

Гарантии доставки и надежность.

Как мы знаем, ни одна система не является постоянно надежной. Могут быть разнообразные проблемы - как на стороне продюсера/консьюмера, так и на стороне брокера. Помимо этого возможны и общие сетевые проблемы, если сервисы работают не на одной сети. Что же делать брокеру? Как обеспечить надежность? Для реализации надежного взаимодействия существуют разные виды гарантий доставок, которые как раз обозначают, как будет взаимодействовать брокер сообщений с продюсерами и консьюмерами, чтобы повысить отказоустойчивость. Таким образом брокер предоставляет определенные обязательства относительно доставки сообщений от продюсера к консьюмеру.

Рассмотрим какие виды гарантий доставок существуют.

At Most Once (не более одного раза)	At Least Once (не менее одного раза)
Гарантируется, что сообщение будет доставлено консьюмеру не более одного раза.	Гарантируется, что сообщение будет доставлено консьюмеру как минимум один раз.
Сообщение может быть потеряно, но не будет доставлено более одного раза. Брокер работает по принципу «отправил и забыл», переотправки сообщения брокером не будет, если произошла ошибка или консьюмер не смог прочитать сообщение.	Брокер пытается доставить сообщение пока не получит подтверждения, что оно получено. Работает по принципу "отправил и проверил". Так как есть переотправка сообщения, оно может быть доставлено несколько раз.

Простота и низкая задержка являются основными преимуществами этой гарантии. Можно выбрать, если для вас важнее производительность и потеря некоторых сообщений не повлияет на работу сервиса, получающего сообщения.	В данном случае мы получаем менее производительную систему, т.к. на каждое сообщение нужно подтверждение о получении. Также могут потребоваться дополнительные настройки брокера/консьюмера, чтобы не дублировать сообщения в системе (если дубли для нас критичны).
Для корректной реализации брокеру не требуется записывать данные на диск, он "отправил и забыл" сообщение.	Для корректной реализации брокеру требуется записывать данные на диск, чтобы была возможность отправить сообщение консьюмеру еще раз.
Продюсер не должен поддерживать повторную отправку сообщений.	Продюсер должен поддерживать повторную отправку сообщений.

Помните мы ранее рассматривали разницу протоколов TCP и UDP? Так вот, как раз at most once можно обозначить как UDP, отправляется поток сообщений и мы не ждем подтверждения. А at least once можно обозначить как TCP, на каждое сообщение нужно подтверждение о получении.

Если работа "не более одного раза" понятна и выглядит просто, это доставка сообщений как в ранее приведенных примерах, без доп.настроек. То "не менее одного раза" работает сложнее:

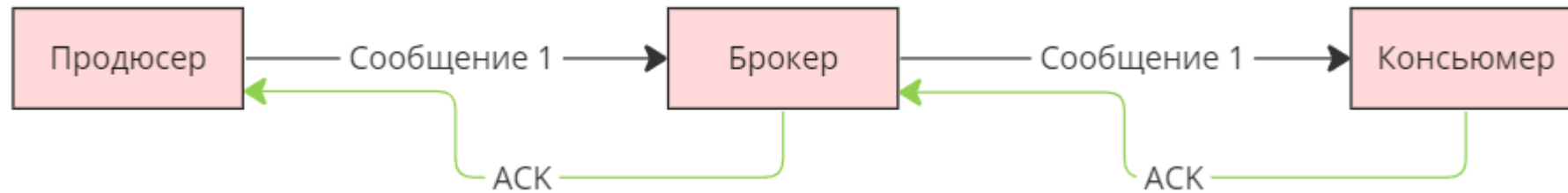
- Используется механизм Ack/Nack. Ack - подтверждено (acknowledge), Nack - не подтверждено (not acknowledge)
 1. Каждое сообщение содержит уникальный идентификатор
 2. Продюсер отправляет сообщение брокеру и ожидает от брокера подтверждения, что сообщение получено

3. Брокер, когда пришло время, отправляет сообщение консьюмеру и ожидает от клиента подтверждения, что сообщение получено

