	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
Type	FRR	FRR	FRR	FRR				
Commit ID	ge1b0c9399	a87315e	gff905c6c3	9931db7				
Commit Date	2020-02-06	2020-06-14	2021-05-27	2021-07-29				
PIM-SM-1.1	RFC 4601 s3 p7 PIM-SM Protocol Overview							
MUST	`Regardless of how it is created, the primary role of the MRIB in the PIM protocol is to provide the next hop router along a multicast-capable path to each destination subnet.The MRIB is used to determine the next hop neighbor to which any PIM Join/Prune message is sent`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.2	NEGATIVE RFC 4601 s3 p7 PIM-SM Protocol Overview							
MUST	`Regardless of how it is created, the primary role of the MRIB in the PIM protocol is to provide the next hop router along a multicast-capable path to each destination subnet. The MRIB is used to determine the next hop neighbor to which any PIM Join/Prune message is sent`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.3	RFC 4601 s3.1 p8 Phase One: RP Tree							
MAY	`In phase one, a multicast receiver expresses its interest in receiving traffic destined for a multicast group.  Typically it does this using IGMP[6] or MLD[4], but other mechanisms might also serve this purpose.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.4	RFC 4601 s3.1 p8 Phase One: RP Tree							
MUST	`Join messages are resent periodically so long as the receiver remains in the group`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-1.5 **MUST**	NEGATIVE: RFC 4601 s3.1 p8 Phase One: RP Tree							
	The RP receives these encapsulated data packets, decapsulates them, and forwards them onto the shared tree.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.6 **MUST**	RFC 4601 s3.2 p9 Phase Two: Register-Stop							
	Although Register-encapsulation may continue indefinitely, for these reasons, the RP will normally choose to switch to native forwarding. To do this, when the RP receives a register-encapsulated data packet from source S on group G, it will normally initiate an (S,G) source-specific Join towards S.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.7 **MUST**	RFC 4601 s3.2 p9 Phase Two: Register-Stop							
	When packets from S also start to arrive natively at the RP, the RP will be receiving two copies of each of these packets. At this point, the RP starts to discard the encapsulated copy of these packets, and it sends a RegisterStop message back to S's DR to prevent the DR unnecessarily encapsulating the packets.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.8 **MUST**	RFC 4601 s3.3 p10 Phase Three: Shortest-Path Tree							
	To obtain lower latencies, a router on the receiver's LAN, typically the DR, may optionally initiate a transfer from the shared tree to a source-specific shortest-path tree (SPT). To do this, it issues an (S,G) Join towards S.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-1.9 **MUST**	RFC 4601 s3.3 p10 Phase Three: Shortest-Path Tree							
	At this point the receiver (or a router upstream of the receiver) will be receiving two copies of the data - one from the SPT and one from the RPT. When the first traffic starts to arrive from the SPT, the DR or upstream router starts to drop the packets for G from S that arrive via the RP tree.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.10 **MUST**	RFC 4601 s3.3 p10 Phase Three: Shortest-Path Tree							
	At this point the receiver (or a router upstream of the receiver) will be receiving two copies of the data - one from the SPT and one from the RPT. When the first traffic starts to arrive from the SPT, the DR or upstream router starts to drop the packets for G from S that arrive via the RP tree. In addition, it sends an (S,G) Prune message towards the RP. This is known as an (S,G,rpt) Prune. (Note: Here DUT is considered as an upstream router. The verification is made that the Join/Prune msg send by DUT has RPT-bit set)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.11 **MAY**	RFC 4601 s3.4 p10 Source-Specific Joins							
	IGMPv3 permits a receiver to join a group and specify that it only wants to receive traffic for a group if that traffic comes from a particular source.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.12 **MAY**	NEGATIVE RFC 4601 s3.4 p10 Source-Specific Joins							
	IGMPv3 permits a receiver to join a group and specify that it only wants to receive traffic for a group if that traffic comes from a particular source. (Note: Send UDP data packet from different source)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-1.13 **MAY**	RFC 4601 s3.4 p10 Source-Specific Joins							
	The range of multicast addresses from 232.0.0.0 to 232.255.255.255 is currently set aside for source-specific multicast in IPv4. For groups in this range, receivers should only issue source-specific IGMPv3 joins. If a PIM router receives a non-source-specific join for a group in this range, it should ignore it.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.14 **MAY**	NEGATIVE RFC 4601 s3.4 p10 Source-Specific Joins							
	The range of multicast addresses from 232.0.0.0 to 232.255.255.255 is currently set aside for source-specific multicast in IPv4. For groups in this range, receivers should only issue source-specific IGMPv3 joins. If a PIM router receives a non-source-specific join for a group in this range, it should ignore it. (Note: Send IGMPv3 Membership Report with empty source list)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.15 **MAY**	RFC 4601 s3.5 p10 Source-Specific Prunes							
	IGMPv3 also permits a receiver to join a group and specify that it only wants to receive traffic for a group if that traffic does not come from a specific source or sources. In this case, the DR will perform a (*,G) join as normal, ...							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.16 **MAY**	RFC 4601 s3.7 p12 RP Discovery							
	PIM-SM routers need to know the address of the RP for each group for which they have (*,G) state. This address is obtained either automatically (e.g., embedded-RP), through a bootstrap mechanism or through static configuration. (Note: through bootstrap mechanism)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-1.17 **MAY**	RFC 4601 s3.7 p12 RP Discovery							
	`PIM-SM routers need to know the address of the RP for each group for which they have (*,G) state. This address is obtained either automatically (e.g., embedded-RP), through a bootstrap mechanism or through static configuration. (Note: through static configuration)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.18 **MUST**	ANVL Setup Verification							
	`Quick test to verify that DUT sends Assert message with metric value correctly`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.19 **MUST**	ANVL Setup Verification							
	`Quick test to verify that DUT sends Assert message with metric preference value correctly`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-1.20 **MUST**	ANVL Setup Verification							
	`Quick test to verify that DUT sends Register message with IP Source set to the IP address where it come from.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-2.1	RFC 4601 s4.1.2 p15 (*,*,RP) State							
MUST	`The upstream (*,*,RP) Join/Prune Timer is used to send out periodic Join(*,*,RP) messages, and to override Prune(*,*,RP) messages from peers on an upstream LAN interface.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-2.2	RFC 4601 s4.1.2 p15 (*,*,RP) State							
MUST	`The last RPF neighbor towards the RP is stored because if the MRIB changes then the RPF neighbor towards the RP may change. If it does so, then we need to trigger a new Join (*,*,RP) to the new upstream neighbor`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-2.3	RFC 4601 s4.1.2 p15 (*,*,RP) State							
MUST	`The last RPF neighbor towards the RP is stored because if the MRIB changes then the RPF neighbor towards the RP may change. If it does so, then we need to trigger a Prune(*,*,RP) to the old upstream neighbor.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-3.1	RFC 4601 s4.1.3 p17 (*,G) State							
MUST	`The upstream (*,G) Join/Prune Timer is used to send out periodic Join(*,G) messages, and to override Prune(*,G) messages from peers on an upstream LAN interface.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-3.2 **MUST**	RFC 4601 s4.1.3 p17 (*,G) State							
	`The last RPF neighbor towards the RP is stored because if the MRIB changes then the RPF neighbor towards the RP may change. If it does so, then we need to trigger a new Join (*,G) to the new upstream neighbor`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-3.3 **MUST**	RFC 4601 s4.1.3 p17 (*,G) State							
	`The last RPF neighbor towards the RP is stored because if the MRIB changes then the RPF neighbor towards the RP may change. If it does so, then we need to trigger a Prune(*,G) to the old upstream neighbor.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.1 **MUST**	RFC 4601 s4.1.4 p19 (S,G) State							
	`The upstream (S,G) Join/Prune Timer is used to send out periodic Join(S,G) messages, and to override Prune(S,G) messages from peers on an upstream LAN interface.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.2 **MUST**	RFC 4601 s4.1.4 p19 (S,G) State							
	`The last RPF neighbor towards the S is stored because if the MRIB changes then the RPF neighbor towards the S may change. If it does so, then we need to trigger a new Join (S,G) to the new upstream neighbor`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-4.3 **MUST**	RFC 4601 s4.1.4 p19 (S,G) State							
	The last RPF neighbor towards the S is stored because if the MRIB changes then the RPF neighbor towards the S may change. If it does so, then we need to trigger a Prune(S,G) to the old upstream neighbor.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.4 **MUST**	RFC 4601 s4.1.4 p19 (S,G) State							
	If the router detects through a changed GenID in a Hello message that the upstream neighbor towards S has rebooted, then it should re-instantiate state by sending a Join(S,G).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.5 **MUST**	NEGATIVE RFC 4601 s4.1.4 p19 (S,G) State							
	The SPTbit is used to indicate whether forwarding is taking place on the (S,G) Shortest Path Tree (SPT) or on the (*,G) tree. A router can have (S,G) state and still be forwarding on (*,G) state during the interval when the source-specific tree is being constructed. When SPTbit is FALSE, only (*,G) forwarding state is used to forward packets from S to G. When SPTbit is TRUE, both (*,G) and (S,G) forwarding state are used. (Note: when SPTbit is FALSE, because JoinDesired(S,G) == FALSE for different source)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.6 **MUST**	NEGATIVE RFC 4601 s4.1.4 p19 (S,G) State							
	The SPTbit is used to indicate whether forwarding is taking place on the (S,G) Shortest Path Tree (SPT) or on the (*,G) tree. A router can have (S,G) state and still be forwarding on (*,G) state during the interval when the source-specific tree is being constructed. When SPTbit is FALSE, only (*,G) forwarding state is used to forward packets from S to G. When SPTbit is TRUE, both (*,G) and (S,G) forwarding state are used. (Note: when SPTbit is FALSE, because JoinDesired(S,G) == FALSE for different group)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-4.7 **MUST**	RFC 4601 s4.1.4 p19 (S,G) State							
	The SPTbit is used to indicate whether forwarding is taking place on the (S,G) Shortest Path Tree (SPT) or on the (*,G) tree. A router can have (S,G) state and still be forwarding on (*,G) state during the interval when the source-specific tree is being constructed. When SPTbit is FALSE, only (*,G) forwarding state is used to forward packets from S to G. When SPTbit is TRUE, both (*,G) and (S,G) forwarding state are used. (Note: when SPTbit is TRUE)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-4.8 **MUST**	RFC 4601 s4.1.4 p20 (S,G) State							
	Amongst other things, this is necessary for the so-called "turnaround rules" - when the RP uses (S,G) joins to stop encapsulation, and then (S,G) prunes to prevent traffic from unnecessarily reaching the RP.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-5.1 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if( iif == RPF_interface(S) AND SPTbit(S,G) == TRUE ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif == RPF_interface(RP(G)) AND SPTbit(S,G) == FALSE) {`
	`  oiflist = inherited_olist(S,G,rpt)`
	`  CheckSwitchToSpt(S,G)`
	`} else {`
	`  # Note: RPF check failed`
	`  # A transition in an Assert FSM, may cause an Assert(S,G)`
	`  # or Assert(*,G) message to be sent out interface iif.`
	`  # See section 4.6 for details.`
	`  if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`  } else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`  }`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`
	`(Note: If the SPT-bit of an (S,G) entry is set, and if incoming interface is the same as a matching (S,G) ifaceIn, the packet is forwarded to the oif-list of (S,G))`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-5.2 **MUST**	NEGATIVE RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if( iif == RPF_interface(S) AND SPTbit(S,G) == TRUE ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif == RPF_interface(RP(G)) AND SPTbit(S,G) == FALSE) {`
	`  oiflist = inherited_olist(S,G,rpt)`
	`  CheckSwitchToSpt(S,G)`
	`} else {`
	`  # Note: RPF check failed`
	`  # A transition in an Assert FSM, may cause an Assert(S,G)`
	`  # or Assert(*,G) message to be sent out interface iif.`
	`  # See section 4.6 for details.`
	`  if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`  } else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`  }`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`
	`(Note: If the SPT-bit of an (S,G) entry is set, and if incoming`
	`interface is same as RPF_interface(s), the packet`
	`is forwarded to the oif-list of (S,G))`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-5.3 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if( iif == RPF_interface(S) AND SPTbit(S,G) == TRUE ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif == RPF_interface(RP(G)) AND SPTbit(S,G) == FALSE) {`
	`  oiflist = inherited_olist(S,G,rpt)`
	`  CheckSwitchToSpt(S,G)`
	`} else {`
	`  # Note: RPF check failed`
	`  # A transition in an Assert FSM, may cause an Assert(S,G)`
	`  # or Assert(*,G) message to be sent out interface iif.`
	`  # See section 4.6 for details.`
	`  if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`  } else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`  }`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`
	`(Note: On receiving multicast data packet if SPT-bit of an (S,G)`
	`entry is cleared, and ifaceIn differs than a matching (S,G) ifaceIn`
	`but matches with a (*,G) ifaceIn, packet is forwarded to the`
	`oif-list of (*,G))`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-5.4 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if( iif == RPF_interface(S) AND SPTbit(S,G) == TRUE ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif == RPF_interface(RP(G)) AND SPTbit(S,G) == FALSE) {`
	`  oiflist = inherited_olist(S,G,rpt)`
	`  CheckSwitchToSpt(S,G)`
	`} else {`
	`  # Note: RPF check failed`
	`  # A transition in an Assert FSM, may cause an Assert(S,G)`
	`  # or Assert(*,G) message to be sent out interface iif.`
	`  # See section 4.6 for details.`
	`  if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`  } else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`  }`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`
	`(Note: On receiving multicast data packet, if incoming interface does`
	`not match (S,G) ifaceIn or (*,G) ifaceIn, the packet is not forwarded)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-5.5 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if( iif == RPF_interface(S) AND SPTbit(S,G) == TRUE ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif == RPF_interface(RP(G)) AND SPTbit(S,G) == FALSE) {`
	`  oiflist = inherited_olist(S,G,rpt)`
	`  CheckSwitchToSpt(S,G)`
	`} else {`
	`  # Note: RPF check failed`
	`  # A transition in an Assert FSM, may cause an Assert(S,G)`
	`  # or Assert(*,G) message to be sent out interface iif.`
	`  # See section 4.6 for details.`
	`  if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`  } else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`  }`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`
	`(Note: If the SPT-bit of an (S,G) entry is not set, and if incoming`
	`interface is the same as a matching RPF_interface(RP(G)), the packet is`
	`forwarded to the oif-list of (S,G,rpt))`						

