

(19)



**Евразийское
патентное
ведомство**

(11) **033980**(13) **B1**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2019.12.16

(21) Номер заявки
201791342

(22) Дата подачи заявки
2015.10.14

(51) Int. Cl. **H04L 12/58** (2006.01)
G06Q 20/02 (2012.01)
G06Q 20/38 (2012.01)

(54) ИНТЕРФЕЙС СПОСОБ И КОМПЬЮТЕРНЫЙ ПРОГРАММНЫЙ ПРОДУКТ ДЛЯ УПРАВЛЕНИЯ ПЕРЕСЫЛКОЙ ЭЛЕКТРОННЫХ СООБЩЕНИЙ

(31) **1422637.7**

(32) **2014.12.18**

(33) **GB**

(43) **2017.11.30**

(86) **PCT/GB2015/053042**

(87) **WO 2016/097675 2016.06.23**

(71)(73) Заявитель и патентовладелец:
ИПКО 2012 ЛИМИТЕД (GB)

(56) **US-A1-2008219207**
US-A-4100533
EP-A2-1025680
US-A1-2003023877

(72) Изобретатель:
Гарлик Стивен Джордж, Мастерс Нил
Энтони (GB)

(74) Представитель:
Медведев В.Н. (RU)

(57) Изобретение обеспечивает интерфейс для управления пересылкой электронных транзакционных сообщений между первым финансовым институтом и множеством коммутаторов, распределенных по множеству коммутационных пунктов, при этом первый финансовый институт и множество коммутаторов соединены через сеть передачи данных, причем указанный интерфейс содержит коммуникационные схемы, обрабатывающие схемы и память, хранящую операционное состояние каждого коммутационного пункта, при этом коммуникационные схемы выполнены с возможностью передачи тестового сообщения на один из коммутационных пунктов через сеть данных, если из указанного коммутационного пункта не было принято ни одного транзакционного сообщения в течение заданного времени, и в ответ на указанное тестовое сообщение коммуникационные схемы выполнены с возможностью приема эхосигнала тестового сообщения из указанного коммутационного пункта, при этом обрабатывающие схемы выполнены с возможностью, если указанный эхосигнал принят в течение некоторого определенного времени, установить операционное состояние указанного коммутационного пункта как рабочее, а если указанный эхосигнал не принят в течение указанного определенного времени, установить операционное состояние указанного коммутационного пункта как нерабочее; в результате чего в случае, когда операционное состояние указанного коммутационного пункта установлено как нерабочее, обрабатывающие схемы выполнены с возможностью не пересылать ни одного электронного сообщения на указанный коммутационный пункт.

B1**033980****033980****B1**

Область техники, к которой относится изобретение

Настоящее изобретение относится к интерфейсу, способу и компьютерному программному продукту для управления пересылкой электронных сообщений.

Уровень техники

Описание уровня техники обеспечено здесь с целью дать общее представление о контексте изобретения. Произведение указанных авторов изобретения в объеме, описанном в разделе "Уровень техники", а также аспекты описания, которые нельзя по-иному квалифицировать как известный уровень техники во время подачи заявки, ни в явном виде, ни косвенно, нельзя признать в качестве известного уровня техники, противопоставленного настоящему изобретению.

Современные электронные банковские системы позволяют осуществлять электронное перечисление денежных средств между банковскими счетами разных банков, используя сообщения с запросом электронной транзакции (транзакционные сообщения). Указанный способ перечисления денежных средств является надежным (позволяет использовать различные известные протоколы защиты, которые позволяют защитить передачу данных по сети) и удобным (поскольку позволяет держателям банковских счетов делать запросы на перечисление денежных средств в любое время, 24 ч в сутки, 365 дней в году).

Однако, хотя указанные сообщения с запросом электронной транзакции могут выполняться в любое время, в действительности денежные средства перечисляются из банка в банк на периодической основе. Этот процесс известен как модель "расчетного цикла". Во время каждого расчетного цикла транзакционные сообщения, выдаваемые банками, принимаются и записываются. Затем, в конце расчетного цикла происходит действительное перечисление денежных средств из банка в банк (известное как "расчеты") на основе записанных транзакционных сообщений. Период времени между расчетными циклами, как правило, составляет 8, 12 или 24 часа, хотя возможно использование любого другого подходящего периода времени.

Поскольку итоговые расчеты между банками основаны на записанных транзакционных сообщениях, созданных в течение расчетного цикла, особенно важно, чтобы хранение и управление записанными транзакционными сообщениями эффективно выполнялось даже в том случае, например, когда произошел технологический сбой или природные катаклизмы. Это гарантирует, что в конце каждого расчетного цикла каждая завершенная транзакция будет учтена во время расчетов, причем учтена только один раз. Это предотвращает любого рода переоплату или недоплату реальных денежных средств в межбанковских расчетах. Таким образом, настоящее изобретение имеет своей целью решить по меньшей мере проблему обеспечения надежной и легко перестраиваемой системы для эффективного хранения и управления сообщениями, касающимися электронных транзакций.

Сущность изобретения

Настоящее изобретение обеспечивает интерфейс для управления пересылкой электронных транзакционных сообщений между первым финансовым институтом и множеством коммутаторов, распределенных по множеству коммутационных пунктов, при этом первый финансовый институт и множество коммутаторов соединены через сеть передачи данных, причем указанный интерфейс содержит коммуникационные схемы, обрабатывающие схемы и память, хранящую операционное состояние каждого коммутационного пункта, при этом коммуникационные схемы выполнены с возможностью передачи тестового сообщения на один из коммутационных пунктов через сеть данных, если из указанного коммутационного пункта не было принято ни одного транзакционного сообщения в течение заданного времени, и в ответ на указанное тестовое сообщение коммуникационные схемы выполнены с возможностью приема экзосигнала тестового сообщения из указанного коммутационного пункта, при этом обрабатывающие схемы выполнены с возможностью, если указанный экзосигнал принят в течение некоторого определенного времени, установить операционное состояние указанного коммутационного пункта как рабочее, а если указанный экзосигнал не принят в течение указанного определенного времени, установить операционное состояние указанного коммутационного пункта как нерабочее; в результате чего в случае, когда операционное состояние указанного коммутационного пункта установлено как нерабочее, обрабатывающие схемы выполнены с возможностью не пересылать ни одного электронного сообщения на указанный коммутационный пункт.

Преимуществом является то, что такая конфигурация означает, что электронные сообщения не посылают в коммутационные пункты, определенные как недоступные и не способные принимать сообщения. К тому же это уменьшает риск искажения или потери сообщений, и это также означает, что возможности обработки и пропускная способность сети не используются впустую на сообщения, которые не могут быть доставлены. Следовательно, повышается производительность обработки и эффективность сети.

В варианте осуществления коммуникационные схемы выполнены с возможностью отправки дополнительного тестового сообщения в нерабочий коммутационный пункт после некоторого точно определенного периода.

В варианте осуществления указанная память выполнена с возможностью хранения операционного состояния второго финансового института, подсоединенного к указанному множеству коммутаторов через сеть передачи данных; причем коммуникационные схемы выполнены с возможностью приема ин-

формации от одного или более из множества коммутаторов, указывающей обновленное операционное состояние указанного второго финансового института; и обрабатывающие схемы выполнены с возможностью обновления операционного состояния второго финансового института, хранящегося в указанной памяти, на основе информации, полученной от одного или более коммутаторов, при этом в случае, когда операционное состояние второго финансового института указывает, что второй финансовый институт находится в нерабочем состоянии, обрабатывающие схемы выполнены с возможностью предотвращения передачи на указанные коммутаторы электронных транзакционных сообщений, поручающих выполнение транзакции от первого финансового института во второй финансовый институт.

В варианте осуществления информация, указывающая обновленное операционное состояние второго финансового института, создается одним или более из множества коммутаторов на основе отсутствия ответа на тестовое сообщение, переданное во второй финансовый институт каждым из коммутаторов в течение заданного периода времени, или указания вторым финансовым институтом одному или более из множества коммутаторов о том, что второй финансовый институт следует отсоединить от сети передачи данных.

Настоящее изобретение обеспечивает способ управления пересылкой электронных транзакционных сообщений между первым финансовым институтом и множеством коммутаторов, распределенных по множеству коммутационных пунктов, при этом первый финансовый институт и множество коммутаторов соединены через сеть передачи данных, причем способ содержит этапы, на которых: передают тестовое сообщение на один из коммутационных пунктов через сеть данных, если из указанного коммутационного пункта не было принято ни одного транзакционного сообщения в течение заданного времени; и в ответ на указанное тестовое сообщение принимают эхосигнал тестового сообщения из указанного коммутационного пункта; при этом если указанный эхосигнал принят в течение некоторого определенного времени операционное состояние указанного коммутационного пункта устанавливают как рабочее, а если указанный эхосигнал не принят в течение указанного определенного времени операционное состояние указанного коммутационного пункта устанавливают как нерабочее; в результате чего в случае, когда операционное состояние указанного коммутационного пункта установлено как нерабочее, способ содержит этап, на котором ни одно электронное сообщение не пересылают на указанный коммутационный пункт.

В варианте осуществления способ содержит этап, на котором посылают дополнительное тестовое сообщение в нерабочий коммутационный пункт после некоторого точно определенного периода.

В варианте осуществления способ содержит этапы, на которых сохраняют операционное состояние второго финансового института, подсоединенного к указанному множеству коммутаторов через сеть передачи данных; принимают информацию от одного или более из множества коммутаторов, указывающую обновленное операционное состояние указанного второго финансового института; и обновляют операционное состояние второго финансового института, хранящееся в указанной памяти, на основе информации, полученной от одного или более коммутаторов, при этом в случае, когда операционное состояние второго финансового института указывает, что второй финансовый институт находится в нерабочем состоянии, способ содержит этап, на котором предотвращают передачу на указанные коммутаторы электронных транзакционных сообщений, поручающих выполнение транзакции от первого финансового института на второй финансовый институт.

В варианте осуществления информация, указывающая обновленное операционное состояние второго финансового института, создается одним или более из множества коммутаторов на основе отсутствия ответа на тестовое сообщение, переданное во второй финансовый институт каждым из коммутаторов в течение заданного периода времени, или указания вторым финансовым институтом одному или более из множества коммутаторов о том, что второй финансовый институт следует отсоединить от сети передачи данных.

Настоящее изобретение обеспечивает компьютерный программный продукт, содержащий запоминающий носитель, на котором хранится компьютерная программа; причем компьютерная программа при ее выполнении на компьютере конфигурирует указанный компьютер для выполнения вышеупомянутого способа по настоящему изобретению.

Вышеуказанные абзацы обеспечены в качестве общего введения и не предусматривают ограничение объема нижеследующей формулы изобретения. Описанные варианты осуществления вместе с дополнительными преимуществами лучше всего станут понятными, если обратиться к последующему подробному описанию вместе с сопроводительными чертежами.

Краткое описание чертежей

Более полную оценку изобретения и понимание многих присущих ему преимуществ легко получить, обратившись к нижеследующему подробному описанию, при его рассмотрении вместе с сопроводительными чертежами, на которых

фиг. 1 - различные компоненты системы согласно варианту осуществления;

фиг. 2А-2В - блок-схема, показывающая различные этапы обработки, применяемые к транзакционному сообщению и выполняемые частью системы, образующей коммутаторы, согласно варианту осуществления;

фиг. 3А - иллюстрация проблемы, связанной с реализацией использования дебетового лимита в системе;

фиг. 3В - иллюстрация технического решения проблемы, показанной на фиг. 3А, согласно варианту осуществления;

фиг. 4 - блок-схема, иллюстрирующая различные этапы обработки, применяемые указанной системой к транзакционному сообщению, согласно варианту осуществления; и

фиг. 5 - компьютер согласно варианту осуществления.

Подробное описание изобретения

Обратимся теперь к чертежам, на которых одинаковые ссылочные позиции обозначают идентичные или соответствующие детали на всех видах.

На фиг. 1 показан общий вид системы 100 согласно варианту осуществления. Система содержит множество банков (в данном случае два банка: банк 1 и банк 2) и множество коммутационных пунктов (в данном случае два коммутационных пункта: коммутационный пункт 1 и коммутационный пункт 2). Каждый банк соединен с обоими коммутационными пунктами через канал обмена данными (не показан). Канал обмена данными можно реализовать через сеть обмена данными, такую как Интернет. Для повышения надежности коммутационные пункты разделены, так что в случае возникновения проблемы, такой как технологический отказ или природный катаклизм на одном коммутационном пункте, обработка может продолжаться на другом коммутационном пункте. Например, коммутационные пункты могут находиться в разных географических точках.

Каждый банк содержит банковское приложение 102А, 102В, приложение-шлюз 104А, 104В (называемое также просто "шлюз") и блок 106А, 106В очереди сообщений. Каждый из коммутационных пунктов содержит блок 108А, 108В очереди сообщений, множество коммутаторов (в данном случае два коммутатора на каждом пункте: коммутаторы SW1 и SW2 на коммутационном пункте 1 и коммутаторы SW3 и SW4 на коммутационном пункте 2), базу 110А, 110В данных и блок 112А, 112В очереди сообщений. Кроме того, один из коммутационных пунктов (в данном случае коммутационный пункт 1) содержит операционный блок 114. Функция каждой из этих компонент подробно описана ниже. Заметим, что соответствующие компоненты соответствующих банков или коммутационных пунктов (указанных одинаковым числом, но разной буквой; например, базы 110А и 110В данных в коммутационных пунктах 1 и 2 соответственно и очереди 106А и 106В сообщений в банках 1 и 2 соответственно) функционально идентичны.

На фиг. 1 показан примерный поток сообщений, реализованный системой 100, с тем чтобы дать возможность выполнить транзакцию между банковскими счетами. В примере, представленном на фиг. 1, выполняется транзакция денежных средств с банковского счета в банке 1 на другой банковский счет в банке 2. Однако следует понимать, что систему 100 можно также использовать для выполнения транзакции денежных средств между разными банковскими счетами одного и того же банка. Более того, банк 2, конечно, может создавать запросы транзакций, которые обеспечивают перечисление денежных средств со счета, находящегося в банке 2 на счет в банке 1.

Поток сообщений для реализации указанной транзакции между банком 1 и банком 2 запускается, когда держатель банковского счета в банке 1 поручает банку 1 перечислить средства на банковский счет, поддерживаемый в банке 2. Результатом этого является создание сообщения М1 с запросом транзакции. Сообщение М1 содержит идентификатор банка и конкретного банковского счета, с которого должны быть перечислены средства (например, код типа банка 1 и номер счета), идентификатор банка и конкретного банковского счета, на который должны быть перечислены средства (например, код типа банка 2 и номер счета) и сумма и денежная валюта, подлежащая перечислению. Сообщение М1 также будет содержать уникальный идентификатор, который уникальным образом идентифицирует транзакцию, к которой относится сообщение М1 (например, уникальное число, уникальная комбинация букв, уникальная комбинация букв и чисел или какая-либо другая комбинация). Эта информация обеспечивается банковским приложением 102А. Банковское приложение 102А может быть обеспечено банком и может быть доступно клиенту, использующему защищенный вебпункт или мобильное приложение, такое как приложение, обеспеченное на мобильном терминале или планшетном компьютере клиента. Этот уникальный идентификатор называется "идентификатором транзакции".

