой

Известен ряд статей [14–18, 20–23, 25], где изложены подходы к построению моделей функционирования SIP-серверов с пороговым управлением в условиях перегрузок. Практически во всех известных на момент написания обзора источниках, включая стандарты комитета IETF, в явном виде отсутствует механизм гистерезисного управления нагрузкой, а все численные результаты получены либо с помощью измерений, либо с использованием имитационных моделей. Поэтому ниже мы строим две СМО с гистерезисным управлением нагрузкой, причем для первой из них приводим решение системы уравнений равновесия (СУР) в явном виде, как это сделано в [37].

Рассматривается СМО типа $M|M|1|\langle L,H\rangle|B$, изображенная на рис. 10, где B — объем буферного накопителя, L — порог нижнего уровня, H — порог верхнего уровня контроля перегрузок.

График интенсивности поступающего на СМО пуассоновского потока $\lambda(s,n)$ изображен на рис. 11, где $s \in \{0,1,2\}$ — статус перегрузки.

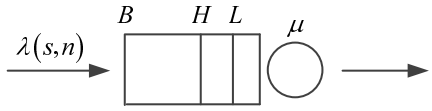

Рис. 10. **Однопотоковая СМО типа** $M|M|1|\langle L,H\rangle|B$

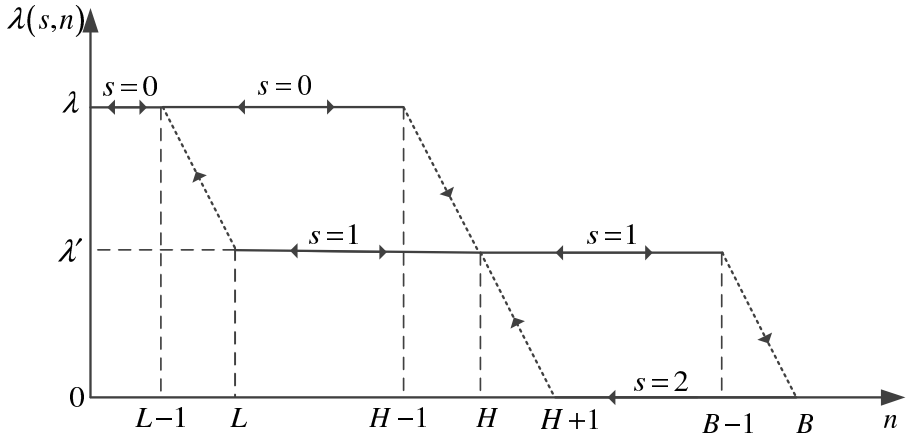

Рис. 11. **Гистерезисное управление сигнальной нагрузкой на SIP-сервере**

Предполагается, что в состоянии нормальной загрузки сервер может обрабатывать поток SIP-сообщений интенсивности λ, а при достижении общей длиной очереди n значения H порога верхнего уровня контроля перегрузок нагрузка уменьшается до величины $\lambda' = p\lambda$, где p - доля сообщений-ответов, показанных на рис.6, которые в зарубежных источниках принято называть сообщениями типа non-INVITE. Множество состояний системы на рис.10 с интенсивностью входящего потока $\lambda(s,n)$ на рис. 11 представимо в виде $\mathcal{Y} = \mathcal{Y}_0 \cup \mathcal{Y}_1 \cup \mathcal{Y}_2$, где $\mathcal{Y}_0$ — множество состояний нормальной нагрузки, $\mathcal{Y}_1$ — множество состояний перегрузки и $\mathcal{Y}_2$ множество состояний сброса нагрузки. Эти множества имеют вид $\mathcal{Y}_0 = \{(s,n) : s = 0, 0 \leqslant n \leqslant H-1\}, \mathcal{Y}_1 = \{(s,n) : s = 1, L \leqslant n \leqslant B-1\}, \mathcal{Y}_2 = \{(s,n) : s = 2, H+1 \leqslant n \leqslant B\}$ и, тогда

$$\lambda(s,n) = \begin{cases} \lambda, & (s,n) \in \mathcal{Y}_0, \\ p\lambda, & (s,n) \in \mathcal{Y}_1, \\ 0, & (s,n) \in \mathcal{Y}_2. \end{cases}$$

