Assignment 3 – Paxos

**Team:**

Livar Cunha

Luciano Mandryk

Sergio Clemente

# Introduction

For this assignment we chose to build up on assignment 1 and improve the reliability of the Storage Server by using Paxos to coordinate a number of replicas and have them agree on a total order of execution for commands sent to any storage server node. The choice of improving the storage server instead of the facebook server was mostly arbitrary; we could just as easily have used the facebook server (or both) by changing a single line of code that delivers the chosen Paxos value for each slot to a particular component for execution.

# Compiling and Executing

Just run ./compile.sh or ./execute.sh

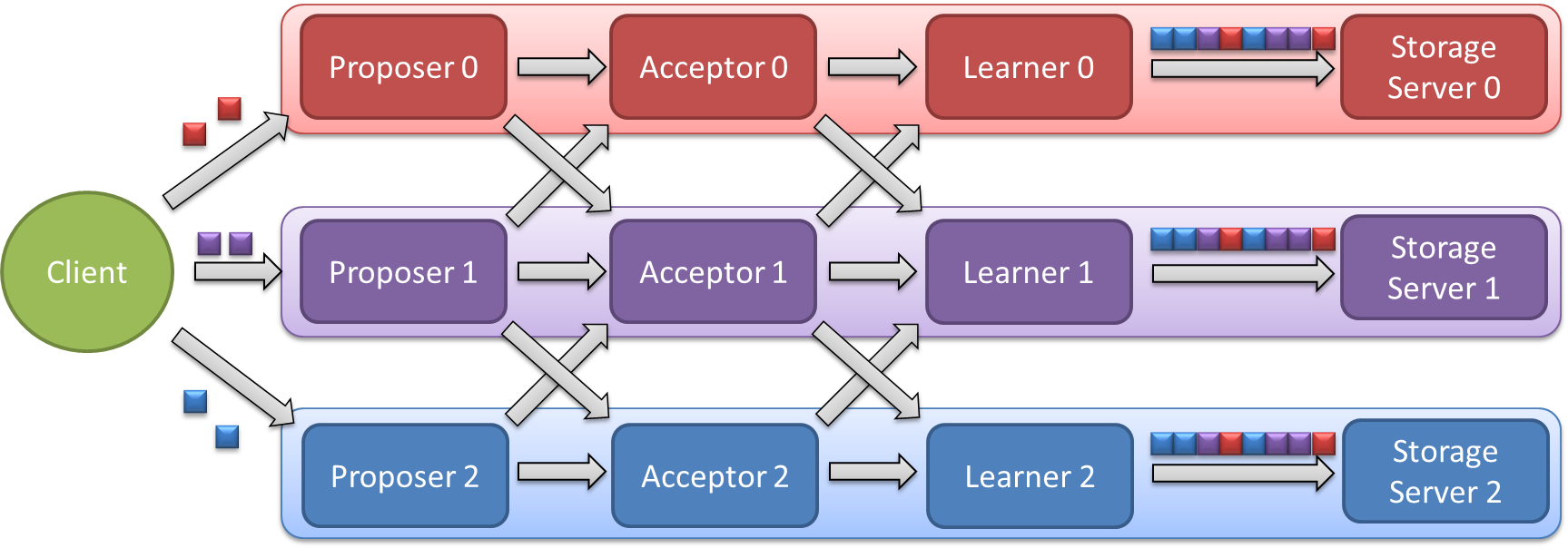
If you run into any issues make sure you set your java home to something like the following:

[~] JAVA\_HOME=/Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents/Home/bin

[~] export JAVA\_HOME

Also, make sure the classes folder exist.

# High-Level Design



We chose to build a 3-node system (nodes 0, 1, 2) where each node implements the roles of proposer, acceptor, and learner, as well as the storage server. This is not a hard requirement, it was chosen for simplicity and debugability. The Paxos roles can be easily separated, mixed and matched across many different nodes by editing the address arrays at the top of src/node/paxos/PaxosNode.java.

The roles are what one would expect from Paxos, with nodes being able to send messages to all other nodes (some arrows are purposely missing in the diagram to keep it readable).

### Commands:

The client sends any number of commands to any of the proposers via the execute\_command command. For example:

0 execute\_command create login\_names.txt

Commands sent to the proposers are kept in a per-proposer queue. Since storage server commands may have dependencies (e.g. a command to append data to a file must not be reordered and executed before the command that created the file), in order to preserve the serialization, commands sent to the same proposer are negotiated (and therefore executed) in the order they are received (FIFO). We make no guarantees as to the order of execution of commands sent to different proposers. One consequence of this design decision is that each proposer will negotiate a slot for only one command at a time ( is 1), although every proposer may be proposing a value at the same time.