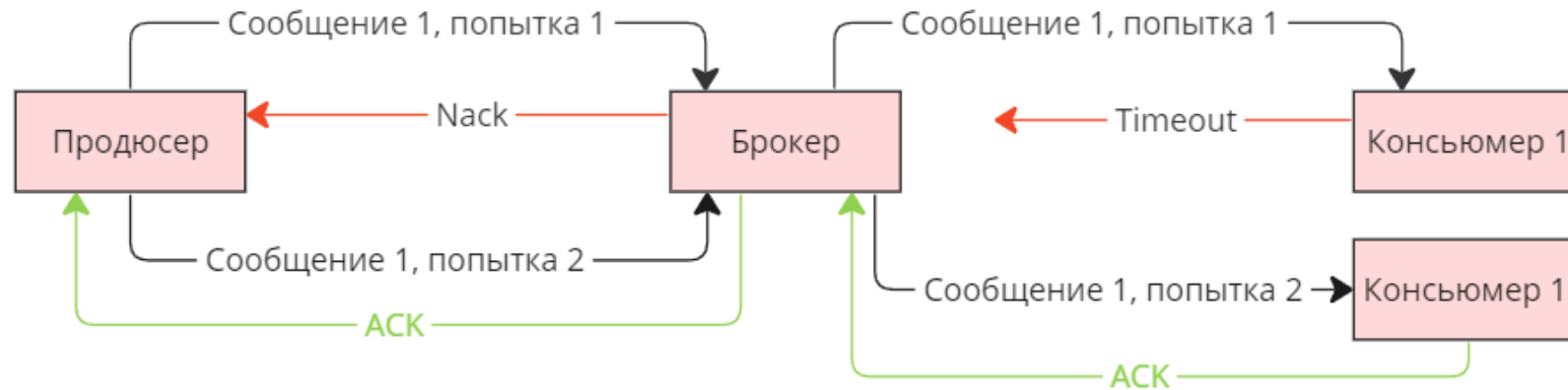
Помните, мы рассматривали push и pull модели? В случае с pull, гарантия доставки на стороне отношений брокер-консьюмер (третий пункт) ложится на "плечи" консьюмера. Механизм, когда консьюмер ответственный:

- Каждое сообщение также содержит уникальный идентификатор, содержащий его порядковый номер
- Когда консьюмер видит, что пропущено какое-либо сообщение, он запросит у продюсера повторно все сообщения с пропущенного оффсета (с определенного места, про оффсет мы говорили ранее). То есть консьюмер ничего не подтверждает, но следит за порядком полученных сообщений.

Успешная схема в случае push модели будет выглядеть так:



То есть на каждом шаге вы будете уверены, что сообщение точно будет доставлено. При этом оно может быть доставлено несколько раз. Приведем пример с двумя инстансами консьюмеров.



В данном случае:

- Продюсер отправил сообщение брокеру, но получил Nack, допустим, у брокера закончилась память и он не смог поместить к себе сообщение. Продюсер повторно переотправил сообщение, получил успешное подтверждение.
- Брокер отправил сообщение на первый инстанс консьюмера 1, консьюмер его получил, но по какой-либо причине не смог отправить подтверждение. Случился Timeout, брокер не получил никакой информации, и отправляет сообщение еще раз на второй инстанс консьюмера 1. Получает подтверждение об успешном получении.
- В данном случае у сервиса с двумя инстансами (консьюмер 1) оказалось 2 сообщения от брокера

Также и брокер может не успеть отправить подтверждение и тогда у него может оказаться 2 сообщения. Что делать, чтобы избежать дублей сообщений в брокере и консьюмерах? Допустим, если это сервис мониторинга, где происходит сбор метрик в большом количестве (например, температура комнаты измеряется каждые 5 секунд), это может быть не критичным случаем. Но если консьюмером будет сервис создания заданий, мы получим дубль задания, что плохо может повлиять на обслуживание клиентов. В таком случае реализуют вид гарантии "не более одного раза". Этот механизм не является типичным и не реализуется "из коробки" при использовании брокера, поэтому мы его обсудим как отдельную историю.