Сообщение М1 поступает в блок 106А очереди сообщений, в котором оно будет временно храниться, пока шлюз 104А не станет доступным для обработки этого сообщения М1. Блок 106А очереди сообщений хранит сообщение М1 вместе с другими сообщениями, которые должны быть посланы на шлюз 104 в порядке очереди (так, что сообщение, полученное раньше, посылается на шлюз 104А перед посылкой сообщения, полученного позднее). Шлюз 104А затем ставит это сообщение во главе очереди, когда он окажется доступным.

Когда шлюз 104А доступен для приема сообщения М1, блок 106А очереди сообщений подает сообщение М1 на шлюз 104А. Шлюз 104А гарантирует, что сообщение М1 будет в заданном стандартизованном формате для использования с системой 100 (например, в стандартизованном формате ISO20022) и ставит цифровую подпись на сообщении М1. Шлюз 104А также добавляет к сообщению М1 маршрутную информацию (например, к заголовку сообщения М1), идентифицирующую коммутационный пункт,

на который следует послать M1. Функционирование шлюза 104A более подробно описывается ниже. Как будет ясно позднее, шлюз 104A является опционным элементом указанной системы, но он обеспечен в вариантах осуществления изобретения. При отсутствии шлюза 104A сообщение M1 необходимо будет правильно отформатировать и подписать, используя банковское приложение 102A, а также правильно направить.

После проверки формата сообщения M1, выполнение его цифровой подписи и добавления в него маршрутной информации сообщение M1 возвращают в блок 106A очереди сообщений для временного хранения перед тем, как передать его в блок 108A, 108B очереди сообщений на указанных коммутационных пунктах. Опять же, блок 106A очереди сообщений хранит сообщение M1 вместе с другими сообщениями, которые должны быть посланы на коммутационные пункты в порядке очереди. Первые сообщения из очереди в блоке 106A очереди сообщений посылают в блок 108A очереди сообщений, входящий в коммутационный пункт 1, или в блок 108B очереди сообщений, входящий в коммутационный пункт 2, в соответствии с маршрутной информацией в каждом сообщении. Шлюз каждого банка, как правило, направляет каждое последующее сообщение попеременно на указанные два коммутационных пункта. Таким образом, если, например, сообщение из начала очереди в блоке 106A очереди сообщений посылается в блок 108A очереди сообщений, то следующее сообщение в очереди будет послано в блок 108B очереди сообщений, следующее сообщение будет послано в блок 108A очереди сообщений и т.д. В этом случае очевидно, что сообщение M1 посылается в блок 108A очереди сообщений, входящий в коммутационный пункт 1.

Сообщение M1 временно хранится в блоке 108A очереди сообщений, пока один из коммутаторов SW1 или SW2 не станет доступным для обработки сообщения M1. Опять же, блок 109A очереди сообщений хранит сообщение M1 вместе с другими сообщениями, которые должны быть посланы на коммутаторы SW1 или SW2 в порядке очереди, так что сообщение из начала очереди посылается на тот из коммутаторов SW1 или SW2, который первым окажется доступным. Заметим, что очереди 106A, 106B сообщений на стороне банка и очереди 108A, 108B на стороне системы можно реализовать, используя такую систему, как IBM® MQ.

В этом случае сообщение M1 посылают на коммутатор SW1. Затем коммутатор SW1 применяет функцию хэширования к сообщению M1. Функция хэширования зависит от транзакционного идентификатора сообщения M1 и количества доступных коммутаторов на всех пунктах (в этом варианте осуществления четыре коммутатора), а выходом функции хэширования является число, идентифицирующее коммутатор, который должен быть использован для обработки сообщения M1 (например, число 1, 2, 3 или 4 для коммутаторов SW1, SW2, SW3 или SW4 соответственно). Коммутатор, идентифицированный функцией хэширования, может совпадать с коммутатором, на который изначально посылается сообщение M1 блоком 108A очереди сообщений, или, как альтернативный вариант, это может быть другой коммутатор. Если коммутатор, идентифицированный функцией хэширования, является другим коммутатором, то тогда исходный принимающий коммутатор (коммутатор SW1 в данном случае) посылает сообщение M1 на коммутатор, идентифицированный функцией хэширования. Однако в варианте осуществления по фиг. 1, коммутатор, идентифицированный функцией хэширования, совпадает с коммутатором, на который было принято сообщение M1 (то есть, коммутатор SW1), и поэтому данное сообщение не пересылается на другой коммутатор. Алгоритм хэширования более подробно описывается ниже.

После приема сообщения M1 коммутатором SW1 в базу 110A данных записывают состояние транзакции, к которой относится сообщение M1. В этом случае в базе 110A данных будет записано, что данная транзакция еще не закончена, и что на настоящий момент имеется одно сообщение (сообщение M1) из трех сообщений, необходимых для завершения транзакции, которое уже принято или передано коммутатором SW1 (этими тремя сообщениями являются сообщения M1, M2 и M3, смотри ниже).

База 110A данных представляет собой кластерную базу данных, созданную на основе данных, хранящихся в обоих блоках 109A и 109B памяти. Базу 110A данных можно реализовать, например, в виде базы данных Mnesia. Коммутаторы SW1, SW2 иницируют запись и обновление состояния транзакции для каждой транзакции, обрабатываемой этими коммутаторами, в базе 110A данных (это обозначено стрелкой TU на фиг. 1). Блоки 109A, 109B памяти синхронизированы друг с другом, так что каждое состояние транзакции записывается и обновляется в обоих блоках 109A и 109B памяти. Таким образом, блоки 109A, 109B памяти содержат идентичные копии одних и тех же данных, и каждое состояние транзакции в базе 110A данных можно извлечь либо из блока 109A памяти, либо из блока 109B памяти. Это значит, что состояния транзакции в базе 110A данных остаются доступными, когда один из блоков 109A, 109B памяти оказывается в нерабочем состоянии (из-за отказа, запланированного технического обслуживания или т.п.), что является преимуществом такого подхода.

База 110B данных имеет настройку, идентичную настройке базы 110A данных в том, что база 110B данных является кластерной базой данных, созданной на основе данных, хранящихся в обоих блоках 109C, 109D памяти. Коммутаторы SW3, SW4 иницируют запись и обновление состояния транзакции для каждой транзакции, обрабатываемой этими коммутаторами, в базе 110B данных. Блоки 109C, 109D памяти синхронизированы друг с другом, так что каждое состояние транзакции записывается и обновляется в обоих блоках 109C и 109D памяти. Таким образом, блоки 109C, 109D памяти содержат идентич-

ные копии одних и тех же данных, и каждое состояние транзакции в базе 110В данных можно извлечь либо из блока 109С памяти, либо из блока 109D памяти.

После записи состояния транзакции в базе 110А данных вслед за приемом сообщения М1 коммутатором SW1, коммутатор SW1 создает информационное сообщение М2 о транзакции. Информационное сообщение М2 о транзакции предназначено для информирования банка, который должен получить транзакционные средства (то есть, банк 2). Банк-получатель также информируется о конкретном банковском счете, на котором должны быть получены денежные средства, сумме и денежной валюте, которая должна быть получена. Таким образом, как и в случае с сообщением М1, содержащим запрос транзакции, сообщение М2 содержит идентификатор банка и конкретного банковского счета, на который должны быть перечислены средства (например, код типа банка 2 и номер счета), сумму и денежную валюту, подлежащую перечислению. Сообщение М2 также содержит тот же идентификатор транзакции, что и сообщение М1 (поскольку сообщение М1 и сообщение М2 относятся к одной и той же транзакции). Сообщение М2 также может содержать идентификатор банка и конкретного банковского счета, с которого посылаются средства (например, код типа банка 1 и номер счета) с тем, чтобы держатель счета банка-получателя смог узнать идентификационные данные отправителя.

После создания сообщения М2 оно передается в банк 2 через блоки 108А и 106В очереди сообщений и шлюз 104В. Блок 106В очереди сообщений и шлюз 104В банка 2 функционально идентичны блоку 106А очереди сообщений и шлюзу 104А банка 1. Таким образом, сообщение М2 временно хранится и находится в очереди в блоке 108А очереди сообщений. Затем оно передается по каналу обмена данными на блок 106В очереди сообщений (на основе идентификатора банка 2 в сообщении М2), где оно вновь временно хранится и содержится в очереди, пока шлюз 104В не станет доступным для его приема. После приема шлюзом 104В сообщения М2 проверяют (с использованием цифровой подписи, созданной указанным коммутатором). Затем сообщение М2 возвращают в блок 106В очереди сообщений, где оно временно хранится и находится в очереди до пересылки в банковское приложение 102В банка 2. Таким образом, банк 2 уведомляют о транзакции и конкретном базовом счете, на который должны быть получены денежные средства. Таким образом, можно выполнить кредитование этого конкретного банковского счета с использованием указанных транзакционных средств, если даже расчеты между банком 1 и банком 2 еще не закончены.

После передачи сообщения М2 банку 2 обновляют состояние транзакции в базе 110А данных, создавая запись о том, что транзакция еще продолжается, и что два сообщения (сообщение М1 и М2) из трех сообщений, необходимых для завершения транзакции, уже принято или передано коммутатором SW1.

После успешной обработки сообщения М2 банковским приложением 102В банка 2 банковское приложение 102В создает первое сообщение М3 о подтверждении транзакции. Сообщение М3 является подтверждением того, что сообщение М2 было успешно принято и обработано банком 2 и содержит тот же идентификатор транзакции, что и сообщения М1 и М2 (поскольку сообщения М1, М2 и М3 относятся к одной и той же транзакции). Сообщение М3 временно хранится и находится в очереди в блоке 106В очереди сообщений, а затем передается на шлюз 104В для обеспечения соответствующего стандартизованного формата и для добавления цифровой подписи и маршрутной информации. Затем сообщение М3 возвращают в блок 106В очереди сообщений для временного хранения и нахождения в очереди до его пересылки на один из коммутационных пунктов. Как и в блоке 106А очереди сообщений, сообщения в начале очереди в блоке 106В очереди сообщений, как правило, попеременно, посылают в блок 108А очереди сообщений, входящего в коммутационный пункт 1, и блок 108В очереди сообщений, входящий в коммутационный пункт 2, в соответствии с маршрутной информацией, добавленной к каждому сообщению шлюзом 104В. В этом случае очевидно, что сообщение М3 посылают в блок 108В очереди сообщений, входящий в коммутационный пункт 2.

Сообщение М3 временно хранится и находится в очереди в блоке 108В очереди сообщений, пока один из коммутаторов SW3 или SW4 не станет доступным для приема сообщения М3. В этом случае очевидно, что коммутатор SW4 будет первым коммутатором, который будет доступен, когда сообщение М3 находится в начале очереди, и поэтому сообщение М3 посылается на коммутатор SW4.

В коммутаторе SW4 к сообщению М3 применяется алгоритм хэширования. Поскольку сообщение М3 имеет такой же идентификатор транзакции, как сообщение М1 (и сообщение М2), на выходе алгоритма хэширования будет такой же результат, как на выходе алгоритма хэширования для сообщения М1. То есть, выходом алгоритма хэширования будет указание на то, что сообщение М3 следует переслать на коммутатор SW1. Таким образом, как видно из фиг. 1, коммутатор SW4 передает сообщение М3 на коммутатор SW1. Эта передача выполняется по каналу обмена данными, соединяющему коммутационные пункты 1 и 2.

После приема сообщения М3 коммутатором SW1 состояние транзакции в базе 110А данных обновляется, чтобы обеспечить запись о завершении транзакции, поскольку все сообщения (сообщения М1, М2 и М3), необходимые для завершения транзакции уже приняты или переданы коммутатором SW1. В этом случае состояние транзакции перемещают из промежуточной таблицы в базе 110А данных, в которую записывают подробности выполняющихся транзакций, в окончательную таблицу в базе 110А дан-

ных, куда записывают подробности завершенных транзакций.

Указанные три сообщения M1, M2 и M3 достаточны для определения того, является ли транзакция завершенной, поскольку после приема сообщения M3 подтверждается, что информация о платеже, содержащаяся в сообщении M2 получена и успешно обработана банком 2. Таким образом, та же сумма, которая дебетуется со счета-отправителя (банк 1) после передачи сообщения M1, кредитруется на счет-получатель (банк 2). Заметим, что это позволяет клиентам банка 1 и банка 2 пользоваться мгновенной пересылкой денежных средств, хотя окончательные расчеты между банком 1 и банком 2 еще не оформлены.

После обновления состояния транзакции для выполнения записи в базе 110A данных о том, что транзакция завершена, создается итоговая запись TS о транзакции на основе указанного переходного состояния. Итоговая запись TS о транзакции содержит достаточную информацию для выполнения расчетов между финансовыми счетами, определенными транзакционными сообщениями, причем эта информация хранится в блоке 112A очереди сообщений. В частности, итоговая запись TS о транзакции содержит имя, адрес, код типа и номер счета отправителя (источника) денежных средств, адрес, код типа и номер счета получателя (бенефициара) этих средств, сумму денежных средств, подлежащую перечислению (включая название валюты, в которой должно быть выполнено перечисление) и уникальный идентификатор транзакции. Итоговая запись TS о транзакции также может содержать дополнительную информацию, такую как время и дата создания указанной записи, временные метки, относящиеся к посылке или приему транзакционных сообщений (M1, M2 и M3), дату и время завершения расчетов (или другой индикатор конкретного расчетного цикла, в котором должна обрабатываться данная транзакция) и индикатор того, оказалась ли данная транзакция успешной. Конечно, этот список не является исчерпывающим, то есть, и в итоговую запись NS о транзакции может быть включена любая информация, которая может оказаться полезной при выполнении расчетов для всех транзакций.

Хотя это на фиг. 1 не показано, блок 112A очереди сообщений, который также имеет вид кластерной базы данных, где в отдельных блоках памяти хранятся две копии всех итоговых записей TS о транзакции в отдельных блоках памяти. Эти две копии, будучи синхронизированными, повышают надежность системы 100, поскольку, если одна копия итоговых записей о транзакциях окажется недоступной (из-за технического отказа или т.п.), можно будет использовать вторую копию. Блок 112A очереди сообщений временно хранит указанную итоговую запись о транзакции, устанавливая для нее очередь, пока операционный блок 114 (смотри ниже) не окажется готовым принять итоговую запись о транзакции для обработки расчетов по транзакции.

Создание итоговой записи TS о транзакции и функционирование блоков 112A и 112B очереди сообщений может быть реализовано с использованием программного приложения, такого как, например, Rabbit MQ®. В действительности, кластерная база 110A данных и блок 112A очереди сообщений (и аналогично кластерная база 110B данных и блок 112B очереди сообщений) можно функционально объединить, используя указанное программное приложение, так что, хотя кластерная база данных и блок очереди сообщений на каждом пункте в действительности является совершенно отдельными объектами, функциональные возможности считывания и записи между ними (включая создание итоговой записи TS о транзакции) реализуются достаточно легко и надежно.

