XORP PIM-SM Test Suite

Version 0.1

XORP Project
International Computer Science Institute
Berkeley, CA 94704, USA
feedback@xorp.org

December 11, 2002

Modification History

Version 0.1 completed	December 5, 2002	`draft-ietf-pim-sm-v2-new-05.{ps,tex}` `draft-ietf-pim-sm-bsr-02.{ps,tex}`

Authors

Pavlin Radoslavov XORP Project, ICSI pavlin@icsi.berkeley.edu

Contents

List of Tables

List of Figures

Chapter 0

Introduction

0.1 Overview

TODO

This document describes a set of tests for evaluating Protocol Independent Multicast - Sparse Mode implementation.

0.2 Test Software

TODO

0.3 Acronyms

Acronyms used in this Test Suite:

- **LAN**: **L**ocal **A**rea **N**etwork
- **M**: **M**onitor or packet capturer
- **MRT**: **M**ulticast **R**outing **T**able
- **MRE**: **M**ulticast **R**outing **E**ntry
- **N**: **N**etwork
- **NBMA**: **N**on-**B**roadcast **M**ulti-**A**ccess
- **RTE**: **R**outing **T**able **E**ntry
- **RUT**: **R**outer **U**nder **T**est
- **Rx**: **R**eceiver
- **S**: **S**ource
- **TN**: **T**esting **N**ode
- **TR**: **T**esting **R**outer

When several entities of the same type are present in a test configuration, a number is appended to the acronym to yield a label for each entity. For example, if there were three testing routers in the test configuration, they would be labeled TR1, TR2, TR3.

0.4 Definitions

TODO

0.5 Timers and Default Values

PIM-SMv2 defines several timers and default values. For the purpose of testing, all configurable timers and values are set to their defaults, unless otherwise noted in the test description. These defaults are given here for reference, taken or calculated from the PIM-SM spec.

- PIM-SM protocol spec:

`PIM_SM_version_default`	2
`LAN_delay_default`	0.5 sec
`t_override_default`	2.5 sec
`Hello_Period`	30 sec
`Triggered_Hello_Delay`	5 sec
`Default_Hello_Holdtime`	105 sec
`Hello_Holdtime`	105 sec
`J/P_HoldTime`	210 sec
`J/P_Override_Interval(I)`	3 sec
`Assert_Override_Interval`	3 sec
`Assert_Time`	180 sec
`t_periodic`	60 sec
`t_suppressed`	rand(66,84) sec
`t_override`	rand(0,2.5) sec
`Keepalive_Period`	210 sec
`RP_Keepalive_Period`	185 sec
`Register_Suppression_Time`	60 sec
`Register_Probe_Time`	5 sec

- Bootstrap mechanism spec:

`BS_Period`	60 sec
`BS_Timeout`	130 sec
`rand_override`	rand(0, 5.0) sec
`C-RP_Timeout`	150 sec
`C-RP-Adv_Period`	60 sec
`SZ_Timeout`	1500 sec

0.6 Test Organization

TODO

0.7 References

The following documents are referenced in this text:

- RFC 2362 – Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification
- `draft-ietf-pim-sm-v2-new-05.{ps,tex}` – Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)
- `draft-ietf-pim-sm-bsr-02.{ps,tex}` – Bootstrap Router (BSR) Mechanism for PIM Sparse Mode

0.8 Acknowledgments

The format and notation in this test-suite is largely based on the format of the IP Consortium Test Suite developed by the InterOperability Laboratory at the University of New Hampshire.

Test Group 1

Interoperability

Scope: The following tests verify the general operation of a PIM-SM router and are not specific to any single section of the specification.

Overview: These tests have been designed to test the Interoperability of the RUT with other PIM-SM capable devices. This test group focuses on testing configurations of the network that could cause problems when deployed if the RUT does not operate properly with the devices that it is connected to. The tests in this group do not determine if a product conforms to the PIM-SM standard but they are designed as interoperability tests. The test routers in this section are complete implementation of PIM-SM.

Please note that in the case of interoperability tests, failure against any other router does not necessarily indicate nonconformance. Rather, it indicates that the two routers are unable to work “properly” together and that further work should be done to isolate the cause of the failure.

1.1 Basic Interoperability

Purpose: To ensure that a router can interoperate with other PIM-SM implementation in a multicast network.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4

Discussion: PIM capable routers detect each other with PIM Hello messages and exchange information for their routing tables through PIM Join/Prune messages. For each multicast group there is a single PIM-SM router that is the Rendezvous Point (RP) for that group. PIM-SM routers that are connected to group members send PIM Join messages toward the RP to create the group-specific shared tree. A PIM-SM router that is directly connected to a multicast source, encapsulates the multicast data in PIM Register messages which are unicast to the RP for that group. The RP decapsulates the PIM Register messages and forwards them down the group-specific shared tree to all receivers for that group.

Test Setup: Connect the RUT, TR1, TN1, and TN2 according to Figure 1.1. Enable PIM-SM on both the RUT and TR1.

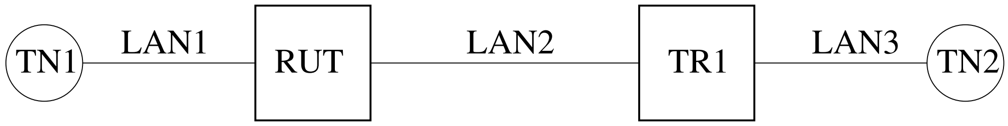

Figure 1.1: Basic interoperability test setup

Procedure:

Part A: The RUT is the RP, TN1 is the receiver, and TN2 is the sender.

Configure both the RUT and TR1 such that the RUT is the RP for group 224.0.1.20. This configuration can be either manual, or implicit through the Bootstrap mechanism. Configure TN1 and TN2 such that TN1 is the receiver, and TN2 is the sender, both for group 224.0.1.20.

1. Start both the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start the receiver and the sender.
3. Observe the data packets received by the receiver.

Part B: TR1 is the RP, TN1 is the receiver, and TN2 is the sender.

Configure both the RUT and TR1 such that TR1 is the RP for group 224.0.1.20. This configuration can be either manual, or implicit through the Bootstrap mechanism. Configure TN1 and TN2 such that TN1 is the receiver, and TN2 is the sender, both for group 224.0.1.20. The rest of the procedure is same as in Part A.

Part C: The RUT is the RP, TN2 is the receiver, and TN1 is the sender.

Configure both the RUT and TR1 such that the RUT is the RP for group 224.0.1.20. This configuration can be either manual, or implicit through the Bootstrap mechanism. Configure TN1 and TN2 such that TN2 is the receiver, and TN1 is the sender, both for group 224.0.1.20. The rest of the procedure is same as in Part A.

Part D: TR1 is the RP, TN2 is the receiver, and TN1 is the sender.

Configure both the RUT and TR1 such that TR1 is the RP for group 224.0.1.20. This configuration can be either manual, or implicit through the Bootstrap mechanism. Configure TN1 and TN2 such that TN2 is the receiver, and TN1 is the sender, both for group 224.0.1.20. The rest of the procedure is same as in Part A.

Observable Results: In all cases the data packets by the sender should be received by the receiver.

In Part A, the data packets from TN2 should be encapsulated in PIM Registers by TR1, unicast to the RUT (the RP), decapsulated by RUT, and then forwarded (using native multicast) to TN1. In Part B, the data packets from TN2 should be forwarded by TR1 to the RUT (using native multicast, because TR1 does not need to encapsulate PIM Registers to itself), and then forwarded to TN1. In Part C, the data packets from TN1 should be forwarded (using native multicast) by the RUT to TR1, and then to TN2. In Part D, the data packets from TN1 should be encapsulated in PIM Registers by the RUT, unicast to TN (the RP), decapsulated by TN, and then forwarded (using native multicast) to TN2.

Possible Problems: If the RP-set is empty, or is not same for the RUT and TR1, no multicast packets will be received by the receiver.

Test Group 2

Designated Routers (DR) and Hello Messages

Scope: Test PIM router neighbor discovery, option exchange, and DR election.

Overview: PIM Hello messages are send periodically on each PIM-enabled interface. They are used to discover neighboring PIM routers, to exchange various options, and to elect a Designated Router (DR) per LAN. Hello messages are also used to provide a keep-alive function to detect a neighbor loss.

2.1 Hello Transmission

Purpose: Verify that a router properly sends Hello messages.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.3.1

Discussion: Neighbor discovery of PIM capable routers is accomplished by sending Hello messages to the ALL-PIM-ROUTERS multicast address (224.0.0.13 for IPv4, and “ff02::d” for IPv6). These Hello messages must contain the following:

- The IP TTL MUST be set to one.
- The Type field must be set to 0, this designates a PIM Hello message.

The router should transmit the Hello messages at an interval of `Hello_Period` (30 sec).

Test Setup: Connect the RUT according to Figure 2.1. Enable PIM-SM on the RUT.

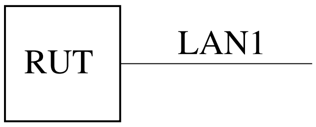

Figure 2.1: Hello transmission test setup

Procedure:

1. Start the RUT.
2. Observe the messages transmitted by the RUT.

Observable Results:

- The RUT should send properly formatted PIM Hello messages to the ALL-PIM-ROUTERS multicast address at `Hello_Period` (30 sec) interval.
- The first Hello message only should be send after a random interval between 0 and `Triggered_Hello_Delay` (5 sec).
- The PIM Hello messages should meet all the requirements in the discussion section.

Possible Problems: None.

2.2 Two-Way Neighbor Adjacency

Purpose: Verify that a router accepts Hello messages and forms a two-way neighbor adjacency.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.3.1

Discussion: Neighbor discovery of PIM capable routers is accomplished by sending Hello messages to the ALL-PIM-ROUTERS address (224.0.0.13 for IPv4, and "ff02::d" for IPv6). As a PIM router receives Hello messages from its PIM neighbors, it records the neighbor addresses on each interface.

Test Setup: Connect the RUT and TR1 according to Figure 2.2. Enable PIM-SM on both the RUT and TR1.

Figure 2.2: Two-way neighbor adjacency test setup

Procedure:

Part A: Neighbor adjacency when there is network connectivity failure.

1. Start the RUT.
2. Observe the neighbor state information in the RUT for at least `Triggered_Hello_Delay` (5 sec).
3. Start TR1.
4. Observe the neighbor state information in the RUT for at least `Triggered_Hello_Delay` (5 sec).
5. Disconnect TR1 from LAN1.
6. Observe the neighbor state information in the RUT for at least `Hello_Holdtime` (105 sec).

Part B: Neighbor adjacency when a neighbor gracefully goes down.

The procedure is same as in Part A, except that TR1 is gracefully shut-down instead of disconnected from LAN1.

Part C: Neighbor adjacency when the RUT is gracefully shut-down.

1. Start the RUT and TR1.
2. Observe the neighbor state information in the RUT.
3. Observe the messages transmitted by the RUT on LAN1.
4. Gracefully shut-down the RUT.

Observable Results:

Part A:

- Before TR1 is started, the neighbor state information in the RUT should show no PIM neighbors:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr    V Mode    Holdtime Timeout
```

- After TR1 is started, and after up to `Triggered_Hello_Delay` (5 sec), the neighbor state information in the RUT must contain TR1:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc1                       1 10.2.0.2         2 Sparse       105     103
```

- After TR1 is disconnected from LAN1, and after up to `Hello_Holdtime` (105 sec), the neighbor state information in the RUT should show no PIM neighbors:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
```

Part B:

The observed results should be same as in Part A, except that after TR1 is gracefully shut-down, it would send a Hello message with zero `HoldTime`. After the RUT receives this Hello message, it should immediately timeout TR1, instead of up to `Hello_Holdtime` (105 sec).

Part C:

- After TR1 is started, and after up to `Triggered_Hello_Delay` (5 sec), the neighbor state information in the RUT must contain TR1:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc1                       1 10.2.0.2         2 Sparse       105     103
```

- Right after the RUT is gracefully shut-down, it should send a Hello message with zero `HoldTime`.

Possible Problems: None.

2.3 Hello Reception

Purpose: Verify that a router properly receives Hello messages.

References:

- draft-ietf-pim-sm-v2-new-05 – Sections 4.3.1, 4.10, and 4.10.2

Discussion: Neighbor discovery of PIM capable routers is accomplished by sending Hello messages to the ALL-PIM-ROUTERS address (224.0.0.13 for IPv4, and "ff02::d" for IPv6). As a PIM router receives Hello messages from its PIM neighbors, it records the neighbor addresses on each interface. The checksum is 16-bit one's complement of the one's complement sum of the PIM messages. The checksum MUST be validated on reception of the message. The unused and reserved bits MUST be ignored upon reception.

Test Setup: Connect the RUT and TR1 according to Figure 2.3. Enable PIM-SM on both the RUT and TR1.

Figure 2.3: Hello reception test setup

Procedure:

Part A: Reception of a PIM Hello message containing an invalid checksum.

1. TR1 transmits a Hello message with an invalid checksum.
2. Observe the neighbor state information in the RUT.

Part B: Reception of a PIM Hello message containing a Reserved field of 0xFF.