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-5.6 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`} else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`}`
	`(Note: On receipt a data from S to G on interface iif, if SPT-bit is TRUE,`
	`it will send an Assert(S,G) on iif.)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-5.7 **MUST**	RFC 4601 s4.2 p27 Data Packet Forwarding Rules							

	`if ( SPTbit(S,G) == TRUE AND iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`} else if ( SPTbit(S,G) == FALSE AND`
	`  iif is in inherited_olist(S,G,rpt) {`
	`  send Assert(*,G) on iif`
	`}`
	`(Note: On receipt a data from S to G on interface iif, if SPT-bit is FALSE,`
	`it will send an Assert(*,G) on iif.)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-6.1 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	`Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:`
	`void`
	`Update_SPTbit(S,G,iif) {`
	`  if ( iif == RPF_interface(S)`
	`  AND JoinDesired(S,G) == TRUE`
	`  AND ( DirectlyConnected(S) == TRUE`
	`  OR RPF_interface(S) != RPF_interface(RP(G))`
	`  OR inherited_olist(S,G,rpt) == NULL`
	`  OR ( ( RPF'(S,G) == RPF'(*,G) ) AND`
	`  ( RPF'(S,G) != NULL ) )`
	`  OR ( I_Am_Assert_Loser(S,G,iif) ) {`
	`  Set SPTbit(S,G) to TRUE`
	`  }`
	`}`
	`Here the JoinDesired(S,G) is set to TRUE because, immediate_olist(S,G)`
	`is not NULL, the RPF interface to S is different from the RPF interface`
	`to the RP. Here RP Tree is built by (*,G) Join message`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-6.2 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE & DirectlyConnected(S)
	is set to TRUE & for (*,G) state							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-6.3 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE because, immediate_olist(S,G)
	is not NULL, the RPF interface to S is different from the RPF interface
	to the RP. Here RP Tree is built by (*,*,RP) Join message							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-6.4 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE because, & DirectlyConnected(S)
	is set to TRUE. Here RP Tree is built by (*,*,RP) Join message							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-6.5 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE, DirectlyConnected(S)
	is set to FALSE. RPF_interface(S) is same as RPF_interface(RP).
	Here no RP Tree is built by (*,G) or (*,*,RP) Join, hence
	inherited_olist(S,G,rpt) is NULL.							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-6.6 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE, DirectlyConnected(S)
	is set to FALSE. RPF_interface(S) is same as RPF_interface(RP).
	inherited_olist(S,G,rpt) is not NULL through (*,G) join.
	RPF'(S,G) == RPF'(*,G) and RPF'(S,G) != NULL							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-6.7 **MUST**	RFC 4601 s4.2.2 p29 Setting and Clearing the (S,G) SPTbit							

	Thus, when a packet arrives, the (S,G) SPTbit is updated as follows:
	void
	Update_SPTbit(S,G,iif) {
	  if (iif == RPF_interface(S)
	  AND JoinDesired(S,G) == TRUE
	  AND (DirectlyConnected(S) == TRUE
	  OR RPF_interface(S) != RPF_interface(RP(G))
	  OR inherited_olist(S,G,rpt) == NULL
	  OR ((RPF'(S,G) == RPF'(*,G)) AND
	  (RPF'(S,G) != NULL))
	  OR (I_Am_Assert_Loser(S,G,iif)) {
	  Set SPTbit(S,G) to TRUE
	  }
	}
	Here the JoinDesired(S,G) is set to TRUE, DirectlyConnected(S)
	is set to FALSE. RPF_interface(S) is same as RPF_interface(RP).
	inherited_olist(S,G,rpt) is not NULL through (*,*,RP) join.
	RPF'(S,G) == RPF'(*,G) and RPF'(S,G) != NULL							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.1 **MUST**	RFC 4601 s4.3.1 p30 Sending Hello Messages							
	PIM Hello messages are sent periodically on each PIM-enabled interface. Hello messages must be sent every <Hello-Period> seconds.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.2 **MUST**	RFC 4601 s4.3.1 p30 Sending Hello Messages							
	Hello messages MUST be sent on all active interfaces, including physical point-to-point links, and are multicast to the `ALL-PIM-ROUTERS' group address (`224.0.0.13' for IPv4 and `ff02::d' for IPv6).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-7.3 **MUST**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	When PIM is enabled on an interface or a router first starts,the hello timer of that interface is set to a random value between 0 and Triggered_Hello_Delay.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.4 **MAY**	NEGATIVE RFC 4601 s4.3.1 p31 Sending Hello Messages							
	Note that neighbors will not accept Join/Prune or Assert messages from a router unless they have first heard a Hello message from that router. (Note: This test is for (*,*,RP) join state)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.5 **MAY**	NEGATIVE RFC 4601 s4.3.1 p31 Sending Hello Messages							
	Note that neighbors will not accept Join/Prune or Assert messages from a router unless they have first heard a Hello message from that router. (Note: This test is for (*,G) join state)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.6 **MAY**	NEGATIVE RFC 4601 s4.3.1 p31 Sending Hello Messages							
	Note that neighbors will not accept Join/Prune or Assert messages from a router unless they have first heard a Hello message from that router. (Note: This test is for (S,G) join state)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-7.7 **MUST**	RFC 4601 s4.3.1 p31 Sending Hello Messages RFC 4601 s4.6 p83 PIM Assert Messages							
	Note that neighbors will not accept Join/Prune or Assert messages from a router unless they have first heard a Hello message from that router. AND If a router receives an Assert message from a particular IP source address and it has not seen a PIM Hello message from that source address, then the Assert message SHOULD be discarded without further processing.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.8 **SHOULD**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	The DR_Priority Option SHOULD be included in every Hello message, even if no DR Priority is explicitly configured on that interface.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.9 **SHOULD**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	The DR_Priority Option SHOULD be included in every Hello message, even if no DR Priority is explicitly configured on that interface. This is necessary because priority-based DR election is only enabled when all neighbors on an interface advertise that they are capable of using the DR_Priority Option. The default priority is 1.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.10 **SHOULD**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	The Generation_Identifier (GenID) Option SHOULD be included in all Hello messages							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-7.11 **MUST**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	The GenID option contains a randomly generated 32-bit value that is regenerated each time PIM forwarding is started or restarted on the interface, including when the router itself restarts.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-7.12 **SHOULD**	RFC 4601 s4.3.1 p31 Sending Hello Messages							
	The LAN Prune Delay Option SHOULD be included in all Hello messages sent on multi-access LANs.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.1 **MUST**	RFC 4601 s4.3.2 p33 DR Election							

	The function used for comparing DR "metrics" on interface I is:
	Bool dr_is_better(a,b,I) {
	  if(there is a neighbor n on I for which n.dr_priority_present
	  is false) {
	  return a.primary_ip_address > b.primary_ip_address
	  } else {
	  return (a.dr_priority > b.dr_priority) OR
	  (a.dr_priority == b.dr_priority AND
	  a.primary_ip_address > b.primary_ip_address)
	  }
	}
	Note: If no DR-priority option is specified in a Hello message, the neighbor with the highest IP address is elected as the DR.							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-8.2 **MUST**	RFC 4601 s4.3.2 p33 DR Election							

	The function used for comparing DR "metrics" on interface I is:
	Bool dr_is_better(a,b,I) {
	if(there is a neighbor n on I for which n.dr_priority_present is false) {
	return a.primary_ip_address > b.primary_ip_address
	} else {
	return (a.dr_priority > b.dr_priority) OR
	(a.dr_priority == b.dr_priority AND
	a.primary_ip_address > b.primary_ip_address)
	}
	}
	Note: If DR-priority option is specified in a Hello message. The DR Priority is a 32-bit unsigned number and the numerically larger priority is always preferred.
	(When DUT is elected as DR)							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.3 **MUST**	RFC 4601 s4.3.2 p33 DR Election							

	The function used for comparing DR "metrics" on interface I is:
	Bool dr_is_better(a,b,I) {
	if(there is a neighbor n on I for which n.dr_priority_present is false) {
	return a.primary_ip_address > b.primary_ip_address
	} else {
	return (a.dr_priority > b.dr_priority) OR
	(a.dr_priority == b.dr_priority AND
	a.primary_ip_address > b.primary_ip_address)
	}
	}
	Note: If DR-priority option is specified in a Hello message. The DR Priority is a 32-bit unsigned number and the numerically larger priority is always preferred.
	(When ANVL is elected as DR)							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-8.4 **MUST**	RFC 4601 s4.3.2 p33 DR Election							

