PIM Protocol

Teldat-Dm 804-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat-Dm 762-I IGMP Protocol

Chapter 1 Introduction

1.1 Introduction

PIM is a family of multicast routing protocols for the Internet protocol IP that provides one-to-many and many-to-many distribution of data over a LAN, WAN or Internet. For an initial introduction to IP multicast, please see manual **Teldat-Dm762-I IGMP Protocol** .

It is termed *protocol-independent* because PIM does not include its own topology discovery mechanism, but instead uses routing information supplied by other traditional routing protocols such as RIP, OSPF or BGP. What PIM does is to construct distribution trees from the sources to the receivers, based on the information received from the topology.

There are four variants of PIM:

(1) **PIM Sparse Mode (PIM-SM)**
PIM-SM explicitly builds unidirectional *shared trees* rooted at a *rendezvous point* (RP) per group, and optionally creates *shortest-path trees* per source. PIM-SM generally scales fairly well for wide-area usage (WAN). RFC 4601 describes the specifications for PIM-SM version 2.

(2) **PIM Dense Mode (PIM-DM)**
PIM-DM implicitly builds shortest-path trees by flooding multicast traffic domain wide, and then pruning back branches of the tree where no receivers are present. PIM-DM is straightforward to implement but generally has poor scaling properties. RFC 3973 describes the specifications for PIM-SM.

(3) **Bidirectional PIM (BIDIR-PIM)**
BIDIR-PIM explicitly builds *shared bi-directional trees*. It never builds a shortest path tree, so may have longer end-to-end delays than PIM-SM, but scales well because it needs no source-specific state. RFC 5015 describes the specifications for BIDIR-PIM.

(4) **PIM Source-Specific Multicast (PIM-SSM)**
PIM-SSM builds trees that are rooted in just one source, offering a more secure and scalable model for a limited amount of applications (mostly broadcasting of content). In SSM, an IP datagram is transmitted by a source S to an SSM destination address G, and receivers can receive this datagram by subscribing to channel (S,G). RFCs 3569, 4607 and 4608 describe the specifications for PIM-SSM.

Of the four variants, PIM-SM has the widest deployment.

The PIM variants that are supported in our routers are explained below.

1.2 PIM Sparse Mode (PIM-SM)

PIM Sparse-Mode is a protocol for efficiently routing IP packets to multicast groups that may span wide-area and inter-domain internets.

The protocol is named *protocol-independent* because it is not dependent on any particular unicast routing protocol for topology discovery, and *sparse-mode* because it is suitable for groups where a very low percentage of the nodes (and their routers) will subscribe to the multicast session.

The following sections present the concepts that PIM handles and describes their functionality.

1.2.1 Definitions

1.2.1.1 Distribution Trees

PIM broadcasts multicast traffic from a source to all receivers that have expressed an interest in the said traffic. PIM ensures that the traffic is only routed over the essential interfaces.

There is a single interface towards the root (*incoming interface* / *iif)* in each node forming part of the tree, and one or various interfaces (*outgoing interfaces* / *oif)*, towards the leaves, which in this case are the receivers.

There are two types of trees.

1.2.1.1.1 Shared Tree/RP Tree

- In this case the root or the traffic source is a node on the network known as *Rendezvous-Point (RP)*, which concentrates all the traffic destined to the same multicast group and ignores the source generating the traffic (IP source for the multicast packets).

1.2.1.1.2 Shortest Path Tree (SPT)

- This tree has a single traffic transmitter source as its root, which is destined to a specific multicast group. As the traffic doesn't have to cross any node where the traffic is concentrated, the tree always provides the shortest path between the source and the receivers.

1.2.1.2 Roles

1.2.1.2.1 Rendezvous Point (RP)

- An RP is a router that has been configured to be used as the distribution tree root for a multicast group that is not from a specified source.

1.2.1.2.2 Designated Router (DR)

- A LAN such as Ethernet can have multiple PIM-SM routers connected to it. Only one of these routers, the DR, acts for the directly connected *hosts* with respect to the PIM-SM protocol. A single DR is selected per interface through a simple election process.

1.2.1.2.3 Last Hop Router (LHR)

- Within a distribution tree, the LHR is the first PIM router the multicast receivers come across which must be directly connected to them.

1.2.1.3 PIM Messages

The PIM sends its control messages using IP number 103. Some of these messages are sent in unicast packets, while others are sent in multicast mode with TTL=1, with destination 'ALL_PIM_ROUTERS' (224.0.0.13).

The following types of messages can be found in PIM-SM:

1.2.1.3.1 HELLO

These are periodically sent through all the interfaces for neighbor discovery.

1.2.1.3.2 JOIN/PRUNE

- Routers send these messages to an *upstream* router going toward the source or toward the RP, to build and maintain the corresponding tree.

1.2.1.3.3 REGISTER

- This is a unicast message that encapsulates a multicast packet which the DR sends to the RP to be distributed through the *Shared-Tree* .

1.2.1.3.4 REGISTER-STOP

- The RP sends this message to the router that sent a REGISTER message, to prevent it from sending REGISTER messages.

1.2.1.3.5 ASSERT

- This is used to resolve resending conflicts between routers in a link.

1.2.1.3.6 BOOTSRAP (BSM)

- Allows a router to advertise itself as a *PIM Bootstrap Router* (BSR) candidate, while including dynamic entries from the RP-Set.

1.2.1.3.7 CANDIDATE-RP-ADVERTISEMENT

- Allows a router to advertise itself as a potential RP. This message is sent directly to the BSR, which is reponsible for collecting all applications and forming the RP-Set.

Figure 1 shows the common header for all the PIM version 2 messages.

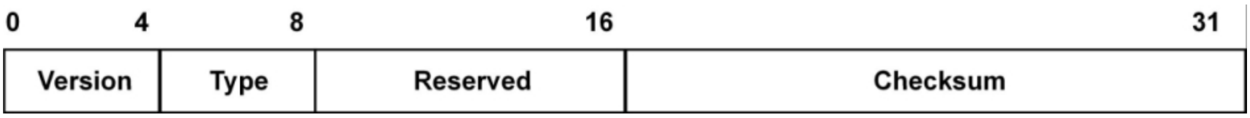

Fig. 1: Messages PIMv2: header format

Table describes the most significant fields in the PIMv2 message header.
PIMv2 header fields: Description.

Field	Description
Version	The PIM version is always 2 as it is the only existing version.
Type	This number changes depending on the type of message: • Hello # 0 • Register # 1 • Register-Stop # 2 • Join/Prune # 3 • Bootstrap # 4 • Assert # 5 • Candidate-RP-Advertisement # 8

1.2.1.4 PIM-SM Protocol States

This section specifies the database for the protocol that the PIM implementation must maintain in order to function correctly.

In RFC 4601 this database is known as *Tree Information Base* (TIB).

The database is made up of different types of states. Each of these states refers to multicast traffic with different specification levels: from the most specific to the most global. These states group all the variables that the PIM considers for the type of traffic in question: incoming and outgoing interfaces, timers, associated RP, etc.

The existing states are as follows:

1.2.1.4.1 State (*,*,RP)

This refers to all the multicast traffic managed by a given RP.

1.2.1.4.2 State (*,G)

This refers to multicast traffic destined for a multicast group 'G', regardless of the source of traffic. This builds and maintains the *Shared -Tree.*

1.2.1.4.3 State (S,G)

This refers to multicast traffic destined for a multicast group 'G', but only from a specific source 'S'. This builds and maintains the *Shortest -Path-Tree* .

1.2.1.4.4 State (S,G,rpt)

This includes the same traffic as the state (S, G), but in the *Shared -Tree* , i.e. where the tree root is the RP associated with Group 'G'. It occurs when the states (S, G) and (*, G) coexist so that traffic can be blocked from (S, G) from the *Shared -Tree* to exclusively arrive through the *Shortest-Path-Tree* .

1.2.2 Neighbor Discovery

When Router A with PIM configured initiates, it sends HELLO messages over all the interfaces where PIM-SM is enabled. Through this it reports its main IP address together with other protocol options for this interface. Each of the messages will only reach those routers that are directly connected. When this message is received by Router B, which also supports PIM, B includes router A in its list of neighbors with the information provided in the message. So that A also knows that B exists, this latter sends its own HELLO message to A so both become PIM neighbors. Being a PIM neighbor is a prerequisite in order to be able to accept any other PIM multicast message.

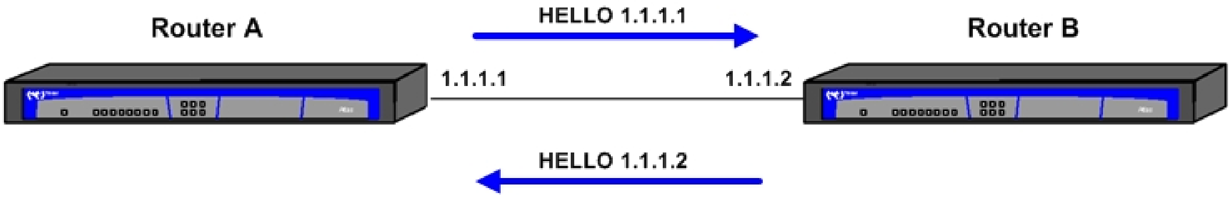

Fig. 2: Neighbor Discovery

To keep track of neighbors, a lifetime is included in the HELLO messages during which it is considered that the neighbor is present. HELLO messages are forwarded periodically, so that if the lifetime is exceeded without receiving a new HELLO from the neighbor, it is considered that the PIM neighbor is no longer available and all information related to it is deleted.

1.2.3 Electing the DR

One of the PIM neighbors on the same link must be elected to execute the role of DR. This element fulfills two functions: to export the multicast traffic generated by the local link sources to the rest of the PIM domain and to deliver the multicast traffic requested by the receiver present in the link to the subnet. It is essential for the second part that the router selected as DR is also configured as an IGMP router, even if it's not the IGMP *Querier*.

There are two criterions established in order to elect the DR, so that all PIM routers in a link reach the same election:

(1) If all PIM routers present on the link include the "DR-Priority" in their HELLO messages, the router with the highest priority value ("DR-Priority" highest) is the elected DR.

(2) If a PIM router does not include the "DR-Priority", or in cases where there is a tie with the highest priority, the router elected as DR is the one with the highest IP address.

Remember that for the IGMP *Querier* election, the router with lower IP address is elected instead. There may be conflicts as to who is responsible for broadcasting multicast traffic on a link if there is a router that does not have PIM IGMP enabled on this interface

1.2.4 Reverse Path Forwarding

In trees built by PIM, the multicast traffic is forwarded by the *oifs* and never forwarded by the interface expecting to receive the said traffic, the *iif*.

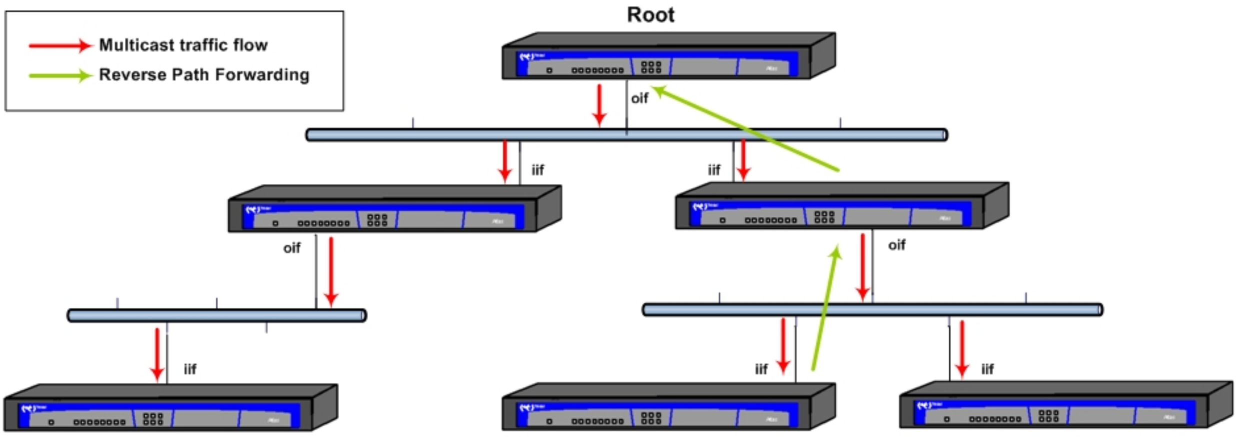

Fig. 3: Reverse Path Forwarding

So a router can determine the incoming interface, it has to use the device's unicast routing table, executing a route search towards the tree root, that could be an RP or an 'S' source. This mechanism is known as *Reverse Path Forwarding* (RPF), i.e. this searches for the opposite route to the one that broadcasts the multicast traffic. This route is checked and special attention is paid to which router is the next hop as this should be a PIM neighbor. This router is known as the *neighbour* (or *RPF neighbour*). In cases when searching for the route towards a source directly connected to the device, there won't be an *upstream neighbour*. All routers that build trees used in the PIM must support PIM in the implicated interfaces.

Other relevant data taken from the unicast route is the cost of the said route. This is used for example to resolve conflicts between various PIM routers through ASSERT messages to clarify which one of them has the lowest cost route to the RP or the 'S' source depending on the case. If static routes are configured, you should bear in mind the costs of the said routes.

1.2.5 Discovering RPs

The *Rendezvous -Point* (RP) is a PIM node responsible for coordinating all multicast traffic from a set of multicast groups.

First, all routers interested in receiving multicast traffic from a group 'G', one of the groups from a range assigned to the RP, contacts the said RP, i.e., joins the *Shared -Tree* (*, G). Secondly, routers behaving as DR encapsulate all multicast traffic generated by the local link sources and forward it to RP. Thus, the RP becomes a meeting point between transmitters and receivers for multicast traffic from group 'G'. The RP is designated by the interface’s IP where the RP role is performed.

All PIM routers in a domain must execute the same group mapping to RP since there can be more than one possible RP. But each group must be able to uniquely associate to an RP. Therefore it’s essential that all PIM routers in the domain share the same RPs information, known as RP-Set. Each of its entries are associated with an RP, defined by its IP address, with a range of multicast groups, described by a range initial group and mask.

1.2.5.1 RPs discovery mechanism

There are different methods to ensure that RP-Set is identical in all the routers. The standardized methods available in our routers are as follows:

1.2.5.1.1 Static RPs

This solution is the simplest, but also the least flexible and there is no redundancy. This consists of all the RPs that make up the RP-Set are statically identically configured in all PIM routers in the domain. To add or remove an RP from the RP-Set, you have to change the configuration in all routers. The distribution of multicast groups should be such that all multicast groups of interest are associated with a single RP; we recommend that there is no static RPs overlapping.

1.2.5.1.2 Bootstrap mechanism

This mechanism is used to execute dynamic discovery of RPs; this method is standardized in RFC 5059. This method can coexist with the static RPs method, completing the statically formed RP-Set with dynamic information. It relies on two elements: the *Bootstrap router* or BSR and *RP -candidate*. This operation is divided into two phases:

(1) Electing the BSR.
There can only be one BSR in one PIM domain, although there can be multiple candidates to be the BSR. The election is carried out by sending BOOTSTRAP messages (BSMs). Once startup is complete, a BSR candidate firstly sends its initial HELLO and subsequently announces it candidature by sending its BSM message over all its PIM interfaces. This message is not limited to the local sphere but is forwarded and broadcast throughout the PIM domain.
If a BSR candidate receives the said BSM and according to a specified algorithm it is considered to have the highest priority to become the BSR than the previous broadcaster, then it sends its own BSR message, which is also broadcast over the entire domain. The previous candidate, on receiving this message assumes that its candidature has not succeeded and takes as the newly selected BSR as the new BSR candidate. Until a better candidate emerges, the elected BSR assumes the role of BSR and sends periodic BSM messages, if a new node is added to the PIM network, the PIM neighbors indicate who the BSR is.

(2) Candidates for RP.
With the information given up until now, the BSR role has no further purpose other than the election and the broadcasting of who is the BSR. But its purpose is to gather all the information on RP candidates. All RP candidates have learned who the BSR is through the BSMs, and they periodically send their candidatures for the RP to the BSR through unicast CANDIDATE-RP-advertisement messages. The BSR adds all the candidatures received and includes them in the BSM messages that it sends. As these messages reach all PIM routers in the domain, all of them can add the said candidatures to their PR-Set, previously made up of statically configured RPs only.

In order to contain the BSM messages within the PIM domain, the role of PIM border router is defined: *PIM Border Router* or PMBR. These routers do not forward BSM messages through border interfaces, nor accept BSMs they come from outside the PIM domain.

1.2.5.2 Group mapping to RP

Starting from an RP-Set, all PIM routers must be capable of resolving RP assignation for a given multicast group.

To do this, there is a set of defined rules in the RFC 6226 that permit you to reduce the number of “RP # range of groups” assignments to consider. When you are only left with one “RP # range of groups” assignment, then the verdict on the said RP arrives indicating that this is the elected RP.

