

SailFin/SJSCAS1.0

Sip Session Replication Test Specification

Please send comments to andrzej.lisowski@ericsson.com & varun.rupela@sun.com

Author(s): Andrzej Lisowski & Varun Rupela

Contributor(s): [Erik van der Velden](#)

Version 0.21

Date: April the 28th, 2008

Table of Contents

1.Introduction.....	3
1.1.General information.....	3
1.2.Test Environment.....	5
1.3.Test Applications.....	6
1.4.Terminology.....	6
1.5.Open Issues.....	7
2.Test Scenarios - Summary.....	7
3.Test Scenarios Details.....	8
3.1.Regression Test [FSD section2.3.1].....	8
3.2.Configuration Tests.....	9
3.3.Single instance failures [FSD section2.3.4].....	10
3.3.1 Failures – INVITE.....	11
3.3.2 Failures – PRESENCE.....	15
3.3.3 Failures – CONVERGED.....	19
3.3.4 Failures – other cases.....	22
3.4.Operation and Maintenance.....	23
3.5.Timers and timeouts.....	25
3.5.1 Session Timers.....	25
3.5.2 Servlet Timers.....	27
3.6.Other.....	29
3.7.Multiple failures.....	30
4.Approvals/Reviewers.....	31
5.References.....	31

1. Introduction

1.1. General information

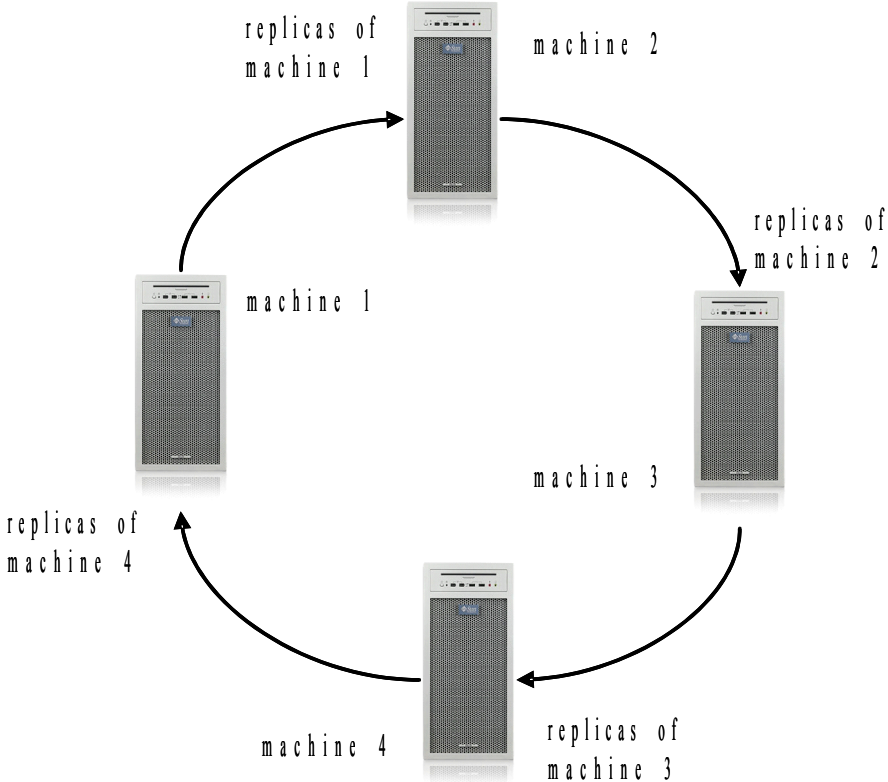
This document specifies the function test cases for the Sip Session Replication functionality. The purpose of the Sip Session Replication is to guarantee high availability to the SailFin cluster. In case of a failure or planned shut-down of the single node in the cluster there will be always replica stored in the cluster of the following data:

- The SIP application Session (SAS)
- The SIP (protocol) sessions (SS)
- The Dialog Fragment
- The Converged HTTP (protocol) sessions - CHS
- Servlet Timers

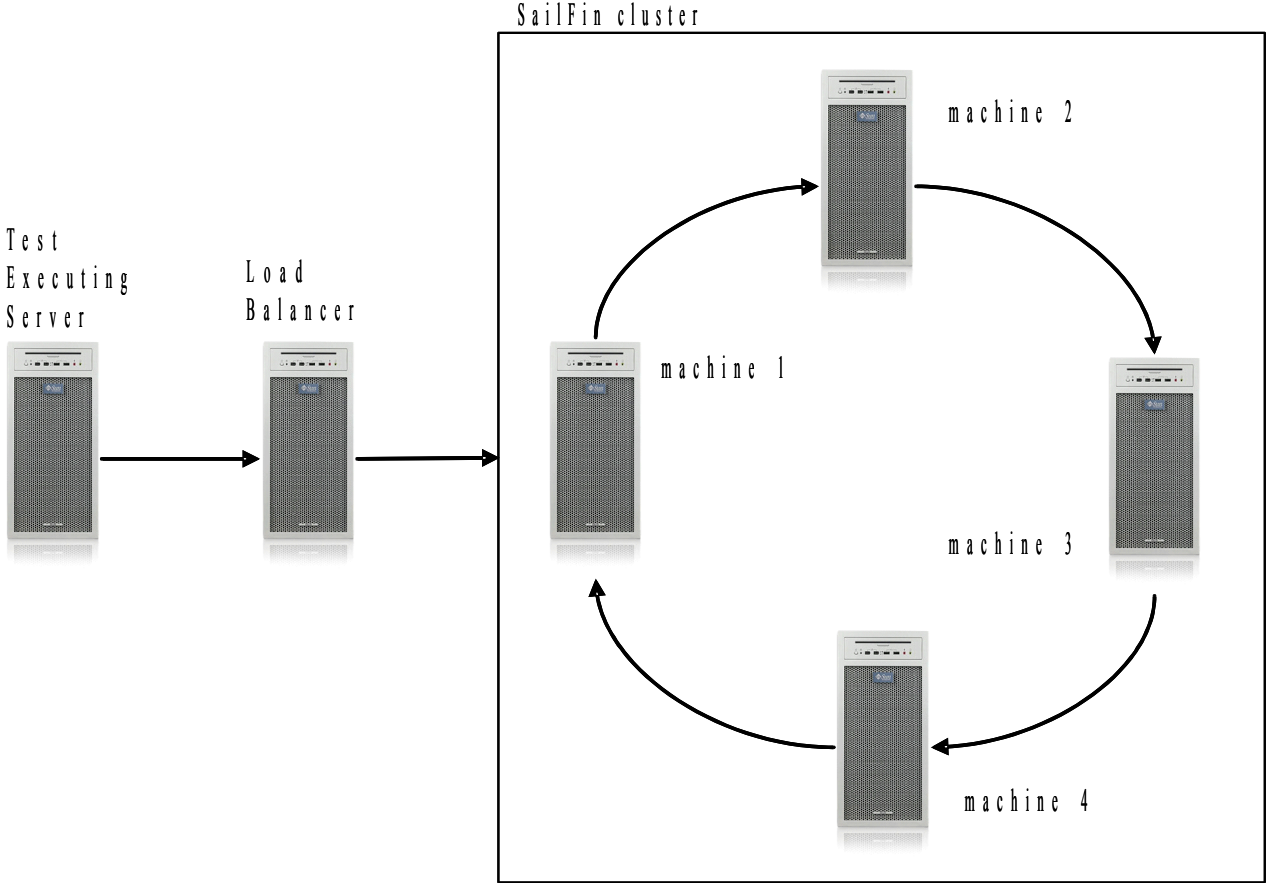
SailFin uses so called buddy replication.

Sip Session Replication Test Specification

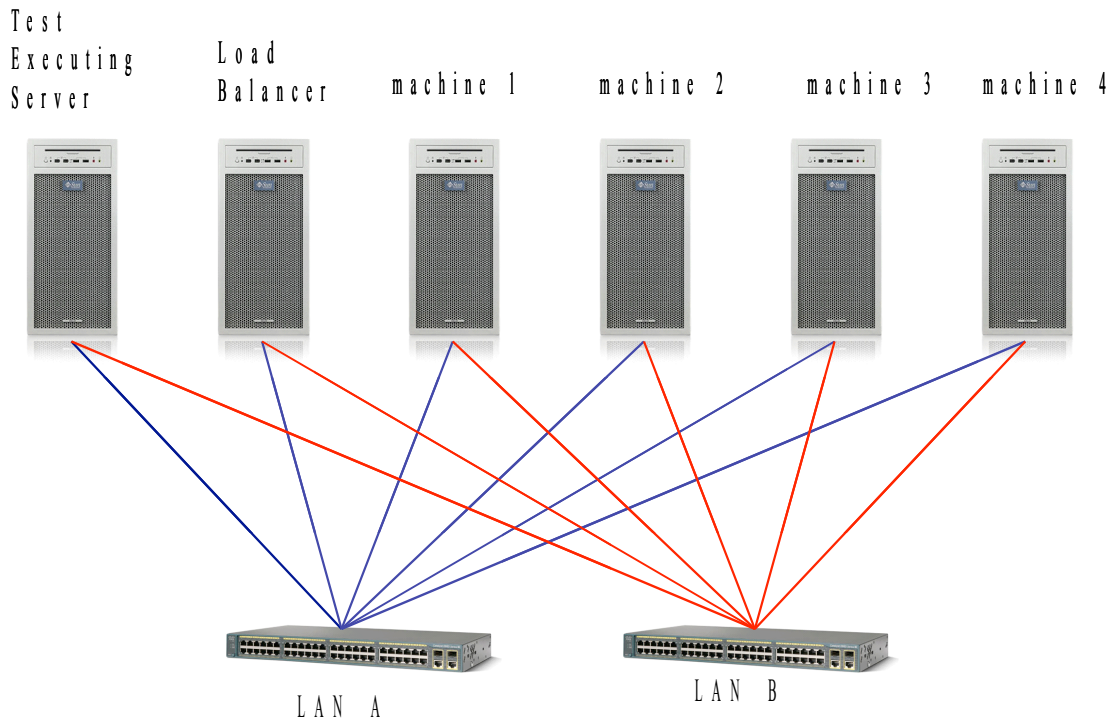
The picture below shows how is the data replicated in the cluster of four machines. The arrows show where is the placement of the replicas.



1.2. Test Environment



Every machine in this network is equipped with two NICs (Network Interface Cards - Ethernet) to the two separate routers so the two separate and independent LANs are created. The picture below represents the actual connections (cabling) in the test network.



1.3. Test Applications

In order to verify the SRR it is necessary to make use of the test applications. There are following three types of the applications distinguished:

- Presence application
It makes use of the PUBLISH, SUBSCRIBE, NOTIFY methods to keep the track of the status of the UAs.
- Call handling application
It handles the SIP calls that are initiated using the INVITE operation.
- Converged application
It makes use of the converged HTTP + SIP session.
- Service chaining
The application(s) that make use of service chaining (can be a combination of the two existing ones).

1.4. Terminology

During writing of this document the effort has been done to comply with the terminology that has been introduced in the “Functional Specification for Sip Session Replication” so please check this document for the terminology.

The definitions of normal load, high load and overload are described in the document “JEE server definitions for normal, high and overload”.

