Converged Load Balancer Functional Test Specification

Please send comments to andreas.xx.nyberg@ericsson.com

Author(s): Andreas Nyberg

Contributor(s): Joel Binnquist

Version 0.2

Date: Oct 5th 2007

Table of contents

1 Introduction

1.1 Scope

This document specifies the functional tests for the Convergence Load Balancer (CLB). The test cases have been specified and designed with the functional specification CLB as test base 6.

2 Test Scenarios - Summary

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

Index	*Priority*	*Test scenario/Assertion-ID*	*Comments*
1	P1	CLB-TS-21-1	
2	P1	CLB-TS-21-2	
3	P1	CLB-TS-21-3	
4	P1	CLB-TS-21-20	
5	P1	CLB-TS-21-21	
6	P1	CLB-TS-21-22	
7	P1	CLB-TS-21-23	
TOTAL=7			
8	P2	CLB-TS-21-10	
9	P2	CLB-TS-21-11	
10	P2	CLB-TS-21-12	
11	P2	CLB-TS-21-13	
12	P2	CLB-TS-21-14	
13	P2	CLB-TS-21-16	
14	P2	CLB-TS-21-17	
15	P2	CLB-TS-21-18	
16	P2	CLB-TS-21-19	
17	P2	CLB-TS-21-30	
TOTAL=10			

3 Abbreviations

Instance	An installed SailFin instance
CLB	Converged Load Balancer

3.1 Test base

Functional Specification Converged Load Balancer: 6

3.2 Functional Requirements

The following functional requirements have been identified in the functional test specification for CLB. All requirements have been evaluated for testability and for applicability.

Req Id	**Requirement**	**Description**	**Testable**	**Testability (0-5)**

LLR-CLB-1	Sticky request	Existing SIP dialogs gets tagged as sticky so that subsequent requests can be handled by the same instance to keep session states intact. The CLB shall make sure that the request ends up on the right listener on the right instance in the right cluster.	Yes	
LLR-CLB-2	Sticky request, unhealthy instance	Sticky dialog where the instance to handle the request is unhealthy. The CLB shall be able to make a decision where to send the request instead, same cluster but the next healthy instance.	Yes	2
LLR-CLB-3	Load balancing	The CLB can forwards requests to instances on multiple clusters and a single cluster can receive requests from multiple CLB's.	Yes	
LLR-CLB-4	CLB runtime, supported protocols	The CLB shall be able to handle request matching for HTTP and SIP requests.	SIP only	
LLR-CLB-5	CLB runtime, request matching	The CLB shall be able to determine whether the request is new or sticky by looking in the SIP header Route: or in the SIP-URI.	Yes	
LLR-CLB-6	CLB runtime, forwarding of new requests	New requests shall be forwarded according to the LB policy.	Yes	
LLR-CLB-7	CLB runtime, forwarding of subsequent requests	The CLB shall be able to determine the the instance WHERE to send the subsequent request based on the BEKey. The request shall then be sent to the handling instance or do a fail-over/quiescing in case of unhealthyness.		
LLR-CLB-8	CLB runtime, determining response stickiness	Shall be able to determine stickiness by looking for the BERoute in the topmost Via:	Yes	
LLR-CLB-9	CLB runtime, logging messages		No	0
LLR-CLB-10	SIP sessions	A session shall be tied to a particular cluster. Sessions shall only be failed over to instances in the same cluster.	Yes	
LLR-CLB-11	Self load balancing	A cluster shall be able to load balance it self or load balances a backend cluster.	Yes	
LLR-CLB-12	Multiple CLBs	A SailFin cluster can be compromised by one or more CLBs.	Yes	
LLR-CLB-13	Multiple listeners on an instance	An instance can have multiple listeners where incoming requests are served.		
LLR-CLB-14	Application deployment	Homogenous deployment of applications in all clusters and on all instances.	No	
LLR-CLB-15	Application deployment, disable	It shall be possible to disable an application on one instance.	No	
LLR-CLB-16	Load balancing policy/rules	A CLB shall be able to handle a hash based policy or data centric rules.	Yes	
LLR-CLB-17	Quiescing	The CLB shall be able to handle quieshing which in practice handles the case where a CLB will disable an instance that gracefully disables itself. Correct timeouts shall be set in the CLB that shall handle the following scenarios: A) the instance timeout has not expired then all sticky requests shall still be handled by the disabling instance. All new requests shall be sent to the next instance determined by the LB policy. B) when timer has expired fail over policy shall be applied.	Yes	
LLR-CLB-18	Monitoring health of instances	The CLB shall be responsible for its own list that states the healthy instances a cluster.	No	
HLR-CLB-1	SIP container	The SIP-container shall be able to make the correct sticky stamping in requests and responses.	Yes	
LLR-CLB-20	SIP container, stamping responses	Responses shall have the Contact: header stamped with BEKey for all responses sent by a SIP container.	Yes	

LLR-CLB-21	SIP container, stamping proxied requests	Record-Route header shall be stamped with a BEKey for all proxied requests sent by a SIP container.	Yes	
LLR-CLB-22	SIP container, outgoing requests non dialog creating	The top-most Via: shall be stamped with a BERoute for all outgoing requests sent by a SIP-container.	Yes	
LLR-CLB-23	SIP container, outgoing requests dialog creating	The Contact header shall be stamped with a BEKey for all initail dialog creating requests sent by the SIP-container.	Yes	

3.3 Test environment

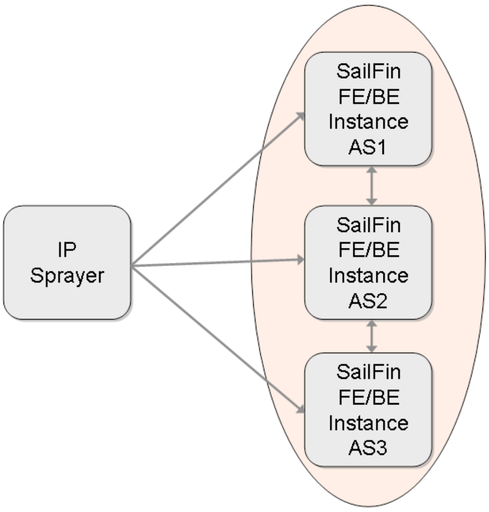
The test environment is described in 6.

3.3.1 Setups

This section describes the different setups that are used through out the test cases in this specification.

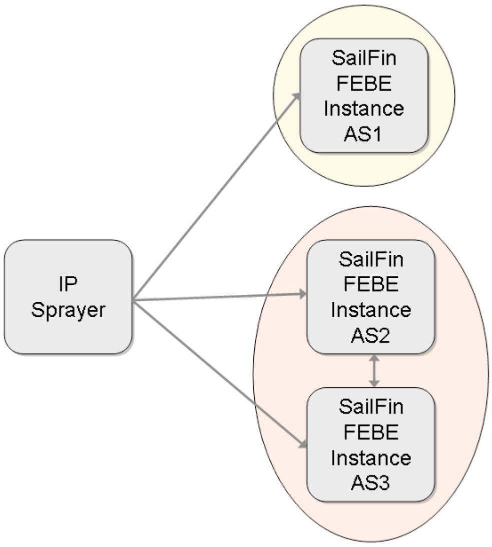
3.3.1.1 A) Three FE/BE instances in a single cluster

All instances have their own load balancer.


