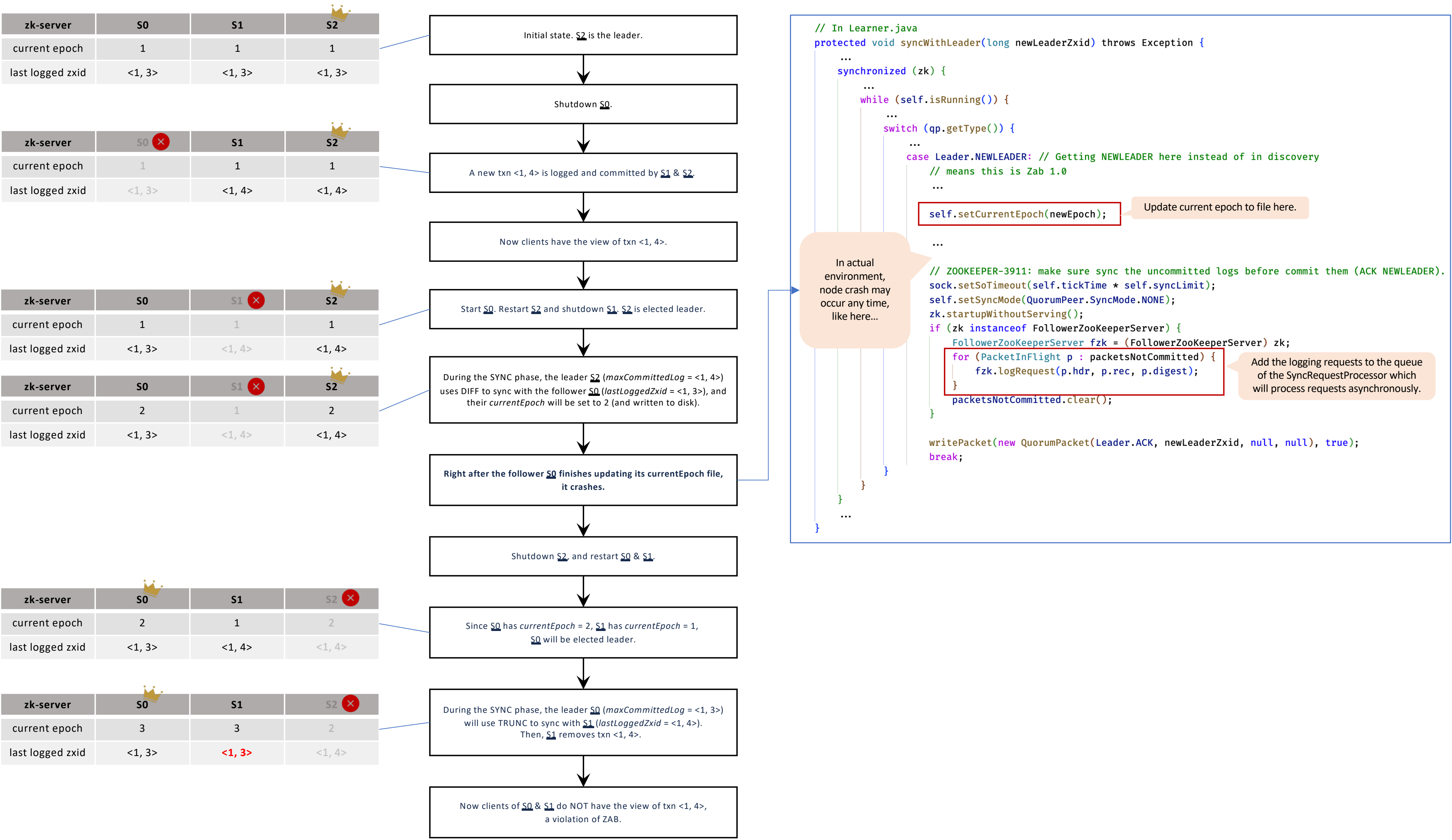
Example trace of ZOOKEEPER-4643



zk-server	S0	S1	S2 👑
current epoch	1	1	1
last logged zxid	<1, 3>	<1, 3>	<1, 3>

Initial state. S2 is the leader.

Shutdown S0.

zk-server	S0 ✖	S1	S2 👑
current epoch	1	1	1
last logged zxid	<1, 3>	<1, 4>	<1, 4>

A new txn <1, 4> is logged and committed by S1 & S2.

Now clients have the view of txn <1, 4>.

zk-server	S0	S1 ✖	S2 👑
current epoch	1	1	1
last logged zxid	<1, 3>	<1, 4>	<1, 4>

Start S0. Restart S2 and shutdown S1. S2 is elected leader.

zk-server	S0	S1 ✖	S2 👑
current epoch	2	1	2
last logged zxid	<1, 3>	<1, 4>	<1, 4>

During the SYNC phase, the leader S2 (*maxCommittedLog* = <1, 4>) uses DIFF to sync with the follower S0 (*lastLoggedZxid* = <1, 3>), and their *currentEpoch* will be set to 2 (and written to disk).

Right after the follower S0 finishes updating its currentEpoch file, it crashes.

```
// In Learner.java
protected void syncWithLeader(long newLeaderZxid) throws Exception {
    ...
    synchronized (zk) {
        ...
        while (self.isRunning()) {
            ...
            switch (qp.getType()) {
                ...
                case Leader.NEWLEADER: // Getting NEWLEADER here instead of in discovery
                    // means this is Zab 1.0
                    ...
                    self.setCurrentEpoch(newEpoch);
                    ...
                    // ZOOKEEPER-3911: make sure sync the uncommitted logs before commit them (ACK NEWLEADER).
                    sock.setSoTimeout(self.tickTime * self.syncLimit);
                    self.setSyncMode(QuorumPeer.SyncMode.NONE);
                    zk.startupWithoutServing();
                    if (zk instanceof FollowerZooKeeperServer) {
                        FollowerZooKeeperServer fzk = (FollowerZooKeeperServer) zk;
                        for (PacketInFlight p : packetsNotCommitted) {
                            fzk.logRequest(p.hdr, p.rec, p.digest);
                        }
                        packetsNotCommitted.clear();
                    }

                    writePacket(new QuorumPacket(Leader.ACK, newLeaderZxid, null, null), true);
                    break;
            }
        }
    }
    ...
}
```

Update current epoch to file here.

In actual environment, node crash may occur any time, like here...

Add the logging requests to the queue of the SyncRequestProcessor which will process requests asynchronously.

Shutdown S2, and restart S0 & S1.

zk-server	S0 👑	S1	S2 ✖
current epoch	2	1	2
last logged zxid	<1, 3>	<1, 4>	<1, 4>

Since S0 has *currentEpoch* = 2, S1 has *currentEpoch* = 1, S0 will be elected leader.

zk-server	S0 👑	S1	S2 ✖
current epoch	3	3	2
last logged zxid	<1, 3>	**<1, 3>**	<1, 4>

During the SYNC phase, the leader S0 (*maxCommittedLog* = <1, 3>) will use TRUNC to sync with S1 (*lastLoggedZxid* = <1, 4>). Then, S1 removes txn <1, 4>.

Now clients of S0 & S1 do NOT have the view of txn <1, 4>, a violation of ZAB.