

Exercise 2

Set election timeout to a very small value, e.g. 5ms. Run the source code, document commands, capture screenshots, explain results

My works:

- Set election timeout to a 5ms:
 - Notice this method in **"raft.go"**:

```
// electionTimeout generates a pseudo-random election timeout duration.
func (cm *ConsensusModule) electionTimeout() time.Duration {
    // If RAFT_FORCE_MORE_REELECTION is set, stress-test by deliberately
    // generating a hard-coded number very often. This will create collisions
    // between different servers and force more re-elections.
    if len(os.Getenv("RAFT_FORCE_MORE_REELECTION")) > 0 && rand.Intn(3)
        return time.Duration(150) * time.Millisecond
    } else {
        return time.Duration(150+rand.Intn(150)) * time.Millisecond
    }
}
```

- Modify to **5 + Range(0→4) ms**

```
time.Duration(5+rand.Intn(5)) * time.Millisecond
```

1. TestElectionBasic

- Terminal command for testcase:

```
go test -v -race -run TestElectionBasic |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
08:35:31.514624 [1] becomes Candidate (currentTerm=70); log=[]
08:35:31.514679 [1] election timer started (8ms), term=70
08:35:31.514653 [2] becomes Candidate (currentTerm=70); log=[]
08:35:31.514708 [2] election timer started (9ms), term=70
08:35:31.514722 [1] sending RequestVote to 2: {Term:70 Candidateld:1 LastLogIn
08:35:31.514729 [2] sending RequestVote to 0: {Term:70 Candidateld:2 LastLogIn
08:35:31.514748 [2] sending RequestVote to 1: {Term:70 Candidateld:2 LastLogIn
08:35:31.514747 [1] sending RequestVote to 0: {Term:70 Candidateld:1 LastLogIn
08:35:31.516052 [0] RequestVote: {Term:70 Candidateld:2 LastLogIndex:0 LastL
08:35:31.516073 [0] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:35:31.516168 [2] received RequestVoteReply {Term:70 VoteGranted:false}
08:35:31.516867 [2] RequestVote: {Term:70 Candidateld:0 LastLogIndex:0 LastL
08:35:31.516887 [2] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:35:31.516991 [0] received RequestVoteReply {Term:70 VoteGranted:false}
08:35:31.518083 [1] RequestVote: {Term:70 Candidateld:2 LastLogIndex:0 LastL
08:35:31.518094 [0] RequestVote: {Term:70 Candidateld:1 LastLogIndex:0 LastL
08:35:31.518104 [1] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:35:31.518116 [0] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:35:31.518208 [1] received RequestVoteReply {Term:70 VoteGranted:false}
08:35:31.518209 [2] received RequestVoteReply {Term:70 VoteGranted:false}
    testharness.go:135: leader not found
08:35:31.519319 [0] becomes Dead
08:35:31.519370 [1] becomes Dead
08:35:31.519818 [2] becomes Dead
--- FAIL: TestElectionBasic (0.75s)
FAIL
exit status 1
FAIL github.com/elib/raft 1.039s
```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result: FAIL**

```
FAIL TestElectionBasic map[0:true 1:true 2:true] ; entries: 2103  
... Emitted file:///tmp/TestElectionBasic.html
```

```
FAIL
```

