	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
Type	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR
Commit ID	36a7e78	30283fd	5dff4ec	7a377a1	85f25d8	c8c2427	5a80b8c	10d4945
Commit Date	2017-11-08	2017-11-08	2018-01-09	2018-03-12	2018-07-05	2018-10-08	2019-02-24	2019-03-01
ANVL-RIPNG-1.1 **MUST**	ANVL, setup verification							
	`Setup verification tests` `DUT sends unsolicited RIPng response.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-1.2 **MUST**	ANVL, setup verification							
	`Setup verification tests` `RIPng process responds to Unicast Request Message at UDP Port 521.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-1.3 **MUST**	ANVL, setup verification							

	`Setup verification tests`
	`Once the entry has been validated, update the metric by adding`
	`the cost of the network on which the message arrived.`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-1.4 **MUST**	ANVL, setup verification							
	`Setup verification tests` `DUT forwards the packet according to routing table entry.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-1.6 **MUST**	ANVL, setup verification							

	`Setup verification tests`
	`When the number of RTEs do not fit in one RIPng Unsolicited Update`
	`then split the RTEs across multiple IPv6 fragments`							

	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: FAIL	FreeBSD 12.0: FAIL
ANVL-RIPNG-2.1 **MUST**	NEGATIVE RFC 2080 s2 p4 Protocol Specification RFC 2080 s2.1 p7 Message Format							

	`RIPng Message Format`
	`The RIPng metric of a network is an integer between 1 and 15,`
	`inclusive, specifying the current metric for the destination;`
	`or the value 16 (infinity), which indicates that the destination`
	`is not reachable.`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-2.2 **MUST**	RFC 2080 s2.1 p5 Message Format							