	The function used for comparing DR "metrics" on interface I is:
	Bool dr_is_better(a,b,I) {
	if(there is a neighbor n on I for which n.dr_priority_present
	is false) {
	return a.primary_ip_address > b.primary_ip_address
	} else {
	return (a.dr_priority > b.dr_priority) OR
	(a.dr_priority == b.dr_priority AND
	a.primary_ip_address > b.primary_ip_address)
	}
	}
	Note: If DR-priority option is specified in a Hello message, the neighbor with the DR-priority is equal to that of the others then the highest IP address is elected as the DR.
	(When DUT is elected as DR)							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.5 **MUST**	RFC 4601 s4.3.2 p33 DR Election							

	The function used for comparing DR "metrics" on interface I is:
	Bool dr_is_better(a,b,I) {
	if(there is a neighbor n on I for which n.dr_priority_present
	is false) {
	return a.primary_ip_address > b.primary_ip_address
	} else {
	return (a.dr_priority > b.dr_priority) OR
	(a.dr_priority == b.dr_priority AND
	a.primary_ip_address > b.primary_ip_address)
	}
	}
	Note: If DR-priority option is specified in a Hello message, the neighbor with the DR-priority is equal to that of the others then the highest IP address is elected as the DR.
	(When ANVL is elected as DR)							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.6 **MUST**	RFC 4601 s4.3.2 p33 DR Election							
	The Neighbor Liveness Timer (NLT(N,I)) is reset to Hello_Holdtime (from the Hello Holdtime option) whenever a Hello message is received containing a Holdtime option, or to Default_Hello_Holdtime if the Hello message does not contain the Holdtime option.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-8.7 **MUST**	RFC 4601 s4.3.2 p33 DR Election							
	The Neighbor Liveness Timer (NLT(N,I)) is reset to Hello_Holdtime (from the Hello Holdtime option) whenever a Hello message is received containing a Holdtime option, or to Default_Hello_Holdtime if the Hello message does not contain the Holdtime option. (Note: ANVL sends Hello message that contains Holdtime option, from <mcast-router-b>, NLT is set to Hello_Holdtime)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.8 **MAY**	RFC 4601 s4.3.2 p34 DR Election							
	A router's idea of the current DR on an interface can change when a PIM Hello message is received, when a neighbor times out, or when a router's own DR Priority changes. (Note: neighbor time out)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.9 **MUST**	RFC 4601 s4.3.2 p34 DR Election							
	A router's idea of the current DR on an interface can change when a PIM Hello message is received, when a neighbor times out, or when a router's own DR Priority changes. (Note: router's own DR Priority changes)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-8.10 **MAY**	RFC 4601 s4.3.2 p34 DR Election							
	A router's idea of the current DR on an interface can change when a PIM Hello message is received, when a neighbor times out, or when a router's own DR priority changes. If the router becomes the DR or ceases to be the DR, this will normally cause the DR Register state-machine to change state. (Here selection of the new DR to be one with the highest IP address)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-9.1 **MUST**	RFC 4601 s4.3.3 p34 Reducing Prune Propagation Delay on LANs							
	`Just like the DR_Priority option, the information provided in the LAN Prune Delay option is not used unless all neighbors on a link advertise the option. (Note: when lan_delay_enabled is FALSE, both Effective_Propagation_Delay(I), & Effective_Override_Interval(I) return Propagation_delay_default & t_override_default respectively)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-9.2 **MUST**	RFC 4601 s4.3.3 p35 Reducing Prune Propagation Delay on LANs							
	`When all routers on a link are in a position to negotiate a different than default Propagation Delay, the largest value from those advertised by each neighbor is chosen. (Note: for Effective_Propagation_Delay(I) & (*,*,RP) state)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-9.3 **MUST**	RFC 4601 s4.3.3 p35 Reducing Prune Propagation Delay on LANs							
	`When all routers on a link are in a position to negotiate a Propagation Delay different from the default, the largest value from those advertised by each neighbor is chosen. (Note: for Effective_Propagation_Delay(I) & (*,G) state)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-9.4 **MUST**	RFC 4601 s4.3.3 p36 Reducing Prune Propagation Delay on LANs							
	`When all routers on a link are in a position to negotiate a different than default Override Interval, the largest value from those advertised by each neighbor is chosen. (Note: for Effective_Override_Interval(I) & (*,*,RP) state)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-9.5 **MUST**	RFC 4601 s4.3.3 p36 Reducing Prune Propagation Delay on LANs							
	When all routers on a link are in a position to negotiate an Override Interval different from the default, the largest value from those advertised by each neighbor is chosen. (Note: for Effective_Override_Interval(I) & (*,G) state)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-10.1 **MUST**	RFC 4601 s4.4 p38 PIM Register Messages							
	The Designated Router (DR) on a LAN or point-to-point link encapsulates multicast packets from local sources to the RP for the relevant group unless it recently received a Register Stop message for that (S,G) or (*,G) from the RP.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-10.2 **MUST**	NEGATIVE RFC 4601 s4.4 p38 PIM Register Messages							
	The Designated Router (DR) on a LAN or point-to-point link encapsulates multicast packets from local sources to the RP for the relevant group unless it recently received a Register Stop message for that (S,G) or (*,G) from the RP.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-10.3 **MUST**	RFC 4601 s4.4 p38 PIM Register Messages							
	The Designated Router (DR) on a LAN or point-to-point link encapsulates multicast packets from local sources to the RP for the relevant group unless it recently received a Register-Stop message for that (S,G) or (*,G) from the RP.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-10.4	RFC 4601 s4.4 p38 PIM Register Messages							
MUST	`When the DR receives a Register Stop message from the RP, it starts a Register Stop timer to maintain this state. Just before the Register Stop timer expires, the DR sends a Null-Register Message to the RP to allow the RP to refresh the Register Stop information at the DR.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.1	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
MUST	`In Join(J) state if DR receives RegisterStop Message, then it will go to Prune(P) state by removing register tunnel and set Register-Stop Timer.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.2	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
MUST	`In Join(J) state if CouldRegister(S,G) becomes false then it will go to NoInfo(NI) State & remove reg tunnel Here CouldRegister(S,G) -> FALSE is achieved by making I_am_DR(RPF_interface(S))->FALSE`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.3	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
MUST	`In Join(J) state if RP(G) changes, then the DR updates Register tunnel`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-11.4 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Join Pending(JP) state if RegStop timer expires then the DR will go to Join(J) state by adding the register tunnel`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.5 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Join Pending(JP) state if RP changed then the DR will go to Join(J) state by adding the register tunnel and cancel the Register-Stop Timer.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.6 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Join Pending(JP) state if CouldRegister(S,G) becomes false then it will go to NoInfo(NI) State Here CouldRegister(S,G) -> FALSE is achieved by making I_am_DR(RPF_interface(S))->FALSE`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.7 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Join Pending(JP) state if RegStop is received The the DR goes to Prune(P) state and set RegStop timer to randomised RSI - probetime`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-11.8 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Prune(P) state if Register-Stop timer expires then the DR will send Null-Register message`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.9 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Prune(P) state if CouldRegister(S,G) becomes false then it will go to NoInfo(NI) State Here CouldRegister(S,G) -> FALSE is achieved by making I_am_DR(RPF_interface(S))->FALSE`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.10 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In Prune(P) state if RP(G) changes, then the DR goes to Join(J) state and adds register tunnel; cancel Register-Stop Timer`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-11.11 **MUST**	RFC 4601 s4.4.1 p39 Sending Register Messages from the DR							
	`In NoInfo(NI) if CouldRegister(S,G) becomes true then DR will go to Join(J) State, adding register tunnel Here CouldRegister(S,G) -> TRUE is achieved by making I_am_DR(RPF_interface(S))->TRUE`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-11.12 **MUST**	RFC 4601 s4.4.1 p42 Sending Register Messages from the DR							
	`A Register-Stop(*,G) should be treated as a Register-Stop(S,G) for all (S,G) Register state machines`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.1 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	`When an RP receives a Register message, the course of action is decided according to the following pseudocode:`
	 `packet_arrives_on_rp_tunnel( pkt ) {`
	 `  ...`
	 `  if(SPTbit(S,G) OR`
	 `  (SwitchToSptDesired(S,G) AND`
	 `  (inherited_olist(S,G) == NULL))) {`
	 `  send Register-Stop(S,G) to outer.src`
	 `  sentRegisterStop = TRUE;`
	 `  }`
	 `  else {`
	 `  ...`
	 `  }`
	 `  ...`
	 `}`
	 `(Note: A "switch on first packet" policy can be implemented by making SwitchToSptDesired(S,G) return true once a single packet has been received for the source and group.`
	 `If ((inherited_olist(S,G) == NULL) AND (SwitchToSptDesired(S,G) == TRUE)) then RP send Register-Stop(S,G) to outer.src i.e., the DRs address.)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-12.2 **MUST**	NEGATIVE RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	<pre>When an RP receives a Register message, the course of action is
	decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	  if(I_am_RP(G) AND outer.dst == RP(G)) {
	  ...
	  if(!SPTbit(S,G) AND !pkt.NullRegisterBit) {
	  decapsulate and forward the inner packet to
	  inherited_olist(S,G,rpt) # Note (+)
	  }
	  }
	}
	(Note: If (S,G) entry with SPT bit set to TRUE, and received Register
	with Null-Register-Bit set to FALSE then RP don't decapsulate and
	pass the inner packet to the normal forwarding path.)</pre>							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.3 **MUST**	NEGATIVE RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	<pre>When an RP receives a Register message, the course of action is
	decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	  if(I_am_RP(G) AND outer.dst == RP(G)) {
	  ...
	  if(!SPTbit(S,G) AND !pkt.NullRegisterBit) {
	  decapsulate and forward the inner packet to
	  inherited_olist(S,G,rpt) # Note (+)
	  }
	  }
	}
	(Note: If (S,G) entry with SPT bit set to TRUE, and received Register
	with Null-Register-Bit set to TRUE then RP don't decapsulate and
	pass the inner packet to the normal forwarding path.)</pre>							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-12.4 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	When an RP receives a Register message, the course of action is decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	  if(I_am_RP(G) AND outer.dst == RP(G)) {
	  ...
	  if(!SPTbit(S,G) AND !pkt.NullRegisterBit) {
	  decapsulate and forward the inner packet to
	  inherited_olist(S,G,rpt) # Note (+)
	  }
	  }
	}
	If there is no (S,G) entry, i.e. SPTbit set to FALSE and received Register has Null-Register-Bit set to FALSE then RP decapsulate and pass the inner packet to the normal forwarding path for forwarding on the (*,G) tree.							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.5 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	When an RP receives a Register message, the course of action is decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	  ...
	  if(I_am_RP(G) && outer.dst == RP(G)) {
	  ...
	  } else {
	  send Register-Stop(S,G) to outer.src
	  # Note (*)
	  }
	}
	Here it is tested if (I_am_RP(G) -> FALSE) RP sent a Register-Stop Message							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-12.6 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	When an RP receives a Register message, the course of action is decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	...
	if(I_am_RP(G) && outer.dst == RP(G)) {
	...
	} else {
	send Register-Stop(S,G) to outer.src
	# Note (*)
	}
	}
	Here it is tested if (I_am_RP(G) -> FALSE) RP does not forward the data							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.7 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	When an RP receives a Register message, the course of action is decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	...
	if (I_am_RP(G) && outer.dst == RP(G)) {
	...
	} else {
	send Register-Stop(S,G) to outer.src
	# Note (*)
	}
	}
	Here (outer.dst == RP(G))->FALSE							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-12.8 **MUST**	RFC 4601 s4.4.2 p43 Receiving Register Messages at the RP							