2. TestElectionLeaderDisconnect + TestElectionLeaderDisconnectThenReconnect + TestElectionLeaderDisconnectThenReconnect5

- Terminal command for testcase:

```
go test -v -race -run TestElectionLeaderDisconnect |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
...  
08:38:38.644731 [1] ... RequestVote reply: &{Term:67 VoteGranted:false}  
08:38:38.644734 [4] received RequestVoteReply {Term:67 VoteGranted:false}  
08:38:38.644780 [2] received RequestVoteReply {Term:67 VoteGranted:false}  
08:38:38.644814 [0] received RequestVoteReply {Term:67 VoteGranted:false}  
08:38:38.644838 [2] received RequestVoteReply {Term:67 VoteGranted:false}  
08:38:38.644967 [4] RequestVote: {Term:67 CandidateId:0 LastLogIndex:0 LastL  
08:38:38.644986 [4] ... RequestVote reply: &{Term:67 VoteGranted:false}  
08:38:38.645030 [4] RequestVote: {Term:67 CandidateId:3 LastLogIndex:0 LastL  
08:38:38.645055 [4] ... RequestVote reply: &{Term:67 VoteGranted:false}  
08:38:38.645136 [0] received RequestVoteReply {Term:67 VoteGranted:false}  
08:38:38.645175 [3] received RequestVoteReply {Term:67 VoteGranted:false}
```

```

testharness.go:135: leader not found
08:38:38.648761 [0] becomes Dead
08:38:38.648799 [1] becomes Dead
08:38:38.648825 [2] becomes Dead
08:38:38.648856 [3] becomes Dead
08:38:38.648875 [4] becomes Dead
08:38:38.649285 [4] in election timer state=Dead, bailing out
08:38:38.649424 [0] in election timer state=Dead, bailing out
08:38:38.649872 [3] in election timer state=Dead, bailing out
08:38:38.649907 [2] in election timer state=Dead, bailing out
08:38:38.649947 [1] in election timer state=Dead, bailing out
--- FAIL: TestElectionLeaderDisconnectThenReconnect5 (0.81s)
FAIL
exit status 1
FAIL github.com/elib/raft 2.978s

```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result: FAIL**

```

FAIL TestElectionLeaderDisconnect map[0:true 1:true 2:true] ; entries: 2104
... Emitted file:///tmp/TestElectionLeaderDisconnect.html

```

```

FAIL TestElectionLeaderDisconnectThenReconnect map[0:true 1:true 2:true] ; en
... Emitted file:///tmp/TestElectionLeaderDisconnectThenReconnect.html

```

```

FAIL TestElectionLeaderDisconnectThenReconnect5 map[0:true 1:true 2:true 3:tr
... Emitted file:///tmp/TestElectionLeaderDisconnectThenReconnect5.html

```

```

FAIL

```

3. TestElectionLeaderAndAnotherDisconnect

- Terminal command for testcase:

```
go test -v -race -run TestElectionLeaderAndAnotherDisconnect |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
...
08:42:30.370032 [0] RequestVote: {Term:70 Candidateld:1 LastLogIndex:0 LastL
08:42:30.370047 [2] RequestVote: {Term:70 Candidateld:1 LastLogIndex:0 LastL
08:42:30.370073 [0] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:42:30.370091 [2] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:42:30.370260 [1] received RequestVoteReply {Term:70 VoteGranted:false}
08:42:30.370298 [1] received RequestVoteReply {Term:70 VoteGranted:false}
08:42:30.370680 [2] RequestVote: {Term:70 Candidateld:0 LastLogIndex:0 LastL
08:42:30.370716 [2] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:42:30.370877 [0] received RequestVoteReply {Term:70 VoteGranted:false}
08:42:30.373130 [0] RequestVote: {Term:70 Candidateld:2 LastLogIndex:0 LastL
08:42:30.373163 [1] RequestVote: {Term:70 Candidateld:2 LastLogIndex:0 LastL
testharness.go:135: leader not found
08:42:30.373175 [0] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:42:30.373201 [1] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:42:30.373400 [2] received RequestVoteReply {Term:70 VoteGranted:false}
08:42:30.373447 [0] becomes Dead
08:42:30.373458 [2] received RequestVoteReply {Term:70 VoteGranted:false}
08:42:30.373527 [1] becomes Dead
08:42:30.373598 [2] becomes Dead
--- FAIL: TestElectionLeaderAndAnotherDisconnect (0.75s)
FAIL
exit status 1
FAIL github.com/elib/raft 1.037s
```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result: FAIL**

```
FAIL TestElectionLeaderAndAnotherDisconnect map[0:true 1:true 2:true] ; entries
... Emitted file:///tmp/TestElectionLeaderAndAnotherDisconnect.html
```