	`RIPng Message Format`
	`Each router that uses RIPng has a routing process that sends`
	`datagrams on UDP port number 521, the RIPng port.`
	`Unsolicited routing update messages have both the source and`
	`destination port equal to the RIPng port.`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-2.3 **MUST**	RFC 2080 s2.1 p5 Message Format							
	RIPng Message Format Each router that uses RIPng has a routing process that receives datagrams on UDP port number 521, the RIPng port.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-2.4 **MUST**	RFC 2080 s2.1 p5 Message Format							
	RIPng Message Format Those sent in response to a request are sent to the port from which the request came.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-3.1 **MUST**	RFC 2080 s2.1.1 p7 Next Hop							
	RIPng Next Hop The route tag and prefix length in the next hop RTE must be set to zero on sending.							
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: FAIL	FreeBSD 12.0: FAIL
ANVL-RIPNG-3.2 **MUST**	NEGATIVE RFC 2080 s2.1.1 p7 Next Hop							
	RIPng Next Hop The route tag in the next hop RTE must be ignored on reception. (Note : Prefix Length is set to zero but route tag set to non-zero so DUT must ignore this non-zero value)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-3.3 **MUST**	NEGATIVE RFC 2080 s2.1.1 p7 Next Hop							
	RIPng Next Hop The prefix length in the next hop RTE must be ignored on receiption. (Note : Prefix Length is set to non-zero but route tag set to zero so DUT must ignore this non-zero value)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-3.4 **SHOULD**	RFC 2080 s2.1.1 p8 Next Hop							
	RIPng Next Hop Specifying a value of 0:0:0:0:0:0:0:0 in the prefix field of a next hop RTE indicates that the next hop address should be the originator of the RIPng advertisement.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-3.5 **MUST**	RFC 2080 s2.1.1 p8 Next Hop							
	RIPng Next Hop An address specified as a next hop must be a link-local address. If the received next hop address is not a link-local address, it should be treated as 0:0:0:0:0:0:0:0							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-4.1 **SHOULD**	RFC 2080 s2.2 p8 Addressing Considerations							
	RIPng Addressing Considerations In general, the system administrator will be provided with a way to specify which routers should create and advertise default route entries.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-5.1 **MUST**	RFC 2080 s2.3 p9 Timers							
	RIPng Timers Every 30 seconds, the RIPng process is awakened to send an unsolicited Response message. - The 30-second timer is offset by a small random time (+/- 0 to 15 seconds) each time it is set. The offset is derived from: 0.5 * the update period (i.e. 30).							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-5.2 **MUST**	RFC 2080 s2.3 p9 Timers							
	RIPng Timers If 180 seconds elapse from the last time the timeout was initialized, the route is considered to have expired.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-5.3 **MUST**	RFC 2080 s2.3 p9 Timers							
	RIPng Timers Deletions can occur for one of two reasons: - the timeout expires. (Note: The received RIPng Update from DUT can be a triggered update or a regular update that will have the metric field for the RTE set to 16 (infinity))							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-5.5 **MUST**	RFC 2080 s2.3 p10 Timers							
	RIPng Timers Until the garbage-collection timer expires, the route is included in all updates sent by this router.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-5.6 **MUST**	RFC 2080 s2.3 p10 Timers							
	RIPng Timers When the garbage-collection timer expires, the route is deleted from the routing table.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-6.1 **SHOULD**	RFC 2080 s2.4.1 p10 Request Messages							
	RIPng Request Messages Normally, Requests are sent as multicasts, from the RIPng port, by routers which have just come up and are seeking to fill in their routing tables as quickly as possible.							
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: FAIL	FreeBSD 12.0: FAIL
ANVL-RIPNG-6.4 **MUST**	RFC 2080 s2.4.1 p10 Request Messages RFC 2080 s2.5.2 p15 Generating Response Messages							
	RIPng Request Messages However, there may be situations ... If such a Request is received, the router responds directly to the requestor"s address and port with a globally valid source address since the requestor may not reside on the directly attached network.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-6.5 **MUST**	RFC 2080 s2.4.1 p11 Request Messages							
	RIPng Request Messages If there are no entries, no response is given.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-6.7 **MUST**	RFC 2080 s2.4.1 p11 Request Messages							
	RIPng Request Messages If there is no explicit route to the specified destination, put infinity in the metric field.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-6.8 **MUST**	RFC 2080 s2.4.1 p11 Request Messages							
	RIPng Request Messages If the request is for specific entries, they are looked up in the routing table and the information is returned as is; no Split Horizon processing is done.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.1 **MUST**	RFC 2080 s2.4.2 p11 Response Messages							
	RIPng Response Messages The Response must be ignored if it is not from the RIPng port.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.2 **MUST**	RFC 2080 s2.4.2 p11 Response Messages							
	RIPng Response Messages The Response must be ignored if it is not from the RIPng port. (Note: Here we are testing that response will be accepted if it is from RIPng Port)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.3 **MUST**	RFC 2080 s2.4.2 p11 Response Messages RFC 2080 s2.5.2 p15 Generating Response Messages							
	RIPng Response Messages The datagram"s IPv6 source address should be checked to see whether the datagram is from a valid neighbor; the source of the datagram must be a link-local address.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.4 **MUST**	NEGATIVE RFC 2080 s2.4.2 p11 Response Messages RFC 2080 s2.5.2 p15 Generating Response Messages							
	RIPng Response Messages The datagram"s IPv6 source address should be checked to see whether the datagram is from a valid neighbor; the source of the datagram must be a link-local address.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.5 **MUST**	NEGATIVE RFC 2080 s2.4.2 p11 Response Messages							
	RIPng Response Messages It is also worth checking to see whether the response is from one of the router"s own addresses. If a router processes its own output as new input, confusion is likely, and such datagrams must be ignored.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.6 **MUST**	RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages As an additional check, periodic advertisements must have their hop counts set to 255, and inbound, multicast packets sent from the RIPng port (i.e. periodic advertisement) must be examined to ensure that the hop count is 255.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.8 **MUST**	NEGATIVE RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages As an additional check, periodic advertisements must have their hop counts set to 255.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.9 **MUST**	RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages Queries and their responses may still cross intermediate nodes and therefore do not require the hop count test to be done.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.10 **SHOULD**	NEGATIVE RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages The basic validation tests of a RTE are: - is the destination prefix valid (e.g., not a multicast prefix and not a link-local address) A link-local address should never be present in an RTE.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.11 **MUST**	RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages If any check fails, ignore that entry and proceed to the next.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.12 **MUST**	RFC 2080 s2.4.2 p12 Response Messages							
	RIPng Response Messages Once the entry has been validated, update the metric by adding the cost of the network on which the message arrived. If the result is greater than infinity, use infinity. That is, metric = MIN (metric + cost, infinity).							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.13 **MUST**	RFC 2080 s2.4.2 p12 Response Messages							