IP Sprayer
SailFin FE/BE Instance AS1
SailFin FE/BE Instance AS2
SailFin FE/BE Instance AS3

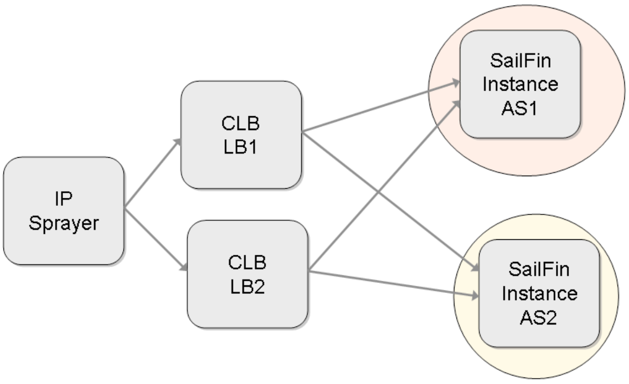
3.3.1.2 B) The FE/BE instances in two clusters

All instances have their own load balancer.


SailFin FEBE Instance AS1
IP Sprayer
SailFin FEBE Instance AS2
SailFin FEBE Instance AS3

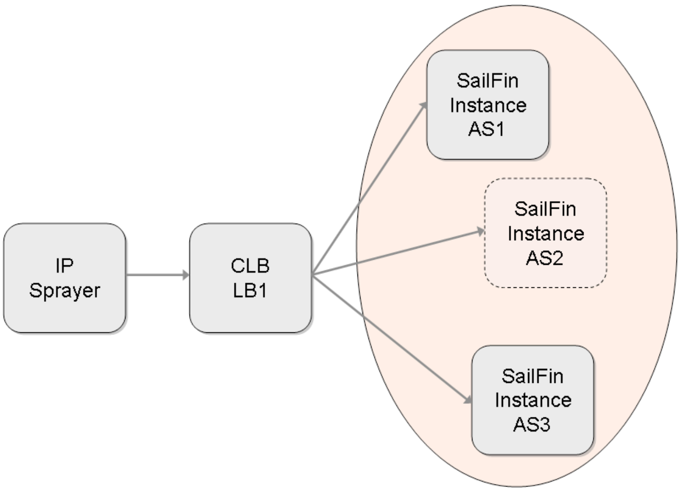
3.3.1.3 C) Many to many

Two front load balancers distribute requests evenly between the clusters and instances.



3.3.1.4 D) One to many

One fronted load balancer distributes requests evenly to all instances in a cluster. The load balancer shall be able to keep track of the “health” of the instances in the cluster.



4 Test cases

The test cases are divided in to two sections. Specification based and use case scenario based.

4.1 Use case based test cases

This section covers the test cases that have been derived from the use case scenarios in the test base. There must be an IP sprayer placed in front of the cluster.

4.1.1 NI CLB-TS-21-1 External party is the initiator

Description: Typical SIP session (establish dialog, re-invite and close) where an external party is the initiator and an application on the TS acts as endpoint. The IP sprayer in from of the cluster shall dispatch responses and subsequent requests to different instances in the cluster so that the front end (FE)/load balancers needs to re-route or proxy.

Requirement: Use case 8.1.1

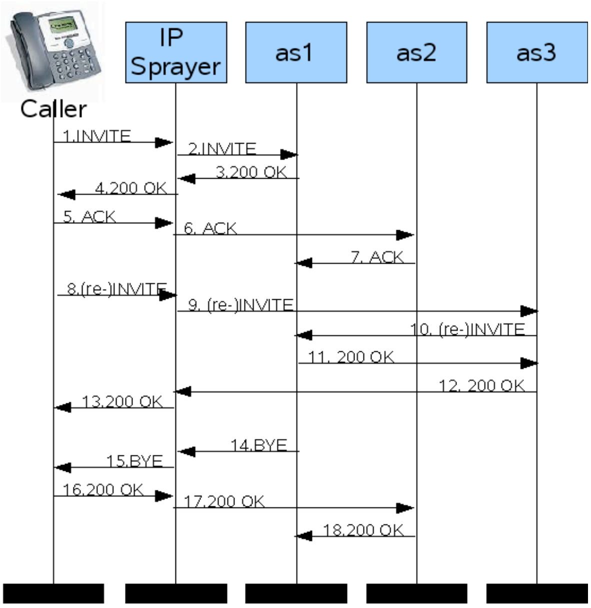


Figure 1: A possible message flow for test case 20.1

Setup: A (3.3.1.1)

Test instructions: Use test setup A.

Verify: That the session stickiness holds for the entire test case. All subsequent requests (re-INVITE) and responses (200 (BYE)) shall be forwarded to the initially serving instance

Scenarios: TDB

4.1.2 NI CLB-TS-21-2 An Application in TS is the initiator

Description: Typical SIP session (establish dialog, re-invite and close) where an application in the TS is the initiator (UAC) and the test client acts as endpoint. The IP sprayer in from of the cluster shall dispatch responses and subsequent requests to different instances in the cluster so that the front end (FE)/load balancers needs to re-route or proxy. All communication is handled over UDP.

Requirement: Use case 8.1.2

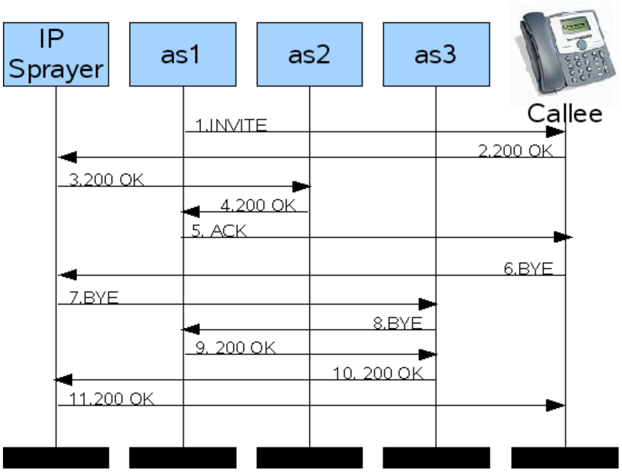


Figure 2: A possible message flow for test case 20.2

Setup: A (3.3.1.1)

Test instructions: Use test setup A.

Verify: That the session stickiness holds for the entire test case. All subsequent requests (BYE) and responses (200(INVITE), 200 (BYE)) shall be forwarded to the initially serving instance

Scenarios: TDB

4.1.3 NI CLB-TS-21-3 TS is Proxy

Description: The cluster acts as a proxy between caller and callee. All messages are distribute in to the cluster through the IP sprayer. All communication is handled over UDP.