FAIL

4. TestDisconnectAllThenRestore

- Terminal command for testcase:

```
go test -v -race -run TestDisconnectAllThenRestore |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
...
08:46:04.053038 [2] becomes Candidate (currentTerm=121); log=[]
08:46:04.053066 [2] sending RequestVote to 1: {Term:121 CandidateId:2 LastLog
08:46:04.053093 [1] becomes Candidate (currentTerm=121); log=[]
08:46:04.053067 [2] sending RequestVote to 0: {Term:121 CandidateId:2 LastLog
08:46:04.053115 [1] sending RequestVote to 0: {Term:121 CandidateId:1 LastLogI
08:46:04.053142 [2] election timer started (5ms), term=121
08:46:04.053163 [1] election timer started (7ms), term=121
08:46:04.053161 [1] sending RequestVote to 2: {Term:121 CandidateId:1 LastLogI
08:46:04.053228 [0] becomes Candidate (currentTerm=121); log=[]
08:46:04.053299 [0] sending RequestVote to 1: {Term:121 CandidateId:0 LastLog
08:46:04.053350 [0] election timer started (5ms), term=121
08:46:04.053391 [0] sending RequestVote to 2: {Term:121 CandidateId:0 LastLog
08:46:04.054362 [0] RequestVote: {Term:121 CandidateId:1 LastLogIndex:0 LastI
08:46:04.054386 [0] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.054488 [1] received RequestVoteReply {Term:121 VoteGranted:false}
```

```

08:46:04.055336 [0] RequestVote: {Term:121 Candidateld:2 LastLogIndex:0 LastL
08:46:04.055359 [0] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.055460 [2] received RequestVoteReply {Term:121 VoteGranted:false}
08:46:04.057405 [2] RequestVote: {Term:121 Candidateld:1 LastLogIndex:0 LastL
08:46:04.057417 [1] RequestVote: {Term:121 Candidateld:0 LastLogIndex:0 LastL
08:46:04.057427 [2] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.057445 [1] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.057513 [1] RequestVote: {Term:121 Candidateld:2 LastLogIndex:0 LastL
08:46:04.057530 [0] received RequestVoteReply {Term:121 VoteGranted:false}
08:46:04.057533 [1] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.057521 [2] RequestVote: {Term:121 Candidateld:0 LastLogIndex:0 LastL
08:46:04.057625 [2] ... RequestVote reply: &{Term:121 VoteGranted:false}
08:46:04.057651 [2] received RequestVoteReply {Term:121 VoteGranted:false}
08:46:04.057606 [1] received RequestVoteReply {Term:121 VoteGranted:false}
08:46:04.057748 [0] received RequestVoteReply {Term:121 VoteGranted:false}
    testharness.go:135: leader not found
08:46:04.059262 [0] becomes Dead
08:46:04.059361 [1] becomes Dead
08:46:04.059417 [2] becomes Dead
--- FAIL: TestDisconnectAllThenRestore (1.31s)
FAIL
exit status 1
FAIL github.com/elib/raft 1.987s

```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result:** **FAIL**

```

FAIL TestDisconnectAllThenRestore map[0:true 1:true 2:true TEST:true] ; entries:
... Emitted file:///tmp/TestDisconnectAllThenRestore.html

FAIL

