	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
Type	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR
Commit ID	3e71b5d	3d7746c	b84ccd4	f731a65	bade23d	f30a732	f92f83b	dceb5f8	c47b10c	fb13970
Commit Date	2017-04-02	2017-04-25	2017-05-16	2017-05-24	2017-06-02	2017-06-27	2017-07-01	2017-07-21	2017-08-09	2017-08-16
ANVL-BGPPLUS-1.1 **MUST**	ANVL, setup verification									
	ANVL, Setup Verification DUT Listens on TCP port 179 for BGP4 Connection									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-1.2 **MUST**	ANVL, setup verification									
	ANVL, Setup Verification Establish BGP4 connection to the DUT and transit to Established state									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-1.3 **MUST**	ANVL, setup verification									
	ANVL, Setup Verification Router adds routes contained in the newly received Update Message to its 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-2.1 **MUST**	RFC4760, Sect. 1: Introduction, p 1, Overview									
	Requirement of IPv4 address for Multiprotocol Extensions This document assumes that any BGP speaker (including the one that supports multiprotocol capabilities defined in this document) has to have an IPv4 address (which will be used, among other things, in the AGGREGATOR attribute).									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-3.1 **MUST**	RFC 4760, Sect. 3, p 2, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14)									
	Purpose of MP_REACH_NLRI attribute This is an optional non-transitive attribute that can be used for the following purposes: (a) to advertise a feasible route to a peer (b) to permit a router to advertise the Network Layer address of the router that should be used as the next hop to the destinations listed in the Network Layer Reachability Information field of the MP_REACH_NLRI attribute.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-3.2 **MUST**	RFC 4760, Sect. 3, p 3, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14) Reserved									
	Purpose of MP_REACH_NLRI attribute A 1 octet field that MUST be set to 0, and SHOULD be ignored upon receipt. Note: Here we check that the Reserved field is set to 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-3.3 **MUST**	RFC 4760, Sect. 3, p 3, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14) Reserved									
	Purpose of MP_REACH_NLRI attribute A 1 octet field that MUST be set to 0, and SHOULD be ignored upon receipt. Note: Here we check that DUT ignores the non-zero reserved field									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-3.4 **MUST**	RFC 4760, Sect. 3, p 4, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14)									
	Purpose of MP_REACH_NLRI attribute An UPDATE message that carries the MP_REACH_NLRI must also carry the ORIGIN and the AS_PATH attributes (for EBGP)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-3.5 **MUST**	RFC 4760, Sect. 3, p 4, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14)									
	Purpose of MP_REACH_NLRI attribute An UPDATE message that carries the MP_REACH_NLRI must also carry the ORIGIN and the AS_PATH attributes (for IBGP)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-3.6 **MUST**	RFC 4760, Sect. 3, p 4, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14)									
	Purpose of MP_REACH_NLRI attribute Moreover, in IBGP exchanges such a message must also carry the LOCAL_PREF attribute.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-3.7 **SHOULD**	NEGATIVE RFC 4760, Sect. 3, p 4, Multiprotocol Reachable NLRI - MP_REACH_NLRI (Type Code 14)									

	`Purpose of MP_REACH_NLRI attribute`
	`An UPDATE message that carries no NLRI, other than the one encoded in`
	`the MP_REACH_NLRI attribute, SHOULD NOT carry the NEXT_HOP attribute.`
	`If such a message contains the NEXT_HOP attribute, the BGP speaker`
	`that receives the message SHOULD ignore this attribute.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-4.1 **MUST**	RFC 4760, Sect. 4, p 5, Multiprotocol Unreachable NLRI - MP_UNREACH_NLRI (Type Code 15):									

	`Purpose of MP_UNREACH_NLRI attribute`
	`An UPDATE message that contains the MP_UNREACH_NLRI is not required`
	`to carry any other path attributes.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-5.1 **MUST**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling									

	`Error Handling`
	`If a BGP speaker receives from a neighbor an Update message that`
	`contains the MP_REACH_NLRI or MP_UNREACH_NLRI attribute, and the`
	`speaker determines that the attribute is incorrect, the speaker must`
	`delete all the BGP routes received from that neighbor whose AFI/SAFI`
	`is the same as the one carried in the incorrect MP_REACH_NLRI or`
	`MP_UNREACH_NLRI attribute.`
	`(Note: ANVL sends two updates, the second update containing`
	`MP_REACH_NLRI attribute with incorrect length of nlri set to 129`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-5.2 **MUST**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling									