Requirement: Use case 8.1.3

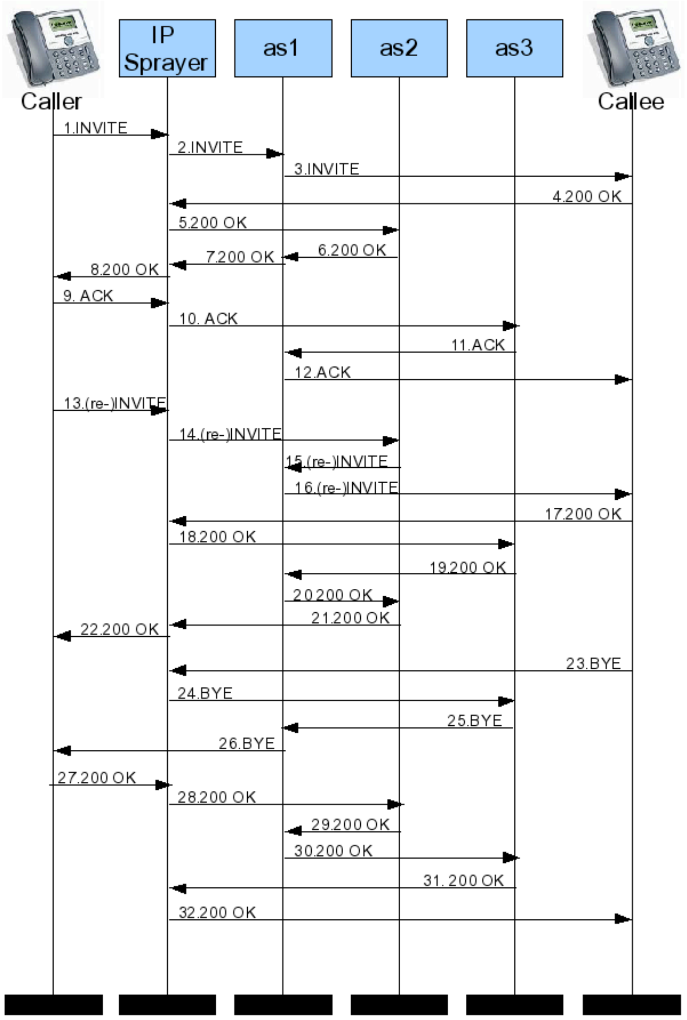


Figure 3: A possible message flow for test case 20.3

Setup: A (3.3.1.1)

Test instructions: Use test setup A.

Verify: That the session stickiness holds for the entire test case. All subsequent requests and responses shall be forwarded to the instance acting as the proxy.

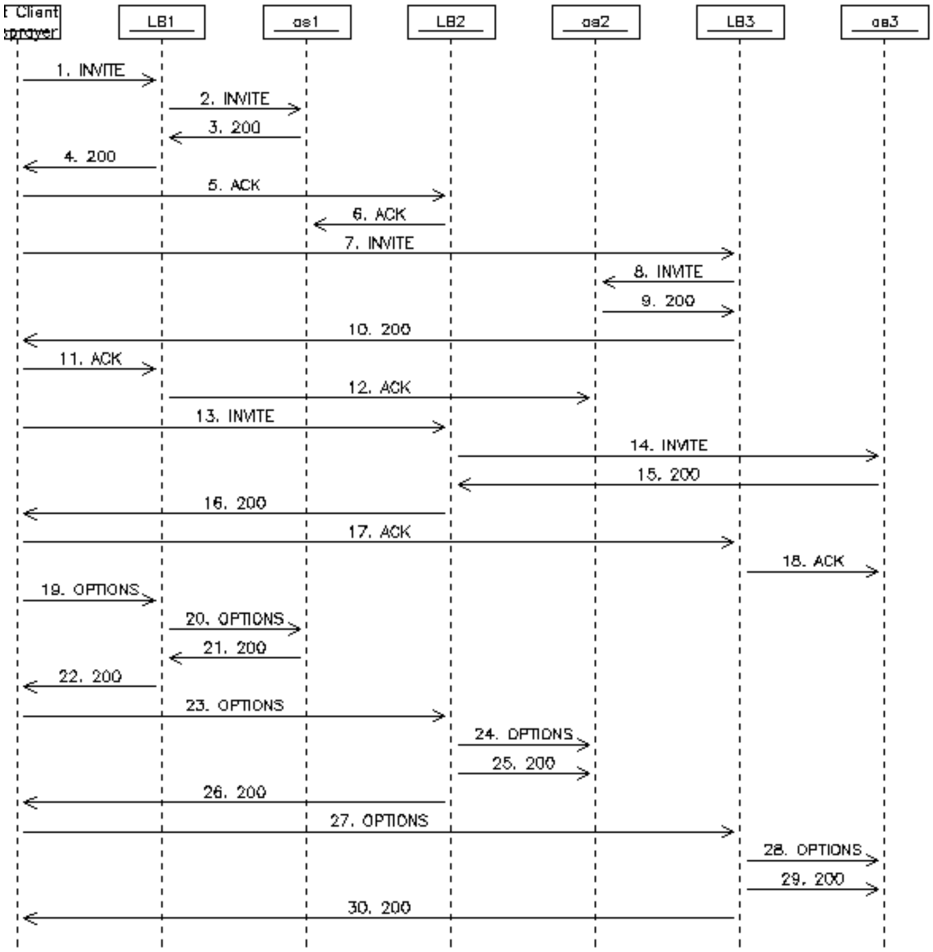
Scenarios: TDB

4.2 Specification based test cases

The requirements derived from the functional specification are listed in section 3.2.

4.2.1 NI CLB-TS-21-10 Sticky requests setup B

Description: This test case test that sticky requests end up on the original instance for setups according to setup B. There is one stand alone load balancer and two clusters involved. It shall be verified that multiple clusters does not confuse the load balancer.



