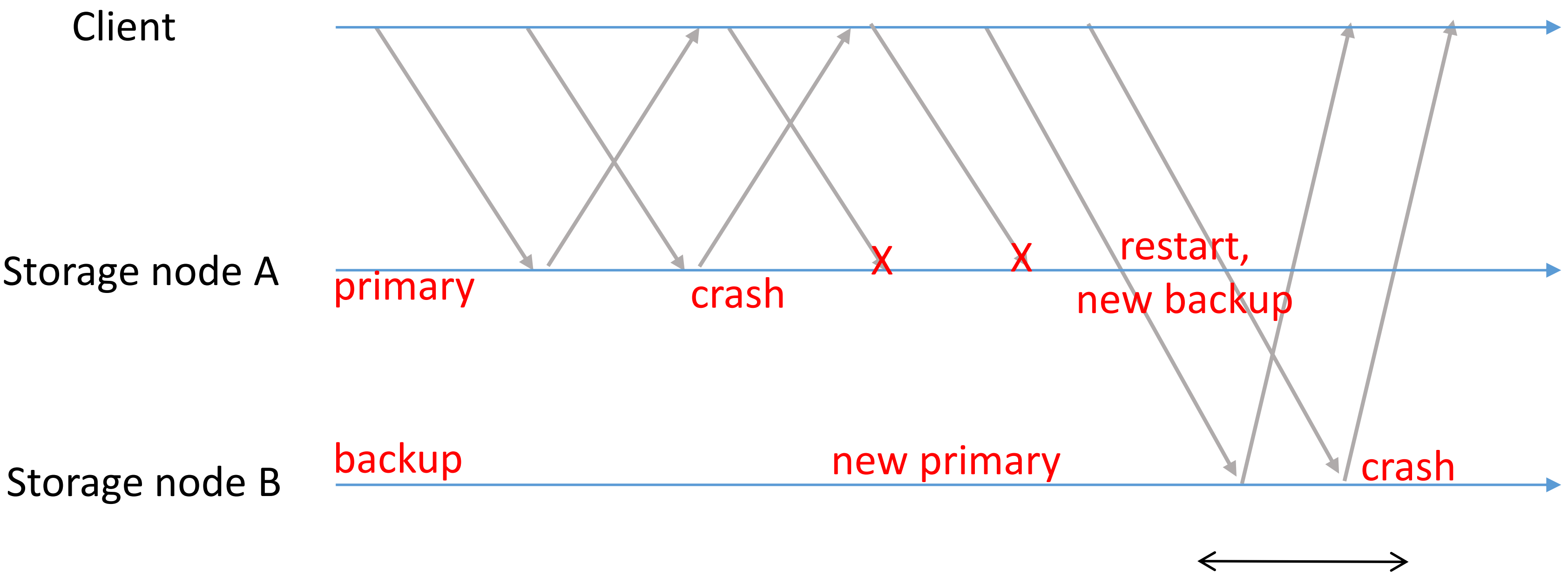
Assignment 3: Problematic Scenarios

ECE 454: Distributed Computing

Instructor: Dr. Wojciech Golab wgolab@uwaterloo.ca

Scenario #1: frequent failures



short window of opportunity to copy old state from primary to new backup

Scenario #1: frequent failures

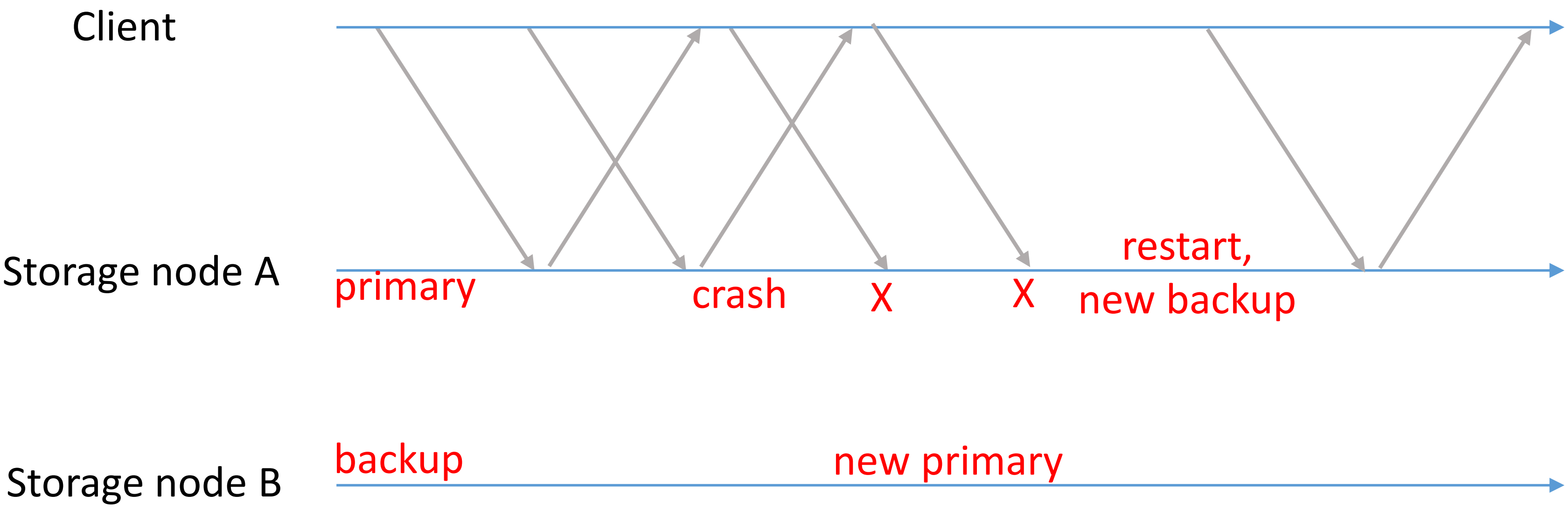
Problem:

If failures are injected every T seconds, and a new process is started T/2 seconds after each failure, then the window of opportunity for a new backup to copy old state from the primary can be as short as T/2. Moreover, during this time it is possible that ZooKeeper still records the ephemeral node of the original primary, which makes it difficult (but not impossible!) to determine the new primary.

Solution:

In the storage node code, don't assume that the first ephemeral znode correctly identifies the current primary. Use a more sophisticated algorithm to identify the primary, and copy a snapshot of the data right away without waiting for ZooKeeper to delete stale ephemeral nodes.