1. TR1 transmits a Hello message containing a Reserved field of 0xFF.
2. Observe the neighbor state information in the RUT.

Part C: Reception of a PIM Hello message containing a Version field of 0xF.

1. TR1 transmits a Hello message containing a Version field of 0xF.
2. Observe the neighbor state information in the RUT.

Part D: Reception of a PIM message containing an unrecognized Type field of 0xF.

1. TR1 transmits a PIM message containing a Type field of 0xF.
2. Observe the warning log messages in the RUT.

Observable Results:

- In Part A, the Hello messages with invalid checksum should be ignored, and the neighbor state information in the RUT should show no PIM neighbors:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
```

- In Part B, after TR1 is started, and after up to `Triggered_Hello_Delay` (5 sec), the neighbor state information in the RUT must contain TR1:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc1                       1 10.2.0.2         2 Sparse       105     103
```

- In Part C, after TR1 is started, and after up to `Triggered_Hello_Delay` (5 sec), the neighbor state information in the RUT must contain TR1 with protocol version of 15 (0xF).

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc2                       1 10.3.0.1        15 Sparse       105     103
```

- In Part D, after TR1 is started and after it sends PIM messages containing a Type field of 0xF, the RUT should log the messages with this unrecognized Type field:

```
[ 2002/08/17 22:45:20 WARNING test_pim.rut PIM ] RX PIM_type_unknown from
10.3.0.2 to 224.0.0.13: message type (15) is unknown
```

Possible Problems:

- Earlier protocol specification (RFC 2117), have “Addr length” field in place of the “Reserved” field. Therefore, the test in Part B may not be appropriate for such implementations.
- At the moment of writing, the lastest PIM-SM spec does not describe the behavior if a PIM message is received with protocol version that is larger than the implemented one. (TODO: if the spec later is modified to include that behavior, remove this bullet). Hence, in Part C, it is possible that an implementation may silently ignore all messages with protocol version larger than `PIM_SM_version_default` (PIM-SMv2). As a result, the RUT will not see TR1 as a neighbor.
- In Part D, the RUT may not log events such as receiving of a message with unrecognized Type field, because the logging is not described in the spec.

2.4 Holdtime Option

Purpose: Verify that a router properly sends and receives Hello messages with Holdtime option.

References:

- draft-ietf-pim-sm-v2-new-05 – Sections 4.3.1 and 4.10.2

Discussion: PIM capable routers periodically send Hello messages messages to the ALL-PIM-ROUTERS address (224.0.0.13 for IPv4, and "ff02::d" for IPv6). If a PIM router does not receive a Hello message from a neighbor for some amount of time, it is assumed that the neighbor (or the connectivity to it) is not operational, therefore the neighbor is timed-out. A Hello message may contain a Holdtime option that specifies the amount of time (in seconds) a router must keep the neighbor reachable. If a Hello message does not contain a Holdtime option, the default value of `Default_Hello_Holdtime` (105 sec) is assumed. If a Hello message contains a Holdtime option with value of 0xFFFF, then only one Hello message should be sent, and the router-recipient of the message should not time-out the router-originator.

Test Setup: Connect the RUT and TR1 according to Figure 2.4. Enable PIM-SM on both the RUT and TR1.

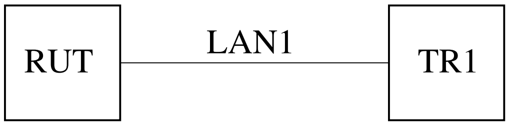

Figure 2.4: Holdtime option test setup

Procedure:

Part A: Transmission of a PIM Hello message containing a Holdtime option.

1. Start the RUT.
2. Observe the messages transmitted by the RUT for at least `Triggered_Hello_Delay` + 3 * `Hello_Holdtime` (*i.e.,* (5 sec) + 3 * (105 sec)).
3. Stop the RUT.
4. Observe the messages transmitted by the RUT for at least `Triggered_Hello_Delay` + 3 * `Hello_Holdtime` (*i.e.,* (5 sec) + 3 * (105 sec)).
5. Change the configuration value of `Hello_Holdtime` and repeat.

Part B: Transmission of a PIM Hello message containing a Holdtime option with value of 0xFFFF.

The procedure is same as in Part A, except that the RUT is configured with Holdtime option value of 0xFFFF. Note that the amount of time to observe the transmitted messages should be same as in Part A.

Part C: Reception of a PIM Hello message containing a Holdtime option.

1. Start the RUT.
2. Observe the neighbor state information in the RUT for at least `Triggered_Hello_Delay` (5 sec).

3. Configure TR1 such that its Hello messages would contain a Holdtime option with the default value of `Hello_Holdtime` (105 sec).

4. Start TR1.

5. Observe the neighbor state information in the RUT for at least `Triggered_Hello_Delay` (5 sec).

6. Stop TR1.

7. Observe the neighbor state information in the RUT for at least `Hello_Holdtime` (105 sec).

8. Change the configuration value of `Hello_Holdtime` in TR1 and repeat.

Part D: Reception of a PIM Hello message containing a Holdtime option with value of 0xFFFF.
The procedure is same as in Part C, except that in Step 3 the Hello message originated by TR1 must be configured contain a Holdtime option with value of 0xFFFF.

Part E: Reception of a PIM Hello message that does not contain a Holdtime option.
The procedure is same as in Part C, except that in Step 3 the the Hello message originated by TR1 must be configured to exclude the Holdtime option.

Observable Results:

Part A:

- After the RUT is started, the first Hello message should be transmitted no later than `Triggered_Hello_Delay` (5 sec).

- After the first Hello message, there should be one Hello message transmitted each `Hello_Period` (30 sec). Each Hello message should contain a Holdtime option with default value of `Hello_Holdtime` (105 sec).

- Right after the RUT is stopped, there should be a Hello message transmitted with Holdtime option value of 0.

- After the Hello message with Holdtime option value of 0, no other Hello messages should be transmitted.

- The results of repeating the test with different value of `Hello_Holdtime` should be similar, except that the Hello messages should be transmitted every 1/3th of the new `Hello_Holdtime`, and the Holdtime Option value in the Hello messages should be the new `Hello_Holdtime`.

Part B:

- After the RUT is started, the first Hello message should be transmitted no later than `Triggered_Hello_Delay` (5 sec). This message should contain a Holdtime option with value of 0xFFFF.

- After the first Hello message, there should be no one Hello messages transmitted before the RUT is stopped.

- Right after the RUT is stopped, there should be a Hello message transmitted with Holdtime option value of 0.

- After the Hello message with Holdtime option value of 0, no other Hello messages should be transmitted.

Part C:

- Before TR1 is started, the neighbor state information in the RUT should show no PIM neighbors:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr    V Mode    Holdtime Timeout
```

- After TR1 is started, and after up to `Triggered_Hello_Delay` (5 sec), the neighbor state information in the RUT must contain TR1. The Holdtime information about TR1 must match the Holdtime value in the Hello messages by TR1:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc1                       1 10.2.0.2         2 Sparse       105     103
```

- After TR1 is stopped, and after it sends a Hello message with Holdtime option value of 0, the RUT must immediately expire it. The neighbor state information in the RUT should show no PIM neighbors:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
```

- The results of repeating the test with different value of `Hello_Holdtime` should be similar, except that the Holdtime value in the state information in the RUT about TR1 should match the chosen value of the `Hello_Holdtime`.

Part D:

The observed results should be same as in Part C, except that after the first Hello message from TR1 is received, the neighbor state information in the RUT must show that the TR1 would never be expired:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc1                       1 10.3.0.2         2 Sparse     65535    None
```

Part E:

The observed results should be same as in Part C, except that the RUT assumes that the Holdtime value about TR1 is always equal to the default value of `Hello_Holdtime` (105 sec).

Possible Problems: None.

2.5 DR Priority Option

Purpose: Verify that a router properly sends and receives Hello messages with DR Priority option.

References:

- draft-ietf-pim-sm-v2-new-05 – Sections 4.3.1, 4.3.2, and 4.10.2

Discussion: One of the PIM routers on each LAN, the Designated Router (DR), needs to act on behalf of the rest of the routers and directly connected hosts with respect to the PIM protocol. The DR election considers the DR priority of each PIM router and its IP address: numerically larger DR priority is always better, with a numerically larger IP address used as a tie-break. Each PIM router includes its DR priority in the DR Priority option of the Hello messages it originates. If there is at least one PIM router on a LAN that does not include the DR Priority option in its Hello messages, then the DR election process considers only the IP addresses of the routers.

Test Setup: Connect the RUT and TR1 according to Figure 2.5. Enable PIM-SM on both the RUT and TR1.

Figure 2.5: DR priority option test setup

Procedure:

Part A: The RUT has a numerically larger IP address than TR1.

1. Start the RUT with default DR priority of 1, and observe the DR state information in the RUT.
2. Start TR1 with default DR priority of 1, and observe the DR state information in the RUT.
3. Change the DR priority of TR1 to 2, and observe the DR state information in the RUT.
4. Change the DR priority of TR1 to 0, and observe the DR state information in the RUT.
5. Change the DR priority of TR1 to 1, and observe the DR state information in the RUT.
6. Change the DR priority of the RUT to 2, and observe the DR state information in the RUT.
7. Change the DR priority of the RUT to 0, and observe the DR state information in the RUT.
8. Change the DR priority of the RUT to 1, and observe the DR state information in the RUT.

Part B: The RUT has a numerically smaller IP address than TR1.

The procedure is same as in Part A, except that TR1 has a numerically smaller IP address than TR1.

Observable Results:

Part A:

1. After the RUT is started with default DR priority of 1, it should be the DR for LAN1:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2             UP       Sparse  2 DR               1 10.2.0.2            0
```

2. After TR1 is started with default DR priority of 1, the RUT should be still the DR for LAN1:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2             UP       Sparse  2 DR               1 10.2.0.2            1
```

3. After the DR priority of TR1 is increased to 2, TR1 should become the DR for LAN1. The DR state information in the RUT should show that the RUT is not the DR anymore:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2             UP       Sparse  2 NotDR            1 10.2.0.1            1
```

4. After the DR priority of TR1 is decreased to 0, the RUT should become the DR for LAN1:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2             UP       Sparse  2 DR               1 10.2.0.2            1
```

5. After the DR priority of TR1 is increased to 1, the RUT should continue to be the DR for LAN1:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2             UP       Sparse  2 DR               1 10.2.0.2            1
```

6. After the DR priority of the RUT is increased to 2, the RUT should continue to be the DR for LAN1:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr     Neighbors
dc2             UP       Sparse  2 DR               2 10.2.0.2           1
```

7. After the DR priority of the RUT is decreased to 0, TR1 should become the DR for LAN1. The DR state information in the RUT should show that the RUT is not the DR anymore:

```
Xorp> show pim interface dc2
Interface       State    Mode    V PIMstate  Priority DRaddr     Neighbors
dc2             UP       Sparse  2 NotDR            0 10.2.0.1           1
```

8. After the DR priority of the RUT is increased to 1, the RUT should become the DR for LAN1:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 DR               1 10.2.0.2            1
```

Part B:

1. After the RUT is started with default DR priority of 1, it should be the DR for LAN1:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 DR               1 10.2.0.2            0
```

2. After TR1 is started with default DR priority of 1, it should become the DR for LAN1. The DR state information in the RUT should show that the RUT is not the DR anymore:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 NotDR            1 10.2.0.3            1
```

3. After the DR priority of TR1 is increased to 2, TR1 should continue to be the DR for LAN1. The DR state information in the RUT should show that the RUT is not the DR:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 NotDR            1 10.2.0.3            1
```

4. After the DR priority of TR1 is decreased to 0, the RUT should become the DR for LAN1:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 DR               1 10.2.0.2            1
```

5. After the DR priority of TR1 is increased to 1, TR1 should become the DR for LAN1. The DR state in the RUT should show that the RUT is not the DR:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 NotDR            1 10.2.0.3            1
```

6. After the DR priority of the RUT is increased to 2, the RUT should become the DR for LAN1:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr      Neighbors
dc2              UP       Sparse  2 DR               2 10.2.0.2            1
```

7. After the DR priority of the RUT is decreased to 0, TR1 should become the DR for LAN1. The DR state in the RUT should show that the RUT is not the DR:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr       Neighbors
dc2              UP       Sparse  2 NotDR            0 10.2.0.3             1
```

8. After the DR priority of the RUT is increased to 1, TR1 should continue to be the DR for LAN1. The DR state information in the RUT should show that the RUT is not the DR:

```
Xorp> show pim interface dc2
Interface        State    Mode    V PIMstate  Priority DRaddr       Neighbors
dc2              UP       Sparse  2 NotDR            1 10.2.0.3             1
```

Possible Problems: None.

2.6 Generation ID Option

Purpose: Verify that a router properly sends and receives Hello messages with Generation ID option.

References:

- draft-ietf-pim-sm-v2-new-05 – Sections 4.3.1 and 4.10.2

Discussion: The Generation ID (GenID) option contains a randomly generated 32-bit value that is regenerated each time PIM forwarding is started or restarted on the interface, including when the router itself restarts. When a Hello message with a new GenID is received from a neighbor, any old Hello information about that neighbor should be discarded and superseded by the information from the new Hello message. In addition, if a router needs to send a Join/Prune or some other control messages to the new neighbor, then it must send a Hello message, immediately followed by the relevant control messages.

Test Setup: Connect the RUT, TR1, and Rx1 according to Figure 2.6. Enable PIM-SM on both the RUT and TR1.

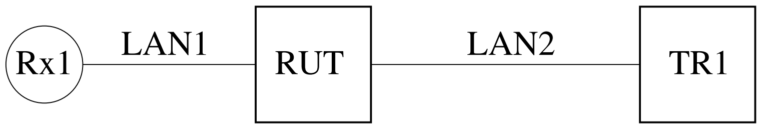

Figure 2.6: Generation ID option test setup

Procedure:

Part A: Transmission of a PIM Hello message containing a Generation ID option.

1. Start the RUT.
2. Observe the messages transmitted by the RUT on LAN2 for at least `Triggered_Hello_Delay` + 3 * `Hello_Holdtime` (*i.e.,* (5 sec) + 3 * (105 sec)).
3. Restart the RUT interface that connects the RUT to LAN2.
4. Observe the messages transmitted by the RUT on LAN2 for at least `Triggered_Hello_Delay` (5 sec).
5. Restart the RUT.
6. Observe the messages transmitted by the RUT on LAN2 for at least `Triggered_Hello_Delay` (5 sec).

Part B: Reception of a PIM Hello message containing a Generation ID option.

1. Start both the RUT and TR1.
2. Observe the messages transmitted by the RUT on LAN2.
3. Quit TR1 (*i.e.,* stop it without graceful shutdown), and start it immediately.

Part C: Reception of a PIM Hello message containing a Generation ID option when the RUT has pending Join messages to send.

Configure both the RUT and TR1 such that it is the RP for group 224.0.1.20. This configuration can be either manual, or implicit through the Bootstrap mechanism. Configure Rx1 such that it is the receiver for group 224.0.1.20.

1. Start both the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start Rx1, and observe the Join state in the RUT and TR1.
3. Observe the messages transmitted by the RUT on LAN2.
4. Quit TR1 (*i.e.,* stop it without graceful shutdown), and start it immediately.

Observable Results:

Part A:

- After the RUT is started, after a random interval between 0 and `Triggered_Hello_Delay` (5 sec) the RUT should start transmitting Hello message with Generation ID option included. There should be a Hello message every `Hello_Holdtime` (105 sec), and the value of the Generation ID must be same for all messages.
- After the RUT interface that connects the RUT to LAN2 is restarted, after a random interval between 0 and `Triggered_Hello_Delay` (5 sec) the RUT should transmit on LAN2 a Hello message with Generation ID option included. The value of this Generation ID must be different from the value of the Hello message before the restart.
- After the RUT is restarted, after a random interval between 0 and `Triggered_Hello_Delay` (5 sec) the RUT should transmit on LAN2 a Hello message with Generation ID option included. The value of this Generation ID must be different from the value of the Hello message before the restart.

Part B:

- After the RUT receives the first Hello message from TR1 that has different Generation ID from the one before the restart, after a random interval between 0 and `Triggered_Hello_Delay` (5 sec) the RUT should transmit on LAN2 a Hello message.

Part C:

- After the RUT receives the first Hello message from TR1 that has different Generation ID from the one before the restart, the RUT should transmit immediately on LAN2 a Hello message followed by a PIM Join/Prune message with group 224.0.1.20 included in the list of joined groups.

Possible Problems: None.

2.7 Two-Way Neighbor Adjacency Without Hello Messages

Purpose: Verify that a router can be configured to form a two-way neighbor adjacency even if no Hello message was received.

References:

- draft-ietf-pim-sm-v2-new-05 – Sections 4.3.1, and 4.5
- draft-ietf-pim-sm-bsr-02 – Section 3

Discussion: The following PIM control messages should only be accepted for processing if they come from a known PIM neighbor: Join/Prune, Bootstrap, Assert, Graft, and Graft-Ack. A PIM router hears about PIM neighbors through PIM Hello messages. However, some older PIM implementations incorrectly fail to send Hello messages on point-to-point interfaces. Hence, the spec recommends that a configuration option should be provided to allow interoperation with such old routers (disabled by default). More specifically, if the option is enabled, then the above PIM control messages should be accepted from a neighbor even if no Hello message was received first from that neighbor.

Test Setup: Connect the RUT, TR1, S1, and Rx1 according to Figure 2.7. Enable PIM-SM on both TR1 and the RUT. Configure TR1 as a Candidate-BSR, and the RUT as a Candidate-RP. Modify or configure TR1 such that it never sends PIM Hello messages. Configure Rx1 and S1 such that Rx1 is a receiver, and S1 is a sender, both for group 224.0.1.20.

Figure 2.7: Two-way neighbor adjacency without Hello messages test setup

Procedure:

1. Start TR1 and the RUT.
2. Observe the messages transmitted by TR1 and the RUT, and the neighbor and Bootstrap state information in the RUT.
3. Wait until TR1 transmits 2-3 Bootstrap messages.
4. Enable the interoperability option in the RUT (without restarting it), such that the RUT would accept PIM control messages from TR1 even if TR1 does not send Hello messages.
5. Wait until TR1 transmits 2-3 Bootstrap messages.
6. Start Rx1.
7. Start S1.

Observable Results:

- After TR1 and the RUT are started, the RUT should start transmitting PIM Hello messages on LAN2.
- After TR1 receives the first Hello message from the RUT, TR1 should unicast a Bootstrap message to the RUT. After that it start transmitting periodically only Bootstrap messages on LAN2.
- Each Bootstrap message received by the RUT should be ignored because it comes from an unknown neighbor. Therefore, the RUT should not have a BSR:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
0.0.0.0            0 0.0.0.0            0 Init                  -1        -1
```

  In addition, the neighbor state information in the RUT should be empty:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
```

- After the interoperability option in the RUT is enabled, the next Bootstrap message from TR1 should be accepted by the RUT. As a result, the Bootstrap state in the RUT should show that TR1 is the BSR:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
10.2.0.1           1 0.0.0.0            0 AcceptPreferred       74      1444
```

  Eventually, the neighbor state information in the RUT may show TR1. If this is the case, the state information in the RUT for TR1 should be refreshed by each Bootstrap originated by TR1:

```
Xorp> show pim neighbors
Interface        DRpriority NeighborAddr     V Mode    Holdtime Timeout
dc2                    none 10.2.0.1         2 Sparse       210     164
```

- After Rx1 is started, TR1 should send PIM Join message for group 224.0.1.20 on LAN2 toward the RP (the RUT).
- After S1 is started, the multicast data packets originated by it should be forwarded by the RUT and TR1 to Rx1.

Possible Problems: None.

Test Group 3

PIM Register Messages

Scope: Test sending and receiving of PIM Register messages.

Overview: The Designated Router (DR) on a LAN or point-to-point link encapsulates multicast packets from local sources to the RP for the relevant group unless it recently received a Register Stop message for that (S,G) or (*,G) from the RP. When the DR receives a Register Stop message from the RP, it starts a Register Stop timer to maintain this state. Just before the Register Stop timer expires, the DR sends a Null-Register message to the RP to allow the RP to refresh the Register Stop information at the DR. If the Register Stop timer actually expires, the DR will resume encapsulating packets from the source to the RP.

3.1 Register Messages Transmission

Purpose: Verify that a DR properly sends Register messages.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.4.1

Discussion: The DR encapsulates multicast packets from local sources into PIM Register messages, and unicasts them to the RP for the relevant multicast group.

Test Setup: Connect the RUT, TR1, TR2, S1, and Rx1 according to Figure 3.1. Configure the RUT, TR1, and TR2 such that TR1 is the RP for group 224.0.1.20, and such that it never attempts to switch to the shortest-path tree by originating an (S,G) SPT Join message toward a source. Enable PIM-SM on the RUT, TR1, and TR2. Configure Rx1 and S1 such that Rx1 is a receiver, and S1 is a sender, both for group 224.0.1.20.

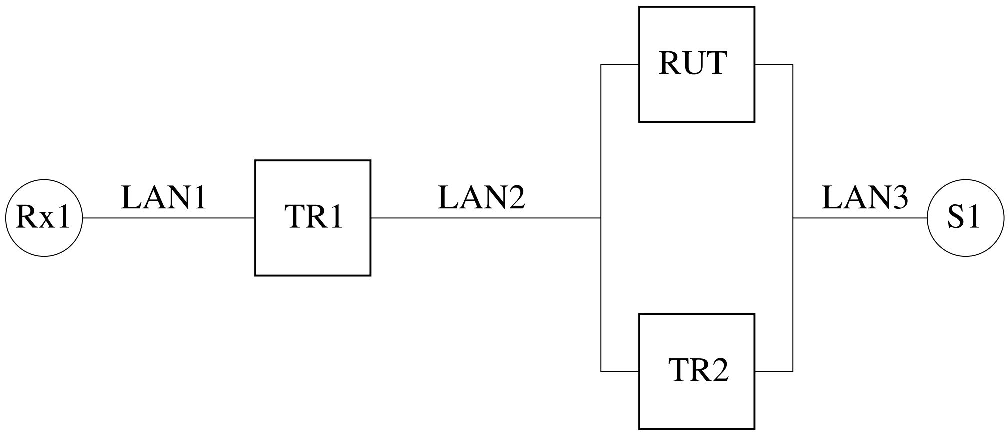

Figure 3.1: Register messages transmission test setup

Procedure:

Part A: Transmission of Register messages.

1. Configure the RUT such that it is the DR on LAN3.
2. Start the RUT, TR1, and TR2. If necessary, wait until the RP-set in the RUT, TR1, and TR1 converges.
3. Start Rx1.
4. Observe the messages transmitted by the RUT and TR2 on LAN2.
5. Start S1.

Part B: Non-transmission of Register messages.

The procedure is same as in Part A, except that TR1 instead of the RUT is the DR on LAN3.

Part C: Switching between transmission and non-transmission of Register messages.

1. Configure the RUT such that it is the DR on LAN3.
2. Start the RUT, TR1, and TR2. If necessary, wait until the RP-set in the RUT, TR1, and TR1 converges.
3. Start Rx1.
4. Observe the messages transmitted by the RUT and TR2 on LAN2.
5. Start S1.
6. Reconfigure the RUT (without stopping it), such that TR2 is the DR on LAN3.
7. Reconfigure the RUT (without stopping it), such that it again is the DR on LAN3.

Part D: Handling of RegisterStop(S,G) messages at the DR.

1. Start the RUT and TR1 (note that in this part we do not use TR2). If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start Rx1.
3. Observe the messages transmitted by the RUT on LAN2.
4. Start S1.
5. Stop Rx1.
6. Start Rx1.

Observable Results:

Part A:

After S1 is started, the RUT should start encapsulating the data packets transmitted by S1 in PIM Register messages, and unicast them to the RP (TR1, that should decapsulate and forward them to Rx1).

Part B:

After S1 is started, the RUT should NOT encapsulate the data packets transmitted by S1 in PIM Register messages. Instead, TR2 should encapsulate and unicast them to the RP (TR1, that should decapsulate and forward them to Rx1).

Part C:

- After S1 is started, the RUT should start encapsulating the data packets transmitted by S1 in PIM Register messages, and unicast them to the RP (TR1, that should decapsulate and forward them to Rx1).
- After the RUT is reconfigured such that it is not the DR on LAN3, it should immediately stop transmitting PIM Register messages to the RP. Instead, the new DR (TR2) should start transmitting them.
- After the RUT is reconfigured such that it is again the DR on LAN3, it should immediately start transmitting PIM Register messages to the RP. The previous DR (TR2) should stop transmitting them.

Part D:

- After S1 is started, the RUT should start encapsulating the data packets transmitted by S1 in PIM Register messages, and unicast them to the RP (TR1, that should decapsulate and forward them to Rx1).

- After Rx1 is stopped, TR1 should send PIM Register Stop message to the RUT for each PIM Register message it receives from the RUT (including the PIM Null Register messages). After receiving the first PIM Register Stop message, the RUT should stop sending PIM Register messages to the RP (TR1) with the encapsulated data packets from S1. However, from time to time the RUT should be sending PIM Null Register messages to the RP (TR1) with time interval between two messages a random value chosen uniformly from the interval
  (0.5 * `Register_Suppression_Time`, 1.5 * `Register_Suppression_Time`)
  - `Register_Probe_Time`
  = ((0.5 * 60 sec, 1.5 * 60 sec) - 5 sec).
  To each PIM Null Register message, the RP (TR1) should respond with a PIM Register Stop message, therefore the RUT should continue not to encapsulate the data packets from S1.

- After Rx1 is started again, the RP (TR1) should send an SPT (S,G) Join message on LAN2 toward the source. As a result, the data packets from S1 should be forwarded to the RP (TR1) natively instead of encapsulating them in PIM Register messages.

Possible Problems: In Part C, if the sender's rate is relatively high, there could be few packet losses at Rx1 when the DR on LAN3 changes.