Пусть μ параметр экспоненциального распределения длительности обслуживания заявок и $p_{s,n}$ стационарное распределение вероятностей состояний системы $(s,n) \in \mathcal{Y}$, удовлетворяющее СУР:

$$\begin{cases} (\lambda+\mu u(k))\,p_{0,k} = \lambda u(k)\,p_{0,k-1} + \mu p_{0,k+1}, \quad k=0,\ldots,L-2, \quad k=L,\ldots,H-2,\\ (\lambda+\mu)\,p_{0,L-1} = \lambda p_{0,L-2} + \mu p_{0,L} + \mu p_{1,L},\\ (\lambda+\mu)\,p_{0,H-1} = \lambda p_{0,H-2},\\ (\lambda'+\mu)\,p_{1,L} = \mu p_{1,L+1},\\ (\lambda'+\mu)\,p_{1,k} = \lambda' p_{1,k-1} + \mu p_{1,k+1}, \quad k=L+1,\ldots,H-1, \quad k=H+1,\ldots,B-2,\\ (\lambda'+\mu)\,p_{1,B-1} = \lambda' p_{1,B-2},\\ (\lambda'+\mu)\,p_{1,H} = \lambda' p_{1,H-1} + \mu p_{1,H+1} + \lambda p_{0,H-1} + \mu p_{2,H+1},\\ p_{2,k} = p_{2,k+1}, \quad i=H+1,\ldots,B-1,\\ \mu p_{2,B} = \lambda' p_{1,B-1}, \end{cases}$$

где $u(k) = \begin{cases} 1, k>0,\\ 0, k\leqslant 0. \end{cases}$ Решение СУР представлено ниже в аналитическом виде:

$$p_{0,k} = \rho^k p_{0,0}, \quad k=1,\ldots,L-1; \quad p_{0,k} = \frac{\rho^k\left(1-\rho^{H-k}\right)}{1-\rho^{H-L+1}} p_{0,0}, \quad k=L,\ldots,H-1,$$

$$p_{1,k} = \frac{\rho^H(1-\rho)\left(1-\rho'^{k-L+1}\right)}{(1-\rho^{H-L+1})(1-\rho')} p_{0,0}, \quad k=L,\ldots,H,$$

$$p_{1,H+k} = \left(a_k - b_k \frac{(\rho'+1)\,a_{B-H-1} - \rho' a_{B-H-2}}{(\rho'+1)\,b_{B-H-1} - \rho' b_{B-H-2}}\right) p_{0,0}, \quad k=1,\ldots,B-H-1,$$

$$p_{2,k} = \rho'\left(a_{B-H-1} - b_{B-H-1} \frac{(\rho'+1)\,a_{B-H-1} - \rho' a_{B-H-2}}{(\rho'+1)\,b_{B-H-1} - \rho' b_{B-H-2}}\right) p_{0,0}, \quad k=H+1,\ldots,B,$$

где $\rho=\lambda/\mu$, $\rho'=\lambda'/\mu$, а $p_{0,0}$ находится из условия нормировки. Элементы числовых последовательностей $\{a_k\}$ и $\{b_k\}$ можно найти по следующим формулам:

$$a_0 = \frac{\rho^H(1-\rho)\left(1-\rho'^{H-L+1}\right)}{(1-\rho^{H-L+1})(1-\rho')},$$

$$a_1 = \left(\frac{1-\rho'^{H-L+2}}{1-\rho'^{H-L+1}} + \frac{\rho'(1-\rho')\left(1-\rho^{H-L}\right)}{\left(1-\rho'^{H-L+1}\right)(1-\rho)}\right) \frac{\rho^H(1-\rho)\left(1-\rho'^{H-L+1}\right)}{(1-\rho^{H-L+1})(1-\rho')} - \rho'\rho^{H-1},$$

$$a_k = (\rho'+1)\,a_{k-1} - \rho' a_{k-2}, \quad k\geqslant 2;$$

$$b_0 = 0, \quad b_1 = 1, \quad b_k = (\rho'+1)\,b_{k-1} - \rho' b_{k-2}, \quad k\geqslant 2.$$

Перейдем теперь к построению модели, которая учитывает поступление на SIP-сервер потоков сообщений двух типов — запросов INVITE и всех остальных сообщений-ответов. Разделение сообщений на два потока в модели сделано потому, что все механизмы (см. раздел 2 обзора) предусматривают в случае перегрузок в первую очередь сброс сообщений-запросов (INVITE), а не сообщений-ответов (non-INVITE). Таким образом, на СМО $M_2|M_2|1|\langle L,H\rangle|B$, изображенную на рис. 12, поступают два пуассоновских потока — ответов non-INVITE с интенсивностью $\lambda_1(s,i,n)$ и запросов INVITE с интенсивностью $\lambda_2(s,i,n)$.

