

git_comments:

1. The second batch should not bump the epoch as txn manager is already in fatal error state
2. Handling b1 should bump the epoch after OutOfOrderSequenceException

git_commits:

1. **summary:** KAFKA-9605; Do not attempt to abort batches when txn manager is in fatal error (#8177)
message: KAFKA-9605; Do not attempt to abort batches when txn manager is in fatal error (#8177) We detected a bug in soak where the producer batches shall be failed in sender loop before the produce response callback. This shall trigger an illegal state exception on the producer batch as it is already aborted. The impact is not severe since sender is on its own thread but should be fixed to avoid unnecessary critical exception. Reviewers: Bob Barrett <bob.barrett@confluent.io>, Jason Gustafson <jason@confluent.io>

github_issues:**github_issues_comments:****github_pulls:**

1. **title:** KAFKA-9605: Do not attempt to abort batches when txn manager is in fatal error
body: We detected a bug in soak where the producer batches shall be failed in sender loop before the produce response callback. This shall trigger an illegal state exception on the producer batch as it is already aborted. The impact is not severe since sender is on its own thread but should be fixed to avoid unnecessary critical exception. ### Committer Checklist (excluded from commit message) - [] Verify design and implementation - [] Verify test coverage and CI build status - [] Verify documentation (including upgrade notes)

github_pulls_comments:

1. @bob-barrett @hachikuji Could you take a look when you got time?
2. ok to test
3. Thanks @bob-barrett ! Note that we got 1/2 green build already.
4. test this please
5. retest this please

github_pulls_reviews:

1. nit: we might want log different message if we're ignoring due to a fatal state instead of due to a bumped epoch or ID.
2. 👍
3. sg

jira_issues:

1. **summary:** EOS Producer could throw illegal state if trying to complete a failed batch after fatal error
description: In the Producer we could see network client hits fatal exception while trying to complete the batches after Txn manager hits fatal fenced error: {code:java} [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) [2020-02-24 21:23:28,673] ERROR [kafka-producer-network-thread | stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer] [Producer clientId=stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer, transactionalId=stream-soak-test-1_0] Aborting producer batches due to fatal error (org.apache.kafka.clients.producer.internals.Sender) [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) org.apache.kafka.common.errors.ProducerFencedException: Producer attempted an operation with an old epoch. Either there is a newer producer with the same transactionalId, or the producer's transaction has been expired by the broker. [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) [2020-02-24 21:23:28,674] INFO [stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3] [Producer clientId=stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-0_0-producer, transactionalId=stream-soak-test-0_0] Closing the Kafka producer with timeoutMillis = 9223372036854775807 ms. (org.apache.kafka.clients.producer.KafkaProducer) [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) [2020-02-24 21:23:28,684] INFO [kafka-producer-network-thread | stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer] [Producer clientId=stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer, transactionalId=stream-soak-test-1_0] Resetting sequence number of batch with current sequence 354277 for partition windowed-node-counts-0 to 354276 (org.apache.kafka.clients.producer.internals.TransactionManager) [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) [2020-02-24 21:23:28,684] INFO [kafka-producer-network-thread | stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer] Resetting sequence number of batch with current sequence 354277 for partition windowed-node-counts-0 to 354276 (org.apache.kafka.clients.producer.internals.ProducerBatch) [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) [2020-02-24 21:23:28,685] ERROR [kafka-producer-network-thread | stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer] [Producer clientId=stream-soak-test-5e16fa60-12a3-4c4f-9900-c75f7d10859f-StreamThread-3-1_0-producer, transactionalId=stream-soak-test-1_0] Uncaught error in request completion: (org.apache.kafka.clients.NetworkClient) [2020-02-24T13:23:29-08:00] (streams-soak-trunk-eos_soak_i-02ea56d369c55eec2_streamslog) java.lang.IllegalStateException: Should not reopen a batch which is already aborted. at org.apache.kafka.common.record.MemoryRecordsBuilder.reopenAndRewriteProducerState(MemoryRecordsBuilder.java:295) at org.apache.kafka.clients.producer.internals.ProducerBatch.resetProducerState(ProducerBatch.java:395) at org.apache.kafka.clients.producer.internals.TransactionManager.lambda\$adjustSequencesDueToFailedBatch\$4(TransactionManager.java:770) at org.apache.kafka.clients.producer.internals.TransactionManager\$TopicPartitionEntry.resetSequenceNumbers(TransactionManager.java:180) at org.apache.kafka.clients.producer.internals.TransactionManager.adjustSequencesDueToFailedBatch(TransactionManager.java:760) at org.apache.kafka.clients.producer.internals.TransactionManager.handleFailedBatch(TransactionManager.java:735) at org.apache.kafka.clients.producer.internals.Sender.failBatch(Sender.java:671) at org.apache.kafka.clients.producer.internals.Sender.failBatch(Sender.java:662) at org.apache.kafka.clients.producer.internals.Sender.completeBatch(Sender.java:620) at org.apache.kafka.clients.producer.internals.Sender.handleProduceResponse(Sender.java:554) at

```
org.apache.kafka.clients.producer.internals.Sender.access$100(Sender.java:69)      at
org.apache.kafka.clients.producer.internals.Sender$1.onComplete(Sender.java:745)    at
org.apache.kafka.clients.ClientResponse.onComplete(ClientResponse.java:109)      at
org.apache.kafka.clients.NetworkClient.completeResponses(NetworkClient.java:571)   at
org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:563)              at
org.apache.kafka.clients.producer.internals.Sender.runOnce(Sender.java:304)        at
org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:239)           at java.lang.Thread.run(Thread.java:748) {code} The proper
fix is to add a check for handle failed batch in txn manager.
```

jira_issues_comments:

1. hachikuji commented on pull request #8177: KAFKA-9605: Do not attempt to abort batches when txn manager is in fatal error URL:
<https://github.com/apache/kafka/pull/8177> ----- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org