	`Error Handling`
	`If a BGP speaker receives from a neighbor an Update message that`
	`contains the MP_REACH_NLRI or MP_UNREACH_NLRI attribute, and the`
	`speaker determines that the attribute is incorrect, the speaker must`
	`delete all the BGP routes received from that neighbor whose AFI/SAFI`
	`is the same as the one carried in the incorrect MP_REACH_NLRI or`
	`MP_UNREACH_NLRI attribute.`
	`(Note: ANVL sends two updates, the second update containing`
	`MP_UNREACH_NLRI attribute with SAFI set to Unicast even when the`
	`route is 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-5.3 **MAY**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling									
	Error Handling In addition, the speaker may terminate the BGP session over which the Update message was received. (Note: Here, the UPDATE sent by ANVL contains incorrect NLRI length 129									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-5.4 **MAY**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling									
	Error Handling In addition, the speaker may terminate the BGP session over which the Update message was received. (Note: Here, the UPDATE sent by ANVL contains incorrect MP_UNREACH_NLRI which causes DUT to close the BGP4 connection with the sending peer)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-5.5 **SHOULD**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling RFC 4271, Sect. 6.3, p 34, UPDATE message error handling									
	Error Handling The session should be terminated with the Notification message code/subcode indicating "Update Message Error"/"Optional Attribute Error". The NLRI field in the UPDATE message is checked for syntactic valid- ity. If the field is syntactically incorrect, then the Error Subcode MUST be set to Invalid Network Field. (Note: Here we are checking this behavior using incorrect MP_REACH_NLRI attribute in the BGP4 UPDATE Message sent by ANVL)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-5.6 **SHOULD**	NEGATIVE RFC 4760, Sect. 7, p 8, Error Handling RFC 4271, Sect. 6.3, p 34, UPDATE message error handling									
	Error Handling The session should be terminated with the Notification message code/subcode indicating "Update Message Error"/"Optional Attribute Error". The NLRI field in the UPDATE message is checked for syntactic valid- ity. If the field is syntactically incorrect, then the Error Subcode MUST be set to Invalid Network Field. (Note: Here we are checking this behavior using incorrect MP_UNREACH_NLRI attribute in the BGP4 UPDATE Message sent by ANVL)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-6.1 **SHOULD**	RFC 4760, Sect. 8, p 8, Use of BGP Capability Advertisement									
	BGP4 Capability Advertisement A BGP speaker that uses Multiprotocol Extensions should use the Capability Advertisement procedures [BGP-CAP] to determine whether the speaker could use Multiprotocol Extensions with a particular peer.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-6.2 **MUST**	RFC 4760, Sect. 8, p 9, Use of BGP Capability Advertisement									
	BGP4 Capability Advertisement A speaker that supports multiple AFI, SAFI> tuples includes them as multiple Capabilities in the Capabilities Optional Parameter.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-6.3 **MUST**	RFC 4760, Sect. 8, p 9, Use of BGP Capability Advertisement									
	BGP4 Capability Advertisement To have a bi-directional exchange of routing information for a particular AFI, SAFI> between a pair of BGP speakers, each such speaker must advertise to the other (via the Capability Advertisement mechanism) the capability to support that particular AFI, SAFI> routes.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-7.1 **MUST**	NEGATIVE RFC 4760, Sect. 9, p 9, IANA Considerations									
	IANA Considerations SAFI value 0 and 255 are reserved.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-8.1 **MUST**	RFC 2545, Sect. 2, p 2, IPv6 Address Scopes									
	IPv6 Address Scopes As this document makes no assumption on the characteristics of a particular routing realm where BGP-4 is used, it makes no distinction between global and site-local addresses and refers to both as "global" or "non-link-local".									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-9.1 **SHOULD**	NEGATIVE RFC 2545, Sect. 3, p 2, Constructing the Next Hop field									
	Next Hop field The value of the Length of Next Hop Network Address field on a MP_REACH_NLRI attribute shall be set to 16, when only a global address is present, or 32 if a link-local address is also included in the Next Hop field. (Note: In this test we send only a link-local address even when we set the length of NEXT_HOP field to 16)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-9.2 **MUST**	RFC 2545, Sect. 3, p 2, Constructing the Next Hop field RFC 2545, Sect. 3, p 3, Constructing the Next Hop field									
	Next Hop field The value of the Length of Next Hop Network Address field on a MP_REACH_NLRI attribute shall be set to 16, when only a global address is present, or 32 if a link-local address is also included in the Next Hop field. In all other cases a BGP speaker shall advertise to its peer in the Network Address field only the global IPv6 address of the next hop (the value of the Length of Network Address of Next Hop field shall be set to 16). (Note: Here we test that DUT correctly sets the NEXT_HOP field of MP_REACH_NLRI attribute when length is set to 16)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-9.3 **SHOULD**	RFC 2545, Sect. 2, p 2, IPv6 Address Scopes RFC 2545, Sect. 3, p 2, Constructing the Next Hop field									
	Next Hop field A BGP speaker shall advertise to its peer in the Network Address of Next Hop field the global IPv6 address of the next hop, potentially followed by the link-local IPv6 address of the next hop. The value of the Length of Next Hop Network Address field on a MP_REACH_NLRI attribute shall be set to 16, when only a global address is present, or 32 if a link-local address is also included in the Next Hop field. The link-local address shall be included in the Next Hop field if and only if the BGP speaker shares a common subnet with the entity identified by the global IPv6 address carried in the Network Address of Next Hop field and the peer the route is being advertised to. (Note: Here, we verify that the DUT correctly sends the link-local address along with the non-link-local address in its UPDATE Message. This test uses FIRST PARTY NEXT_HOP)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-9.4 **SHOULD**	NEGATIVE RFC 2545, Sect. 3, p 2, Constructing the Next Hop field									
	Next Hop field The link-local address shall be included in the Next Hop field if and only if the BGP speaker shares a common subnet with the entity identified by the global IPv6 address carried in the Network Address of Next Hop field and the peer the route is being advertised to. (Note: Here, we test that the DUT does not accept a UPDATE sent by ANVL containing an off-net non-link-local IPv6 Address following by a link-local IPv6 Address of sending interface. This test verifies FIRST PARTY NEXT_HOP)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-9.5 **MAY**	RFC 2545, Sect. 3, p 2, Constructing the Next Hop field RFC 2545, Sect. 3, p 3, Constructing the Next Hop field									