Если нужно спроектировать работу брокера и консьюмера так, чтобы избегать дублей, используется механизм доставки **Exactly Once** (ровно один раз, иногда называют **Effectively Once**) - гарантируется, что сообщение будет доставлено консьюмеру ровно один раз, без дублирования или потери. Для данного механизма также требуется запись сообщений на стороне брокера, и поддержка отправки повторного сообщения на стороне продюсера. Иначе при неуспешной доставке переотправить сообщения не получится.

Это наиболее сложный тип гарантии и требует дополнительных механизмов и поддержки от брокера и консьюмера. В данном случае также используется механизм Ack/Nack, т.е. также как и в at least once, но еще реализуют дополнительные техники, т.к. дубли могут повлиять на работу системы:

- Транзакции: брокеры могут поддерживать транзакции, позволяющие гарантировать атомарность обмена сообщениями. Это означает, что сообщения будут успешно доставлены консьюмерам только в том случае, если все операции с сообщениями в рамках транзакции были успешно завершены. В случае сбоя на каком-либо этапе, транзакции (все действия) могут быть отменены, чтобы избежать нежелательных эффектов
- Дедупликация: для предотвращения дублирования сообщений можно использовать механизм дедупликации, который идентифицирует и удаляет дубликаты сообщений, прежде чем они будут доставлены консьюмерам

- Идемпотентность: консьюмеры должны обрабатывать сообщения идемпотентно, чтобы повторное получение сообщений не привело к нежелательным эффектам. Идемпотентность в данном случае означает, что обработка сообщения не изменит состояние системы, если оно будет обработано несколько раз

Какой же тип гарантий доставки использовать? Все зависит от конкретных требований вашего приложения. Например, где потеря некоторых сообщений не является проблемой, можно выбрать гарантию at most once для повышения производительности. А где надежность и точность доставки критичны, гарантия exactly once может быть предпочтительной, но она может быть более сложной и затратной в реализации. Большинство брокеров предоставляют гарантии at least once и at most once из коробки, а exactly once требует дополнительной разработки.

Отдельно отметим про запись сообщений на стороне брокера. Могут быть проблемы такого уровня: допустим, «упадет» брокер или потеряет все сообщения. Тогда для реализации схемы at least once или exactly once нам необходимо включить запись полученных сообщений на диск, иначе все сообщения могут быть утеряны, и ретраи будут невозможны. Брокеры умеют хранить сообщения у себя в оперативной памяти или записывать их на диск. Как вы понимаете, хранение в оперативной памяти может быть не надежным способом. Но при записи на диск могут быть потери производительности, однако получаем гарантию более высокого уровня. Здесь вам тоже нужно выбрать, какой вариант для вас предпочтительнее.

Чтобы выбрать гарантию доставки, можете для начала ориентироваться на эти вопросы:

1. Критично ли для потребителей получать каждое сообщение без потерь?

1. Да (at least once или exactly once)

2. Нет (at most once)

2. Повлияют ли дубли сообщений на работу потребителей?

1. Да (exactly once)

2. Нет (at least once)

3. Готовы ли вы пожертвовать гарантиями доставки в пользу быстрой передачи сообщений?

1. Да (at most once)

2. Нет (at least once или exactly once)

Давайте составим таблицу-ориентир, какая гарантия доставки будет эффективна в том или ином случае.

At most once	At least once	Exactly once
Сервис собирает данные о температуре помещения	Сервис собирает данные о действиях пользователя	Сервис создает задания после получения сообщения
Сервис собирает какие-либо координаты (местоположение)	Сервис отправляет некритичные push-уведомления клиенту (нам не критична повторная доставка)	Сервис формирует доставку заказа после получения сообщения

Заметим, что на одном проекте (в одном веб-приложении) вы можете использовать разные виды гарантий, все зависит от требований к вашему приложению.