Requirement: LLR-CLB-1 -> (3.2)

Setup: B (3.3.1.2)

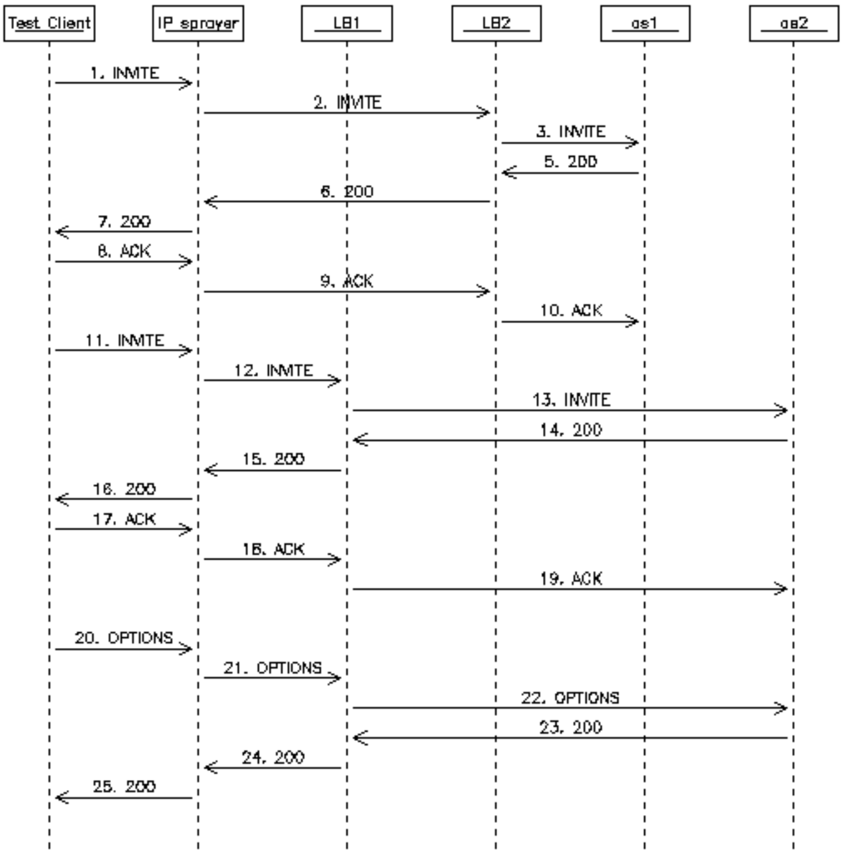
Test instructions: close all sessions by sending a BYE from the test client after message 30.

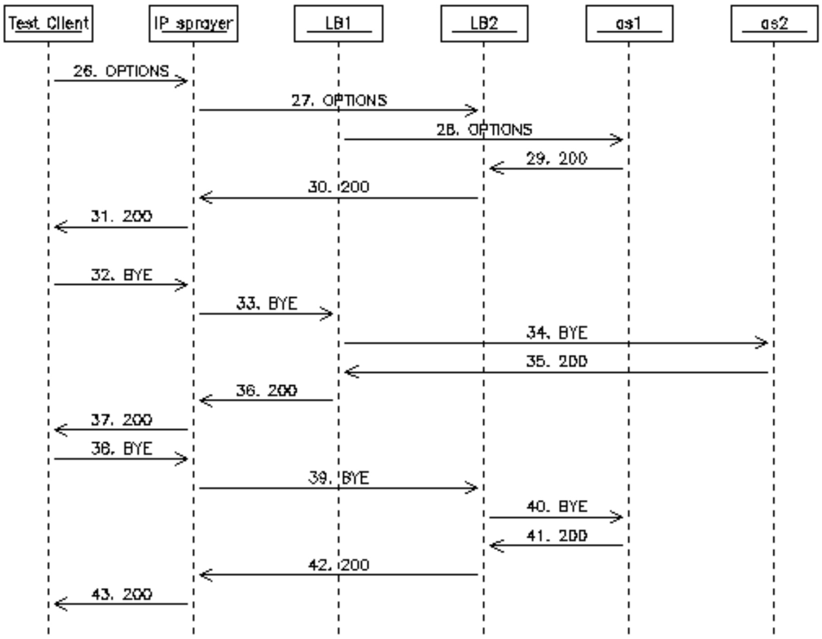
Verify: That subsequent requests (OPTIONS) end up on the original instance.

Scenarios: TDB

4.2.2 NI CLB-TS-21-11 Sticky requests setup C

Description: This test case test that sticky requests end up on the original instance for setups according to setup C which includes multiple stand alone load balancers and multiple clusters. It shall be verified that multiple clusters does not confuse the load balancers.


Test Client
IP sprayer
LB1
LB2
as1
as2
1. INVITE
2. INVITE
3. INVITE
5. 200
6. 200
7. 200
8. ACK
9. ACK
10. ACK
11. INVITE
12. INVITE
13. INVITE
14. 200
15. 200
16. 200
17. ACK
18. ACK
19. ACK
20. OPTIONS
21. OPTIONS
22. OPTIONS
23. 200
24. 200
25. 200


Test Client
IP sprayer
LB1
LB2
as1
as2
26. OPTIONS
27. OPTIONS
28. OPTIONS
29. 200
30. 200
31. 200
32. BYE
33. BYE
34. BYE
35. 200
36. 200
37. 200
38. BYE
39. BYE
40. BYE
41. 200
42. 200
43. 200

Requirement: LLR-CLB-1 -> (3.2)

Setup: C (3.3.1.23.3.1.3)

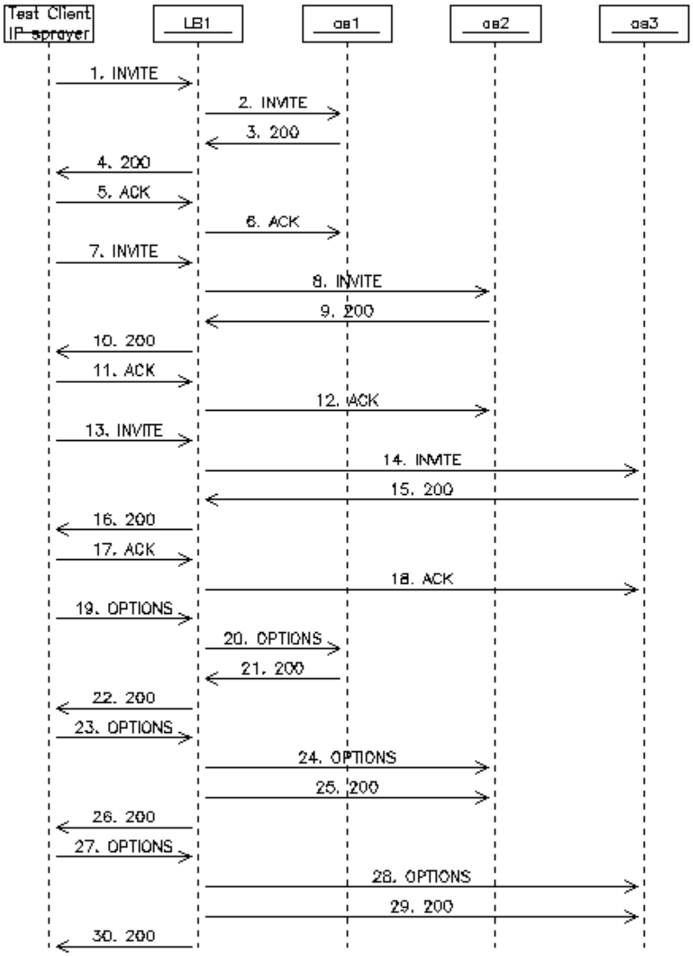
Test instructions: -

Verify: That subsequent requests ends up on the original instance.

