

Асинхронные каналы для протокола BEEP

Asynchronous Channels for the Blocks Extensible Exchange Protocol (BEEP)

Статус документа

Этот документ определяет экспериментальный протокол для сообщества Internet. Документ не задаёт каких-либо стандартов Internet и служит приглашением к дискуссии в целях совершенствования протокола. Распространение документа не ограничивается.

Авторские права

Авторские права принадлежат IETF Trust и лицам, указанным в качестве авторов. Все права защищены ((c) 2009).

К этому документу применимы права и ограничения, перечисленные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно.

Аннотация

Протокол BEEP¹ обеспечивает протокольную модель для разработки прикладных протоколов. В этом документе определяется функция (свойство) BEEP, обеспечивающая асинхронный обмен для отдельных каналов.

Оглавление

1. Введение.....	1
2. Асинхронные каналы BEEP.....	2
2.1. Асинхронная функция.....	2
2.2. Организация асинхронного канала.....	2
2.3. Поведение асинхронного канала.....	3
2.4. Обработка ошибок.....	3
3. Альтернативы.....	3
3.1. Повышение пропускной способности.....	3
3.2. Асинхронность в прикладном протоколе.....	3
4. Вопросы безопасности.....	4
5. Взаимодействии с IANA.....	4
6. Литература.....	4
6.1. Нормативные документы.....	4
6.2. Дополнительная литература.....	4

1. Введение

Протокол BEEP обеспечивает модель, которая управляет многими аспектами разработки прикладных протоколов - кадрирование, кодирование, конфиденциальность, проверка подлинности, асинхронное взаимодействие. Однако асинхронность BEEP ограничена асинхронностью между каналами, а отклики на сообщения, переданные по любому каналу, строго упорядочены.

Последовательная обработка желательна для многих приложений. Однако она не совсем подходит для приложений, которые больше полагаются на асинхронность. В частности, если создание отклика занимает много времени, канал по сути блокируется на время обработки и отправки отклика. Конвейер лишь гарантирует, что задержка в сети не добавится к этому времени, однако последующие запросы не могут обрабатываться, пока не будет отклика на первый запрос.

Асинхронным приложениям нужен протокол, способный поддерживать большое число остающихся одновременно запросов. Представление каналов как потоков (нитей) не позволяет решить задачу расширяемости. Аналогия канала и потока на обеспечивает расширяемости до большого числа потоков, используемых в современных системах. Современные приложения обычно имеют множество одновременно обрабатываемых потоков. Таким образом, нужен более эффективный способ мультиплексирования большого числа одновременных запросов.

В этом документе описывается функция BEEP - расширение BEEP, позволяющее создать асинхронный канал. Асинхронным каналом является канал, в котором порядок откликов не привязан к порядку запросов, переданных партнером-клиентом. Асинхронный канал идентичен другим каналам, используемым не изменённое кадрирование, за исключением того, что запросы могут обрабатываться параллельно, а отклики могут передаваться в любом порядке.

Асинхронные каналы позволяют эффективно использовать один канал для множества одновременных запросов. Это не влияет на запросы, вызванные своевременными откликами на другие запросы. Запрашивающий партнёр может обрабатывать отклики на переданные им запросы по мере их поступления, а обслуживающий партнёр может использовать преимущества параллельной обработки без искусственных ограничений, связанных с порядком откликов.

¹Blocks Extensible Exchange Protocol - расширяемый протокол обмена блоками.

Асинхронные каналы обеспечивают высокую пропускную способность, когда обслуживаемому узлу требуется какое-то время для обработки запросов. Это особенно актуально, когда обслуживающий узел должен выполнять продолжительные расчёты или делать сетевые запросы в процессе обслуживания.

Согласование функции BEEP используется для того, чтобы гарантировать желание обеих сторон создать асинхронные каналы. Описаны способы организации асинхронного канала.

Этот документ публикуется как экспериментальный RFC, чтобы выяснить, будет ли расширение развёрнуто в различных вариантах и приложениях.

2. Асинхронные каналы BEEP

Этот документ определяет функцию BEEP, позволяющую применять асинхронные каналы. Асинхронным каналом является канал BEEP, к которому не применяются ограничения параграфа 2.6.1 из [RFC3080] в части упорядочения откликов. Запросы могут обрабатываться, а отклики на них передаваться обслуживающим партнёром в любом порядке.

Асинхронные каналы используют элемент msgno в заголовке кадра BEEP для сопоставления запросов и откликов. Обычные каналы BEEP не используют msgno для сопоставления откликов с запросами, несмотря на обеспечиваемые этим возможности. В обычном канале BEEP поле msgno служит лишь для контроля протокольных ошибок.

Асинхронные каналы не подходят для приложений, где созданное запросом состояние зависит от последующих запросов и порядок сообщений имеет значение.