Scenario #2: port number reuse



Client

Storage node A

primary

crash

X

X

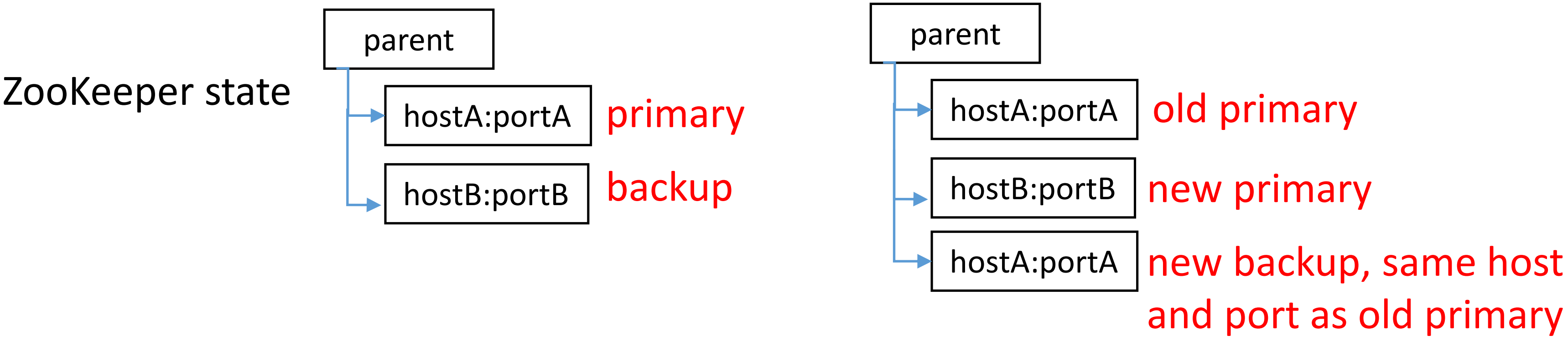
restart,

new backup

Storage node B

backup

new primary



ZooKeeper state

parent

hostA:portA

primary

hostB:portB

backup

parent

hostA:portA

old primary

hostB:portB

new primary

hostA:portA

new backup, same host and port as old primary

Scenario #2: frequent failures

Problem:

If the primary fails, its ephemeral znode may not be deleted until after the new backup has initialized and created its own znode. During this time, the client may mistake the new backup for the primary.

Solution:

If this scenario applies to your particular solution, then don't assume in the storage node that the first ephemeral znode correctly identifies the current primary. (Same advice as for scenario #1.) Also, don't let a backup storage node answer get/put RPCs from the client. Throw exceptions instead.