Federation Design Note

Design Note

The information below is quite stale. The outstanding issues listed have been resolved as of M3.

Information needed for this design note:

- Mapping of federation features to source files
- Full description of the dynamic binding protocol and its associated algorithms
- A discussion of how changes to AMQP could improve the protocol (the main thing needed is an arguments map in the unbind method)

Old Content

Formatted mail from Gordon...

Regarding federation, what we have now in the c++ broker is really inter-broker routing.

Links between brokers can be setup to transfer messages from one to another.

In the current terminology a 'link' is a connection between two brokers. Such a link is setup using the management system, by asking one broker to establish a connection to another broker given the host and port.

Once a link is established, a 'bridge' can be created. A bridge is essentially a subscription for messages between two brokers, requesting the transfer of messages from a source to a destination. The 'source' for a bridge can logically be either an exchange or a queue; the destination is an exchange on the receiving broker.

The current implementation of bridges relies on the symmetry of the message.transfer command in the 0-10 AMQP specification. A bridge is created by issuing a subscribe request to one broker using the exchange name to which the messages should be delivered as the 'destination' argument. So once the subscription is setup the bridge apperas to be a standard consumer to the source broker and messages routed from that broker appear as standard publications at the source broker.

If the logical source of the bridge was an exchange rather than a queue, an exclusive queue is created for the bridge and bound with the relevant binding details (currently only a binding key is supported, but thats easy to extend).

Bridges can be established to support different types of message flow. A common case is where you have two or more brokers over which you want to offer a 'federated exchange'. I.e. you want messages published to that exchange on one broker to be routed through the equivalent exchanges on all the brokers in the federation, allowing queues bund locally at those brokers to receive such messages. This common case is supported by the qpid-route tool.

There are currently a few outstanding issues needing to be resolved.

- One is preventing messages from looping in configurations where there are circularities in the defined routes (such as those described for the 'federated exchange'). I plan to address that next week. The solution I have in mind is to have the exclusive queues used form bridging from exchanges append an identifier to a custom property ('x-qpid-route' or whatever) in each message that passes through them. It will then be possible to specify a list of exclusions when establishing a bridge and messages where the route property contains any of the excluded identifiers will be silently dropped. I'm not entirely delighted with that approach, but it will have to do in the short term I think.

- Another is ensuring that links are re-established when lost (e.g. due to network failures or brokers being taken down) and that the details of configured bridges survive restart. These will also be addressed quite soon I hope.

This is obviously just the beginning of full federation capabilities. There are many ways it can be made more sophisticated and I for one would be interested in debating ideas, use cases and directions.