	When an RP receives a Register message, the course of action is decided according to the following pseudocode:
	packet_arrives_on_rp_tunnel(pkt) {
	  if(I_am_RP(G) AND outer.dst == RP(G)) {
	  ...
	  if(!SPTbit(S,G) AND !pkt.NullRegisterBit) {
	  decapsulate and forward the inner packet to
	  inherited_olist(S,G,rpt) # Note (+)
	  }
	  }
	}
	If there is no (S,G) entry, i.e. SPTbit set to FALSE and received Register has Null-Register-Bit set to TRUE then RP doesn't decapsulate and pass the inner packet to the normal forwarding path for forwarding on the (*,G) tree.							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.9 **MUST**	RFC 4601 s4.4.2 p44 Receiving Register Messages at the RP							
	When forwarding a packet from the Register Tunnel, the TTL of the original data packet is decremented after it is decapsulated.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.10 **MUST**	NEGATIVE RFC 4601 s4.4.2 p44 Receiving Register Messages at the RP							
	When forwarding a packet from the Register Tunnel, the TTL of the original data packet is decremented after it is decapsulated.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-12.11 **MUST**	RFC 4601 s4.4.2 p44 Receiving Register Messages at the RP							
	The IP ECN bits should be copied from the IP header of the Register packet to the decapsulated packet.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-13.1 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(*,*,RP) message the (*,*,RP) downstream state machine on interface I remains in the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.2 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.3 **MAY**	NEGATIVE RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.4 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Join(J) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I remains in Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-13.5 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Join(J) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I remains in Join state, and the Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When Current value is less than HoldTime from Join/Prune message.)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.6 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Join(J) state by receiving Prune(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending Timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface; otherwise it is set to zero causing it to expire immediately. (Note: the Prune-Pending timer expires immediately)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.7 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Join(J) state by receiving Prune(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface;`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.8 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Join(J) state if the Expiry Timer for the (*,*,RP) downstream state machine on interface I expires. The (*,*,RP) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-13.9 **MUST**	RFC 4601 s4.5.1 p46 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Prune(*,*,RP) message the (*,*,RP) downstream state machine on interface I remains into the Prune-Pending state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.10 **SHOULD**	RFC 4601 s4.5.1 p47 Receiving (*,*,RP) Join/Prune Messages							
	`Note that it is possible to receive a Join(*,*,RP) message for an RP that we do not have information telling us that it is an RP. In the case of (*,*,RP) state, so long as we have a route to the RP, this will not cause a problem, and the transition should still take place.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.11 **MUST**	RFC 4601 s4.5.1 p48 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.12 **MUST**	NEGATIVE RFC 4601 s4.5.1 p48 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-13.13 **MUST**	NEGATIVERFC 4601 s4.5.1 p48 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,*,RP) message the (*,*,RP) downstream state machine on interface I transitions to the Join state. The Expiry Timer is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.14 **MUST**	RFC 4601 s4.5.1 p48 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Expiry Timer for the (*,*,RP) downstream state machine on interface I expires. The (*,*,RP) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-13.15 **MUST**	RFC 4601 s4.5.1 p48 Receiving (*,*,RP) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Prune-Pending Timer for the (*,*,RP) downstream state machine on interface I expires. The (*,*,RP) downstream state machine on interface I transitions to the NoInfo state. A PruneEcho(*,*,RP) is sent onto the subnet connected to interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.1 **MAY**	RFC 4601 s4.5.2 p49 Receiving (*,G) Join/Prune Messages							
	`If the RP in the message does not match RP(G) the Join(*,G) should be silently dropped.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-14.2 **MAY**	RFC 4601 s4.5.2 p49 Receiving (*,G) Join/Prune Messages							
	`If a router has no RP information (e.g. has not recently received a BSR message) then it may choose to accept Join(*,G) and treat the RP in the message as RP(G).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.3 **MUST**	RFC 4601 s4.5.2 p49 Receiving (*,G) Join/Prune Messages							
	`Received Prune(*,G) messages are processed even if the RP in the message does not match RP(G).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.4 **MAY**	RFC 4601 s4.5.2 p49 Receiving (*,G) Join/Prune Messages							
	`If a router has no RP information (e.g. has not recently received a BSR message) then it may choose to accept Prune(*,G) and treat the RP in the message as RP(G).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.5 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(*,G) message the (*,G) downstream state machine on interface I remains in the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-14.6 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.7 **MUST**	NEGATIVE RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.8 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(*,G) message the (*,G) downstream state machine on interface I remains in the NoInfo(NI) state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.9 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I remains in Join state, and the Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune mesage. (Note: When current value is smaller than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-14.10 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I remains in Join state, and the Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is greater than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.11 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I remains in Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.12 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state by receiving Prune(*,G) message The (*,G) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending Timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface; otherwise it is set to zero causing it to expire immediately. (Note: Prune-Pending Timer expires immediately)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.13 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state by receiving Prune(*,G) message the (*,G) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface;`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-14.14 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Join(J) state if the Expiry Timer for the (*,G) downstream state machine on interface I expires. The (*,G) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.15 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Prune(*,G) message the (*,G) downstream state machine on interface I remains into the Prune-Pending state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.16 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.17 **MUST**	NEGATIVE RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-14.18 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state. The Expiry Timer is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is greater than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.19 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Expiry Timer for the (*,G) downstream state machine on interface I expires. The (*,G) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.20 **MUST**	RFC 4601 s4.5.2 p50 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Prune-Pending Timer for the (*,G) downstream state machine on interface I expires. The (*,G) downstream state machine on interface I transitions to the NoInfo state. A PruneEcho(*,G) is sent onto the subnet connected to interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-14.21 **MUST**	RFC 4601 s4.5.2 p52 Receiving (*,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(*,G) message the (*,G) downstream state machine on interface I transitions to the Join state. The Expiry Timer is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is smaller than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-15.1 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(S,G) message the (S,G) downstream state machine on interface I remains in the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.2 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.3 **MUST**	NEGATIVE RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.4 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Join(J) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I remains in Join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-15.5 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	In Join(J) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I remains in Join state, and the Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (When current value is greater than HoldTime from the triggering Join/Prune message)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.6 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	In Join(J) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I remains in Join state, and the Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (When current value is smaller than HoldTime from the triggering Join/Prune message)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.7 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	In Join(J) state by receiving Prune(S,G) message the (S,G) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending Timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface; otherwise it is set to zero causing it to expire immediately. (Note: Prune-Pending timer expires immediately)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-15.8 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Join(J) state by receiving Prune(S,G) message the (S,G) downstream state machine on interface I transitions to the Prune-Pending state. The Prune-Pending timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface; (Note: router has more than one neighbor on that interface)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.9 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Join(J) state if the Expiry Timer for the (S,G) downstream state machine on interface I expires. The (S,G) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.10 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Prune(S,G) message the (S,G) downstream state machine on interface I remains into the Prune-Pending state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.11 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-15.12 **MUST**	NEGATIVE RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state. The Prune-Pending timer is canceled (without triggering an expiry event).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.13 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state. ...The Expiry Timer is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is greater than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.14 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state by receiving Join(S,G) message the (S,G) downstream state machine on interface I transitions to the Join state. ...The Expiry Timer is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is smaller than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-15.15 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Expiry Timer for the (S,G) downstream state machine on interface I expires. The (S,G) downstream state machine on interface I transitions to the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-15.16 **MUST**	RFC 4601 s4.5.3 p54 Receiving (S,G) Join/Prune Messages							
	`In Prune-Pending(PP) state if the Prune-Pending Timer for the (S,G) downstream state machine on interface I expires. The (S,G) downstream state machine on interface I transitions to the NoInfo state. A PruneEcho(S,G) is sent onto the subnet connected to interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.1 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Join(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I remains in the NoInfo state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.2 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I transitions to Prune-Pending(PP) state. ...The Prune-Pending timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface; otherwise it is set to causing it to expire immediately (Note: Here DUT has only one downstream neighbor)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.3 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In NoInfo(NI) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I transitions to Prune-Pending(PP) state. The Prune-Pending timer is started; it is set to the J/P_Override_Interval(I) if the router has more than one neighbor on that interface (Note: Here DUT has more than one downstream neighbor)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-16.4 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Prune-Pending (PP) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I remains in the Prune-Pending(PP) state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.5 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Prune-Pending (PP) state by receiving Join(*,G) message the (S,G,rpt) downstream state machine on interface I transitions to the Prune-Pending-Tmp(PP') state. If the (*,G) message does not contain (S,G,rpt) Join/Prune information the downstream state machine on interface I transitions to NoInfo state`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.6 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Prune-Pending (PP) state by receiving Join(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I transitions to NoInfo state. ET and PPT are canceled.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.7 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Prune-Pending (PP) state if the Prune-Pending Timer for the (S,G,rpt) downstream state machine on interface I expires. The (S,G,rpt) downstream state machine on interface I transitions to the Pruned state`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-16.8 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	In Pruned(P) state by receiving Join(*,G) message the (S,G,rpt) downstream state machine on interface I transitions to PruneTmp state. The end of the compound Join/Prune message is reached. The (S,G,rpt) downstream state machine on interface I transitions to the NoInfo state. ET is canceled. (Note: Here DUT has only one downstream neighbor)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.9 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	In Pruned(P) state by receiving Join(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I transitions to NoInfo state.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.10 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	In Pruned(P) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I remains in Pruned state.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.11 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	In Pruned(P) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I remains in Pruned state. The Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is larger than HoldTime from the triggering Join/Prune message)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-16.12 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Pruned(P) state by receiving Prune(S,G,rpt) message the (S,G,rpt) downstream state machine on interface I remains in Pruned state. The Expiry Timer (ET) is restarted, set to maximum of its current value and the HoldTime from the triggering Join/Prune message. (Note: When current value is smaller than HoldTime from the triggering Join/Prune message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-16.13 **MUST**	RFC 4601 s4.5.4 p58 Receiving (S,G,rpt) Join/Prune Messages							
	`In Pruned(P) state if the Expiry Timer for the (S,G,rpt) downstream state machine on interface I expires. The (S,G,rpt) downstream state machine on interface I transitions to the NoInfo state`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-17.1 **MUST**	RFC 4601 s4.5.5 p63 Sending (*,*,RP) Join/Prune Messages							
	`When the upstream (*,*,RP) state-machine is in NotJoined state, if JoinDesired(*,*,RP) becomes True then the upstream (*,*,RP) state machine transitions to Joined state. Send Join(*,*,RP) to the appropriate upstream neighbor, which is MRIB.next_hop(RP). (Here Join List verified)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-17.2 **MUST**	RFC 4601 s4.5.5 p63 Sending (*,*,RP) Join/Prune Messages							
	`When the upstream (*,*,RP) state-machine is in NotJoined state, if JoinDesired(*,*,RP) becomes True then the upstream (*,*,RP) state machine transitions to Joined state. Send Join(*,*,RP) to the appropriate upstream neighbor, which is MRIB.next_hop(RP). (Note: Here WC and RPT Bit are checked)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-17.3 **MUST**	RFC 4601 s4.5.5 p63 Sending (*,*,RP) Join/Prune Messages							
	`JoinDesired(*,*,RP) becomes False The downstream state for (*,*,RP) has changed so no interface is in immediate_olist(*,*,RP), making JoinDesired(*,*,RP) become False. The upstream (*,*,RP) state machine transitions to NotJoined state. Send Prune(*,*,RP) to the appropriate upstream neighbor, which is MRIB.next_hop(RP). (Here Prune List verified)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-17.4 **MUST**	RFC 4601 s4.5.5 p63 Sending (*,*,RP) Join/Prune Messages							
	`JoinDesired(*,*,RP) becomes False The downstream state for (*,*,RP) has changed so no interface is in immediate_olist(*,*,RP), making JoinDesired(*,*,RP) become False. The upstream (*,*,RP) state machine transitions to NotJoined state. Send Prune(*,*,RP) to the appropriate upstream neighbor, which is MRIB.next_hop(RP). (Note: Here WC and RPT Bit are checked)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-17.5 **MUST**	RFC 4601 s4.5.5 p63 Sending (*,*,RP) Join/Prune Messages							
	`When the upstream (*,*,RP) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(*,*,RP). Send Join(*,*,RP) to the appropriate upstream neighbor, which is MRIB.next_hop(RP). Restart the Join Timer (JT) to expire after t_periodic seconds.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-17.6 **MUST**	RFC 4601 s4.5.5 p64 Sending (*,*,RP) Join/Prune Messages							
	`When the upstream (*,*,RP) state-machine is in Joined state, if the MRIB.next_hop(RP) GenID changes then the upstream (*,*,RP) state machine remains in Joined state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-18.1 **MUST**	RFC 4601 s4.5.6 p66 Sending (*,G) Join/Prune Messages							
	`If a (*,G) Assert occurs on the upstream interface, and this changes this router's idea of the upstream neighbor, it should be prepared to ensure that the Assert winner is aware of downstream routers by sending a Join(*,G) almost immediately.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.2 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	`JoinDesired(*,G) becomes True The downstream state for (*,G) has changed so that at least one interface is in immediate_olist(*,G), making JoinDesired(*,G) become True. The upstream (*,G) state machine transitions to Joined state. Send Join(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). (Here Join List verified)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.3 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	`JoinDesired(*,G) becomes True The downstream state for (*,G) has changed so that at least one interface is in immediate_olist(*,G), making JoinDesired(*,G) become True. The upstream (*,G) state machine transitions to Joined state. Send Join(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). (Note: Here WC and RPT Bit are checked)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.4 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	`JoinDesired(*,G) becomes False The downstream state for (*,G) has changed so no interface is in immediate_olist(*,G), making JoinDesired(*,G) become False. The upstream (*,G) state machine transitions to NotJoined state. Send Prune(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). (Note: Here Prune List verified)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-18.5 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							