### Proposers:

Upon receiving a command, the proposer starts a Paxos round for the next available slot. Other proposers may be proposing other commands for the same slot, and eventually the algorithm reaches a consensus on which command should be associated with that particular slot. Along with the command, the Paxos value also contains the identity of the proposer, making the {proposer\_id,command} tuple unique even when multiple proposers are proposing the same command for the same slot. This also prevents a bug where if multiple proposers propose the same command at the same time, the command would be chosen and executed only once despite the intent of it being executed as many times as it was proposed by different proposers.

Once a value has been chosen for that particular slot, the proposer learns the chosen value by tracking the accept responses from the acceptors. If the chosen value was the command that it proposed, the proposer moves on to the next slot and proposes the next command in its queue, if there is any. If a different command was chosen for that slot, the proposer initiates a Paxos round for the next available slot, repeating this process as many times as necessary until its command is chosen.

If the proposal round fails (i.e. the proposer doesn’t receive enough responses by some timeout period or a majority of acceptors did not promise to accept the proposal), the proposer backs out for a random number of ticks before retrying the proposal. The goal with the random back out is to perturb the determinism of the messaging layer and allow one of the competing proposers to eventually win the round.

### Acceptors and Learners:

Acceptors and learners behave like any vanilla Paxos implementation. Upon accepting a value (obviously subject to the conditions imposed by the algorithm), the acceptor notifies all learns of the accepted value. The learners learn that a value has been chosen when it detects that a majority of acceptors accepted the same proposal.

Once a learner learns that a value has been chosen, it stores the chosen command in a data structure. If there are no gaps in the slots, the command is passed to the storage server component, which executes it. If there happened to be gaps, the learner waits until the gap is filled before passing the commands to the storage server component.

The proposers, acceptors and learners properly persist the necessary state to ensure that slot numbers, chosen values and accepted proposal numbers are never forgotten even in case of total failure. Additionally, learners persist their state before passing the commands to the storage server component in order to maintain that at-most-once semantics that was required in assignment 1.

Because of Paxos, all learners learn the same values for the same slots, so eventually all storage servers end up executing the same commands in the same other, therefore achieving the same state.

# Examples

Here we show a series of execution scenarios and their outcomes. For brevity, we only show the executed commands and the final result produced. The complete output of each scenario containing all the relevant message traffic and debug spew is included at the end of this document and can be used to verify the correctness of the execution.

### Scenario 1: consensus over a single command

In this scenario we send a command to create a file called scenario1\_file.txt to one of the proposers. All nodes (proposers, acceptors, learners) are online and there is no competition for slots, so this is Paxos at its best case scenario. The expectation is that all learners will learn and execute the command. We use the dump\_values command to have each node display its view of the commands chosen for each slot.

The sequence of commands provided to the program is:

**# initialize the nodes**

**start 0**

**start 1**

**start 2**

**# tell node 0 to propose the command “create scenario1\_file.txt”**

**0 execute\_command create scenario1\_file.txt**

**# after consensus is reached, ask each node to dump the chosen values**

**0 dump\_values**

**1 dump\_values**

**2 dump\_values**

The relevant output snippets are shown below:

Time: 0

Please input a sequence of commands terminated by a blank line or the TIME command:

**start 0**

**start 1**

**start 2**

0: START 0

1: START 1

2: START 2

In what order should the events happen? (enter for in-order)

[...]

Time: 1

Please input a sequence of commands terminated by a blank line or the TIME command:

**0 execute\_command create scenario1\_file.txt**

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: COMMAND 0 executes execute\_command create scenario1\_file.txt

[...]

Time: 21

Please input a sequence of commands terminated by a blank line or the TIME command:

**0 dump\_values**

**1 dump\_values**

**2 dump\_values**

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: COMMAND 0 executes dump\_values

1: COMMAND 1 executes dump\_values

2: COMMAND 2 executes dump\_values

In what order should the events happen? (enter for in-order)

**\*\*\* 0: PAXOS: node 0: slot 0: 0.create scenario1\_file.txt**

**\*\*\* 1: PAXOS: node 1: slot 0: 0.create scenario1\_file.txt**

**\*\*\* 2: PAXOS: node 2: slot 0: 0.create scenario1\_file.txt**

Note in the output above that all nodes (0, 1, 2) reached consensus on the command to run for slot 0. The chosen command was “**create scenario1\_file.txt**”, which was proposed by proposer 0, as indicated by the “**0.**” prefix.

The following output shows that the command has been executed and the file has been created in the storage area of each node:

$ ls -FalR .