Вслед за последующим хранением итоговой записи TS о транзакции в блоке 112A очереди сообщений коммутатор SW1 создает второе сообщение M4 подтверждения транзакции. Сообщение M4 является сообщением для подтверждения для банка 1 успешного выполнения транзакции в банке 2, причем это сообщение содержит тот же идентификатор транзакции, что и сообщения M1, M2 и M3 (поскольку все сообщения M1, M2, M3 и M4 относятся к одной и той же транзакции). Сообщение M4 передается в банк 1 через блоки 108A и 106A очереди сообщений и шлюз 104A. То есть, сообщение M4 временно хранится и находится в очереди в блоке 108A очереди сообщений. Затем оно передается по каналу обмена данными в блок 106A очереди сообщений, где оно снова временно хранится и находится в очереди, пока шлюз 104A не станет доступным для его приема. После приема шлюзом 104A сообщение M4 проверяется (с использованием цифровой подписи, созданной упомянутым коммутатором). Затем сообщение M4 возвращают в блок 106A очереди сообщений, где оно временно хранится и находится в очереди, прежде чем будет послано в банковское приложение 102A банка 1. Таким образом, банк 1 получает подтверждение об успешном перечислении денежных средств.

В случае, когда держатель счета в банке-поручителе (банк 1 в примере на фиг. 1) сначала передает сообщение M1 на коммутатор-приемник через Интернет, используя запрашивающее приложение, и обрабатывающий коммутатор (определенный функцией хэширования) не совпадает с исходным принимающим коммутатором (заметим, что в примере по фиг. 1 рассматривается другой случай, поскольку обрабатывающий коммутатор SW1 также является коммутатором, который первоначально принял сообщение M1), то сообщение M4 сначала будет направлено на исходный принимающий коммутатор от обрабатывающего коммутатора, а затем передается в банк-поручитель (через соответствующие блоки 108A, 108B, 106A, 106B) от исходного принимающего коммутатора. Это необходимо для обеспечения возврата сообщения M4 в то же запрашивающее приложение, которое послало сообщение M1 для данной транзакции.

Операционный блок 114 сконфигурирован для обработки итоговых записей о транзакции, хранящихся и находящихся в очереди в блоках 112А, 112В очереди сообщений для предоставления возможности выполнения всех расчетов между банком 1 и банком 2. Это возможно, поскольку каждая из итоговых записей о транзакции содержит всю информацию, необходимую для отслеживания суммы средств, перечисляемых из банка 1 в банк 2 (или наоборот) для каждой транзакции, и, следовательно, в конце расчетного цикла общая сумма средств, подлежащая перечислению от одного банка к другому, может быть вычислена с использованием упомянутых итоговых записей о транзакции. Это гарантирует соответствие кредитования и дебетования банковских счетов для каждой транзакции с реальной денежной суммой при нетто расчетах между банками. Итоговые записи TS о транзакциях хранятся и находятся в очереди 112А сообщений перед их передачей в операционный блок 114 для обработки. Обработка в операционном блоке 114 содержит запоминание итоговых записей TS о транзакции и поддержание нарастающим итогом двусторонних (банк-банк) и многосторонних (банк-все-другие банки) обязательств для каждого банка (то есть, задолженность) для использования во взаимных расчетах. Затем в конце расчетного цикла (например, каждые 8, 12 или 24 часа, как было описано выше) происходят взаимные расчеты на основе указанных данных, поддерживаемых операционным блоком 114.

Заметим, что в данном варианте осуществления операционный блок 114 находится на коммутационном пункте 1. Таким образом, итоговые записи TS о транзакции, хранящиеся в блоке 112В очереди сообщений на коммутационном пункте 2, должны передаваться по каналу обмена данными между коммутационными пунктами для их приема операционным блоком 114. В альтернативном варианте осуществления операционный блок 114 может находиться на коммутационном пункте 2, и в этом случае итоговые записи TS о транзакции, хранящиеся в блоке 112А очереди сообщений на коммутационном пункте 1, должны передаваться по каналу обмена данными между коммутационными пунктами, с тем чтобы их мог принимать операционный блок 114. В другом варианте осуществления имеется множество операционных блоков 114 по одному на каждом из коммутационных пунктах 1 и 2. В этом случае для расчетов будет использоваться только один из операционных блоков. Однако наличие множества операционных блоков 114 означает, что, если один из операционных блоков 114 окажется неработоспособным (из-за отказа или запланированного технического обслуживания или т.п.), то итоговые записи TS о транзакции, хранящиеся в блоках 112А, 112В очереди сообщений, могут быть перенаправлены на остальные операционные блоки 114 для расчетов. Таким образом, система 100 отличается повышенной надежностью, поскольку расчеты могут выполняться с использованием итоговых записей TS о транзакции, несмотря на отказ одного из операционных блоков 114.

Распределение запросов транзакции

Как уже упоминалось, в варианте осуществления по фиг. 1 транзакционные сообщения (в частности, сообщения M1 и M3, которые получают от банков) распределяются между коммутаторами SW1, SW2, SW3 и SW4 посредством функции хэширования (также называемой алгоритмом хэширования). Это помогает обеспечить равномерное распределение сообщений по коммутаторам, что улучшает баланс нагрузки системы 100. Это также помогает обеспечить обработку сообщений, относящихся к одной и той же транзакции, одним и тем же коммутатором или по меньшей мере теми коммутаторами, которые имеют доступ к одной и той же базе 110А и 110В данных. Далее более подробно раскрывается функция хэширования.

Как пояснялось ранее, функция хэширования отображает каждое сообщение на тот или иной коммутатор. В частности, функция хэширования использует уникальный транзакционный идентификатор сообщения и количество коммутаторов в качестве входных данных и выдает число, указывающее, какой из коммутаторов SW1, SW2, SW3 и SW4 должен быть использован для обработки данного сообщения. То есть, функция хэширования имеет вид:

Номер коммутатора=функция (транзакционный идентификатор, общее количество коммутаторов).

Конкретные примеры функций хэширования (то есть, функций, которые только выводят одно из фиксированного количества заданных выходных значений, хотя возможно большое или бесконечное количество входов), которые можно использовать, известны в данной области техники и далее подробно не обсуждаются.

В вариантах осуществления коммутатор, который первым принимает сообщение от одного из банков (или, в частности, от одного из блоков 108А, 108В очереди сообщений) выполняет хэширование для данного сообщения. Это идентифицирует коммутатор, который будет использован для обработки данного сообщения. Затем сообщение посылают (или, другими словами, направляют) на этот идентифицированный коммутатор. Эта функция реализуется в схемах обработки, находящихся в коммутаторе, с использованием машиночитаемых команд, хранящихся в блоке памяти в коммутаторе. Направление сообщения после применения функции хэширования в качестве примера соответствует фиг. 1 (где сообщение M3, которое изначально послано на коммутатор SW4, направляется на коммутатор SW1 после применения алгоритма хэширования). Функция хэширования используется для сообщений, полученных от банков (то есть, сообщений M1 и M3 в примере по фиг. 1), поскольку это те сообщения, которые изначально могли быть посланы на любой из коммутаторов SW1, SW2, SW3 и SW4.

Все сообщения, связанные с одной и той же транзакцией, содержат одинаковый транзакционный

идентификатор. Это позволяет направлять и обрабатывать такие сообщения одним и тем же коммутатором или по меньшей мере коммутаторами, которые имеют доступ к одной и той же базе 110А, 110В данных, если это возможно. Это является преимуществом, так как позволяет своевременно обновлять эффективно и легко соответствующую базу 110А, 110В данных в ходе конкретной транзакции.

Между тем, как преимуществом является возможность выделения одного идентифицированного коммутатора для обработки всех сообщений, связанных с конкретной транзакцией, в том случае, когда этот идентифицированный коммутатор перестает работать из-за повреждения, запланированного технического обслуживания или т.п., в вариантах осуществления изобретения обеспечен список приоритетов альтернативных коммутаторов, которые должны обрабатывать сообщения, связанные с данной транзакцией.

Пример порядка приоритетов коммутаторов поясняется в табл. 1, представленной ниже.

Таблица 1

Первый коммутатор	Второй коммутатор	Третий коммутатор	Четвертый коммутатор
SW1	SW2	SW3	SW4
SW2	SW1	SW4	SW3
SW3	SW4	SW1	SW2
SW4	SW3	SW2	SW1

В табл. 1 показан порядок приоритетов коммутаторов. Первый коммутатор - это коммутатор, выбранный алгоритмом хэширования для определения равномерного распределения сообщений по всем коммутаторам. Второй коммутатор - это следующий коммутатор в порядке приоритета, то есть, если первый коммутатор недоступен, сообщение для конкретной транзакции посылают на второй коммутатор. В этом случае второй коммутатор находится в том же месте, что и первый коммутатор. Другими словами, в случае, когда первый коммутатор находится в коммутационном пункте 1, второй коммутатор также будет находиться в коммутационном пункте 1. Третий коммутатор - это следующий коммутатор в порядке приоритета, то есть, если второй коммутатор недоступен, сообщение посылают на третий коммутатор. В этом случае третий коммутатор должен находиться в другом месте по отношению к первому и второму коммутаторам, поскольку никакие другие коммутаторы недоступны в коммутационном пункте первого и второго коммутаторов. Наконец, четвертый коммутатор - это следующий коммутатор в порядке приоритета, то есть, если третий коммутатор недоступен, сообщение посылают на четвертый коммутатор. В этом случае четвертый коммутатор (который является единственным оставшимся коммутатором) должен находиться в том же пункте, что и третий коммутатор.

Существует два отдельных преимущества, связанных с предоставлением коммутаторов в порядке приоритета, как это показано в табл. 1. Первое преимущество состоит в том, что, если первый коммутатор отказал, то сообщение будет быстро обрабатываться активным коммутатором. Это обеспечивает быстрое перераспределение запроса транзакции. Во-вторых, поскольку второй коммутатор в порядке приоритета находится на том же пункте, что и первый коммутатор, обеспечивается, что в случае неработоспособности первого коммутатора сообщения посылаются на коммутатор, имеющий прямой доступ к той же базе 110А, 110В, что и первый коммутатор. Это позволяет продолжать обработку сообщений, как если бы первый коммутатор все еще находился в рабочем состоянии, поскольку второй коммутатор способен немедленно осуществить доступ к записям о состоянии транзакции и обновить их для незаконченных транзакций, обработка которых ранее осуществлялась первым коммутатором. Преимуществом является то, что такой подход сокращает прерывание обработки транзакций в том случае, когда один из коммутаторов на конкретном коммутационном пункте оказался в нерабочем состоянии.

В случае, когда оба коммутатора на конкретном коммутационном пункте оказались в нерабочем состоянии, наличие третьего и четвертого коммутаторов в списке приоритетов позволяет обрабатывать новые транзакции (то есть, транзакции, для которых сообщение M1 было выдано после того, как оба коммутатора на конкретном пункте оказались в нерабочем состоянии), продолжая свою работу. Преимуществом является то, что это позволяет выполнять новые порученные транзакции даже в том случае, когда оба коммутатора на указанном пункте оказались в нерабочем состоянии (возможно, например, в результате природного катаклизма в одном из коммутационных пунктов). Однако для транзакций, которые частично уже завершены, в коммутационном пункте, потерявшем работоспособность, транзакция не сможет продолжаться, поскольку доступ к данным о состоянии транзакции (которые хранятся в базе 110А, 110В данных в потерявшем работоспособность пункте) будет невозможен. Эта ситуация подробно описывается ниже.

Следует иметь в виду, что конкретные списки приоритетов для каждого коммутатора, показанные в табл. 1, являются лишь примером, и что также можно использовать альтернативные списки приоритетов. Более того, хотя в вышеприведенном описании обеспечиваются два коммутатора в двух пунктах, предусмотрено, что указанные принципы можно применить к любому количеству коммутаторов в любом количестве пунктов.

Повышение надежности системы

Как уже говорилось, в системе 100 данные о состоянии транзакции хранятся в базе 110A, 110B данных коммутационного пункта, используемого для обработки транзакции. Затем данные о состоянии транзакции используют для создания итоговой записи TS о транзакции, которая хранится и находится в очереди в блоке 112A, 112B очереди сообщений, пока операционный блок 114 не станет доступным для приема итоговой записи TS о транзакции в конце расчетного цикла. После успешного запоминания итоговой записи TS в соответствующем блоке 112A, 112B очереди сообщений данные о состоянии транзакции могут быть удалены из базы 110A, 110B данных. Кроме того, когда операционный блок 114 подтвердил соответствующему блоку 112A, 112B очереди сообщений, что суммарная запись TS о транзакции успешно получена, итоговую запись TS о транзакции можно удалить из блока 112A, 112B очереди сообщений.

Однако, если имеется проблема в коммутационном пункте, в котором хранится итоговая запись TS транзакции, которая препятствует доступу операционного блока 114 к блоку 112A, 112B очереди сообщений коммутационного пункта (несмотря на компоновку кластерной базы данных блоков 112A, 112B очереди сообщений), то тогда существует опасность того, что итоговая запись TS о транзакции не будет учтена в конце расчетного цикла. Это приведет к некорректным расчетам между банками (и, следовательно, к некорректному перечислению реальной денежной суммы между банками).

Чтобы избежать этой проблемы при передаче каждой итоговой записи TS о транзакции в блок 112A, 112B очереди сообщений обрабатывающего ее коммутационного пункта, на один из коммутаторов другого коммутационного пункта также направляется копия этой записи для запоминания в блоке резервной памяти (не показан). Например, каждая итоговая запись TS о транзакции, созданная в коммутационном пункте 1, и временно хранящаяся, находясь в очереди, в блоке 112A очереди сообщений, также будет скопирована и направлена на коммутатор SW3 или SW4 в коммутационном пункте 2 и будет храниться в блоке резервной памяти в коммутационном пункте 2. Это означает что, если по какой-то причине итоговые записи TS о транзакции, хранящиеся в блоке 112A, 112B очереди сообщений в конкретном пункте стали недоступными операционному блоку 114 в конце расчетного цикла (например, из-за отказа в блоке 112A, 112B очереди сообщений), то тогда резервные копии итоговых записей TS о транзакции можно будет извлечь и сделать доступными для операционного блока 114 с целью выполнения необходимых расчетов по транзакции. Вместе с использованием конфигурации кластерной базы данных для блоков 112A, 112B очереди сообщений это помогает обеспечить еще и возможность корректного выполнения расчетов между банками.

Как обсуждалось выше, в вариантах осуществления коммутаторы распределены по множеству пунктов, причем каждый пункт содержит кластер коммутаторов. В варианте осуществления, показанном на фиг. 1, имеется, например, четыре коммутатора SW1, SW2, SW3 и SW4, причем коммутаторы SW1 и SW2 образуют кластер в пункте 1, а коммутаторы SW3 и SW4 образуют кластер в пункте 2.