	Next Hop field The link-local address shall be included in the Next Hop field ... In all other cases a BGP speaker shall advertise to its peer in the Network Address field only the global IPv6 address of the next hop... As a consequence, a BGP speaker that advertises a route to an internal peer may modify the Network Address of Next Hop field by removing the link-local IPv6 address of the next hop.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-10.1 **MUST**	RFC 2545, Sect. 4, p 3 Transport									
	Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. IPv6/IPv6 AFI and Unicast SAFI (Note: This test is to verify that DUT correctly specifies the NLRI and NEXT_HOP field types in MP_REACH_NLRI attribute as IPv6 in its BGP4 Update Message over TCP/IPv6 through AFI/SAFI> combination)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-10.2 **MUST**	RFC 2545, Sect. 4, p 3 Transport									
	Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies its IPv6 route advertisement capabilities in BGP4 Open Message when runing over TCP/IPv4)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-10.3 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies the NLRI and NEXT_HOP field types in MP_REACH_NLRI attribute as IPv6 in its BGP4 Update Message over TCP/IPv4 through AFI/SAFI> combination)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-10.4 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies its IPv4 route advertisement capabilities in BGP4 Open Message when runing over TCP/IPv6)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-10.5 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies the NLRI and NEXT_HOP field types in MP_REACH_NLRI attribute as IPv4 in its BGP4 Update Message over TCP/IPv6 through AFI/SAFI> combination)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: pass
ANVL-BGPPLUS-10.6 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies its IPv4 route advertisement capabilities in BGP4 Open Message when runing over TCP/IPv4)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-10.7 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies the NLRI and NEXT_HOP field types in MP_REACH_NLRI attribute as IPv4 in its BGP4 Update Message over TCP/IPv4 through AFI/SAFI> combination)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: pass
ANVL-BGPPLUS-10.8 **MUST**	RFC 2545, Sect. 4, p 3 Transport Transport layer independance TCP connections, on top of which BGP-4 messages are exchanged, can be established either over IPv4 or IPv6. While BGP-4 itself is independent of the particular transport used it derives implicit configuration information from the address used to establish the peering session. This information (the network address of a peer) is taken in account in the route dissemination procedure. (Note: This test is to verify that DUT correctly specifies the NLRI and Next Hop when sending an update to a peer over TCP-V4> received from a different peer over TCP-V6>)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-11.1 **MUST**	RFC 4271, Sect. 4, p 10, Message Formats Message Formats The maximum message size is 4096 octets. All implementations are required to support this maximum message size.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-12.1 **MUST**	NEGATIVE RFC 4271, Sect. 4.2, p 12, OPEN Message Format OPEN Message Format Upon receipt of an OPEN message, a BGP speaker MUST calculate the value of the Hold Timer by using the smaller of its configured Hold Time and the Hold Time received in the OPEN 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-12.2 **MUST**	RFC 4271, Sect. 4.2, p 12, OPEN Message Format OPEN Message Format The Hold Time MUST be either zero or at least three seconds. (Note: Here we test the Hold Time value with 0 or 3 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-12.3 **MUST**	NEGATIVE RFC 4271, Sect. 4.2, p 12, OPEN Message Format RFC 4271, Sect. 6.2, p 31, OPEN message error handling									
	OPEN Message Format The Hold Time MUST be either zero or at least three seconds. If the Hold Time field of the OPEN message is unacceptable, then the Error Subcode MUST be set to Unacceptable Hold Time. An implementation MUST reject Hold Time values of one or two seconds. (Note: Here we test the Hold Time value with 1 second and 2 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-12.4 **MUST**	NEGATIVE RFC 4271, Sect. 4.2, p 13, OPEN Message Format									
	OPEN Message Format The calculated value for Hold Time indicates the maximum number of seconds that may elapse between the receipt of successive KEEPALIVE,and/or UPDATE messages by the sender. (Note: Here, we test that the DUT sends a NOTIFICATION message due to not receiving successive UPDATE/KEEPALIVE messages within Hold Time Period)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-12.5 **MUST**	NEGATIVE RFC 4271, Sect. 4.2, p 13, OPEN Message Format									
	OPEN Message Format The calculated value for Hold Time indicates the maximum number of seconds that may elapse between the receipt of successive KEEPALIVE, and/or UPDATE messages by the sender. (Note: Here, we test that the DUT sends a NOTIFICATION message due to not receiving successive KEEPALIVE messages within Hold Time Period)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.1 **MAY**	RFC 4271, Sect. 4.3, p 14, UPDATE Message Format									
	UPDATE Message Format An UPDATE message MAY simultaneously advertise a feasible route and withdraw multiple unfeasible routes from service.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.2 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes, the Transitive bit must be set to 1. (Note: Here we test with the path attribute type ORIGIN)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-13.3 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes, the Transitive bit must be set to 1. (Note: Here we test with the path attribute type AS_PATH)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.4 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes, the Transitive bit must be set to 1. (Note: Here we test with the path attribute type LOCAL_PREF)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.5 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes, the Transitive bit must be set to 1. (Note: Here we test with the path attribute type ATOMIC_AGGREGATE)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.6 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type ORIGIN)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.7 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type AS_PATH)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.8 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type MP_REACH_NLRI)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-13.9 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type LOCAL_PREF)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.10 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type ATOMIC_AGGREGATE)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.11 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format For well-known attributes and for optional non-transitive attributes the Partial bit MUST be set to 0. (Note: Here we test with the path attribute type MULTI_EXIT_DISC)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.12 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format The lower-order four bits of the Attribute Flags octet are unused. They MUST be zero when sent and MUST be ignored when received. (Note: Here we test that DUT sends UPDATE message with lower-order four bits of the ORIGIN Attribute Flags octets set to 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.13 **MUST**	RFC 4271, Sect. 4.3, p 16, UPDATE Message Format									