3.2 Register Tunnel Interface

Purpose: Verify that a Register tunnel virtual interface is properly created, removed, or changed.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.4.1

Discussion: When the DR has to encapsulate the data packets from local sources into PIM Register messages, and unicasts them to the RP for the relevant multicast group, it creates a Register tunnel virtual interface with its encapsulation target being the RP. If the DR should stop encapsulating the data packets, it removes the Register tunnel virtual interface. If the RP for the multicast group changes, the DR should update the Register tunnel virtual interface.

Test Setup: Connect the RUT, TR1, TR2, S1, and Rx1 according to Figure 3.2. Enable PIM-SM on the RUT, TR1, and TR2. In all the tests, configure the RP such that it never attempts to switch to the shortest-path tree by originating an (S,G) SPT Join message toward a source. Enable PIM-SM on the RUT, TR1, and TR2. Configure Rx1 and S1 such that Rx1 is a receiver, and S1 is a sender, both for group 224.0.1.20.

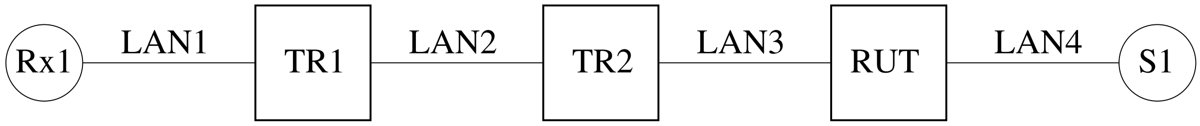

Figure 3.2: Register tunnel interface test setup

Procedure:

Part A: Add and remove Register tunnel.

1. Configure TR1 such that it is the RP.
2. Start the RUT, TR1, and TR2. If necessary, wait until the RP-set in the RUT, TR1, and TR2 converges.
3. Start Rx1.
4. Observe the Register state machine at the RUT, and the messages transmitted by the RUT on LAN3.
5. Start S1.
6. Stop Rx1.
7. Stop S1.

Part B: Update Register tunnel.

1. Configure TR1 such that it is the RP.
2. Start the RUT, TR1, and TR2. If necessary, wait until the RP-set in the RUT, TR1, and TR2 converges.
3. Start Rx1.
4. Observe the Register state machine at the RUT, and the messages transmitted by the RUT on LAN3.

5. Start S1.
6. Reconfigure TR1 and TR2 such that TR2 becomes the RP.

Observable Results:

Part A:

- After Rx1 is started, the Register state machine in the RUT for source S1 and group 224.0.1.20 should be in No Info state:

  ```
  Xorp> show pim join 224.0.1.20
  Group           Source          RP              Flags
  ```

  Further, no PIM Register messages should be transmitted by the RUT.

- After S1 is started, the Register state machine in the RUT for source S1 and group 224.0.1.20 should be in Join state, and the Register tunnel virtual interface should be created between the RUT and the RP (TR1):

  ```
  Xorp> show pim join 224.0.1.20
  Group           Source          RP              Flags
  224.0.1.20      10.4.0.2        10.2.0.1        SG SPT DirectlyConnectedS
      Upstream interface (S):   dc0
      Upstream interface (RP):  dc2
      Upstream MRIB next hop (RP): 10.3.0.1
      Upstream MRIB next hop (S):  UNKNOWN
      Upstream RPF'(*,G):       UNKNOWN
      Upstream RPF'(S,G):       UNKNOWN
      Upstream RPF'(S,G,rpt):   UNKNOWN
      Upstream state:           Joined
      Register state:           RegisterJoin RegisterCouldRegister
      Join timer:               13
      Local include:            .............
      Local exclude:            .............
      Local include WC:         .............
      Local include SG:         .............
      Local exclude SG:         .............
      Joins RP:                 .............
      Joins WC:                 .............
      Joins SG:                 ............O
      Prunes SG_RPT:            .............
      Join state:               ............O
      Prune pending state:      .............
      Prune state:              .............
      I am assert winner state: .............
      I am assert loser state:  .............
      Assert winner WC:         .............
  ```

```
    Assert winner SG:          .............
    Assert lost WC:            .............
    Could assert SG:           ............O
    I am DR:                   ....O.O......
    Immediate olist RP:        .............
    Immediate olist WC:        .............
    Immediate olist SG:        ............O
    Inherited olist SG:        ............O
    Inherited olist SG_RPT:    .............
    PIM include WC:            .............
```

Further, each data packet from S1 should be encapsulated by the RUT in a PIM Register message and unicast to the RP (TR1).

- After Rx1 is stopped, the RP (TR1) should send PIM Register Stop message to the RUT for each PIM Register message it receives from the RUT (PIM Null Register messages excluded). After the RUT receives the first PIM Register Stop message, the Register tunnel virtual interface to the RP (TR1) should be in Prune state:

```
Xorp> show pim join 224.0.1.20
Group           Source          RP              Flags
224.0.1.20      10.4.0.2        10.2.0.1        SG DirectlyConnectedS
    Upstream interface (S):   dc0
    Upstream interface (RP):  dc2
    Upstream MRIB next hop (RP): 10.3.0.1
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):       UNKNOWN
    Upstream RPF'(S,G):       UNKNOWN
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           NotJoined
    Register state:           RegisterPrune RegisterCouldRegister
    Join timer:               -1
    Local include:            .............
    Local exclude:            .............
    Local include WC:         .............
    Local include SG:         .............
    Local exclude SG:         .............
    Joins RP:                 .............
    Joins WC:                 .............
    Joins SG:                 .............
    Prunes SG_RPT:            .............
    Join state:               .............
    Prune pending state:      .............
    Prune state:              .............
    I am assert winner state: .............
    I am assert loser state:  .............
    Assert winner WC:         .............
```

```
    Assert winner SG:          .............
    Assert lost WC:            .............
    Could assert SG:           .............
    I am DR:                   ....O.O......
    Immediate olist RP:        .............
    Immediate olist WC:        .............
    Immediate olist SG:        .............
    Inherited olist SG:        .............
    Inherited olist SG_RPT:    .............
    PIM include WC:            .............
```

Further, from time to time the RUT should be sending PIM Null Register messages to the RP (TR1) with time interval between two messages a random value chosen uniformly from the interval (0.5 * `Register_Suppression_Time`, 1.5 * `Register_Suppression_Time`) - `Register_Probe_Time`
= ((0.5 * 60 sec, 1.5 * 60 sec) - 5 sec).
To each PIM Null Register message, TR1 should respond with a PIM Register Stop message, therefore the RUT should continue not to encapsulate the data packets from S1.

- After S1 is stopped, and after period of `Keepalive_Period` (210 sec), the state for source S1 and group 224.0.1.20 in the RUT should expire:

```
Xorp> show pim join 224.0.1.20
Group           Source          RP              Flags
```

Further, no PIM Register messages should be transmitted by the RUT.

Part B:

- The results until after S1 is started should be same as in Part A.
- After TR1 and TR2 are reconfigured such that TR2 becomes the RP, and after the RP-set in the RUT, TR1, and TR2 converges, the Register tunnel virtual interface in the RUT should be updated to point to the new RP (TR2). The Register state machine should still be in Join state:

```
Xorp> show pim join 224.0.1.20
Group           Source          RP              Flags
224.0.1.20      10.4.0.2        10.3.0.1        SG SPT DirectlyConnectedS
    Upstream interface (S):    dc0
    Upstream interface (RP):   dc2
    Upstream MRIB next hop (RP): 10.3.0.1
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):        UNKNOWN
    Upstream RPF'(S,G):        UNKNOWN
    Upstream RPF'(S,G,rpt):    UNKNOWN
    Upstream state:            Joined
    Register state:            RegisterJoin RegisterCouldRegister
```

```
Join timer:                48
Local include:             ............
Local exclude:             ............
Local include WC:          ............
Local include SG:          ............
Local exclude SG:          ............
Joins RP:                  ............
Joins WC:                  ............
Joins SG:                  ...........O
Prunes SG_RPT:             ............
Join state:                ...........O
Prune pending state:       ............
Prune state:               ............
I am assert winner state: ............
I am assert loser state:   ............
Assert winner WC:          ............
Assert winner SG:          ............
Assert lost WC:            ............
Could assert SG:           ...........O
I am DR:                   ....O.O.....
Immediate olist RP:        ............
Immediate olist WC:        ............
Immediate olist SG:        ...........O
Inherited olist SG:        ...........O
Inherited olist SG_RPT:    ............
PIM include WC:            ............
```

Further, each data packet from S1 should be encapsulated by the RUT in a PIM Register message and unicast to the new RP (TR2).

Possible Problems: In Part B, if the RP-set converges such that the RUT learns that TR2 is the RP before TR2 itself, the PIM Register messages the RUT sends to TR2 may result in TR2 sending-back PIM Register Stop messages. Those PIM Register Stop messages would change the state for source S1 and group 224.0.1.20 in the RUT to Prune, and will stop the PIM Register encapsulation of the data packets from S1. Further, after TR2 learns that it is the RP, it may actually send an SPT (S,G) Join message on LAN2 toward the source. As a result, the data packets from S1 will be forwarded to Rx1 natively instead of encapsulating them in PIM Register messages.

3.3 Register Messages Reception

Purpose: Verify that an RP properly receives and decapsulates Register messages.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.4.2

Discussion: If the RP for a multicast group receives Register messages for that group, and if there are receivers that have joined that group, the RP decapsulates the data packets, and forwards them natively to the receivers. If the multicast group has no receivers, the RP sends Register Stop to the originator of the Register messages.

Test Setup: Connect the RUT, TR1, TR2, S1, and Rx1 according to Figure 3.3. Enable PIM-SM on the RUT, TR1, and TR2. In all the tests, configure the RP such that it never attempts to switch to the shortest-path tree by originating an (S,G) SPT Join message toward a source. Enable PIM-SM on the RUT, TR1, and TR2. Configure Rx1 and S1 such that Rx1 is a receiver, and S1 is a sender, both for group 224.0.1.20.

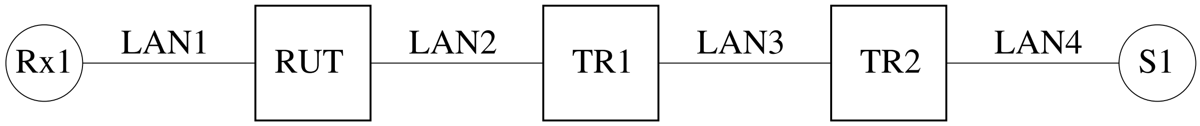

Figure 3.3: Register messages reception test setup

Procedure:

Part A: Receiving Register messages at the RP when there is a receiver.

1. Configure the RUT such that it is the RP.
2. Start the RUT, TR1, and TR2. If necessary, wait intil the RP-set in the RUT, TR1, and TR2 converges.
3. Start Rx1.
4. Observe the messages transmitted by the RUT on LAN2, and the messages received by Rx1.
5. Start S1.
6. Stop Rx1.
7. Start Rx1.

Part B: Receiving Register messages at the RP when there is no receiver.

1. Configure the RUT such that it is the RP.
2. Start the RUT, TR1, and TR2. If necessary, wait intil the RP-set in the RUT, TR1, and TR2 converges.
3. Observe the messages transmitted by the RUT on LAN2, and the messages received by Rx1.
4. Start S1.

5. Start Rx1.
6. Stop Rx1.
7. Start Rx1.

Part C: Receiving Register messages at the non-RP.

1. Manually configure TR2 only to appears that the RUT is the RP. However, configure the RUT itself, and TR1 such that for both of them appear that TR1 is the RP.
2. Start Rx1.
3. Start S1.

Observable Results:

Part A:

- After S1 is started, the data messages it transmits should be encapsulated in PIM Register messages by TR2, and sent to the RP (the RUT). The RUT should decapsulate them, and forward the inner multicast packets down the shared multicast tree to Rx1.
- After Rx1 is stopped, the RUT should send PIM Register Stop message to TR2 for each PIM Register message it receives from the RUT (including the PIM Null Register messages). After receiving the first PIM Register Stop message, TR2 should stop sending PIM Register messages to the RP (the RUT) with the encapsulated data packets from S1. However, from time to time TR2 should be sending PIM Null Register messages to the RP (the RUT) with time interval between two messages a random value chosen uniformly from the interval
  (0.5 * `Register_Suppression_Time`, 1.5 * `Register_Suppression_Time`)
  - `Register_Probe_Time`
  = ((0.5 * 60 sec, 1.5 * 60 sec) - 5 sec).
  To each PIM Null Register message, the RP (the RUT) should respond with a PIM Register Stop message, therefore TR2 should continue not to encapsulate the data packets from S1.
- After Rx1 is started again, the RUT should send an SPT (S,G) Join message on LAN2 toward the source. As a result, the data packets from S1 should be forwarded to the RP (the RUT) natively instead of encapsulating them in PIM Register messages.