Scenarios: TDB

4.2.3 NI CLB-TS-21-12 Sticky requests setup D

Description: This test case test that sticky requests end up on the original instance for setups according to setup D which includes one stand alone load balancer and a cluster with multiple instances. It shall be verified that multiple instances does not confuse the load balancers.



Requirement: LLR-CLB-1 -> (3.2)

Setup: D (3.3.1.4)

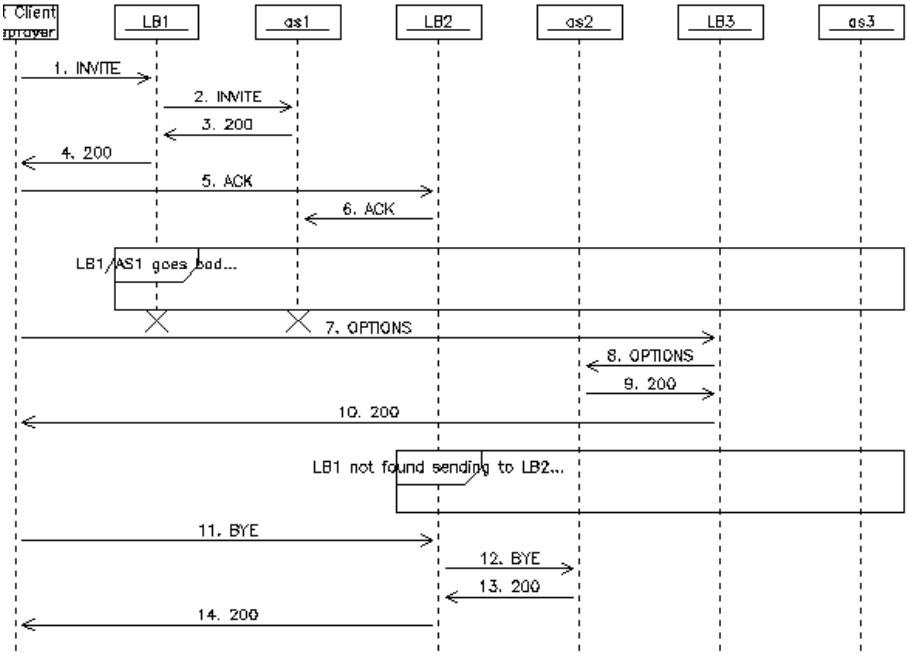
Test instructions: -

Verify: That subsequent requests ends up on the original instance.

Scenarios: TDB

4.2.4 NI CLB-TS-21-13 Fail-over

Description: This test case test that sticky requests end up on a correct instance in the case that the original instance becomes unhealthy.



Requirement: LLR-CLB-2 -> (3.2)

Setup: A (3.3.1.1)

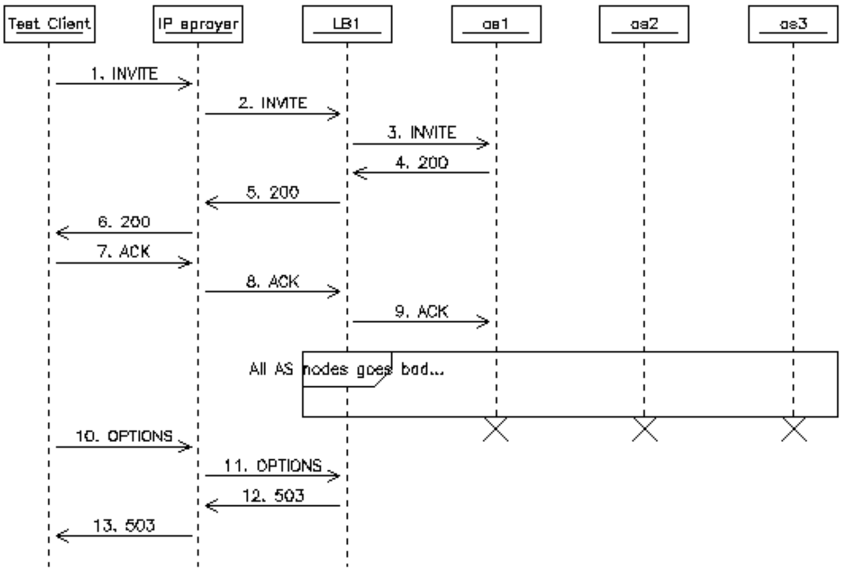
Test instructions: -

Verify: That no 503's are returned and that subsequent request still works.

Scenarios: TDB

4.2.5 NI CLB-TS-21-14 Fail-over no healthy instances

Description: This test case tests sticky requests and where the original instance goes bad and that there are no other instances that can take over the session.



Requirement: LLR-CLB-2 -> (3.2)

Setup: D (3.3.1.4), UDP transport

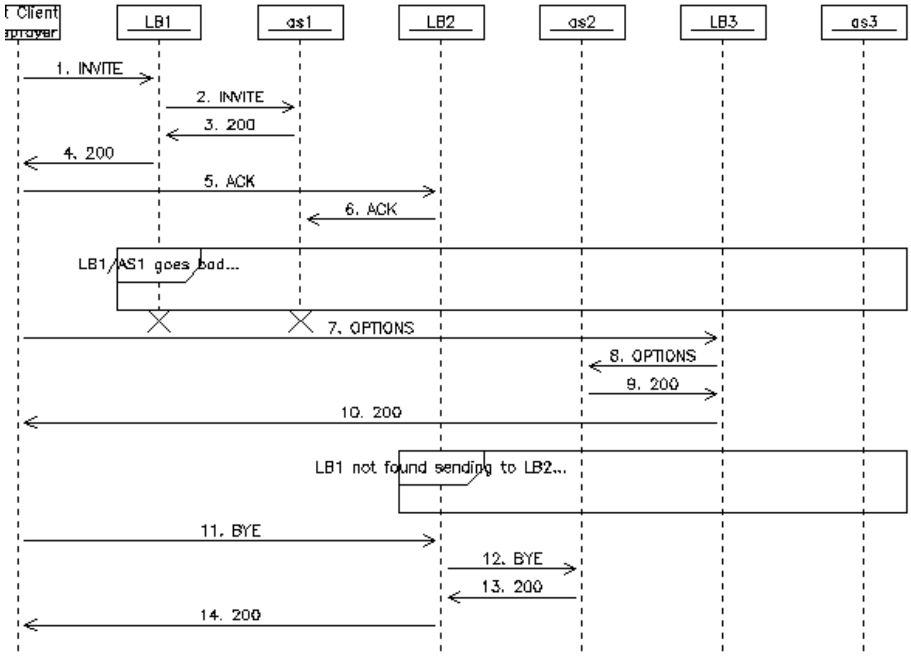
Test instructions: Take down all instances in the cluster during the test case.

Verify: Verify that the front load balancer can handle the scenario and that 503's are returned for all subsequent requests.