.:

total 4

drwxr-xr-x 5 Luciano Administ 0 Jun 6 01:00 ./

drwxr-xr-x 11 Luciano Administ 8192 Jun 6 01:00 ../

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 0/

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 1/

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 2/

./0:

total 1

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 ./

drwxr-xr-x 5 Luciano Administ 0 Jun 6 01:00 ../

-rw-r--r-- 1 Luciano Administ 290 Jun 6 01:00 acceptor\_state.txt

-rw-r--r-- 1 Luciano Administ 168 Jun 6 01:01 node.txt

-rw-r--r-- 1 Luciano Administ 0 Jun 6 01:01 scenario1\_file.txt

./1:

total 1

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 ./

drwxr-xr-x 5 Luciano Administ 0 Jun 6 01:00 ../

-rw-r--r-- 1 Luciano Administ 290 Jun 6 01:00 acceptor\_state.txt

-rw-r--r-- 1 Luciano Administ 168 Jun 6 01:01 node.txt

-rw-r--r-- 1 Luciano Administ 0 Jun 6 01:01 scenario1\_file.txt

./2:

total 1

drwxr-xr-x 2 Luciano Administ 0 Jun 6 01:01 ./

drwxr-xr-x 5 Luciano Administ 0 Jun 6 01:00 ../

-rw-r--r-- 1 Luciano Administ 290 Jun 6 01:00 acceptor\_state.txt

-rw-r--r-- 1 Luciano Administ 168 Jun 6 01:01 node.txt

-rw-r--r-- 1 Luciano Administ 0 Jun 6 01:01 scenario1\_file.txt

The full output of this scenario execution is included at the end of this document.

### Scenario 2: multiple commands

### Scenario 3:

# Appendix

### Scenario 1: full output

Starting simulation with seed: 1338973184225

Warning: unable to open logfile '' for writing.

...continuing

Warning: unable to open logfile '' for writing.

...continuing

Time: 0

Please input a sequence of commands terminated by a blank line or the TIME command:

0: START 0

1: START 1

2: START 2

In what order should the events happen? (enter for in-order)

\*\*\* 0: PAXOS: Starting acceptor node on address 0

\*\*\* 0: PAXOS: connecting to learner address 0

\*\*\* 0: PAXOS: connecting to learner address 1

\*\*\* 0: PAXOS: connecting to learner address 2

\*\*\* 0: PAXOS: Starting proposer node on address 0

\*\*\* 0: PAXOS: connecting to acceptor address 0

\*\*\* 0: PAXOS: connecting to acceptor address 1

\*\*\* 0: PAXOS: connecting to acceptor address 2

\*\*\* 0: PAXOS: Starting learner node on address 0

\*\*\* 0: PAXOS: connecting to acceptor address 0

\*\*\* 0: PAXOS: connecting to acceptor address 1

\*\*\* 0: PAXOS: connecting to acceptor address 2

\*\*\* 0: PAXOS: Starting storage system on address 0

\*\*\* 1: PAXOS: Starting acceptor node on address 1

\*\*\* 1: PAXOS: connecting to learner address 0

\*\*\* 1: PAXOS: connecting to learner address 1

\*\*\* 1: PAXOS: connecting to learner address 2

\*\*\* 1: PAXOS: Starting proposer node on address 1

\*\*\* 1: PAXOS: connecting to acceptor address 0

\*\*\* 1: PAXOS: connecting to acceptor address 1

\*\*\* 1: PAXOS: connecting to acceptor address 2

\*\*\* 1: PAXOS: Starting learner node on address 1

\*\*\* 1: PAXOS: connecting to acceptor address 0

\*\*\* 1: PAXOS: connecting to acceptor address 1

\*\*\* 1: PAXOS: connecting to acceptor address 2

\*\*\* 1: PAXOS: Starting storage system on address 1

\*\*\* 2: PAXOS: Starting acceptor node on address 2

\*\*\* 2: PAXOS: connecting to learner address 0

\*\*\* 2: PAXOS: connecting to learner address 1

\*\*\* 2: PAXOS: connecting to learner address 2

\*\*\* 2: PAXOS: Starting proposer node on address 2

\*\*\* 2: PAXOS: connecting to acceptor address 0

\*\*\* 2: PAXOS: connecting to acceptor address 1

\*\*\* 2: PAXOS: connecting to acceptor address 2

\*\*\* 2: PAXOS: Starting learner node on address 2

\*\*\* 2: PAXOS: connecting to acceptor address 0

\*\*\* 2: PAXOS: connecting to acceptor address 1

\*\*\* 2: PAXOS: connecting to acceptor address 2

\*\*\* 2: PAXOS: Starting storage system on address 2

Time: 1

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: COMMAND 0 executes execute\_command create scenario1\_file.txt

In what order should the events happen? (enter for in-order)

\*\*\* 0: PAXOS: prepare() on slot 0

Time: 2

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->1 protocol: 1 contents: ?L?