Therefore, starting from a complete RP-Set, the assignation is delimited while strictly following these rules in the following order:

(1) If the group is within the *Source Specific Multicast* (SSM) range, then you can't associate the RP and this one doesn't continue.

(2) The group must be included in the assignation group range. If it isn't in an assignation group, then it can't obtain an RP.

(3) If one of the assignations is static and it has the '*override' (rp-address <IP> override)* option configured, it is immediately elected.

(4) Assignments that have the longest group mask are selected.

(5) If there are dynamic assignments, then the static ones are dropped.

(6) If the assignments are BSR, then the RP that announces its candidature with a higher priority is selected: a lower value than "RP-priority".

(7) If the assignments are BSR and we are dealing with PIM-SM, then the RP that obtains a higher value depending on the Hash defined in RFC 4601, section 4.7.2 is selected.

(8) If there is more than one assignation still available, the RP with the highest IP address value is selected.

1.2.6 Protocol: Description

The following describes the protocol functionality once the PIM routers know their neighbors and share the RP-Set.

1.2.6.1 Building the Shared-Tree

When a receiver wants to receive traffic destined to a multicast 'G' group, regardless of its source, it requests this through an *IGMP Report* message received by the DR in its link.

If the DR did not previously have anything related to the 'G' group in their database, it creates a state (*, G) for the said group, and reports through the interface that received the *IGMP Report* that there are interested receivers. This interface then becomes part *oifs* list.

In order for the group 'G' multicast traffic to reach the receiver, the corresponding RP must be informed that it wants all the multicast traffic addressed to 'G'. This is achieved by constructing the *Shared -Tree* (*, G), where the root is the RP and this DR is one of the branches. To do this, a new PIM message is used again: JOIN/PRUNE; in this case this is dealing with a JOIN (*,G). This message is sent by the incoming interface, *iif*, with TTL 1 and addressed to the ALL-PIM-ROUTERS multicast address, although the message says that it is addressed to the router considered by the DR as its *upstream neighbour*. The said upstream is in the path towards the RP and therefore the tree is built from the branches down to the root, following *Reverse Path Forwarding (RPF)* .

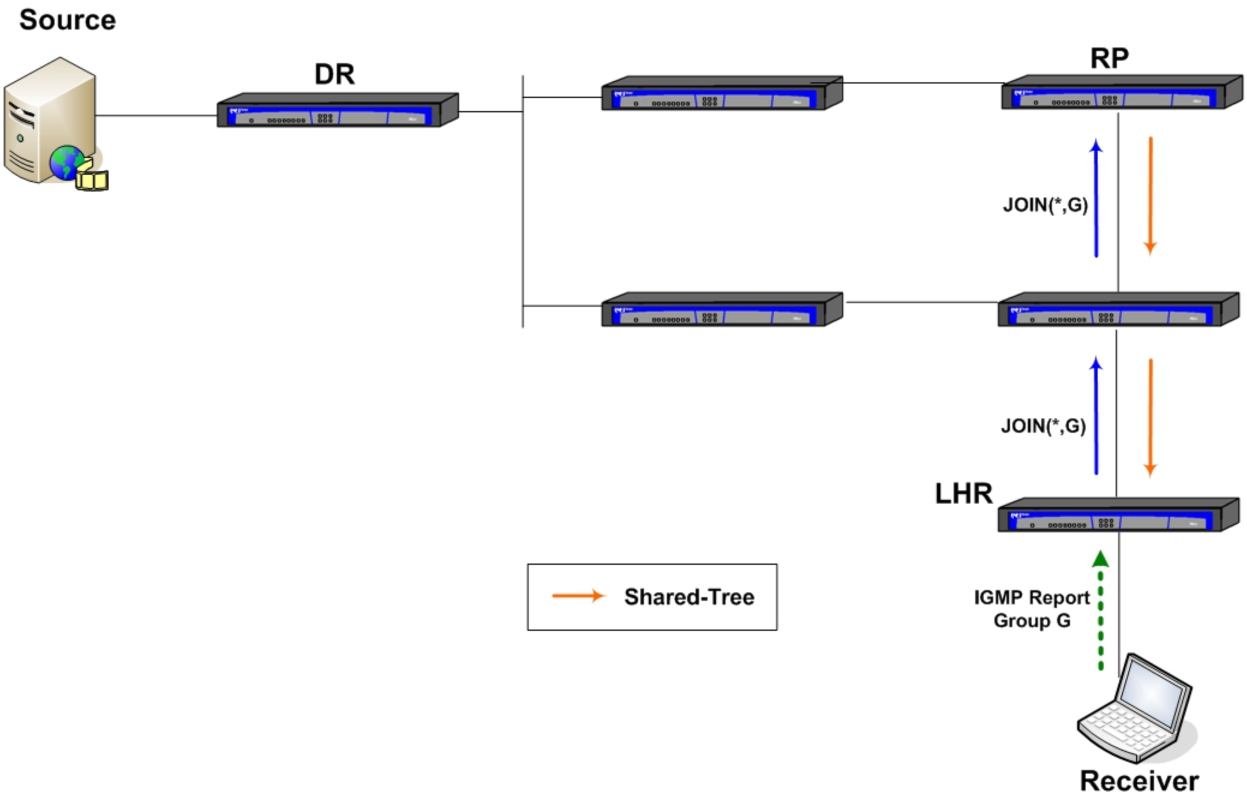

Fig. 4: Building the Shared-Tree

When the *upstream neighbour* router receives the JOIN(*,G), it creates the (*,G) state if this doesn't already exist, and reports that the interface the JOIN(*,G) arrived through has become part of the *oifs* list. The JOIN/PRUNE message has a *holdtime* so it must be periodically forwarded while the router is still interested in this particular traffic. If the *holdtime* times out and new JOIN(*,G) are not received, then this interface is deleted from the *oifs* list. The behavior of this *upstream neighbour* from here is similar to that of the DR: it sends its own JOIN(*,G) message to its own *upstream* router, towards the RP to continue building the *Shared-Tree* .

There comes a point where the JOIN(*,G) reaches the RP or reaches a router that already has the (*,G) state; in this latter case, this is already on the *Shared-Tree* so it doesn't require further building.

When the RP receives a JOIN(*,G for the first time, this also creates the (*,G) state and adds the interface the message came through to the *oifs* list.

In this case, the *iif*, the interface through which the 'G' multicast traffic arrives, isn't obtained through the unicast routing table, as this is the tree root. Instead of this, a "*virtual interface*" (*vif*) is defined, irrelevant to the rest of the router protocols, known as "*register vif*" that indicates that the multicast traffic should arrive encapsulated, as explained below:

1.2.6.2 Sending multicast traffic to the RP

One of the functions carried out by the DR is to forward all the multicast traffic from its local link to the RP. It doesn't forward it using the multicast packets, but encapsulates the multicast packet content in a new PIM message: the REGISTER message. This deals with a unicast packet that takes the DR IP address as source and whose direct destination is the RP.

When the RP receives the REGISTER message, two situations can arise:

(1) There is no *Shared-Tree*

The RP must respond to the DR with a REGISTER-STOP message indicating that it is not interested in this multicast traffic. Once a random time has elapsed and if there is still multicast traffic, the DR sends a new REGISTER message to ask the RP.

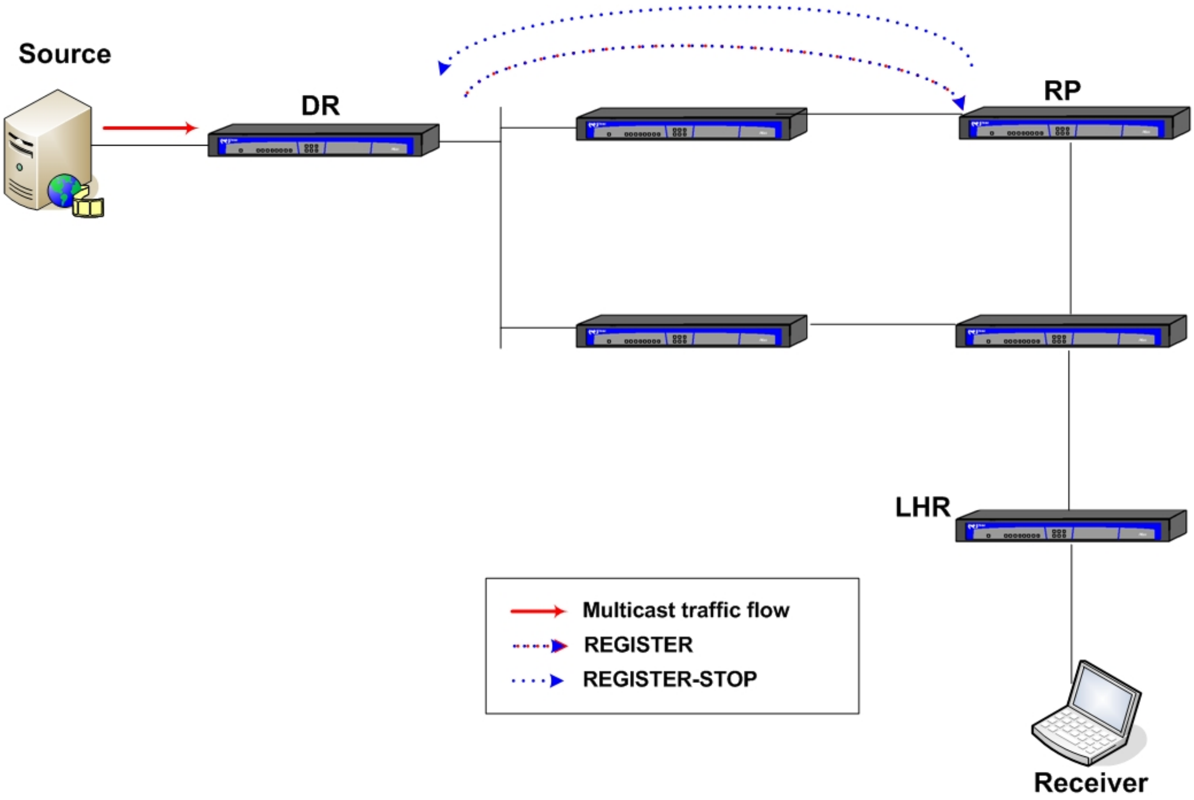

Fig. 5: Sending multicast traffic to the RP when there is no Shared-Tree

(2) When there is a *Shared-Tree*

In this case the RP decapsulates the multicast traffic and distributes it through the *oifs*. The encapsulation process is very inefficient; therefore after receiving the first REGISTER message, the RPP begins to build the *Shortest-Path-Tree* (SPT) towards the source. The aim is for the multicast traffic to arrive without being encapsulated.

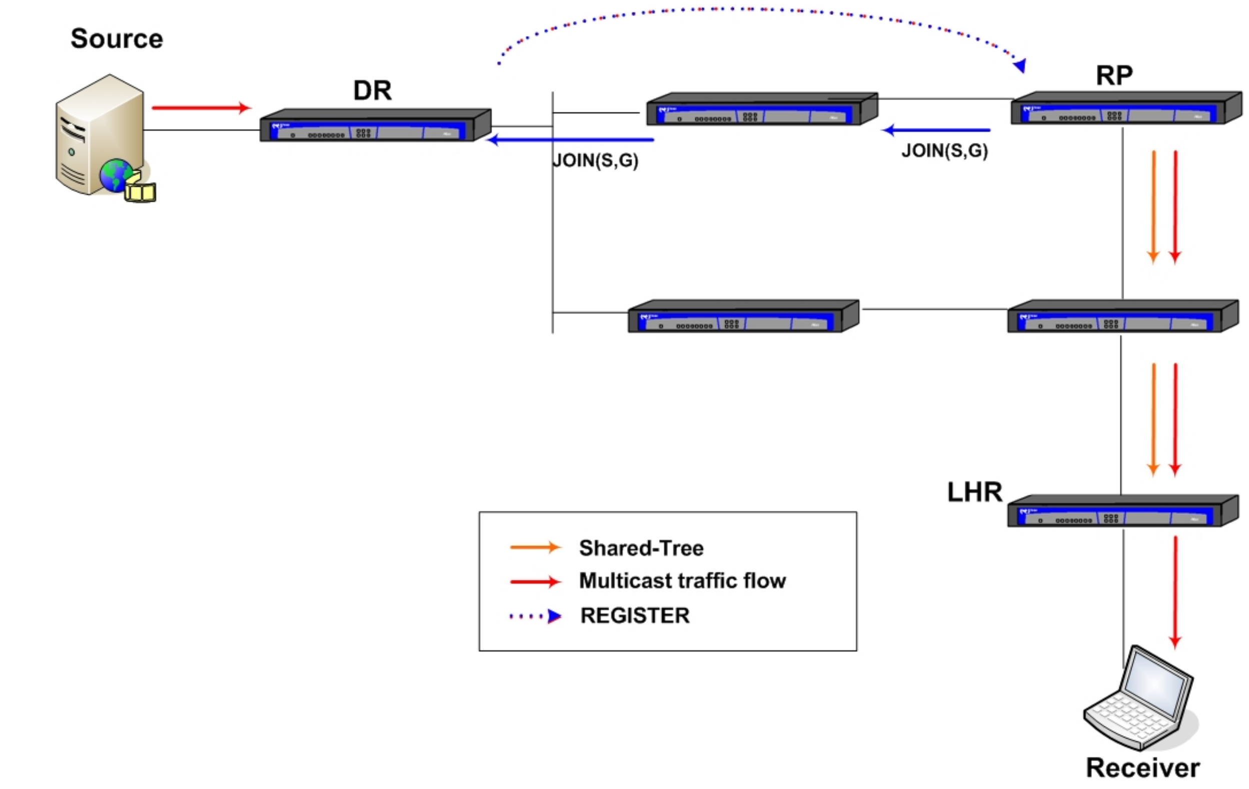

Fig. 6: Sending multicast traffic to the RP when there is a Shared-Tree

The process of building the SPT is similar to that for the Shared-Tree but using JOIN(S,G) instead of JOIN(*G). When the SPT is built, the traffic reaches the RP duplicated (encapsulated in the REGISTER message, and without encapsulation, and with the aim of stopping the REGISTER messages, the RP sends a REGISTER-STOP message to the DR.

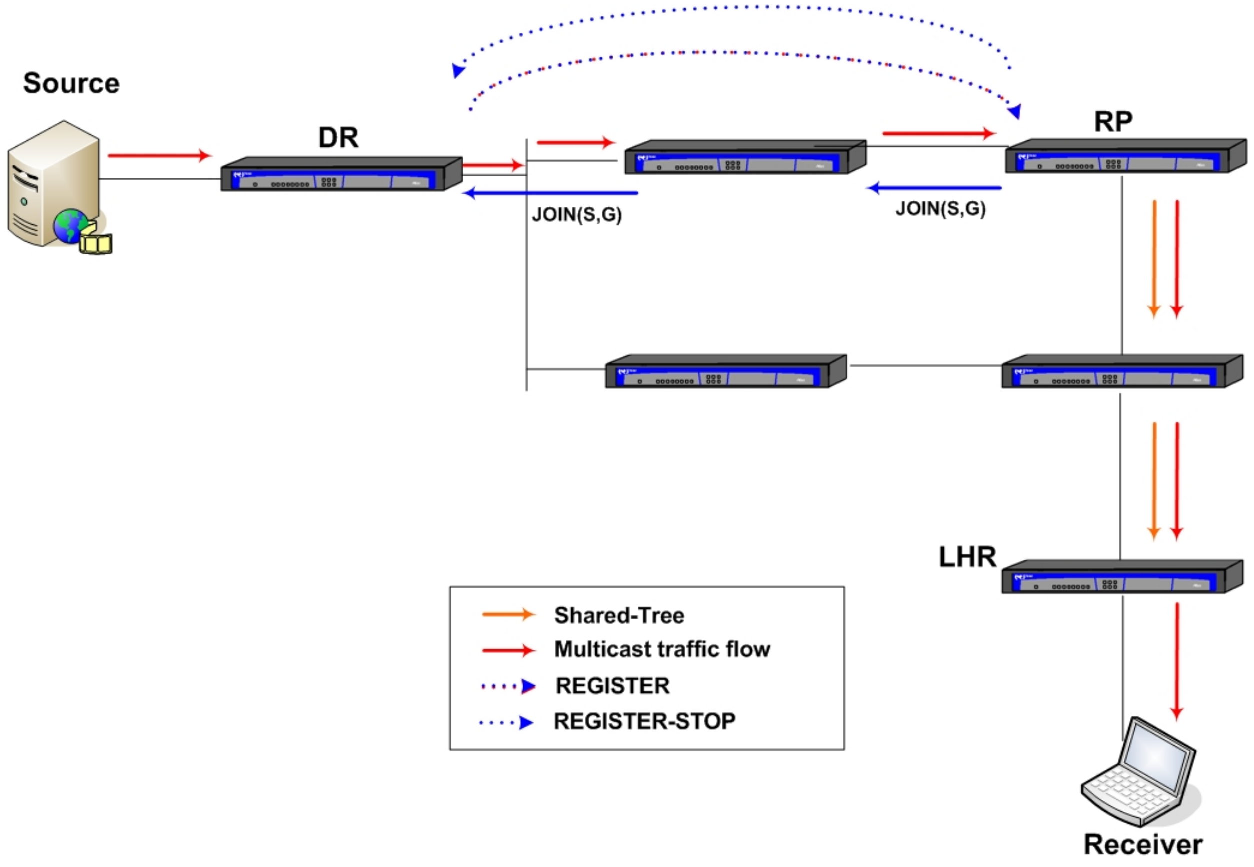

Fig. 7: Receiving duplicated multicast traffic in the RP

1.2.6.3 Building the Shortest-Path-Tree (SPT)

In this particular state, the multicast traffic reaches the receiver without undergoing encapsulation in the route, but it doesn't necessarily arrive through the shortest route.