The other terms are described below:

- Machine
Machine is a physical computer/blade/server together with operating system and one SailFin instance.
- Server
Is single SailFin instance running on the machine.
- Temporary failure
The failure that will be signaled by GMS with FailureSuspectedNotification and will be restored before the machine/server is marked as failed.
- Permanent failure
The failure is permanent when the GMS marks the machine/server as failed.

1.5. Open Issues

1. The eager and lazy recovery `repair-during-failure=false/true` must be taken into the test cases during the implementation.

2. Test Scenarios - Summary

This table below will be filled if time permits and if it seems necessary to do the task.

Tests summary table grouped by priority where P1 is highest priority and P2 and so on.. You can have till P5.

<i>Index</i>	<i>Priority</i>	<i>Test Scenario/ Assertion ID</i>	<i>Comments</i>
1.	P1	SEC-TS-1	
2.	P1		
TOTAL =			
3.	P2		
4.	P2		
5.	P2		
TOTAL=			
6.	P3		
TOTAL=			

<i>Index</i>	<i>Priority</i>	<i>Test Scenario/ Assertion ID</i>	<i>Comments</i>

3. Test Scenarios Details

Note: All references to the FSD are to the latest version of the FSD as pointed to in References section.

3.1. Regression Test [FSD section2.3.1]

SRR-TS-1: Regression test with SRR enabled

Priority: P1

Description/Strategy:

The purpose of running regression test after enabling of the SRR is to make sure that it has no impact on the existing functionality. The regression test suite consists of the following modules:

1. SIP Traffic, INVITE
2. SIP Traffic, NOTIFY
3. SIP Traffic, PUBLISH
4. SIP Traffic, SUBSCRIBE
5. SIP Traffic, Requests and Responses
6. SIP Traffic, Scenarios
7. SIP Servlet API
8. Protocol Timers

Open Issues:

-

SRR-TS-2: Regression test with SRR disabled

Priority: P1

Description/Strategy:

The purpose of running regression test after disabling of the SRR is to make sure that it has no impact on the existing functionality. These tests are already run with no SSR enabled when run in non-clustered domain. The regression test suite consists of the following modules:

9. SIP Traffic, INVITE
10. SIP Traffic, NOTIFY

11. SIP Traffic, PUBLISH
12. SIP Traffic, SUBSCRIBE
13. SIP Traffic, Requests and Responses
14. SIP Traffic, Scenarios
15. SIP Servlet API
16. Protocol Timers

Open Issues:

-

3.2. Configuration Tests

SRR-TS-3: Persistence Type and Scope [FSD section2.3.1.5]

Priority: P2

Description/Strategy:

1. Persistence-type “replicated” and “memory” both will be tested. When using “memory” as the persistence-type, it will be verified that replication actually does not happen. Persistence-scope “session”, “modified-session” and “modified-attribute” will also be tested. For “modified-attribute” a Warning will be expected in the logs on deploy or load of the application.

Open Issues:

SRR-TS-4: Persistence Type and Scope override setting with sun-sip.xml [FSD section2.3.1.6]

Priority: P3

Description/Strategy:

1. Some persistence-type and persistence-scope values will be chosen from TS3, different values used in the sun-sip.xml. It will be checked if the values in sun-sip.xml are the ones that are applied.

Open Issues:

SRR-TS-5: Dialog Replication Activation [FSD section2.3.1.8]

Priority: P2

Description/Strategy:

1. Have session-replication enabled on all the applications
2. Option - - availabilityenabled = true will be used when deploying each of the

applications.

Open Issues:

SRR-TS-6: Dialog Replication Disabled [FSD section2.3.1.9]

Priority: P3

Description/Strategy:

1. Have session-replication NOT enabled on any application.
2. Clarification: This is not the same as testing persistence-type “memory”. The way to enable SSR would be to use the option - - availabilityenabled=false option when deploying the applications.

Open Issues:

SRR-TS-7: Dialog Replication Mis-configuration [FSD section2.3.1.10]

Priority: P3

Description/Strategy:

1. Have session-replication enabled on some and disabled on some applications
2. Expect warning stating mis-configuration.
3. Clarification: This test will only look for the warning.

Open Issues:

1. Should the test also try replication and then ensure that dialog replication did not actually happen and user was warned with appropriate errors/warning.

3.3. Single instance failures [FSD section2.3.4]

Comment From Joe Fialli: Only impacts test scenarios involving a stop/kill followed by a subsequent restart. Test scenario can validate lazy recovery, repair-during-failure=false, by adding a step that explicitly accesses all SIP artifacts AFTER a server instance rejoins the cluster. After accessing all SIP artifacts, this ensures that all SIP artifacts in the cluster are replicated once again, moving the session data in the cluster from the repair period to normal period described in FSD-SSR v2.3, Section 2.3.4.

When eager recovery is being used, repair-during-failure=true, and this is the default, I am not certain what the test strategy is to know when this period has passed and the session data in the cluster moves from the "repair period" to the "normal" fully replicated in the cluster state.

The FSD-SSR requirement document states that eager repair shortens the repair period but places no way to know when the period is complete or requirement this must complete before the server

instance rejoins the cluster. Whatever the recovery period is, any test that wishes to validate that sessions are not lost needs to wait at least this period of time before kill/stop the next server instance in the cluster. (Referring to SRR-TS-14)

3.3.1 Failures – INVITE

SRR-TS-8: Short network failure at machine - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The purpose of this test case is to verify that Session Replication can correctly handle the case of the network failure towards one of the machines. One SAS is created per instance. The network connection towards one machine are removed. The interaction with all SAS is performed. The instance is started and all the SAS in the cluster are terminated.

This test case verifies additionally the versioning mechanism of the Session Replication since at the certain point in time there will be two same SAS present in the cluster with different version numbers.

Open Issues:

1. Comments from Joe Fialli: Unsure if goal of this test is testing intermittent network failure or a complete server instance stop and start. Removing the network connection sounds like an intermittent failure. "Starting the instance" sounds like the instance was previously stopped. Is the steps for this test case:
 - a. Create one SAS per instance. (which implies access to CLB hashing mechanism)
 - b. Remove network connection. (and the server instance is assumed to be failed by the cluster since it can no longer reach it.)
 - c. Update all SAS. (Failing over when accessing the SASes to the machines that are isolated by network connection being disconnected)
 - d. Reinsert network connection and the server instance rejoins the cluster. (instead of "The instance is started")

Not sure which requirement from the FSD this test case is verifying.

Open Issues:

-

SRR-TS-9: Permanent network failure at machine - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of permanent network failure on SailFin cluster.

There is one presence session created per SailFin instance. Afterwards there is network connection removed manually from a server for a longer period of time (longer than five minutes). When the connection is down the update of all the SASes is performed. Than the connection is restored and all the sessions are explicitly terminated.

Open Issues:

-

SRR-TS-10: Short network failure towards router/switch - - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of short network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is for very short moment (few seconds) network connection remover from switch/router. Once the connection is restored the update of the sessions is performed and in the end the sessions are terminated.

Open Issues:

-

SRR-TS-11: Permanent network failure towards router/switch - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of permanent network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is network connection removed manually from a switch/router for a longer period of time (longer than five minutes). When the connection is down the update of all the SASes is performed. Than the connection is restored and all the sessions are explicitly terminated.

Open Issues:

-

SRR-TS-12: Temporary power failure - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

Open Issues:

1. Comments from Joe Fialli: The description/strategy for this test is identical to SSR-TS-12. Both tests remove a network cable. Neither test states specifically what network cable is removed relative to network topology description in described in Section 1.2 Test Environment. This section describes that each machine has two NICs and belong to two lans. Does one machine have the network cable to both of its NICs pulled? Or is only one network cable pulled from either lan A or lan B for a given machine?

Lastly, does SRR-TS-13 simulate machine failure by additionally stopping and restarting the server instance. Or killing and restarting the server instance. If neither of the two proposed steps above are performed, then SSR-TS-12 is identical to SRR-TS-13.

SRR-TS-13: Permanents power failure - INVITE

Priority: P1

Description/Strategy:

This is MANUAL test case.

Open Issues:

1. Comments from Joe Fialli: The description/strategy for this test is identical to SSR-TS-12. Both tests remove a network cable. Neither test states specifically what network cable is removed relative to network topology description in described in Section 1.2 Test Environment. This section describes that each machine has two NICs and belong to two lans. Does one machine have the network cable to both of its NICs pulled? Or is only one network cable pulled from either lan A or lan B for a given machine?

Lastly, does SRR-TS-13 simulate machine failure by additionally stopping and restarting the server instance. Or killing and restarting the server instance. If neither of the two proposed steps above are performed, then SSR-TS-12 is identical to SRR-TS-13.

SRR-TS-14: Instance process failure - INVITE

Priority: P2

Description/Strategy:

The instance is killed and node agent restarts the process immediately.

Open Issues:

-

SRR-TS-15: Temporary node agent and instance failure - INVITE

Priority: P2

Description/Strategy:

Open Issues:

-

SRR-TS-16: Permanent node agent and instance failure - INVITE

Priority: P2

Description/Strategy:

Open Issues:

SRR-TS-17: Temporary planned shutdown of the instance - INVITE

Priority: P2

Description/Strategy:

Stop the instance using asadmin stop-instance and start it back immediately.

Open Issues:

-

SRR-TS-18: Permanent planned shutdown of the instance - INVITE

Priority: P2

Description/Strategy:

Stop the instance using asadmin stop-instance and start it back after considerable amount of time (say > 5 minutes).

Open Issues:

SRR-TS-19: Reboot machine - INVITE

Priority: P4

Description/Strategy:

Reboot machine using reboot command. SailFin starts immediately after machine is rebooted.

Open Issues:

SRR-TS-20: Shutdown machine - INVITE

Priority: P4

Description/Strategy:

Shutdown machine using shutdown command. The machine remains down for considerable amount of time (say > 5 minutes).

Open Issues:

SRR-TS-21: Planned shutdown instance one by one - INVITE

Priority: P2

Description/Strategy:

Shutdown all instances one by one using asadmin command. Every instance remains down for considerable amount of time i.e. few minutes.

Open Issues:

3.3.2 Failures – PRESENCE

SRR-TS-22: Short network failure at machine – PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of short network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is for very short moment (few seconds) network connection removed manually from a server. Once the connection is restored the update of the sessions is performed and in the end the sessions are terminated.

Open Issues:

-

SRR-TS-23: Permanent network failure at machine - PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of permanent network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is network connection removed manually from a server for a longer period of time (longer than five minutes). When the connection is down the update of all the SASes is performed. Than the connection is restored and all the sessions are explicitly terminated.

Open Issues:

-

SRR-TS-24: Short network failure towards router/switch - PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of short network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is for very short moment (few seconds) network connection remover from switch/router. Once the connection is restored the update of the sessions is performed and in the end the sessions are terminated.

Open Issues:

-

SRR-TS-25: Permanent network failure towards router/switch - PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

The goal of this test case is to verify the impact of permanent network failure on SailFin cluster. There is one presence session created per SailFin instance. Afterwards there is network connection removed manually from a switch/router for a longer period of time (longer than five minutes). When the connection is down the update of all the SASes is performed. Than the connection is restored and all the sessions are explicitly terminated.

