	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
Type	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR
Commit ID	3e71b5d	3d7746c	f731a65	f92f83b	c47b10c	fb13970	511684d	5cf0c43	2d67d5a
Commit Date	2017-04-02	2017-04-25	2017-05-24	2017-07-01	2017-08-09	2017-08-16	2017-08-24	2017-09-08	2017-09-14
ANVL-RIP-1.1 **MUST**	RFC 2453 s3.6 p20 Message Format								
	RIP Message and Packet Formats Each router that uses RIP has a routing process that sends datagrams on UDP port number 520.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-2.1 **MUST**	NEGATIVE: RFC 2453 s3.6 p21 Message Format RFC 2453 s3.10.2 p30 Generating Response Messages								
	RIP Packet Formats There may be between 1 and 25 (inclusive) RIP entries. Recall that there is a limit of 25 RTEs to a Response.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-2.2 **MUST**	NEGATIVE: RFC 2453 s4 p31 Protocol Extensions RFC 2453 s3.6 p20-21 Message Format								
	RIP Packet Formats The RIP Message Format is: 0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | command (1) | version (1) | must be zero (2) | +---------------+---------------+-------------------------------+ | | ~ RIP Entry (20) ~ | | +---------------+---------------+---------------+---------------+ There may be between 1 and 25 (inclusive) RIP entries. (NOTE: Here we are testing that only valid RIP packets may be accepted.)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-2.3 **MUST**	NEGATIVE: RFC 2453 s3.1 p21 Message Format								
	RIP Packet Formats The commands implemented in version 1 and 2 are request and response								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-2.4 **MUST**	NEGATIVE RFC 2453 s3.6 p21 Message Format								
	RIP Packet Formats For RIP-1, only AF_INET (2) is generally supported.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-2.5 **MUST**	NEGATIVE: RFC 2453 p21 Message Format								
	RIP Packet Formats The metric field contains a value between 1 and 15 (inclusive) which specifies the current metric for the destination; or the value 16, which indicates that the destination is not reachable.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-2.8 **MUST**	RFC 2453 s3.6 p20 Message Format RFC 2453 s4 p31 Protocol Extensions								
	RIP Packet Formats The RIP Response Message Format is: 0 1 2 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | command (1) | version (1) | must be zero (2) | +---------------+---------------+-------------------------------+ | ~ RIP Entry (20) ~ | +---------------+---------------+---------------+---------------+ There may be between 1 and 25 (inclusive) RIP entries.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-3.1 **MUST**	RFC 2453 s3.7 p22 Addressing Considerations								
	RIP Addressing Consierations If host routes are not supported, they are to be dropped when they are received in response messages.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-3.2 **MUST**	NEGATIVE: RFC 2453 s3.7 p22-23 Addressing considerations								
	RIP Addressing Consierations The destinations appearing in request and response messages can be networks, hosts, or a special code used to indicate a default address. Normally hosts only know the subnet masks for directly-connected networks. (NOTE: Here we are testing the DUT does not accept bad values in address fields.)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-3.3 **MUST**	RFC 2453 s3.7 p22 Addressing Considerations								
	RIP Addressing Consierations RIP-1 routes to a subnet must not be sent outside the network of which the subnet is a part.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-3.5 **SHOULD**	RFC 2453 s3.7 p23 Addressing Considerations								
	RIP Addressing Consierations These routers should create RIP entries for the address 0.0.0.0, just as if it were a network to which they are connected. The decision as to how routers create entries for 0.0.0.0 is left to the implementor. Most commonly, the system administrator will be provided with a way to specify which routers should create entries for 0.0.0.0								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-4.3 **SHOULD**	RFC 2453 s3.8 p24 Timers								
	RIP Timers Route expiration timer should be 180 seconds and garbage collection timer should be 120 seconds.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-4.4 **MUST**	RFC 2453 s3.8 p23-24 Timers								
	RIP Timers The garbage-collection timer is reset upon the reception of a new route to an unreachable network.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-5.1 **MUST**	RFC 2453 s5 p34 Compatability								
	Input Processing RIP messages of version 0 are to be discarded.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-5.2 **MUST**	RFC 2453 s5 p34 Compatability								
	Input Processing RIP messages of version 1 are to be discarded if any Must Be Zero (MBZ) field is non-zero.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-5.3 **SHOULD**	RFC 2453 s5 p34 Compatability								
	Input Processing RIP messages of any version greater than 1 should not be discarded simply because an MBZ field contains a value other than zero.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-6.1 **MUST**	RFC 2453 s3.9.1 p25 Request Messages								