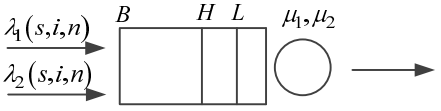

Рис. 12. **Двухпотоковая СМО типа** $M_2|M_2|1|\langle L,H\rangle|B$

Изменение статуса перегрузки s для рассматриваемой модели показано на рис. 13, где i число запросов INVITE и n общее число сообщений в очереди на обработку процессором сервера.

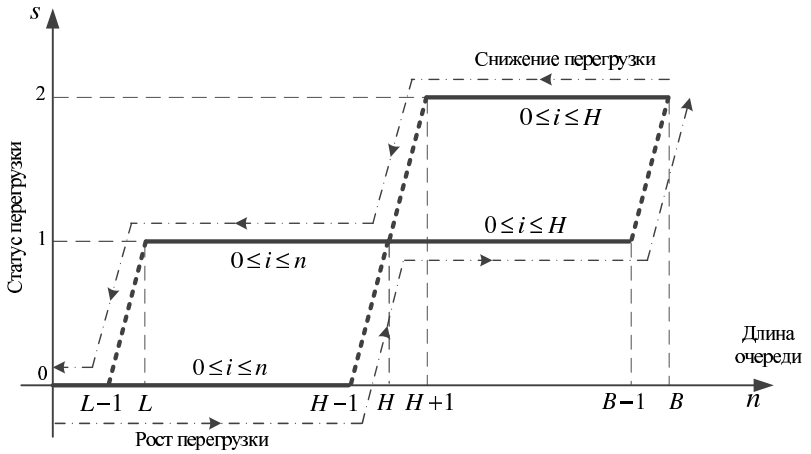

Рис. 13. **Изменение статуса перегрузки SIP-сервера**

Графики интенсивностей потоков сигнальной нагрузки $\lambda_1(s,i,n)$ и $\lambda_2(s,i,n)$ показаны на рис. 14 (а) и рис. 14 (б) соответственно, причем на рис. 14 (а) порог H является порогом снижения перегрузки, порог B — порогом обнаружения перегрузки, а на рис. 14 (б) порог H является порогом обнаружения перегрузки и порог L — порогом снижения перегрузки.

Пространство состояний $\mathcal{Y}$ модели СМО на рис. 12 представляется в виде

$$\mathcal{Y} = \mathcal{Y}_0 \cup \mathcal{Y}_1 \cup \mathcal{Y}_2,$$

где

$$\mathcal{Y}_0 = \{(s,i,n) : s = 0, 0 \leqslant i \leqslant n, 0 \leqslant n \leqslant H-1\};$$

$$\mathcal{Y}_1 = \{(s,i,n) : s = 1, 0 \leqslant i \leqslant n, L \leqslant n \leqslant H\} \cup \\ \cup \{(s,i,n) : s = 1, 0 \leqslant i \leqslant H, H+1 \leqslant n \leqslant B-1\};$$

$$\mathcal{Y}_2 = \{(s,i,n) : s = 2, 0 \leqslant i \leqslant H, H+1 \leqslant n \leqslant B\}.$$

Теперь из графиков на рис. 14 видно, что интенсивности потоков сообщений аналитически могут быть записаны следующим образом:

$$\lambda_1(s,i,n) = \begin{cases} \lambda_1, & (s,i,n) \in \mathcal{Y}_0 \cup \mathcal{Y}_1, \\ 0, & (s,i,n) \in \mathcal{Y}_2, \end{cases} \qquad \lambda_2(s,i,n) = \begin{cases} \lambda_2, & (s,i,n) \in \mathcal{Y}_0, \\ 0, & (s,i,n) \in \mathcal{Y}_1 \cup \mathcal{Y}_2. \end{cases}$$

Ясно, что аналогично однопотоковой СМО могут быть получены СУР и алгоритм для расчета стационарных вероятностей двухпотоковой СМО на рис. 12, что мы не делаем для краткости изложения.

В заключение к данному разделу отметим, что из статей [14–18, 20–23, 25, 26] известны и другие алгоритмы контроля перегрузок SIP-серверов, а построение соответствующих моделей СМО с гистерезисным управлением может быть сделано по той же схеме, что и для двух вышерассмотренных СМО.