Part B:

- After S1 is started, the first one or few data messages it transmits should be encapsulated in PIM Register messages by TR2, and sent to the RP (the RUT). The RUT should respond to each PIM Register message with a PIM Register Stop message for source S1 and group 224.0.1.20. After TR2 receives the first Register Stop message, it should stop encapsulating the data packets and sending them to the RP (the RUT).
- From time to time TR2 should be sending PIM Null Register messages to the RP (the RUT) with time interval between two messages a random value chosen uniformly from the interval
  (0.5 * `Register_Suppression_Time`, 1.5 * `Register_Suppression_Time`)

- `Register_Probe_Time`
= ((0.5 * 60 sec, 1.5 * 60 sec) - 5 sec).
To each PIM Null Register message, the RP (the RUT) should respond with a PIM Register Stop message, therefore TR2 should continue not to encapsulate the data packets from S1.

- After Rx1 is started, the RUT should send an SPT (S,G) Join message on LAN2 toward the source. As a result, the data packets from S1 should be forwarded to the RP (the RUT) natively instead of encapsulating them in PIM Register messages.
- After Rx1 is stopped, the RUT should send an SPT (S,G) Prune message on LAN2 toward the source. As a result, the data packets from S1 should not be forwarded by TR2 from LAN4 on LAN3.
- After Rx1 is started again, the RUT should send an SPT (S,G) Join message on LAN2 toward the source. As a result, the data packets from S1 should be again be forwarded to the RP (the RUT) natively instead of encapsulating them in PIM Register messages.

Part C:

- After S1 is started, the first one or few data messages it transmits should be encapsulated in PIM Register messages by TR2, and sent to the RUT, which TR2 thinks is the RP for group 224.0.1.20. However, because the RUT is not the RP, it should respond to each PIM Register message with a PIM Register Stop message for source S1 and group 224.0.1.20. After TR2 receives the first Register Stop message, it should stop encapsulating the data packets and sending them to the RUT.
- From time to time TR2 should be sending PIM Null Register messages to the RUT with time interval between two messages a random value chosen uniformly from the interval
(0.5 * `Register_Suppression_Time`, 1.5 * `Register_Suppression_Time`)
- `Register_Probe_Time`
= ((0.5 * 60 sec, 1.5 * 60 sec) - 5 sec).
To each PIM Null Register message, the RUT should respond with a PIM Register Stop message, therefore TR2 should continue not to encapsulate the data packets from S1.

Possible Problems: None.

Test Group 4

PIM Join/Prune Messages

Scope: Test sending and receiving of PIM Join/Prune messages.

Overview: A PIM Join/Prune message consists of a list of groups and a list of Joined and Pruned source for each group. It is used by the router originating it to express interest (or lack of interest) in receiving multicast traffic for specific groups and sources.

A Join/Prune message may contain four types of entries. An (*,G) Join/Prune entry is sent toward the RP for group G, and is used to express interest in receiving multicast packets from all sources for that group. An (S,G) Join/Prune entry is sent toward the specified source S, and is used to express interest in receiving multicast packets from the specified source S and group G. An (S,G,rpt) Prune entry is sent toward the RP for group G, and is used to stop receiving multicast packets on the shared tree for the specified source S (note that there is no (S,G,rpt) Join entry, because the (*,G) entry is used for that purpose). An (*,*,RP) Join/Prune entry is sent toward the specified RP, and is used to express interest in receiving multicast packets for all multicast groups that use the specified RP as the root of their shared trees.

The Join/Prune messages are sent either periodic or are triggered by some events. Typically, all Join messages and the (S,G,rpt) Prune messages are sent periodically.

4.1 Receiving (*,*,RP) Join/Prune Messages

Purpose: Verify that (*,*,RP) Join/Prune messages are received and processed properly.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.5.1

Discussion: When a PIM-SM router receives a PIM (*,*,RP) Join/Prune message, the per-interface (*,*,RP) state-machine should be updated appropriately. Typically, if an (*,*,RP) Join message is received, the interface it was received on should be added to the set of outgoing interfaces to forward packets destined for any group handled by the specified RP. If that set has just became non-empty, an (*,*,RP) Join should be sent toward the RP. If an (*,*,RP) Prune message is received, typically it should remove the interface it was received on from the set of outgoing interfaces to forward packets destined to any group handled by the specified RP. If that set has just became empty, an (*,*,RP) Prune message should be sent toward the RP.

Test Setup: Connect the RUT, TR1, TR2, TR3, S1, and S2 according to Figure 4.1 [1]. Enable PIM-SM on the RUT, TR1, TR2, and TR3. Configure S1 and S2 as senders for group 224.0.1.20.

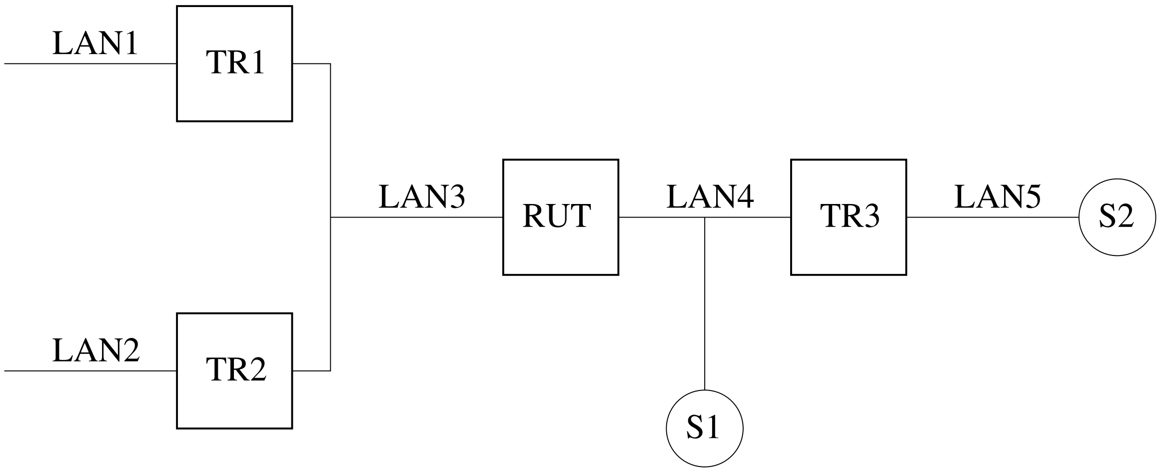

Figure 4.1: Receiving (*,*,RP) Join/Prune messages test setup

Procedure:

Part A: Receiving (,*,RP) Join messages at the RP.*

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.

2. Start observing the downstream (*,*,RP) per-interface state machine at the RUT.

3. Compose an (*,*,RP) Join message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Start S1, and observe the data packets transmitted by the RUT on LAN3.

[1] Note that S1 is used only when TR3 and S2 are not used, hence it is not necessary to have two senders at a time.

5. Wait until the downstream (*,*,RP) per-interface state in the RUT expires.

6. Observe the data packets transmitted by the RUT on LAN3.

Part B: Receiving (,*,RP) Prune messages at the RP.*

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.

2. Start observing the downstream (*,*,RP) per-interface state machine at the RUT.

3. Compose an (*,*,RP) Prune message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Compose an (*,*,RP) Join message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

5. Start S1, and observe the data packets transmitted by the RUT on LAN3.

6. Compose an (*,*,RP) Prune message at TR1 with the RP address set to the address of the RUT, and send it to the RUT.

7. Observe the messages and data packets transmitted by the RUT on LAN3.

Part C: Receiving (,*,RP) Join messages at non-RP router.*

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.

2. Start observing the downstream (*,*,RP) per-interface state machine at the RUT.

3. Compose an (*,*,RP) Join message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Observe the messages transmitted by the RUT on LAN4.

5. Start S2, and observe the data packets transmitted by the RUT on LAN3.

6. Wait until the downstream (*,*,RP) per-interface state in the RUT expires.

7. Observe the data packets transmitted by the RUT on LAN3.

Part D: Receiving (,*,RP) Prune messages at non-RP router.*

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.

2. Start observing the downstream (*,*,RP) per-interface state machine at the RUT.

3. Compose an (*,*,RP) Prune message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Compose an (*,*,RP) Join message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

5. Observe the messages transmitted by the RUT on LAN4.

6. Start S2, and observe the data packets transmitted by the RUT on LAN3.

7. Compose an (*,*,RP) Prune message at TR1 with the RP address set to the address of TR3, and send it to the RUT.

8. Observe the messages transmitted by the RUT on LAN3 and LAN4, and the data packets transmitted by the RUT on LAN3.

Part E: Receiving (,*,RP) Prune messages on a LAN.*
This part is same as Part D, except that in Step 1 we start TR2 as well.

Part F: Receiving (,*,RP) Join and Prune messages on a LAN.*
This part is same as Part E, except that in Step 2 we compose and send same (*,*,RP) Join message from TR2 as well.

Observable Results:

Part A:

- After the (*,*,RP) Join message is received by the RUT, it should create the appropriate (*,*,RP) multicast routing entry for that RP, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.0.0       10.3.0.1        10.3.0.1        RP
    Upstream interface (S):   UNKNOWN
    Upstream interface (RP):  register_vif
    Upstream MRIB next hop (RP): UNKNOWN
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):       UNKNOWN
    Upstream RPF'(S,G):       UNKNOWN
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           Joined
    Register state:
    Join timer:               52
    Local include:            ..............
    Local exclude:            ..............
    Local include WC:         ..............
    Local include SG:         ..............
    Local exclude SG:         ..............
    Joins RP:                 ......O.......
    Joins WC:                 ..............
    Joins SG:                 ..............
    Prunes SG_RPT:            ..............
```

```
Join state:                 ......O.......
Prune pending state:        ..............
Prune state:                ..............
I am assert winner state:   ..............
I am assert loser state:    ..............
Assert winner WC:           ..............
Assert winner SG:           ..............
Assert lost WC:             ..............
Could assert SG:            ..............
I am DR:                    .....OO.......
Immediate olist RP:         ......O.......
Immediate olist WC:         ..............
Immediate olist SG:         ..............
Inherited olist SG:         ..............
Inherited olist SG_RPT:     ..............
PIM include WC:             ..............
```

- After S1 is started, the multicast data packets should be forwarded by the RUT on LAN3.
- After `J/P_HoldTime` (210), the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state (*i.e.,* the (*,*,RP) state in the RUT should expire). As a result of that transition, no multicast packets should be forwarded by the RUT on LAN3.

Part B:

- After the (*,*,RP) Prune message is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (*,*,RP) multicast routing entry should be created).
- After that, the results until after S1 is started should be same as in Part A.
- After the (*,*,RP) Prune message is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state (*i.e.,* the (*,*,RP) state in the RUT should expire). As a result of that transition no multicast packets should be forwarded by the RUT on LAN3.

Part C:

- After the (*,*,RP) Join message is received by the RUT, it should create the appropriate (*,*,RP) multicast routing entry for that RP, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.0.0       10.9.0.1        10.9.0.1        RP
    Upstream interface (S):    UNKNOWN
    Upstream interface (RP):   dc1
    Upstream MRIB next hop (RP): 10.3.0.2
```

```
Upstream MRIB next hop (S):  UNKNOWN
Upstream RPF'(*,G):        UNKNOWN
Upstream RPF'(S,G):        UNKNOWN
Upstream RPF'(S,G,rpt):    UNKNOWN
Upstream state:            Joined
Register state:
Join timer:                47
Local include:             ..............
Local exclude:             ..............
Local include WC:          ..............
Local include SG:          ..............
Local exclude SG:          ..............
Joins RP:                  ......O.......
Joins WC:                  ..............
Joins SG:                  ..............
Prunes SG_RPT:             ..............
Join state:                ......O.......
Prune pending state:       ..............
Prune state:               ..............
I am assert winner state:  ..............
I am assert loser state:   ..............
Assert winner WC:          ..............
Assert winner SG:          ..............
Assert lost WC:            ..............
Could assert SG:           ..............
I am DR:                   ......O.......
Immediate olist RP:        ......O.......
Immediate olist WC:        ..............
Immediate olist SG:        ..............
Inherited olist SG:        ..............
Inherited olist SG_RPT:    ..............
PIM include WC:            ..............
```

Further, the RUT itself should originate an (*,*,RP) Join message toward the RP (TR3).

- After S2 is started, the multicast data packets forwarded by TR3 on LAN4 should be forwarded by the RUT on LAN3.
- After `J/P_HoldTime` (210), the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state (*i.e.,* the (*,*,RP) state in the RUT should expire). As a result of that transition, the RUT should send (*,*,RP) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3.

Part D:

- After the (*,*,RP) Prune message is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (*,*,RP) multicast routing entry should be created).

- After that, the results until after S2 is started should be same as in Part C.
- After the (*,*,RP) Prune message from TR1 is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state (*i.e.,* the (*,*,RP) state in the RUT should expire). As a result of that transition, the RUT should send (*,*,RP) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3. Note that because the RUT has only one PIM neighbor on LAN3, it does not need to send (*,*,RP) PruneEcho on LAN3.

Part E:

- The results until after S2 is started should be same as in Part C and D.
- After the (*,*,RP) Prune message from TR1 is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). After `J/P_Override_Interval(I)` (3 sec), the PrunePending timer on that interface should expire, and the (*,*,RP) state machine for the interface should send (*,*,RP) PruneEcho on LAN3 and transit to NoInfo state (*i.e.,* the (*,*,RP) state in the RUT should expire). As a result of that transition, the RUT should send (*,*,RP) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3.