```

5. TestElectionFollowerComesBack

- Terminal command for testcase:

```
go test -v -race -run TestElectionFollowerComesBack |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
...
08:48:30.190362 [2] becomes Candidate (currentTerm=69); log=[]
08:48:30.190369 [0] election timer started (8ms), term=69
08:48:30.190414 [1] becomes Candidate (currentTerm=69); log=[]
08:48:30.190417 [2] sending RequestVote to 0: {Term:69 Candidateld:2 LastLogI
08:48:30.190481 [0] sending RequestVote to 2: {Term:69 Candidateld:0 LastLogI
08:48:30.190486 [1] sending RequestVote to 2: {Term:69 Candidateld:1 LastLogI
08:48:30.190428 [0] sending RequestVote to 1: {Term:69 Candidateld:0 LastLogI
08:48:30.190487 [2] sending RequestVote to 1: {Term:69 Candidateld:2 LastLogI
08:48:30.190699 [2] election timer started (5ms), term=69
08:48:30.190613 [1] election timer started (5ms), term=69
08:48:30.190868 [1] sending RequestVote to 0: {Term:69 Candidateld:1 LastLogI
08:48:30.191949 [2] RequestVote: {Term:69 Candidateld:1 LastLogIndex:0 LastL
08:48:30.192005 [1] RequestVote: {Term:69 Candidateld:0 LastLogIndex:0 LastL
08:48:30.192016 [2] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.192075 [1] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.192084 [0] RequestVote: {Term:69 Candidateld:2 LastLogIndex:0 LastL
08:48:30.192198 [0] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.192274 [1] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.192325 [0] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.192460 [2] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.195255 [1] RequestVote: {Term:69 Candidateld:2 LastLogIndex:0 LastL
08:48:30.195294 [2] RequestVote: {Term:69 Candidateld:0 LastLogIndex:0 LastL
08:48:30.195316 [1] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.195360 [2] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.195563 [2] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.195610 [0] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.196253 [0] RequestVote: {Term:69 Candidateld:1 LastLogIndex:0 LastL
```



```

08:48:30.196311 [0] ... RequestVote reply: &{Term:69 VoteGranted:false}
08:48:30.196556 [1] received RequestVoteReply {Term:69 VoteGranted:false}
08:48:30.200904 [1] becomes Candidate (currentTerm=70); log=[]
08:48:30.200930 [0] becomes Candidate (currentTerm=70); log=[]
08:48:30.200966 [1] sending RequestVote to 0: {Term:70 Candidateld:1 LastLogI
08:48:30.200980 [1] sending RequestVote to 2: {Term:70 Candidateld:1 LastLogI
08:48:30.201001 [0] sending RequestVote to 1: {Term:70 Candidateld:0 LastLogI
08:48:30.201086 [1] election timer started (6ms), term=70
08:48:30.201129 [0] sending RequestVote to 2: {Term:70 Candidateld:0 LastLogI
08:48:30.201155 [0] election timer started (7ms), term=70
08:48:30.201221 [2] becomes Candidate (currentTerm=70); log=[]
08:48:30.201306 [2] election timer started (7ms), term=70
08:48:30.201312 [2] sending RequestVote to 0: {Term:70 Candidateld:2 LastLogI
    testharness.go:135: leader not found
08:48:30.201322 [2] sending RequestVote to 1: {Term:70 Candidateld:2 LastLogI
08:48:30.201758 [0] becomes Dead
08:48:30.202467 [2] RequestVote: {Term:70 Candidateld:1 LastLogIndex:0 LastL
08:48:30.202514 [2] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:48:30.203485 [1] RequestVote: {Term:70 Candidateld:0 LastLogIndex:0 LastL
08:48:30.203533 [1] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:48:30.204847 [1] becomes Dead
08:48:30.205520 [2] RequestVote: {Term:70 Candidateld:0 LastLogIndex:0 LastL
08:48:30.205572 [2] ... RequestVote reply: &{Term:70 VoteGranted:false}
08:48:30.206942 [2] becomes Dead
08:48:30.211762 [0] in election timer state=Dead, bailing out
08:48:30.211782 [1] in election timer state=Dead, bailing out
08:48:30.211804 [2] in election timer state=Dead, bailing out
--- FAIL: TestElectionFollowerComesBack (0.82s)
FAIL
exit status 1
FAIL github.com/elib/raft 1.097s

```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result: FAIL**

```
FAIL TestElectionFollowerComesBack map[0:true 1:true 2:true] ; entries: 2100
... Emitted file:///tmp/TestElectionFollowerComesBack.html
```

FAIL

6. TestElectionDisconnectLoop

- Terminal command for testcase:

```
go test -v -race -run TestElectionDisconnectLoop |& tee ./log.txt
```