2: Packet: 0->2 protocol: 1 contents: ??oD

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->1 protocol: 1 contents: ?L?

2: DELIVERY Packet: 0->2 protocol: 1 contents: ??oD

In what order should the events happen? (enter for in-order)

Time: 3

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 1->0 protocol: 1 contents: ?L?

2: Packet: 2->0 protocol: 1 contents: ??oD

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 1->0 protocol: 1 contents: ?L?

2: DELIVERY Packet: 2->0 protocol: 1 contents: ??oD

In what order should the events happen? (enter for in-order)

Time: 4

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ???? 7 prepare 10 -698351615 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

1: Packet: 0->1 protocol: 1 contents: ?L? 7 prepare 10 -698351614 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

2: Packet: 0->2 protocol: 1 contents: ??oD 7 prepare 10 -698351613 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 7 prepare 10 -698351615 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

1: DELIVERY Packet: 0->1 protocol: 1 contents: ?L? 7 prepare 10 -698351614 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

2: DELIVERY Packet: 0->2 protocol: 1 contents: ??oD 7 prepare 10 -698351613 55 {

"slotNumber": 0,

"number": {

"value": 0

}

}

3: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1b9f88b) at 4

4: TIMEOUT 0: onTimeout([Ljava.lang.Object;@6c9ec6) at 4

5: TIMEOUT 0: onTimeout([Ljava.lang.Object;@12beeec) at 4

In what order should the events happen? (enter for in-order)

Crash node 0 before creation of acceptor\_state.txt? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 92)? (y/n)

Crash node 1 before creation of acceptor\_state.txt? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 92)? (y/n)

Crash node 2 before creation of acceptor\_state.txt? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 92)? (y/n)

Time: 5

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->0 protocol: 1 contents: ???? 13 reply\_prepare 10 -698351615 1 0 132 {

"hostIdentifier": 0,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

2: Packet: 1->0 protocol: 1 contents: ?L?

3: Packet: 1->0 protocol: 1 contents: ?

4: Packet: 2->0 protocol: 1 contents: ??oD

5: Packet: 2->0 protocol: 1 contents: ???

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 13 reply\_prepare 10 -698351615 1 0 132 {

"hostIdentifier": 0,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

2: DELIVERY Packet: 1->0 protocol: 1 contents: ?L?

3: DELIVERY Packet: 1->0 protocol: 1 contents: ?

4: DELIVERY Packet: 2->0 protocol: 1 contents: ??oD

5: DELIVERY Packet: 2->0 protocol: 1 contents: ???

In what order should the events happen? (enter for in-order)

Time: 6

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->1 protocol: 1 contents: ?

2: Packet: 0->2 protocol: 1 contents: ???

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->1 protocol: 1 contents: ?

2: DELIVERY Packet: 0->2 protocol: 1 contents: ???

3: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1353154) at 6

4: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1398cac) at 6

5: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1429cf1) at 6

In what order should the events happen? (enter for in-order)

Time: 7

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 1->0 protocol: 1 contents: ? 13 reply\_prepare 10 -698351614 1 0 132 {

"hostIdentifier": 1,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

1: Packet: 2->0 protocol: 1 contents: ???  
 13 reply\_prepare 10 -698351613 1 0 132 {

"hostIdentifier": 2,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 1->0 protocol: 1 contents: ? 13 reply\_prepare 10 -698351614 1 0 132 {

"hostIdentifier": 1,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

1: DELIVERY Packet: 2->0 protocol: 1 contents: ???  
 13 reply\_prepare 10 -698351613 1 0 132 {

"hostIdentifier": 2,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"promised": true

}

2: TIMEOUT 0: onTimeout([Ljava.lang.Object;@39452f) at 7

3: TIMEOUT 1: onTimeout([Ljava.lang.Object;@1ed13da) at 7

4: TIMEOUT 2: onTimeout([Ljava.lang.Object;@1b25a82) at 7

In what order should the events happen? (enter for in-order)

\*\*\* 0: PAXOS: accept() on slot 0

Time: 8

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->1 protocol: 1 contents: ?

1: Packet: 0->0 protocol: 1 contents: ???? 6 accept 10 -698351612 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

2: Packet: 0->1 protocol: 1 contents: ?L? 6 accept 10 -698351611 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

3: Packet: 0->2 protocol: 1 contents: ??oD 6 accept 10 -698351610 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

4: Packet: 0->2 protocol: 1 contents: ???

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->1 protocol: 1 contents: ?

1: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 6 accept 10 -698351612 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

2: DELIVERY Packet: 0->1 protocol: 1 contents: ?L? 6 accept 10 -698351611 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

3: DELIVERY Packet: 0->2 protocol: 1 contents: ??oD 6 accept 10 -698351610 169 {

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

4: DELIVERY Packet: 0->2 protocol: 1 contents: ???