	UPDATE Message Format The lower-order four bits of the Attribute Flags octet are unused. They MUST be zero when sent and MUST be ignored when received. (Note: Here we test that DUT ignores lower-order four bits of the ORIGIN Attribute Flag after receiving an UPDATE 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-13.14 **MUST**	RFC 4271, Sect. 4.3, p 17, UPDATE Message Format									
	UPDATE Message Format ORIGIN is a well-known mandatory attribute that defines the origin of the path information. The data octet can assume the following value: 2 INCOMPLETE - Network Layer Reachability Information learned by some other means.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.15 **MUST**	RFC 4271, Sect. 4.3, p 18, UPDATE Message Format									
	UPDATE Message Format ATOMIC_AGGREGATE is a well-known discretionary attribute of length 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-13.16 **MUST**	RFC 4271, Sect. 4.3, p 18, UPDATE Message Format									
	UPDATE Message Format AGGREGATOR is an optional transitive attribute of length 6.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-14.1 **MUST**	RFC 4271, Sect. 4.4, p 21, KEEPALIVE Message Format RFC 4271, Sect. 4.2, p 13, OPEN Message Format									
	KeepAlive Message Format KEEPALIVE messages MUST NOT be sent more frequently than one per second. The Hold Time MUST be either zero or at least three seconds.									
	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: FAIL	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict
	FreeBSD 10.3: unpredict	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict	FreeBSD 10.3: unpredict	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict
ANVL-BGPPLUS-15.1 **MUST**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes BGP implementations MUST recognize all well-known attributes (Note : This test checks for External Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.2 **MUST**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes BGP implementations MUST recognize all well-known attributes (Note : This test checks for Internal Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-15.3 **MUST**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Some of the well-known attributes are mandatory and must be included in every UPDATE message that contains NLRI.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.4 **MUST**	NEGATIVE RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Some of the well-known attributes are mandatory and must be included in every UPDATE message that contains NLRI. This test checks for EBGP									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.5 **MUST**	NEGATIVE RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Some of the well-known attributes are mandatory and must be included in every UPDATE message that contains NLRI. This test checks for IBGP									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.6 **MUST**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Once a BGP peer has updated any well-known attributes, it MUST pass these attributes in any updates it transmits to its peers. (Note: This test verifies AS_PATH as well-known attribute)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.7 **SHOULD**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Paths with unrecognized transitive optional attributes SHOULD 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-15.8 **SHOULD**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes If a path with unrecognized transitive optional attribute is accepted and passed along to other BGP peers, then the unrecognized transitive optional attribute of that path MUST be passed along with the path to other BGP peers									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.9 **SHOULD**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes If a path with unrecognized transitive optional attribute is accepted and passed along to other BGP peers, then the unrecognized transitive optional attribute of that path MUST be passed along with the path to other BGP peers with the Partial bit in the Attribute Flags octet set to 1.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.10 **MUST**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes Unrecognized non-transitive optional attributes must be quietly 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.11 **MUST**	RFC 4271, Sect. 5, p 24, Path Attributes									
	Path Attributes Unrecognized non-transitive optional attributes must not be passed along to other BGP peers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-15.12 **MAY**	RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes New transitive optional attributes may be attached to the path by the originator or by any other AS (BGP Speaker) in the path. (Note: This test checks the case when originator attaches the transitive optional attribute AGGREGATOR)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-15.13 **MAY**	NEGATIVE RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes If new transitive optional attributes are not attached by the originator, the Partial bit in the Attribute Flags octet is set to 1.									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-15.14 **MUST**	NEGATIVE RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes The sender of an UPDATE message should order path attributes within the UPDATE message in ascending order of attribute type. The receiver of an UPDATE message MUST be prepared to handle path attributes within the UPDATE message that are out of order.									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-15.15 **MUST**	NEGATIVE RFC 4271, Sect. 5, p 23, Path Attributes									
	Path Attributes The same attribute (attribute with the same type) can not appear more than once within the path Attributes field of a particular UPDATE 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-16.1 **MUST**	RFC 4271, Sect. 5.1.2, p 24, AS_PATH									
	AS_PATH When a given BGP speaker advertises the route to an internal peer, the advertising speaker SHALL not modify the AS_PATH attribute associated with the 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-16.2 **MUST**	RFC 4271, Sect. 5.1.2, p 24-25, AS_PATH									
	AS_PATH When a given BGP speaker advertises the route to an external peer, then the advertising speaker updates the AS_PATH attribute as follows If the first path segment of the AS_PATH is of type AS_SEQUENCE, the local system shall prepend its own AS number as the last element of the sequence.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-16.3 **MUST**	RFC 4271, Sect. 5.1.2, p 25, AS_PATH									
	AS_PATH If the first path segment of the AS_PATH of the route to be Updated is of type AS_SET, the local system shall prepend a new path segment of type AS_SEQUENCE to the AS_PATH, including its own AS number in that segment.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-16.4 **MUST**	RFC 4271, Sect. 5.1.2, p 25, AS_PATH									
	AS_PATH When a BGP speaker originates a route then the originating speaker shall include an empty AS_PATH attribute in all UPDATE messages sent to internal peers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-16.5 **MUST**	RFC 4271, Sect. 5.1.2, p 25, AS_PATH									
	AS_PATH When a BGP speaker originates a route then the originating speaker shall include its own AS number in a path segment of type AS_SEQUENCE in the AS_PATH attribute of all UPDATE messages sent to an external peer.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-17.1 **MAY**	RFC 4271, Sect. 5.1.3, p 25-26, NEXT_HOP									