2.1. Асинхронная функция

Атрибут feature в приветствии BEEP содержит список разделённых пробелами функций, поддерживаемых каждым партнёром. Если оба списка содержат некую функцию, она может применяться любым из партнёров.

Этот документ регистрирует функцию async. Если какой-либо из партнёров не указал эту функцию в приветственном сообщении, ни одна из сторон не может создать асинхронный канал.

На рисунке 1 показан пример обмена между партнёрами, выразившими желание использовать эту функцию.

```
I: <ожидание входящего соединения>
I: <организация соединения>
I: RPY 0 0 . 0 133
L: Content-Type: application/beep+xml
I:
L: <greeting features="async x-foo">
L:   <profile uri="http://example.com/beep/APP" />
L: </greeting>
I: END
I: RPY 0 0 . 0 69
I: Content-Type: application/beep+xml
I:
I: <greeting features="async" />
I: END
```

Рисунок 1. Приветствие BEEP с асинхронной функцией.

Регистрационный шаблон для функций BEEP приведён в разделе 5.

2.2. Организация асинхронного канала

Для создания асинхронного канала параметр async=true включается в запрос start. Если параметр опущен или установлено значение false, канал не будет асинхронным.

На рисунке 2 показано как можно использовать атрибут async для старта асинхронного канала.

```
C: MSG 0 1 . 52 130
C: Content-Type: application/beep+xml
C:
C: <start number="1" async="true">
C:   <profile uri="http://example.org/protocol"/>
C: </start>
C: END
S: RPY 0 1 . 221 102
S: Content-Type: application/beep+xml
S:
S: <profile uri="http://example.org/protocol"/>
S: END
```

Рисунок 2. Старт асинхронного канала.

Если обслуживающий партнёр не способен по какой-либо причине создать асинхронный канал, создание канала отвергается. Это может происходить, когда выбранный профиль не подходит для асинхронного канала. Отклик может включать код 553 (недействительный параметр) и подходящее сообщение, как показано на рисунке 3.

```
C: MSG 0 1 . 52 128
C: Content-Type: application/beep+xml
C:
C: <start number="1" async="true">
C:   <profile uri="http://example.org/serial"/>
C: </start>
C: END
S: ERR 0 1 . 221 152
S: Content-Type: application/beep+xml
S:
S: <error code="553">Profile &lt;http://example.org/serial&gt;
S: cannot be used for asynchronous channels.</error>
S: END
```

Рисунок 3. Ошибка при старте асинхронного канала.

2.3. Поведение асинхронного канала

Асинхронный канал отличается от обычных каналов BEEP лишь в том, что к нему не применяются ограничения параграфа 2.6.1 из [RFC3080] в части порядка обработки и откликов. Обслуживающий партнёр может обрабатывать запросы и отвечать на них в выбранном им порядке.

В асинхронном канале элемент msgno заголовка кадра служит для сопоставления запросов и откликов. Узел BEEP, получающий отклики в порядке, отличающемся от порядка вызвавших их запросов, не должен считать это ошибкой протокола.

Сообщения MSG, переданные в асинхронный канал, могут параллельно обрабатываться обслуживающим партнёром. Отклики (сообщения RPY, ANS, NUL, ERR) могут передаваться в любом порядке. Разные сообщения ANS, передаваемые в обменах «один со многими», могут чередоваться с другими сообщениями MSG.

Асинхронный канал должен соблюдать правила [RFC3080] в части сегментированных сообщений. Каждое сообщение должно быть завершено до того, как можно будет передать через тот же канал другое сообщение.

Примечание. Исключение из этого правила сделано в [RFC3080] для чередующихся сегментов ANS, передаваемых в ответ на одно сообщение MSG. Партнёрам BEEP рекомендуется не генерировать чередующихся сегментов ANS.

Канал управления BEEP (канал 0) никогда не бывает асинхронным.

2.4. Обработка ошибок

BEEP не предоставляет никакого механизма для управления узлом, не отвечающим на запрос. Синхронные каналы невозможно использовать и даже закрыть, если партнёр не предоставляет отклика на запрос. Единственным доступным способом остаётся разрыв транспортного соединения. Хотя асинхронный канал невозможно закрыть, он всё ещё может использоваться для дальнейших запросов. Однако остающиеся запросы продолжают потреблять ресурсы состояний. Партнёры-клиенты могут избавиться от таких состояний по истечении настроенного интервала, но должны быть готовы отказаться в таких случаях от нераспознанных откликов.

3. Альтернативы

Представленная здесь опция обеспечивает асинхронное взаимодействие. Асинхронные каналы могут подходить не в каждом случае, особенно если важен порядок запросов. В зависимости от требований прикладных протоколов рассмотренные в этом разделе варианты могут быть более применимыми.