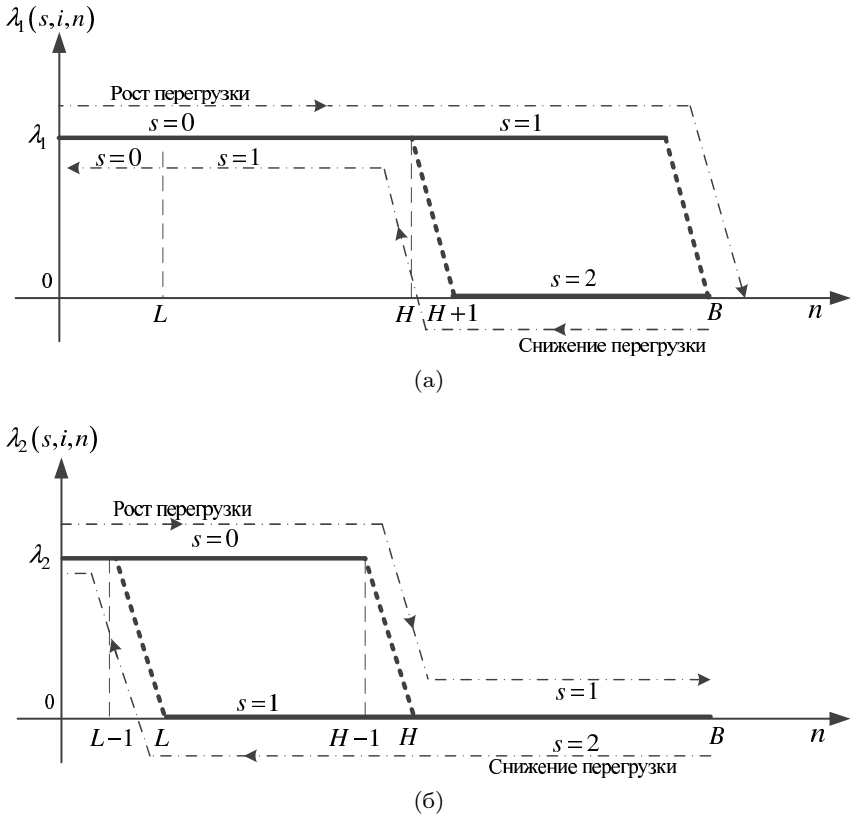

Рис. 14. **Гистерезисное управление сигнальной нагрузкой: а) запросы INVITE; б) ответы non-INVITE**

5. Заключение

Комитет IETF ведет интенсивные работы по стандартизации в области контроля перегрузок в сетях сигнализации следующих поколений. Принятые в настоящее время стандарты и проекты стандартов, а также целый ряд научных публикаций, до сих пор не дают ответов на проблемы в механизмах контроля перегрузок протокола SIP.

Включенные в библиографию настоящего обзора работы отражают основные тенденции развития как базовых механизмов контроля перегрузок протокола SIP, так и принципов построения моделей гистерезисного управления нагрузкой. Вследствие небольшого объема данной статьи ряд механизмов и моделей приведен весьма схематично, а многие из них по той же причине даже не анализируются. Одним из достоинств обзора мы считаем взаимосвязанное описание моделей гистерезисного управления в сетях ОКС7 и в сетях SIP-серверов. Заинтересованному читателю мы рекомендуем ознакомиться более детально с источниками, указанными в библиографии обзора, которые в большинстве случаев содержат достаточно подробный анализ особенностей управления нагрузкой в сетях сигнализации.

Среди актуальных направлений дальнейших исследований прежде всего следует указать развитие теории СМО с гистерезисным управлением нагрузкой, конечной емкостью буферного накопителя и групповым поступлением заявок. При

этом следует стремиться к созданию инженерных методов расчета наиболее актуальных вероятностных параметров качества функционирования системы гистерезисного управления, причем наибольший интерес представляют параметры, определенные нами в данном обзоре.