In what order should the events happen? (enter for in-order)

Crash node 0 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 93)? (y/n)

Crash node 0 before creation of acceptor\_state.txt? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 216)? (y/n)

Crash node 0 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 0 before delete ofacceptor\_state.txt.tmp? (y/n)

Crash node 0 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 217)? (y/n)

Crash node 0 before creation of acceptor\_state.txt? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 290)? (y/n)

Crash node 0 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 0 before delete ofacceptor\_state.txt.tmp? (y/n)

\*\*\* 0: PAXOS: learn() on slot 0

Crash node 1 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 93)? (y/n)

Crash node 1 before creation of acceptor\_state.txt? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 216)? (y/n)

Crash node 1 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 1 before delete ofacceptor\_state.txt.tmp? (y/n)

Crash node 1 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 217)? (y/n)

Crash node 1 before creation of acceptor\_state.txt? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 290)? (y/n)

Crash node 1 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 1 before delete ofacceptor\_state.txt.tmp? (y/n)

\*\*\* 1: PAXOS: learn() on slot 0

Crash node 2 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 93)? (y/n)

Crash node 2 before creation of acceptor\_state.txt? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 216)? (y/n)

Crash node 2 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 2 before delete ofacceptor\_state.txt.tmp? (y/n)

Crash node 2 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 217)? (y/n)

Crash node 2 before creation of acceptor\_state.txt? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 290)? (y/n)

Crash node 2 before creation of acceptor\_state.txt.tmp? (y/n)

Crash node 2 before delete ofacceptor\_state.txt.tmp? (y/n)

\*\*\* 2: PAXOS: learn() on slot 0

Time: 9

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->0 protocol: 1 contents: ???? 5 learn 10 -698351609 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

2: Packet: 0->1 protocol: 1 contents: ?L? 5 learn 10 -698351608 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

3: Packet: 0->2 protocol: 1 contents: ??oD 5 learn 10 -698351607 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

4: Packet: 0->0 protocol: 1 contents: ???? 12 reply\_accept 10 -698351612 1 0 262 {

"hostIdentifier": 0,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

5: Packet: 1->0 protocol: 1 contents: ?L?

6: Packet: 1->0 protocol: 1 contents: ? 5 learn 10 -698351606 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

7: Packet: 1->1 protocol: 1 contents: ??%

8: Packet: 1->2 protocol: 1 contents: =??

9: Packet: 1->0 protocol: 1 contents: ? 12 reply\_accept 10 -698351611 1 0 262 {

"hostIdentifier": 1,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

10: Packet: 2->0 protocol: 1 contents: ??oD

11: Packet: 2->0 protocol: 1 contents: ???  
 5 learn 10 -698351603 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

12: Packet: 2->1 protocol: 1 contents: ????

13: Packet: 2->2 protocol: 1 contents: Y[??

14: Packet: 2->0 protocol: 1 contents: ???  
 12 reply\_accept 10 -698351610 1 0 262 {

"hostIdentifier": 2,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 5 learn 10 -698351609 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

2: DELIVERY Packet: 0->1 protocol: 1 contents: ?L? 5 learn 10 -698351608 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

3: DELIVERY Packet: 0->2 protocol: 1 contents: ??oD 5 learn 10 -698351607 219 {

"slotNumber": 0,

"hostIdentifier": 0,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

4: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 12 reply\_accept 10 -698351612 1 0 262 {

"hostIdentifier": 0,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

5: DELIVERY Packet: 1->0 protocol: 1 contents: ?L?

6: DELIVERY Packet: 1->0 protocol: 1 contents: ? 5 learn 10 -698351606 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

7: DELIVERY Packet: 1->1 protocol: 1 contents: ??%

8: DELIVERY Packet: 1->2 protocol: 1 contents: =??

9: DELIVERY Packet: 1->0 protocol: 1 contents: ? 12 reply\_accept 10 -698351611 1 0 262 {

"hostIdentifier": 1,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

10: DELIVERY Packet: 2->0 protocol: 1 contents: ??oD

11: DELIVERY Packet: 2->0 protocol: 1 contents: ???  
 5 learn 10 -698351603 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

12: DELIVERY Packet: 2->1 protocol: 1 contents: ????

13: DELIVERY Packet: 2->2 protocol: 1 contents: Y[??