Каждый из коммутаторов SW1, SW2, SW3 и SW4 действуют независимо друг от друга. Следовательно, в случае, когда один из коммутаторов в кластере отказал (например, из-за ошибки или нарушения функционирования) или если один коммутатор в кластере должен быть отключен для технического обслуживания, то обработка транзакции все же может продолжаться на другом коммутаторе в этом кластере (который имеет доступ к той же базе 110A, 110B данных).

Этот принцип распространяется на более сложные ситуации, чем отказ одного коммутатора, в которых даже если отказали дополнительные коммутаторы, пока имеется по меньшей мере один коммутатор в системе в рабочем состоянии, обработка транзакции (по меньшей мере для вновь запущенных транзакций) может продолжаться. Например, если один коммутатор в каждом из пунктов 1 и 2 отказал (например, коммутаторы SW1 и SW3), если оба коммутатора в одном пункте отказали (например, коммутаторы SW1 и SW2 в пункте 1) или, если даже все коммутаторы отказали, кроме одного (например, коммутаторы SW1, SW2 и SW3 отказали и только оставшийся SW4 работоспособен, то тогда транзакционное сообщение будет перенаправлено на доступный коммутатор (с использованием функции хэширования), и обработка транзакции будет продолжаться.

Кроме того, указанный принцип не ограничивается отказами коммутаторов. Если произошел отказ других компонентов системы 100 (например, один из блоков 109A-D памяти), то это будет обнаружено соответствующим коммутатором, и тогда этот коммутатор инициирует направление сообщений на альтернативный коммутатор как это определено согласно функции хэширования. Следовательно, обработка транзакции будет продолжаться с использованием одного из других коммутаторов.

Указанный принцип также не сводится к случаю отказов компонент. Например, если коммутационный пункт закрыт на техническое обслуживание (например, обновление программного или аппаратного обеспечения или т.п.), тогда коммутаторы данного пункта могут оказаться недоступными для приема транзакционных сообщений. Однако прием и обработка транзакционных сообщений может выполняться коммутаторами в другом пункте, что позволяет продолжать обработку транзакций. Это также применимо, если весь пункт отключен, например, из-за природного катаклизма (такого, как пожар, затопление или т.п.).

Как уже упоминалось, транзакционные сообщения распределяют между коммутаторами SW1, SW2,

SW3, SW4 согласно спискам приоритетов коммутаторов, определенных функцией хэширования. Когда коммутатор сначала принимает транзакционное сообщение, он применяет к нему функцию хэширования, а затем направляет это сообщение (если необходимо) на первый коммутатор в списке приоритетов, который является доступным (с точки зрения системы), для обработки (непрерывный контроль доступности коммутаторов посредством алгоритма опроса описан более подробно ниже). Чтобы направить данное сообщение исходный принимающий коммутатор пытается установить соединение (такое как соединение Протокола управления передачами (TCP)) с первым коммутатором в списке приоритетов, а затем направить это сообщение после того, как будет успешно установлено указанное соединение. Если соединение не может быть успешно установлено (это может случиться, например, если имеет место отказ коммутатора с первым приоритетом), тогда определяют, что первый коммутатор в списке приоритетов недоступен, и выполняется попытка направления указанного сообщения на следующий коммутатор в списке приоритетов, который система считает доступным для обработки (опять же пытаюсь установить соединение со следующим коммутатором в списке приоритетов). В этом случае процесс повторяется, пока не будет найден доступный коммутатор.

Для сообщения M1 (которое представляет новую порученную транзакцию), пока один коммутатор в списке приоритетов доступен для приема транзакционного сообщения, обработка данной транзакции может быть завершена этим коммутатором с созданием итоговой записи TS о транзакции. Причина этого состоит в том, что данные о состоянии для конкретной транзакции сначала записывают на основе информации в сообщении M1, откуда следует, что не требуются предшествующие данные о состоянии (которые хранятся в базе 110A, 110B данных одного из коммутационных пунктов), подлежащие доступу, чтобы продолжать обработку транзакции. С другой стороны, для сообщения M3 обработка транзакции может быть завершена, только если коммутатор, на который направлено сообщение M3, находится в том же коммутационном пункте, что и коммутатор, на который было направлено сообщение M1 (и где было создано сообщение M2). Причина этого состоит в том, что данные о состоянии транзакции, хранящиеся в указанном коммутационном пункте, должны быть доступны, чтобы продолжать обработку транзакции. Следовательно, в случае, когда оба коммутатора в коммутационном пункте, где обрабатывалось сообщение M1, вышли из строя до приема сообщения M3, транзакция не может продолжаться, поскольку к данным о состоянии транзакции, хранящимся в неработающем коммутационном пункте, доступ невозможен. Этот сценарий более подробно описан ниже.

Для сокращения ширины непроизводительно используемой сетевой пропускной способности каждый коммутатор SW1, SW2, SW3, SW4 сконфигурирован для периодического опроса всех других коммутаторов, чтобы оценить, являются ли они доступными. Этот опрос содержит посылку тестового сообщения на каждый из других коммутаторов и прослушивание ответа. Если ответ от конкретного коммутатора не получен в течение заданного временного периода, то определяют, что этот конкретный коммутатор недоступен. Затем сообщения, связанные с транзакциями коммутаторами, определенные для посылки на этот коммутатор, посылают на следующий коммутатор в списке приоритетов. Благодаря периодическому выполнению этого опроса сетевая пропускная способность не затрачивается на попытки коммутатора установить соединение с недоступным коммутатором. Временной период между опросами определяют таким образом, чтобы избыточная пропускная способность, используемая для запроса, превышала компенсированную пропускную способность в результате ее сокращения, используемую в попытках установления соединений с недоступными коммутаторами. Соответствующий временной период может составить 100 мс, хотя возможны и другие периоды. Опрос коммутаторов с целью определения их недоступности продолжается, и, как только опрашиваемый коммутатор вновь окажется доступным, это станет известно другим коммутаторам, и тогда на коммутатор, который снова стал доступным, могут снова передаваться сообщения.

Таким путем коммутатор определяет, что другой коммутатор недоступен, после отрицательного результата опроса (то есть, когда ответ не получен), или, когда попытка соединения с коммутатором не удалась. В любом случае сообщение, передаваемое на недоступный коммутатор, будет передано на следующий доступный коммутатор согласно списку приоритетов. Этот способ также может указать на другие проблемы в системе, например, отказ линии обмена данными между коммутационными пунктами (например, если оба коммутатора SW1 и SW2 не способны к соединению с SW3 и SW4 или постоянно получают отрицательные результаты их опроса, то возможно этот тот случай, когда произошел отказ линии обмена данными между коммутационным пунктом 1 и коммутационным пунктом 2). В случае возникновения проблем обеспечивается техническое обслуживание отказавшего коммутатора, или коммутационного пункта быстрее, чем это было бы обеспечено без указанного опроса. Например, если опрос не обеспечен, отказавший коммутатор или коммутационный пункт будут идентифицированы только в том случае, когда сообщение, связанное с транзакцией, не удалось правильно передать. Частота такого события может быть гораздо меньше, чем частота появления сигнала опроса.

Банк-отправитель (банк 1 в примере по фиг. 1) узнает об успешном завершении транзакции, как только получит сообщение M4. Если сообщение M4 банком-отправителем не получено в течение заданного временного периода (известного как таймаут), то банк-отправитель не будет знать, была ли транзакция успешной. Этот таймаут может возникнуть по нескольким причинам. Например, коммутацион-

ный пункт, в котором обрабатывалось сообщение M1, оказался в неработоспособном состоянии до завершения транзакции (например, до приема сообщения M3), и это означает, что итоговая запись TS никогда не создавалась, и что никогда не создавалось сообщение M4. Эта транзакция оказывается неудачной. С другой стороны, возможен случай, когда транзакция была успешно завершена, и было создано сообщение M4, которое затем потерялось в системе 100 или задержалось настолько, что период таймаута истек. В случае, когда сообщение M4 не принято в течение периода таймаута, банк-отправитель может передать сообщение M1 повторно. На фиг. 2А-2В показано, как отдельные коммутаторы обрабатывают транзакционные сообщения, в том числе, как отдельные коммутаторы обрабатывают повторно переданные транзакционные сообщения от банка-отправителя.

На фиг. 2А представлена блок-схема, иллюстрирующая процесс, выполняемый коммутатором, который изначально принял транзакционное сообщение от одного из блоков 108А, 108В очереди сообщений, согласно варианту осуществления.

Процесс начинается с этапа 200. На этапе 202 транзакционное сообщение (например, сообщение M1 или M3, как показано на фиг. 1) принимается коммутатором из блока 108А, 108В очереди сообщений. На этапе 204 к сообщению применяется функция хэширования, идентифицирующая список приоритетов коммутаторов, на который данное сообщение должно быть направлено. На этапе 206 определяют, доступен ли первый коммутатор в списке приоритетов для приема указанного сообщения. Определяют это на основе или опроса первого коммутатора, или предпринимая попытку установить соединение с первым коммутатором, как было описано ранее. Если определено, что первый коммутатор доступен, то процесс переходит к этапу 208, на котором указанное сообщение направляют на первый коммутатор. Затем на этапе 210 процесс заканчивается. С другой стороны, если определено, что первый коммутатор недоступен, то процесс переходит к этапу 212.

На этапе 212 определяют, является ли доступным следующий коммутатор в списке приоритетов для приема указанного сообщения. Опять же определить это можно на основе либо опроса, либо попытаться установить соединение со следующим коммутатором. Если определено, что следующий коммутатор доступен, то процесс переходит к этапу 214, на котором сообщение направляют на следующий коммутатор. Затем процесс на этапе 210 заканчивается. С другой стороны, если определено, что следующий коммутатор недоступен, то этап 212 повторяют для следующего коммутатора в списке приоритетов (например, если второй коммутатор в списке приоритетов был определен как недоступный, то тогда определяют, доступен ли третий коммутатор в списке приоритетов). Этап 212 будет повторяться пока не будет обнаружен доступный коммутатор, и тогда процесс переходит к этапу 214, и на этот доступный коммутатор будет направлено указанное сообщение. Заметим, что доступный коммутатор в конце концов обязательно должен быть найден, если процесс по фиг. 2А выполняется, поскольку коммутатор, способный выполнять процесс по фиг. 2А, конечно будет доступен для приема сообщения.

На фиг. 2В представлена блок-схема, иллюстрирующая процесс, выполняемый коммутатором, на который направлено сообщение согласно списку приоритетов.

Процесс начинается на этапе 216. На этапе 218 коммутатор принимает указанное сообщение. На этапе 220 определяют, является ли данное сообщение первым сообщением, относящимся к транзакции (то есть, сообщение M1). Если сообщением является сообщение M1, то процесс переходит к этапу 222, на котором определяют, является ли полученное сообщение M1 повторением сообщения M1. Повторение сообщения M1 - это повторная передача сообщения M1 для конкретной транзакции из банка-отправителя, которое может появиться, когда банк-отправитель ранее послал сообщение M1, но не принял обратно сообщение M4 в течение периода таймаута.

Определение на этапе 222 можно выполнить путем проверки идентификатора транзакции из указанного сообщения M1. Если сообщение M1 - это повтор, то исходное сообщение M1 возможно уже было обработано в коммутационном пункте, и, следовательно, идентификатор транзакции (который одинаков для исходного сообщения M1 и его повтора для конкретной транзакции) будет записан в коммутационном пункте (например, в базе 110А, 110В данных в качестве части данных о состоянии транзакции). Таким образом, совпадение идентификатора транзакции из вновь полученного сообщения M1 с идентификатором транзакции, записанным в коммутационном пункте, укажет, что сообщение M1 является повтором. Вдобавок или в качестве альтернативы, повторное сообщение M1 может включать в себя информацию о повторе (например, в заголовке сообщения), указывающую, что это повторное сообщение (указанная информация о повторе может быть добавлена в сообщение M1, например, банковским приложением 102А, 102В банка-отправителя).

В случае определения сообщения M1 как повторного, процесс переходит на этап 224. С другой стороны, если в коммутационном пункте не было обработано ни одно из предшествующих сообщений, относящихся к данной транзакции, с которой связано сообщение M1, то определяют, что сообщение M1 не было повторным, и действительно является первым сообщением, связанным с конкретной транзакцией. В этом случае процесс переходит к этапу 230, на котором выполняется обычная обработка сообщения M1.

Заметим, что использование информации о повторе, включенной в повторное сообщение M1 является значительным преимуществом. Например, рассмотрим сценарий, в котором не используется инфор-

мация о повторе, и в котором весь коммутационный пункт оказывается неработоспособным после приема сообщения M3 и создания итоговой записи TS о транзакции (указывающей об успешном завершении данной транзакции), но перед передачей сообщения M4. Отсутствие сообщения M4 инициирует повторную передачу банком-отправителем сообщение M1, которое затем будет перенаправлено в альтернативный коммутационный пункт. У альтернативного коммутационного пункта не будет записи с идентификатором транзакции (поскольку итоговая запись TS о транзакции хранится в базе 110A, 110B данных исходного коммутационного пункта) и, следовательно, он не сможет определить является ли указанное сообщение M1 повтором, просто полагаясь на совпадение идентификаторов транзакции. Таким образом, транзакция, связанная с сообщением M1, будет обрабатываться как новая транзакция в альтернативном коммутационном пункте, хотя данная транзакция в действительности уже завершена в исходном коммутационном пункте (путем создания итоговой записи TS о транзакции). В результате транзакция окажется некорректно обработанной, то есть, дважды (имея ввиду перечисление удвоенной денежной суммы, по сравнению с суммой, перечисляемой отправителем согласно расчетам по данной транзакции). Однако использование информации о повторе позволяет определить, что данное сообщение является повторно переданным сообщением M1, любым коммутатором в любом коммутационном пункте (а не только в коммутационном пункте, где было обработано исходное сообщение M1 и был запомнен идентификатор транзакции). Таким образом, использование информации о повторе снижает риск некорректной обработки сообщения M1 в качестве нового сообщения M1 больше одного раза, что уменьшает риск обработки одной и той же транзакции более одного раза.

На этапе 224 определяют, доступно ли в коммутационном пункте сообщение M4 для транзакции, с которой связано повторное сообщение M1 (например, если сообщение M4 хранится в базе 110A, 110B данных в указанном коммутационном пункте). Если сообщение M4 доступно, из это вытекает, что итоговая запись TS для данной транзакции была создана, и что транзакция в указанном коммутационном пункте была завершена. В этом случае процесс переходит к этапу 226, на котором сообщение M4 повторно передается банку-отправителю. Таким образом, это подтверждает банку, что транзакция была завершена успешно, даже если исходно переданное сообщение M4 не было принято банком-отправителем (или по меньшей мере не принято до окончания периода таймаута). Затем процесс заканчивается на этапе 232.