Scenarios: TDB

4.2.6 NI CLB-TS-21-16 Fail-over strict check according to BEKey

Description: This test case tests sticky requests and where the original serving instance goes bad. A subsequent request shall end up at specific instance which shall be determined by the value in the BEKey. It shall be made sure that the instance that actually takes over the session is the one predetermined in the BEKey value.



Requirement: LLR-CLB-7 -> (3.2)

Setup: A (3.3.1.1)

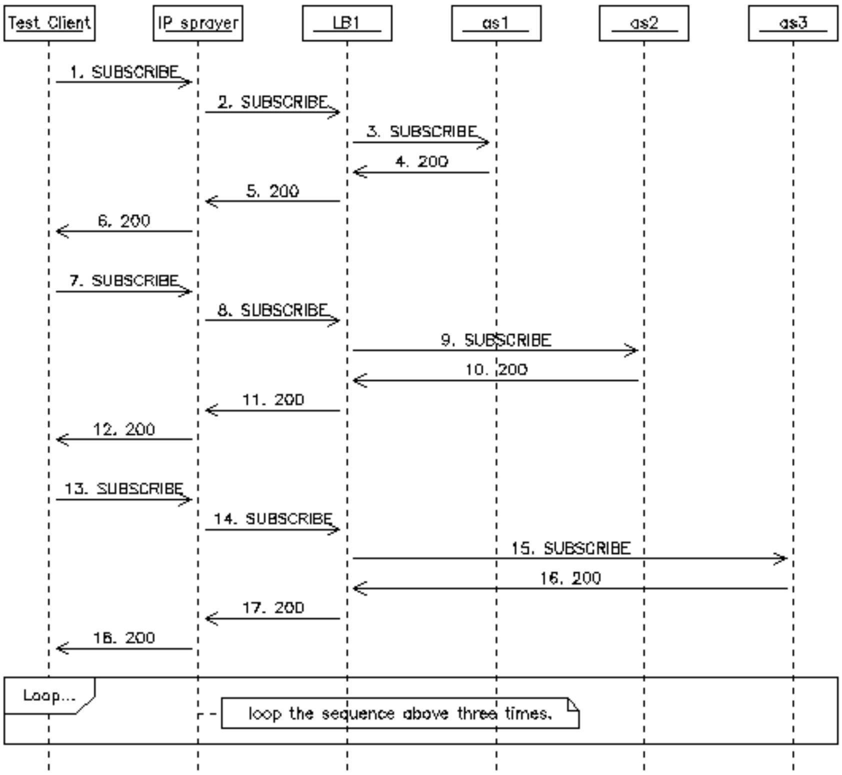
Test instructions: Take down a predefined instance in the cluster.

Verify: That the subsequent requests end up on the right instance, not any instance.

Scenarios: TDB

4.2.7 NI CLB-TS-21-17 Load balancing, even distribution, to all clusters to all instances setup B using data centric policy

Description: This test case tests that load balancing is working as expected using the Data Centric6 load balancing policy. All new requests shall be distributed evenly between all instances in all clusters.



Requirement: LLR-CLB-3 -> (3.2)

Setup: B (3.3.1.2)

Test instructions: -

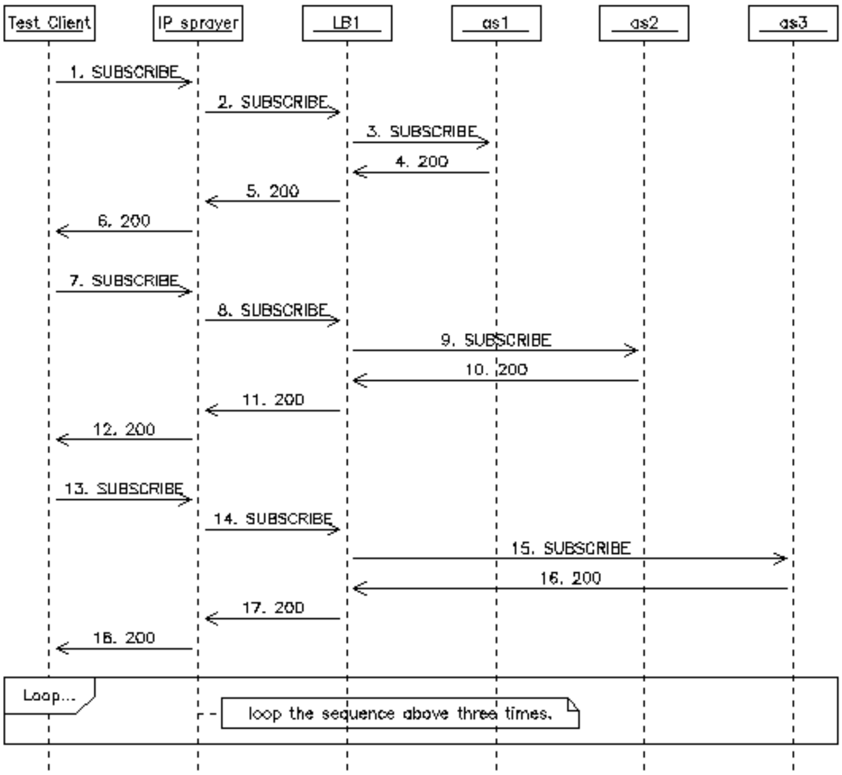
Verify: That all new requests are evenly distributed between all instances within all clusters. For setup B, 9 new requests shall be sent and the distribution shall be accordingly.

Instance	New Requests
AS1	3
AS2	3
AS3	3

Scenarios: TDB

4.2.8 NI CLB-TS-21-18 Load balancing, even distribution, to all clusters to all instances setup B using basic hash policy

Description: This test case tests that load balancing is working as expected using the basic hash load balancing policy. All new requests shall be distributed evenly between all instances in all clusters.



Requirement: LLR-CLB-3 -> (3.2)

Setup: B (3.3.1.2)