	NEXT_HOP When sending a message to an external peer X, and the peer is one IP hop away from the speaker: ... the BGP speaker can use for the NEXT_HOP attribute an interface address of the internal peer router (or the internal router) through which the announced network is reachable for the speaker, provided that peer X shares a common subnet with this address.									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-17.2 **SHOULD**	RFC 4271, Sect. 5.1.3, p 26, NEXT_HOP									
	NEXT_HOP - Otherwise, if the route being announced was learned from an external peer, the speaker can use in the NEXT_HOP attribute an IP address of any adjacent router (known from the received NEXT_HOP attribute) that the speaker itself uses for local route calculation, provided that peer X shares a common subnet with this address. This is a second form of "third party" NEXT_HOP attribute.									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-17.3 **MUST**	NEGATIVE RFC 4271, Sect5.1.3, p 27, NEXT_HOP									
	NEXT_HOP A route originated by a BGP speaker SHALL NOT be advertised to a peer using an address of that peer as NEXT_HOP. (Note : Here we test that DUT does not accept an Update Message advertising a route with next hop set to an interface address of DUT which is in the same subnet as the peer sending the Update)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-17.4 **MAY**	NEGATIVE RFC 4271, Sect5.1.3, p 27, NEXT_HOP									
	NEXT_HOP A route originated by a BGP speaker SHALL NOT be advertised to a peer using an address of that peer as NEXT_HOP. (Note : Here we test that DUT does not accept an Update Message advertising a route with next hop set to an interface address of DUT which is not in the same subnet as the peer sending the Update)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-18.1 **SHOULD**	RFC 4271, Sect. 5.1.4, p 27, MULTI_EXIT_DISC									
	MULTI_EXIT_DISC All other factors being equal, the exit or entry points with lower metric SHOULD be preferred.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-18.2 **MAY**	RFC 4271, Sect. 5.1.4, p 28, MULTI_EXIT_DISC									
	MULTI_EXIT_DISC If received over EBGP, the MULTI_EXIT_DISC attribute MAY be propagated over IBGP to other BGP speakers within the same AS									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-18.3 **MUST**	RFC 4271, Sect. 5.1.4, p 27, MULTI_EXIT_DISC									
	MULTI_EXIT_DISC The MULTI_EXIT_DISC attribute received from a neighboring AS MUST NOT be propagated to other neighboring ASs.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-18.4 **MUST**	RFC 4271, Sect. 5.1.4, p 27-28, MULTI_EXIT_DISC									
	MULTI_EXIT_DISC A BGP speaker MUST IMPLEMENT a mechanism based on local configuration which allows the MULTI_EXIT_DISC attribute to be removed from a route. If a BGP speaker is configured to remove the MULTI_EXIT_DISC attribute from a route, then this removal MUST be done prior to determining the degree of preference of the route and performing route selection (Note : In this test, we test if DUT removes MED on configuration and treats the update as having lowest MED)									
	Ubuntu 16.04: unpredict	Ubuntu 16.04: FAIL	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-18.5 **MAY**	RFC 4271, Sect. 5.1.4, p 28, MULTI_EXIT_DISC									
	MULTI_EXIT_DISC An implementation MAY also (based on local configuration) alter the value of the MULTI_EXIT_DISC attribute received over an external link.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-19.1 **MUST**	RFC 4271, Sect. 5.1.5, p 28, LOCAL_PREF									
	LOCAL_PREF LOCAL_PREF is a well-known attribute that SHALL be included in all UPDATE messages that a given BGP speaker sends to the other internal peers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-19.2 **MUST**	RFC 4271, Sect. 5.1.5, p 28, LOCAL_PREF									
	LOCAL_PREF A BGP speaker SHALL calculate the degree of preference for each external route based on the locally configured policy, and include the degree of preference when advertising a route to its internal peers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-19.3 **MUST**	RFC 4271, Sect. 5.1.5, p 28, LOCAL_PREF									
	LOCAL_PREF The higher degree of preference MUST be preferred.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-19.4 **MUST**	RFC 4271, Sect. 5.1.5, p 28, LOCAL_PREF									
	LOCAL_PREF A BGP speaker MUST NOT include the LOCAL_PREF attribute in UPDATE messages that it sends to external peers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-19.5 **MUST**	RFC 4271, Sect. 5.1.5, p 28, LOCAL_PREF									
	LOCAL_PREF If the LOCAL_PREF attribute in an UPDATE message is received from an external peer, then this attribute MUST be ignored by the receiving speaker.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-20.1 **SHOULD**	RFC 4271, Sect. 5.1.6, p 29 ATOMIC_AGGREGATE									
	ATOMIC_AGGREGATE A BGP speaker that receives a route with the ATOMIC_AGGREGATE attribute SHOULD NOT remove the attribute from the route when propagating it to other speakers.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-21.1 **MUST**	NEGATIVE RFC 4271, Sect. 4.5, p 20, NOTIFICATION message format									
	BGP Error Handling The BGP4 Connection is closed immediately after sending a NOTIFICATION 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-21.2 **MUST**	NEGATIVE RFC 4271, Sect. 6, p 29, BGP Error Handling									
	BGP Error Handling If no Error Subcode is specified in an Error message, then a zero must be used.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-21.3 **MUST**	RFC 4271, Sect. 6, p 31, BGP Error Handling									
	BGP Error Handling The phrase "the BGP4 Connection is closed" means that the transport protocol connection has been closed.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-21.4 **MUST**	RFC 4271, Sect. 6, p 29, BGP Error Handling									
	BGP Error Handling When "the BGP4 Connection is closed" then before the invalid routes are deleted from the system advertises to its peers either withdraws for the routes marked as invalid, or the new best routes before the invalid routes are deleted from the system.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-21.5 **MUST**	NEGATIVE RFC 4271, Sect. 6, p 29, BGP Error Handling									
	BGP Error Handling Unless specified explicitly, the Data field of the NOTIFICATION message that is sent to indicate an error is empty.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-22.1 **MUST**	NEGATIVE RFC 4271, Sect. 6.1, p 30, Message Header error handling									
	Message Header Error Handling If the Marker field of the message header is not the expected one, then a synchronization error has occurred and the Error Subcode is set to Connection Not Synchronized.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-22.2 **MUST**	NEGATIVE RFC 4271, Sect. 6.1, p 30, Message Header error handling									
	Message Header Error Handling If the Length field of an OPEN message is less than the minimum length of the OPEN message, then the Error Subcode is set to Bad Message Length. The Data field contains the erroneous Length field.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-22.3 **MUST**	NEGATIVE RFC 4271, Sect. 6.1, p 30, Message Header error handling									
	Message Header Error Handling If the Length field of an UPDATE message is less than the minimum length of the UPDATE message, then the Error Subcode is set to Bad Message Length. The Data field contains the erroneous Length field.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-22.4 **MUST**	NEGATIVE RFC 4271, Sect. 6.1, p 30, Message Header error handling									