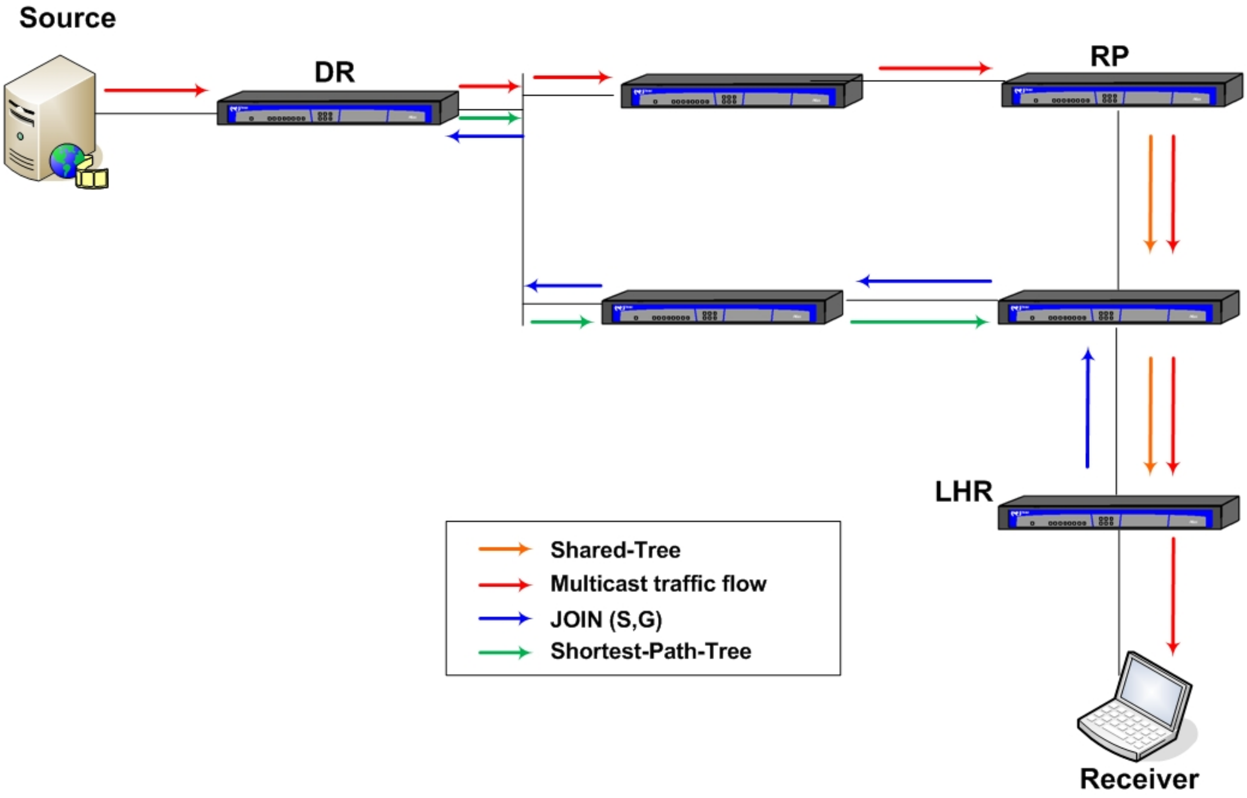

Fig. 8: Building the Shortest-Path-Tree

In this point, the *LastHop Router* (LHR) is in a situation where it has to decide whether it wants the traffic to arrive through the shortest path, attaching itself to the SPT .

There are various possible options:

- Switch to SPT after receiving the first multicast traffic packet.
- Build the SPT if the traffic is greater than a configurable bandwidth (*spt-threshold*).
- Never switch to the SPT, maintaining the minimum information on the protocol.

1.2.6.4 Excluding a source in the Shared-Tree

Once the SPT has been established, the traffic begins to arrive through the shortest path.

If a tree node detects that traffic is arriving through a different path from the *Shared-Tree* , in order to avoid this redundancy, this indicates to its *upstream neighbour* in the *Shared-Tree* that it does not want to receive traffic from this source. This is achieved through a PRUNE (S,G,rpt) message.

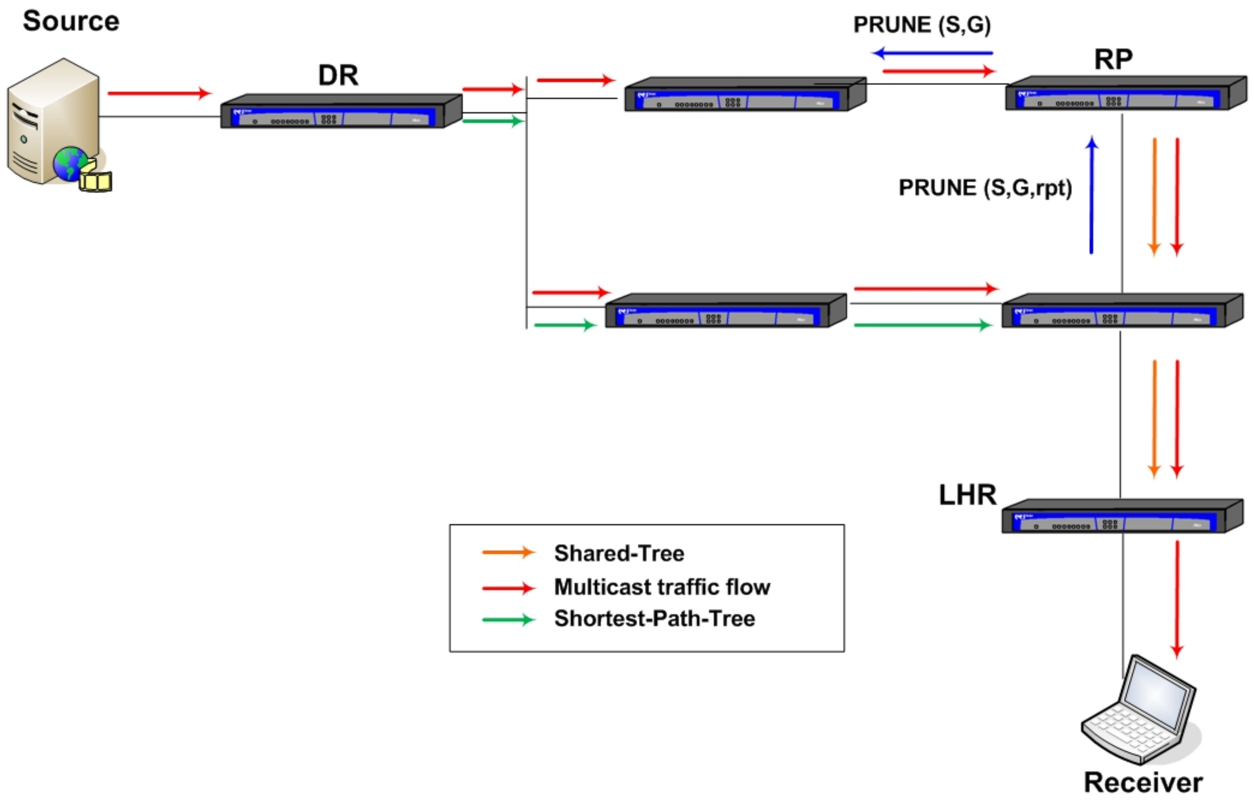

Fig. 9: Excluding a source in the Shared-Tree

As shown in Figure 9, if no other router in the Shared-Tree is interested in traffic from this source, the PRUNE (S, G, rpt) message reaches the RP, which from that point does not want traffic from this source. Consequently, SPT eliminates its membership by sending a PRUNE (S,G) to the source.

The multicast traffic generated by the source is now only sent through the shortest path to the receivers without having to go through the RP Figure 10.

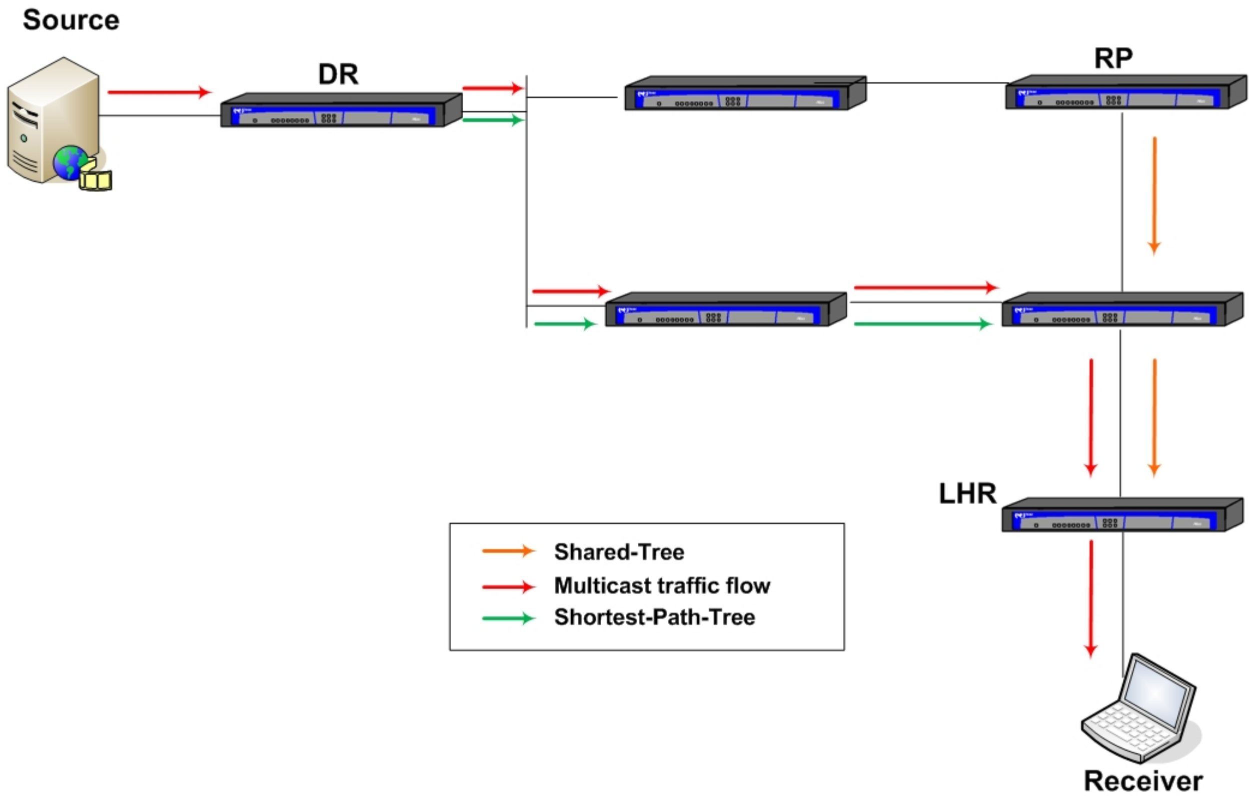

Fig. 10: Traffic being sent through the Shortest-Path-Tree

1.2.7 Routing Multicast

The complete database managed by the PIM in our routers is known as *Multicast Routing Table* (MRT), which is the equivalent to the TIB set out in RFC 4601.

This consults the MRT for each multicast packet that it must forward. This is not very efficient therefore our routers have a *Multicast Forwarding Cache* (MFC). This cache maintains the minimum information necessary for *forwarding*: incoming and outgoing interfaces and traffic statistics and is coherent with the information maintained in the MRT. If on making another consult to the MFC the entry being searched for is not found, this tries to create the entry from the information in the MRT. The cache entries are maintained while there is traffic and after an idle period are deleted.

1.3 PIM Source Specific Multicast (PIM-SSM)

The PIM-SSM variant can be considered a subset of PIM-SM and can co-exist in the same router. PIM-SSM imposes a series of additional conditions to the PIM operations in a range of multicast IP addresses. IANA has reserved the 232.0.0.0/8 address range for use in SSM, although from CIT version 11.00.04 Teldat routers also allow you to set any multicast address range for SSM, given an extended access list (see section 2.10). The main characteristic is that the receivers a priori must know the source they want to receive the traffic from, always maintaining even numbers (S,G). So a receiver can announce it wants to receive traffic from a specific source, it must use IGMP version 3, as previous versions do not support source information. When a DR receives an IGMPv3 petition from a receiver, it immediately builds the SPT. This functioning mode is not exclusive to the SSM range, the difference being that in the said range it cannot operate in a different way. This means that they can never create Shared-Trees, that an SSM group can never have an RP associated, and IGMP petitions that do not include specific sources are rejected, etc. So the router operates correctly, the IGMP protocol must also be adapted for SSM.

Chapter 2 Configuration

2.1 Accessing the Configuration menu

To access the PIM protocol configuration menu, enter the following commands from the general prompt:

Syntax:

```
*config

Config>protocol pim

-- PIM protocol user configuration --
PIM Config>
```

The PIM functions independently through VRF. To configure a VRF other than the primary, specify this with the **vrf** command followed by the VRF name.

Syntax:

```
PIM Config>vrf <vrf name>

-- PIM protocol user configuration for a VRF --
PIM vrf Config>
```

2.2 PIM Protocol Configuration Commands

The following options appear at the PIM protocol configuration prompt:

```
PIM Config>?
  accept-register    Registers accept filter
  accept-rp          RP accept filter
  bsr-candidate      Candidate bootstrap router
  enable             Enable PIM Protocol
  no                 Negate a command or set its defaults
  register-source    Source address for PIM Register
  rp-address         PIM RP-address (Rendezvous Point)
  rp-candidate       To be a PIM RP candidate
  sparse             Adjust the (S,G) expiry timer interval
  spt-threshold      Source-tree switching threshold
  ssm                Configure Source Specific Multicast
  vrf                Specify parameters for a VPN Routing/Forwarding instance
  exit
PIM Config>
```

Command	Function
ACCEPT-REGISTER	Configures an RP router to filter REGISTER messages.
ACCEPT-RP	Configures the router to accept JOIN/PRUNE messages destined for an RP and a specific list of groups.
BSR-CANDIDATE	Configures the router to announce its candidacy as PIM Bootstrap Router (BSR).
ENABLE	Enables PIM in a device.
NO	Allows you to configure default values or to negate a command.
REGISTER-SOURCE	Configures the IP source address for the REGISTER messages, the IP address for the outgoing interface towards the RP being different.
RP-ADDRESS	Statically configures the RP IP address.
RP-CANDIDATE	Configures the router to advertise as a RP candidate to the BSR.
SPARSE SG-EXPIRY-TIMER	Configures the expiry time for the MRT entries (S, G).
SPT-THRESHOLD	Configures when a PIM router should join the SPT for a specific group.
SSM	Lets you assign IP multicast address ranges to SSM, specifying a previously created extended access list.
VRF	Configures PIM in a *routing/forwarding* domain in virtual private networks (*VPN*).

EXIT	Exits the PIM protocol configuration.

2.2.1 ACCEPT-REGISTER

The **accept-register** command is used to prevent unauthorized sources from sending REGISTER messages to the RP. If an unauthorized source sends a REGISTER message to the RP, then the RP immediately sends a REGISTER-STOP in response.

Syntax:

```
PIM Config>accept-register list <access-list-number>
```

access-list-number:	Extended access list number (100-1999) that allows REGISTER messages to be filtered.

Example:

```
PIM Config>accept-register list 110
PIM Config>
```

The selected access list must have been previously configured; if not the following error appears:

```
PIM Config>accept-register list 200
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

This command is unique. If you configure a new one, it will substitute the existing one.

To delete the “accept-register” command, use the “no” form of this command.

```
PIM Config>no accept-register
PIM Config>
```

2.2.2 ACCEPT-RP

The **accept-rp** command allows the router to accept JOIN(*,G) messages destined for a specific RP address only. Additionally, the multicast group address must be within the range specified by the access list.

Syntax:

```
PIM Config>accept-rp <rp-address> [<access-list-number>]
```

rp-address:	RP IP address.
access-list-number:	(Optional) Generic access list number (1-99), that allows you to specify the range of multicast groups that this command will affect.

Example:

```
PIM Config>accept-rp 2.2.2.1 10
PIM Config>
```

The same RP address can only be configured once in the same VRF for the **accept-rp** command. If you configure the same address again with different options, then the last one configured takes precedence.

If you do not specify an access list number, it’s considered that the **accept-rp** command affects the whole range of multicast groups: 224.0.0.0/4.

The selected access list must already exist, otherwise the following error message appears:

```
PIM Config>accept-rp 2.2.2.1 20
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

When the RP address is a local address of the router, the router is the RP only for the specified group range specified by the access list. When the multicast group address is not in the defined group range, the RP will not accept JOIN or REGISTER messages and will respond immediately to REGISTER messages with REGISTER-stop messages.

