DesignNotesOnReliableDistributedTopic

This is very much a work in progress!

Context

This is a small step in the direction of clarifying what the 'dispatch router' might be in more detail. It is looking into one potential requirement to understand if /how that could be delivered in keeping with what I understand to be the design principles of that component. This isn't necessarily advocating that this is the most important feature however.

Plan of action:

First explore design for exactly-once pub-sub using AMQP throughout.

Then consider how this design would be affected by (a) relaxing the delivery guarantee to at-least-once and, orthogonally, (b) using MQTT at the edges.

Overview

Requirement:

- support for exactly-once delivery in a pub-sub distribution pattern where publishers and subscribers are connected through a network of dispatch router instances
- delivery guarantee holds even in the face of individual router failure

Initial Assumptions:

- using AMQP at the edges
- single, non-hierarchical topic
- any filters on receiving links are applied only at the edge

Design principles:

- router network never assumes ultimate responsibility for messages (i.e. no store-and-forward, only accept message from publishers when it has already been accepted by subscribers)
- only rely on weak consistency between router instances in the network; no synchronously replicated state between routers

Basic pattern:

When a receiving link (aka a subscriber) for the topic attaches to a router, it will communicate that fact to all other routers, including the container-id and link name of the link and an indication of whether there was any unsettled state associated with the link at the time of attachment (i.e. loosely whether the link can be considered 'new' link attaching or an 'old' link resuming).

Any router receiving messages for the topic on a sending link (i.e. from a publisher), will forward that message to all other routers in the network that have subscribers attached to them (along the best path available).

Each router needs to track the acceptance of a given message from its local subscribers. It will signal its own acceptance when all its local subscribers have accepted.

The router to which the publishing link is attached will additionally track the acceptance from all the other router instances to which it forwarded the message and will only communicate its acceptance to that publisher on receipt of acceptance both from all its local subscribers and all those other router instances.

The publisher, on receiving acceptance, will then settle the message. The settlement will be relayed to each local subscriber and each router that sent an acceptance. All the other routers likewise settle the delivery for their respective local subscribers.

Delivery Ids for outgoing messages:

Scenario: There are two connected routers, A and B, serving a given topic. A publisher and a subscriber for that topic are both connected to router A. The publisher sends a message which the router forwards to the subscriber. Before the subscriber accepts this, router A fails. Both the publisher and the subscriber failover to router B and attempt to recover their links.

It must be possible to match a delivery to a subscriber with the corresponding delivery from a publisher. However since there may be multiple publisher streams being combined for delivery to a subscriber, the publisher's delivery tag is not on its own sufficiently unique.

Ideally, the delivery tag assigned by the router network would include the publishers container id and link name as well as the original delivery tag.

However the delivery-tag is limited in size (it can contain 32 bytes at most).

Since we don't want to have to maintain a per-message mapping, we either need to rely on the publishers tag being less than 32 bytes or we need to use a delivery tag larger than that for subscribers in order to be able to retain the original while ensuring uniqueness.

Routers also need to be able to identify the original publisher of any message they have received. That can be done by adding an annotation to the message on the first receving router instance.

Inter-router communication:

Each router will forward messages to all other routers that have subscribers for the topic. This needs to be 'reliable' in the sense that messages can't go missing. However we can resend and it doesn't need to use the AMQP defined exactly-once procedure between routers. To start with I'll assume it doesn't for simplicity, this can be revisited later.

(Question: If routers C and D are both reached by A through an intermediary router B, and all of B, C and D have subscribers, should one message be sent by A to all of them, or would three messages each with an explicit target router be sent?)

So, delivery record at a given node tracks:

(a) local subscribers and the state of the delivery to each of them, and

(b) the other router instances with subscribers for the topic

It will signal its own acceptance when all its local subscribers have accepted. The router to which the publishing link is attached will send its acceptance to that publisher only when both all its local subscribers have accepted, and all the router nodes it is expecting to have accepted it.