	`JoinDesired(*,G) becomes False`
	`The downstream state for (*,G) has changed so no interface is in immediate_olist(*,G), making JoinDesired(*,G) become False. The upstream (*,G) state machine transitions to NotJoined state. Send Prune(*,G) to the appropriate upstream neighbor, which is RPF'(*,G).`
	`(Note: Here WC and RPT Bit are checked)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.6 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	`When the upstream (*,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(*,G). Send Join(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). Restart the Join Timer (JT) to expire after t_periodic seconds.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.7 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	`When the upstream (*,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(*,G). Send Join(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). Restart the Join Timer (JT) to expire after t_periodic seconds.` `(Note: See Join(*,G) to RPF'(*,G), Increase Join Timer to t_joinsuppress)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-18.8 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	When the upstream (*,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(*,G). Send Join(*,G) to the appropriate upstream neighbor, which is RPF'(*,G). Restart the Join Timer (JT) to expire after t_periodic seconds. (Note: See Prune(*,G) to RPF'(*,G), Decrease Join Timer to t_override)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-18.9 **MUST**	RFC 4601 s4.5.6 p67 Sending (*,G) Join/Prune Messages							
	When the upstream (*,G) state-machine is in Joined state, if the RPF'(*,G) GenID changes then the upstream (*,G) state machine remains in Joined state.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.1 **MUST**	RFC 4601 s4.5.7 p71 Sending (S,G) Join/Prune Messages							
	If a (S,G) Assert occurs on the upstream interface, and this changes this router's idea of the upstream neighbor, it should be prepared to ensure that the Assert winner is aware of downstream routers by sending a Join(S,G) almost immediately.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.2 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	When the upstream (S,G) state machine is in NotJoined state, JoinDesired(S,G) becomes True, The downstream state for (S,G) has changed so that at least one interface is in inherited_olist(S,G). (Note: Verify (S,G) Join List contains the Source Address in Join List)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-19.3 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages RFC 4601 s4.9.5.1 p124 Group Set Source List Rules							
	`(S,G) source list entries have the Source-Address set to the address of the source S, the Source-Address Mask-Len set to the full length of the IP address, and both the WC and RPT bits of the Encoded-Source-Address cleared. (Note: Here WC and RPT Bit are checked)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.4 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	`JoinDesired(S,G) becomes False The downstream state for (S,G) has changed so no interface is in inherited_olist(S,G), making JoinDesired(S,G) become False. The upstream (S,G) state machine transitions to NotJoined state. Send Prune(S,G) to the appropriate upstream neighbor, which is RPF'(S,G) (Here Prune List verified)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.5 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	`JoinDesired(S,G) becomes False The downstream state for (S,G) has changed so no interface is in inherited_olist(S,G), making JoinDesired(S,G) become False. The upstream (S,G) state machine transitions to NotJoined state. Send Prune(S,G) to the appropriate upstream neighbor, which is RPF'(S,G) (Note: Here WC and RPT Bit are checked)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.6 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	`When the upstream (S,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(S,G). Send Join(S,G) to the appropriate upstream neighbor, which is RPF'(S,G). Restart the Join Timer (JT) to expire after t_periodic seconds.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-19.7 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	`When the upstream (S,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(S,G). Send Join(S,G) to the appropriate upstream neighbor, which is RPF'(S,G). Restart the Join Timer (JT) to expire after t_periodic seconds. (Note: See Join(S,G) to RPF'(S,G), Increase Join Timer to t_joinsuppress)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.8 **MUST**	RFC 4601 s4.5.7 p72 Sending (S,G) Join/Prune Messages							
	`When the upstream (S,G) state-machine is in Joined state, if the Join Timer (JT) expires, indicating time to send a Join(S,G). Send Join(S,G) to the appropriate upstream neighbor, which is RPF'(S,G). Restart the Join Timer (JT) to expire after t_periodic seconds. (Note: See Prune(S,G) to RPF'(S,G), Decrease Join Timer to t_override)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.9 **MUST**	RFC 4601 s4.5.7 p75 Sending (S,G) Join/Prune Messages							
	`When the upstream (S,G) state-machine is in Joined state, if it sees Prune(*,G) to RPF'(S,G), If the Join Timer is set to expire in more than t_override seconds, reset it so that it expires after t_override seconds.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-19.10 **MUST**	RFC 4601 s4.5.7 p76 Sending (S,G) Join/Prune Messages							
	`When the upstream (S,G) state-machine is in Joined state, if the RPF'(S,G) GenID changes then the upstream (S,G) state machine remains in Joined state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-20.1 **MUST**	RFC 4601 s4.5.9 p78 State Machine for (S,G,rpt) Triggered Messages							
	In "NotPruned" State, if PruneDesired(S,G,rpt)->TRUE the action is to send a Prune(S,G,rpt) to RPF'(S,G,rpt)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-20.2 **MUST**	RFC 4601 s4.5.9 p78 State Machine for (S,G,rpt) Triggered Messages							
	If the router is in the Pruned(S,G,rpt) state, and PruneDesired(S,G,rpt) changes to FALSE, this could be because the router no longer has RPTJoinDesired(G) true, or it now wishes to receive traffic from S again. If it is not the former the action is to send a Join(S,G,rpt) to RPF'(S,G,rpt)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-20.3 **MUST**	RFC 4601 s4.5.9 p78 State Machine for (S,G,rpt) Triggered Messages							
	In "NotPruned" State, When the Override Timer expires, we must send a Join(S,G,rpt) to RPF'(S,G,rpt) to override the Prune message that caused the timer to be running.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-20.4 **MUST**	RFC 4601 s4.5.10 p82 Background: (*,*,RP) and (S,G,rpt) Interaction							
	We first note that reception of a Join(*,*,RP) by itself does not cancel (S,G,rpt) prune state on that interface, whereas receiving a Join(*,G) by itself does cancel (S,G,rpt) prune state on that interface. (Note: for (*,*,RP) Join)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-20.5 **MUST**	RFC 4601 s4.5.10 p82 Background: (*,*,RP) and (S,G,rpt) Interaction							
	`We first note that reception of a Join(*,*,RP) by itself does not cancel (S,G,rpt) prune state on that interface, whereas receiving a Join(*,G) by itself does cancel (S,G,rpt) prune state on that interface. (Note: for (*,G) Join)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-20.6 **MUST**	RFC 4601 s4.5.10 p82 Background: (*,*,RP) and (S,G,rpt) Interaction							
	`Similarly, reception of a Prune(*,G) on an interface with (*,*,RP) join state does not by itself prevent forwarding of G using the (*,*,RP) state; this is because a Prune(*,G) only serves to cancel (*,G) join state.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.1 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`This router has lost an (S,G) assert on interface I. It must not forward packets for G onto interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.2 **MUST**	NEGATIVE: RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`This router has lost an (S,G) assert on interface I. It must not forward packets for G onto interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.3 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`In NoInfo state, if an Inferior Assert is received with RPT bit set CouldAssert(S,G,I) is TRUE, then Send Assert(S,G) Set Assert Timer to (Assert_Time - Assert_Override_Interval) Store self as AssertWinner(S,G,I) Store spt_assert_metric(S,I) as AssertWinnerMetric(S,G,I) (Note: The winning router sends an Assert message containing its own metric to that outgoing interface(State machine))`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.4 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`When in NoInfo state, if an inferior assert is received for (S,G) with the RPT bit cleared and CouldAssert(S,G,I) == TRUE, We transition to the "I am Assert Winner" state`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.5 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`When in NoInfo state, if an assert is received for (S,G) with the RPT bit set(it's a (*,G) assert) and CouldAssert(S,G,I) == TRUE, We Send Assert(S,G).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.6 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	`When in NoInfo state, if an (S,G) data packet comes on Interface I and CouldAssert(S,G,I) == TRUE, We transition to the "I am Assert Winner" state`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.7 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in NoInfo state, if an (S,G) data packet comes on Interface I and CouldAssert(S,G,I) == TRUE, we Send Assert(S,G)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.8 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we remains in "I am Assert Winner" State (Note: for (S,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.9 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we re-send an (S,G) Assert, and restart the Assert Timer (Action A3 below). (Note: for (S,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.10 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we re-send an (S,G) Assert, and restart the Assert Timer (Action A3 below). Set Assert Timer to (Assert_Time - Assert_Override_Interval) (Note: for (S,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.11 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we remains in "I am Assert Winner" State (Note: for (*,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.12 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we re-send an (S,G) Assert, and restart the Assert Timer (Action A3 below). (Note: for (*,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.13 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if we receive an (S,G) assert or (*,G) assert mentioning S that has a worse metric than our own. Whoever sent the assert is in error, and so we re-send an (S,G) Assert, and restart the Assert Timer (Action A3 below). Set Assert Timer to (Assert_Time - Assert_Override_Interval) (Note: for (*,G) assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.14 **MUST**	RFC 4601 s4.6.1 p84 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if We receive an (S,G) assert that has a better metric than our own. We transition to "I am Assert Loser" state							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.15 **MUST**	RFC 4601 s4.6.1 p88 (S,G) Assert Message State Machine							
	When in "I am Assert Winner" State, if CouldAssert(S,G,I) become FALSE, we can no longer perform the actions of the assert winner, and so we transition to NoInfo state and perform actions A4 (below). This includes sending a "canceling assert" with an infinite metric. ... Send AssertCancel(S,G) Delete assert info (AssertWinner(S,G,I) and AssertWinnerMetric(S,G,I) will then return their default values).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.16 **MUST**	RFC 4601 s4.6.1 p88 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, we receive an assert that is better than that of the current assert winner. We stay in Loser state, and perform actions A2 below. ... Store new assert winner as AssertWinner(S,G,I) and assert winner metric as AssertWinnerMetric(S,G,I). Set Assert Timer to Assert_Time							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.17 **MUST**	RFC 4601 s4.6.1 p88 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, we receive an assert from the current assert winner that is better than our own metric for this (S,G) (although the metric may be worse than the winner's previous metric). We stay in Loser state, and perform actions A2 below. ... Store new assert winner as AssertWinner(S,G,I) and assert winner metric as AssertWinnerMetric(S,G,I). Set Assert Timer to Assert_Time							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.18 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, The (S,G) Assert Timer expires. We transition to NoInfo state, deleting the (S,G) assert information (action A5 below). (Note: Set Assert Timer to Assert_Time according to A6)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.19 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, if we receive an assert from the current assert winner that is worse than our own metric for this group (typically the winner's metric became worse or because it is an assert cancel). We transition to NoInfo state, deleting the (S,G) assert information and allowing the normal PIM Join/Prune mechanisms to operate.