14: DELIVERY Packet: 2->0 protocol: 1 contents: ???  
 12 reply\_accept 10 -698351610 1 0 262 {

"hostIdentifier": 2,

"prepareRequest": {

"slotNumber": 0,

"number": {

"value": 0

}

},

"acceptedProposalNumber": {

"value": 0

},

"accepted": true,

"value": {

"command": "create scenario1\_file.txt",

"proposer": 0

}

}

15: TIMEOUT 1: onTimeout([Ljava.lang.Object;@2982d8) at 9

16: TIMEOUT 2: onTimeout([Ljava.lang.Object;@14d55de) at 9

In what order should the events happen? (enter for in-order)

##### L0: Value NOT chosen yet

##### L1: Value NOT chosen yet

##### L2: Value NOT chosen yet

##### L0: Value was chosen: (0,create scenario1\_file.txt)

Crash node 0 before creation of node.txt? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 136)? (y/n)

Crash node 0 before creation of node.txt.tmp? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 137)? (y/n)

Crash node 0 before creation of node.txt? (y/n)

Crash node 0 before write(str)? (y/n)

Crash node 0 before write(s, 0, 168)? (y/n)

Crash node 0 before creation of node.txt.tmp? (y/n)

Crash node 0 before delete ofnode.txt.tmp? (y/n)

\*\*\* 0: PAXOS: S0: executing chosen command: 0.create scenario1\_file.txt

Crash node 0 before creation of scenario1\_file.txt? (y/n)

Time: 10

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->0 protocol: 1 contents: ???? 11 reply\_learn 10 -698351609 1 0 6 <null>

2: Packet: 1->0 protocol: 1 contents: ?L?

3: Packet: 1->0 protocol: 1 contents: ? 11 reply\_learn 10 -698351608 1 0 6 <null>

4: Packet: 2->0 protocol: 1 contents: ??oD

5: Packet: 2->0 protocol: 1 contents: ???  
 11 reply\_learn 10 -698351607 1 0 6 <null>

6: Packet: 0->0 protocol: 1 contents: ????

7: Packet: 0->1 protocol: 1 contents: ?

8: Packet: 0->1 protocol: 1 contents: ?L? 11 reply\_learn 10 -698351606 1 0 6 <null>

9: Packet: 1->1 protocol: 1 contents: ??%

10: Packet: 2->1 protocol: 1 contents: =??

11: Packet: 0->1 protocol: 1 contents: ?

12: Packet: 0->2 protocol: 1 contents: ???

13: Packet: 0->2 protocol: 1 contents: ??oD 11 reply\_learn 10 -698351603 1 0 6 <null>

14: Packet: 1->2 protocol: 1 contents: ????

15: Packet: 2->2 protocol: 1 contents: Y[??

16: Packet: 0->2 protocol: 1 contents: ???

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->0 protocol: 1 contents: ???? 11 reply\_learn 10 -698351609 1 0 6 <null>

2: DELIVERY Packet: 1->0 protocol: 1 contents: ?L?

3: DELIVERY Packet: 1->0 protocol: 1 contents: ? 11 reply\_learn 10 -698351608 1 0 6 <null>

4: DELIVERY Packet: 2->0 protocol: 1 contents: ??oD

5: DELIVERY Packet: 2->0 protocol: 1 contents: ???  
 11 reply\_learn 10 -698351607 1 0 6 <null>

6: DELIVERY Packet: 0->0 protocol: 1 contents: ????

7: DELIVERY Packet: 0->1 protocol: 1 contents: ?

8: DELIVERY Packet: 0->1 protocol: 1 contents: ?L? 11 reply\_learn 10 -698351606 1 0 6 <null>

9: DELIVERY Packet: 1->1 protocol: 1 contents: ??%

10: DELIVERY Packet: 2->1 protocol: 1 contents: =??

11: DELIVERY Packet: 0->1 protocol: 1 contents: ?

12: DELIVERY Packet: 0->2 protocol: 1 contents: ???

13: DELIVERY Packet: 0->2 protocol: 1 contents: ??oD 11 reply\_learn 10 -698351603 1 0 6 <null>

14: DELIVERY Packet: 1->2 protocol: 1 contents: ????

15: DELIVERY Packet: 2->2 protocol: 1 contents: Y[??

16: DELIVERY Packet: 0->2 protocol: 1 contents: ???

17: TIMEOUT 0: onPrepareTimeout([Ljava.lang.Object;@4f8358) at 10

18: TIMEOUT 0: onTimeout([Ljava.lang.Object;@38462a) at 10

19: TIMEOUT 0: onTimeout([Ljava.lang.Object;@869470) at 10

20: TIMEOUT 0: onTimeout([Ljava.lang.Object;@17b79a6) at 10

In what order should the events happen? (enter for in-order)

Time: 11

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 0->0 protocol: 1 contents: ????

1: Packet: 0->1 protocol: 1 contents: ?

2: Packet: 0->2 protocol: 1 contents: ???