Open Issues:

-

SRR-TS-26: Temporary power failure - PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-27: Permanents power failure - PRESENCE

Priority: P1

Description/Strategy:

This is MANUAL test case.

Open Issues:

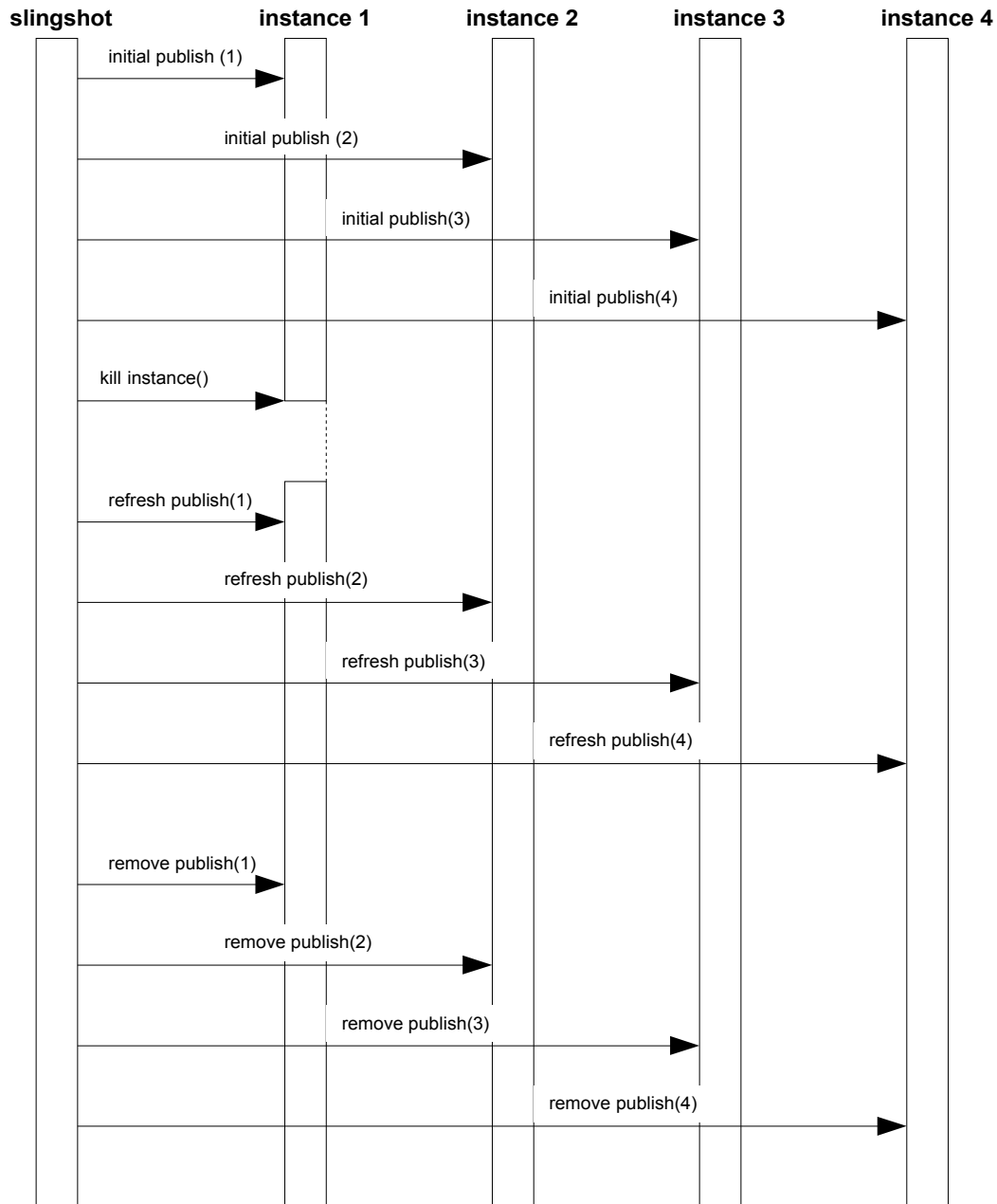
-

SRR-TS-28: Instance process failure - PRESENCE

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that the replica SIP Application Session is properly restored from the replica cash after the restart of the instance. The instance is killed and node agent restarts the instance process immediately.



Open Issues:

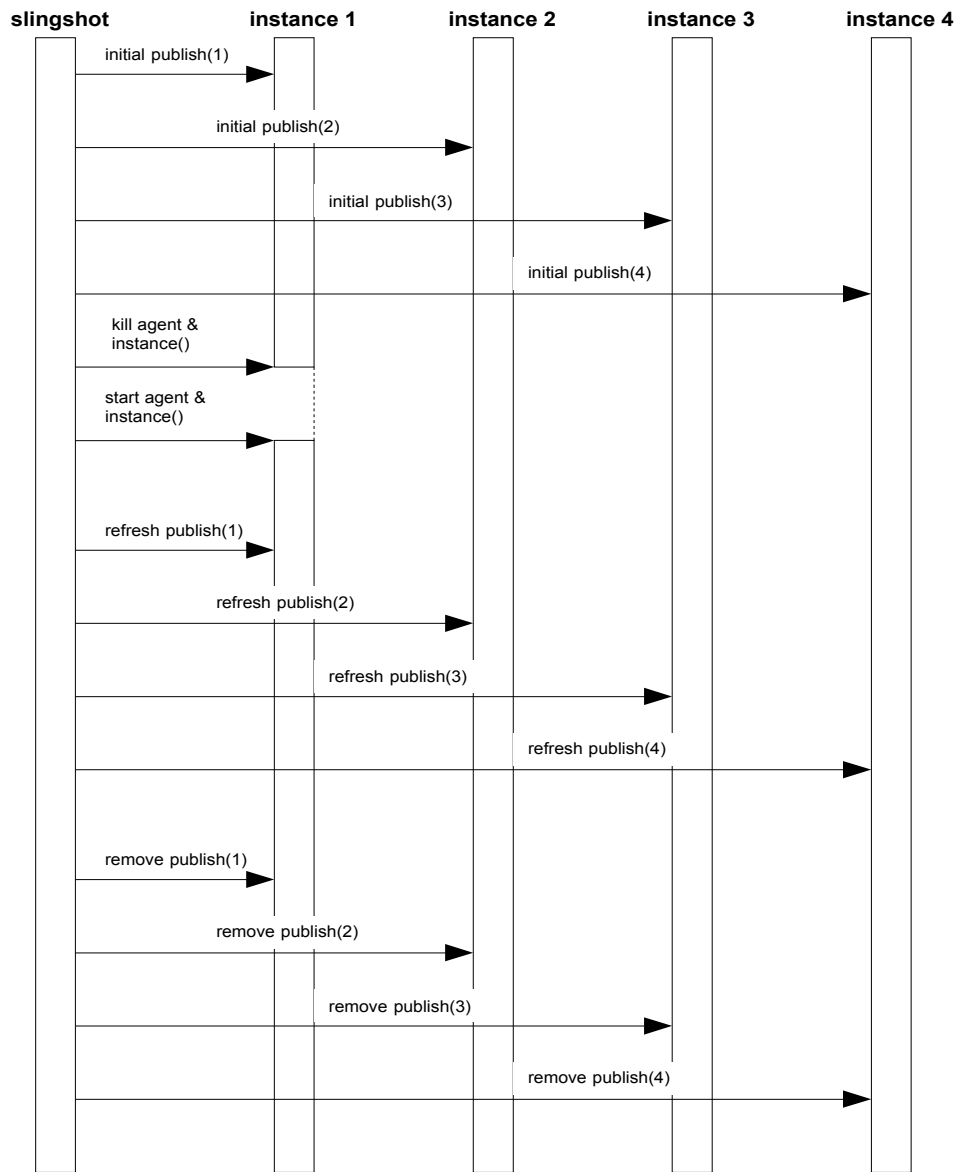
-

SRR-TS-29: Temporary node agent and instance failure - PRESENCE

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that the session replication can properly handle the short failure of both node agent and instance processes. After the instantiating of both Java processes Slingshot tries to refresh the SAS data on the crashed instance. The crash instance before refreshing the SAS data it obtains its copy from the replica cache. The flow for the test case looks as follows.



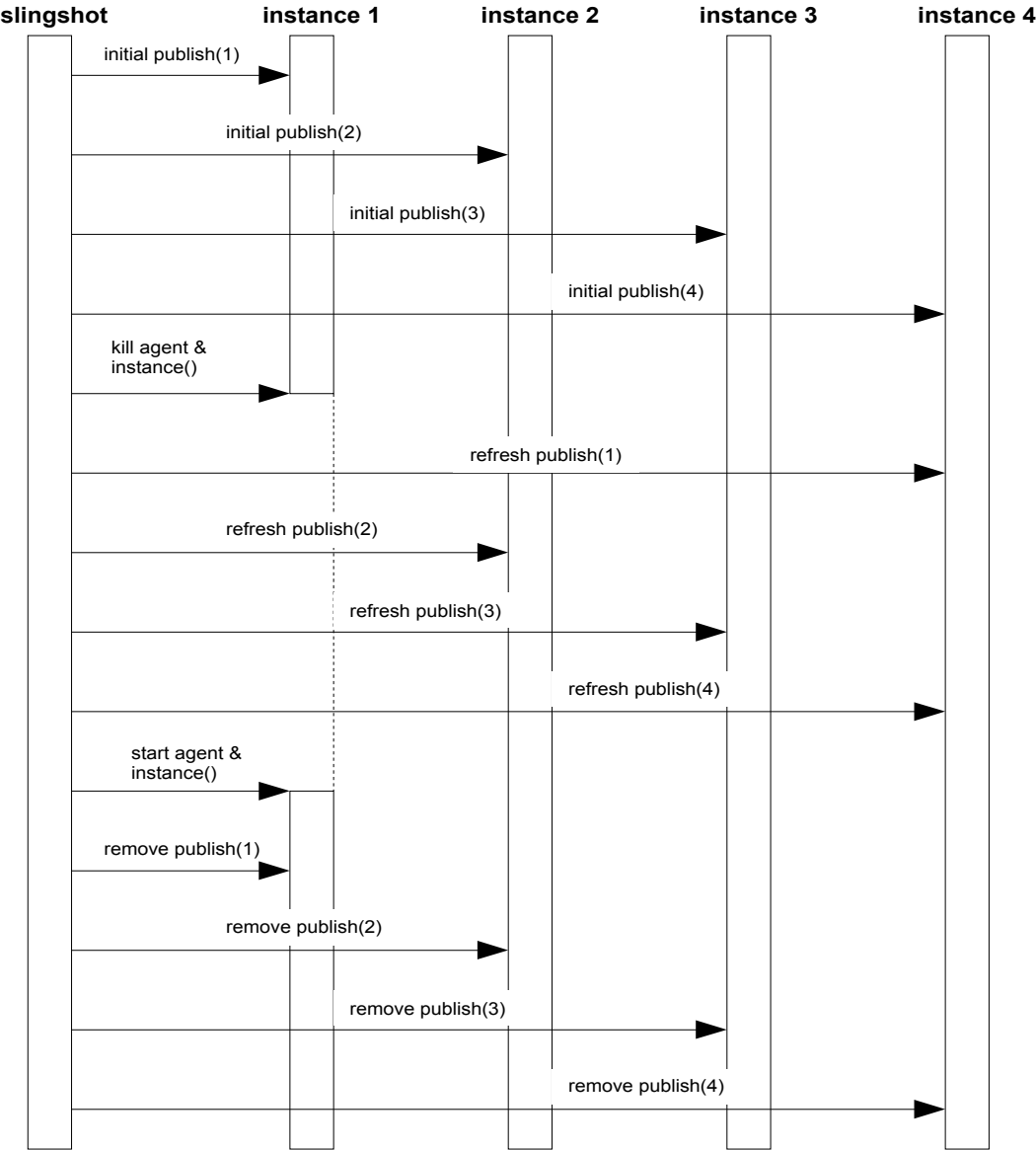
Open Issues:

SRR-TS-30: Permanent node agent and instance failure - PRESENCE

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that the session replication can properly handle the migration of the SAS data in case of the instance failure and restore after the crashed instance is restored. The event flow for the test case look as follows:



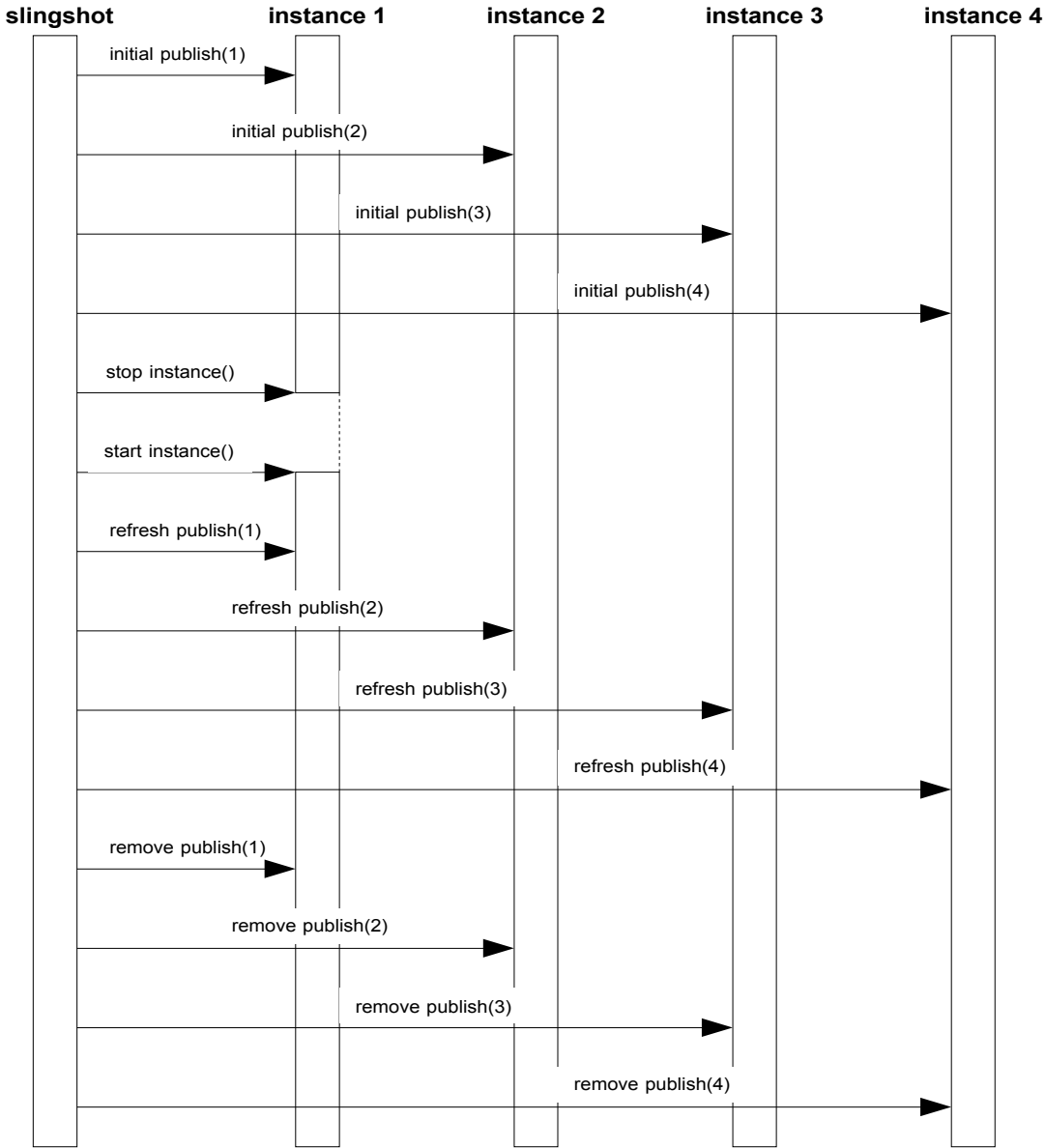
Open Issues:

SRR-TS-31: Temporary planned shutdown of the instance - PRESENCE

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that session replication can properly handle stop and start of the instance using asadmin without migration of the SAS data. The event sequence for the test case looks as follows:



Open Issues:

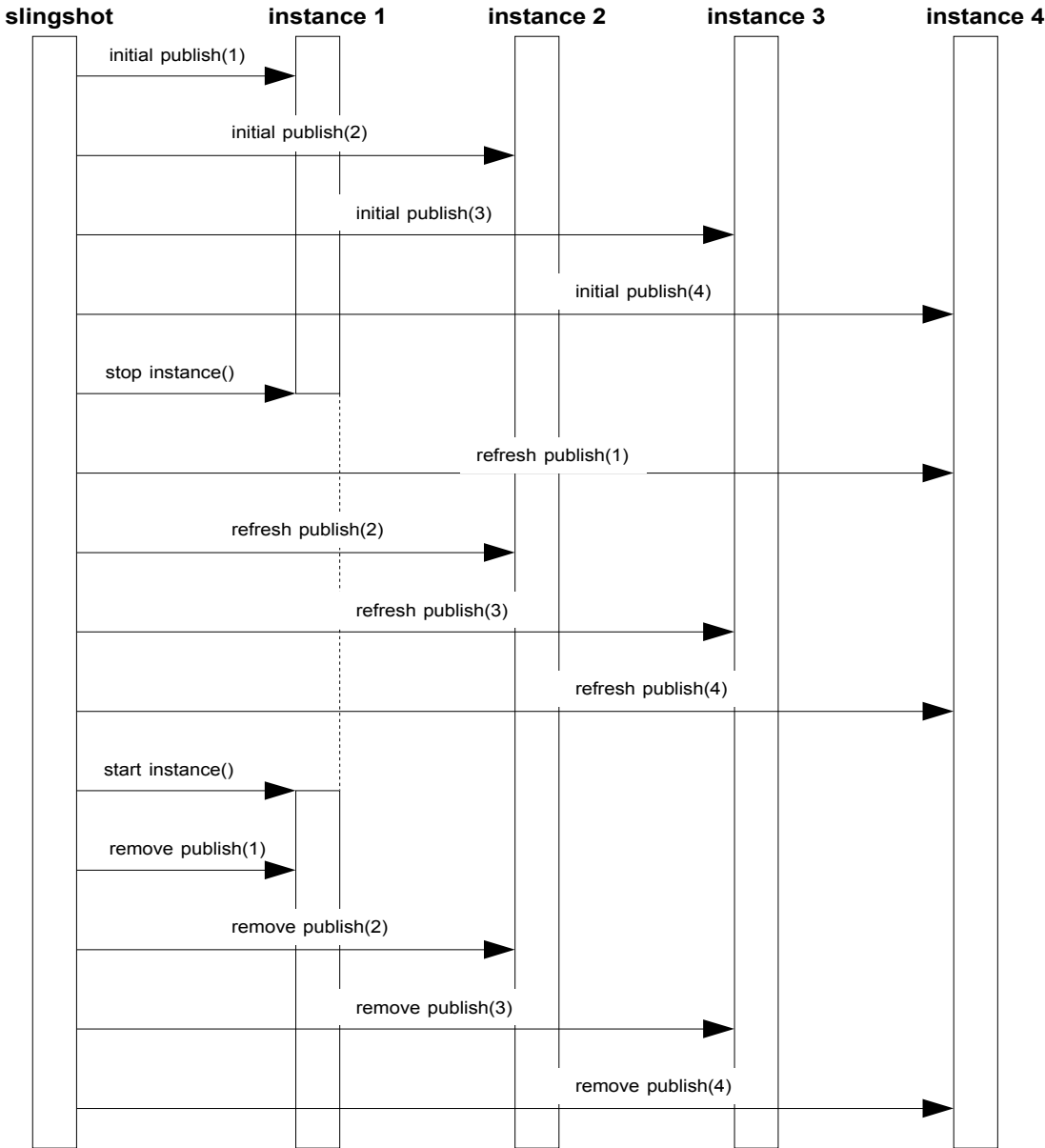
-

SRR-TS-32: Permanent planned shutdown of the instance - PRESENCE

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that session replication can properly handle stop and start of the instance using asadmin including migration of the SAS data. The event sequence for the test case looks as follows:



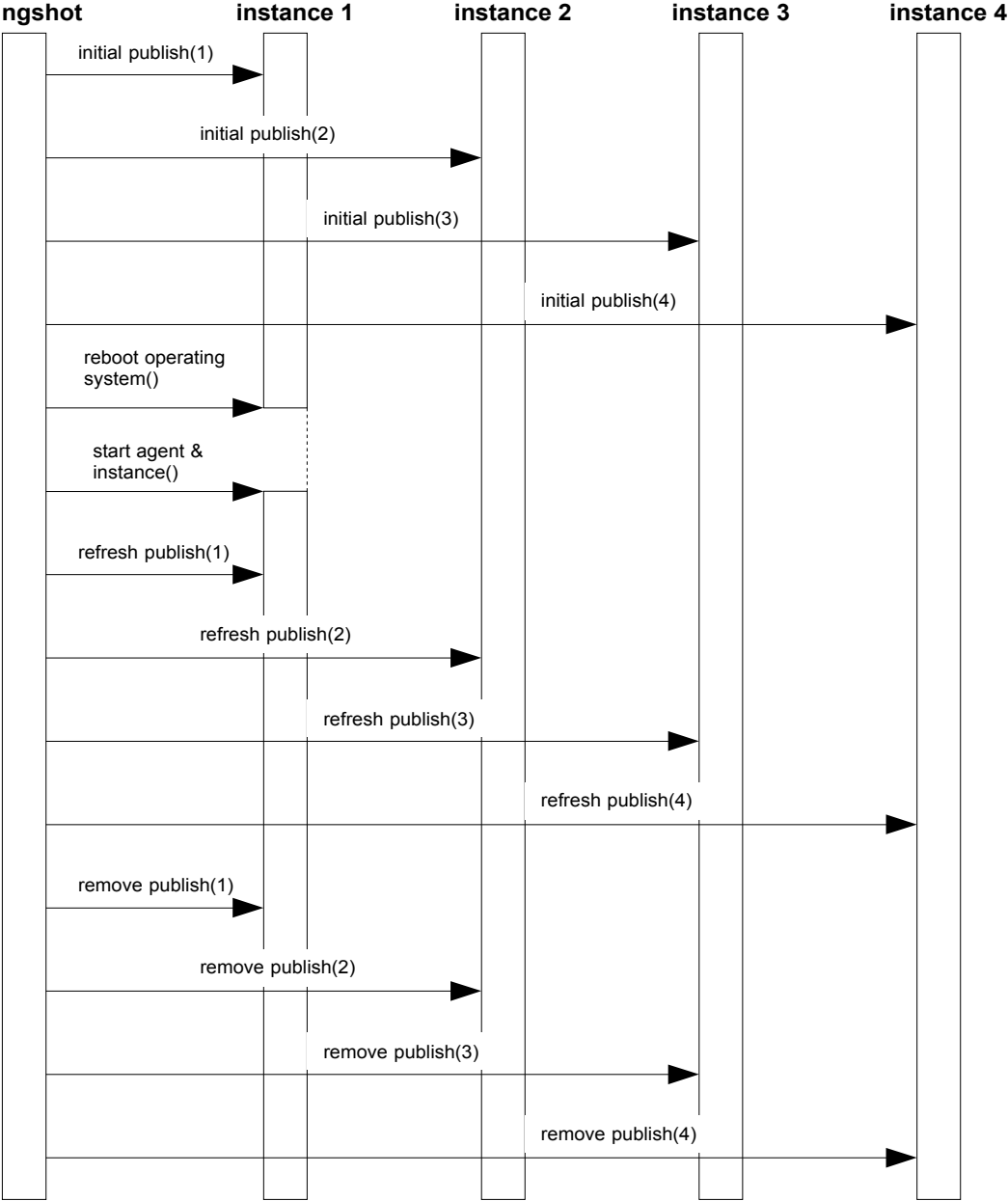
Open Issues:

SRR-TS-33: Reboot machine - PRESENCE

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that the session replication can properly handle reboot of the operating system without migration of the SAS data. The event sequence for the test case is as follows:



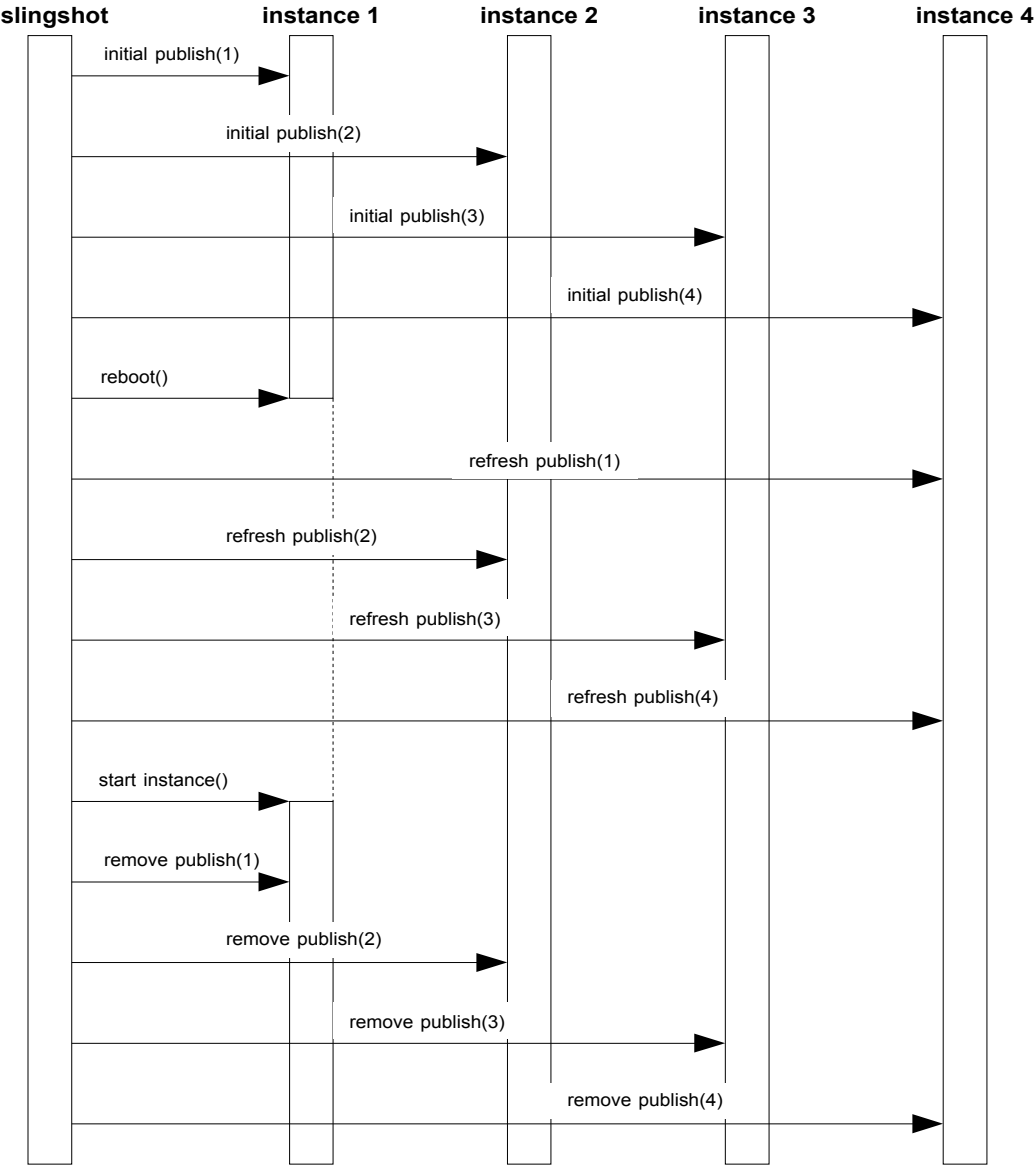
Open Issues:

SRR-TS-34: Shutdown machine - PRESENCE

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that the session replication can properly handle shutdown of the operating system with migration of the SAS data. In order to be able to automate this TC that shutdown has been replaced with reboot command. The event sequence for the test case is as follows.



Open Issues: -

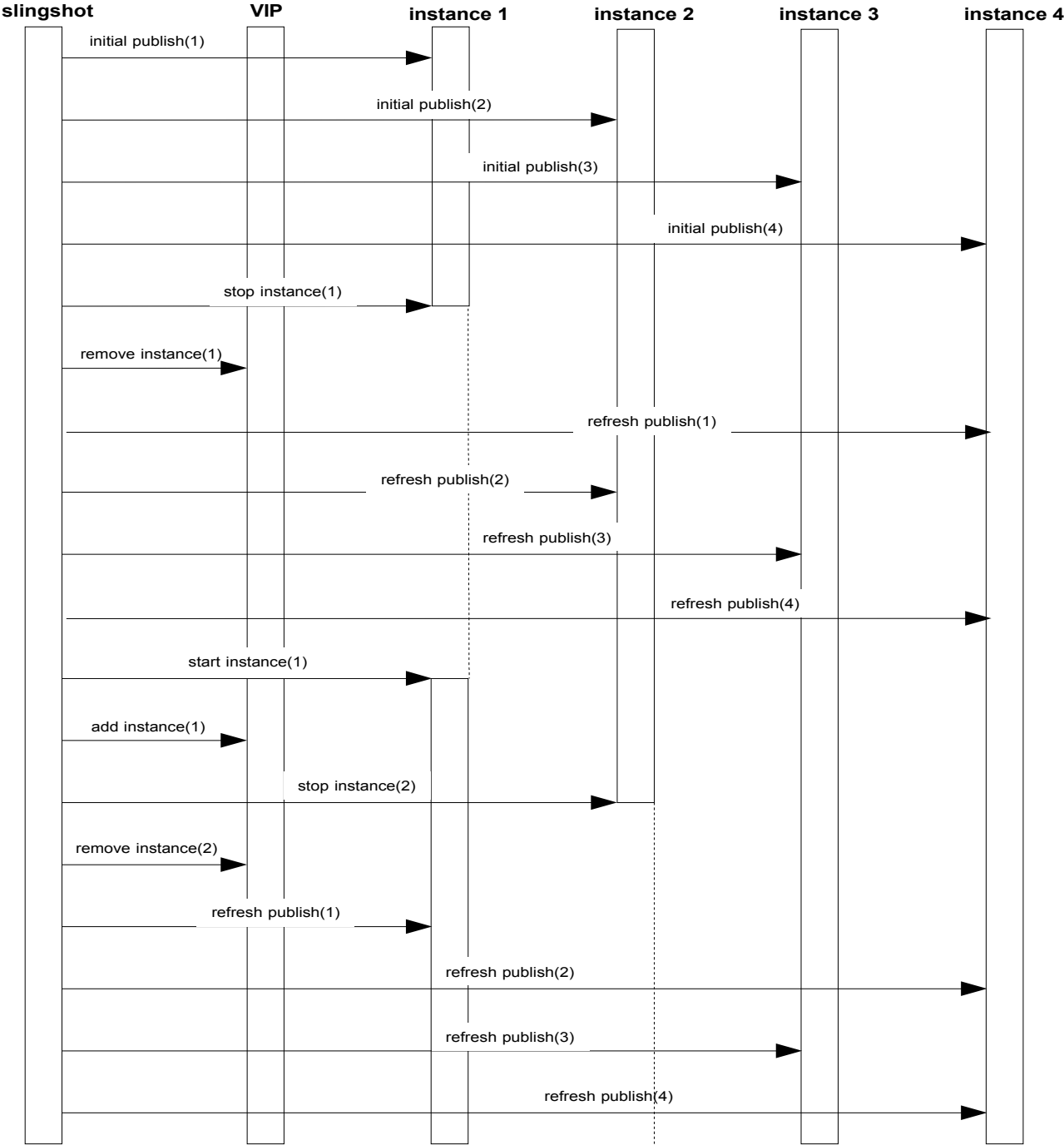
SRR-TS-35: Planned shutdown instance one by one - PRESENCE