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.20 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, the (S,G) Assert Timer expires. We transition to NoInfo state, deleting the (S,G) assert information (action A5 below). ... Delete assert info (AssertWinner(S,G,I) and AssertWinnerMetric(S,G,I) will then return their default values).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.21 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, ... we receive a Hello message from the current winner reporting a different GenID from the one it previously reported. This indicates that the current winner's interface or router has gone down (and may have come back up), and so we must assume it no longer knows it was the winner. We transition to the NoInfo state, deleting this (S,G) assert information (action A5 below).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.22 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	When in "I am Assert Loser" State, my_assert_metric(S,G,I) has changed so that now my assert metric for (S,G) is better than the metric we have stored for current assert winner. This might happen the underlying routing metric changes, or when CouldAssert(S,G,I) becomes true; for example, when SPTbit(S,G) becomes true. We transition to NoInfo state, delete this (S,G) assert state (action A5 below), and allow the normal PIM Join/Prune mechanisms to operate. (Note: underlying routing metric changed)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-21.23 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, interface I used to be the RPF interface for S, and now it is not. We transition to NoInfo state, deleting this (S,G) assert state (action A5 below). ... Delete assert info (AssertWinner(S,G,I) and AssertWinnerMetric(S,G,I) will then return their default values).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-21.24 **MUST**	RFC 4601 s4.6.1 p89 (S,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, we receive a Join(S,G) that has the Upstream Neighbor Address field set to my primary IP address on interface I. The action is to transition to NoInfo state, and delete this (S,G) assert state (action A5 below), and allow the normal PIM Join/Prune mechanisms to operate.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.1 **MUST**	RFC 4601 s4.6.2 p91 (*,G) Assert Message State Machine							
	`This router has lost an (*,G) assert on interface I. It must not forward packets for G onto interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.2 **MUST**	NEGATIVE: RFC 4601 s4.6.2 p91 (*,G) Assert Message State Machine							
	`This router has lost an (*,G) assert on interface I. It must not forward packets for G onto interface I.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-22.3 **MUST**	RFC 4601 s4.6.2 p92 (*,G) Assert Message State Machine							
	`The winning router send Assert(*,G) Set Assert Timer to (Assert_Time - Assert_Override_Interval)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.4 **MUST**	RFC 4601 s4.6.2 p94 (*,G) Assert Message State Machine							
	`When in NoInfo state, if A data packet destined for G arrives on interface I, AND CouldAssert(*,G,I)==TRUE ... we transition to the "I am Assert Winner" state, and perform Actions A1 (below). ... Send Assert(*,G) Set Assert Timer to (Assert_Time - Assert_Override_Interval) Store self as AssertWinner(*,G,I). Store rpt_assert_metric(G,I) as AssertWinnerMetric(*,G,I). (Note: only state transition is tested)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.5 **MUST**	RFC 4601 s4.6.2 p94 (*,G) Assert Message State Machine							
	`When in NoInfo state, if A data packet destined for G arrives on interface I, AND CouldAssert(*,G,I)==TRUE ... we transition to the "I am Assert Winner" state, and perform Actions A1 (below). ... Send Assert(*,G) Set Assert Timer to (Assert_Time - Assert_Override_Interval) Store self as AssertWinner(*,G,I). Store rpt_assert_metric(G,I) as AssertWinnerMetric(*,G,I). (Note: sends Assert (*,G))`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.6 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	`When in "I am Assert Winner" State, The (*,G) Assert Timer expires. As we're in the Winner state, then we must still have (*,G) forwarding state that is actively being kept alive. To prevent unnecessary thrashing of the forwarder and periodic flooding of duplicate packets, we resend the (*,G) Assert and restart the Assert Timer (Actions A3 below). (Note: Set Assert Timer to (Assert_Time - Assert_Override_Interval) according to A1)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-22.7 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	When in "I am Assert Winner" State, The (*,G) Assert Timer expires. The (*,G) Assert Timer expires. As we're in the Winner state, then we must still have (*,G) forwarding state that is actively being kept alive. To prevent unnecessary thrashing of the forwarder and periodic flooding of duplicate packets, we re-send the (*,G) Assert, and restart the Assert Timer (Action A3 below). (Note: we must still have (*,G) forwarding state)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.8 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	When in "I am Assert Winner" State, We receive a (*,G) assert that has a worse metric than our own. Whoever sent the assert has lost, and so we re-send a (*,G) Assert, and restart the Assert Timer (Action A3 below). ... Send Assert(*,G) Set Assert Timer to (Assert_Time - Assert_Override_Interval) (Note: Here check that RPT bit is set for the Assert sent by Assert Winner)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.9 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	When in "I am Assert Winner" State, we receive a (*,G) assert that has a better metric than our own. We transition to "I am Assert Loser" state and perform actions A2 (below).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.10 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	When in "I am Assert Winner" State,if our (*,G) forwarding state or RPF interface changed so as to make CouldAssert(*,G,I) become false. We can no longer perform the actions of the assert winner, and so we transition to NoInfo state and perform actions A4 (below). ... Send AssertCancel(*,G) Delete assert info (AssertWinner(*,G,I) and AssertWinnerMetric(*,G,I) will then return their default values). (Note: send AssertCancel(*,G))							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-22.11 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, We receive a (*,G) assert that is better than that of the current assert winner. We stay in Loser state, and perform actions A2 below.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.12 **MUST**	RFC 4601 s4.6.2 p95 (*,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, We receive a (*,G) assert from the current assert winner that is better than our own metric for this group (although the metric may be worse than the winner's previous metric). We stay in Loser state, and perform actions A2 below.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.13 **MUST**	RFC 4601 s4.6.2 p96 (*,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, We receive an assert from the current assert winner that is worse than our own metric for this group (typically because the winner's metric became worse or is now an assert cancel). We transition to NoInfo state, delete this (*,G) assert state (action A5), and allow the normal PIM Join/Prune mechanisms to operate.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.14 **MUST**	RFC 4601 s4.6.2 p96 (*,G) Assert Message State Machine							
	`When in "I am Assert Loser" State, The (*,G) Assert Timer expires. We transition to NoInfo state and delete this (*,G) assert info (action A5). ... Delete assert info (AssertWinner(*,G,I) and AssertWinnerMetric(*,G,I) will then return their default values). (Note: transition to NoInfo state)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-22.15 **MUST**	RFC 4601 s4.6.2 p96 (*,G) Assert Message State Machine							
	When in "I am Assert Loser" State, we receive a Hello message from the current winner reporting a different GenID from the one it previously reported. This indicates that the current winner's interface or router has gone down (and may have come back up), and so we must assume it no longer knows it was the winner. We transition to the NoInfo state, deleting the (*,G) assert information (action A5).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.16 **MUST**	RFC 4601 s4.6.2 p96 (*,G) Assert Message State Machine							
	When in "I am Assert Loser" State, My routing metric, rpt_assert_metric(G,I), has changed so that now my assert metric for (*,G) is better than the metric we have stored for current assert winner. We transition to NoInfo state, and delete this (*,G) assert state (action A5), and allow the normal PIM Join/Prune mechanisms to operate.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.17 **MUST**	RFC 4601 s4.6.2 p97 (*,G) Assert Message State Machine							
	When in "I am Assert Loser" State, Interface I used to be the RPF interface for RP(G), and now it is not. We transition to NoInfo state, and delete this (*,G) assert state (action A5).							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-22.18 **MUST**	RFC 4601 s4.6.2 p97 (*,G) Assert Message State Machine							
	When in "I am Assert Loser" State, We receive a Join(*,G) or a Join(*,*,RP(G)) that has the Upstream Neighbor Address field set to my primary IP address on interface I. The action is to transition to NoInfo state, and delete this (*,G) assert state (action A5), and allow the normal PIM Join/Prune mechanisms to operate. (Note: transition to NoInfo state for Join(*,G))							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-22.19 **MUST**	RFC 4601 s4.6.2 p97 (*,G) Assert Message State Machine							
	When in "I am Assert Loser" State, We receive a Join(*,G) or a Join(*,*,RP(G)) that has the Upstream Neighbor Address field set to my primary IP address on interface I. The action is to transition to NoInfo state, and delete this (*,G) assert state (action A5), and allow the normal PIM Join/Prune mechanisms to operate. (Note: transition to NoInfo state for Join(*,*,RG(G)))							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-23.1 **MUST**	RFC 4601 s4.6.3 p98 Assert Metrics							
	If all fields are equal, the primary IP address of the router that sourced the Assert message is used as a tie-breaker, with the highest IP address winning. (Note: This is for (*,G) Assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-23.2 **MUST**	RFC 4601 s4.6.3 p98 Assert Metrics							
	If all fields are equal, the primary IP address of the router that sourced the Assert message is used as a tie-breaker, with the highest IP address winning. (Note: This is for (S,G) Assert)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-24.1 **MAY**	RFC 4601 s4.7.1 p105 Group-to-RP Mapping							
	Note that if the set of possible group-range-to-RP mappings changes, each router will need to check whether any existing groups are affected. This may, for example, cause a DR or acting DR to re-join a group, or cause it to re-start register encapsulation to the new RP. (Note: This is done for (*,*,RP) Join)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-24.2 **MAY**	RFC 4601 s4.7.1 p105 Group-to-RP Mapping							
	Note that if the set of possible group-range-to-RP mappings changes, each router will need to check whether any existing groups are affected. This may, for example, cause a DR or acting DR to re-join a group, or cause it to re-start register encapsulation to the new RP. (Note: This is done for (*,G) Join)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-25.1 **MUST**	RFC 4601 s4.8 p106 Source-Specific Multicast							
	A range of multicast addresses, currently 232.0.0.0/8 in IPv4 and FF3x::/32 for IPv6, is reserved for SSM, and the choice of semantics is determined by the multicast group address in both data packets and PIM messages. ((*,G) Join Message with group address is in SSM range)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-25.2 **MUST**	RFC 4601 s4.8 p106 Source-Specific Multicast							
	A range of multicast addresses, currently 232.0.0.0/8 in IPv4 and FF3x::/32 for IPv6, is reserved for SSM, and the choice of semantics is determined by the multicast group address in both data packets and PIM messages. ((S,G) Join Message with group address is in SSM range)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-26.1 **MUST**	RFC 4601 s4.8.1 p106 Protocol Modifications for SSM Destination Addresses							
	A router MUST NOT send a Register message for any packet that is destined to an SSM address.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-26.2 **MUST**	RFC 4601 s4.8.1 p106 Protocol Modifications for SSM Destination Addresses							
	`A router acting as an RP MUST NOT forward any Register-encapsulated packet that has an SSM destination address.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-26.3 **SHOULD**	RFC 4601 s4.8.1 p107 Protocol Modifications for SSM Destination Addresses							
	`A router MAY be configured to advertise itself as a Candidate RP for an SSM address. If so, it SHOULD respond with a Register-Stop message to any Register message containing a packet destined for an SSM address.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-27.1 **MUST**	RFC 4601 s4.8.2 p108 PIM-SSM-Only Routers							