С другой стороны, если сообщение M4 недоступно, из этого вытекает, что имеет место неопределенность, касающаяся того, была ли успешно завершена указанная транзакция. Это может случиться, если, например, оба коммутатора в коммутационном пункте, где изначально обрабатывалась транзакция (и где создается сообщение M4, если транзакция была успешно завершена) оказались недоступны (из-за технического отказа или т.п.). В этом случае состояние транзакции неизвестно, и невозможно получить сообщение M4. Таким образом, процесс заканчивается на этапе 232. Следовательно, транзакция должна быть изучена вручную в конце расчетного цикла (смотри ниже).

Если на этапе 220 не определено, что сообщение является сообщением M1, тогда этим сообщением должно быть сообщение M3, которое получают от банка, принимающего денежные средства по транзакции. В этом случае транзакция может быть завершена, если только коммутатор, реализующий процесс по фиг. 2B, имеет доступ к данным о состоянии данной транзакции (то есть, этот коммутатор должен находиться в том же пункте, в котором было обработано сообщение M1, и который имеет доступ к базе 110A, 110B данных, где хранятся данные о состоянии транзакции). Если коммутатор имеет доступ к данным о состоянии транзакции, то процесс переходит к этапу 230, на котором сообщение M3 обрабатывается обычным образом. Это позволяет завершить транзакцию (с созданием в результате сообщения M4). Затем процесс на этапе 232 заканчивается. С другой стороны, если коммутатор не имеет доступа к данным о состоянии транзакции (что получается, если все коммутаторы в пункте, где база данных с данными о состоянии транзакции оказалась в нерабочем состоянии до выдачи сообщения M4), то данную транзакцию завершить будет невозможно. Таким образом, процесс просто закончится на этапе 232 (без создания сообщения M4).

Таким образом, можно видеть, что, если банк-отправитель не получает сообщение M4, подтверждающее успешное выполнение транзакции, исходное сообщение M1, дающее команду на выполнение транзакции, может быть повторно передано банком-отправителем. Если имеется сообщение M4 (указывающее на успешную обработку транзакции в коммутационном пункте, который еще находится в рабочем состоянии), но оно было просто потеряно или задержано, то сообщение M4 будет повторно передано банку-отправителю в ответ на повторное сообщение M1.

Однако, если сообщение M4 недоступно, это указывает на то, что данная транзакция либо не была успешно обработана (а это значит, что сообщение M4 никогда не создавалось), или на то, что все коммутаторы в указанном коммутационном пункте, где обрабатывалась транзакция, оказались неработоспособными после создания сообщения M4 (а это значит, что сообщение M4 недоступно для повторной посылки). В этом случае дополнительная обработка повторно переданного сообщения M1 не выполняется. Это помогает обеспечить, чтобы ни одна транзакция не обрабатывалась системой 100 дважды (помогая таким образом избежать ситуации, когда денежная сумма дебетуется дважды со счета банка-отправителя). В этом случае сообщение M4 еще не принято банком-отправителем (несмотря на повтор-

ную передачу сообщения M1), и банковское приложение 102A, 102B указанного банка может сообщить пользователю, что данная транзакция не может быть успешно выполнена. В конце расчетного цикла внутренние записи банка-отправителя, внутренние записи банка-получателя и итоговые записи TS о транзакциях, созданные системой 100, можно проанализировать, чтобы определить результат транзакции и обеспечить, чтобы все средства транзакции были учтены.

В вышеупомянутом варианте осуществления период таймаута банка-отправителя для приема сообщения M4 определяется специалистами в данной области техники как временной период, в рамках которого обоснованно ожидается поступление сообщения M4, с учетом ожидаемого времени обработки каждой компонентой в системе 100, сетевой задержки и т.д. Также следует иметь в виду, что (смотри фиг. 2A) этапы, включающие в себя "направление" транзакционного сообщения на подходящий коммутатор определяется списком приоритетов коммутатора, если коммутатор, который принимает сообщение на этапе 202, действительно является коммутатором, который должен обработать данное сообщение (согласно списку приоритетов), то транзакционное сообщение в действительности не будет направлено на другой коммутатор. Скорее всего, оно будет обработано тем же принимающим коммутатором.

Вышеупомянутые признаки помогают обеспечить возможность непрерывной обработки надежно и эффективно, даже если одна или более компонент системы 100 вышли из строя. В качестве дополнительной проверки, в конце расчетного цикла перед выполнением операционным блоком 114 обработки взаимных расчетов с использованием суммарных записей TS о транзакции, проверяется общая сумма транзакции (то есть, общая сумма перечисляемых денежных средств), поступающая к/от каждого банка, определенная итоговыми записями TS о транзакции в сопоставлении с соответствующими записями транзакционных сумм, поддерживаемыми указанными коммутаторами (коммутатор, который обрабатывает сообщение M1 каждой транзакции будет, например, поддерживать запись транзакционной суммы и банка-отправителя и банка-получателя). Если обработка сообщений и создание итоговой записи TS о транзакции были выполнены корректно, то тогда общие транзакционные суммы, перечисляемые банку/от банка, зафиксированные итоговыми записями TS о транзакции, и записи коммутатора должны точно совпадать. В этом случае известно, что обработка расчетов может выполняться надежно. С другой стороны, если между общими суммами транзакции имеется несоответствие, тогда, как известно, имеет место проблема, связанная с созданием итоговых записей TS о транзакции. Следовательно, эта проблема может быть исследована и решена до обработки взаимных расчетов.

Таким образом, в свете вышеописанных признаков следует иметь в виду, что система 100 помогает обеспечить, чтобы электронные транзакции, инициируемые банками, не отменялись, искажались или дублировались, прежде чем они не будут обработаны в конце расчетного цикла. Это помогает обеспечить учет каждой инициированной транзакции, причем только один раз в конце расчетного цикла. Это обеспечивается, даже если отказала какая-либо компонента системы 100, или если имеет место запланированное техническое обслуживание какой-либо компоненты системы 100, для которого необходимо временное отключение компоненты. В то же время, пока еще имеется по меньшей мере один работоспособный коммутатор системы 100, держатели банковских счетов могут продолжать инициировать новые транзакции и пользоваться обычными услугами.

Управление долговыми обязательствами

В вышеописанной системе нетто-расчеты между банками выполняются на периодической основе. Другими словами, реальные денежные средства перечисляются от банка к банку периодически. Это означает, что итоговые записи TS о транзакции являются обязательствами оплаты при взаимных расчетах. Расчеты в этом варианте осуществления выполняются отдельно от системы 100 через систему, уполномоченную банком-хранителем обыкновенных депозитных вкладов, таким как Bank of England. Указанные системы расчетов известны специалистам в данной области техники. Соответственно, обеспечивается дебетовый лимит в зависимости от депозитов, поддерживаемых для расчетов. Банку категорически не разрешается превышать дебетовый лимит.

При реализации дебетового лимита по множеству пунктов в указанной надежной системе согласно вариантам осуществления, возникает проблема. Она поясняется на фиг. 3A.

В примере, показанном на фиг. 3A, положим, что банк А имеет дебетовый лимит -10000 британских фунтов (602). Так как система 100 распределена по коммутационному пункту 1 (603) (который для простоты называется "пункт 1") и коммутационному пункту 2 (604) (который для простоты называется "пункт 2"), для обеспечения надежности дебетовый лимит 602 распределен поровну между пунктом 1 и пунктом 2. В любом случае коммутаторы в каждом пункте записывают дебетовый лимит для данного пункта. Таким образом, в этом случае дебетовый лимит в пункте 1, связанный с банком А, составляет -50000 британских фунтов, и дебетовый лимит, хранящийся в пункте 2, связанном с банком А, также равен -5000 британских фунтов. Конечно, возможны и другие варианты разделения, например, -2500 британских фунтов для системы, распределенной по четырем пунктам и т.п.

Предположим, что в пункте 1 обрабатывается дебетовая транзакция (транзакция 1). Транзакция 1 является дебетовой транзакцией на -2500 британских фунтов. Это означает, что позиция расчетного риска (SRP), являющаяся профессиональным термином, и что сумма по всем транзакциям, которые уже появились в каждом пункте, для коммутационного пункта 1 составляет -2500 британских фунтов. Следова-

тельно, оставшаяся часть дебетового лимита, доступная для будущих транзакций (для простоты также называемые "доступные дебиты") в пункте 1 составляет -2500 британских фунтов. Так как в пункте 2 транзакций нет, то SRP для пункта 2 равно 0, а доступные дебиты для банка А в пункте 2 все еще равны 5000 британских фунтов. Следовательно, общие доступные дебиты составляют -7500 британских фунтов, а общая SRP составляет 2500 британских фунтов (605).

Положим теперь, что в пункте 2 обрабатывается вторая дебетовая транзакция (транзакция 2), где транзакция 2 выполняется на сумму -300 британских фунтов. Следовательно, доступные дебиты для пункта 2 составляют -4700 британских фунтов, а SRP в пункте 2 равна -300 британских фунтов. SRP в пункте 1 остается равной -2500 фунтов, и доступные дебиты для пункта 1 остаются равными -2500 фунтов. Общий SRP для банка А (то есть, сумма SRP в пункте 1 и SRP в пункте 2) составляет -2800 фунтов, а доступные дебиты (например, сумма доступных дебетов для пунктов 1 и 2) для банка А составляет -7200 фунтов, заметим, что вычисления доступных дебетов на фиг. 3А не показаны.

Положим теперь, что в пункте 1 предпринимается попытка обработки третьей дебетовой транзакции (транзакция 3). Транзакция 3 - это дебет на сумму -2600 фунтов. Это означает, что SRP для пункта 1 (если бы транзакция была обработана) составит -5100 фунтов и, следовательно, превысит дебетовый лимит -5000 фунтов, выделенный банку А в пункте 1. Следовательно, транзакция 3 будет отвергнута.

Однако при условии, что общая SRP для банка А, пока не поступила транзакция 3, составляла только -2800 фунтов, и что общие доступные дебиты для банка А составляют -7200 фунтов, транзакцию 3 следует обработать. Это означает, что обеспечение дополнительной устойчивости к технологическим отказам прекращает выполнение транзакций, которые в ином случае следовало бы разрешить. Варианты осуществления изобретения нацелены на решение этой проблемы, как объясняется ниже.

На фиг. 3В показан пример согласно вариантам осуществления изобретения. В этом примере дебетовый лимит 651 остается равным -10000 фунтов. По аналогии с примером, показанным на фиг. 3А, в примере на фиг. 3В дебетовый лимит 651 поровну распределен между двумя указанными пунктами. Другими словами, пункт имеет дебетовый лимит -5000 фунтов, и пункт 2 имеет дебетовый лимит -5000 фунтов.

В подобном примере (смотри фиг. 3А) транзакция 1 поступает в пункт 1, имея дебет -5000 фунтов. Соответственно, SRP в пункте 1 для банка А составляет -2500 фунтов, а SRP для банка А в пункте 2 равна 0, и это значит, что общая SRP для банка А в обоих пунктах составляет 2500 фунтов. Кроме того, доступные дебиты пункта 1 сократились до -2500 фунтов, а доступные дебиты пункта 2 остались на уровне -5000 фунтов. Таким образом, общие доступные дебиты составляют -7500 фунтов. Заметим, что вычисления доступных дебетов на фиг. 3В не показаны.

Затем транзакция 2 поступает в пункт 2 с дебетом -300 фунтов. Соответственно, SRP банка А в пункте 1 остается равной -2500 фунтов, SRP банка А в пункте 2 составляет -300 фунтов, и общая SRP составляет 2800 фунтов. Более того, доступные дебиты пункта 2 сократились до -4700 фунтов, а имеющиеся дебиты пункта 1 остались на уровне -2500 фунтов. Следовательно, общие доступные дебиты для банка А составляют -7200 фунтов.

Как пояснялось со ссылками на фиг. 3А, если транзакция 3 (дебет 2600 фунтов) поступила на коммутационный пункт 1, она будет отвергнута, так как SRP пункта 1 будет превышать дебетовый лимит 5000 фунтов, выделенных для пункта 1. Во избежание этой проблемы в системе периодически выполняется цикл настройки. В частности, цикл 656 настройки периодически выполняется коммутаторами в пунктах 1 и 2. Цикл настройки содержит вычисление значения настройки путем суммирования текущих величин SRP в каждом пункте. То есть, текущая величина SRP1, равная -2500 фунтов, и текущая величина SRP2, равная 300 фунтов, дает в сумме -2800 фунтов. Затем эту сумму усредняют по всем пунктам. В этом случае среднее по двум пунктам составит $-2800/2 = -1400$ фунтов. Конечно, могут быть использованы и другие виды усреднения, такие как среднестатистическое или т.п. Целью определения средней SRP по всем пунктам в системе (в данном случае это пункты 1 и 2) является достижение эффективного баланса обязательств банка между всеми коммутационными пунктами. Это означает, что, если один коммутационный пункт принимает транзакции на очень большие суммы по сравнению с другими коммутационными пунктами, то тогда дебетовый лимит системы 100 не будет превышен (если дебетовый лимит для данного банка по всем коммутаторам не превышен).

Для вычисления значения настройки `adjustment_site` для применения в конкретном коммутационном пункте используют следующее уравнение: $\text{adjustment_site} = \text{averaged_SRP} - \text{SRP}$, где `adjustment_site` - значение настройки, применяемой к конкретному коммутационному пункту, `averaged_SRP` - среднее значение SRP по коммутационным пунктам в системе и `SRP` - SRP в конкретном коммутационном пункте.

Из этого уравнения вытекает, что настройка в пункте 1 $= -2800/2 - (-2500) = +1100$, а настройка в пункте 2 $= -2800/2 - (-300) = -1100$.

Таким образом, SRP конкретного пункта настраивают, применяя указанное значение настройки. Это значение настройки не влияет на общую SRP для банка (не разрешая указанному банку превысить общий дебетовый лимит). Вместо этого используют настроенное значение SRP для распространения обязательств банка по всем коммутационным пунктам.

В формализованном виде настроенная $SRP = \text{значение } SRP + \text{adjustment_site}$.

Таким образом, в случае, показанном на фиг. 3B, после выполнения цикла настройки настроенная SRP для пункта 1 $= -2500 + 1100 = -1400$ фунтов, а настроенная SRP для пункта 2 $= -300 + 1100 = -1400$ фунтов. Другими словами, действующее значение SRP в каждом пункте одинаково.

Если сумма настроенных SRP и новой транзакции не превышает дебетовый лимит, на данном пункте, то новая транзакция одобряется. Однако, если сумма настроенного SRP и новой транзакции превышает дебетовый лимит на данном пункте, то новая транзакция отвергается.

Обратимся теперь к фиг. 3B, где использование настроенной SRP пункта 1 (-1400 фунтов) и суммы транзакции 3, новой транзакции (-2600 фунтов), дает в итоге 4000 фунтов. Это не превышает дебетовый лимит пункта 1, равный -5000 фунтов, и, следовательно, она будет разрешена системой 100. Таким путем значения настройки для каждого пункта используют для уменьшения вероятности того, что действительная сумма транзакции (то есть, сумма транзакции, разрешенная исходя из общего дебетового лимита) будет отвергнута на основе имеющихся дебетов в отдельных коммутационных пунктах.