Priority: P2

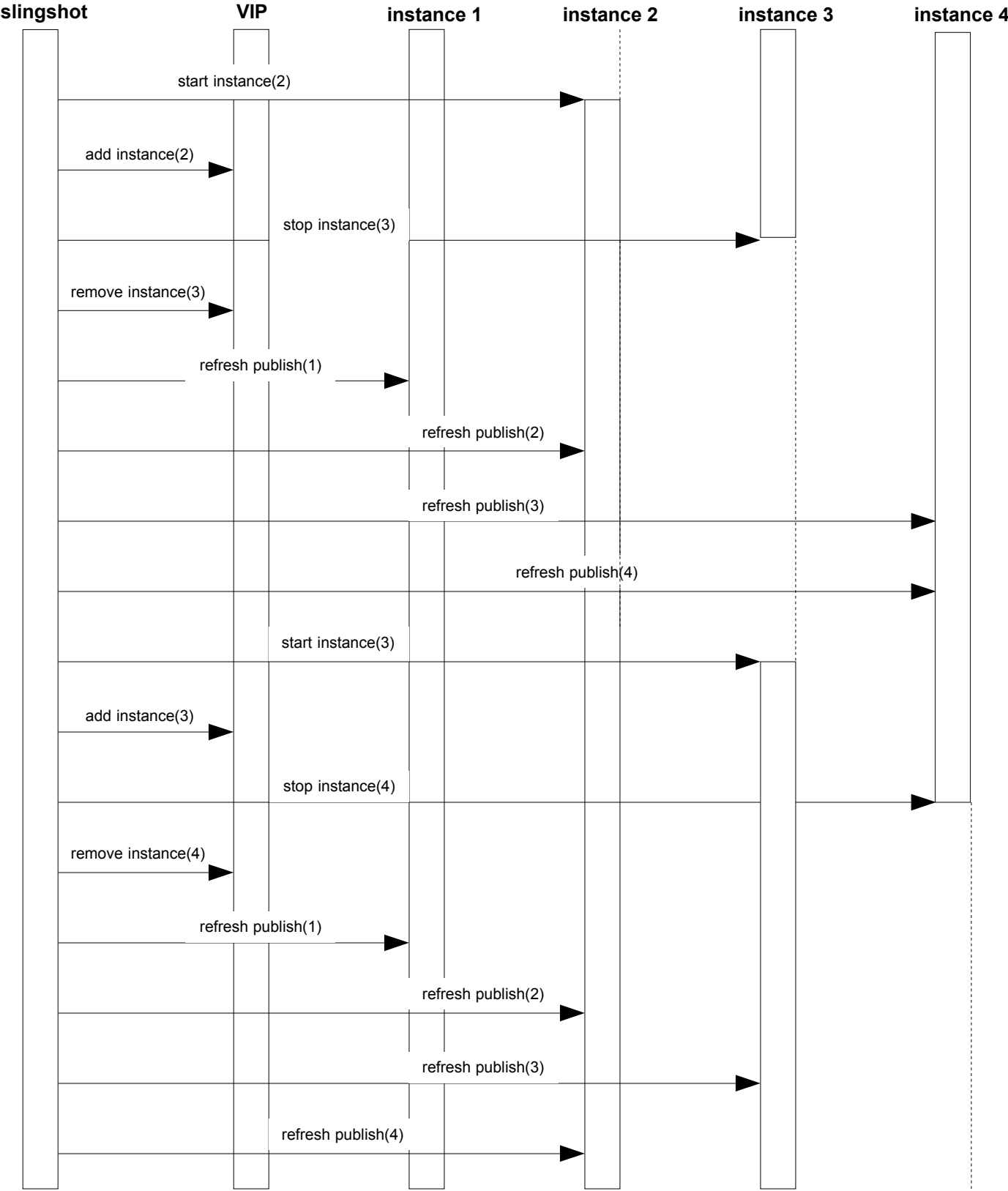
Description/Strategy:

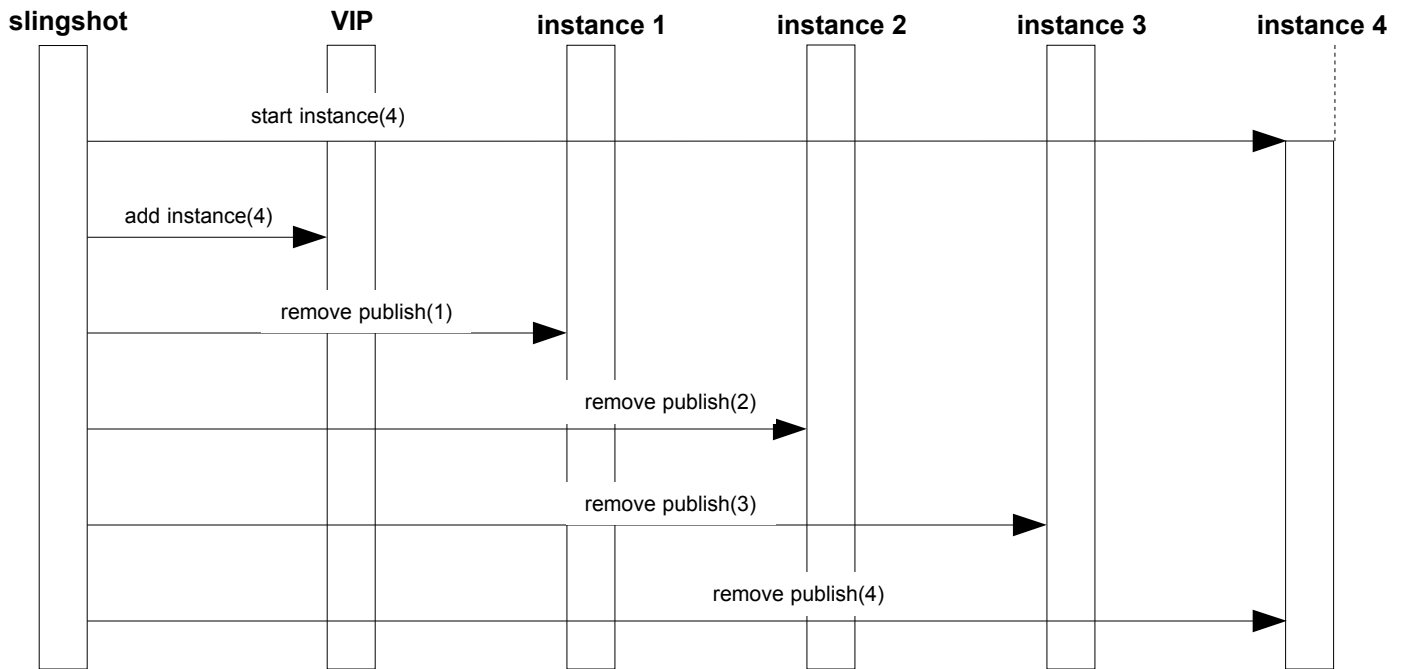
The goal of this test case is to verify whether the Session Replication can properly handle the

planned shutdown from asadmin command line interface of all the instances in the cluster including the migration of the data for every of the shutdown events.



Sip Session Replication Test Specification





Open Issues:

3.3.3 Failures – CONVERGED

SRR-TS-36: Short network failure at machine – CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-37: Permanent network failure at machine - CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-38: Short network failure towards router/switch - CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-39: Permanent network failure towards router/switch - CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-40: Temporary power failure - CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-41: Permanent power failure - CONVERGED

Priority: P3

Description/Strategy:

This is MANUAL test case.

Open Issues:

-

SRR-TS-42: Instance process failure - CONVERGED

Priority: P3

Description/Strategy:

The instance is killed and node agent restarts the process immediately.

Open Issues:

-

SRR-TS-43: Temporary node agent and instance failure - CONVERGED

Priority: P3

Description/Strategy:

Open Issues:

-

SRR-TS-44: Permanent node agent and instance failure - CONVERGED

Priority: P3

Description/Strategy:

Open Issues:

SRR-TS-45: Temporary planned shutdown of the instance - CONVERGED

Priority: P3

Description/Strategy:

Stop the instance using asadmin stop-instance and start it back immediately.

Open Issues:

-

SRR-TS-46: Permanent planned shutdown of the instance - CONVERGED

Priority: P4

Description/Strategy:

Stop the instance using asadmin stop-instance and start it back after considerable amount of time (say > 5 minutes).

Open Issues:

SRR-TS-47: Reboot machine - CONVERGED

Priority: P4

Description/Strategy:

Reboot machine using reboot command. SailFin starts immediately after machine is rebooted.

Open Issues:

SRR-TS-48: Shutdown machine - CONVERGED

Priority: P3

Description/Strategy:

Shutdown machine using shutdown command. The machine remains down for considerable amount of time (say > 5 minutes).

Open Issues:

3.3.4 Failures – other cases

SRR-TS-49: Session replication with large sessions

Priority: P4

Description/Strategy:

The purpose of this test case is to verify whether session replication can properly replicate large session of several megabytes (say > 20 MB).

Open Issues:

SRR-TS-50: Instance failure – service chaining

Priority: P4

Description/Strategy:

The goal of this test case is to verify the basic functionality of Session Replication for service chaining.

Open Issues:

It is not really clear at the moment how this test case should look like in details.

SRR-TS-51: Fail an instance during different type of actions on the session attributes

Priority: P4

Description/Strategy:

1. Do Add / Modify / Delete to a session attribute, fail the instance with the active session and then continue session to ensure that the replica had the updated session.

Open Issues:

SRR-TS-52: Fail an instance when it is serving a request that corresponds to a large sized session

Priority: P4

This test case makes sure that the Session Replication can properly handle the failure of one of the instances when a replication of a large instance is ongoing. There is one SAS created per instance with a size of 100 MB. During the replication one of the instances is stopped thus the replication is unsuccessful and the session is lost. Afterwards the interaction is performed with all the sessions. The interaction with the session that was not replicated fails.

Description/Strategy:

1. Comment From Joe Fialli: Difficult to write a test that deterministically fails a server instance before asynchronous replication message is sent but after the request to create the SAS returns. Would need to place a hook in sailfin replication implementation code to have a chance of deterministically achieving the described test case.

Open Issues:

SRR-TS-53: Inject failures (stop/kill instances) during different stages of a Call

Priority: P4

Description/Strategy:

1. Different stages of a dialog meaning – invite, ack, 200 ok etc.

Open Issues:

SRR-TS-54: Invalidate session on and then fail the instance that held the session

Priority: P4

Description/Strategy:

1. Access the session after the failure, the session should remain invalidated

Open Issues:

3.4. Operation and Maintenance

SRR-TS-55: Add a new instance to the cluster

Priority: P3

Description/Strategy:

1. The purpose of this test case is to verify whether adding a new instance to the cluster is properly handled by the Session Replication. Session replication mechanisms should take new instance into the account.
Parameter `repair-during-failure=true`.
2. Possibility: Have a few sessions created on the cluster. (Perhaps do an instance failure before moving on). Add a new instance, create more sessions, fail a different instance. Check that no sessions are lost.

Open Issues:

1. Take another look at the utility of this test.

SRR-TS-56: Add a new instance to the cluster

Priority: P3

Description/Strategy:

1. The purpose of this test case is to verify whether adding a new instance to the cluster is properly handled by the Session Replication. Session replication mechanisms should take new instance into the account.
Parameter `repair-during-failure=false`.
2. Possibility: Have a few sessions created on the cluster. (Perhaps do an instance failure before moving on). Add a new instance, create more sessions, fail a different instance. Check that no sessions are lost.

Open Issues:

1. Take another look at the utility of this test.

SRR-TS-57: Add multiple new instances to the cluster

Priority: P4

The purpose of this test case is to verify whether adding several new instances to the cluster is properly handled by the Session Replication.

Description/Strategy:

Parameter `repair-during-failure=true`.

Open Issues:

1. Take another look at the utility of this test.

SRR-TS-58: Add multiple new instances to the cluster

Priority: P4

Description/Strategy:

The purpose of this test case is to verify whether adding several new instances to the cluster is properly handled by the Session Replication.

Parameter `repair-during-failure=false`.

Open Issues:

1. Take another look at the utility of this test.

SRR-TS-59: Add new instances to the cluster in compromised state

Priority: P4

Description/Strategy:

The purpose of this test case is to verify whether adding several new instances to the cluster is properly handled by the Session Replication at the moment when one of the instances has recently crashed.

Parameter `repair-during-failure=false`.

