KIP-359: Verify leader epoch in produce requests

Status

Current state: Adopted

Discussion thread: here

JIRA: ⚠ Unable to render Jira issues macro, execution error.

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

Motivation

In KIP-320, we introduced fencing support to the Fetch API, which gives consumers the ability to reason about the fetch state relative to the current leader and detect stale metadata. This KIP proposes a similar extension to the Produce API. We will add the leader epoch to the produce request and the partition leader will verify that it matches its own epoch before appending to the local log.

This change improves the debuggability of the producer and makes its behavior consistent with the consumer in its metadata usage. There is no substantial benefit in terms of the achievable semantics, but it does make it less likely that a write will be accepted by a stale leader, which can be problematic when there are consecutive leader elections. Additionally, it is advantageous for the producer to know whether its metadata is ahead or behind the leader in order to avoid unnecessary metadata refreshing.

Public Interfaces

We will bump the version of the Produce request API to include the leader epoch. For old versions, we will skip the epoch validation and retain the current behavior.

```
ProduceRequest => TransactionalId RequiredAcks Timeout [TopicData]
 TransactionalId => NULLABLE_STRING
 RequiredAcks => INT16
 Timeout => INT32
 TopicData =>  [TopicName [PartitionData]]
   PartitionData => Partition CurrentLeaderEpoch MessageSetSize MessageSet]
     Partition => INT32
     CurrentLeaderEpoch => INT32 // New
     MessageSetSize => INT32
     MessageSet => BYTES
```

The Produce response schema will match the previous version.

As in KIP-320, we will use the error codes FENCED_LEADER_EPOCH and UNKNOWN_LEADER_EPOCH when the leader epoch does not match.

Proposed Changes

Now that we include the leader epoch in the Metadata API, the producer will be able to add it to Produce requests to verify metadata consistency between the client and the leader. If the producer sees the FENCED_LEADER_EPOCH in a produce response, it will refresh metadata and retry. Alternatively, in the case of UNKNOWN_LEADER_EPOCH, which means the producer has more recent metadata than the broker, the producer will simply backoff and retry.

We think it may also be useful to add some randomness to the producer's periodic metadata refreshes to reduce the likelihood of the producer getting stuck on stale metadata, but this is an internal detail and not explicitly in the scope of the KIP.

Compatibility, Deprecation, and Migration Plan

There will be no compatibility impact. Older versions of the produce API will skip the epoch check. Additionally, newer clients will continue to work with older brokers.

Rejected Alternatives

None at the moment.