Part F:

- The results until after S2 is started should be same as in Part C, D, and E.
- After the (*,*,RP) Prune message from TR1 is received by the RUT, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). Assuming that TR2 has (*,*,RP) multicast routing entry in Joined state for the RP it had originated (*,*,RP) Join message earlier, then after very short random interval `t_override` (rand(0,2.5) sec) TR2 should send another (*,*,RP) Join message to the RUT. After the RUT receives that (*,*,RP) Join message from TR2, the (*,*,RP) state machine for the interface that connects the RUT to LAN3 should transition back to Join state. As a result of that transition, the RUT should not send (*,*,RP) Prune message toward the RP (TR3), and the multicast packets should continue to be forwarded by the RUT on LAN3.

Possible Problems: None.

4.2 Receiving (*,G) Join/Prune Messages

Purpose: Verify that (*,G) Join/Prune messages are received and processed properly.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.5.2

Discussion: When a PIM-SM router receives a PIM (*,G) Join/Prune message, the per-interface (*,G) state-machine should be updated appropriately. Typically, if an (*,G) Join message is received, the interface it was received on should be added to the set of outgoing interfaces to forward packets destined for the specified multicast group address. If that set has just became non-empty, an (*,G) Join should be sent toward the RP for that group. If an (*,G) Prune message is received, typically it should remove the interface it was received on from the set of outgoing interfaces for that group. If that set has just became empty, an (*,G) Prune message should be sent toward the RP for that group.

Test Setup: Connect the RUT, TR1, TR2, TR3, S1, and S2 according to Figure 4.2 [2]. Enable PIM-SM on the RUT, TR1, TR2, and TR3. Configure S1 and S2 as senders for group 224.0.1.20.

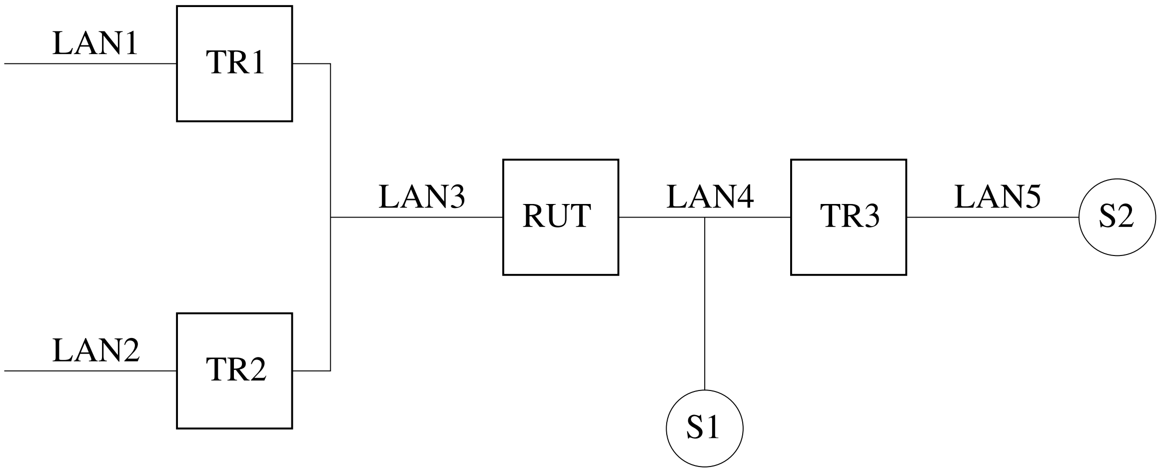

Figure 4.2: Receiving (*,G) Join/Prune messages test setup

Procedure:

Part A: Receiving (,G) Join messages at the RP.*

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start observing the downstream (*,G) per-interface state machine at the RUT.
3. Compose an (*,G) Join message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Start S1, and observe the data packets transmitted by the RUT on LAN3.

[2] Note that S1 is used only when TR3 and S2 are not used, hence it is not necessary to have two senders at a time.

5. Wait until the downstream (*,G) per-interface state in the RUT expires.
6. Observe the data packets transmitted by the RUT on LAN3.

Part B: Receiving (,G) Prune messages at the RP.*

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start observing the downstream (*,G) per-interface state machine at the RUT.
3. Compose an (*,G) Prune message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Compose an (*,G) Join message at TR1 with the RP address set to the address of the RUT, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
5. Start S1, and observe the data packets transmitted by the RUT on LAN3.
6. Compose an (*,G) Prune message at TR1 with the RP address set to the address of the RUT, and send it to the RUT.
7. Observe the messages and data packets transmitted by the RUT on LAN3.

Part C: Receiving (,G) Join messages at non-RP router.*

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.
2. Start observing the downstream (*,G) per-interface state machine at the RUT.
3. Compose an (*,G) Join message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Observe the messages transmitted by the RUT on LAN4.
5. Start S2, and observe the data packets transmitted by the RUT on LAN3.
6. Wait until the downstream (*,G) per-interface state in the RUT expires.
7. Observe the data packets transmitted by the RUT on LAN3.

Part D: Receiving (,G) Prune messages at non-RP router.*

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.
2. Start observing the downstream (*,G) per-interface state machine at the RUT.
3. Compose an (*,G) Prune message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Compose an (*,G) Join message at TR1 with the RP address set to the address of TR3, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

5. Observe the messages transmitted by the RUT on LAN4.

6. Start S2, and observe the data packets transmitted by the RUT on LAN3.

7. Compose an (*,G) Prune message at TR1 with the RP address set to the address of TR3, and send it to the RUT.

8. Observe the messages transmitted by the RUT on LAN3 and LAN4, and the data packets transmitted by the RUT on LAN3.

Part E: Receiving (,G) Prune messages on a LAN.*
This part is same as Part D, except that in Step 1 we start TR2 as well.

Part F: Receiving (,G) Join and Prune messages on a LAN.*
This part is same as Part E, except that in Step 2 we compose and send same (*,G) Join message from TR2 as well.

Part G: Receiving (,G) Join messages with mismatch RP address at non-RP router*

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.

2. Start observing the downstream (*,G) per-interface state machine at the RUT.

3. Compose an (*,G) Join message at TR1 with the RP address set to an address that is different from the address of TR3, but such that TR3 is the next-hop router toward it (*e.g.,* the address of S), and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).

4. Observe the messages transmitted by the RUT on LAN4.

Observable Results:

Part A:

- After the (*,G) Join message is received by the RUT, it should create the appropriate (*,G) multicast routing entry for that group, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      0.0.0.0         10.3.0.1        WC
    Upstream interface (S):   UNKNOWN
    Upstream interface (RP):  register_vif
    Upstream MRIB next hop (RP): UNKNOWN
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):       UNKNOWN
    Upstream RPF'(S,G):       UNKNOWN
    Upstream RPF'(S,G,rpt):   UNKNOWN
```

```
Upstream state:           Joined
Register state:
Join timer:               42
Local include:            ..............
Local exclude:            ..............
Local include WC:         ..............
Local include SG:         ..............
Local exclude SG:         ..............
Joins RP:                 ..............
Joins WC:                 ......O.......
Joins SG:                 ..............
Prunes SG_RPT:            ..............
Join state:               ......O.......
Prune pending state:      ..............
Prune state:              ..............
I am assert winner state: ..............
I am assert loser state:  ..............
Assert winner WC:         ..............
Assert winner SG:         ..............
Assert lost WC:           ..............
Could assert SG:          ..............
I am DR:                  .....OO.......
Immediate olist RP:       ..............
Immediate olist WC:       ......O.......
Immediate olist SG:       ..............
Inherited olist SG:       ..............
Inherited olist SG_RPT:   ..............
PIM include WC:           ..............
```

- After S1 is started, the multicast data packets should be forwarded by the RUT on LAN3.
- After `J/P_HoldTime` (210), the (*,G) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state (*i.e.,* the (*,G) state in the RUT should expire). As a result of that transition, no multicast packets should be forwarded by the RUT on LAN3.

Part B:

- After the (*,G) Prune message is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (*,G) multicast routing entry should be created).
- After that, the results until after S1 is started should be same as in Part A.
- After the (*,G) Prune message is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state (*i.e.,* the (*,G) state in the RUT should expire). As a result of that transition no multicast packets should be forwarded by the RUT on LAN3.

Part C:

- After the (*,G) Join message is received by the RUT, it should create the appropriate (*,G) multicast routing entry for that group, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      0.0.0.0         10.4.0.1        WC
    Upstream interface (S):   UNKNOWN
    Upstream interface (RP):  dc1
    Upstream MRIB next hop (RP): 10.3.0.2
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):       10.3.0.2
    Upstream RPF'(S,G):       UNKNOWN
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           Joined
    Register state:
    Join timer:               50
    Local include:            ..............
    Local exclude:            ..............
    Local include WC:         ..............
    Local include SG:         ..............
    Local exclude SG:         ..............
    Joins RP:                 ..............
    Joins WC:                 ......O.......
    Joins SG:                 ..............
    Prunes SG_RPT:            ..............
    Join state:               ......O.......
    Prune pending state:      ..............
    Prune state:              ..............
    I am assert winner state: ..............
    I am assert loser state:  ..............
    Assert winner WC:         ..............
    Assert winner SG:         ..............
    Assert lost WC:           ..............
    Could assert SG:          ..............
    I am DR:                  ......O.......
    Immediate olist RP:       ..............
    Immediate olist WC:       ......O.......
    Immediate olist SG:       ..............
    Inherited olist SG:       ..............
    Inherited olist SG_RPT:   ..............
    PIM include WC:           ..............
```

  Further, the RUT itself should originate an (*,G) Join message toward the RP (TR3).

- After S2 is started, the multicast data packets forwarded by TR3 on LAN4 should be forwarded by the RUT on LAN3.

- After `J/P_HoldTime` (210), the (*,G) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state (*i.e.,* the (*,G) state in the RUT should expire). As a result of that transition, the RUT should send (*,G) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3.

Part D:

- After the (*,G) Prune message is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (*,G) multicast routing entry should be created).
- After that, the results until after S2 is started should be same as in Part C.
- After the (*,G) Prune message from TR1 is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state (*i.e.,* the (*,G) state in the RUT should expire). As a result of that transition, the RUT should send (*,G) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3. Note that because the RUT has only one PIM neighbor on LAN3, it does not need to send (*,G) PruneEcho on LAN3.

Part E:

- The results until after S2 is started should be same as in Part C and D.
- After the (*,G) Prune message from TR1 is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). After `J/P_Override_Interval(I)` (3 sec), the PrunePending timer on that interface should expire, and the (*,G) state machine for the interface should send (*,G) PruneEcho on LAN3 and transit to NoInfo state (*i.e.,* the (*,G) state in the RUT should expire). As a result of that transition, the RUT should send (*,G) Prune message toward the RP (TR3), and no multicast packets should be forwarded by the RUT on LAN3.

Part F:

- The results until after S2 is started should be same as in Part C, D, and E.
- After the (*,G) Prune message from TR1 is received by the RUT, the (*,G) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). Assuming that TR2 has (*,G) multicast routing entry in Joined state for the RP it had originated (*,G) Join message earlier, then after very short random interval `t_override` (rand(0,2.5) sec) TR2 should send another (*,G) Join message to the RUT. After the RUT receives that (*,G) Join message from TR2, the (*,G) state machine for the interface that connects the RUT to LAN3 should transition back to Join state. As a result of that transition, the RUT should not send (*,G) Prune message toward the RP (TR3), and the multicast packets should continue to be forwarded by the RUT on LAN3.

Part G:

- After the RUT receives the (*,G) Join message with the mismatch RP address inside, it should silently ignore it without any further action.

Possible Problems: None.

4.3 Receiving (S,G) Join/Prune Messages

Purpose: Verify that (S,G) Join/Prune messages are received and processed properly.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.5.3

Discussion: When a PIM-SM router receives a PIM (S,G) Join/Prune message, the per-interface (S,G) state-machine should be updated appropriately. Typically, if an (S,G) Join message is received, the interface it was received on should be added to the set of outgoing interfaces to forward packets destined for the specified source address and multicast group. If that set has just became non-empty, an (S,G) Join should be sent toward the specified source. If an (S,G) Prune message is received, typically it should remove the interface it was received on from the set of outgoing interfaces for that source and group. If that set has just became empty, an (S,G) Prune message should be sent toward the source.

Test Setup: Connect the RUT, TR1, TR2, TR3, S1, and S2 according to Figure 4.3 [3]. Enable PIM-SM on the RUT, TR1, TR2, and TR3. Configure S1 and S2 as senders for group 224.0.1.20.

