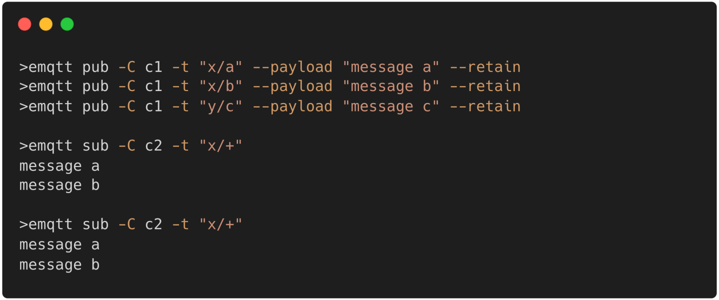
Retainer topic indexing

Ilya Averyanov

2022

Retaining feature



```
>emqtt pub -C c1 -t "x/a" --payload "message a" --retain
>emqtt pub -C c1 -t "x/b" --payload "message b" --retain
>emqtt pub -C c1 -t "y/c" --payload "message c" --retain

>emqtt sub -C c2 -t "x/+"
message a
message b

>emqtt sub -C c2 -t "x/+"
message a
message b
```

Retaining feature
Externally

- Each topic can have a *retained* message
- This message is sent to each subscriber on topic subscribe
- A subscriber can subscribe to many topics at once using wildcards and receive many retained messages

Retaining feature
Internally

- A broker needs to keep a *database* of topics and their corresponding retained messages
- On topic subscribe, a broker makes a lookup of corresponding retained messages

Retainer database

Internally

- A subscribed topic can be "concrete", like `a/b/c/d`. Then we need a simple lookup in a key-value storage to find a retained message
- A subscribed topic can be a wildcard topic, like `+/b/+/d`. Then we first need to select relevant topics and only then lookup the corresponding retained messages
- There may be a lot of topics with retained messages so selecting relevant topics may be resourse consuming https://github.com/emqx/emqx/issues/6014

Retainer indexing

Solution

To provide effective topic search we introduce topic indexing. Topic indexing allows to specify topic segment positions that are supposed to be meaningful and not wildcard during subscriptions. Retained messages will be searched effectively during such subscriptions.

Retainer indexing
Example

Assume we have the following topic structure:

```
<Country>/<City>/<Office>/<DeviceModel>/<Owner>
```

For example:

```
UK/London/HQ/ModelA/User4534
```

Retainer indexing

Example

We want to subscribe to topics corresponding to some particular `DeviceModel` and `Owner`.

The wildcard will be: `+/+/+/<DeviceModel>/<Owner>`

Retainer indexing
Example

To optimize finding retained messages for such a subscription we can specify the following index:

```
retainer {
    backend {
        ...
        index_specs = [ [4, 5] ]
    }
}
```

Retainer indexing

Notes

- By default there are indices that help to handle wilcard subscriptions having any meaningful segments in the first 3 positions
- To enable new indices after config change one should explicitly run reindex: `emqx_ctl retainer reindex start`

Thank you!