	`RIPng Response Messages`
	`If there is no such route, add this route to the routing table,`
	`unless the metric is infinity (there is no point in adding a`
	`route which unusable).`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.14 **MUST**	NEGATIVE RFC 2080 s2.4.2 p12 Response Messages							

	`RIPng Response Messages`
	`If there is no such route, add this route to the routing table,`
	`unless the metric is infinity (there is no point in adding a`
	`route which unusable).`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.15 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							

	`RIPng Response Messages`
	`Adding a route to the routing table consists of:`
	`- Signal the output process to trigger an update.`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.16 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							

	`RIPng Response Messages`
	`If there is an existing route, compare the next hop address to`
	`the address of the router from which the datagram came.  If this`
	`datagram is from the same router as the existing route,`
	`reinitialize the timeout.`							

	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.17 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages If the datagram is from the same router as the existing route, and the new metric is different than the old one; - Adopt the route from the datagram. That is, put the new metric in.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.18 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages If the datagram is from the same router as the existing route, and the new metric is different than the old one; or, if the new metric is lower than the old one; do the following actions: - Adopt the route from the datagram. That is, put the new metric in. (Note : Here we send RIPng updates from two different routers)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.19 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages If the datagram is from the same router as the existing route, and the new metric is different than the old one; - Adopt the route from the datagram. That is, adjust the next hop address (if necessary).							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-7.20 **MUST**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages If the datagram is from the same router as the existing route, and the new metric is different than the old one; or, if the new metric is lower than the old one; - Adopt the route from the datagram. That is, adjust the next hop address (if necessary). (Note : Here we send RIPng updates from two different routers)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.22 **SHOULD**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages Therefore, if the new metric is the same as the old one, examine the timeout for the existing route. If it is at least halfway to the expiration point, switch to the new route. (Note : Here we test updates from the same router)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-7.23 **SHOULD**	RFC 2080 s2.4.2 p13 Response Messages							
	RIPng Response Messages Therefore, if the new metric is the same as the old one, examine the timeout for the existing route. If it is at least halfway to the expiration point, switch to the new route. (Note : Here we test updates from two different routers)							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-8.2 **MUST**	RFC 2080 s2.5.1 p14 Triggered Updates							
	RIPng Triggered Updates After a triggered update is sent, a timer should be set for a random interval between 1 and 5 seconds. If other changes that would trigger updates occur before the timer expires, a single update is triggered when the timer expires. (Note : In this test we check that the time difference between two successive RIPng triggered updates is within the range of 1 - 5 seconds)							
	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-9.1 **MUST**	RFC 2080 s2.5.2 p16 Generating Response Messages							
	Generating RIPng Response Messages The version described in this document is version 1 and the bytes labeled "must be zero" to zero.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-9.2 **MUST**	NEGATIVE RFC 2080 s2.5.2 p16 Generating Response Messages							
	Generating RIPng Response Messages The version described in this document is version 1.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-9.3 **MUST**	RFC 2080 s2.5.2 p16 Generating Response Messages							
	Generating RIPng Response Messages Routes to link-local addresses must never be included in an RTE.							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass

	Release 2.0.2	Release 3.0.2	Release 3.0.3	Release 4.0	Release 5.0.1	Release 6.0	Master 2019-02-24	Release 7.0
ANVL-RIPNG-9.4 **MUST**	RFC 2080 s2.5.2 p16 Generating Response Messages							
	`Generating RIPng Response Messages` `Routes must be included in the datagram even if their metrics are infinite.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-10.1 **MUST**	RFC 2080 s2.6 p16 Split Horizon							
	`Split Horizon` `The basic split horizon algorithm omits routes learned from one neighbor in updates sent to that neighbor.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass
ANVL-RIPNG-10.2 **MUST**	RFC 2080 s2.6 p16 Split Horizon							
	`Split Horizon` `Split Horizon with Poisoned Reverse (more simply, Poison Reverse) does include such routes in updates, but sets their metrics to infinity.`							
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: untested
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: untested	FreeBSD 12.0: pass	FreeBSD 12.0: pass