To delete an “accept-rp” command, use the “no” form of this command, specifying the specific entry you want to delete:

```
PIM Config>no accept-rp 2.2.2.1
PIM Config>
```

2.2.3 BSR-CANDIDATE

The **bsr-candidate** command allows you to configure the router as a candidate for a PIM Bootstrap Router (BSR).

The router sends BOOTSTRAP messages to all PIM neighbors, with the address of the interface designated as BSR.

Syntax:

```
PIM Config>bsr-candidate <interface> [<hash-mask-length> [<priority>]]
```

interface:	Name of the interface announcing its IP address as candidate to the BSR address.
hash-mask-length:	(Optional) Length of a mask (0-32), going to be used in the RP election for a specific group, using the RP-Set sent by the BSR.
priority:	(Optional) Priority of the candidate BSR (0-255). The default priority is 64. The BSR candidate with the highest priority value is chosen.

Example:

```
PIM Config>bsr-candidate ethernet0/0
PIM Config>
```

The selected interface must have PIM enabled; otherwise the following error message appears:

```
PIM Config>bsr-candidate ethernet0/0.1
CLI Error: PIM not configured on that interface
CLI Error: Command error
PIM Config>
```

This command is unique. If you configure a new one, it will substitute the existing one.

To delete a “bsr-candidate” command, use the “no” form of this command:

```
PIM Config>no bsr-candidate
PIM Config>
```

2.2.4 ENABLE

This command allows you to globally enable the PIM protocol in the device.

Syntax:

```
PIM Config>enable
```

Example:

```
PIM Config>enable
PIM Config>
```

2.2.5 REGISTER-SOURCE

The **register-source** command is required only when the IP source address of a REGISTER message is not a uniquely routed address to which the RP can send packets. This situation may occur if the source address is filtered as such that packets sent to it will not be forwarded or if the source address is not unique to the network. In these cases, the replies sent from the RP to the source address will fail to reach the DR, resulting in PIM protocol failures.

If the **register-source** command is not configured or the configured source name is not in service, the IP address of the outgoing interface of the DR leading toward the RP is used as the IP source address of the REGISTER message.

Syntax:

```
PIM Config>register-source <interface>
```

interface: Name of the interface whose IP address is used as source address for the REGISTER messages.

Example:

```
PIM Config>register-source ethernet0/0
PIM Config>
```

The selected interface must have PIM enabled; otherwise the following error message appears:

```
PIM Config>register-source ethernet0/0.1
CLI Error: PIM not configured on that interface
CLI Error: Command error
PIM Config>
```

This command is unique. If you configure a new one, it will substitute the existing one.

To delete a “register-source” command, use the “no” form of this command:

```
PIM Config>no register-source
PIM Config>
```

2.2.6 RP-ADDRESS

The **rp-address** command allows you to specify the IP addresses for the static RPs.

PIM uses RPs to connect sources and receivers. All routers in a PIM domain need to have a consistent configuration for the multicast groups.

Syntax:

```
PIM Config>rp-address <ip-address> [<access-list-number>][override]
```

ip-address:	IP address of the static RP, which is used for the indicated groups.
access-list-number:	(Optional) Generic access list number (1-99), that defines the multicast groups that are associated with the previous RP.
override:	(Optional) This option indicates that there is a dynamic and static RP-group mapping and there is a conflict with the RP address, the RP address configured as static takes precedence over the dynamic.

Example:

```
PIM Config>rp-address 2.2.2.1
PIM Config>
```

The selected access list must have been previously configured; otherwise the following appears:

```
PIM Config>rp-address 2.2.2.1 10
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

You can configure a single RP for more than one group using an access list. If no access list is specified, the configured address is the RP for the entire range of multicast groups: 224.0.0.0/4.

You can configure multiple RPs, but only one RP per multicast group range.

If you configure more than one static RP, the following rules apply:

- The highest RP IP address is selected:

  - If a multicast group is included in various configured static RP access lists.

- One RP address per command:

  - Each range of multicast groups can only have one RP assigned.

- One access list per command:

  - Each access list is unique for each RP, i.e. this access list cannot be used with a different RP. In cases where this occurs, the access list will then be assigned to the second RP.

To delete an ”rp-address” command use the “no” form of this command, specifying the specific entry you want to delete:

```
PIM Config>no rp-address 2.2.2.1
PIM Config>
```

2.2.7 RP-CANDIDATE

The **rp -candidate** command allows the router to send CANDIDATE-RP-ADVERTISEMENT messages to the device to announce itself as a candidate for RP. This must configure itself in routers that have good connectivity to all parts of the PIM domain.

Syntax:

```
PIM Config>rp-candidate <interface> [group-list <access-list-number>]
                                    [interval <seconds>][priority <value>]
```

interface:	Name of the interface that announces its IP address as candidate for RP.
access-list-number:	(Optional) Generic access list number (1-99), that defines the multicast groups that are advertised with the address of the previous RP.
seconds:	(Optional) Specifies in seconds the time interval used to send the CANDIDATE-RP-ADVERTISEMENT messages. Default is 60 seconds.
value:	(Optional) Specifies the priority (0-255) that the CANDIDATE-RP-ADVERTISEMENT messages are sent with. The priority default is 192. The candidate with the lowest priority is elected.

Example:

```
PIM Config>rp-candidate ethernet0/0 group-list 10
PIM Config>
```

The IP address associated with the interface specified in the command **rp-candidate** is the address that is sent as the candidate for RP. You can configure as many candidates for RP as interfaces that the PIM has enabled.

The selected interface must have PIM enabled; otherwise the following error message appears:

```
PIM Config>rp-candidate ethernet0/1
CLI Error: PIM not configured on that interface
CLI Error: Command error
PIM Config>
```

The selected access list must have been previously configured; otherwise the following error message appears:

```
PIM Config>rp-candidate ethernet0/0 group-list 20
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

To delete an “rp-candidate” command, use the “no” form of this command, specifying the specific entry you want to delete:

```
PIM Config>no rp-candidate ethernet0/0
PIM Config>
```

2.2.8 SPARSE SG-EXPIRY-TIMER

The **sparse sg-expiry-timer** command allows you to adjust the timer interval for the PIM MRT (S,G) entries. This command can be used to protect the STP from intermittent sources in PIM-SM environments.

When a source stops sending traffic to a multicast group, the corresponding entry (S,G) eventually times out and the MRT is removed. When the source resumes sending traffic to the group, the (S,G) entry is rebuilt. During the short time before the (S, G) entry is rebuilt, the traffic is forwarded on the (*, G) forwarding entry. There is a small window of time before the (S, G) entry is completely built in which packets may be dropped. This command can be used to maintain the (S;G) entries, so they will not be removed and will not potentially suffer packet loss.

Syntax:

```
PIM Config>sparse sg-expiry-timer <seconds> [sg-list <access-list-number>]
```

seconds:	Duration of the expiry timer for the (S,G) entries in seconds (180-57600). The default value for the expiry timer is 210 seconds.
access-list-number:	(Optional). Number from the extended access list (100-1999), that indicates that the MRT (S,G) entries must be applied to the previous interval. If no list is specified, the time value configured is applied to all MRT (S,G) entries.

Example:

```
PIM Config>sparse sg-expiry-timer 200
PIM Config>
```

The selected access list must have been previously configured; otherwise the following error message appears:

```
PIM Config>sparse sg-expiry-timer 200 sg-list 210
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

This command is unique. If you configure a new one, it will substitute the existing one.

To delete a “sparse sg-expiry-timer” command, use the “no” form of this command:

```
PIM Config>no sparse
PIM Config>
```

2.2.9 SPT-THRESHOLD

The **spt-threshold** command allows you to configure the traffic threshold so the PIM router must join the SPT for a specific group.

When this command is not used, the *LastHop Router* in the domain immediately joins the SPT after receiving the first multicast traffic packet.

Syntax:

```
PIM Config>spt-threshold {<traffic-rate-kbps> | infinity} [group-list <acl-number>]
```

traffic-rate-kbps:	Traffic threshold in kbps (0-4294967).
infinity:	Makes all the sources from a specific group only use the *Shared-Tree.*
group-list:	(Optional) Specifies through a generic access list (1-99), the multicast groups that the configured traffic threshold is applied to. If a list has not been specified, the threshold is applied to all multicast groups.

Example:

```
PIM Config>spt-threshold 100
PIM Config>
```

The selected access list must have been previously configured; otherwise the following error message appears:

```
PIM Config>spt-threshold 100 group-list 20
CLI Error: Unable to assign empty list
CLI Error: Command error
PIM Config>
```

If a source sends traffic with a speed equal or greater than the configured threshold, the *LastHop Router* sends a JOIN message towards the source to build the SPT.

This command has different configuration combinations, with the *traffic-rate-kbps* and *infinity* options.

- The *traffic-rate-kbps* and *infinity* options are exclusive, i.e., you cannot have both configured at the same time:

```
PIM Config>spt-threshold infinity
PIM Config>show config
;
        spt-threshold infinity
PIM Config>
PIM Config>spt-threshold 10
PIM Config>show config
;
        spt-threshold 10
```

- The *traffic-rate-kbps* and *infinity* options can both be configured at the same time with and without an access list:

```
PIM Config>spt-threshold infinity
PIM Config>spt-threshold infinity group-list 10
PIM Config>show config
;
        spt-threshold infinity
        spt-threshold infinity group-list 10
```

- Additionally you can configure the *traffic-rate-kbps* option together with the previous configuration provided that it is accompanied by the *group-list* option:

```
PIM Config>spt-threshold 100 group-list 15
PIM Config>show config
;
        spt-threshold infinity
        spt-threshold 100 group-list 15
        spt-threshold infinity group-list 10
```

To delete a “stp-threshold” command, use the “no” form of this command, specifying the specific entry you want to delete:

```
PIM Config>no spt-threshold infinity
PIM Config>no spt-threshold 100 group-list 15
PIM Config>no spt-threshold infinity group-list 10
```

2.2.10 SSM

The **ssm** command allows you to configure one or more multicast IP address ranges for clients who wish to join an SSM group. If the command is not in use, only the default reserved range may be used (232.0.0.0/8).

Syntax:

```
PIM Config>ssm range <acl-number>
```

Range: Specifies multicast ranges for SSM using an extended access list (100-1999).

Example:

```
PIM Config>ssm range 105
PIM Config>
```

The access list selected must exist, otherwise the following error message is displayed:

```
PIM Config>ssm range 103
CLI Error: Unable to assign empty list
```

```
CLI Error: Command error
PIM Config>
```

The list number specified must also be within the extended range, otherwise the following error will be displayed:

```
PIM Config>ssm range 95
CLI Error: Unrecognized command or invalid value
PIM Config>
```

If you want to use the default range (232.0.0.0/8), disable the command as follows:

```
PIM Config>no ssm
PIM Config>
```

Command history:

Release	Modification
11.00.04	Command introduced in version 11.00.04
11.00.03.01.02	Command introduced in version 11.00.03.01.02
11.01.00	Command introduced in version 11.01.00

2.2.11 VRF

The **vrf** command accesses the PIM configuration for the specified VRF. By accessing the PIM configuration menu for a VRF, the prompt changes to “*PIM vrf Config>*”. On using the *exit* command, this returns to the PIM configuration for the main VRF.

Syntax:

```
PIM Config>vrf <vrf_name>
```

Example:

```
PIM Config>vrf vrf_1
```

```
-- PIM protocol user configuration for a VRF --
PIM vrf Config>
```

This VRF menu includes the same configuration commands as in the main VRF.

2.2.12 EXIT

The **exit** command is used to exit the protocol configuration menu and to return to the device configuration prompt.

Syntax:

```
PIM Config>exit
```

Example:

```
PIM Config>exit
Config>
```

2.3 PIM Configuration Commands per Interface

Lastly the PIM configuration is explained. This must be carried out through the interface and remain within the set of the interface IP parameters.

The first step is to access the configuration menu for the interface itself. Given that each interface is associated with a VRF, you don't need to specify it. The following example uses the ethernet0/0 interface.

Example:

```
Config>network ethernet0/0

-- Ethernet Interface User Configuration --
ethernet0/0 config>ip pim ?
  bsr-border          Border of PIM domain
  dr-priority         PIM router DR priority
  neighbour-filter    PIM peering filter
  query-interval      PIM router query interval
  sparse-mode         Enable PIM sparse-mode operation
```

The PIM configuration commands per interface are always preceded by **ip pim**. As we have seen in the help, the following options are available in order to configure the PIM.

Command	**Function**
BSR-BORDER	Prevents the sending and reception of BOOTSTRAP messages in the interface.
DR-PRIORITY	Allows you to configure a different priority value to that configured by default for electing the DR.
NEIGHBOUR-FILTER	Prevents the reception of PIM messages from unwanted routers.
NO	Allows you to configure default values or negate a command.
QUERY-INTERVAL	Allows you to configure the frequency the HELLO messages are sent.
SPARSE-MODE	Allows you to enable the PIM *Sparse Mode* in the interface.

2.3.1 BSR-BORDER

The **bsr-border** command allows you to configure a border router for a PIM domain. Configuring an interface as **bsr-border** prevents BOOTSTRAP messages being exchanged between two domains.

BOOTSTRAP messages should not be exchanged between different domains, because routers in one domain may elect an RP in the other domain, resulting in protocol malfunction or loss of isolation between the two domains.

Syntax:

```
ethernet0/0 config>ip pim bsr-border
```

To delete a "bsr-border" command, use the "no" form of this command:

```
ethernet0/0 config>no ip pim bsr-border
ethernet0/0 config>
```

2.3.2 DR-PRIORITY

The default priority value for a router for DR election is 1. The **dr-priority** command allows you to modify this priority value in the corresponding interface.

Syntax:

```
ethernet0/0 config>ip pim dr-priority <priority-value>
```

Example:

```
ethernet0/0 config>ip pim dr-priority 100
```

priority-value:	The range of priority values is from 0 to 4294967295.

For further information on DR election, please see section 2.3 “Electing the DR” in the Chapter 1 “Introduction” in this document.

To delete a “dr-priority” command, use the “no” form of this command:

```
ethernet0/0 config>no ip pim dr-priority
ethernet0/0 config>
```

2.3.3 NEIGHBOUR-FILTER

The **neighbour-filter** command allows you to reject PIM packets from a specific source. This is used when you want to prevent a router from participating in the PIM domain.

Syntax:

```
ethernet0/0 config>ip pim neighbour-filter <access-list-number>
```

Example:

```
ethernet0/0 config>ip pim neighbour-filter 10
```

access-list-number:	Indicates the number on the generic access list (1-99), that bars PIM packets from a specific source.

The selected access list must have been previously configured; otherwise the following error message appears:

```
ethernet0/0 config>ip pim neighbour-filter 20
CLI Error: Unable to assign empty list
CLI Error: Command error
ethernet0/0 config>
```

To delete a “neighbour-filter” command, use the “no” form of this command:

```
ethernet0/0 config>no ip pim neighbour-filter
ethernet0/0 config>
```

2.3.4 QUERY-INTERVAL

The default time interval between HELLO messages is 30 seconds. The **query-interval** command allows you to modify this value so that messages are sent with more or less frequency.

Syntax:

```
ethernet0/0 config>ip pim query-interval <period>
```

Example:

```
ethernet0/0 config>ip pim query-interval 45
```

period:	Number of seconds that can be configured as a time interval between HELLO messages. This goes from 1 to 65535

This value is also used to determine the time that a PIM router remains as a neighbor. A PIM router is maintained as a neighbor three times the time interval between HELLO messages. If during this time HELLO messages are not received, then it is deleted

To delete a “query-interval” command, use the “no” form of this command:

```
ethernet0/0 config>no ip pim query-interval
ethernet0/0 config>
```

2.3.5 SPARSE-MODE

The **sparse-mode** command is used to enable the PIM-SM mode in the interface.

Syntax:

```
ethernet0/0 config>ip pim sparse-mode
```

To delete a “sparse-mode” command, use the “no” form of this command:

```
ethernet0/0 config>no ip pim sparse-mode
ethernet0/0 config>
```

Chapter 3 Monitoring

3.1 Accessing the Monitoring Menu

PIM protocol monitoring displays the information relative to the protocol functionality.

To access the PIM protocol monitoring menu, enter the following commands from the general prompt:

```
*monitor

Console Operator
+protocol pim
-- PIM protocol monitor --
PIM+
```

The PIM protocol operates independently through VRF. If you want to access a VRF that is not the main one where PIM is active, you can specify this using the **vrf** command followed by the name of the VRF. If PIM is not enabled on the VRF, you can't access the VRF PIM monitoring menu even if the said VRF exists.