Each router tracks the set of subscribers connected at each other router. On the failure of a router instance, any other router that was waiting for an acceptance from that failed router will track the reconnection of the subscribers on that failed router to (an)other router(s). There will also be a timeout such that if they don't retattach, any state associated with the link is discarded. (This would be the value for the timeout field of a terminus that the router would accept).

When a receiving link attaches to a router, it will send out a message describing that to all other routers in the network. Likewise it will notify the network if/when that link is closed.

Routers track the subscriber identifiers (container-id and link name) attached to all other nodes in the network. (TODO: This is not ideal in terms of scalability, and there may be some ways to relax the requirement a little).

On failure of a router, the rest of the network can then determine when all the receivers that were attached to that failed node have reattached, and can update their records of any deliveries for which a response was expected from the failed node.

Settlement:

Once settled, a router no longer needs to hold on to the message itself, but we do need to track the delivery until we are confident that every receiver knows it has been settled. This allows us to assume that when resuming a receiver link, any unsettled deliveries declared by the receiver that the router is unware of, have yet to make it to that router.

With AMQP 1.0 there is no indication sent back by receivers to indicate that they know a given delivery has been settled (since it is assumed that the sender has already discarded any record of the delivery). We could however **infer** that they know it has been settled when they have responded to something sent after the disposition notifying them of that settlement.

The router to which the publisher of a message is attached will relay the settlement and will also subsquently communicate to all nodes that the delivery record can be deleted when a subsequent delivery from a publisher who is also attached to this router is accepted.

[TODO: Ordering assumption here needs to be scrutinised in the face of failover....]

---

resuming links on failover

Resuming a publisher:

On having a publishing link attach with unsettled state, the router to which it attaches will examine its delivery records to see which if any of the unsettled deliveries it has any record of.

For those deliveries for which it already has a record, it will start tracking acceptance from all other interested routers, to which these deliveries will now be resent.

It will request that the publisher resend any deliveries it was not aware of, which will be then added to its records and forwarded under a disambiguated delivery tag to its local subscribers if any exist, as well as all interested routers in the network.

The router will not indicate acceptance of any of these deliveries until all interested routers and local subscribers have accepted them.

Resuming a subscriber:

On having a receiving link attach with unsettled state, the router will compare the unsettled delivery states as presented by the receiver with its own records. It can respond with its own view of delivery state for any delivery it already has a record of. Those deliveries it does not have a record of could have been settled or not yet received.

For deliveries the receiver considers unsettled and the router has a record of, if the record indicates the delivery was settled, then the router can omit the delivery from its own unsettled map, otherwise the router will start tracking this receiver against that record and include these in the unsettled map we send back. If receiver indicates it has accepted, we can mark it as accepted by them.

For deliveries the receiver considers unsettled but the router has no record of, the router assumes it has yet to receive those messages. It creates a placeholder record for them and starts tracking this receiver against that. It shouldn't deliver any messages to that receiver until all preceding placeholders have been 'fulfilled', i.e. until it has received a message matching that delivery, updated the placeholder record accordingly and forwarded that message to this receiver (and any others that may have also resumed including that delivery in their unsettled state).

Any deliveries receiver doesn't report in its unsettled map either have been settled (and are no longer relevant) or have yet to be delivered to that receiver. Any unsettled records for the topic that the router has that are not in the receivers unsettled map, it should resend to that receiver and track the receivers acceptance of it.

Delivery records kept for topic:

- the message
- the publishers identity
- the publishers original delivery tag
- the outgoing delivery tag (includes the publisher tag, adds disambiguating pre- or post- fix)
- a map of local subscribers and their respective delivery statuses (i.e. in-doubt, accepted, settled)
- a map of other interested routers and their respective delivery statuses (i.e. in-doubt, accepted, settled) Note that this will only be tracked by the router to which the publisher of the message is attached.