KIP-272: Add API version tag to broker's RequestsPerSec metric

Status

Current state: *Accepted*

Discussion thread: here

JIRA: *here*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Kafka changes message format from release to release. When upgrading Kafka version, typically the message format version recommended to set to old version before all clients upgrade to the latest release. Otherwise broker has to pay significant performance penalty to down convert message. Only when most clients have upgraded can the broker change to the latest message format.

In this process, there is no visibility from broker side to see what portion of the clients have been upgraded. Therefore changing the message format version becomes a difficult decision.

Proposed Changes

Previous metric: kafka.network:type=RequestMetrics,name=RequestsPerSec,request={Produce|FetchConsumer|FetchFollower|...}

*New metric: kafka.network:type=RequestMetrics,name=RequestsPerSec,request={Produce|FetchConsumer|FetchFollower|...},**version=INTEGER***

We want to amend RequestsPerSec metric to have "version" tag so that we have the insight of versions that clients are using. The value is an integer that represent the version for the specific request type. This can be done in RequestChannel where it already parses an keeps the client API version in memory and updates the RequestsPerSec metric. With this change, an additional hash lookup is needed when updating the metric to locate the metric corresponding to a specific version.

The metrics for all versions will be cleaned up at broker shutdown.

Details of the change can be viewed from the pull request

Compatibility, Deprecation, and Migration Plan

- *This may break the user who build monitoring tools which does not expect additional tag for RequestsPerSec metric for a specific request. To see the total count, they need to aggregate over all versions.*

Rejected Alternatives

- We can create a new metric for it. However, it will incur more more storage cost for any metric system since we will be duplicating some request information in the new metric. It also makes it less user friendly for metric systems that can already "slice and dice" on different dimensions (tags) of metrics. In this case, it is easy to "drill down" to a tag without having to look at a completely different metric.
- Create separate meter for each version, but also keep the meter for the request type and increment both at the same time. This creates a problem for metric systems that automatically aggregate for the same metric name and create a "double count" problem.
- Use the new metrics in KIP-188. The problem is that users have to upgrade message format first before seeing the impact, which might be too late and does not help users to make informed decisions.