	RIP Requests Normally, Requests are sent as broadcasts, from the RIP port, by routers which have just come up and are seeking to fill in their routing tables as quickly as possible. However, there may be situations (e.g., router monitoring) where the routing table of only a single router is needed. In this case, the Request should be sent directly to that router from a UDP port other than the RIP port. If such a Request is received, the router responds directly to the requestor"s address and port.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-6.5 **MUST**	NEGATIVE: RFC 2453 s3.9.1 p25 Request Messages								
	RIP Requests If there is exactly one entry in the request, and it has an address family identifier of zero and a metric of infinity (i.e., 16), then this is a request to send the entire routing table.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-6.6 **MUST**	RFC 2453 s3.9.1 p25 Request Messages								
	RIP Requests Validate RIP Response Message in reply to Request Message.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.1 **MUST**	RFC 2453 s3.9.2 p26 Response Messages								
	RIP Responses The Response must be ignored if it is not from the RIP port. (UDP Port 520).								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.2 **MUST**	NEGATIVE: RFC 2453 s3.9.2 p26 Response Messages								
	RIP Responses The datagram"s IPv4 source address should be checked to see whether the datagram is from a valid neighbor								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-7.3 **MUST**	NEGATIVE: RFC 2453 s3.9.2 p26 Response Messages								
	RIP Responses It is also worth checking to see whether the response is from one of the router"s own addresses. Interfaces on broadcast networks may receive copies of their own broadcasts/multicasts immediately. If a router processes its own output as new input, confusion is likely so such datagrams must be ignored.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-14.1 **MUST**	RFC 2453 s4.4 p33 Next hop								
	RIP Next Hop An address specified as a next hop must, per force, be directly reachable on the logical subnet over which the advertisement is made.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-14.2 **MUST**	RFC 2453 s4.4 p33 Next hop								
	RIP Next Hop The purpose of the Next Hop field is to eliminate packets being routed through extra hops in the system. It is particularly useful... If the received Next Hop is not directly reachable, it should be treated as 0.0.0.0.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-15.1 **MUST**	RFC 2453 s4.5 p33 Multicasting								
	RIP Multicasting In order to reduce unnecessary load on those hosts which are not listening to RIP-2 messages, an IP multicast address will be used for periodic broadcasts. The IP multicast address is 224.0.0.9. In order to maintain backwards compatibility, the use of the multicast address will be configurable (NOTE: Here we are testing DUT sends multicast RIP-2 update)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-15.2 **MUST**	RFC 2453 s4.5 p33 Multicasting								
	RIP Multicasting In order to reduce unnecessary load on those hosts which are not listening to RIP-2 messages, an IP multicast address will be used for periodic broadcasts. The IP multicast address is 224.0.0.9. In order to maintain backwards compatibility, the use of the multicast address will be configurable (NOTE: Here we are testing DUT accepts multicast RIP-2 update)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-16.1 **MUST**	RFC 2453 s5.1 p34 Compatibility switch								
	RIP Version Compatibility The switch has four settings: RIP-1, in which only RIP-1 messages are sent; RIP-1 compatibility, in which RIP-2 messages are broadcast; RIP-2, in which RIP-2 messages are multicast; and "none", which disables the sending of RIP messages. CASE: Only RIP-1 messages are sent								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.2 **MUST**	RFC 2453 s5.1 p34 Compatibility switch								
	RIP Version Compatibility The switch has four settings: RIP-1, in which only RIP-1 messages are sent; RIP-1 compatibility, in which RIP-2 messages are broadcast; RIP-2, in which RIP-2 messages are multicast; and "none", which disables the sending of RIP messages. CASE: RIP-2 messages are broadcast								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-17.1 **MAY**	RFC 2453 s3.10 p29 Output Processing								
	RIP Parameter Setting It may be necessary to specify an actual list of neighboring routers and send a datagram to each one explicitly								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-1.2 **MUST**	RFC 2453 s3.6 p20 Message Format								
	RIP Message and Packet Formats Unsolicited routing update messages have both source and destination port equal to the RIP port (UDP port number 520).								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-1.3 **MUST**	RFC 2453 s3.6 p20 Message Format								
	RIP Message and Packet Formats Update messages sent in response to a request are sent to the port from which the request came.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.9 **MUST**	NEGATIVE: RFC 2453 s3.10.2 p30 Generating Response Messages RFC 2453 s5 p34 Compatibility								
	RIP Responses Set the command to Response. Set the bytes labeled "must be zero" to zero. RIP messages of version 1 are to be discarded if any Must Be Zero (MBZ) field is non-zero								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-7.10 **MUST**	RFC 2453 s3.4.2 p27 Response Messages								
	RIP Responses 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)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.12 **MUST**	RFC 2453 s3.9.2 p27 Response Messages								