	`Additionally, the Packet forwarding rules of Section 4.2 can be simplified in a PIM-SSM-only router:`
	`If (iif == RPF_interface(S) AND UpstreamJPState(S,G) == Joined ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if(iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`}`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-27.2 **MUST**	RFC 4601 s4.8.2 p108 PIM-SSM-Only Routers							

	`Additionally, the Packet forwarding rules of Section 4.2 can be simplified in a PIM-SSM-only router:`
	`if (iif == RPF_interface(S) AND UpstreamJPState(S,G) == Joined ) {`
	`  oiflist = inherited_olist(S,G)`
	`} else if( iif is in inherited_olist(S,G) ) {`
	`  send Assert(S,G) on iif`
	`}`
	`oiflist = oiflist (-) iif`
	`forward packet on all interfaces in oiflist`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-28.1 **MUST**	RFC 4601 s4.9 p108 PIM Packet Formats							
	`All PIM control messages have IP protocol number 103.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-28.2 **MUST**	RFC 4601 s4.9 p109 PIM Packet Formats							
	`Set to zero on transmission.  Ignored upon receipt.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-28.3 **MUST**	RFC 4601 s4.9 p109 PIM Packet Formats							
	`The checksum is a standard IP checksum,i.e. the 16-bit one's Complement of the one's complement sum of the entire PIM message, excluding the "Multicast data packet" section of the Register message.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-28.4 **MUST**	RFC 4601 s4.9 p110 PIM Packet Formats							
	`If a message is received with an unrecognized PIM Ver or Type field or a message's destination does not correspond to the table above, it MUST be discarded and an error message SHOULD be logged to the administrator in a rate limited manner. (Note: wrong Type field, DUT discards packet)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-29.1 **MUST**	RFC 4601 s4.9.1 p111 Encoded Source and Group Address Formats							
	`If the message is sent for a single group then the Mask length must equal the address length in bits for the given Address Family and Encoding Type. (e.g. 32 for IPv4 native encoding, 128 for IPv6 native encoding).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.2 **MUST**	RFC 4601 s4.9.1 p111 Encoded Source and Group Address Formats							

	`[B]idirectional PIM`
	`Indicates the group range should use Bidirectional PIM [13].`
	`For PIM-SM defined in this specification, this bit MUST be zero.`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.3 **MUST**	RFC 4601 s4.9.1 p111 Encoded Source and Group Address Formats							

	`Admin Scope [Z]one indicates the group range is an admin scope zone.`
	`This is used in the Bootstrap Router Mechanism [11] only. For all`
	`other purposes, this bit is set to zero and ignored on receipt.`
	`(Here we are considering Non-BSR message)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.4 **MUST**	NEGATIVE RFC 4601 s4.9.1 p111 Encoded Source and Group Address Formats							

	`Admin Scope [Z]one indicates the group range is an admin scope zone.`
	`This is used in the Bootstrap Router Mechanism [11] only. For all`
	`other purposes, this bit is set to zero and ignored on receipt.`
	`(Here we are considering Non-BSR message)`							