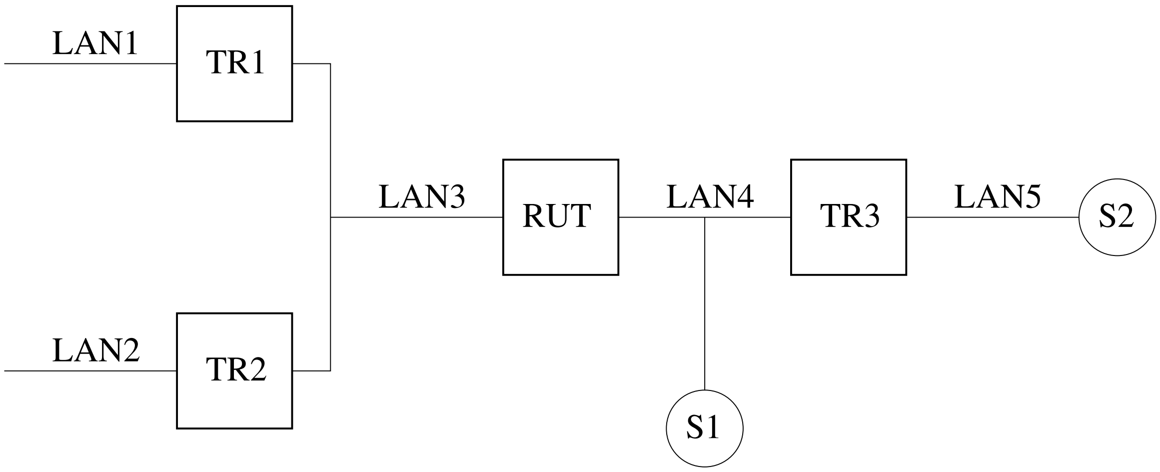

Figure 4.3: Receiving (S,G) Join/Prune messages test setup

Procedure:

Part A: Receiving (S,G) Join messages at the first-hop router.

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start observing the downstream (S,G) per-interface state machine at the RUT.
3. Compose an (S,G) Join message at TR1 with the source address set to the address of S1, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Start S1, and observe the data packets transmitted by the RUT on LAN3.

[3]Note that S1 is used only when TR3 and S2 are not used, hence it is not necessary to have two senders at a time.

5. Wait until the downstream (S,G) per-interface state in the RUT expires.
6. Observe the data packets transmitted by the RUT on LAN3.
7. Stop S1.

Part B: Receiving (S,G) Prune messages at the first-hop router.

1. Configure the RUT as the RP. Start the RUT and TR1. If necessary, wait until the RP-set in the RUT and TR1 converges.
2. Start observing the downstream (S,G) per-interface state machine at the RUT.
3. Compose an (S,G) Prune message at TR1 with the source address set to the address of S1, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Compose an (S,G) Join message at TR1 with the source address set to the address of S1, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
5. Start S1, and observe the data packets transmitted by the RUT on LAN3.
6. Compose an (S,G) Prune message at TR1 with the source address set to the address of S1, and send it to the RUT.
7. Observe the messages and data packets transmitted by the RUT on LAN3.
8. Stop S1.

Part C: Receiving (S,G) Join messages at non-first-hop router.

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.
2. Start observing the downstream (S,G) per-interface state machine at the RUT.
3. Compose an (S,G) Join message at TR1 with the source address set to the address of S2, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Observe the messages transmitted by the RUT on LAN4.
5. Start S2, and observe the data packets transmitted by the RUT on LAN3.
6. Wait until the downstream (S,G) per-interface state in the RUT expires.
7. Observe the data packets transmitted by the RUT on LAN3.
8. Stop S2.

Part D: Receiving (S,G) Prune messages at non-first-hop router.

1. Configure TR3 as the RP. Start the RUT, TR1, and TR3. If necessary, wait until the RP-set in the RUT, TR1, and TR3 converges.

2. Start observing the downstream (S,G) per-interface state machine at the RUT.
3. Compose an (S,G) Prune message at TR1 with the source address set to the address of S2, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
4. Compose an (S,G) Join message at TR1 with the source address set to the address of S2, and send it to the RUT. The `J/P_HoldTime` of the message should be set to its default value (210).
5. Observe the messages transmitted by the RUT on LAN4.
6. Start S2, and observe the data packets transmitted by the RUT on LAN3.
7. Compose an (S,G) Prune message at TR1 with the source address set to the address of S2, and send it to the RUT.
8. Observe the messages transmitted by the RUT on LAN3 and LAN4, and the data packets transmitted by the RUT on LAN3.
9. Stop S2.

Part E: Receiving (S,G) Prune messages on a LAN.
This part is same as Part D, except that in Step 1 we start TR2 as well.

Part F: Receiving (S,G) Join and Prune messages on a LAN.
This part is same as Part E, except that in Step 2 we compose and send same (S,G) Join message from TR2 as well.

Observable Results:

Part A:

- After the (S,G) Join message is received by the RUT, it should create the appropriate (S,G) multicast routing entry for that source and group, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.3.0.2        10.3.0.1        SG DirectlyConnectedS
    Upstream interface (S):    dc1
    Upstream interface (RP):   register_vif
    Upstream MRIB next hop (RP): UNKNOWN
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):        UNKNOWN
    Upstream RPF'(S,G):        UNKNOWN
    Upstream RPF'(S,G,rpt):    UNKNOWN
    Upstream state:            Joined
    Register state:            RegisterNoinfo RegisterNotCouldRegister
    Join timer:                51
    Local include:             ..............
    Local exclude:             ..............
    Local include WC:          ..............
```

```
    Local include SG:         ..............
    Local exclude SG:         ..............
    Joins RP:                 ..............
    Joins WC:                 ..............
    Joins SG:                 ......O.......
    Prunes SG_RPT:            ..............
    Join state:               ......O.......
    Prune pending state:      ..............
    Prune state:              ..............
    I am assert winner state: ..............
    I am assert loser state:  ..............
    Assert winner WC:         ..............
    Assert winner SG:         ..............
    Assert lost WC:           ..............
    Could assert SG:          ..............
    I am DR:                  .....OO.......
    Immediate olist RP:       ..............
    Immediate olist WC:       ..............
    Immediate olist SG:       ......O.......
    Inherited olist SG:       ......O.......
    Inherited olist SG_RPT:   ..............
    PIM include WC:           ..............
```

- After S1 is started, the multicast data packets should be forwarded by the RUT on LAN3. In addition, the SPT flag for the entry should be set:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.3.0.2        10.3.0.1        SG SPT DirectlyConnectedS
    Upstream interface (S):   dc1
    Upstream interface (RP):  register_vif
    Upstream MRIB next hop (RP): UNKNOWN
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):       UNKNOWN
    Upstream RPF'(S,G):       UNKNOWN
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           Joined
    Register state:           RegisterNoinfo RegisterNotCouldRegister
    Join timer:               19
    Local include:            ..............
    Local exclude:            ..............
    Local include WC:         ..............
    Local include SG:         ..............
    Local exclude SG:         ..............
    Joins RP:                 ..............
    Joins WC:                 ..............
    Joins SG:                 ......O.......
```

```
    Prunes SG_RPT:              ..............
    Join state:                 ......O.......
    Prune pending state:        ..............
    Prune state:                ..............
    I am assert winner state: ..............
    I am assert loser state:  ..............
    Assert winner WC:           ..............
    Assert winner SG:           ..............
    Assert lost WC:             ..............
    Could assert SG:            ......O.......
    I am DR:                    .....OO.......
    Immediate olist RP:         ..............
    Immediate olist WC:         ..............
    Immediate olist SG:         ......O.......
    Inherited olist SG:         ......O.......
    Inherited olist SG_RPT:     ..............
    PIM include WC:             ..............
```

- After `J/P_HoldTime` (210), the (S,G) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state. As a result of that transition, no multicast packets should be forwarded by the RUT on LAN3. However, the (S,G) state itself should not be removed yet, because the directly-connected sender is still active:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.3.0.2        10.3.0.1        SG DirectlyConnectedS
    Upstream interface (S):    dc1
    Upstream interface (RP):   register_vif
    Upstream MRIB next hop (RP): UNKNOWN
    Upstream MRIB next hop (S):  UNKNOWN
    Upstream RPF'(*,G):        UNKNOWN
    Upstream RPF'(S,G):        UNKNOWN
    Upstream RPF'(S,G,rpt):    UNKNOWN
    Upstream state:            NotJoined
    Register state:            RegisterNoinfo RegisterNotCouldRegister
    Join timer:                -1
    Local include:             ..............
    Local exclude:             ..............
    Local include WC:          ..............
    Local include SG:          ..............
    Local exclude SG:          ..............
    Joins RP:                  ..............
    Joins WC:                  ..............
    Joins SG:        B          ..............
    Prunes SG_RPT:             ..............
    Join state:                ..............
    Prune pending state:       ..............
```

```
    Prune state:                 ..............
    I am assert winner state:    ..............
    I am assert loser state:     ..............
    Assert winner WC:            ..............
    Assert winner SG:            ..............
    Assert lost WC:              ..............
    Could assert SG:             ..............
    I am DR:                     .....OO.......
    Immediate olist RP:          ..............
    Immediate olist WC:          ..............
    Immediate olist SG:          ..............
    Inherited olist SG:          ..............
    Inherited olist SG_RPT:      ..............
    PIM include WC:              ..............
```

- After the sender is stopped, and after it has been inactive for `Keepalive_Period` (210 sec), the (S,G) state should expire.

Part B:

- After the (S,G) Prune message is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (S,G) multicast routing entry should be created).
- After that, the results until after S1 is started should be same as in Part A.
- After the (S,G) Prune message is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state. As a result of that transition no multicast packets should be forwarded by the RUT on LAN3. However, the (S,G) state itself should not be removed yet, because the directly-connected sender is still active (see Part A).
- After the sender is stopped, and after it has been inactive for `Keepalive_Period` (210 sec), the (S,G) state should expire.

Part C:

- After the (S,G) Join message is received by the RUT, it should create the appropriate (S,G) multicast routing entry for that source and group, and the interface toward LAN3 should be in Join state and added to the set of outgoing interface for that entry:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.4.0.2        10.4.0.1        SG
    Upstream interface (S):    dc1
    Upstream interface (RP):   dc1
    Upstream MRIB next hop (RP): 10.3.0.2
    Upstream MRIB next hop (S):  10.3.0.2
    Upstream RPF'(*,G):        UNKNOWN
```

```
    Upstream RPF'(S,G):       10.3.0.2
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           Joined
    Register state:
    Join timer:               47
    Local include:            ..............
    Local exclude:            ..............
    Local include WC:         ..............
    Local include SG:         ..............
    Local exclude SG:         ..............
    Joins RP:                 ..............
    Joins WC:                 ..............
    Joins SG:                 ......O.......
    Prunes SG_RPT:            ..............
    Join state:               ......O.......
    Prune pending state:      ..............
    Prune state:              ..............
    I am assert winner state: ..............
    I am assert loser state:  ..............
    Assert winner WC:         ..............
    Assert winner SG:         ..............
    Assert lost WC:           ..............
    Could assert SG:          ..............
    I am DR:                  ......O.......
    Immediate olist RP:       ..............
    Immediate olist WC:       ..............
    Immediate olist SG:       ......O.......
    Inherited olist SG:       ......O.......
    Inherited olist SG_RPT:   ..............
    PIM include WC:           ..............
```

Further, the RUT itself should originate an (S,G) Join message toward the source (S2).

- After S2 is started, the multicast data packets forwarded by TR3 on LAN4 should be forwarded by the RUT on LAN3. Further, the SPT-bit for the entry should be set:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.4.0.2        10.4.0.1        SG SPT
    Upstream interface (S):   dc1
    Upstream interface (RP):  dc1
    Upstream MRIB next hop (RP): 10.3.0.2
    Upstream MRIB next hop (S):  10.3.0.2
    Upstream RPF'(*,G):       UNKNOWN
    Upstream RPF'(S,G):       10.3.0.2
    Upstream RPF'(S,G,rpt):   UNKNOWN
    Upstream state:           Joined
```

```
    Register state:
    Join timer:                 19
    Local include:              ..............
    Local exclude:              ..............
    Local include WC:           ..............
    Local include SG:           ..............
    Local exclude SG:           ..............
    Joins RP:                   ..............
    Joins WC:                   ..............
    Joins SG:                   ......O.......
    Prunes SG_RPT:              ..............
    Join state:                 ......O.......
    Prune pending state:        ..............
    Prune state:                ..............
    I am assert winner state:   ..............
    I am assert loser state:    ..............
    Assert winner WC:           ..............
    Assert winner SG:           ..............
    Assert lost WC:             ..............
    Could assert SG:            ......O.......
    I am DR:                    ......O.......
    Immediate olist RP:         ..............
    Immediate olist WC:         ..............
    Immediate olist SG:         ......O.......
    Inherited olist SG:         ......O.......
    Inherited olist SG_RPT:     ..............
    PIM include WC:             ..............
```

- After `J/P_HoldTime` (210), the (S,G) state machine for the interface that connects the RUT to LAN3 should timeout and transition to NoInfo state As a result of that transition, the RUT should send (S,G) Prune message toward the source (S2), and no multicast packets should be forwarded by the RUT on LAN3. Note that if the implementation does not remove (S,G) states that have the Keepalive Timer running, the entry may not be removed yet:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      10.4.0.2        10.4.0.1        SG
    Upstream interface (S):    dc1
    Upstream interface (RP):   dc1
    Upstream MRIB next hop (RP): 10.3.0.2
    Upstream MRIB next hop (S):  10.3.0.2
    Upstream RPF'(*,G):        UNKNOWN
    Upstream RPF'(S,G):        10.3.0.2
    Upstream RPF'(S,G,rpt):    UNKNOWN
    Upstream state:            NotJoined
    Register state:
    Join timer:                -1
```

```
Local include:              ..............
Local exclude:              ..............
Local include WC:           ..............
Local include SG:           ..............
Local exclude SG:           ..............
Joins RP:                   ..............
Joins WC:                   ..............
Joins SG:                   ..............
Prunes SG_RPT:              ..............
Join state:                 ..............
Prune pending state:        ..............
Prune state:                ..............
I am assert winner state:   ..............
I am assert loser state:    ..............
Assert winner WC:           ..............
Assert winner SG:           ..............
Assert lost WC:             ..............
Could assert SG:            ..............
I am DR:                    ......O.......
Immediate olist RP:         ..............
Immediate olist WC:         ..............
Immediate olist SG:         ..............
Inherited olist SG:         ..............
Inherited olist SG_RPT:     ..............
PIM include WC:             ..............
```

- After the sender is stopped, and after it has been inactive for `Keepalive_Period` (210 sec), the (S,G) state should expire if it was not removed earlier.