	RIP Responses 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 is unusable).								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.13 **MUST**	RFC 2453 s3.9.2 p28 Response Messages								
	RIP Responses If the new metric is infinity, start the deletion process								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-7.14 **MUST**	RFC 2453 s3.9.2 p27 Response Messages								
	RIP Responses Any entry that fails these tests is ignored, as it is no better than the current route.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-8.1 **MUST**	RFC 2453 s3.10 p28 Output Processing								
	Output Processing This processing may be triggered by input processing, when a Request is received (this Response is unicast to the requestor)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-8.3 **MUST**	RFC 2453 s3.10 p28 Output Processing								
	Output Processing This processing may be triggered by triggered updates (broadcast/multicast when a route changes)								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-8.5 **SHOULD**	RFC 2453 s3.10.1 p29 Triggered Updates								
	Output Processing 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. The timer is then reset to another random value between 1 and 5 seconds.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-8.17 **MUST**	RFC 2453 s3.4.3 p15-16 Split horizon								
	Output Processing The "simple split horizon" scheme omits routes learned from one neighbor in updates sent to that neighbor. Thus implementors may at their option implement simple split horizon rather than split horizon with poisoned reverse The router requirements RFC [11] specifies that all implementation of RIP must use split horizon								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-9.1 **MUST**	RFC 2453 s3.6 p20 Message format								
	RIP Version 2 Packet Formats The RIP Header format is: 0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | command (1) | version (1) | must be zero (2) | +---------------+---------------+-------------------------------+								
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-RIP-9.2 **MUST**	RFC 2453 s4 p31 Protocol Extensions								
	RIP Version 2 Packet Formats The format for the 20-octet route entry (RTE) for RIP-2 is: 0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Address Family Identifier (2) | Route Tag (2) | +---------------+---------------+-------------------------------+ | IP Address (4) | +---+ | Subnet Mask (4) | +---+ | Next Hop (4) | +---+ | Metric (4) | +---+								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-10.1 **MUST**	RFC 2453 s4.1 p31 Authentication								
	RIP Version 2 Authentication If the Address Family Identifier of the first (and only the first) entry in the message is 0xFFFF, then the remainder of the entry contains the authentication.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-10.2 **MUST**	NEGATIVE: RFC 2453 s4.1 p31 Authentication								
	RIP Version 2 Authentication If authentication is not in use, then no entries in the message should have an Address Family Identifier of 0xFFFF.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-10.3 **MUST**	NEGATIVE: RFC 2453 s4.1 p32 Authentication								
	RIP Version 2 Authentication Currently, the only Authentication Type is simple password and it is type 2. The remaining 16 octets contain the plain text password. If the password is under 16 octets, it must be left-justified and padded to the right with nulls (0x00).								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.3 **MUST**	RFC 2453 s5.1 p34 Compatibility switch								
	RIP Version Compatibility The switch has four settings: RIP-1, in which only RIP-1 messages are sent; RIP-1 compatibility, in which RIP-2 messages are broadcast; RIP-2, in which RIP-2 messages are multicast; and "none", which disables the sending of RIP messages. CASE: RIP-2 messages are multicast								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.4 **MUST**	RFC 2453 s5.1 p34 Compatibility switch								
	RIP Version Compatibility The switch has four settings: RIP-1, in which only RIP-1 messages are sent; RIP-1 compatibility, in which RIP-2 messages are broadcast; RIP-2, in which RIP-2 messages are multicast; and "none", which disables the sending of RIP messages. CASE: No RIP messages are sent								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.5 **SHOULD**	RFC 2453 s5.1 p34 Compatibility Switch								
	RIP Version Compatibility For completeness, routers should also implement a receive control switch which would determine whether to accept RIP-1 only.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-RIP-16.6	RFC 2453 s5.1 p34 Compatibility Switch								
SHOULD	RIP Version Compatibility For completeness, routers should also implement a receive control switch which would determine whether to accept RIP-2 only								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.7	RFC 2453 s5.1 p34 Compatibility Switch								
SHOULD	RIP Version Compatibility For completeness, routers should also implement a receive control switch which would determine whether to accept both								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass
ANVL-RIP-16.8	RFC 2453 s5.1 p34 Compatibility Switch								
SHOULD	RIP Version Compatibility For completeness, routers should also implement a receive control switch which would determine whether to accept none.								
	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	Ubuntu 16.04: pass
	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: pass	FreeBSD 10.3: pass