→ **Log:** The log is too large so I paste a small part of it

```
...
08:50:00.113855 [0] becomes Follower with term=65; log=[]
08:50:00.113915 [0] election timer started (8ms), term=65
08:50:00.114737 [0] AppendEntries: {Term:65 LeaderId:2 PrevLogIndex:0 PrevLo
08:50:00.114826 [0] AppendEntries reply: {Term:65 Success:true}
08:50:00.117254 [2] in election timer state=Leader, bailing out
08:50:00.117305 [1] becomes Candidate (currentTerm=66); log=[]
08:50:00.117383 [1] election timer started (8ms), term=66
08:50:00.117416 [1] sending RequestVote to 2: {Term:66 Candidateld:1 LastLogIn
08:50:00.117415 [1] sending RequestVote to 0: {Term:66 Candidateld:1 LastLogIn
08:50:00.117698 [1] AppendEntries: {Term:65 LeaderId:2 PrevLogIndex:0 PrevLo
08:50:00.117783 [1] AppendEntries reply: {Term:66 Success:false}
08:50:00.117994 [2] term out of date in heartbeat reply
08:50:00.118025 [2] becomes Follower with term=66; log=[]
08:50:00.118078 [2] election timer started (5ms), term=66
08:50:00.118908 [2] RequestVote: {Term:66 Candidateld:1 LastLogIndex:0 LastLc
08:50:00.118967 [2] ... RequestVote reply: &{Term:66 VoteGranted:true}
08:50:00.119224 [1] received RequestVoteReply {Term:66 VoteGranted:true}
08:50:00.119274 [1] wins election with 2 votes
08:50:00.119305 [1] becomes Leader; term=66, log=[]
```

```

08:50:00.119366 [1] sending AppendEntries to 2: ni=0, args={Term:66 LeaderId:1
08:50:00.119373 [1] sending AppendEntries to 0: ni=0, args={Term:66 LeaderId:1
08:50:00.119965 [0] RequestVote: {Term:66 CandidateId:1 LastLogIndex:0 LastLo
08:50:00.120011 [0] ... term out of date in RequestVote
08:50:00.120044 [0] becomes Follower with term=66; log=[]
08:50:00.120083 [0] ... RequestVote reply: &{Term:66 VoteGranted:true}
    testharness.go:135: leader not found
08:50:00.120186 [0] election timer started (9ms), term=66
08:50:00.120352 [1] received RequestVoteReply {Term:66 VoteGranted:true}
08:50:00.120388 [1] while waiting for reply, state = Leader
08:50:00.120452 [0] becomes Dead
08:50:00.120783 [1] becomes Dead
08:50:00.120861 [2] becomes Dead
08:50:00.121032 [0] in election timer state=Dead, bailing out
08:50:00.124005 [0] in election timer state=Dead, bailing out
08:50:00.127589 [1] in election timer state=Dead, bailing out
08:50:00.128216 [2] in election timer state=Dead, bailing out
08:50:00.130545 [0] in election timer state=Dead, bailing out
--- FAIL: TestElectionDisconnectLoop (0.81s)
FAIL
exit status 1
FAIL github.com/elib/raft 1.091s

```

- Terminal command for HTML visualization:

```
Get-Content ./log.txt | go run ../tools/raft-testlog-viz/main.go
```

→ **Result: FAIL**

```

FAIL TestElectionDisconnectLoop map[0:true 1:true 2:true] ; entries: 1999
... Emitted file:///tmp/TestElectionDisconnectLoop.html

FAIL

```

7. Results

Test case	Result
TestElectionBasic	FAIL
TestElectionLeaderDisconnect	FAIL
TestElectionLeaderDisconnectThenReconnect	FAIL
TestElectionLeaderDisconnectThenReconnect5	FAIL
TestElectionLeaderAndAnotherDisconnect	FAIL
TestDisconnectAllThenRestore	FAIL
TestElectionFollowerComesBack	FAIL
TestElectionDisconnectLoop	FAIL

→ Explain Result:

Because the **election timeout is too short** (e.g., 5ms) compared to the travel time of heartbeats between nodes, initial **Followers** do not receive heartbeats in time. As a result, they all become **Candidates** simultaneously, leading to frequent and continuous elections. Due to the short election timeout, multiple re-elections occur before a **Leader** is successfully elected. Even after a **Leader** is chosen, some **Followers** may still transition to **Candidates** because they do not receive heartbeats before their election timeout expires. This repeated cycle can cause a **Candidate** to start a new election with a higher term, potentially replacing the current **Leader**.