Part D:

- After the (S,G) Prune message is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should continue to stay in the NoInfo state (*i.e.,* no (S,G) multicast routing entry should be created).
- After that, the results until after S2 is started should be same as in Part C.
- After the (S,G) Prune message from TR1 is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should transition to NoInfo state. This may or may not remove the (S,G) state itself (see the observable results in Part C). As a result of that transition, the RUT should send (S,G) Prune message toward the source (S2), and no multicast packets should be forwarded by the RUT on LAN3. Note that because the RUT has only one PIM neighbor on LAN3, it does not need to send (S,G) PruneEcho on LAN3.

Part E:

- The results until after S2 is started should be same as in Part C and D.

- After the (S,G) Prune message from TR1 is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). After `J/P_Override_Interval(I)` (3 sec), the PrunePending timer on that interface should expire, and the (S,G) state machine for the interface should send (S,G) PruneEcho on LAN3 and transit to NoInfo state. This may or may not remove the (S,G) state itself (see the observable results in Part C). As a result of that transition, the RUT should send (S,G) Prune message toward the source (S2), and no multicast packets should be forwarded by the RUT on LAN3.

Part F:

- The results until after S2 is started should be same as in Part C, D, and E.
- After the (S,G) Prune message from TR1 is received by the RUT, the (S,G) state machine for the interface that connects the RUT to LAN3 should transition to PrunePending state (the reason that it does not transit to NoInfo instead is because the RUT has more than one PIM neighbors on that interface). Assuming that TR2 has (S,G) multicast routing entry in Joined state for the source it had originated (S,G) Join message earlier, then after very short random interval `t_override` (rand(0,2.5) sec) TR2 should send another (S,G) Join message to the RUT. After the RUT receives that (S,G) Join message from TR2, the (S,G) state machine for the interface that connects the RUT to LAN3 should transition back to Join state. As a result of that transition, the RUT should not send (S,G) Prune message toward the source (S2), and the multicast packets should continue to be forwarded by the RUT on LAN3.

Possible Problems: None.

Test Group 5

PIM Assert Messages

Scope: Test sending and receiving of PIM Assert messages.

Overview: TODO

Test Group 6

Bootstrap Mechanism

Scope: TODO

Overview: TODO

6.1 Single BSR Election

Purpose: Test the Bootstrap self-election on a single router.

References:

- draft-ietf-pim-sm-bsr-02 – Section 3

Discussion: A candidate-BSR must be able to elect itself as the Bootstrap router if there are no other candidate-BSRs. The time to complete the election is `BS_Timeout` (130 sec).

Test Setup: Connect the RUT according to Figure 6.1. Configure the RUT as a Cand-BSR.

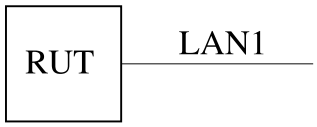

Figure 6.1: Single BSR election test setup

Procedure:

1. Start the RUT.
2. Check immediately the BSR status on the RUT. Initially the RUT should be the BSR in Pending state:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout
10.3.0.1           1 10.3.0.1           1 Pending               11
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout
10.3.0.1           1 10.3.0.1           1 Init                  -1
```

3. After `BS_Timeout` (130 sec), the RUT must become the elected BSR:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout
10.3.0.1           1 10.3.0.1           1 Elected               59
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout
10.3.0.1           1 10.3.0.1           1 Init                  -1
```

Observable Results: TODO

Possible Problems: The RUT might become the BSR earlier than `BS_Timeout` (130 sec) if the implementation uses a shorter timeout interval on startup (*e.g.,* if it has tuned for fast start-up).

6.2 Single Router RP-Set

Purpose: Test if the Bootstrap-derived RP-Set works on a single router.

References:

- draft-ietf-pim-sm-bsr-02 – Section 3

Discussion: A router that is both a candidate-BSR and a candidate-RP must be able to include itself to the Cand-RP-Set. The time to complete the election is `BS_Timeout` + up to (`BS_Period` + `C-RP-Adv-Period`) (*i.e.,* $\leq$ (130 sec) + (130 sec) + (60 sec)).

Test Setup: Connect the RUT according to Figure 6.2. Configure the RUT as a Cand-BSR and a Cand-RP.

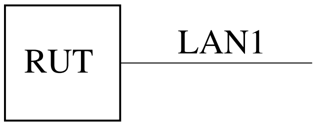

Figure 6.2: Single router RP-Set test setup

Procedure:

1. Start the RUT.

2. Check immediately the RP-Set on the RUT. Initially the RUT should be empty:

```
Xorp> show pim rps
RP              Type      Pri Holdtime Timeout ActiveGroups GroupPrefix
```

3. After up to `BS_Timeout` + `BS_Period` + `C-RP-Adv-Period` (*i.e.,* $\leq$ (130 sec) + (130 sec) + (60 sec)), the RUT must have a non-empty RP-Set with itself included in that set:

```
Xorp> show pim rps
RP              Type      Pri Holdtime Timeout ActiveGroups GroupPrefix
10.3.0.1        bootstrap 192       60      -1            0 224.0.0.0/4
```

Observable Results: TODO

Possible Problems: The RP-Set in the RUT might become non-empty much earlier than `BS_Timeout` + `BS_Period` + `C-RP-Adv-Period` (*i.e.,* $\leq$ (130 sec) + (130 sec) + (60 sec)) if the implementation uses shorter timeout intervals on startup (*e.g.,* if it has tuned for fast start-up).

6.3 Bootstrap Timeout

Purpose: Test the Bootstrap timeout between two routers.

References:

- draft-ietf-pim-sm-bsr-02 – Section 3

Discussion: If the elected BSR goes away, it should timeout in the state of other routers. If it is gracefully shutdown, the BSR should send a Bootstrap message with holdtime of 0, *i.e.,* the timeout should be almost immediately. If the BSR crashes, it should take up to `BS_Timeout` (130 sec) to timeout the state.

Test Setup: Connect the RUT and TR1 according to Figure 6.3. Configure TR1 as a Cand-BSR and a Cand-RP. Do not configure the RUT as a Cand-BSR or a Cand-RP.

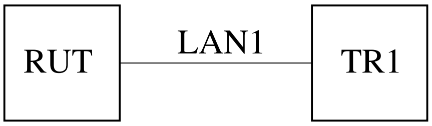

Figure 6.3: Bootstrap timeout test setup

Procedure:

1. Start the RUT.

2. Check that the RUT does not have any BSR or RP-related entries:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout

Xorp> show pim rps
RP              Type      Pri Holdtime Timeout ActiveGroups GroupPrefix

Xorp> show pim join all
Group           Source          RP              Flags
```

3. Start TR1.

4. After up to `BS_Timeout` + `BS_Period` + `C-RP-Adv-Period` (*i.e.,* $\leq$ (130 sec) + (130 sec) + (60 sec)), the RUT must have a non-empty RP-Set with TR1 included in that set:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout
```

```
10.3.0.1           1 0.0.0.0           0 AcceptPreferred      122
Configured zones:
BSR               Pri LocalAddress     Pri State            Timeout

Xorp> show pim rps
RP               Type      Pri Holdtime Timeout ActiveGroups GroupPrefix
10.3.0.1         bootstrap 192       60      19            0 224.0.0.0/4

Xorp> show pim join all
Group            Source           RP               Flags
224.0.0.0        10.3.0.1         10.3.0.1         RP
    Upstream interface (S): UNKNOWN
    Upstream interface (RP): dc2
    Upstream state: NotJoined
    Register state:
    Join timer: -1
    Local include:             .............
    Local exclude:             .............
    Local include WC:          .............
    Local include SG:          .............
    Local exclude SG:          .............
    Joins RP:                  .............
    Joins WC:                  .............
    Joins SG:                  .............
    Prunes SG_RPT:             .............
    Join state:                .............
    Prune pending state:       .............
    Prune state:               .............
    I am assert winner state: .............
    I am assert loser state:  .............
    Assert winner WC:          .............
    Assert winner SG:          .............
    Assert lost WC:            .............
    Could assert SG:           .............
    I am DR:                   ....O........
    Immediate olist RP:        .............
    Immediate olist WC:        .............
    Immediate olist SG:        .............
    Inherited olist SG:        .............
    Inherited olist SG_RPT:    .............
    PIM include WC:            .............
```

5. Stop TR1.

6. Wait until the BSR entry in the RUT timeout (up to `BS_Period` (60 sec)). After that the RUT should show the BSR entry as expired, though it may still cache it in the scope zone entry. However, the RP-related entry might be deleted (though the spec allows caching them):

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
10.3.0.1           1 0.0.0.0            0 AcceptAny              0      1369
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout

Xorp> show pim rps
RP              Type      Pri Holdtime Timeout ActiveGroups GroupPrefix
Xorp> show pim join all
Group           Source          RP              Flags
```

7. After the Scope Zone in the RUT timeout, the RUT should not contain any scope zone entries:

```
Xorp> show pim bootstrap
Active zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
Configured zones:
BSR              Pri LocalAddress     Pri State            Timeout SZTimeout
```

Observable Results: TODO

Possible Problems: TODO

Test Group 7

Multicast Routing Table

Scope: TODO

Overview: TODO

7.1 Single Router Same-LAN Membership

Purpose: Test the same-LAN membership on a single router.

References:

- draft-ietf-pim-sm-v2-new-05 – Section 4.1.6

Discussion: A router that has same-LAN group-specific or source-group-specific member must be able to detect it and create the appropriate MRE state in the MRT.

Test Setup: Connect the RUT and Rx1 according to Figure 7.1. Configure the RUT as a Cand-BSR and a Cand-RP.

Figure 7.1: Single router Same-LAN membership test setup

Procedure:

Part A: Group-specific same-LAN membership.

1. Start the RUT.
2. Wait until the RP-Set of the RUT becomes non-empty (*e.g.,* see 6.2).
3. Test that the RUT has no local membership state:

```
Xorp> show pim join
Group           Source          RP              Flags
```

4. Start receiver Rx for group 224.0.1.20. For example:

```
root@xorp1[42] msend -i 10.2.0.1 -r 224.0.1.20/12345
```

5. Test that the RUT now has local membership for group 224.0.1.20:

```
Xorp> show pim join
Group           Source          RP              Flags
224.0.1.20      0.0.0.0         10.3.0.1        WC
    Upstream interface (S): UNKNOWN
    Upstream interface (RP): register_vif
    Upstream state: Joined
    Register state:
    Join timer: 57
    Local include:             ......O.......
```

```
Local exclude:              ..............
Local include WC:           ......O.......
Local include SG:           ..............
Local exclude SG:           ..............
Joins RP:                   ..............
Joins WC:                   ..............
Joins SG:                   ..............
Prunes SG_RPT:              ..............
Join state:                 ..............
Prune pending state:        ..............
Prune state:                ..............
I am assert winner state:   ..............
I am assert loser state:    ..............
Assert winner WC:           ..............
Assert winner SG:           ..............
Assert lost WC:             ..............
Could assert SG:            ..............
I am DR:                    .....OO.......
Immediate olist RP:         ..............
Immediate olist WC:         ......O.......
Immediate olist SG:         ..............
Inherited olist SG:         ..............
Inherited olist SG_RPT:     ..............
PIM include WC:             ......O.......
```

6. Stop receiver Rx

7. Test that the RUT now has no local membership for group 224.0.1.20:

```
Xorp> show pim join
Group            Source          RP              Flags
```

Part B: Source-group-specific same-LAN membership.

1. Start the RUT.

2. Wait until the RP-Set of the RUT becomes non-empty (*e.g.,* see 6.2).

3. Test that the RUT has no local membership state:

```
Xorp> show pim join
Group            Source          RP              Flags
```

4. TODO: Start receiver Rx for...

5. TODO: Test that the RUT now has local membership for...

6. Stop receiver Rx

7. TODO: Test that the RUT now has no local membership for...

```
Xorp> show pim join
Group           Source          RP              Flags
```

Observable Results: TODO

Possible Problems: TODO

7.2 Template

Purpose: TODO

References:

- draft-ietf-pim-sm-v2-new-05 – Section TODO

Discussion: TODO

Test Setup: TODO

Procedure: TODO

Observable Results: TODO

Possible Problems: TODO