Test instructions: -

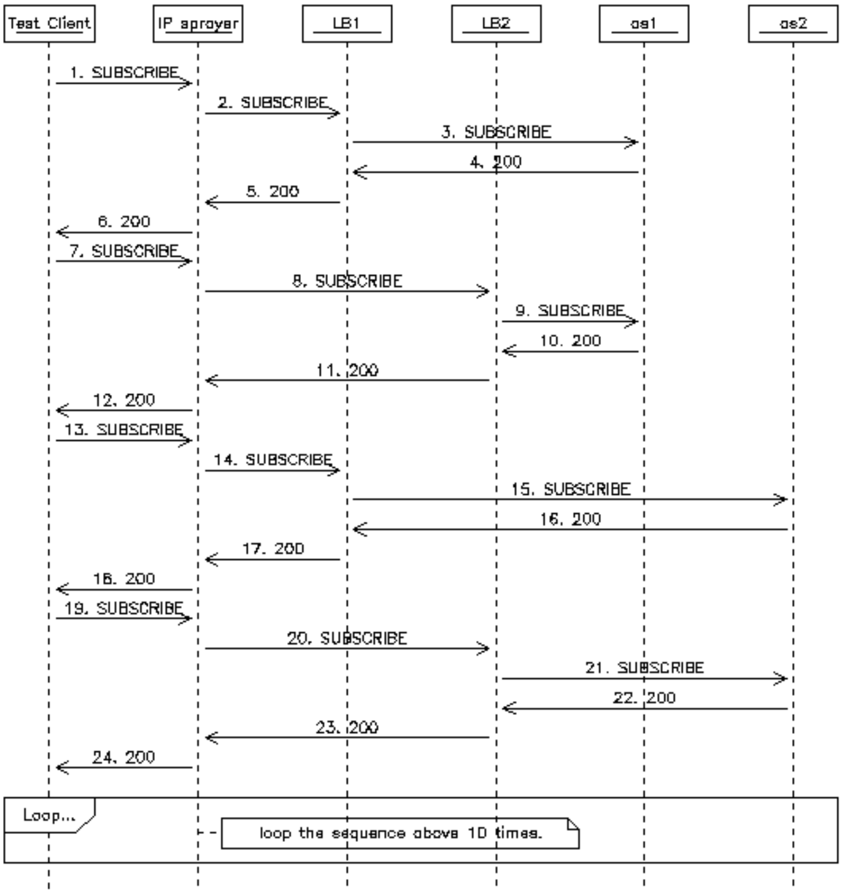
Verify: That all new requests are evenly distributed between the clusters and all instances within a cluster. For setup B, 9 new requests shall be sent and the distribution shall be accordingly.

Instance	New Requests
AS1	3
AS2	3
AS3	3

Scenarios: TDB

4.2.9 NI CLB-TS-21-19 Load balancing, even distribution, to all clusters to all instances setup C

Description: This test case tests that load balancing is working as expected using the Data Centric6 load balancing policy. All new requests shall be distributed evenly between all instances in all clusters from all involved load balancers.



Requirement: LLR-CLB-3 -> (3.2)

Setup: C (3.3.1.3)

Test instructions: -

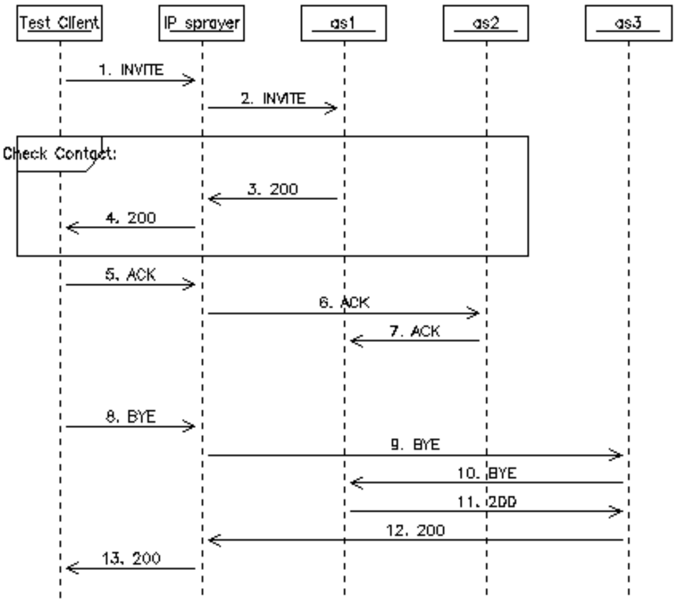
Verify: That all new requests are evenly distributed between all instances within all clusters. For setup C, 40 new requests shall be sent and the distribution shall be accordingly.

Instance	New Requests
AS1	20
AS2	20

Scenarios: TDB

4.2.10 NI CLB-TS-21-20 Sip container stamping requests

Description: Test that the *Contact* header has been stamped with the *BEKey* in responses from a SIP-container callee instance.



Requirement: LLR-CLB-20 -> (3.2)

Setup: A (3.3.1.1)

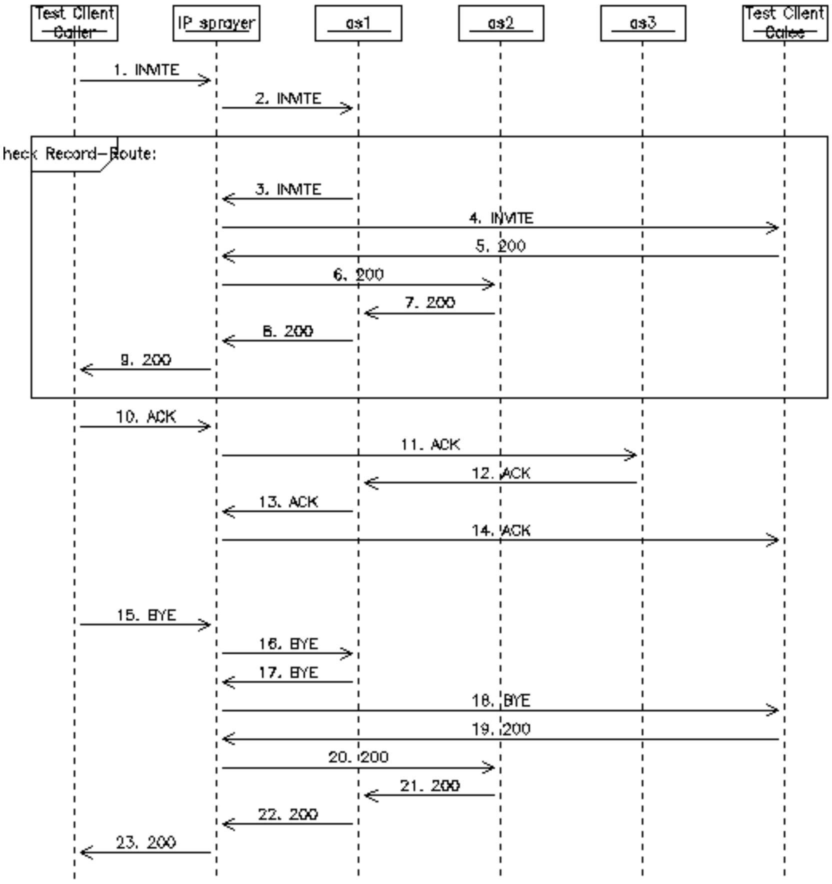
Test instructions: -

Verify: Verify that the 200 response (message 4) has the *Contact* header correctly stamped with a *BEKey*.

Scenarios: TBD

4.2.11 NI CLB-TS-21-21 Sip container stamping proxied requests

Description: Test that the *Record-Route* header has been stamped with the *BEKey* in requests that are proxied by a SIP-container.