	`Message Header Error Handling`
	`If the Length field of a KEEPALIVE message is not equal to 19 then`
	`the Error Subcode is set to Bad Message Length. The Data field`
	`contains the erroneous Length field.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-22.5 **MUST**	NEGATIVE RFC 4271, Sect. 6.1, p 30, Message Header error handling									

	`Message Header Error Handling`
	`If the Type field of the message header is not recognized, then the`
	`Error Subcode is set to Bad Message Type. The Data field contains`
	`the erroneous Type field.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-23.1 **MUST**	NEGATIVE RFC 4271, Sect. 6.2, p 31, OPEN message error handling									

	`Open Message Error Handling`
	`If the Autonomous System field of the OPEN message is unacceptable,`
	`then the Error Subcode is set to Bad Peer AS.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-23.3 **MUST**	NEGATIVE RFC 4271, Sect. 6.2, p 32, OPEN message error handling									

	`Open Message Error Handling`
	`If the BGP Identifier field of the OPEN message is syntactically`
	`incorrect, then the Error Subcode is set to Bad BGP Identifier.`
	`Syntactic correctness means that the BGP Identifier field represents`
	`a valid IP host address.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-23.4 **MUST**	NEGATIVE RFC 4271, Sect. 6.2, p 32, OPEN message error handling									

	`Open Message Error Handling`
	`If one of the Optional Parameters in the OPEN message is not`
	`recognized, then the Error Subcode MUST be set to Unsupported`
	`Optional Parameters.`									

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

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-25.1 **SHOULD**	NEGATIVE RFC 4271, Sect. 6.4, p 33, NOTIFICATION message error handling									
	`Notification Message Error Handling` `If a peer sends a NOTIFICATION message, and there is an error in that message, such as an unrecognized Error Code or Error Subcode, it should be noticed, logged locally, and brought to the attention of the administration of the peer.`									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-26.1 **MAY**	RFC 4271, Sect. 6.7, p 34, Cease									
	`Error Code Cease` `In absence of any fatal errors (that are indicated in this section), a BGP peer may choose at any given time to close its BGP4 Connection by sending the NOTIFICATION message with Error Code Cease.`									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-26.2 **MUST**	NEGATIVE RFC 4271, Sect. 6.7, p 34, Cease									

	`Error Code Cease`
	`The Cease NOTIFICATION message must not be used when a fatal error indicated by this section does exist.`
	`(Note : This test checks the case when the error is in message Header)`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-26.3 **MUST**	NEGATIVE RFC 4271, Sect. 6.7, p 34, Cease									

	`Error Code Cease`
	`The Cease NOTIFICATION message must not be used when a fatal error indicated by this section does exist.`
	`(Note : This test checks the case when the error is in OPEN 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-26.4 **MUST**	NEGATIVE RFC 4271, Sect. 6.7, p 34, Cease									

	`Error Code Cease`
	`The Cease NOTIFICATION message must not be used when a fatal error indicated by this section does exist.`
	`(This test checks the case when the error is in UPDATE Message fields)`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-27.1 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	Connection Collision Detection In case when a connection collision is detected, if the value of the local BGP Identifier is less than the remote one, the local system closes BGP4 Connection that already exists (the one that is already in the OpenConfirm state), and accepts BGP4 Connection initiated by the remote system.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-27.2 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	Connection Collision Detection In case when a connection collision is detected, if the value of the local BGP Identifier is greater than the remote one, the local system closes newly created BGP4 Connection, and continues to use the existing one (the one that is already in the OpenConfirm state).									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-27.3 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	Connection Collision Detection Unless allowed via configuration, a connection collision with an existing BGP4 Connection that is in Established state causes closing of the newly created connection.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-27.4 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	Connection Collision Detection Note that a connection collision cannot be detected with connections that are in Idle, or Connect, or Active states. (Note: This test is for Connect state)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-27.5 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	Connection Collision Detection Note that a connection collision cannot be detected with connections that are in Idle, or Connect, or Active states. (This test is for Active State)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-27.6 **MUST**	RFC 4271, Sect. 6.8, p 35, Connection collision detection									
	`Connection Collision Detection` `Closing the BGP4 Connection (that results from the collision resolution procedure) is accomplished by sending the NOTIFICATION message with the Error Code Cease.`									
	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-28.1 **MUST**	NEGATIVE RFC 4271, Sect. 6.2, p 30, OPEN message error handling RFC 4271, Sect. 7, p 35, BGP Version Negotiation									