Open Issues:

1. Comment from Joe Fialli: More details are needed for this test. What was being done before server instance crashed? How is it verified that several new instances are added to the cluster are properly handled by replication.

3.5. Timers and timeouts

3.5.1 Session Timers

SRR-TS-60: HTTP session timeout at active partner for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that HTTP session timeout is properly handled by session replication for converged session by active partner. There is one converged session created per instance. The session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-61: HTTP session timeout at replication partner for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that HTTP session timeout is properly handled by session replication for converged session by replication partner. There is one converged session created per instance. One of the instances is stopped. The session replica is deactivated after the session timer elapses. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-62: SAS session timeout at active partner

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that SAS session timeout is properly handled by active partner. There is one SAS session created per instance. The SAS session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-63: SAS session timeout at replication partner [FSD section2.3.4.7]

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that SAS session timeout is properly handled by replication partner. There is one SAS session created per instance. One of the instances is stopped. The SAS session is deactivated after the session timer elapses for replica. Any interactions with the timed out replica fail.

Open Issues:

-

SRR-TS-64: SAS session timeout at active partner for converged session

Priority: P3

Description/Strategy:

The purpose of this test case is to verify that SAS session timeout is properly handled by session replication for converged session at active partner. There is one converged session created per instance. The session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-65: SAS session timeout of replica for converged session [FSD section2.3.4.7]

Priority: P3

Description/Strategy:

The purpose of this test case is to verify that SAS session timeout is properly handled by session replication for converged session by replication partner. There is one converged session created per instance. The session is deactivated after the session timer elapses for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-66: Test Timers after upscale/downscale of the cluster

Priority: P3

Description/Strategy:

1. Have multiple timers set to expire and around that time do a upscale or downscale of the system such that the session is migrated ? The system should continue to behave normally.

Open Issues:

-

SRR-TS-67: Stop/start an instance on which a session's timer just expired

Priority: P4

Description/Strategy:

1. Ensure that timer expires only once.

Open Issues:

-

SRR-TS-68: Application timer missed 9 times

Priority: P4

Description/Strategy:

Application timer expires 9 times on the replica with missed triggers. The last trigger is triggered by the timer + delta.

Open Issues:

Needs to be implemented in the test application.

3.5.2 Persistent Servlet Timers

SRR-TS-69: Persistent servlet timer timeout for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by session replication for converged session by active partner. There is one converged session created per instance. The session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-70: Persistent servlet timer timeout at replication partner for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by session replication for converged session by replication partner. There is one converged session created per instance. One of the instances is stopped. The session replica is deactivated after the session timer elapses. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-71: Persistent servlet timer timeout at active partner for SIP session

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by active partner. There is one SAS session created per instance. The SAS session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-72: Persistent servlet timer timeout at replication partner for SIP session [FSD section2.3.4.7]

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by replication partner. There is one SAS session created per instance. One of the instances is stopped. The SAS session is deactivated after the session timer elapses for replica. Any interactions with the timed out replica fail.

Open Issues:

-

SRR-TS-73: Persistent Servlet Timers after upscale/downscale of the cluster

Priority: P3

Description/Strategy:

1. Have multiple timers set to expire and around that time do a upscale or downscale of the system such that the session is migrated ? The system should continue to behave normally.

Open Issues:

-

SRR-TS-74: Stop/start an instance on which a session's persistent servlet timer just expired

Priority: P4

Description/Strategy:

1. Ensure that timer expires only once.

Open Issues:

-

3.5.3 Non-persistent Servlet Timers

SRR-TS-75: Non-persistent servlet timer timeout for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by session replication for converged session by active partner. There is one converged session created per instance. The session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-76: Non-persistent servlet timer timeout at replication partner for converged session

Priority: P4

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by session replication for converged session by replication partner. There is one converged session created per instance. One of the instances is stopped. The session replica is deactivated after the session timer elapses. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-77: Non-persistent servlet timer timeout at active partner for SIP session

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by active partner. There is one SAS session created per instance. The SAS session is deactivated after the session timer elapses both in primary memory and for replica. Any interactions with the timed out sessions fail.

Open Issues:

-

SRR-TS-78: Non-persistent servlet timer timeout at replication partner for SIP session [FSD section2.3.4.7]

Priority: P2

Description/Strategy:

The purpose of this test case is to verify that servlet timer timeout is properly handled by replication partner. There is one SAS session created per instance. One of the instances is stopped. The SAS session is deactivated after the session timer elapses for replica. Any interactions with the timed out replica fail.

Open Issues:

-

SRR-TS-79: Non-persistent servlet timers after upscale/downscale of the cluster

Priority: P3

Description/Strategy:

1. Have multiple timers set to expire and around that time do a upscale or downscale of the system such that the session is migrated ? The system should continue to behave normally.

Open Issues:

-

SRR-TS-80: Stop/start an instance on which a session's persistent servlet timer just expired

Priority: P4

Description/Strategy:

1. Ensure that timer expires only once.

Open Issues:

-

3.6. Other

SRR-TS-81: Lazy Creation [FSD section2.3.9.2]

Priority: P5

Description/Strategy:

1. Create Request, Dispatch it and fail instance from where it was dispatched.
2. Now, let dialog be established

Open Issues:

1. Will the session exist before the Dialog state is established ?

SRR-TS-82: Spiraling

Priority: P5

Description/Strategy:

-

Open Issues:

The detail of this TC are unclear yet.

SRR-TS-83: Test Session Replication while using different types of LB – Default Consistent Hash or Specific Data Centric Rules

Priority: P5

Description/Strategy:

-

Open Issues:

The details of this TC are unclear yet.

SRR-TS-84: Need to think up some test case for Session clean up ?

Priority: P4

Description/Strategy:

-

Open Issues:

The details of this TC are unclear yet.