3: Packet: 1->0 protocol: 1 contents: ?L?

4: Packet: 1->1 protocol: 1 contents: ??% 5 learn 10 -698351605 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

5: Packet: 1->2 protocol: 1 contents: =?? 5 learn 10 -698351604 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

6: Packet: 2->0 protocol: 1 contents: ??oD

7: Packet: 2->1 protocol: 1 contents: ???? 5 learn 10 -698351602 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

8: Packet: 2->2 protocol: 1 contents: Y[?? 5 learn 10 -698351601 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 0->0 protocol: 1 contents: ????

1: DELIVERY Packet: 0->1 protocol: 1 contents: ?

2: DELIVERY Packet: 0->2 protocol: 1 contents: ???

3: DELIVERY Packet: 1->0 protocol: 1 contents: ?L?

4: DELIVERY Packet: 1->1 protocol: 1 contents: ??% 5 learn 10 -698351605 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

5: DELIVERY Packet: 1->2 protocol: 1 contents: =?? 5 learn 10 -698351604 219 {

"slotNumber": 0,

"hostIdentifier": 1,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

6: DELIVERY Packet: 2->0 protocol: 1 contents: ??oD

7: DELIVERY Packet: 2->1 protocol: 1 contents: ???? 5 learn 10 -698351602 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

8: DELIVERY Packet: 2->2 protocol: 1 contents: Y[?? 5 learn 10 -698351601 219 {

"slotNumber": 0,

"hostIdentifier": 2,

"learnedValue": {

"slotNumber": 0,

"content": {

"command": "create scenario1\_file.txt",

"proposer": 0

},

"number": {

"value": 0

}

}

}

9: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1d47f59) at 11

10: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1f4cdd2) at 11

11: TIMEOUT 0: onTimeout([Ljava.lang.Object;@f4ca49) at 11

12: TIMEOUT 0: onTimeout([Ljava.lang.Object;@10c94a7) at 11

13: TIMEOUT 1: onTimeout([Ljava.lang.Object;@f3770c) at 11

14: TIMEOUT 1: onTimeout([Ljava.lang.Object;@cbf92) at 11

15: TIMEOUT 1: onTimeout([Ljava.lang.Object;@f549de) at 11

16: TIMEOUT 1: onTimeout([Ljava.lang.Object;@1c4a2d3) at 11

17: TIMEOUT 2: onTimeout([Ljava.lang.Object;@9e0c2d) at 11

18: TIMEOUT 2: onTimeout([Ljava.lang.Object;@318293) at 11

19: TIMEOUT 2: onTimeout([Ljava.lang.Object;@739aa3) at 11

20: TIMEOUT 2: onTimeout([Ljava.lang.Object;@1ad6c98) at 11

In what order should the events happen? (enter for in-order)

##### L1: Value was chosen: (0,create scenario1\_file.txt)

Crash node 1 before creation of node.txt? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 136)? (y/n)

Crash node 1 before creation of node.txt.tmp? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 137)? (y/n)

Crash node 1 before creation of node.txt? (y/n)

Crash node 1 before write(str)? (y/n)

Crash node 1 before write(s, 0, 168)? (y/n)

Crash node 1 before creation of node.txt.tmp? (y/n)

Crash node 1 before delete ofnode.txt.tmp? (y/n)

\*\*\* 1: PAXOS: S1: executing chosen command: 0.create scenario1\_file.txt

Crash node 1 before creation of scenario1\_file.txt? (y/n)

##### L2: Value was chosen: (0,create scenario1\_file.txt)

Crash node 2 before creation of node.txt? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 136)? (y/n)

Crash node 2 before creation of node.txt.tmp? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 137)? (y/n)

Crash node 2 before creation of node.txt? (y/n)

Crash node 2 before write(str)? (y/n)

Crash node 2 before write(s, 0, 168)? (y/n)

Crash node 2 before creation of node.txt.tmp? (y/n)

Crash node 2 before delete ofnode.txt.tmp? (y/n)

\*\*\* 2: PAXOS: S2: executing chosen command: 0.create scenario1\_file.txt

Crash node 2 before creation of scenario1\_file.txt? (y/n)

Time: 12

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 1->1 protocol: 1 contents: ??%

1: Packet: 1->1 protocol: 1 contents: ??% 11 reply\_learn 10 -698351605 1 0 6 <null>

2: Packet: 2->1 protocol: 1 contents: =??

3: Packet: 2->1 protocol: 1 contents: ???? 11 reply\_learn 10 -698351604 1 0 6 <null>

4: Packet: 1->2 protocol: 1 contents: ????

5: Packet: 1->2 protocol: 1 contents: =?? 11 reply\_learn 10 -698351602 1 0 6 <null>

6: Packet: 2->2 protocol: 1 contents: Y[??