Requirement: LLR-CLB-21 -> (3.2)

Setup: A (3.3.1.1)

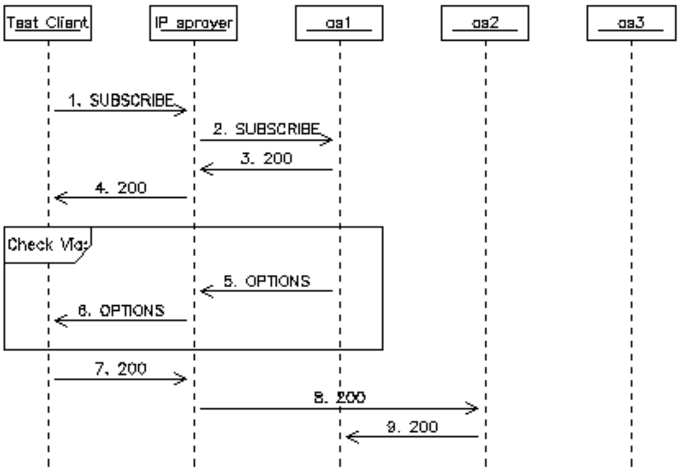
Test instructions: -

Verify: Verify that returned 200 (message 9) have the *Record-Route* header correctly stamped with a *BEKey*.

Scenarios: TBD

4.2.12 NI CLB-TS-21-22 Sip container stamping outgoing requests

Description: This test case tests that non dialog creating outgoing requests have their top most *Via* header stamped with the *BERoute* parameter.



Requirement: LLR-CLB-22 -> (3.2)

Setup: A (3.3.1.1)

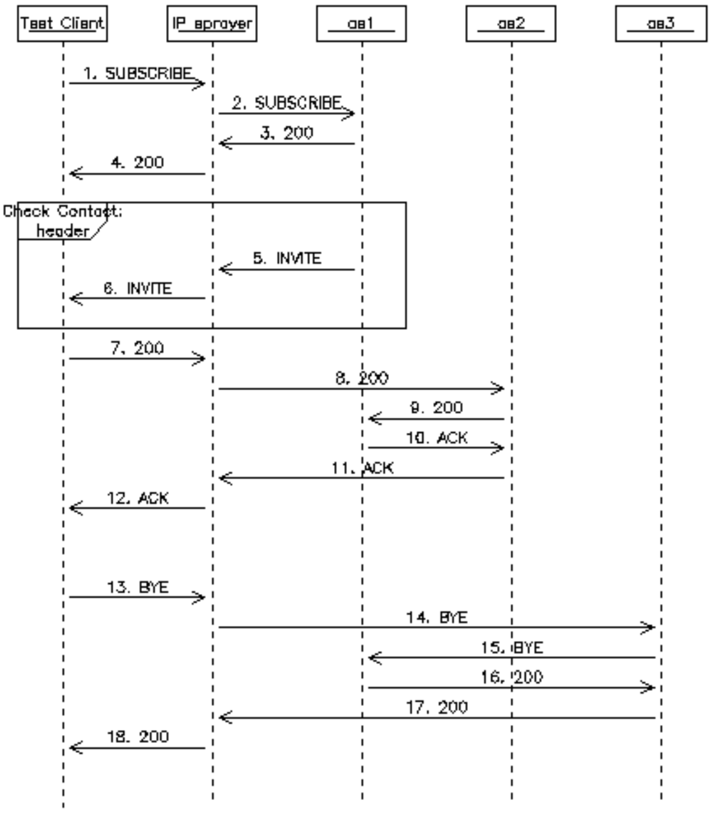
Test instructions: -

Verify: That the top most *Via* header is stamped with the *BERoute* parameters in the outgoing request (message 6).

Scenarios: TDB

4.2.13 NI CLB-TS-21-23 Sip container stamping subsequent requests

Description: This test case tests that any initial dialog creating request sent from a SIP-container caller has the *Contact* header stamped with the *BEKey* parameter.



Requirement: LLR-CLB-22 -> (3.2)

Setup: A (3.3.1.1)

Test instructions: Setup as SIP session with the *caller* inside a SailFin cluster and make sure the *caller* does a subsequent request.

Verify: That the *Contact* header has been stamped with the *BEKey* parameter in the request (message 6).

Scenarios: TDB

4.2.14 NI CLB-TS-21-30 Cluster-wide Failover

Description: A front end machines handles traffic for two clusters, cluster 1 and cluster 2. Cluster 1 is handling requests for a particular user but is taken offline for some reason. The front end realizes this and routes all subsequent traffic to cluster 2. Later on back end 1 of cluster 2 is taken offline and so back end 2 of cluster 2 should take over the traffic the previously was routed to back end 1 of cluster 2 (phew). (The yellow notes in the picture below represent the consistent hash tables)

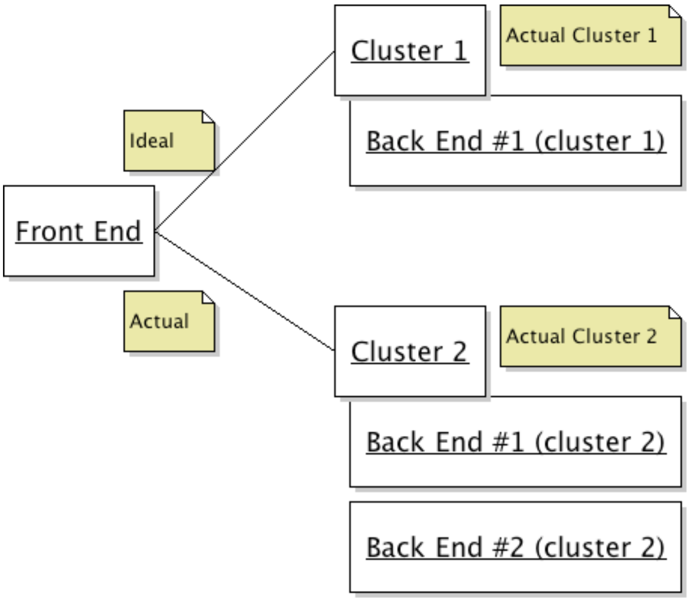


Figure 4: Setup for Cluster Wide failover test case

Requirement: LLR-CLB-22 -> (3.2)

Setup: Figure 4.

Test instructions: Setup as SIP session with the *caller* inside a SailFin cluster and make sure the *caller* does a subsequent request.

Verify: That the *Route* header has been stamped with the *BEKey* parameter in subsequent requests.

Scenarios: TDB

5 Approvals / Reviewers

Name	*Function*	*Date*	*Approval/Reviewer*	*Comments*
Joel Binnquist	Quality Engineer		Reviewer	
Jens Tinfors	Test Engineer		Reviewer	
<<Engineering counterpart>>	Engineering module owner		Approver	
	Lead		Approver	
	Manager		Approver	

6 References

[1] Functional Specification Converged Load Balancer
Document url:
http://wiki.glassfish.java.net/PageInfo.jsp?page=FunctionalSpecsOnePagers/ConvergedLB-FSD.pdf

[2] Functional Test Environment
Document url: Ericsson internal.

[3] Data Centric specification
Document url: Ericsson internal