```
PIM+vrf <vrf name>

-- PIM protocol monitor for a VRF --

PIM vrf+
```

3.2 Monitoring Commands

The following options appear at the PIM protocol monitoring prompt:

```
PIM+?
  clear    Clear information
  list     List PIM information
  stats    Show statistics
  vrf      PIM monitoring in a VPN Routing/Forwarding instance
  exit
PIM+
```

Command	Function
CLEAR	Allows you to clear the sent and received message statistics or initialize the cache.
LIST	Lists different information about the current state of the PIM protocol.
STATS	Presents the sent and received message statistics.
VRF	Selects another VRF instance where the PIM protocol performance is monitored.
EXIT	Exits the PIM protocol monitoring.

3.2.1 CLEAR

The **clear** command allows you to initialize various options.

Syntax:

```
PIM+clear ?
  cache    Flush Multicast Forwarding Cache
  stats    Clear PIM statistics
PIM+clear
```

3.2.1.1 CLEAR CACHE

Obsoletes all the *Multicast Forwarding Cache* (MFC) entries in the device; therefore each first multicast traffic flow packet reaching the device will produce a new query in the *Multicast Routing Table* (MRT).

Syntax:

```
PIM+clear cache
```

Example:

```
PIM+list cache

 Multicast Forwarding Cache
-------------------------------------------------------------------------------
Ch#  Life  Source            Group            Interfaces
---  ----  ---------------   ---------------  --------------------------------
 19   10   192.168.212.18  -> 238.1.1.1        [fr1]

Number of MFC entries listed: 1
PIM+clear cache
PIM+list cache
Multicast Forwarding Cache
-------------------------------------------------------------------------------
Ch#  Life  Source            Group            Interfaces
---  ----  ---------------   ---------------  --------------------------------
 19    4D  192.168.212.18  -> 238.1.1.1        [fr1]
Number of MFC entries listed: 1
PIM+
```

The cache entry has not as yet been deleted, however it has been marked as obsolete or it's in the process of being deleted (letter 'D' after the seconds counter in the lifetime).

3.2.1.2 CLEAR STATS

Resets the counters for messages sent and received by the PIM protocol.

Syntax:

```
PIM+clear stats
```

Example:

```
PIM+stats global

Global packet statistics
========================
                          sent /   received
     Hello                 602 /        388
     Register               75 /          0
     Null-register         154 /          0
     Register-stop           0 /        200
     Join-prune              0 /         11
     Bootstrap             189 /        191
     Assert                  0 /          0
     Cand-RP-adv            92 /          0

PIM+clear stats

PIM+stats global

Global packet statistics
========================
                          sent /   received
     Hello                   0 /          0
     Register                0 /          0
     Null-register           0 /          0
     Register-stop           0 /          0
     Join-prune              0 /          0
     Bootstrap               0 /          0
     Assert                  0 /          0
     Cand-RP-adv             0 /          0

PIM+
```

3.2.2 LIST

The **list** command displays the information gathered by the PIM protocol. The majority of this refers to the variables used in the *Multicast Routing Table* (MRT), but also provides different points of view depending on what the user is interested in.

Syntax:

```
PIM+list ?
  cache         Multicast Forwarding Cache
  groups        Multicast groups list used in PIM
  interfaces    Interface state for PIM
  mrt           Multicast Routing Table
  rp-set        Current RP Set
  sources       Sources list used in PIM
  timers        Timers list used in PIM
PIM+list
```

3.2.2.1 LIST CACHE

Allows you to dump the MFC information.

Syntax:

```
PIM+list cache?
  detailed    Show also iif and stats (pkt/bytes/wrong-iif)
  <cr>
```

3.2.2.1.1 LIST CACHE

Displays the summarized information on the MFC entries

Syntax:

```
PIM+list cache
```

Example:

```
PIM+list cache

 Multicast Forwarding Cache
-------------------------------------------------------------------------------
 Ch# Life Source Group Interfaces
---  ----   ---------------    ---------------  --------------------------------
240   10   192.168.212.10  -> 239.255.255.250  [register]
Number of MFC entries listed: 1
PIM+
```

Each entry indicates the following values:

- **Ch#**: The position number in the cache table.

- **Life**: The remaining life time in seconds for the entry if there is no activity.

- **Source**: Source IP for the multicast traffic affected by the entry.

- **Group**: Destination IP for the multicast traffic affected by the entry.

- **Interfaces**: Outgoing interfaces where the traffic exits. In this example, the register interface indicates that the traffic in REGISTER messages must be sent encapsulated.

3.2.2.1.2 LIST CACHE DETAILED

Displays detailed information on the MFC entries.

Syntax:

```
PIM+list cache detailed
```

Example:

```
PIM+list cache detailed

 Multicast Forwarding Cache
-------------------------------------------------------------------------------
Ch#  Life  Source             Group            Interfaces
---  ----  ---------------    ---------------  --------------------------------
 19    9   192.168.212.18  -> 238.1.1.1        [fr1]
            Inc: ethernet0/0       Stats: (        92p/     94576B/         0W)
Number of MFC entries listed: 1
PIM+
```

In addition to the same information on the summarized version, each entry has a second line added where the following is also listed:

- **Inc**: Incoming interface through which the traffic is received.

- **p**: Number of multicast packets that have been forwarded after consulting the MFC.

- **B**: Bytes counter for the said packets.

- **W**: Number of times that a multicast packet has been received through an interface that is not the incoming one. This is indicated with a ‘W’ that stands for *wrong interface*.

3.2.2.2 LIST GROUPS

Displays the list of multicast groups currently present in the MRT.

Syntax:

```
PIM+list groups?
  one-group    Show only one group if present
  one-rp       Show only groups associated to one RP
  <cr>         Show complete groups list
```

3.2.2.2.1 LIST GROUPS

Displays the full list of the groups without any kind of filtering.

Syntax:

```
PIM+list groups
```

Example:

```
PIM+list groups
 Groups list
------------------------------------------------------------------------
Group RP address RP next grp (*,G) (S,G,rpt) (S,G)

238.1.1.1        2.2.2.1          239.255.255.250    N        0      1
239.255.255.250  2.2.2.1          ---                N        0      5
Number of groups listed: 2
PIM+
```

The groups are shown from smallest to largest group IP. The following is cited from each one:

- **Group**: Multicast group IP address.

- **RP address**: RP IP address used to map the group.

- **RP next grp**: Next group mapped with the same RP. The first group on the RP list is the most recently created.

- **(*,G)**: Existence of state (*,G); this can be Yes (**Y**) or No (**N**).

- **(S,G,rpt)**: Number of states (S,G,rpt) with the said ‘G’ group.

- **(S,G)**: Number of states (S,G) with the said ‘G’ group.

3.2.2.2.2 LIST GROUPS ONE-GROUP <GROUP>

Only lists the group specified in the '*group'* option.

Syntax:

```
PIM+list groups one-group <group>
```

Example:

```
PIM+list groups one-group 238.1.1.1

Group            RP address       RP next grp      (*,G)  (S,G,rpt)  (S,G)
238.1.1.1        2.2.2.1          239.255.255.250    N        0      1
PIM+
```

The information shown on the group is the same as that shown in the full list of groups.

3.2.2.2.3 LIST GROUPS ONE-RP <RP_ADDRESS>

Lists the groups that have an '*rp_address*' as RP.

Syntax:

```
PIM+list groups one-rp <rp_address>
```

Example:

```
PIM+list groups one-rp 2.2.2.1

Group            RP address       RP next grp      (*,G)  (S,G,rpt)  (S,G)
238.1.1.1        2.2.2.1          239.255.255.250    N        0      1
239.255.255.250  2.2.2.1          ---                N        0      2
PIM+
```

3.2.2.3 LIST INTERFACES

Presents information on the interfaces associated with the VRF in question.

Syntax:

```
PIM+list interfaces ?
  detailed    Show also neighbour information
  <cr>
```

3.2.2.3.1 LIST INTERFACES

Displays the list of the VRF interfaces with brief amount of information on them.

Syntax:

```
PIM+list interfaces
```

Example:

```
PIM+list interfaces

 Virtual Interface Table
-------------------------------------------------------------------------------

 # Name               LocalAddress    Subnet             TTL  Flags
 0 ethernet0/0        192.168.213.246 192.168.212/23       1  NO-NBR DR
 1 ethernet0/1        1.1.1.2         1.1.1/24             1  NBRS PRIO LPD DR
 2 serial0/0          ---             ---                  1  DISABLED DOWN
 3 bri0/0             ---             ---                  1  DISABLED DOWN
 4 x25-node           ---             ---                  1  DISABLED DOWN
 5 fr1                5.5.5.2         5.5.5/24             1  NBRS PRIO LPD
 6 register           ---             ---                  1  REG

PIM+
```

This table is known as the *Virtual Interface Table* as it also includes the virtual register interface (*vif* register). Firstly the number of the *vif* and the name is presented for each interface and at the end the flags that condition the behavior. The intermediate fields are only logical if the interfaces have PIM configured.

This is the interpretation of the fields describing each interface:

- **#**: *vif* number.
- **Name**: *vif* name.
- **LocalAddress**: Main local IP address. This is used to advertise in the HELLOs.
- **Subnet**: Subnet the main IP pertains to with the subnet and mask.
- **TTL**: Minimum TTL that a multicast packet must have in order to be forwarded.
- **Flags**:
    - **DISABLED**: the interface is not configured for PIM.
    - **DOWN**: the interface is not up.
    - **DR**: the router has been elected as DR in this interface.
    - **NO-NBR**: there are no PIM neighbors through the interface.
    - **NBRS**: PIM neighbors have been detected through the interface.
    - **PRIO**: you need to use the *DR priority* option to elect the DR.
    - **LPD**: the HELLO *LAN Prune Delay* option has taken effect.
    - **BSR-BRDR**: the interface is configured as BSR Border.
    - **P2P**: the interface is considered point-to-point.
    - **REG**: register interface. This interface is not accessible to other applications and is not linked to any VRF.

3.2.2.3.2 LIST INTERFACES DETAILED

Displays in-depth information on the interfaces present in the VRF.

Syntax:

```
PIM+list interfaces detailed
```

Example:

```
PIM+list interfaces detailed

 Virtual Interface Table
-------------------------------------------------------------------------------
 # Name              LocalAddress    Subnet             TTL  Flags
 0 ethernet0/0       192.168.213.246 192.168.212/23       1  NO-NBR DR
          Hellos:    Interval: 30s    Next: 7s           Uptime: 3h33m26s
                                      GenID: 0x2CF9BE27  Prio: 1
                                      PropDelay: 500ms   OverrInt: 2500ms
 1 ethernet0/1       1.1.1.2         1.1.1/24             1  NBRS PRIO LPD DR
          Hellos:    Interval: 30s    Next: 20s          Uptime: 3h22m13s
                                      GenID: 0x4AFE536C  Prio: 1
                                      PropDelay: 500ms   OverrInt: 2500ms
      Neighbours:    1.1.1.1          Life: 94s          Uptime: 3h17m43s
                                      GenID: 0x02536D5F  Prio: 1
                                      PropDelay: 500ms   OverrInt: 2500ms
 2 serial0/0         ---             ---                  1  DISABLED DOWN
 3 bri0/0            ---             ---                  1  DISABLED DOWN
 4 x25-node          ---             ---                  1  DISABLED DOWN
 5 fr1               5.5.5.2         5.5.5/24             1  NBRS PRIO LPD
          Hellos:    Interval: 30s    Next: 7s           Uptime: 3h21m0s
                                      GenID: 0x0F7B13A0  Prio: 1
                                      PropDelay: 500ms   OverrInt: 2500ms
      Neighbours:    5.5.5.3          Life: 84s          Uptime: 3h20m54s    DR
                                      GenID: 0x66225484  Prio: 1
                                      PropDelay: 500ms   OverrInt: 2500ms
 6 register          ---             ---                  1  REG

PIM+
```

The first line shown for each interface is as described in the previous point, containing the summarized list of interfaces. Information is not provided on interfaces that don’t have PIM configured and on the register interface.

The following three lines for each interface describe the information sent by the router in the HELLO messages:

- **Interval**: Time interval between HELLOs, known in the configuration as *query-interval.*
- **Next**: Remaining seconds before sending the next periodic HELLO.
- **Uptime**: Time the PIM interface has been up.
- **GenID**: GenID value. This is a randomly selected value for each interface.
- **Prio**: Advertised DR priority.
- **PropDelay**: Advertised *Propagation delay*.
- **OverrInt**: Advertised *Override interval*.

Subsequently a list of PIM neighbors detected on the interface is shown and each of these provides the same information received in their HELLOs. The difference is that it doesn't show the interval between HELLOs, nor does it indicate when the next one is going to arrive. What it does do is to show the life time that the neighbor has left if it doesn't receive a new HELLO.

Additionally, in cases regarding the interfaces that haven't been elected as DR and consequently do not have the DR flag, the DR mark is included together with the neighbor that has been elected as DR.

3.2.2.4 LIST MRT

This displays the information on the *Multicast Routing Table* (MRT).

Syntax:

```
PIM+list mrt?
  one-group     Show (*,G), (S,G) and (S,G,rpt) states for one group
  groups        Show all (*,G), (S,G) and (S,G,rpt) states
  one-source    Show (S,G) and (S,G,rpt) states for one source
  any-group     Show all (*,*,RP) states
  one-rp        Show all states for one Rendezvous Point
  ssm           Show all (S,G) states for SSM groups
  <cr>          Show all states from MRT
```

3.2.2.4.1 LIST MRT

This displays the entire MRT without any type of filtering.

Syntax:

```
PIM+list mrt
```

Example:

```
PIM+list mrt
================================================================================
                          MULTICAST   ROUTING   TABLE
================================================================================


                         +----------------------------+
                         |                            |
                         |  ( * ,       238.1.1.1 )   |
                         |                            |
                         +----------------------------+

-------------------------------------(*,G)--------------------------------------
 +---------------------------------+          RP-addr   Flags
 |     ANY SOURCE,       238.1.1.1|           2.2.2.1   WC RP
 +---------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif    InhOif   Iif
ethernet0/1      |   167s  |         |       |          |    X    |    X    |
register         |         |         |       |          |         |         |  X

Uptime: 43s             Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Joined         JoinPruneT: 17s

-----------------------------------(S,G,rpt)------------------------------------
 +---------------------------------+          RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SGRPT RP PRUNE_DESIRED
 +--------------------------------+

Vif                Pruned   PrunePend Exclude LostAssert  ImmOif   InhOif   Iif
ethernet0/1      |         |         |       |          |         |    X   |
register         |         |         |       |          |         |        |  X

Uptime: 0s             Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Pruned

-------------------------------------(S,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SG SPT CACHE
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |         |        |  X
ethernet0/1      |         |         |       |          |         |    X   |

Uptime: 31s            Preference: 60          Metric: 2           KAT: 185s
Upstream Neighbour: 1.1.1.2           State: Joined         JoinPruneT: 29s

===============================================================================
Number of Groups listed: 1
Entries listed: 1 (*,G) - 1 (S,G,rpt) - 1 (S,G)
Number of Cache Mirrors: 1

PIM+
```

The MRT is divided into two parts: entries related to the ‘G’ multicast group (ordered by the group IP from lowest to highest) and the (*,*,RP) entries.

Entries related to the same ‘G’ group are preceded by a (*,G) text box:

```
                         +----------------------------+
                         |                            |
                         |  ( * ,       238.1.1.1 )   |
                         |                            |
                         +----------------------------+
```

Subsequently each new state included in the ‘G’ group begins with a dividing line where the type of state is indicated:

...**----(*,G)----**..., ...**----(S,G,rpt)----**..., ...**----(S,G)----**... E.g.:

```
-------------------------------------(*,G)-------------------------------------
```

Only the existing states in the MRT are displayed. The order of appearance is:

- (*,G) if it exists.

- Subsequently the entries depending on the sources, are ordered by the source IP address from lower to higher. For each source available in the MRT, first the (S,G,rpt) is presented followed by existing (S,G).

Within each state, the source is first indicated on the left (if applicable) and the group:

```
 +--------------------------------+
 | 192.168.212.18,       238.1.1.1|
 +--------------------------------+
```

In the (*,G) states that do not have a source associated, an **ANY SOURCE** text is displayed.

On the right the RP appears mapped to the multicast group and the flags applicable to the state. The possible flags are:

- **PMBR**: (*,*,RP) state.
- **WC**: (*,G) state.
- **SGRPT**: (S,G,rpt) state.
- **SG**: (S,G) state.
- **RP**: the tree root to be applied is RP.
- **SPT**: in an (S,G) state this indicates that the traffic is arriving through the *Shortest-Path-Tree* .

- **CACHE**: this state is referenced to an MFC cache entry.
- **PRUNE_DESIRED**: PruneDesired(S,G,rpt) algorithm gives a positive result.
- **A**:when in an (S,G) state, this indicates that the source is a candidate to be advertised by the MDSP protocol. This flag only appears if the MSDP protocol is configured in the device.
- **M**: when in an (S,G) state, this indicates that the state has been learnt through the MSDP protocol. This flag only appears if the MSDP protocol is configured in the device.

Below you will see a table that provides information depending on the interfaces (or *vifs*) affected in this state:

```
Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/1      |   167s  |         |       |          |    X   |    X   |
register         |         |         |       |          |        |        |  X
```

Each row in the table refers to an affected interface; in this example the **ethernet0/1** and **register**.

Each column describes the different aspects gathered by the PIM protocol on the interface:

- **Joined** or **Pruned** if this is (S,G,rpt): the interface in question has received a JOIN or a PRUNE if it is (S,G,rpt). This information is considered valid during the time indicated by the *Join/Prune Expiry Timer.* In the *register* interface, the **Joined** column indicates REGISTER messages are being sent and are only expressed with an '**X**'.
- **PrunePend**: the *Prune-Pending Timer* appears if it is active.
- **Include**, or **Exclude** if this is (S,G,rpt): IGMP has indicated there is a *host* interested in including or excluding traffic from the source. In (*,G) and (S,G) this means the *host* wishes to receive the corresponding traffic; in (S,G,rpt) this means that the traffic is not wanted (Exclude).
- **LostAssert**: indicates an ASSERT situation in the interface, if it exists:

  - Firstly the '**X**' or '-' character, depending if the RFC 4601 *lost_assert* macro has a positive or negative result.

  - Subsequently, '**W**' if it is the ASSERT winner in this interface, or '**L**' if it has lost.

  - Finally the interface *Assert Timer*.
- **ImmOif**: an '**X**' if the interface belongs to the *immediate_olist* described in RFC 4601.
- **InhOif**: an '**X**' if the interface belongs to the *inherited_olist* described in RFC 4601. The exception is the (S,G) state, where this field indicates whether it belongs to the *inherited_olist(S,G,rpt)*. In order to find out the *inherited_olist(S,G)*, you need to join **ImmOif** and **InhOif** for the state.
- **Iif**: an '**X**' in the interface which is the *incoming interface*. There can only be one *iif* in the table. If this is the *register* vif, this means this is RP.

Finally, a series of variables are added from the state:

- **Uptime**: time the state has been up.
- **Preference**: Route's administrative distance towards the tree root.
- **Metric**: Cost of the route towards the tree root.
- **KAT**: *Keepalive Timer*, only for (S,G) states.
- **Upstream neighbour**: Next hop on the route to the tree root.
- **State**: *Upstream Join/Prune State*.
- In (*,*,RP), (*,G) and (S,G):

  - NotJoined

  - **Joined**
- In (S,G,rpt):

  - **RPTNotJoined**

  - **NotPruned**

  - **Pruned**
- **JoinPruneT**: *Upstream Join/Prune Timer.*
- **OverrideT**: *Override Timer* for (S,G,rpt).
- **Register State**: only applicable for (S,G) if the register state is different from the **NoInfo** value. Possibilities:

  - **Join**

  - **Prune**

- **Join-Pending**

- **RegisterStopT:** Register-Stop Timer. Present in (S,G) if the Register State appears.

3.2.2.4.2 LIST MRT ONE-GROUP <GROUP> ONE-SOURCE <SOURCE>

Displays the S,G,rpt) and (S,G) states for the '*group*' and the '*source*' if they exist in the MRT. Only a maximum of two possible states can be presented.

Syntax:

```
PIM+list mrt one-group <group> one-source <source>
```

Example:

```
PIM+list mrt one-group 238.1.1.1 one-source 192.168.212.18



                         +----------------------------+
                         |                            |
                         |  ( * ,       238.1.1.1 )   |
                         |                            |
                         +----------------------------+

-----------------------------------(S,G,rpt)------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SGRPT RP PRUNE_DESIRED
 +--------------------------------+


Vif                Pruned   PrunePend Exclude LostAssert  ImmOif   InhOif   Iif
ethernet0/1      |   155s  |         |       |          |        |        |
register         |         |         |       |          |        |        |  X

Uptime: 3h23m12s       Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Pruned

-------------------------------------(S,G)--------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SG
 +--------------------------------+


Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |        |  X

Uptime: 3h23m43s       Preference: 60          Metric: 2           KAT: 169s
Upstream Neighbour: 1.1.1.2           State: NotJoined

PIM+
```

The interpretation of the states is indicated in the **list mrt.**

3.2.2.4.3 LIST MRT GROUPS

Displays the first half of the MRT, the part that includes all the (*,G), (S,G,rpt) and (S,G) states; i.e. it does not include the (*,*,RP).

Syntax:

```
PIM+list mrt groups
```

Example:

```
PIM+list mrt groups



================================================================================
                          MULTICAST   ROUTING   TABLE
================================================================================


                         +----------------------------+
                         |                            |
                         |  ( * ,       238.1.1.1 )  |
                         |                           |
                         +---------------------------+

-------------------------------------(*,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 |     ANY SOURCE,       238.1.1.1|           2.2.2.1   WC RP
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/1      |   183s  |         |       |          |    X   |    X   |
register         |         |         |       |          |        |        |  X

Uptime: 3h41m27s       Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Joined         JoinPruneT: 33s

-----------------------------------(S,G,rpt)-----------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SGRPT RP PRUNE_DESIRED
 +--------------------------------+

Vif                Pruned   PrunePend Exclude LostAssert  ImmOif   InhOif   Iif
ethernet0/1      |   183s  |         |       |          |        |        |
register         |         |         |       |          |        |        |  X

Uptime: 3h40m44s       Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Pruned

-------------------------------------(S,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SG
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |        |  X

Uptime: 3h41m15s       Preference: 60          Metric: 2           KAT: 133s
Upstream Neighbour: 1.1.1.2           State: NotJoined

PIM+
```

The interpretation of the states is indicated in the **list mrt**.

3.2.2.4.4 LIST MRT ONE-SOURCE <SOURCE>

Displays all the MRT (S,G,rpt) and (S,G) states that have a ‘*source*’. On this occasion, the group is not specified so there may be states from different multicast groups.

Syntax:

```
PIM+list mrt one-source <source>
```

Example:

```
PIM+list mrt one-source 192.168.213.88

                         +---------------------------+
                         |                           |
                         |  ( * , 239.255.255.250 )  |
                         |                           |
                         +---------------------------+

-------------------------------------(S,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.213.88, 239.255.255.250|           2.2.2.1   SG
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |        |  X
```

```
Uptime: 3m8s           Preference: 0           Metric: 0           KAT: 23s
Upstream Neighbour: ---               State: NotJoined
Register State: Prune          RegisterStopT: 52s

PIM+
```

The interpretation of the states is indicated in the **list mrt**.

3.2.2.4.5 LIST MRT ANY-GROUP

Lists all the states that aren’t associated with any multicast group. These are the (*,*,RP) states. This is the second part of the MRT where it’s normal to have few or no (*,*,RP) states.

Syntax:

```
PIM+list mrt any-group
```

Example:

```
PIM+list mrt any-group

===============================================================================
                          MULTICAST   ROUTING   TABLE
===============================================================================


                         +============================+
                         |                            |
                         |      (*,*,RP) STATES       |
                         |                            |
                         +============================+

-----------------------------------(*,*,RP)------------------------------------
 RP = 2.2.2.1            Flags: PMBR RP

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |   206s  |         |       |          |    X   |    X   |
ethernet0/1      |         |         |       |          |        |        |  X

Uptime: 4s             Preference: 60          Metric: 1
Upstream Neighbour: 1.1.1.1           State: Joined         JoinPruneT: 56s

PIM+
```

The interpretation of the states is very similar to that indicated in the **list mrt.** There are some small discrepancies in the state headers.

3.2.2.4.6 LIST MRT ONE-RP <RP_ADDRESS>

Lists all the states whose group is associated with the RP ‘*rp_address*’ together with the (*,*,RP) state whose RP is ‘*rp_address*’ if this exists.

Syntax:

```
PIM+list mrt one-rp <rp_address>
```

Example:

```
PIM+list mrt one-rp 2.2.2.1

                         +----------------------------+
                         |                            |
                         |  ( * ,       238.1.1.1 )   |
                         |                            |
                         +----------------------------+

-------------------------------------(*,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 |     ANY SOURCE,       238.1.1.1|           2.2.2.1   WC RP
 +--------------------------------+
Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |    X   |
ethernet0/1      |   191s  |         |       |          |    X   |    X   |
register         |         |         |       |          |        |        |  X

Uptime: 4h7m19s        Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Joined         JoinPruneT: 41s

-----------------------------------(S,G,rpt)-----------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SGRPT RP PRUNE_DESIRED
 +--------------------------------+

Vif                Pruned   PrunePend Exclude LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |    X   |
ethernet0/1      |   191s  |         |       |          |        |        |
register         |         |         |       |          |        |        |  X

Uptime: 4h6m36s        Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Pruned

-------------------------------------(S,G)-------------------------------------
 +--------------------------------+           RP-addr   Flags
 | 192.168.212.18,       238.1.1.1|           2.2.2.1   SG SPT CACHE
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |         |         |       |          |        |    X   |  X

Uptime: 4h7m7s         Preference: 60          Metric: 2           KAT: 133s
Upstream Neighbour: 1.1.1.2           State: Joined         JoinPruneT: 35s



===============================================================================

                         +===========================+
                         |                           |
                         |      (*,*,RP) STATES      |
                         |                           |
                         +===========================+

-----------------------------------(*,*,RP)------------------------------------
 RP = 2.2.2.1            Flags: PMBR RP

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |   205s  |         |       |          |    X   |    X   |
register         |         |         |       |          |        |        |  X

Uptime: 5s             Preference: 0           Metric: 0
Upstream Neighbour: ---               State: Joined         JoinPruneT: 55s

PIM+
```

The interpretation of the states is similar to that indicated in the **list mrt.**

3.2.2.4.7 LIST MRT SSM

Lists all the states whose group is an SSM multicast address. The SSM range is defined by IANA as 232.0.0.0/8.

Syntax:

```
PIM+list mrt ssm
```

Example:

```
PIM+list mrt ssm

                         +---------------------------+
                         |                           |
                         |  ( * ,       232.1.2.3 )  |
                         |                           |
                         +---------------------------+

-------------------------------------(S,G)-------------------------------------
 +--------------------------------+           SSM       Flags
 |       4.4.4.10,       232.1.2.3|                     SG
 +--------------------------------+

Vif                Joined   PrunePend Include LostAssert  ImmOif   InhOif   Iif
ethernet0/0      |   178s  |         |       |          |    X   |        |
fr1              |         |         |       |          |        |        |  X

Uptime: 5s             Preference: 60          Metric: 2           KAT: idle
Upstream Neighbour: 5.5.5.3           State: Joined         JoinPruneT: 55s

PIM+
```

The interpretation of the states is similar to that indicated in the **list mrt**. RP here is irrelevant as no RP is mapped to the SSM groups

3.2.2.5 LIST RP-SET

Lists information on the *RP-Set* for the VRF in question. Firstly information is provided on the BSR, subsequently the *RP-Set* content, which is made up of a group of RP assignments and a range of multicast groups.

Syntax:

```
PIM+list rp-set ?
  detailed    Show advanced information about RP and group prefixes
  one-rp      Show information of only one RP if present
  <cr>
```

3.2.2.5.1 LIST RP-SET

Lists all the *RP-Set.*

Syntax:

```
PIM+list rp-set
```

Example:

```
PIM+list rp-set

 RP Set
-------------------------------------------------------------------------------


Current BSR
Address:  1.1.1.1              Pri: 64       State: AP          Timer: 118s
Upstream: 2.2.2.1              Incoming vif: ethernet0/0



RP addr         Incoming vif      Upstr neighb    Group prefix       Pri T Time
5.5.5.2         serial0/0         5.5.5.2         225/8              192 B 50s
                                                  228/6              192 B 50s
2.2.2.1         ethernet0/0       2.2.2.1         224/4              192 S Frvr
1.1.1.2         ethernet0/0       2.2.2.1         234/7              160 B 50s

PIM+
```

The first part describes the BSR information. The current BSR is described first:

- **Address**: current BSR IP address.
- **Pri**: current BSR priority.
- **State**: BSR mechanism state; this can be:

  - Devices that are not candidates for BSR:

NOINFO: *NoInfo*

AA: *Accept Any*

AP: *Accept Preferred*

- Devices that are candidates for BSR:

C-BSR: *Candidate-BSR*

P-BSR: *Pending-BSR*

E-BSR: *Elected-BSR*

- **Timer**: *Bootstrap Timer*
- **Upstream**: next hop towards the BSR.
- **Incoming vif**: interface used to reach the BSR.

If the device is also a candidate for BSR, the IP address for its candidature and the priority are subsequently given.

The table with the “RP # multicast groups range” assignments appear after the BSR information. The entries are ordered by RP, with their IP addresses ordered from highest to lowest. The common information for an RP is only presented for the first RP assignment; the following assignments are added in successive rows. Each assignment presents the following:

- **RP addr**: IP address of the candidate for RP.
- **Incoming vif**: interface the RP is reached through.
- **Upstr neighb**: next hop towards the RP (*incoming vif* neighbor).
- **Group prefix**: range of multicast groups for the assignment. This is made up of:

  - Starting multicast group in the range, omitting the 0 on the right, e.g., 225 means IP 225.0.0.0 .

  - Length in bits of the range’s mask, e.g., 8 means a mask of 255.0.0.0 .
- According to the example, the range defined as 225/8 is the 225.0.0.0 to 225.255.255.255 range.
- **Pri**: priority of the candidate for RP.
- **T**: type of assignment. The possible options are:

  - **S**: static assignment.

  - **B**: dynamic assignment through the Bootstrap mechanism.
- **Time**: time that the assignment is valid for if it is not updated. **Frvr** (*Forever)* if it doesn’t expire.

3.2.2.5.2 LIST RP-SET DETAILED

Displays the complete *RP -Set* in more detail by breaking down the table.

Syntax:

```
PIM+list rp-set detailed
```

Example:

```
PIM+list rp-set detailed

 RP Set
--------

Current BSR
Address:  1.1.1.1              Pri: 64       State: E-BSR       Timer:  14s
Upstream: ---                  Incoming vif: ---

Own BSR candidacy
Address:  1.1.1.1              Pri: 64

RP address:              5.5.5.2
Incoming vif:            ethernet0/0
Upstream neighbour:      1.1.1.2
Metric:                  2
Preference:              60
```

```
(*,*,RP) entry:          no
RP-group 1:
   Group prefix:         225/8
     Hash mask:          255.255.255.252
     Fragment tag:       59824
     Group RP number:    1
   Origin type:          Bootstrap
   Holdtime:             131s
   Priority:             192
   Fragment tag:         59824
   Link to any group:    no
RP-group 2:
   Group prefix:         228/6
     Hash mask:          255.255.255.252
     Fragment tag:       59824
     Group RP number:    1
   Origin type:          Bootstrap
   Holdtime:             131s
   Priority:             192
   Fragment tag:         59824
   Link to any group:    no
RP address:              2.2.2.1
Incoming vif:            register
Upstream neighbour:      ---
Metric:                  0
Preference:              0
(*,*,RP) entry:          no
RP-group 1:
   Group prefix:         224/4
     Hash mask:          0.0.0.0
     Fragment tag:       0
     Group RP number:    1
   Origin type:          Static
   Holdtime:             Forever
   Priority:             192
   Fragment tag:         0
   Link to any group:    yes
RP address:              1.1.1.2
Incoming vif:            ethernet0/0
Upstream neighbour:      1.1.1.2
Metric:                  0
Preference:              0
(*,*,RP) entry:          no
RP-group 1:
   Group prefix:         234/7
     Hash mask:          255.255.255.252
     Fragment tag:       59824
     Group RP number:    1
   Origin type:          Bootstrap
   Holdtime:             131s
   Priority:             160
   Fragment tag:         59824
   Link to any group:    no

PIM+
```

The first part describes the BSR information, in the same way as the **list rp-set**. The result of the second part due to executing the command displays the allocation blocks, grouped by RP candidates. Some data on the RP in question is provided, followed by the assignment itself, with more detail and information on the variables defined in RFC 4601.

3.2.2.5.3 LIST RP-SET ONE-RP <RP_ADDRESS>

This only lists the assignments where the candidate for RP is '*rp_address*'.

Syntax:

```
PIM+list rp-set one-rp <rp_address>
```

Example:

```
PIM+list rp-set one-rp 2.2.2.1

RP addr         Incoming vif      Upstr neighb    Group prefix       Pri T Time
                        2.2.2.1         ethernet0/0       2.2.2.1          224/4             192 S Frvr

PIM+
```

This does not include BSR information and only displays the rows corresponding to the assignments table.

3.2.2.5.4 LIST RP-SET ONE-RP <RP_ADDRESS> DETAILED

This lists the assignments where the candidate for RP is ‘*rp_address*’, but with more in-depth details than in the table given with the previous command.

Syntax:

```
PIM+list rp-set one-rp <rp_address> detailed
```

Example:

```
PIM+list rp-set one-rp 2.2.2.1 detailed

RP address:              2.2.2.1
Incoming vif:            ethernet0/0
Upstream neighbour:      2.2.2.1
Metric:                  0
Preference:              0
(*,*,RP) entry:          no
RP-group 1:
   Group prefix:         224/4
     Hash mask:          0.0.0.0
     Fragment tag:       0
     Group RP number:    1
   Origin type:          Static
   Holdtime:             Forever
   Priority:             192
   Fragment tag:         0
   Link to any group:    yes

PIM+
```

3.2.2.6 LIST SOURCES

Lists the sources referred to in the VRF MRT states.

Syntax:

```
PIM+list sources?
  one-source    Show only one source if present
  <cr>          Show complete sources list
```

3.2.2.6.1 LIST SOURCES

Displays a complete list of sources, with the sources ordered by their IPs from lowest to highest.

Syntax:

```
PIM+list sources
```

Example:

```
PIM+list sources

 Sources list
-------------------------------------------------------------------------------
Source           Incoming vif      Upstr neighb     Metr  Pref  (S,G) (S,G,rpt)
192.168.212.10   ethernet0/0       ---                 0     0     1     0
192.168.212.18   ethernet0/0       ---                 0     0     1     0
192.168.212.31   ethernet0/0       ---                 0     0     1     0
```

```
192.168.212.70   ethernet0/0       ---                 0     0     1     0
192.168.212.89   ethernet0/0       ---                 0     0     1     0
192.168.212.100  ethernet0/0       ---                 0     0     1     0
192.168.213.88   ethernet0/0       ---                 0     0     1     0

Number of sources listed: 7

PIM+
```

Each source provides the following information:

- **Source**: source IP address.
- **Incoming vif**: interface used to reach the source.
- **Upstr neighb**: next hop in the route towards the source. This is empty if it is directly connected.
- **Metr**: cost of the route towards the source. The cost is 0 if it is directly connected.
- **Pref**: administrative distance of the route towards the source. The distance is 0 if it is directly connected.
- **(S,G)**: number of (S,G) states where the source is ‘S’.
- **(S,G,rpt)**: number of (S,G,rpt) states where the source is ‘S’.

3.2.2.6.2 LIST SOURCES ONE-SOURCE <SOURCE>

Lists the information on the source ‘*source*’ if it is in the list of sources.

Syntax:

```
PIM+list sources one-source <source>
```

Example:

```
PIM+list sources one-source 192.168.212.18

Source           Incoming vif      Upstr neighb     Metr  Pref  (S,G)  (S,G,rpt)

                        192.168.212.18   ethernet0/0       1.1.1.2             2    60     0     1

PIM+
```

The information on each source is the same as that explained in the previous point.

3.2.2.7 LIST TIMERS

Lists the advanced and descriptive information on the state of the timers used by the PIM.

Syntax:

```
PIM+list timers ?
  global         Select global PIM timers
  <interface>    Select PIM timers for one interface
  <cr>           List all PIM timers
```

3.2.2.7.1 LIST TIMERS

Displays a complete list of all timers used by the PIM in the VRF. This list begins with the global timers that aren’t associated with any specific interface. Subsequently the PIM own timers are listed for each PIM interface.

Syntax:

```
PIM+list timers
```

Example:

```
PIM+list timers

 Global timer list
-------------------------------------------------------------------------------

Type         Expiration    OrigDurat  Description
bsr            42s799ms      1m0s0ms  BSR 1.1.1.1, state E-BSR
join-prune     58s806ms      1m0s0ms  (*,238.1.1.1)
grp-rp-exp   1m39s501ms     2m30s0ms  RP 1.1.1.2 -> 234.0.0.0/7
```

```
grp-rp-exp   1m39s502ms     2m30s0ms  RP 5.5.5.2 -> 225.0.0.0/8
grp-rp-exp   1m39s502ms     2m30s0ms  RP 5.5.5.2 -> 228.0.0.0/6

Number of timers listed: 5

 Interface ethernet0/0 timer list
-------------------------------------------------------------------------------


Type         Expiration    OrigDurat  Description
hello           4s360ms       30s0ms  Periodic timer 30s
nbr-hello    1m22s948ms     1m45s0ms  Neighbour 1.1.1.2

Number of timers listed: 2

 Interface ethernet0/1 timer list
-------------------------------------------------------------------------------


Type         Expiration    OrigDurat  Description
hello          28s740ms       30s0ms  Periodic timer 30s
nbr-hello    1m43s741ms     1m45s0ms  Neighbour 2.2.2.3
jp-expiry    3m28s870ms     3m30s0ms  (*,238.1.1.1)
jp-expiry    3m28s871ms     3m30s0ms  (192.168.212.18,238.1.1.1,rpt)

Number of timers listed: 4

PIM+
```

The active timers are displayed beginning with those that are going to expire first. Each timer provides the following information:

- **Type**: type of timer.

  The possible types of global timers are as follows:

  - **keep-alive**: entry *Keepalive Timer* (S,G).
  - **reg-stop**: entry *Register-Stop Timer* (S,G).
  - **join-prune**: *Upstream Join/Prune Timer.*
  - **override**: entry *Override Timer* (S,G,rpt).
  - **bsr**: *Bootstrap Timer.*
  - **grp-rp-exp**: *Group-to-RP mapping Expiry Timer.*
  - **data-rate**: timer to measure the traffic *throughput* in the cache.
  - **ucast-rout**: timer used to update the states after changes in the unicast routing table.
- The types of the interface’s own timers are as follows:
  - **jp-expiry**: *Join/Prune Expiry Timer.*
  - **p-pending**: *Prune-Pending Timer.*
  - **assert**: *Assert Timer.*
  - **hello**: timer to send the next HELLO.
  - **nbr-hello**: timer with the lifetime of a PIM neighbor.
  - **c-rp-adv**: *C-RP Advertisement Timer.*
  - **pend-jp-msg**: timer to group various JOIN or PRUNE entries in the same JOIN/PRUNE.
- **Expiration**: remaining time until the timer expires.
- **OrigDurat**: time the timer was programmed with the last time.
- **Description**: description of the timer.

3.2.2.7.2 LIST TIMERS GLOBAL

Only displays the list of global timers that are not related to any specific interface.

Syntax:

```
PIM+list timers global
```

Example:

```
PIM+list timers global

 Global timer list
-------------------------------------------------------------------------------
Type         Expiration    OrigDurat  Description
bsr             1s444ms      1m0s0ms  BSR 1.1.1.1, state E-BSR
join-prune     17s430ms      1m0s0ms  (*,238.1.1.1)
grp-rp-exp   1m58s181ms     2m30s0ms  RP 1.1.1.2 -> 234.0.0.0/7
grp-rp-exp   1m58s181ms     2m30s0ms  RP 5.5.5.2 -> 225.0.0.0/8
grp-rp-exp   1m58s181ms     2m30s0ms  RP 5.5.5.2 -> 228.0.0.0/6

Number of timers listed: 5

PIM+
```

3.2.2.7.3 LIST TIMERS <INTERFACE>

Displays the list of timers associated with a specific '*interface*'.

Syntax:

```
PIM+list timers <interface>
```

Example:

```
PIM+list timers ethernet0/0

 Interface ethernet0/0 timer list
-------------------------------------------------------------------------------

Type         Expiration    OrigDurat  Description
hello           21s23ms       30s0ms  Periodic timer 30s
nbr-hello    1m39s639ms     1m45s0ms  Neighbour 1.1.1.2

Number of timers listed: 2

PIM+
```

3.2.3 STATS

The **stats** command displays the PIM protocol message counters.

Syntax:

```
PIM+stats ?
  global        Global statistics
  interfaces    Statistics per all interfaces
  <cr>          All statistics
PIM+stats
```

3.2.3.1 STATS

Presents all the information on the message counters.

Syntax:

```
PIM+stats
```

Example:

```
PIM+stats
```

```
Global packet statistics
========================
                          sent /   received
     Hello               1346 /       1347
     Register               0 /         74
     Null-register          0 /        261
     Register-stop        326 /          0
     Join-prune             8 /        339
     Bootstrap            679 /        675
     Assert                 0 /          0
     Cand-RP-adv            0 /       1354


Packet statistics per interface
===============================

 Vif   0:   ethernet0/0
                              sent /   received
         Hello                 632 /        633
         Register                0 /         61
         Null-register           0 /        216
         Register-stop         272 /          0
         Join-prune              5 /          0
         Bootstrap             318 /        317
         Assert                  0 /          0
         Cand-RP-adv             0 /       1260

 Vif   1:   ethernet0/1
                              sent /   received
         Hello                 672 /        671
         Register                0 /          0
         Null-register           0 /          0
         Register-stop           0 /          0
         Join-prune              0 /        339
         Bootstrap             339 /        337
         Assert                  0 /          0
         Cand-RP-adv             0 /          0

 Vif   2:   serial0/0            DISABLED

 Vif   3:   serial0/1            DISABLED

 Vif   4:   bri0/0               DISABLED

 Vif   5:   x25-node             DISABLED

 Vif   6:   voip1/0              DISABLED

 Vif   7:   register             No packets (virtual)

PIM+
```

The counter information is divided into two blocks. First the global counters, which are the ones that are added, are displayed. Subsequently the message counters broken down per interface and the sent and received messages are noted for each type of message.

3.2.3.2 STATS GLOBAL

Presents the added information on the message counters.

Syntax:

```
PIM+stats global
```

Example:

```
PIM+stats global
```

```
Global packet statistics
========================
                          sent /   received
     Hello                2053 /       2056
     Register                0 /          0
     Null-register           0 /          0
     Register-stop           0 /          0
     Join-prune            683 /        682
     Bootstrap            1031 /       1034
     Assert                  0 /          0
     Cand-RP-adv             0 /          0

PIM+
```

3.2.3.3 STATS INTERFACE

Displays the message counters broken down by interface.

Syntax:

```
PIM+stats interfaces
```

Example:

```
PIM+stats interfaces

Packet statistics per interface
===============================

 Vif   0:   ethernet0/0
                              sent /   received
         Hello                2642 /       2643
         Register                0 /        512
         Null-register           0 /       1775
         Register-stop        2281 /          0
         Join-prune              7 /          0
         Bootstrap            1323 /       1322
         Assert                  0 /          0
         Cand-RP-adv             0 /       5280

 Vif   1:   ethernet0/1
                              sent /   received
         Hello                2682 /       2681
         Register                0 /          0
         Null-register           0 /          0
         Register-stop           0 /          0
         Join-prune              0 /       1346
         Bootstrap            1344 /       1342
         Assert                  0 /          0
         Cand-RP-adv             0 /          0

 Vif   2:   serial0/0            DISABLED
 Vif   3:   serial0/1            DISABLED
 Vif   4:   bri0/0               DISABLED
 Vif   5:   x25-node             DISABLED
 Vif   6:   voip1/0              DISABLED
 Vif   7:   register             No packets (virtual)

PIM+
```

3.2.4 VRF

The **vrf** command accesses the monitoring menu for the VRF *vrf_name* PIM protocol. To make this possible, the said VRF must be already created and have the PIM protocol enabled. The device's main VRF is monitored in the main PIM protocol monitoring menu and is available even though the PIM isn't enabled in it. The **vrf** command isn't in the protocol menu for a VRF if it isn't the router's main VRF.

Syntax:

```
PIM+vrf <vrf name>
```

Example:

```
PIM+vrf secondary

-- PIM protocol monitor for a VRF --

PIM vrf+?
  clear    Clear information
  list     List PIM information
  stats    Show statistics
  exit
PIM vrf+
```

3.2.5 EXIT

Use the **exit** command to exit the protocol's monitoring menu and return to the device's general monitoring prompt.

Syntax:

```
PIM+exit
```

Example:

```
PIM+exit
+
```

Chapter 4 Examples

4.1 PIM-SM Scenario with static RP

This first example shows how to configure different PIM devices in a simple scenario, with the main roles: DR, LHR, and RP with an intermediate router on the PIM network.

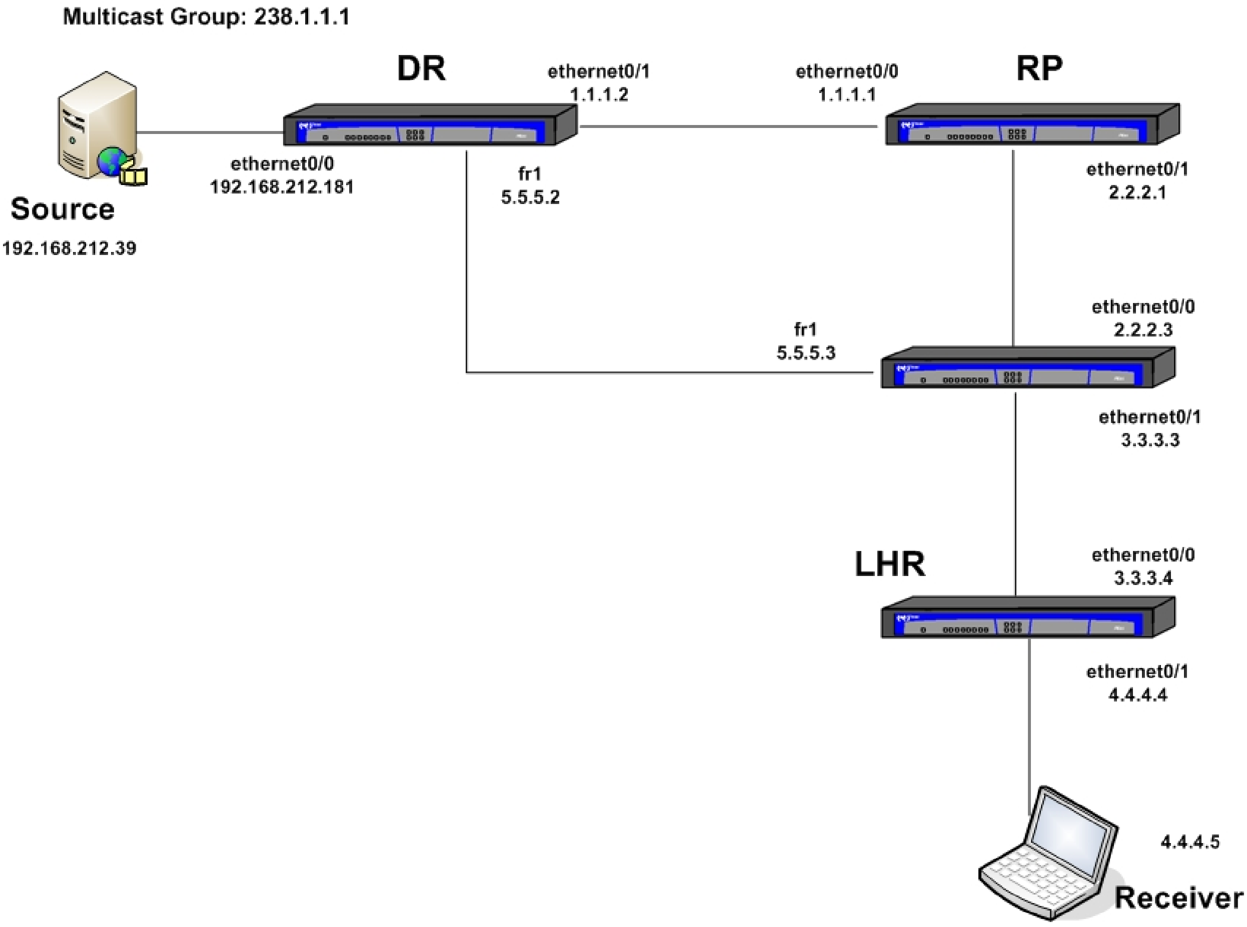

Fig. 11: Scenario for an example with static RP

4.1.1 Configuring the devices

4.1.1.1 Configuring RP

The device has the *Rendezvous-Point* feature and is determined from the beginning to be the network static RP. It only has two PIM interfaces and doesn't have IGMP configured in either of them as it isn't in direct contact with any IGMP *host*.

The configuration is as follows:

```
   log-command-errors
   no configuration
   set hostname RP
   set inactivity-timer disabled
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 1.1.1.1 255.255.255.0
;
                     ip pim sparse-mode
                     exit
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
```

```
      ip address 2.2.2.1 255.255.255.0
;
                     ip pim sparse-mode
   exit
;
   protocol pim
; -- PIM protocol user configuration --
                     enable
;
                     rp-address 2.2.2.1
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 3.3.3.0 255.255.255.0 2.2.2.3
      route 4.4.4.0 255.255.255.0 2.2.2.3 2
      route 192.168.212.0 255.255.254.0 1.1.1.2 2
      route 5.5.5.0 255.255.255.0 1.1.1.2 2
   exit
;
   dump-command-errors
   end
```

PIM-SM has been configured on the Ethernet interfaces, and the PIM protocol enabled. Address 2.2.2.1 has been configured as RP, which happens to be the device's own address, so it assumes the RP role.

Static routes are also configured, adding a cost 2 to those networks that are two hops away.

4.1.1.2 Configuring DR

The DR device executes the role of *Designated Router* for network 192.168.212.0/23, where the source is located. Thanks to this, multicast traffic generated by the source (S) is encapsulated so it can reach the RP.

The device configuration is as follows:

```
   log-command-errors
   no configuration
   set hostname DR
   set inactivity-timer disabled
   add device fr 1
   set data-link sync serial0/0
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.181 255.255.254.0
;
                     ip pim sparse-mode
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 1.1.1.2 255.255.255.0
;
                     ip pim sparse-mode
;
   exit
;
;
   network fr1
; -- Generic FR User Configuration --
      ip address 5.5.5.2 255.255.255.0
;
                     ip pim sparse-mode
```

```
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
      fr
; -- Frame Relay user configuration --
         pvc 21 default
;
         point-to-point-line 21
         no lmi
      exit
;
   exit
;
;
   protocol pim
; -- PIM protocol user configuration --
                     enable
;

                     rp-address 2.2.2.1
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 2.2.2.0 255.255.255.0 1.1.1.1
      route 3.3.3.0 255.255.255.0 5.5.5.3
      route 4.4.4.0 255.255.255.0 5.5.5.3 2
;
   exit
;
   dump-command-errors
   end
```

PIM-SM has been configured in the Ethernet and frame-relay interfaces and the PIM protocol enabled. Address 2.2.2.1 has been configured as RP. In this case, IGMP hasn't been configured in any interface either, as there is no direct contact with any IGMP *host*.

Static routes are also configured, adding a cost 2 to those networks that are two hops away.

4.1.1.3 Configuring LHR

The LHR device executes the role of *LastHop Router* in this PIM domain. This is the first router the receivers come across when requesting determined multicast traffic. It only has two PIM interfaces and has IGMP configured in one of the ethernet interfaces as this is in direct contact with IGMP *hosts*.

The device configuration is as follows:

```
   log-command-errors
   no configuration
   set hostname LHR
   set inactivity-timer disabled
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 3.3.3.4 255.255.255.0
;
                      ip pim sparse-mode
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
```

```
      ip address 4.4.4.4 255.255.255.0
;
                     ip pim sparse-mode
;

                     ip igmp downstream default
;
   exit
;
;
   protocol pim
; -- PIM protocol user configuration --

                      enable
;
                     rp-address 2.2.2.1
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 192.168.212.0 255.255.254.0 3.3.3.3 2
      route 2.2.2.0 255.255.255.0 3.3.3.3
      route 1.1.1.0 255.255.255.0 3.3.3.3 2
      route 5.5.5.0 255.255.255.0 3.3.3.3
;
   exit
;
;
   dump-command-errors
   end
```

4.1.1.4 Configuring the intermediate router

The intermediate device (the one shown in Figure 11 without a label) only forwards multicast traffic through the corresponding interfaces.

As the interfaces forming part of the tree must have PIM enabled, PIM-SM has been configured in the Ethernet and frame-relay interfaces and the PIM protocol has been enabled. Address 2.2.2.1 has been configured as the RP.

The device configuration is as follows:

```
   log-command-errors
   no configuration
   set hostname intermed
   set inactivity-timer disabled
   add device fr 1
   set data-link sync serial0/0
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 2.2.2.3 255.255.255.0
;
                     ip pim sparse-mode
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 3.3.3.3 255.255.255.0
;
                     ip pim sparse-mode
;
   exit
;
   network fr1
; -- Generic FR User Configuration --
```

```
      ip address 5.5.5.3 255.255.255.0
;
                     ip pim sparse-mode
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
      fr
; -- Frame Relay user configuration --
         pvc 21 default
;
         point-to-point-line 21
         no lmi
      exit
;
   exit
;
   protocol pim
; -- PIM protocol user configuration --
                     enable
;
                     rp-address 2.2.2.1
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 192.168.212.0 255.255.254.0 5.5.5.2
      route 1.1.1.0 255.255.255.0 2.2.2.1
      route 4.4.4.0 255.255.255.0 3.3.3.4
;
   exit
;
;
   dump-command-errors
   end
```

Once you have configured all the devices, you need to save the configuration and restart them so the changes activate.

4.1.2 User scenario with static RP

The receiver must execute a request from the traffic from the 238.1.1.1 multicast group through an *IGMP Report* message.

When the source begins to send traffic for the ‘G’ multicast group, this flows through the *Shared-Tree* , whose path to the receiver is through the DR Ethernet, the RP, the intermediate router and the LHR interfaces.

With the current configuration, the LHR will decide to switch to the *Shortest-Path-Tree* with the first packet it receives so the traffic at this point flows through the DR frame-relay and the intermediate interfaces to later pass to the LHR, thus avoiding traffic encapsulation and the path through the RP Figure 12.

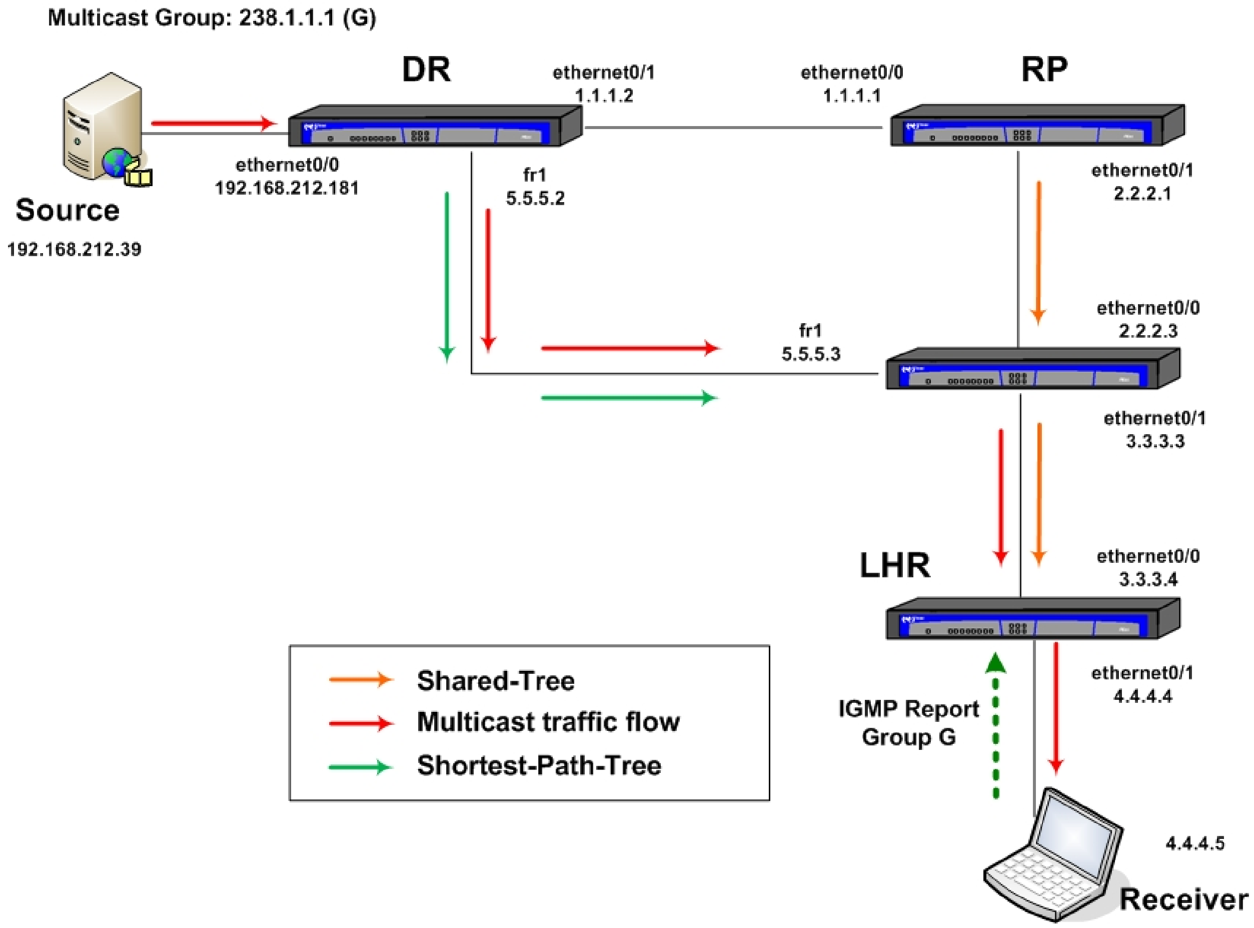

Fig. 12: User scenario with static RP – SPT

4.2 PIM-SSM Scenario

This example shows how to configure different PIM devices for a *Source Specific Multicast* (SSM) scenario.

There are no RPs in the PIM-SSM so only the DR and the LHR are playing their parts. The client IGMP petitions must be IGMP version 3, which allow us to select the traffic for the multicast group from a specific source.

The scenario for PIM-SSM is the same as for the previous example, except that there is no RP and the multicast group must be within the 232/8 range, since you are **not** going to specify any other ranges using the “ssm” command from the PIM menu in this example.

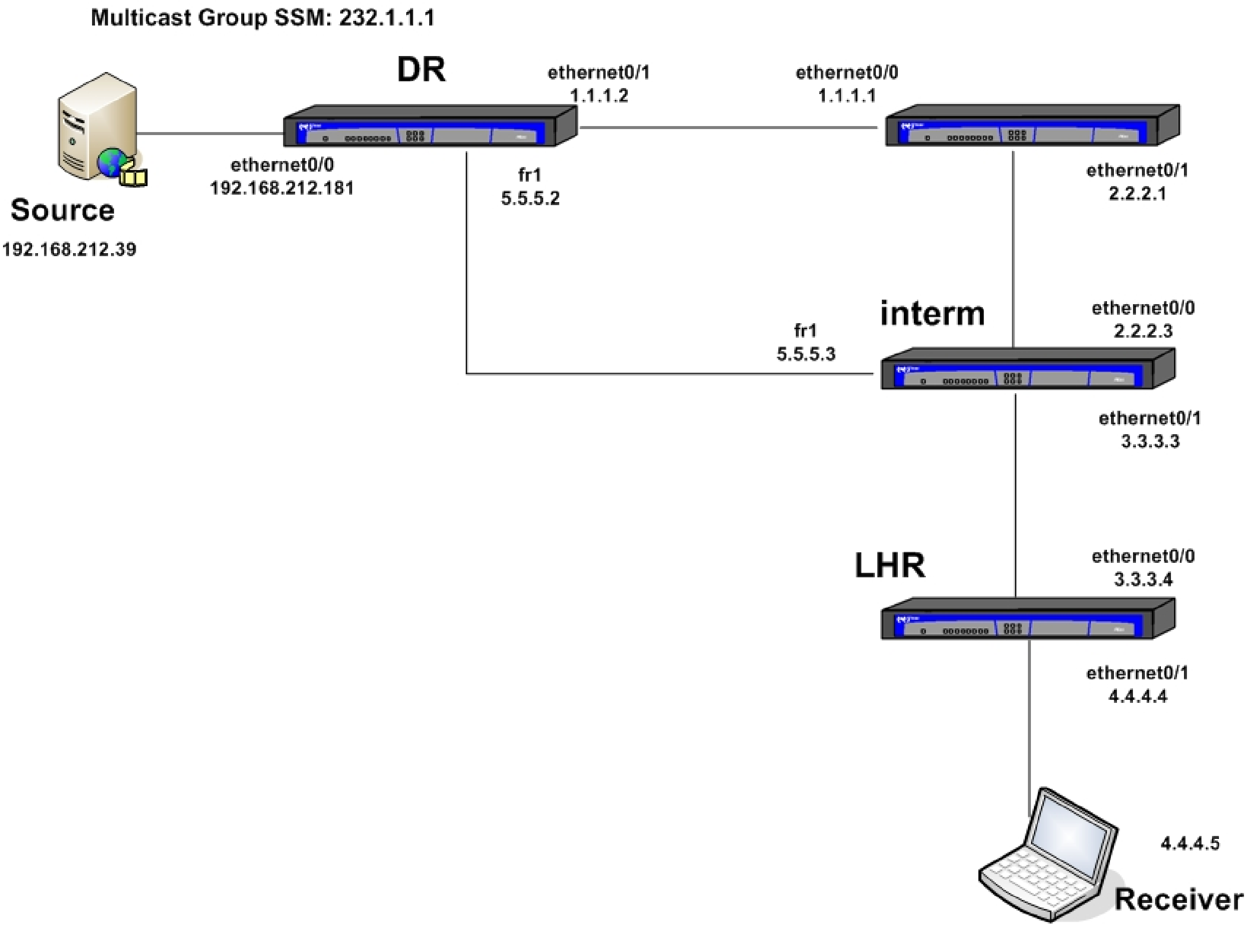

Fig. 13: PIM-SSM example scenario

4.2.1 Configuring the devices

As already mentioned, in this scenario there are no RPs, so the only difference in the devices' configuration compared to the previous example is that they do not need to specify the RP address.

4.2.2 PIM-SSM scenario operation

When the LHR receives the traffic petition from the receiver for a 'G' multicast group from the SSM range and a specific 'S' source, an entry (S,G) in its MRT is created and a JOIN (S,G) message is sent to its *upstream neighbour*, the intermediate router.

The intermediate router after receiving the JOIN (S,G) from the LHR, proceeds to do the same as the LHR and sends a JOIN (S,G) message to the DR.

The traffic sent by the 'S' source for the 'G' group flows through the created tree towards the source (SPT), without REGISTER messages, without RP roles and without *Shared-Tree* .

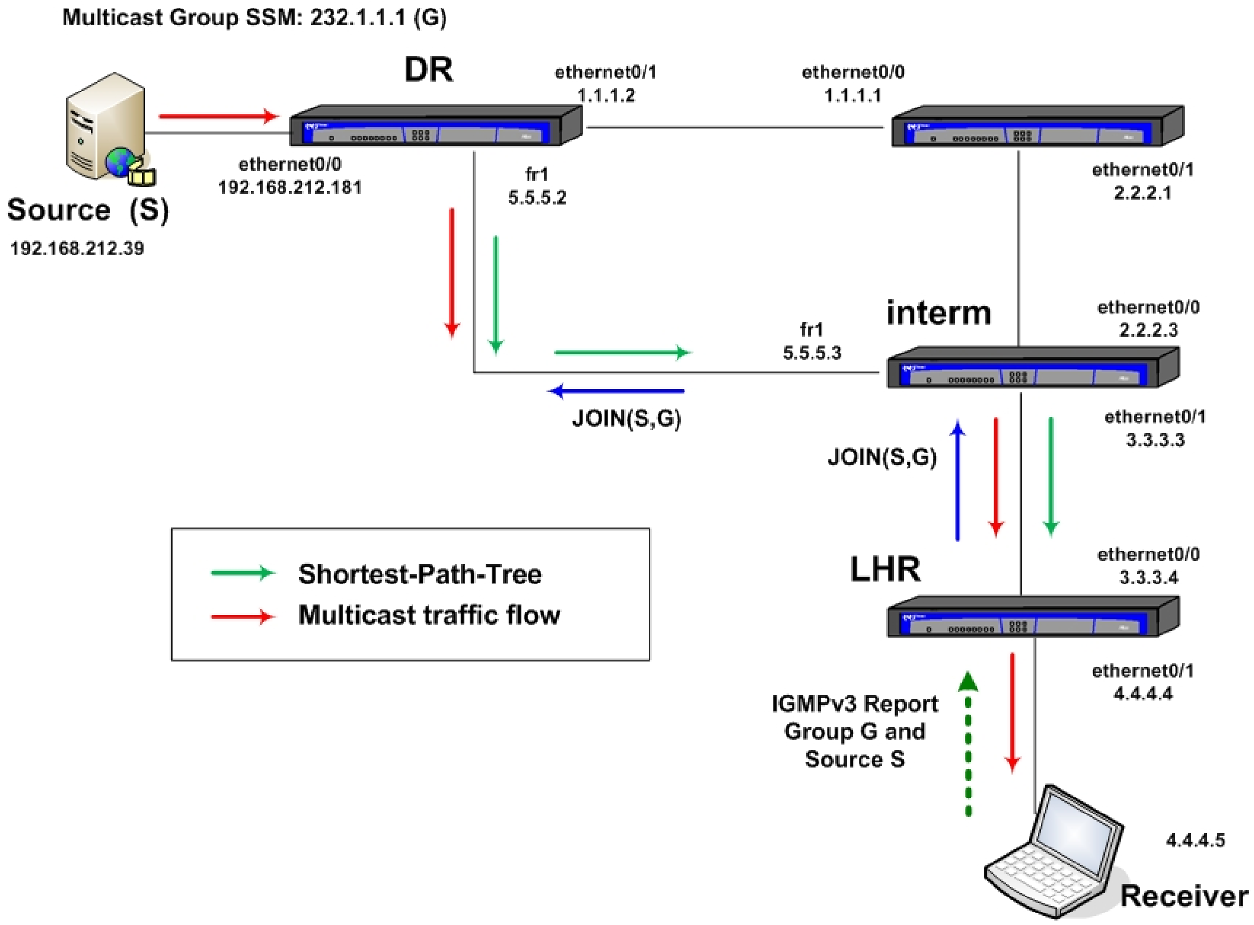

Fig. 14: PIM-SSM operating scenario