7: Packet: 2->2 protocol: 1 contents: Y[?? 11 reply\_learn 10 -698351601 1 0 6 <null>

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 1->1 protocol: 1 contents: ??%

1: DELIVERY Packet: 1->1 protocol: 1 contents: ??% 11 reply\_learn 10 -698351605 1 0 6 <null>

2: DELIVERY Packet: 2->1 protocol: 1 contents: =??

3: DELIVERY Packet: 2->1 protocol: 1 contents: ???? 11 reply\_learn 10 -698351604 1 0 6 <null>

4: DELIVERY Packet: 1->2 protocol: 1 contents: ????

5: DELIVERY Packet: 1->2 protocol: 1 contents: =?? 11 reply\_learn 10 -698351602 1 0 6 <null>

6: DELIVERY Packet: 2->2 protocol: 1 contents: Y[??

7: DELIVERY Packet: 2->2 protocol: 1 contents: Y[?? 11 reply\_learn 10 -698351601 1 0 6 <null>

8: TIMEOUT 0: onTimeout([Ljava.lang.Object;@188f506) at 12

9: TIMEOUT 1: onTimeout([Ljava.lang.Object;@8e85b5) at 12

10: TIMEOUT 2: onTimeout([Ljava.lang.Object;@ec42e0) at 12

11: TIMEOUT 0: onTimeout([Ljava.lang.Object;@11a9310) at 12

12: TIMEOUT 0: onTimeout([Ljava.lang.Object;@1ad093c) at 12

In what order should the events happen? (enter for in-order)

Time: 13

Please input a sequence of commands terminated by a blank line or the TIME command:

The following messages are in transit:

0: Packet: 1->1 protocol: 1 contents: ??%

1: Packet: 1->2 protocol: 1 contents: ????

2: Packet: 2->1 protocol: 1 contents: =??

3: Packet: 2->2 protocol: 1 contents: Y[??

Which should be dropped? (space delimited list or just press enter to drop none)

Which should be delayed? (space delimited list or just press enter to delay none)

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: DELIVERY Packet: 1->1 protocol: 1 contents: ??%

1: DELIVERY Packet: 1->2 protocol: 1 contents: ????

2: DELIVERY Packet: 2->1 protocol: 1 contents: =??

3: DELIVERY Packet: 2->2 protocol: 1 contents: Y[??

4: TIMEOUT 1: onTimeout([Ljava.lang.Object;@d38976) at 13

5: TIMEOUT 1: onTimeout([Ljava.lang.Object;@1e5c339) at 13

6: TIMEOUT 2: onTimeout([Ljava.lang.Object;@17414c8) at 13

7: TIMEOUT 2: onTimeout([Ljava.lang.Object;@7a17) at 13

In what order should the events happen? (enter for in-order)

Time: 14

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: TIMEOUT 1: onTimeout([Ljava.lang.Object;@381a53) at 14

1: TIMEOUT 2: onTimeout([Ljava.lang.Object;@1484a8a) at 14

2: TIMEOUT 1: onTimeout([Ljava.lang.Object;@115c974) at 14

3: TIMEOUT 2: onTimeout([Ljava.lang.Object;@15d0a50) at 14

In what order should the events happen? (enter for in-order)

Time: 15

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

In what order should the events happen? (enter for in-order)

Time: 16

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

In what order should the events happen? (enter for in-order)

Time: 17

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

In what order should the events happen? (enter for in-order)

Time: 18

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

In what order should the events happen? (enter for in-order)

Time: 19

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: TIMEOUT 0: onInvokeTimeout([Ljava.lang.Object;@8aaff4) at 19

1: TIMEOUT 0: onInvokeTimeout([Ljava.lang.Object;@125d92c) at 19

2: TIMEOUT 0: onInvokeTimeout([Ljava.lang.Object;@1d4e49a) at 19

In what order should the events happen? (enter for in-order)

Time: 20

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

In what order should the events happen? (enter for in-order)

Time: 21

Please input a sequence of commands terminated by a blank line or the TIME command:

Live nodes: 0, 1, 2

Crash which nodes? (space-delimited list of addresses or just press enter)

0: COMMAND 0 executes dump\_values

1: COMMAND 1 executes dump\_values

2: COMMAND 2 executes dump\_values

In what order should the events happen? (enter for in-order)

\*\*\* 0: PAXOS: node 0: slot 0: 0.create scenario1\_file.txt

\*\*\* 1: PAXOS: node 1: slot 0: 0.create scenario1\_file.txt

\*\*\* 2: PAXOS: node 2: slot 0: 0.create scenario1\_file.txt

### Scenario 2: full output

### Scenario 3: full output