В вариантах осуществления цикл настройки (для вычисления нового значения настройки) выполняется на периодической основе. Этот период может быть временным, например, каждые 20 или 30 с или может наступить после того как пункт обработал конкретное количество транзакций, или даже когда SRP в пункте превысила заданную величину.

В случае отказа пункта дебетовый лимит на каждом из оставшихся пунктов может быть увеличен таким образом, чтобы общее значение дебетового лимита было допустимым. Так, например, если пункт 2 вышел из строя, то тогда общий дебетовый лимит, равный 10000 фунтов, мог бы быть выделен пункт 1, с тем чтобы могла продолжаться обработка транзакций на общую сумму вплоть до значения общего дебетового лимита.

Заметим, что действительное значение SRP для каждого пункта (в настоящем примере это -5100 для пункта 1 и -300 для пункта 2 после выполнения всех транзакций по фиг. 3B) очень важно, поскольку значение SRP в пункте равно денежной сумме, owed банку (если SRP положительный) или являющейся задолженностью перед банком (если SRP отрицательный), после транзакций в этом пункте. Таким образом, действующее значение SRP в каждом пункте должно оставаться неизменным, с тем чтобы его можно было использовать для обеспечения корректного выполнения расчетов. Таким образом, в соответствующем пункте продолжает храниться действующее значение SRP для каждого пункта в любой заданный момент времени, притом, что это настроенное значение SRP , которое используют для того, чтобы общий дебетовый лимит не был превышен. Заметим, что для точного (насколько это возможно) поддержания SRP в каждом пункте, оно будет обновлено для банка-отправителя после успешной обработки сообщения M1, и обновлено банком-получателем после успешной обработки сообщения M3.

Применение шлюзов

Как обсуждалось выше, шлюз 104A, 104B каждого банка помогает управлять пересылкой сообщений между банком и коммутаторами SW1, SW2, SW3 и SW4. Каждый шлюз 104A, 104B выполняет ряд функций по поручению банка и коммутаторов для коррекции ошибок обработки в банках и коммутаторах. Заметим, что использование шлюзов во всей системе является опциональным. Однако, если шлюз опционально не используется, банковское приложение 102A, 102B должно быть способно выполнять структурированную проверку (смотри ниже на конкретном этапе 906 по фиг. 4), подписывать сообщения, выполнять передачу и проверку (опять же смотри ниже на конкретных этапах 910, 912 и 914 по фиг. 4), выполнять маршрутизацию сообщений и добавлять информацию о повторении сообщения M1 (когда это требуется - смотри выше) с использованием подобных функциональных возможностей.

Одна из функций шлюза связана с ситуацией, когда банк (по поручению держателя счета в этом банке) требует обработку транзакции системой 100. Каждый банк обычно использует разные системы собственности для управления счетами и для разрешения держателям счетов поручать выполнение транзакций и т.п. Однако по соображениям эффективности транзакционные сообщения, обрабатываемые коммутаторами, должны соответствовать структуре определенного стандарта (или формату, такому как обсужденный выше стандарт ISO20022). Таким образом, любое транзакционное сообщение, созданное держателем счета в банке, должно иметь форму транзакционного сообщения стандартного типа, чтобы оно могло быть принято коммутаторами. Кроме того, по соображениям безопасности необходимо, чтобы любое транзакционное сообщение, передаваемое на коммутаторы и обратно имело, цифровую подпись и было проверено (с использованием цифровой подписи) с тем, чтобы гарантировать легитимность реального источника сообщения.

Проверка структуры транзакционных сообщений выполняется шлюзом. Кроме того, подпись сообщений, подлежащих передаче от банка к коммутаторам (эти сообщения последовательно проверяются принимающим коммутатором), и проверка сообщений, переданных от коммутаторов в банк (эти сообщения подписаны ранее передающим коммутатором) реализуется с использованием шлюза. То есть, шлюз банка обеспечивает интерфейс передачи между системой собственности данного банка и коммутаторами, чтобы гарантировать наличие цифровой подписи транзакционных сообщений, их проверку и соответствие требуемой структуре стандарта. Для структуры транзакционных сообщений цифровой подписи и проверки можно использовать любой подходящий стандарт. Такие стандарты известны специали-

стам в данной области техники. В примерном способе цифровой подписи и проверки используется хэш данных сообщения (хэшированных с использованием SHA1) с последующим шифрованием (с использованием RSA шифрования) для передачи. В вариантах осуществления транзакционные сообщения, подлежащие обработке шлюзами 104A, 104B, ставят в очередь соответственно в очереди 106A, 106B. Если шлюзы 104A и 104B были предусмотрены, то банковскому приложению 102A, 102B потребуется обеспечить эти функциональные возможности.

Шлюз также направляет сообщение в коммутационные пункты, добавляя маршрутную информацию в заголовок каждого сообщения, который затем считывается блоками 106A, 106B, 108A, 108B очереди сообщений для направления сообщения в подходящий коммутационный пункт. В примере по фиг. 1 очередь 108A сообщений можно обозначить как MQ1, а очередь 108B сообщений можно обозначить как MQ2. Таким образом, обозначение MQ1 или MQ2 добавляется к заголовку каждого сообщения, подлежащего передаче в коммутационный пункт указанным шлюзом, так что очередь 106A, 106B сообщений может направить указанное сообщение в соответствующий коммутационный пункт. Также предусмотрено, что один банк может иметь множество блоков 106A, 106B очередей сообщений на стороне банка (например, банк 1 может иметь множество очередей 106A1, 106A2, ..., 106An сообщений - не показаны), и что маршрутная информация, добавленная шлюзом, может включать в себя информацию, идентифицирующую, на какой из блоков очереди сообщений на стороне банка должно быть направлено указанное сообщение. Это помогает обеспечить равномерную загрузку данных в системе 100, что является преимуществом.

В качестве примерного варианта осуществления со ссылками на процесс, показанный на фиг. 4, описывается работа шлюза 104A для банка 1 и работа коммутатора, когда сообщение передается от банка 1 на указанный коммутатор. Процесс начинается на этапе 900. На этапе 902 шлюз 104A принимает транзакционное сообщение от банковского приложения 102A банка 1, которое должно быть передано на один из коммутаторов для обработки. На этапе 904 шлюз 104A анализирует структуру транзакционного сообщения. В частности, шлюз 104A обеспечивает стандартную структуру этого сообщения, необходимую для обработки указанными коммутаторами. Это включает в себя обеспечение ввода в сообщение всей необходимой информации. Например, если сообщение содержит поручение на выплату денежных средств со счета банка 1 на счет банка 2, то гарантируется, что сумма платежа и валюта, код типа банка, подробности счета банка 1, а также код типа банка 2 и подробности счета банка 2 будут присутствовать в этом сообщении, причем в правильном формате (поскольку эта информация необходима для завершения транзакции). Это также включает в себя проверку того, содержит ли данное сообщение уникальный идентификатор транзакции, и содержит ли это сообщение подходящую маршрутную информацию.

На этапе 906, если определено, что структура сообщения соответствует указанному стандарту, то процесс переходит к этапу 910. С другой стороны, если определено, что структура сообщения не соответствует указанному стандарту (например, если какая-либо существенная информация выпала из указанного сообщения), то процесс переходит к этапу 908. На этапе 908 сообщение возвращают в банк 1 вместе с сообщением об ошибке. Сообщение об ошибке информирует банк 1 о том, что транзакционное сообщение не соответствует стандарту сообщения, и что его необходимо повторно послать с исправленной ошибкой (например, с добавленной пропавшей информацией). Затем процесс переходит к этапу 922.

На этапе 910 под транзакционным сообщением ставится цифровая подпись банка 1. Эта цифровая подпись позволяет принимающему коммутатору гарантировать аутентичность данного сообщения (то есть, что это сообщение исходит от банка 1). Затем на этапе 912 по каналу обмена данными между банком 1 и подходящим коммутационным пунктом (как это определено маршрутной информацией) передается запрос транзакции. Заметим, что шлюз реализован на основе схем, выполняющих программное обеспечение, хранящееся на запоминающем носителе (не показан) в банке 1. Преимуществом является то, что эта конфигурация позволяет выполнить проверку и передачу транзакционного сообщения, полностью обрабатываемого схемами шлюза, что сокращает нагрузку, связанную с обработкой, как в системе собственности банка 1, так и в коммутационном пункте. Это происходит потому, что сообщения, не содержащее правильную информацию, в коммутационный пункт не направляются. Более того, предусмотрено, что, если шлюз в системе не предусмотрен (как может быть в данном случае) банковское приложение обеспечивает цифровую подпись под сообщениями, которые содержат всю необходимую информацию и сформированы в подходящем формате.

Как только коммутатор в коммутационном пункте примет указанное сообщение, выполняется его проверка с использованием упомянутой цифровой подписи. Это выполняется на этапе 914. Как было описано выше, этапы цифровой подписи и проверки (этапы 910 и 914 соответственно) помогают гарантировать, что любое транзакционное сообщение, полученное данным коммутатором, в действительности поступило от одного из банков, подписавшихся использовать систему 100. Верификация подписи может быть выполнена на коммутаторе, который принял указанное сообщение первым, или, в качестве альтернативы, коммутатором, на который было направлено указанное сообщение (если таковой имеется).

На этапе 916, если цифровая подпись была успешно проверена процесс переходит к этапу 920, на котором транзакционное сообщение обрабатывается подходящим коммутатором (как было описано). Затем процесс заканчивается на этапе 922. С другой стороны, если проверка цифровой подписи дала от-

рицательный результат, то полагают, что транзакционное сообщение возможно содержит ошибку или т.п., связанную с ней. Таким образом, процесс переходит к этапу 918, на котором указанное транзакционное сообщение возвращают банку-отправителю. В этом случае коммутатором, который выполнил проверку и возврат транзакционного сообщения (или его части), создает информацию, указывающую что в процессе проверки, имела место ошибка.

Преимуществом является то, что банк-отправитель уведомляется о проблеме, связанной с указанным транзакционным сообщением, и тогда банк-отправитель сможет ее исследовать. Затем процесс на этапе 922 заканчивается. Верификация цифровой подписи всех входящих транзакционных сообщений также помогает предотвратить обработку поддельных или фальсифицированных сообщений (например, от авторизованной стороны, пытающейся поручить выполнение транзакции).

Таким образом, использование шлюза помогает обеспечить правильную стандартную структуру всех транзакционных сообщений, получаемых коммутаторами для обработки. Следовательно, эти коммутаторы должны быть сконфигурированы для работы только с одной стандартной структурой (а не с множеством разных структур сообщений, создаваемых разными банками), что повышает эффективность системы 100. Кроме того, поскольку сообщения действительно передаются по сети от банков в систему 100, если они соответствуют правильной структуре сообщения, сетевая пропускная способность не используется впустую на посылку сообщений на коммутаторы, которые не имеют правильную структуру (и которые, следовательно, будут отвергнуты). Дополнительным преимуществом является то, что процесс формирования цифровой подписи и верификации, реализованный шлюзом и коммутатором, помогает повысить безопасность системы.

Как уже упоминалось, в вариантах осуществления каждый шлюз 104A, 104B реализован в виде схем, выполняющих программное обеспечение, хранящееся на запоминающем носителе (не показан), имеющемся в месте расположения соответствующего банка. Это программное обеспечение функционально отделено от программного обеспечения, реализующего внутренние функции в каждом банке (например, система, имеющаяся в каждом банке, для управления счетами и банковское приложение, позволяющее держателям счетов давать поручения на выполнение транзакций и т.п.) и от любого программного обеспечения, которое реализует внутренние функции в коммутационных пунктах (например, любое программное обеспечение, реализуемое коммутаторами SW1, SW2, SW3, SW4, блоками 108A, 108B очереди сообщений и т.д.). Преимуществом является то, что это позволяет реализовать выгоды от функционирования шлюза (повышенная эффективность и безопасность, как обсуждалось выше) без увеличения нагрузки на ядерные процессоры в банках или коммутационных пунктах.

Вторая функция шлюза 104A, 104B относится к непрерывному контролю за тем, какие банки и какие коммутационные пункты доступны для посылки и приема сообщений, и для выравнивания загрузки данных в системе 100.

Если сообщение не доставлено в коммутационный пункт через соответствующие блоки 106A, 106B, 108A, 108B очереди сообщений в течение заданного временного периода, то соответствующий блок очереди сообщений добавляет в заголовок сообщения информацию, указывающую на то, что заданное время истекло, и возвращает это сообщение на шлюз. Эта ситуация может возникнуть потому что, например, блок 108A, 108B очереди сообщений в коммутационном пункте не смог зафиксировать сообщение от соответствующего блока 106A, 106B очереди сообщений в течение заданного временного периода, или потому, что коммутатор на стороне коммутаторов не смог зафиксировать данное сообщение от соответствующего блока 108A, 108B очереди сообщений в течение указанного заданного временного периода. Может иметь место множество различных причин этого, в том числе отказ блока 108A, 108B очереди сообщений в коммутационном пункте, отказ коммутаторов или отказ сети. В этом случае шлюз передает такое сообщение повторно в альтернативный пункт. Так, например, если ни SW1, ни SW2 в коммутационном пункте 1 не зафиксировал сообщение в течение заданного временного периода, то шлюз банка-поручителя после возвращения указанного сообщения перенаправит это сообщение на коммутационный пункт 2. Однако здесь возникает проблема, состоящая в том, что, если, например, оба коммутатора в конкретном коммутационном пункте готовы работать в течение увеличенного периода времени, этот способ перенаправления сообщений потребует интенсивного использования процессора и бесполезного использования пропускной способности сети. Причина этого заключается в том, что сообщения будут непрерывно посылаться в неактивный пункт лишь только для того, чтобы получать ответы с сообщением об ошибке.

Аналогичным образом, внутренние системы в одном или более банках также могут временно оказаться недоступными для приема сообщений. Например, в примере по фиг. 1 банк 2 может оказаться недоступным для приема сообщения M2, информирующего банк 2 об ожидании выплаты денежных средств от банка 1, в случае, когда транзакционное сообщение M1 выдано держателем счета в банке 1 для выплаты денежных средств держателю счета в банке 2. В этом случае сообщение M3 от банка 2 не принимается после передачи сообщения M2 на банковское приложение 102B, и, следовательно, данная транзакция не может быть завершена. Однако, если банк 2 доступен в течение удлиненного периода времени, то данный способ, заключающийся в продолжении посылки сообщений M2 в банк 2, окажется проблемным, поскольку он связан с весьма интенсивным использованием процессора и производи-

тельным использованием пропускной способности сети. Это происходит потому, что сообщения непрерывно посылаются в неактивный банк.

Для решения этих проблем шлюз 104A, 104B каждого банка сконфигурирован для отслеживания того, какие другие банки и коммутационные пункты доступны для приема сообщений. Это определено в виде кэша доступности.