	`BGP Version Negotiation`
	`If the version number contained in the Version field of the received OPEN message is not supported then Data field contains a 2-octet unsigned integer, which indicates the largest locally supported version number less than the version the remote BGP peer bid.`
	`If an open attempt fails with an Error Code OPEN Message Error, and an Error Subcode Unsupported Version Number, then if the two peers do support one or more common versions, then they will rapidly determine the highest common version.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.1 **MUST**	RFC 4271, Sect. 8.2.2, p 52, BGP Finite State machine									
	`BGP Finite State Machine` `At Idle state in response to the Manual Start event the local system initiates a TCP connection to other BGP peer.`									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.2 **MUST**	RFC 4271, Sect. 8.2.2, p 52, BGP Finite State machine									
	`BGP Finite State Machine` `At idle state in response to the Manual Start event the local system starts the ConnectRetry timer with initial value.`									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.3 **MUST**	RFC 4271, Sect. 8.2.2, p 52, BGP Finite State machine									
	`BGP Finite State Machine` `At idle state in response to the Manual Start event the local system listens for a connection that may be initiated by the remote BGP peer`									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-29.4 **MUST**	RFC 4271, Sect. 8.2.2, p 54, BGP Finite State machine									
	BGP Finite State Machine In response to the ConnectRetryTimer_Expires event, the local system: - restarts the ConnectRetryTimer									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.5 **MAY**	RFC 4271, Sect. 8.2.2, p 58, BGP Finite State machine									
	BGP Finite State Machine While in Active state in response to the ConnectRetry timer expired event : - continues to listen for TCP connection that may be initiated by remote BGP peer									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.6 **MUST**	RFC 4271, Sect. 8.2.2, p 62, BGP Finite State machine									
	BGP Finite State Machine Start event is ignored in the OpenSent state.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.7 **MUST**	NEGATIVE RFC 4271, Sect. 8.2.2, p 63, BGP Finite State machine									
	BGP Finite State Machine In state OpenSent if the Hold Timer expires, the local system sends NOTIFICATION message with Error Code Hold Timer Expired.									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict
	FreeBSD 10.3: unpredict	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: unpredict	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: pass	FreeBSD 10.3: FAIL	FreeBSD 10.3: unpredict	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-29.8 **MUST**	RFC 4271, Sect. 8.2.2, p 63, BGP Finite State machine									
	BGP Finite State Machine In OpenSent state if a TcpConnectionFails event is received, the local system: - closes the BGP4 Connection									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.9 **MAY**	RFC 4271, Sect. 8.2.2, p 63, BGP Finite State machine									
	BGP Finite State Machine In OpenSent state if a TcpConnectionFails event (Event18) is received, the local system: - continues to listen for a connection that may be initiated by the remote BGP peer									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-29.10 **MUST**	RFC 4271, Sect. 8.2.2, p 64, BGP Finite State machine									
	BGP Finite State Machine At OpenSent state if there are no errors in the OPEN message, the local system: - sends a KEEPALIVE message, and - sets a KeepaliveTimer									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.11 **MUST**	RFC 4271, Sect. 8.2.2, p 66, BGP Finite State machine									
	BGP Finite State Machine Any start event is ignored in the OpenConfirm state.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.12 **MUST**	RFC 4271, Sect. 8.2.2, p 66, BGP Finite State machine									
	BGP Finite State Machine In OpenConfirm state in response to a ManualStop event initiated by the operator, the local system: - sends the NOTIFICATION message with Cease									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.13 **MUST**	RFC 4271, Sect. 8.2.2, p 66, BGP Finite State machine									
	BGP Finite State Machine In OpenConfirm state in response to a ManualStop event initiated by the operator, the local system: - changes its state to Idle.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.14 **MUST**	RFC 4271, Sect. 8.2.2, p 70, BGP Finite State machine									
	BGP Finite State Machine Any start event is ignored in the Established state.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-29.15 **MUST**	RFC 4271, Sect. 8.2.2, p 71, BGP Finite State machine									
	BGP Finite State Machine In the Established state, if the KeepaliveTimer_Expires event occurs the local system: - sends a KEEPALIVE message, and - restarts its KeepaliveTimer unless the negotiated HoldTime value is 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-29.16 **MUST**	NEGATIVE RFC 4271, Sect. 8.2.2, p 73, BGP Finite State machine									
	BGP Finite State Machine In the Established state, if the local system receives an UPDATE or KEEPALIVE message, it restarts its Hold Timer, if the negotiated Hold Time value 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-30.1 **MAY**	NEGATIVE RFC 4271, Sect. 9, p 74, UPDATE Message Handling									
	Update Message Handling An UPDATE message may be received only in the Established state. (Note : This test checks by sending Update Message immediately after TCP connection is estabilised)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-30.2 **MAY**	NEGATIVE RFC 4271, Sect. 9, p 74, UPDATE Message Handling									
	Update Message Handling An UPDATE message may be received only in the Established state. (This test checks by sending Update Message in OpenConfirm state)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-31.1 **SHOULD**	NEGATIVE RFC 4271, Sect. 9.1.2, p 77 Phase 2: Route Selection									
	Phase 2: Route Selection If the AS_PATH attribute of a BGP route contains an AS loop, the BGP route should be excluded from the Phase 2 decision function.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-31.2 **MUST**	RFC 4271, Sect. 9.1.2, p 78 Phase 2: Route Selection									
	Phase 2: Route Selection Notice that even though BGP routes do not have to be installed in the Routing Table with the immediate next hop(s), implementations MUST take care that before any packets are forwarded along a BGP route, its associated NEXT_HOP address is resolved to the immediate (directly connected) next-hop address and this address (or multiple addresses) is finally used for actual packet forwarding.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-31.3 **MUST**	RFC 4271, Sect. 9.1.2, p 77 Phase 2: Route Selection									
	Phase 2: Route Selection The local speaker MUST determine the immediate next-hop address from the NEXT_HOP attribute of the selected route (see Section 5.1.3). If either the immediate next hop or the IGP cost to the NEXT_HOP (where the NEXT_HOP is resolved through an IGP route) changes, Phase 2 Route Selection MUST be performed again.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-31.4 **MUST**	RFC 4271, Sect. 9.1.2, p 77, Phase 2: Route Selection									
	Phase 2: Route Selection The local speaker MUST determine the immediate next-hop address from the NEXT_HOP attribute of the selected route (see Section 5.1.3). If either the immediate next hop or the IGP cost to the NEXT_HOP (where the NEXT_HOP is resolved through an IGP route) changes, Phase 2 Route Selection MUST be performed again.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-31.5 **SHOULD**	RFC 4271, Sect. 9.1.2, p 78, Phase 2: Route Selection									
	Phase 2: Route Selection Unresolvable routes SHALL be removed from the Loc-RIB and the routing table. However, corresponding unresolvable routes SHOULD be kept in the Adj-RIBs-In (in case they become resolvable).									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-32.1 **MUST**	NEGATIVE RFC 4271, Sect. 9.1.2.1, p 78, Route Resolvability Condition RFC 4271, Sect. 9.1.2.1, p 78-79, Route Resolvability Condition									
	Route Resolvability Condition 1. A route Rte1, referencing only the intermediate network address, is considered resolvable if the Routing Table contains at least one resolvable route Rte2 that matches Rte1"s intermediate network address and is not recursively resolved (directly or indi- rectly) through Rte1. Mutually recursive routes (routes resolving each other or themselves), also fail the resolvability check. It is also important that implementations do not consider feasible routes that would become unresolvable if they were installed in the Routing Table even if their NEXT_HOPs are resolvable using the cur- rent contents of the Routing Table (an example of such routes would be mutually recursive routes).									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-33.1 **MUST**	RFC 4271, Sect. 9.1.2.2, p 77-78, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`a) Remove from consideration all routes which are not tied for`
	`having the smallest number of AS numbers present in their AS_PATH`
	`attributes. Note, that when counting this number, an AS_SET counts`
	`as 1, no matter how many ASs are in the set.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-33.2 **MUST**	RFC 4271, Sect. 9.1.2.2, p 77-78, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`b) Remove from consideration all routes which are not tied for`
	`having the lowest Origin number in their Origin attribute.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-33.3 **MUST**	RFC 4271, Sect. 9.1.2.2, p 78, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`Routes which do not have the MULTI_EXIT_DISC attribute are considered`
	`to have the lowest possible MULTI_EXIT_DISC value.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-33.4 **MUST**	RFC 4271, Sect. 9.1.2.2, p 79, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`d) If at least one of the candidate routes was received via EBGP,`
	`remove from consideration all routes which were received via IBGP.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-33.5 **MUST**	RFC 4271, Sect. 9.1.2.2, p 79, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`e) Remove from consideration any routes with less-preferred inte-`
	`rior cost.  The interior cost of a route is determined by calcu-`
	`lating the metric to the NEXT_HOP for the route using the 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-33.6 **MUST**	RFC 4271, Sect. 9.1.2.2, p 79, Breaking Ties (Phase 2)									