	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-29.5 **MUST**	RFC 4601 s4.9.1 p112 Encoded Source and Group Address Formats							
	The Sparse bit is a 1 bit value, set to 1 for PIM-SM.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.6 **MUST**	RFC 4601 s4.9.1 p112 Encoded Source and Group Address Formats							
	The WC(or WildCard) bit is a 1 bit value for use with PIM Join/Prune messages. (S,G) source list entries have the Source-Address set to the address of the source S, the Source-Address Mask-Len set to the full length of the IP address and have both the WC and RPT bits of the Encoded-Source-Address cleared. (Note: check the WC bit & RPT bit)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.7 **MUST**	RFC 4601 s4.9.1 p112 Encoded Source and Group Address Formats							
	The RPT (or Rendezvous Point Tree) bit is a 1 bit value for use with PIM Join/Prune messages (see Section 4.9.5.1). If the WC bit is 1, the RPT bit MUST be 1.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-29.8 **MUST**	NEGATIVERFC 4601 s4.9.1 p112 Encoded Source and Group Address Formats							
	The RPT (or Rendezvous Point Tree) bit is a 1 bit value for use with PIM Join/Prune messages (see Section 4.9.5.1). If the WC bit is 1, the RPT bit MUST be 1.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-30.1 **SHOULD**	RFC 4601 s4.9.2 p114 Hello Message Format							
	`Hello messages with a Holdtime value set to `0' are also sent by a router on an interface about to go down or changing IP address (see Section 4.3.1). These are effectively goodbye messages and the receiving routers should immediately time out the neighbor information for the sender. (Here the testing is done on whether DUT correctly times out a neighbor)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-30.2 **MUST**	RFC 4601 s4.9.2 p114 Hello Message Format							
	`Hello messages with a Holdtime value set to `0' are also sent by a router on an interface about to go down or changing IP address (see Section 4.3.1). These are effectively goodbye messages and the receiving routers should immediately time out the neighbor information for the sender. (Note: change of IP address)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-30.3 **MUST**	RFC 4601 s4.9.2 p114 Hello Message Format							
	`Hello messages with a Holdtime value set to `0' are also sent by a router on an interface about to go down or changing IP address (see Section 4.3.1). These are effectively goodbye messages and the receiving routers should immediately time out the neighbor information for the sender. (Note: interface goes down)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-31.1 **MUST**	RFC 4601 s4.9.3 p117 Register Message Format							
	`Note that in order to reduce encapsulation overhead, the checksum for Registers is done only on the first 8 bytes of the packet, including the PIM header and the next 4 bytes, excluding the data packet portion.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-31.2	RFC 4601 s4.9.3 p117 Register Message Format							
MUST	`If the router is a DR for a source that it is directly connected to, it sets the B bit to 0`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-32.1	RFC 4601 s4.9.4 p119 Register-Stop Message Format							
MUST	`For Register-Stops, the Mask Len field contains full address length * 8 (e.g. 32 for IPv4 native encoding), if the message is sent for a single group`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-33.1	RFC 4601 s4.9.5 p122 Join/Prune Message Format							
MUST	`Within one PIM Join/Prune message, all the Multicast Group Addresses, Joined Source addresses and Pruned Source addresses MUST be of the same address family.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-34.1	RFC 4601 s4.9.5.1 p122 Group Set Source List Rules							
MUST	`The wildcard group set is represented by the entire multicast range - the beginning of the multicast address range in the group address field and the prefix length of the multicast address range in the mask length field of the Multicast Group Address, e.g. 224.0.0.0/4 for IPv4 or ff00::/8 for IPv6. (This test is for IPv4)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-34.2	RFC 4601 s4.9.5.1 p123 Group Set Source List Rules							
MUST	`(*,G) source list entries have the Source-Address set to the address of the RP for group G, the Source-Address Mask-Len set to the full length of the IP address and have both the WC and RPT bits of the Encoded-Source-Address set.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-34.3	RFC 4601 s4.9.5.1 p124 Group Set Source List Rules							
MUST	`(S,G) source list entries have the Source-Address set to the address of the source S, the Source-Address Mask-Len set to the full length of the IP address and have both the WC and RPT bits of the Encoded-Source-Address cleared.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.1	RFC 4601 s4.9.6 p127 Assert Message Format							
MUST	`RPT-bit is a 1 bit value. The RPT-bit is set to 1 for Assert(*,G) messages and 0 for Assert(S,G) messages. (Note: for (*,G) Asserts)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.2	RFC 4601 s4.9.6 p128 Assert Message Format							
MUST	`Source specific asserts are sent by routers forwarding a specific source on the shortest-path tree(SPT bit is TRUE). (S,G) Asserts have the Group-Address field set to the group G and Source-Address field set to source S`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-35.3 **MUST**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`RPT-bit is a 1 bit value. The RPT-bit is set to 1 for Assert(*,G) messages and 0 for Assert(S,G) messages. (Note: for (S,G) Assert)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.4 **MUST**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`Group specific asserts are sent by routers forwarding data for the group and source(s) under contention on the shared tree. (*,G) Asserts have the Group-Address field set to the group G`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.5 **MAY**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`For data-triggered Asserts the Source-Address field MAY be set to the IP source address of the data packet that triggered the Assert and is set to zero otherwise.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.6 **MUST**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`For data-triggered Asserts the Source-Address field MAY be set to the IP source address of the data packet that triggered the Assert and is set to zero otherwise. The RPT-bit is set to 1, the Metric-Preference is set to MRIB.pref(RP(G)) and the Metric is set to MRIB.metric(RP(G)). (Note: for Source-Address field & Metric-Preference)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-35.7 **MUST**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`For data-triggered Asserts the Source-Address field MAY be set to the IP source address of the data packet that triggered the Assert and is set to zero otherwise. The RPT-bit is set to 1, the Metric-Preference is set to MRIB.pref(RP(G)) and the Metric is set to MRIB.metric(RP(G)). (Note: for Source-Address field & Metric)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-35.8 **MUST**	RFC 4601 s4.9.6 p128 Assert Message Format							
	`For data-triggered Asserts the Source-Address field MAY be set to the IP source address of the data packet that triggered the Assert and is set to zero otherwise. The RPT-bit is set to 1, the Metric-Preference is set to MRIB.pref(RP(G)) and the Metric is set to MRIB.metric(RP(G)). (Note: for (*,G) Assert message RPT-bit)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-36.1 **MUST**	RFC 4601 s4.11 p130 Timer Values							
	`Hello Timer (HT(I)). Periodic interval for Hello messages.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-36.2 **MUST**	RFC 4601 s4.11 p132 Timer Values							
	`Assert Timer (AT(*,G,I), AT(S,G,I)). This timer is used for period after last assert before assert state is timed out. Default: 180 secs.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-36.3 **MUST**	RFC 4601 s4.11 p133 Timer Values							
	`Upstream Join Timer (JT(*,*,RP), JT(*,G), JT(S,G)). This timer is used for period between Join/Prune messages. Default: 60 seconds`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-36.4 **MUST**	RFC 4601 s4.11 p133 Timer Values							
	`Upstream Join Timer (JT(*,*,RP), JT(*,G), JT(S,G)). Suppression period when someone else sends a J/P message so we don't need to do so. Value: rand(1.1 * t_periodic, 1.4 * t_periodic) when Suppression_Enabled(I) is true, 0 otherwise.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-36.5 **MUST**	RFC 4601 s4.11 p133 Timer Values							
	`Upstream Join Timer (JT(*,*,RP), JT(*,G), JT(S,G)). This timer is used for period between Join/Prune messages (Here JT(*,*,RP)) is tested)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-36.6 **MUST**	RFC 4601 s4.11 p133 Timer Values							
	`Upstream Join Timer (JT(*,*,RP), JT(*,G), JT(S,G)). This timer is used for period between Join/Prune messages (Here JT(S,G)) is tested`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-36.7	RFC 4601 s4.11 p134 Timer Values							
MUST	`Keepalive Timer (KAT(S,G)). This timer is the Period after last (S,G) data packet during which (S,G) Join state will be maintained even in the absence of (S,G) Join messages. Default : 210 seconds.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: FAIL	Ubuntu 18.04: FAIL				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.1	draft-ietf-pim-sm-bsr-12.txt s1.2 p7 Protocol Overview							
MUST	`BSMs are originated periodically to ensure consistency after failure restoration.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.2	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
MUST	`If Bootstrap Timer expires, and current state is `P-BSR', the router goes to E-BSR state and after receiving a non-preferred BSM, it remains in the E-BSR state and originates a BSM that contains the BSR priority value of the included BSR & the address of the bootstrap router for the domain.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.3	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
MUST	`In E-BSR state and after receiving a preferred BSM, it goes to the C-BSR state & forward BSM; store RP-Set; set Bootstrap timer to BS_Timeout.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.4 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
	`In P-BSR state and after receiving a preferred BSM, it goes to the C-BSR state & forward BSM; store RP-Set; set Bootstrap timer to BS_Timeout.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.5 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
	`In P-BSR state and after receiving a non-preferred BSM, it remains in the P-BSR state & forward BSM`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.6 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
	`In C-BSR state and after receiving a preferred BSM, it remains in the C-BSR state & forward BSM; store RP-Set; set bootstrap timer to BS_Timeout`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.7 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
	`In C-BSR state and after receiving a preferred BSM, it remains in the C-BSR state & forward BSM; store RP-Set; set bootstrap timer to BS_Timeout (Note: A Bootstrap message is also preferred if it is from the current BSR with a lower weight than the previous BSM it sent, provided that if the router is a Candidate BSR the current BSR still has a weight higher or equal than the router itself.)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.8	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
MUST	In C-BSR state and after receiving a non-preferred BSM, it goes to the P-BSR state & forward BSM; set bootstrap timer to <BS_Rand_Override> (Note:A Bootstrap message is received from the elected BSR, but the BSR Priority field in the received message has changed, so that now the currently elected BSR has lower weight that the router itself.)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.9	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
MUST	In C-BSR state when bootstrap timer expires, it goes to the P-BSR state & set bootstrap timer to BS_Rand_Override							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.10	draft-ietf-pim-sm-bsr-12.txt s3.1.1 p11 Per-Scope-Zone Candidate-BSR State Machine							
MUST	In E-BSR state if the BS Timer expires the BSR originates BSM and set BS Timer to BS_Period							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.11	draft-ietf-pim-sm-bsr-12.txt s3.1.2 p13 Per-Scope-Zone State Machine for Non-Candidate-BSR Routers							
MUST	If the included BSR is not preferred over, and not equal to, the currently active BSR If the Bootstrap Timer has expired and the receiving router is not a C-BSR, the Bootstrap message is then forwarded (Note: Per-Scope-Zone State-machine for Non-Candidate-BSR Routers)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.12 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.1.2 p13 Per-Scope-Zone State Machine for Non-Candidate-BSR Routers							
	`The router knows the identity of the current BSR, and is using the RP-Set provided by that BSR. Only bootsrap messages from that BSR or from a C-BSR with higher weight than the current BSR will be accepted`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.13 **MUST**	NEGATIVE draft-ietf-pim-sm-bsr-12.txt s3.1.2 p13 Per-Scope-Zone State Machine for Non-Candidate-BSR Routers							
	`The router knows the identity of the current BSR, and is using the RP-Set provided by that BSR. Only bootsrap messages from that BSR or from a C-BSR with higher weight than the current BSR will be accepted`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.14 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.2 p19 Sending Candidate-RP-Advertisement Messages							
	`Every C-RP periodically unicasts a C-RP-Adv to the BSR... (Note: Here the unicast test is performed)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.15 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.2 p19 Sending Candidate-RP-Advertisemnt Message							
	`Every C-RP periodically unicasts a C-RP-Adv to the BSR... (Note: Here the periodic test is performed)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.16 **SHOULD**	draft-ietf-pim-sm-bsr-12.txt s3.2 p19 Sending Candidate-RP-Advertisement Messages							
	C-RPs should by default send C-RP-Adv messages withthe Priority field set to 192.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.17 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.2 p19 Sending Candidate-RP-Advertisement Messages							
	If the C-RP is a ZBR for an admin scope zone, then the Admin Scope Zone bit MUST be set in the C-RP-Adv messages it sends for that scope zone; otherwise this bit MUST NOT be set. (Note: Admin Scope Zone bit is unset)							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.18 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.3 p21 Creating the RP-Set at the BSR							
	For each RP-address, the "RP-Holdtime" field is set to the Holdtime from the C-RP-Set, subject to the constraint that it MUST be larger than BS_Period and SHOULD be larger than 2.5 times BS_Period to allow for some Bootstrap messages getting lost.							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.19 **SHOULD**	draft-ietf-pim-sm-bsr-12.txt s3.3 p21 Creating the RP-Set at the BSR							
	For each RP-address, the "RP-Holdtime" field is set to the Holdtime from the C-RP-Set, subject to the constraint that it MUST be larger than BS_Period and SHOULD be larger than 2.5 times BS_Period to allow for some Bootstrap messages getting lost. (Note: Here we test the SHOULD part							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.20	draft-ietf-pim-sm-bsr-12.txt s3.3 p21 Creating the RP-Set at the BSR							
MUST	`There MUST however be a minimum of BS_Min_Interval between each time a BSM is sent.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.21	draft-ietf-pim-sm-bsr-12.txt s3.4 p23 Forwarding Bootstrap Messages							
MUST	`One is that a bootstrap message is not forwarded if its No-Forward bit is set, ...`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.22	draft-ietf-pim-sm-bsr-12.txt s3.4 p23 Forwarding Bootstrap Messages							
MUST	`When a Bootstrap message is forwarded, it is forwarded out of every multicast-capable interface which has PIM neighbors (including the one over which the message was received).`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.23	draft-ietf-pim-sm-bsr-12.txt s3.5 p24 Bootstrap Messages to New and Rebooting Routers							
MAY	`one router on the LAN sends a stored copy of the Bootstrap message for each admin scope zone to the new or rebooting router...This message SHOULD be sent as a No-Forward Bootstrap message ... For backwards compatibility, this message MAY instead or in addition be sent as a Unicast Bootstrap message,...` `(Note: Here ANVL checks that whether the Bootstrap MSG send by DUT has Multicast or Unicast destination. If the destination is Multicast then it should be No-Forward Bootstrap message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.24 **MUST**	NEGATIVE draft-ietf-pim-sm-bsr-12.txt s3.5 p24 Bootstrap Messages to New and Rebooting Routers RFC4601 s4.9, p110 PIM Packet Formats							
	`To allow new or rebooting routers to learn the RP-Set quickly, when a Hello message is received from a new neighbor, or a Hello message with a new GenID is received from an existing neighbor, one router on the LAN sends a stored copy of the Bootstrap message for each admin scope zone to the new or rebooting router. NOTE: <CASE-1> Sending PIM Hello MSG with Unrecognized Version field <CASE-2> Sending PIM Hello MSG with incorrect checksum`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.26 **MUST**	draft-ietf-pim-sm-bsr-12.txt s4 p25 Message Formats							
	`Usually, Bootstrap messages are multicast with TTL 1 to the ALL-PIM-ROUTERS group, ... (Note: Here DUT originates the Bootstrap Message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.27 **MUST**	draft-ietf-pim-sm-bsr-12.txt s4 p25 Message Formats							
	`Usually, Bootstrap messages are multicast with TTL 1 to the ALL-PIM-ROUTERS group, ... (Note: Here DUT forwards the Bootstrap Message)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.28 **MUST**	draft-ietf-pim-sm-bsr-12.txt s4 p25 Message Formats							
	`Usually, Bootstrap messages are multicast with TTL 1 to the ALL-PIM-ROUTERS group, ... (Note: here we check IP TTL value)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-41.29 **MUST**	draft-ietf-pim-sm-bsr-12.txt s4 p25 Message Formats							
	`Usually, Bootstrap messages are multicast with TTL 1 to the ALL-PIM-ROUTERS group, but in some circumstances (described in section 3.5.2) Bootstrap messages are unicast to a specific PIM neighbor. (Note: here we check IP TTL value for forwarded BSM)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.30 **MAY**	draft-ietf-pim-sm-bsr-12.txt s4.1 p28 Bootstrap Message Format							
	`The length (in bits) of the mask to use in the hash function. For IPv4 we recommend a value of 30.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-41.31 **MUST**	draft-ietf-pim-sm-bsr-12.txt s4.2 p32 Candidate-RP-Advertisement Message Format							
	`C-RPs MUST NOT send C-RP-Adv messages with a Prefix Count of `0'.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 18.04: unpredict	Ubuntu 18.04: unpredict				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-42.1 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.6 p25 Receiving and Using the RP-Set							
	`If a mapping is not already part of the RP-Set, it is added to the RP-Set and the associated Group-to-RP mapping Expiry Timer (GET) is initialized to the holdtime from the Bootstrap message. Its priority is set to the Priority from the Bootstrap message.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				

	Release 7.2.1	Release 7.3	Release 7.5.1	Release 8.0	Release x.x.x	Release x.x.x	Release x.x.x	Release x.x.x
PIM-SM-42.2 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.6 p25 Receiving and Using the RP-Set							
	`If a mapping is already part of the RP-Set, it is updated with the Priority from the Bootstrap message and its associated GET is reset to the holdtime from the Bootstrap message.`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-42.3 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.6 p25 Receiving and Using the RP-Set							
	`If a mapping is not already part of the RP-Set, it is added to the RP-Set and the associated Group-to-RP mapping Expiry Timer (GET) is initialized to the holdtime from the Bootstrap message. Its priority is set to the Priority from the Bootstrap message. (Note: This test is for rp-priority)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				
PIM-SM-42.4 **MUST**	draft-ietf-pim-sm-bsr-12.txt s3.6 p25 Receiving and Using the RP-Set							
	`If a mapping is already part of the RP-Set, it is updated with the Priority from the Bootstrap message and its associated GET is reset to the holdtime from the Bootstrap message. (Note: This test is for rp-priority)`							
	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested	Free BSD 10.3 untested				
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 18.04: pass	Ubuntu 18.04: pass				
	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested	Free BSD 12.0 untested				