Литература

1. ITU-T Recommendation Q.704: Signalling System No.7 – Message Transfer Part, Signalling Network Functions and Messages. — 1996.
2. ITU-T Recommendation Q.752: Monitoring and Measurements for Signalling System No. 7 Networks. — 1997.
3. *Rosenberg J.* Requirements for Management of Overload in the Session Initiation Protocol // RFC5390. — 2008.
4. Design Considerations for Session Initiation Protocol (SIP) Overload Control / V. Hilt, E. Noel, C. Shen, A. Abdelal // draft-ietf-soc-overload-design-06. — 2011.
5. *Красносельский М. А.* Системы с гистерезисом. — М.: Наука, 1984. — 272 с. [*Krasnoseljskiyj M. A.* Sistemih s gisterezisom. — M.: Nauka, 1984. — 272 s.]
6. *Жарков М. А., Закошанский А. Э., Самуйлов К. Е.* Об одном методе анализа процедуры гистерезисного управления нагрузкой в системе сигнализации №7 МККТТ // Труды 14 Всесоюзной школы-семинара по вычислительным сетям. — ВИНИТИ, 1989. — Т. 3. — С. 58–67. [*Zharkov M. A., Zakoshanskiyj A. Eh., Samuyjlov K. E.* Ob odnom metode analiza procedurih gisterezisnogo upravleniya nagruzkoyj v sisteme signalizacii No7 MKKTT // Trudih 14 Vsesoyuznoyj shkolih-seminara po vihchisliteljnihm setyam. — VINITI, 1989. — T. 3. — S. 58–67.]
7. *Самуйлов К. Е.* Методы анализа и расчета сетей ОКС 7. — М.: РУДН, 2002. — 291 с. [*Samuyjlov K. E.* Metodih analiza i rascheta seteyj OKS 7. — M.: RUDN, 2002. — 291 s.]
8. *Юнаков П. А., Иванов М. Б.* Метод оптимизации структуры местных сетей ОКС при применении цифровых систем коммутации // Электросвязь. — 1988. — Т. 10. — С. 32–35. [*Yunakov P. A., Ivanov M. B.* Metod optimizacii strukturih mestnihkh seteyj OKS pri primenenii cifrovihkh sistem kommutacii // Ehlektrosvyazj. — 1988. — T. 10. — S. 32–35.]
9. Системы сигнализации сетей коммутации каналов и коммутации пакетов / А. И. Летников, А. П. Пшеничников, Ю. В. Гайдамака, А. В. Чукарин. — М.: МТУСИ, 2008. — 195 с. [Sistemih signalizacii seteyj kommutacii kanalov i kommutacii paketov / A. I. Letnikov, A. P. Pshenichnikov, Yu. V. Gayjdamaka, A. V. Chukarin. — M.: MTUSI, 2008. — 195 s.]
10. SIP: Session Initiation Protocol / J. Rosenberg, H. Schulzrinne, G. Camarillo et al. // RFC3261. — 2002.
11. Session Initiation Protocol (SIP) Basic Call Flow Examples / A. Johnston, S. Donovan, R. Sparks et al. // RFC3665. — 2003.
12. *Гольдштейн Б. С., Соколов Н. А., Яновский Г. Г.* Сети связи. — СПб.: БХВ-Петербург, 2010. — 400 с. [*Goljdshteyjn B. S., Sokolov N. A., Yanovskiyj G. G.* Seti svyazi. — SPb.: BKhV-Peterburg, 2010. — 400 s.]
13. Fast and Robust Signaling Overload Control / S. Kasera, J. Pinheiro, C. Loader et al. // Ninth International Conference on Network Protocols. — 2001. — Pp. 323–331.
14. *Ohta M.* Overload Protection in a SIP Signaling Network // International Conference on Internet Surveillance and Protection. — 2006. — Pp. 205–210.
15. *Ohta M.* Overload Control in a SIP Signaling Network // International Journal of Electrical and Electronics Engineering. — 2009. — Pp. 87–92.
16. *Hilt V., Widjaja I.* Controlling Overload in Networks of SIP Servers // IEEE International Conference on Network Protocols. — 2008. — Pp. 83–93.