Что касается непрерывного контроля за коммутационными пунктами, то шлюз делает это, посылая тестовое сообщение на коммутационный пункт по сети через блок 108A, 108B очереди сообщений этого коммутационного пункта и прослушивая эхосигнал на это сообщение, поступающий от коммутационного пункта. Шлюз выполняет это, как правило, если он не получил сообщение от конкретного коммутационного пункта в течение определенного периода времени. Эхосигнал (точнее, "дистанционный эхоконтроль") представляет собой копию тестового сообщения, посланного в коммутационный пункт, которая передается из коммутационного пункта обратно на шлюз. Эхосигнал создается одним из коммутаторов в коммутационных пунктах в соответствии с командой, содержащейся в тестовом сообщении. Если в ответ на посылку тестового сообщения в конкретный коммутационный пункт, шлюз принимает эхосигнал этого сообщения в течение заданного временного периода (например, период времени, равный периоду таймаута для приема сообщения M4), то шлюз узнает, что коммутационный пункт еще доступен для приема сообщений через блок 108A, 108B очереди сообщений. С другой стороны, если эхосигнал не получен, то шлюз определяет, что коммутационный пункт недоступен для приема сообщений (возможно из-за отказа) через блок 108A, 108B очереди сообщений, и тем самым предотвращает передачу дополнительных сообщений в указанный коммутационный пункт по этому маршруту.

В случае, когда шлюз передает тестовое сообщение в коммутационный пункт вышеописанным путем и по истечении заданного периода времени не принимает обратно эхосигнал, шлюз определяет, что связь с данным коммутационным пунктом через блок 108A, 108B очереди сообщений недоступна. Это происходит потому, что, если бы этот маршрут был бы доступен, то тогда тестовое сообщение в конце концов поступило бы на коммутатор данного коммутационного пункта, и этот коммутатор создал бы копию тестового сообщения, возвращаемого обратно на шлюз в виде эхосигнала. В случае, когда эхосигнал не принят, шлюз больше не будет направлять сообщение в данный коммутационный пункт через блок 108A, 108B очереди сообщений. Другими словами, этот коммутационный пункт определяют как недоступный. Таким образом, шлюз управляет последующим распределением сообщений, как только один из коммутационных пунктов будет определен как недоступный.

Шлюз также определяет, что коммутационный пункт недоступен, если сообщение M1 вернулось на шлюз из коммутационного пункта, поскольку оно не было зафиксировано коммутатором в указанном коммутационном пункте в течение заданного временного периода (смотри выше). Опять же, в этом случае шлюз больше не будет направлять сообщения на указанный коммутационный пункт через блок 108A, 108B очереди сообщений в этом пункте.

Для ясности заметим, что, если коммутационный пункт определен шлюзом как недоступный, то это не обязательно означает, что коммутаторы в указанном коммутационном пункте недоступны. Скорее, это свидетельствует о том, что сообщения не могут передаваться на коммутаторы в указанном коммутационном пункте через блок 108A, 108B в указанном коммутационном пункте. Например, это может быть случай, когда произошел сетевой отказ, который препятствует передаче сообщений на коммутаторы в конкретном коммутационном пункте через блок 108A, 108B, но при этом коммутаторы в указанном коммутационном пункте все еще полностью работоспособны. В этом случае сообщение все еще можно направлять на указанные коммутаторы через другой коммутационный пункт и линию обмена данными между пунктами (например, в соответствии с функцией хэширования).

После определения коммутационного пункта как недоступного (либо из-за отсутствия эхосигналов, принимаемых на шлюзе, или из-за возврата сообщения M1 на шлюз) тестовые сообщения периодически посылают на указанный коммутационный пункт через блок 108A, 108B очереди сообщений, с тем чтобы определить, стал ли он снова доступным. Если коммутационный пункт остается недоступным, то никакие эхосигналы не принимаются, и, следовательно, сообщения не будут посылаться на указанный коммутационный пункт. С другой стороны, если доступность коммутационного пункта восстановлена (например, после завершения ремонта или технического обслуживания), то, когда следующее тестовое сообщение будет послано в указанный коммутационный пункт, от него будет получен эхосигнал. В этом случае шлюз определяет, что коммутационный пункт снова стал доступен, и разрешает передачу сообщений в указанный коммутационный пункт через блок 108A, 108B очереди сообщений. Шлюз также установит, что коммутационный пункт, определенный как недоступный, вновь доступен в том случае, когда транзакционное сообщение (например, сообщение M2) получено из того же коммутационного пункта (поскольку указанное сообщение не может быть получено, если коммутационный пункт все еще недоступен). Что касается непрерывного контроля за другими банками, то шлюз банка руководствуется информацией, сформированной коммутаторами, и/или информацией, которую можно получить исходя из факта приема сообщения M4 для транзакций для конкретных банков.

В частности, каждый из коммутаторов может периодически посылать тестовое сообщение в каждый из банков, используя систему 100, и ждать получение эхосигнала от каждого из этих банков.

Если эхосигнал от конкретного банка в течение заданного временного периода на конкретном коммутаторе не получен, то определяют, что данный банк недоступен для приема сообщений от указанного конкретного коммутатора (возможно из-за отказа сетевого соединения между указанным коммутатором и банком). В этом случае сообщения M2, передаваемые в указанный банк из указанного коммутатора, будут перенаправлены по альтернативному маршруту (например, через другой коммутатор в коммутационном пункте, или даже через один из коммутаторов в альтернативном коммутационном пункте через соединение для обмена данными между пунктами). Заметим, что в этом случае сам коммутатор находится в рабочем состоянии и, следовательно, может еще принимать сообщения согласно функции хэширования и обрабатывать эти сообщения. Единственная проблема - это действительная передача сообщений M2 в рассматриваемый банк, которая решается с помощью описанного здесь перенаправления сообщений. Заметим, что это не применяется к сообщению M4, поскольку, как уже было сказано, сообщение M4 необходимо передать обратно в банк-отправитель, выдавший запрос транзакции через тот же коммутатор, на котором было первоначально получено сообщение M1 (а не коммутатор, определенный функцией хэширования). В случае, когда соединение между исходным коммутатором и банком-отправителем нарушено, сообщение M4 нельзя будет передать на банк-отправитель через этот коммутатор. Следовательно, система должна ждать повторную передачу сообщения M1 от банка-отправителя, которое в случае, когда соединение между исходным коммутатором и банком-отправителем остается нарушенным, будет направлено на альтернативный принимающий коммутатор, который имеет соединение с банком отправителем через подходящие очереди 106A, 106B, 108A, 108B сообщений. Затем это сообщение M1 о повторе будет направлено на обрабатывающий коммутатор (согласно функции хэширования), что приведет к повторной передаче сообщения M4 с последующей передачей этого сообщения M4 обратно в банк-отправитель через альтернативный принимающий коммутатор.

В случае, когда все коммутаторы не в состоянии получить ответ от конкретного банка, определяют, что данный банк недоступен в целом (то есть, он недоступен для приема сообщений с любого коммутатора), и информация, указывающая на это, возвращается на шлюз всех других банков.

Вдобавок, в случае, например, планового технического обслуживания внутренних банковских систем каждый банк, использующий систему 100, имеет возможность отсоединиться от системы 100. В этом случае банк намеренно отсоединяется (или, другими словами, он отключается) от указанных коммутаторов, и один или более из этих коммутаторов будут передавать информацию на шлюз всех других банков, указывающую, что данный банк, который отключился, недоступен.

Кроме того, в случае, когда банк недоступен (из-за отсутствия ответных эхосигналов на тестовые сообщения от всех коммутаторов или из-за его отключения от системы), коммутатор, который получает сообщение M1, содержащее поручение осуществить транзакцию средств на недоступный банк, может создать сообщение M4 и передать это сообщение обратно в банк-отправитель. Это сообщение M4 отличается от обычного сообщения M4, указывающего на успешную транзакцию, поскольку оно содержит информацию, уведомляющую банк-отправитель о том, что банк-получатель недоступен и, следовательно, транзакция не была успешно обработана. В этом случае дебетовый счет в банке-отправителе может быть перекредитован. В частности, сообщение M4 будет содержать данные, указывающие причину отказа выполнения транзакции (например, код причины, указывающий сетевой отказ в банке-получателе, что указывается отсутствием принимаемых ответных эхосигналов, или код причины, указывающий, что банк-получатель отключен от системы). Шлюз в банке-отправителе, следовательно, сможет использовать это сообщение M4, чтобы определить, что банк-получатель в данный момент недоступен. Заметим, что коммутаторы узнают о недоступности банка по отсутствию принимаемых эхосигналов или из-за отключения банка от коммутаторов, как обсуждалось ранее. Хотя определение коммутаторами недоступности банка посредством использования эхосигналов или через процедуру отключения банка, будет транслироваться на каждый из других банков, использование сообщения M4 для указания отказа в запросе транзакции для конкретного банка обеспечивает дополнительный способ для других банков узнать о недоступности указанного банка (что может оказаться преимуществом, например, если имеет место задержка стандартной трансляции этой информации коммутаторами).

Шлюз каждого банка поддерживает запись о доступности всех других банков (например: банк А - доступен, банк В - доступен, банк С - недоступен) согласно информации, транслируемой коммутаторами (эта информация основана на использовании ответов и/или процедуры отключения банка) и/или приеме сообщений M4, указывающих, что конкретный банк недоступен. Эта запись составляет часть кэша доступности, поддерживаемого в каждом банке (кэш доступности также включает в себя информацию, касающуюся того, какие коммутационные пункты доступны). Благодаря этой записи сокращается непроизводительное использование пропускной способности в попытках послать сообщения в недоступные банки. Например, когда клиент банка А запрашивает новую транзакцию средств, предназначенных для банка, определенного как недоступный (например, банк С, как в этом случае), тогда шлюз отвергнет эту транзакцию, так что сообщение M1 никогда не будет послано, и держатель счета этого банка будет уведомлен банковским приложением банка А о том, что банк-получатель (банк С) в данный момент недоступен. Таким образом, сокращается непроизводительное использование пропускной способности в результате отправки транзакционных сообщений через систему 100 в недоступный банк.

Когда определено, что банк недоступен посредством использования эхосигналов (как обсуждалось выше), можно периодически посылать тестовые сообщения на этот недоступный банк с тем, чтобы определить, не стал ли он вновь доступным. Если банк остается недоступным, эхосигнал не будет получен и, следовательно, сообщения будут продолжать посылаться, но только не в указанный банк. С другой стороны, если доступность банка восстановлена (например, после повторного восстановления отказавшего сетевого соединения), то при посылке на этот банк следующего тестового сообщения, будет получен эхосигнал. В этом случае шлюз определяет, что банк снова доступен и сразу разрешает передачу на этот банк сообщений.

Если определено, что банк недоступен из-за его отключения от коммутаторов, то банк остается недоступным, пока он снова не восстановит соединение с коммутаторами (или, другими словами, вновь подключится). Как только банк снова соединится с коммутаторами, один или несколько коммутаторов передадут информацию на шлюз всех других банков, указывающую, что данный банк стал снова доступным.

Обычно текстовые сообщения могут посылаться указанными коммутаторами периодически в каждый из банков с достаточно малым временным периодом с тем, чтобы попытаться минимизировать количество транзакционных сообщений, посланных в банк, являющийся недоступным, что позволяет экономно использовать пропускную способность. В то же время указанный временной период не должен быть слишком маленьким, поскольку для посылки тестовых сообщений с большой частотой потребуется использовать слишком большую пропускную способность сети, что приведет к снижению эффективности сети (практически сводя на нет изначальный эффект экономного использования пропускной способности при передаче тестовых сообщений и эхосигналов). В качестве примера, временной период между тестовыми сообщениями, посылаемыми в конкретный банк, может составлять примерно от 5 до 30 с. Конечно, эти цифры могут меняться, как очевидно специалистам в данной области техники.

В одном варианте осуществления шлюз посылает множество тестовых сообщений до определения того, что коммутационный пункт недоступен. Подобным же образом, каждый коммутатор посылает множество тестовых сообщений до определения того, что конкретный банк недоступен. Например, для конфигурации, в которой временной период между передачами тестовых сообщений составляет 30 секунд, шлюз или коммутатор может послать первое тестовое сообщение и прослушать эхосигнал 30 секундами позднее; если ответ не был получен, то шлюз или коммутатор пошлет второе тестовое сообщение и прослушает эхосигнал 30 секундами позднее; если эхосигнал еще не был принят, то, коммутационный пункт (в данном случае, шлюз, посылающий тестовые сообщения) или банк (в данном случае, коммутаторы, посылающие тестовые сообщения), на который были переданы тестовые сообщения, определяется как недоступный, и передача транзакционных сообщений на этот коммутационный пункт или банк приостанавливается. Посылка множества тестовых сообщений до определения того, является ли коммутационный пункт или банк доступным, помогает обеспечить непрерывность передачи транзакционных сообщений на функционирующий коммутационный пункт или банк в том случае, когда принимаемый эхосигнал от коммутационного пункта или банка испытывает, например, сетевую задержку или т.п.

Преимуществом является то, что из-за наличия кэша доступности, поддерживаемого шлюзом каждого банка, транзакционные сообщения не посылают в банки или коммутационные пункты, которые определены как недоступные, и, следовательно, сокращается количество попыток передач сообщений на коммутационные пункты или банки, которые не могут принять сообщение. Наряду с уменьшением риска появления искажений или потери сообщений это также означает, что ресурсы обработки и пропускная способность сети не тратятся впустую в связи с сообщениями, которые невозможно было доставить. Таким образом, повышается производительность обработки и эффективность сети.

Вдобавок, заметим, что часто имеется заданное ограничение, накладываемое банками на время, которое затрачивается на транзакцию. То есть, время между посылкой исходного сообщения M1 и приема сообщения M4 банком не должно превышать заданный временной лимит. Может быть установлен любой подходящий лимит времени, хотя, как правило, его устанавливают примерно от 5 до 15 секунд. Таким образом важно обеспечить передачу транзакционных сообщений через систему насколько возможно эффективно, чтобы помочь обеспечить завершение транзакции в пределах указанного лимита времени. Преимуществом является то, что из-за использования кэша доступности в каждом банке время не тратится впустую на попытки посылки транзакционных сообщений на коммутаторы через коммутационные пункты, являющиеся недоступными (что приводит к возвращению сообщения и необходимости изменения маршрута, на что впустую тратится время). Аналогичным образом, поскольку каждый коммутатор поддерживает запись о маршрутах к каждому банку, являющемуся доступным (в соответствии с посланными тестовыми сообщениями и ответами, полученными от каждого банка), время не тратится впустую на попытки посылки транзакционных сообщений в банки через недоступные маршруты. Также, поскольку каждый коммутатор опрашивает другие коммутаторы по поводу их доступности, время не тратится впустую на попытки посылки транзакционных сообщений на недоступные коммутаторы. Таким образом, все эти функции помогают обеспечить эффективную передачу транзакционных сообщений по системе 100 и обеспечить возможность завершения транзакции в пределах лимита времени, определенного бан-

ками.

Как дополнительное преимущество, использование шлюза для непрерывного контроля доступности коммутационных пунктов и банков и для выдачи команд разгружает банковские приложения 102A, 102B. Следовательно, повышается общая эффективность обработки в банках.