3.7. Multiple failures

SRR-TS-85: Multiple instance failure

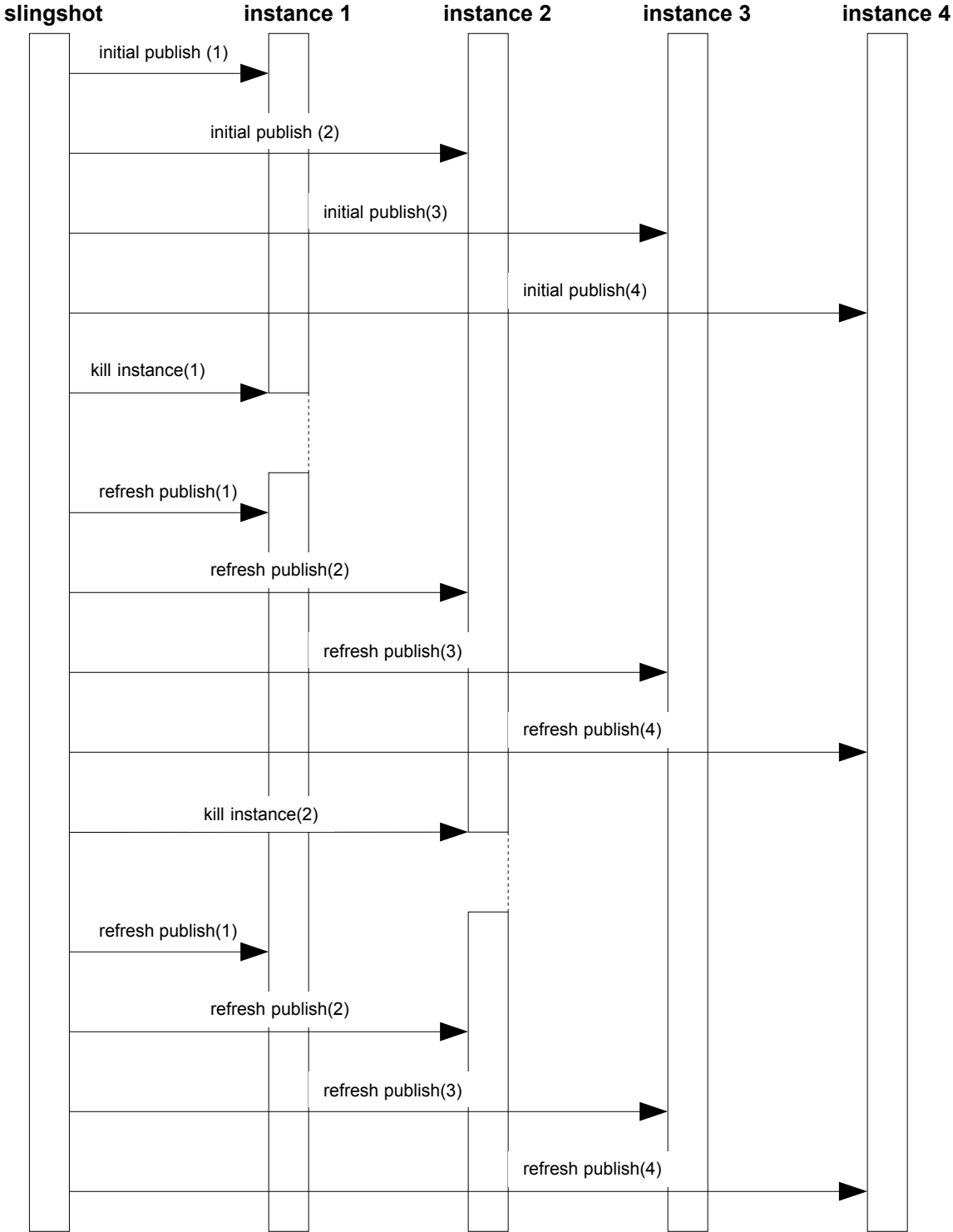
Priority: P3

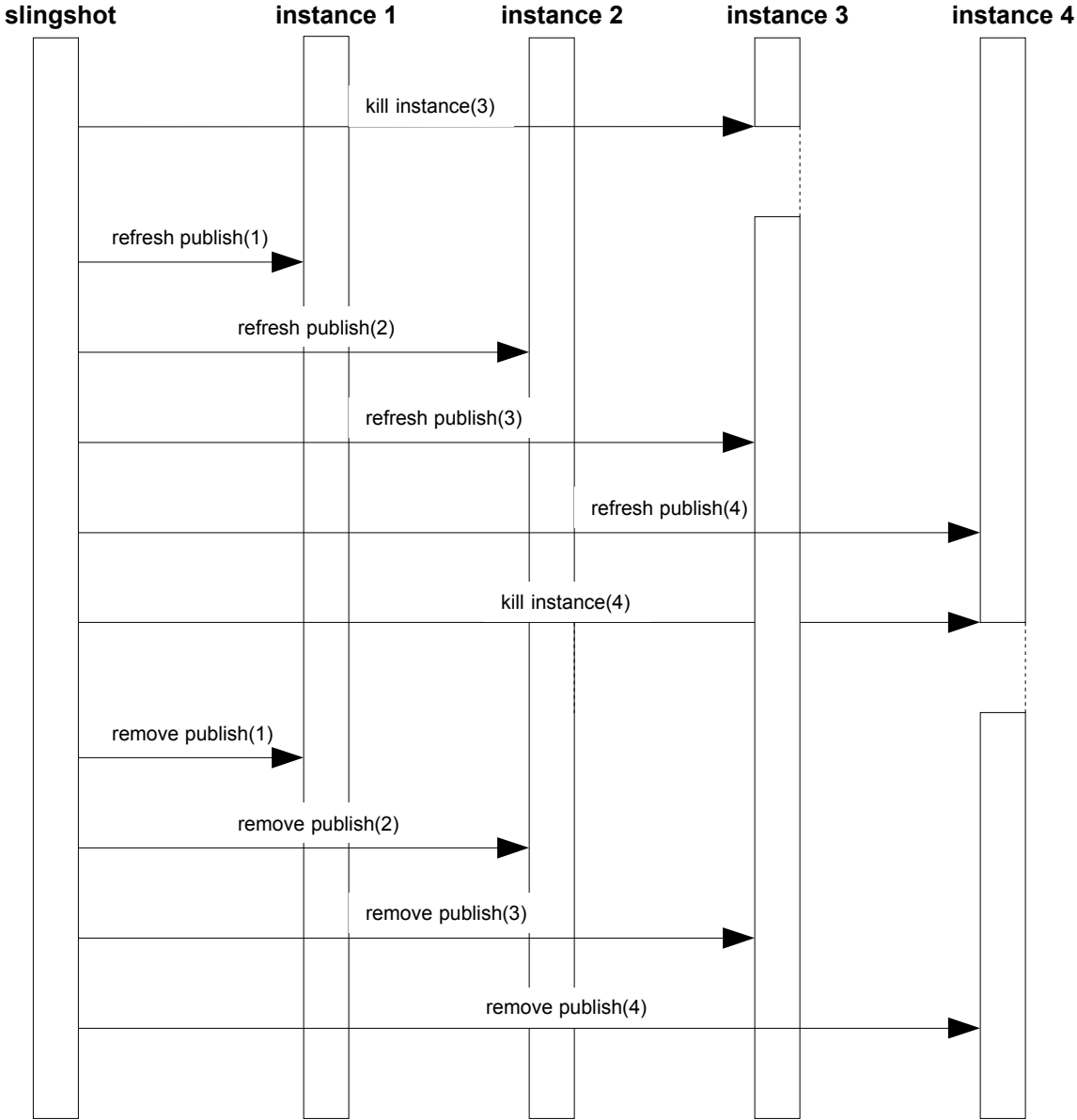
Description/Strategy:

Multiple instance failure one after another ensuring repair in between. The instance failure is caused by killing of the instance process the is immediately restarted by the node agent. The SAS data is not migrated.

Although multiple failures are not high priority, these two tests must be considered P3 since the scenario is realistic and we expect there to be NO Session Loss.

Sip Session Replication Test Specification





Open Issues:

-

SRR-TS-86: Multiple machine failure

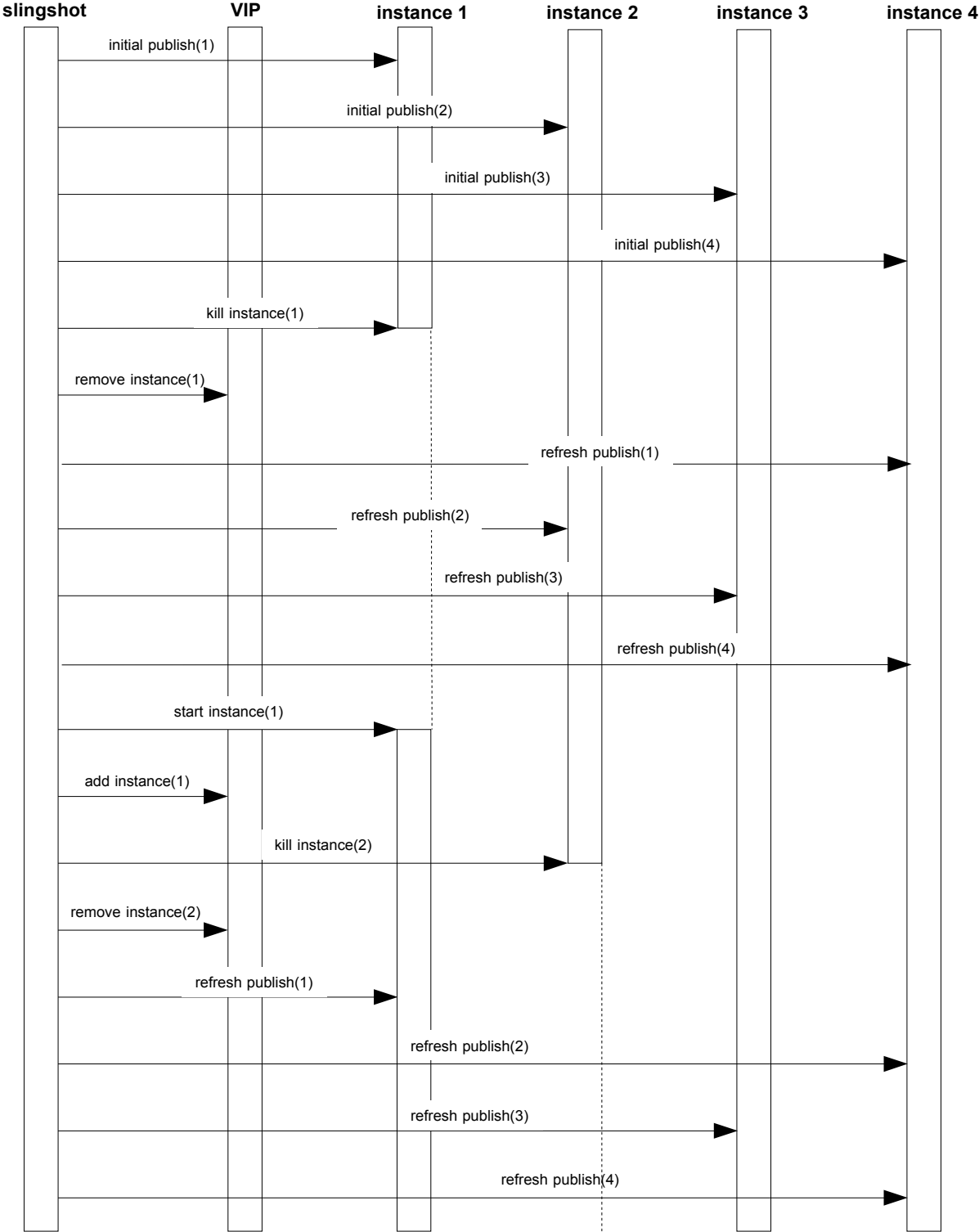
Priority: P3

Description/Strategy:

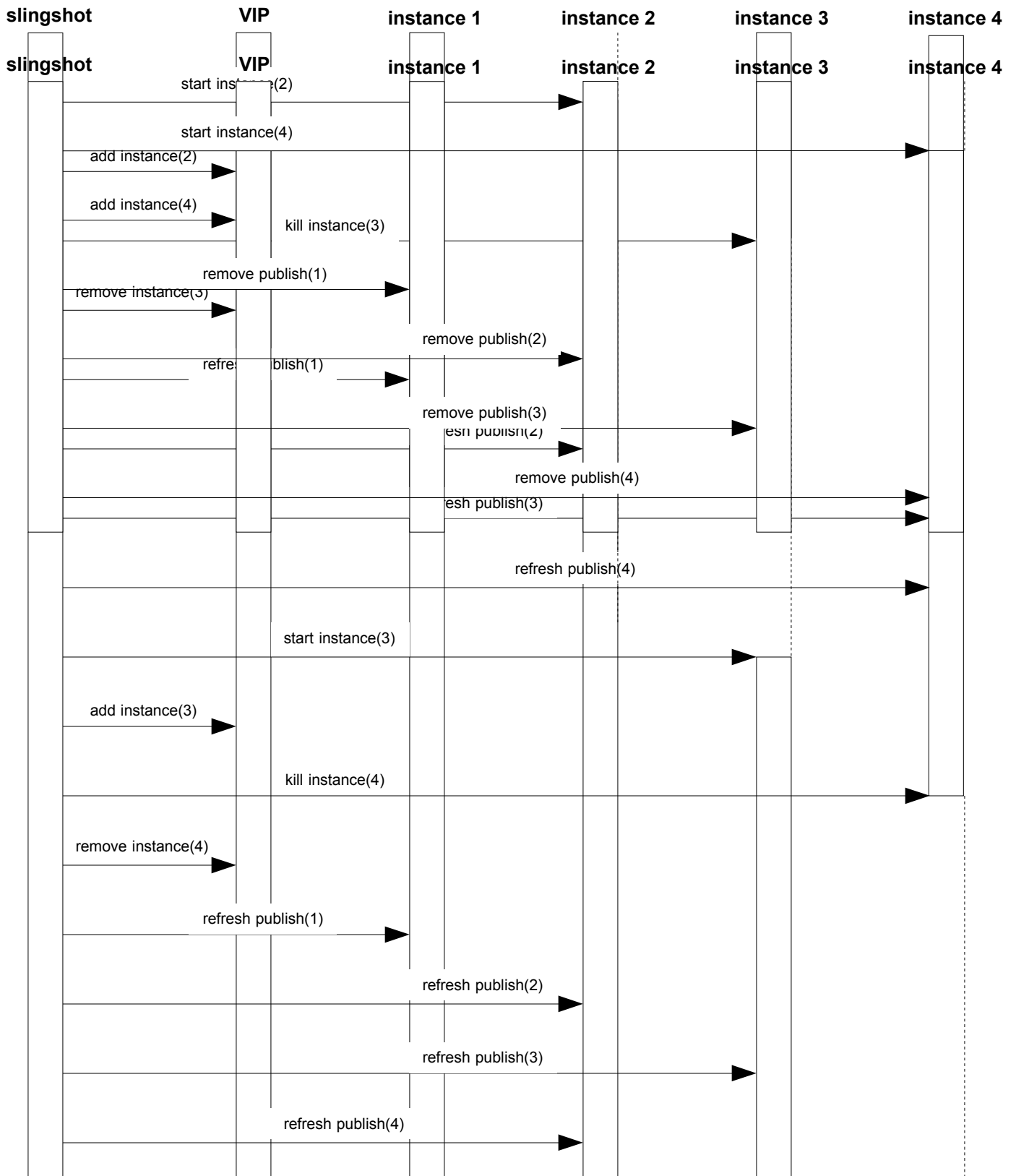
Multiple instance failure one after another ensuring repair in between. The instance failure is caused by killing of the instance process. The SAS data is migrated.

Although multiple failures are not high priority, these two tests must be considered P3 since the scenario is realistic and we expect there to be NO Session Loss.

Sip Session Replication Test Specification



Sip Session Replication Test Specification



Open Issues:

-

4. Approvals/Reviewers

Name	Function	Date	Approval/Reviewer	Comments
<<Quality engineer>>	Quality Engineer		Reviwer	Any buddy engineer should be ok.
<<Engineering counterpart>>	Engineering Module Owner		Approver	Mandatory email needed from the engineering module owner.
Binod P.G./ Prasad Subramanian/ Jagadesh Munta	Lead		Approver	Any one should be ok.
Srikanth Anandal/ Sriram Lakkaraju	Manager		Approver	Optional.

5. References

I. Functional Specification – Sip Session Replication at

<http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/FSD-SSR-2.3.odt>

II. Geek Notes at <http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/SAD-SSR-2.2.odt>