17. *Shen C., Schulzrinne H., Nahum E.* Session Initiation Protocol (SIP) Server Overload Control: Design and Evaluation // Lecture Notes in Computer Science. — Springer, 2008. — Vol. 5310. — Pp. 149–173.
18. *Montagna S., Pignolo M.* Performance Evaluation of Load Control Techniques in SIP Signaling Servers // Proceedings of Third International Conference on Systems (ICONS). — 2008. — Pp. 51–56.
19. *Yang J., Huang F., Gou S.* An Optimized Algorithm for Overload Control of SIP Signaling Network // Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM). — 2009.
20. Queueing Strategies for Local Overload Control in SIP Server / R. G. Garroppo, S. Giordano, S. Spagna, S. Niccolini // IEEE Global Telecommunications Conference. — 2009. — Pp. 1–6.
21. *Montagna S., Pignolo M.* Load Control Techniques in SIP Signaling Servers using Multiple Thresholds // 13th International Telecommunications Network Strategy and Planning Symposium, NETWORKS. — 2008. — Pp. 1–17.
22. A Prediction-Based Overload Control Algorithm for SIP Servers / R. G. Garroppo, S. Giordano, S. Niccolini, S. Spagna // Network and Service Management, IEEE Transactions on. — 2011. — Vol. 8, No 1. — Pp. 39–51.
23. Controlling Overload in SIP Proxies: An Adaptive Window Based Approach Using No Explicit Feedback / M. Homayouni, H. Nemati, V. Azhari, A. Akbari // GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference. — 2010. — Pp. 1–5.
24. *Abdelal A., Matragi W.* Signal-Based Overload Control for SIP Servers // Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE. — 2010. — Pp. 1–7.
25. *Montagna S., Pignolo M.* Comparison Between Two Approaches to Overload Control in a Real Server: "Local" or "Hybrid" Solutions? // MELECON 2010 — 2010 15th IEEE Mediterranean Electrotechnical Conference. — 2010. — Pp. 845–849.
26. *Hong Y., Huang C., Yan J.* Mitigating SIP Overload Using a Control-Theoretic Approach // GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference. — 2010. — Pp. 1–5.
27. *Gebhart R. F.* A Queuing Process with Bilevel Hysteretic Service-Rate Control // Naval Research Logistics Quarterly. — ВИНИТИ, 1967. — Vol. 14. — Pp. 55–68.
28. *Brown P., Chemouil P., Delosme B.* A Congestion Control Policy for Signalling Networks // Proceedings of 7th. IeCC. — 1984. — Pp. 717–724.
29. *Filipiak J.* Modelling and Control of Dynamic Flows in Communication Networks. — New York: Springer-Verlag, 1988. — 202 p.
30. *Roughan M., Pearce C.* A Martingale Analysis of Hysteretic Overload Control // Advances in Performance Analysis: A Journal of Teletraffic Theory and Performance Analysis of Communication Systems and Networks. — 2000. — Vol. 3. — Pp. 1–30.
31. *Takagi H.* Analysis of a Finite-Capacity $M/G/1$ Queue with a Resume Level // Performance Evaluation. — 1985. — Vol. 5. — Pp. 197–203.
32. *Yum T. P., Yen H. M.* Design Algorithm for a Hysteresis Buffer Congestion Control Strategy // In Proceedings of the IEEE International Conference on Communications. — 1983. — Pp. 499–503.
33. *Jennings C., Mahy R., Audet F.* Managing Client-Initiated Connections in the Session Initiation Protocol (SIP) // RFC5626. — 2009.
34. Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping / G. Camarillo, A. Roach, J. Peterson, L. Ong // RFC3398. — 2002.
35. *Gurbani V., Hilt V., Schulzrinne H.* Session Initiation Protocol (SIP) Overload Control // draft-gurbani-soc-overload-control-02. — 2010.
36. *Schulzrinne H., Polk J.* Communications Resource Priority for the Session Initiation Protocol (SIP) // RFC4412. — 2006.
37. *Abaev P. O.* Algorithm for Computing Steady-State Probabilities of the Queuing

System with Hysteretic Congestion Control and Working Vacations // Bulletin of Peoples' Friendship University of Russia. — 2011. — No 3. — Pp. 58–62.

UDC 621.39

Hysteretic Overload Control in a SIP Signaling Network

P. O. Abaev, Y. V. Gaidamaka, K. E. Samouylov

Telecommunication Systems Department
Peoples' Friendship University of Russia
Miklukho-Maklaya str., 6, Moscow, Russia, 117198

This review deals with the research of load control mechanisms in signaling networks that use three types of thresholds for congestion control. The main objective of this paper is to analyze the congestion control mechanisms and mathematical models for SIP-servers. The study is based on hysteresial techniques of flow control, which originally was developed for Signaling System 7. We propose general methods for describing hysteresis signaling flow control techniques. We study the current situation and problems of SIP built-in overload control mechanism, proposed by IETF. Our approaches to mathematical models construction in the form of queuing systems with hysteresis control are presented.

Key words and phrases: signaling network, SS7, SIP server, hysteretic overload control, overload control mechanism, threshold control.