На фиг. 5 показан компьютер 700 для использования с системой 100. В варианте осуществления функции, выполняемые указанным элементом системы 100 (то есть, банковские приложения 102A, 102B, шлюзы 104A, 104B, блоки 106A, 106B, 108A, 108B, 112A, 112B, коммутаторы SW1, SW2, SW3, SW4, базы 110A, 110B данных и операционный блок 114) реализованы одним или несколькими указанными компьютерами 700. Эти компьютеры также могут являться серверами (которые, в свою очередь, могут представлять собой физические серверы или виртуальные серверы). Компьютер 700 управляется центральным процессором (CPU) 702, причем CPU 702 сконфигурирован для обработки команд, хранящихся в памяти 704. Обмен данными с компьютером 700 происходит через сетевой интерфейс 706. Компьютер 700 также содержит запоминающий носитель 708 (такой как накопитель на жестких дисках, твердотельная память или накопитель на ленте) для хранения данных.

Хотя система 100 была описана со ссылками на обработку финансовых транзакционных сообщений, следует иметь в виду, что систему можно использовать для обработки и хранения единиц данных любого вида, управление которыми должно быть организовано так, чтобы во время любой дополнительной обработки каждая единица данных определенно учитывалась, причем только один раз. В этом случае, независимо от типа используемой единицы данных система 100 поможет избежать удаления, искажения или дублирования единицы данных.

Например, систему 100 можно использовать для обработки или хранения данных, созданных во время научных или инженерных экспериментов до анализа этих данных. В этом случае каждая единица данных может представлять собой, например, отдельное экспериментальное измерение. Такие единицы данных часто очень трудно или дорого получить, и поэтому важно избежать удаления или искажения таких данных. Кроме того, для последующего анализа важно, чтобы данные не дублировались, так как это может привести к некорректным выводам, получаемым из их анализа. Таким образом, использование системы 100 выгодно для организации и управления данными этого типа.

Хотя вышеизложенное было описано со ссылками на банк, настоящее изобретение также относится к любому финансовому институту, который выполняет денежные транзакции, такому как компания, работающая с кредитными картами или т.п.

Заметим, что конкретное преимущество вышеописанной системы 100 состоит в том, что поскольку компоненты системы скомпонованы таким образом, чтобы обеспечить ее надежность (это означает, что система может продолжать надежно обрабатывать транзакции, даже если некоторые ее компоненты оказались в нерабочем состоянии), тот факт, что, сразу после приема транзакционного сообщения коммутатором (в порядке, определенном функцией хэширования), оно запоминается и обрабатывается в одном пункте перед пересылкой в операционный блок 114, а это значит, что уменьшается задержка в системе (из-за пересылки транзакционного сообщения между разными компонентами в разных пунктах). Таким образом, задержка системы остается небольшой, даже если надежность системы возросла.

Очевидно, что, хотя на фиг. 1 показано два банка, два коммутационного пункта и два коммутатора для каждого из них (для облегчения объяснения), варианты осуществления изобретения этим не ограничены. В действительности может быть большое количество банков, сконфигурированных для использования системы 100, где каждый из них имеет конфигурацию, показанную на фиг. 1 для банка 1 и банка 2 (то есть, содержит банковское приложение, шлюз и блок очереди сообщений). Также возможно использование более двух коммутационных пунктов, сконфигурированных таким же образом, как коммутационные пункты 1 и 2 (причем каждый коммутационный пункт имеет возможность обмена данными с другими коммутационными пунктами и каждым из банков). Кроме того, каждый коммутационный пункт может содержать более двух коммутаторов, каждый из которых сконфигурирован для хранения транзакционных сообщений в одной и той же базе данных, чтобы иметь возможность обработки сообщений, связанных с одной и той же транзакцией, используя любой из коммутаторов в конкретном пункте. Специалистам в данной области техники очевидно, каким образом система 100, описанная со ссылками на фиг. 1, может быть расширена с тем, чтобы включать в себя большее количество банков, коммутационных пунктов и/или коммутаторов.

Хотя вышеизложенное было описано со ссылками на запросы транзакции, настоящее изобретение этим не ограничивается, и может работать с электронными сообщениями любого вида. Кроме того, хотя вышеизложенное было описано со ссылками на запросы транзакции, временно хранящиеся в очереди 106A сообщений перед их посылкой на шлюз 104A, настоящее изобретение этим не ограничено. Например, банковское приложение 102A может осуществлять связь непосредственно со шлюзом 104A без хранения запроса транзакции в очереди 106A сообщений. Это может быть достигнуто, если иметь буфер в шлюзе 104A, в котором хранят запросы транзакций, пока данный запрос транзакции не будет обслужен шлюзом 104A.

Очевидно, что возможно множество модификаций и вариантов настоящего изобретения в свете вышеописанных ключевых принципов. Таким образом, должно быть ясно, что в рамках объема прила-

гаемой формулы изобретения оно может быть на практике реализовано иначе, чем конкретно здесь описано.

Учитывая, что изобретение до настоящего момента было описано в виде, реализуемом по меньшей мере частично, устройством обработки данных, управляемым программным обеспечением, очевидно, что для представления варианта осуществления настоящего изобретения также рассматривается машиночитаемый носитель долговременного хранения, такой как оптический диск, магнитный диск, полупроводниковая память или т.п.

Следует понимать, что в вышеприведенном описании для ясности представлены варианты осуществления со ссылками на разные функциональные блоки, схемы и/или процессы. Однако очевидно, что возможно использование любого подходящего распределения функциональных возможностей между разными функциональными блоками, схемами и/или процессами, не умаляя упомянутые варианты осуществления.

Описанные варианты осуществления можно реализовать в любой подходящей форме, включая аппаратное обеспечение, программное обеспечение, программно-аппаратное обеспечение или любую их комбинацию. Описанные варианты осуществления могут, но не обязательно, быть реализованы по меньшей мере частично в виде компьютерного программного обеспечения, выполняющегося на одном или более процессорах данных и/или цифровых процессорах сигналов. Элементы и компоненты любого варианта осуществления могут быть физически, функционально или логически реализованы любым подходящим образом. В действительности, указанные функциональные возможности можно реализовать в одном блоке, во множестве блоков или как часть других функциональных блоков. Как таковые, раскрытые варианты осуществления можно реализовать в едином блоке или можно физически и функционально распределить между разными блоками, схемами и процессорами.

Хотя настоящее изобретение было описано в связи с рядом вариантов его осуществления, это не предполагает ограничение изобретения в изложенном здесь конкретном виде. Вдобавок, хотя в связи с конкретными вариантами осуществления может появиться признак, подлежащий описанию, специалисты в данной области техники понимают, что различные признаки описанных вариантов осуществления могут быть объединены любым образом, подходящим для реализации указанного здесь способа.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Интерфейс для управления пересылкой электронных транзакционных сообщений между первым финансовым институтом и множеством коммутаторов, распределенных по множеству коммутационных пунктов, при этом первый финансовый институт и множество коммутаторов соединены через сеть передачи данных, причем указанный интерфейс содержит коммуникационные схемы, обрабатывающие схемы и память, хранящую операционное состояние каждого коммутационного пункта, при этом коммуникационные схемы выполнены с возможностью передачи тестового сообщения на один из коммутационных пунктов через сеть передачи данных, если из указанного коммутационного пункта не было принято ни одного транзакционного сообщения в течение заданного времени, и в ответ на указанное тестовое сообщение коммуникационные схемы выполнены с возможностью приема эхосигнала тестового сообщения из указанного коммутационного пункта, при этом обрабатывающие схемы выполнены с возможностью, если указанный эхосигнал принят в течение некоторого определенного времени, установить операционное состояние указанного коммутационного пункта как рабочее, а если указанный эхосигнал не принят в течение указанного определенного времени, установить операционное состояние указанного коммутационного пункта как нерабочее; в результате чего в случае, когда операционное состояние указанного коммутационного пункта установлено как нерабочее, обрабатывающие схемы выполнены с возможностью не пересылать ни одного электронного сообщения на указанный коммутационный пункт; при этом память выполнена с возможностью хранения операционного состояния второго финансового института, подсоединенного к указанному множеству коммутаторов через сеть передачи данных; коммуникационные схемы выполнены с возможностью приема информации от одного или более из множества коммутаторов, указывающей обновленное операционное состояние указанного второго финансового института; и обрабатывающие схемы выполнены с возможностью обновления операционного состояния второго финансового института, хранящегося в указанной памяти, на основе информации, полученной от одного или более коммутаторов, при этом в случае, когда операционное состояние второго финансового института указывает, что второй финансовый институт находится в нерабочем состоянии, обрабатывающие схемы выполнены с возможностью предотвращения передачи на указанные коммутаторы электронных транзакционных сообщений, поручающих выполнение транзакции от первого финансового института во второй финансовый институт.

2. Интерфейс по п.1, в котором коммуникационные схемы выполнены с возможностью послышки дополнительного тестового сообщения в нерабочий коммутационный пункт после точно определенного периода.

3. Интерфейс по п.1, в котором информация, указывающая обновленное операционное состояние второго финансового института, создается одним или более из множества коммутаторов на основе от-

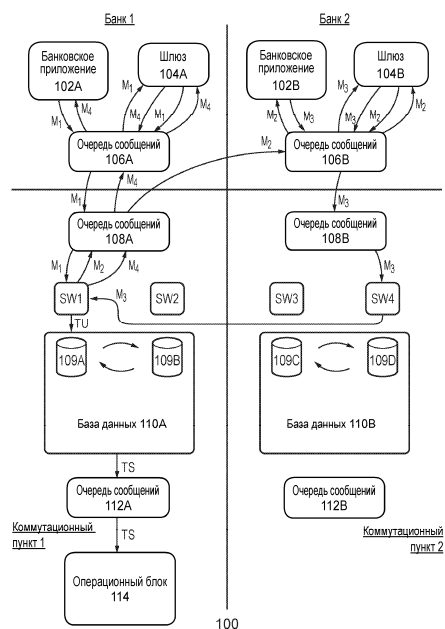
сутствия ответа на тестовое сообщение, переданное во второй финансовый институт каждым из коммутаторов в течение заданного периода времени, или указания вторым финансовым институтом одному или более из множества коммутаторов о том, что второй финансовый институт следует отсоединить от сети передачи данных.

4. Способ управления пересылкой электронных транзакционных сообщений между первым финансовым институтом и множеством коммутаторов, распределенных по множеству коммутационных пунктов, при этом первый финансовый институт и множество коммутаторов соединены через сеть передачи данных, включая использование интерфейса по любому из пп.1-3.

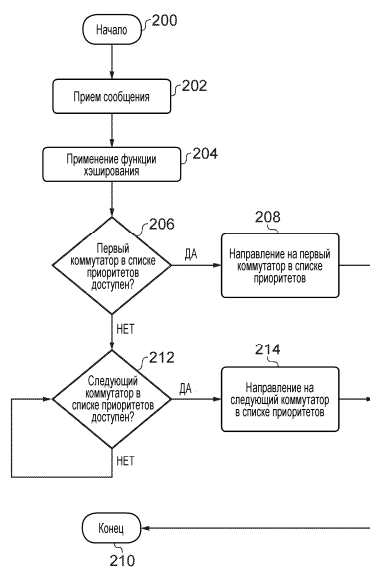
5. Способ по п.4, содержащий этап, на котором посылают дополнительное тестовое сообщение в нерабочий коммутационный пункт после некоторого точно определенного периода.

6. Способ по п.4, в котором информация, указывающая обновленное операционное состояние второго финансового института, создается одним или более из множества коммутаторов на основе отсутствия ответа на тестовое сообщение, переданное во второй финансовый институт каждым из коммутаторов в течение заданного периода времени, или указания вторым финансовым институтом одному или более из множества коммутаторов о том, что второй финансовый институт следует отсоединить от сети передачи данных.

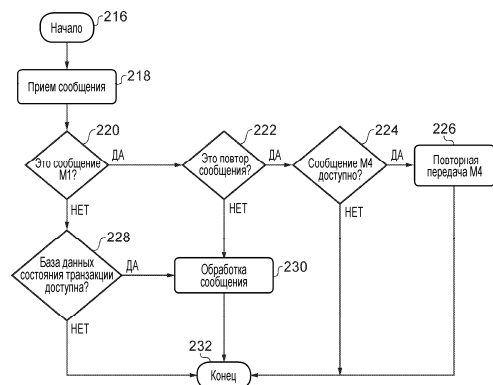
7. Машиночитаемый носитель, хранящий машиночитаемые команды, которые при выполнении на компьютере конфигурируют указанный компьютер для выполнения этапов способа по любому из пп.4, 5 или 6.



Фиг. 1



Фиг. 2А



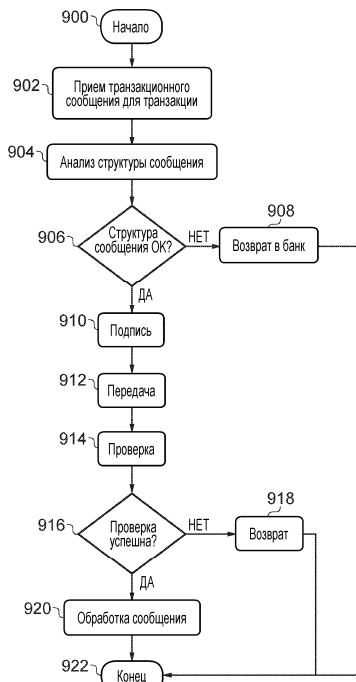
Фиг. 2В

602 Дебетовый лимит = -10000			
603 Дебетовый лимит = -5000 SRP пункта 1	604 Дебетовый лимит = -5000 SRP пункта 2	605 Общий дебетовый лимит = -10000 ОБЩИЙ SRP	
Дебет 1 2500		0	-2500
		Дебет 2 300	-2800
Дебет 3 2600		-300	-5400
↑ Не допускается, так как превышен дебетовый лимит пункта			

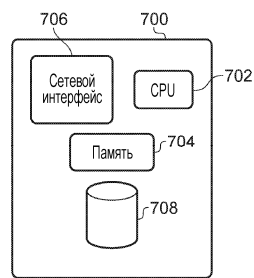
Фиг. 3А

Дебетовый лимит = -10000 651			
Дебетовый лимит = -5000 SRP пункта 1	Дебетовый лимит = -5000 SRP пункта 2	Общий дебетовый лимит = -10000 ОБЩИЙ SRP	
Дебет 1 2500		0	-2500
	Дебет 2 300	-300	-2800
Цикл настройки 656			
Настройка пункт 1 $= \left(\frac{-2800}{2}\right) - 2500$ $= +1100$	Настройка пункт 2 $= \frac{-2800}{2} - 300$ $= -1100$		
Настроенный SRP Дебет 3 2600 $= -2500 + 1100 + (-2600)$ $= -4000$	Настроенный SRP Новая транзакция $= -300 + 1100 + 0$ $= -1400$		-5400
Разрешено, так как не превышен дебетовый лимит £5000			

Фиг. 3В



Фиг. 4



Фиг. 5