3.1. Повышение пропускной способности

В некоторых случаях асинхронные каналы могут применяться для снятия ограничений для пропускной способности при обработке сообщений. Конвейерная обработка может значительно повысить производительность в тех случаях, когда сетевые задержки являются ограничивающим фактором. Распределение запросов по нескольким каналам повышает общую пропускную способность, если она является единственной проблемой.

Примечание. Следует остерегаться ложной оптимизации, основанной на конвейерной обработке запросов. Если более поздние запросы в последовательности конвейерных запросов зависят от состояния, созданного предшествующими запросами, ошибки в ранних запросах могут сделать недействительными более поздние запросы.

Окно управления потоком данных, применяемое в отображении TCP [RFC3081] может вносить ограничения на пропускную способность отдельных каналов. Выбор размера окна TCP также ограничивает пропускную способность (см. [RFC1323]). Чтобы избежать ограничений, принимающие партнёры могут увеличивать окно, а передающие партнёры могут создавать дополнительные каналы с тем же профилем. Корректное управление потоком данных также применимо к асинхронным каналам.

3.2. Асинхронность в прикладном протоколе

При внесении изменений в прикладные протоколы последовательные каналы можно использовать для асинхронного обмена. Асинхронность может быть обеспечена на уровне протокола выше BEEP путём разделения запросов и откликов. Это требует добавления фирменных заголовков MIME или изменения прикладного протокола.

Обслуживающий партнёр обеспечивает незамедлительный отклик RPY (или NUL) на запросы. Это освобождает канал для последующих запросов. Фактический отклик передаётся в отдельном сообщении MSG с использованием специального идентификатора, включённого в исходный запрос для сопоставления откликов с запросами. Это вторичное сообщение MSG может быть передано по тому же каналу (поскольку он полнудуплексный) или по специально созданному для этой цели каналу.

Этот метод не является предпочтительным, поскольку требует от прикладного протокола решать задачу сопоставления откликов с запросами. BEEP стремится обеспечить основу для создания прикладного протокола и

отсутствие сопоставления откликов с запросами будет ограничивать его полезность. Использование заголовка MIME возможно, но msgho обеспечивает более элегантное решение.

4. Вопросы безопасности

Разрешение асинхронного обмена сообщениями для канала потенциально требует поддержки дополнительной информации о состоянии. Не отвечающий на сообщения обслуживающий партнёр может вызвать скопление состояний на стороне партнера-клиента. Если такая информация о состоянии не была ограничена, это может быть использовано для DoS¹-атаки. Эта проблема уже имеется в BEEP, но её значение возрастает из-за характера обработки на обслуживающем узле запросов, полученных по асинхронному каналу. Уровень отказов в обслуживании ограничен тем, что воспринимает обслуживающий партнёр - число остающихся не выполненными до конца запросов можно ограничить для защиты от накопления избыточных данных о состоянии.

Клиент поддерживает состояние для каждого отправленного запроса и ему следует устанавливать настраиваемое ограничение числа запросов, которые остаются незавершёнными. Это ограничение может быть применено в канале, на соединении или в области действия приложения. По достижении предела клиент может прекратить или заблокировать генерацию запросов.

Узлы, обслуживающие запросы на асинхронных каналах, также накапливают информацию о состоянии при восприятии запроса для обработки. Обслуживающие узлы тоже могут ограничивать число одновременно обрабатываемых запросов. По достижении заданного предела принимающий партнёр может остановить считывание новых запросов или начать отклонять их, возвращая сообщения об ошибках. Дополнительно может применяться управление потоком данных [RFC3081] - кадры SEQ могут быть подавлены, что позволит закрыть окно управления потоком данных и предотвратить получение новых запросов.

5. Взаимодействие с IANA

Этот раздел регистрирует функцию BEEP async в реестре параметров BEEP в соответствии с шаблоном, представленным в параграфе 5.2 [RFC3080].

Идентификация функции - async

Семантика функции - эта функция позволяет создавать асинхронные каналы (см. раздел 2 в RFC 5573).

Контактные данные - Martin Thomson <martin.thomson@andrew.com>

6. Литература

6.1. Нормативные документы

[RFC3080] Rose, M., "The Blocks Extensible Exchange Protocol Core", [RFC 3080](#), March 2001.

6.2. Дополнительная литература

[RFC3081] Rose, M., "Mapping the BEEP Core onto TCP", [RFC 3081](#), March 2001.

[RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", [RFC 1323](#), May 1992.

Адрес автора

Martin Thomson

Andrew

PO Box U40

Wollongong University Campus, NSW 2500

AU

Phone: +61 2 4221 2915

E-Mail: martin.thomson@andrew.com

URI: <http://www.andrew.com/>

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

¹Denial of service - отказ в обслуживании.