	`Breaking Ties (Phase 2)`
	`f) Remove from consideration all routes other than the route that`
	`was advertised by the BGP speaker whose BGP Identifier has the`
	`lowest value.`									

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

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-33.7 **MUST**	RFC 4271, Sect. 9.1.2.2, p 79, Breaking Ties (Phase 2)									
	`Breaking Ties (Phase 2)` `g) Prefer the route received from the lowest peer address.`									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-34.1 **SHOULD**	RFC 4271, Sect. 9.1.4, p 81, Overlapping Routes									

	`Overlapping Routes`
	`If a more specific route is later withdrawn, the set of destinations`
	`described by the overlap will still be reachable using the less`
	`specific 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-34.2 **MUST**	RFC 4271, Sect. 9.1.4, p 81, Overlapping Routes									

	`Overlapping Routes`
	`If both a less and a more specific route are accepted, then the`
	`Decision Process MUST install both the less and the more specific`
	`routes if it does not aggregate the two routes.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-34.3 **MUST**	RFC 4271, Sect. 9.1.4, p 81 Overlapping Routes									

	`Overlapping Routes`
	`In particular, a route that carries ATOMIC_AGGREGATE attribute`
	`MUST NOT be de-aggregated`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-35.1 **MUST**	RFC 4271, Sect. 9.2, p 81, Update-Send Process									

	`Update-Send Process`
	`When a BGP speaker receives an UPDATE message from an internal peer,`
	`the receiving BGP speaker SHALL NOT re-distribute the routing`
	`information contained in that UPDATE message to other internal peers`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-36.1 **MUST**	RFC 4271, Sect. 9.2.1.1, p 83, Frequency of Route Advertisement,									

	`Frequency of Route Advertisement`
	`If new routes are selected multiple times while awaiting the`
	`expiration of MinRouteAdvertisementInterval, the last route selected`
	`SHALL be advertised at the end of MinRouteAdvertisementInterval.`									

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	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: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-37.1 **MUST**	RFC 4271, Sect. 9.2.1.2, p 83 Frequency of Route Origination RFC 4271, Sect. 10, p 88 BGP Timers									

	`Frequency of Route Origination`
	`The parameter MinASOriginationIntervalTimer determines the minimum`
	`amount of time that must elapse between successive advertisements of`
	`UPDATE messages that report changes within the advertising BGP`
	`speaker"s own autonomous systems.`
	`The suggested default value for the MinASOriginationIntervalTimer-`
	`Timer on EBGP4 Connections is 30 seconds.`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-37.2 **MUST**	RFC 4271, Sect. 9.2.1.2, p 83 Frequency of Route Origination RFC 4271, Sect. 10, p 88 BGP Timers									

	`Frequency of Route Origination`
	`The parameter MinASOriginationIntervalTimer determines the minimum`
	`amount of time that must elapse between successive advertisements of`
	`UPDATE messages that report changes within the advertising BGP`
	`speaker"s own autonomous systems.`
	`The suggested default value for the MinASOriginationIntervalTimer-`
	`Timer on IBGP4 Connections is 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-38.1 **SHOULD**	RFC 4271, Sect. 9.2.2.2, p 84, Aggregating Routing Information									

	`Aggregating Routing Information`
	`Routes that have different MULTI_EXIT_DISC attribute SHALL NOT be`
	`aggregated`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-38.2 **SHOULD**	RFC 4271, Sect. 9.2.2.2, p 84, Aggregating Routing Information									

	`Aggregating Routing Information`
	`If the aggregated route has an AS_SET as the first element in its`
	`AS_PATH attribute, then the router that originates the route SHOULD`
	`NOT advertise the MULTI_EXIT_DISC attribute with this route.`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-38.3 **MUST**	RFC 4271, 9.2.2.2, p 84, Aggregating Routing Information									

	`Aggregating Routing Information`
	`When aggregating routes that have different NEXT_HOP attribute,`
	`the NEXT_HOP attribute of the aggregated route SHALL identify`
	`an interface on the BGP speaker that performs the aggregation.`									

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

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-38.4 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information,									

	`Aggregating Routing Information`
	`If at least one route among routes that are aggregated has ORIGIN`
	`with the value INCOMPLETE, then the aggregated route must have the`
	`ORIGIN attribute with the value INCOMPLETE.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-38.5 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information,									

	`Aggregating Routing Information`
	`If at least one route among routes that are aggregated has ORIGIN with`
	`the value EGP, then the aggregated route must have the ORIGIN`
	`attribute with the value EGP.`									

	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-38.6 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information									

	`Aggregating Routing Information`
	`If routes to be aggregated have identical AS_PATH attributes,`
	`then the aggregated route has the same AS_PATH attribute as`
	`each individual route.`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-38.7 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information									

	`Aggregating Routing Information`
	`- all tuples of type AS_SEQUENCE in the aggregated AS_PATH SHALL`
	`appear in all of the AS_PATH in the initial set of routes to be`
	`aggregated.`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-38.8 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information									

	`Aggregating Routing Information`
	`- all tuples of type AS_SET in the aggregated AS_PATH SHALL`
	`appear in at least one of the AS_PATH in the initial set`
	`(they may appear as either AS_SET or AS_SEQUENCE types).`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-38.9 **MUST**	RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information									
	Aggregating Routing Information - for any tuple X of type AS_SEQUENCE in the aggregated AS_PATH which precedes tuple Y in the aggregated AS_PATH, X precedes Y in each AS_PATH in the initial set which contains Y, regardless of the type of Y.									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-38.10 **MUST**	NEGATIVE RFC 4271, Sect. 9.2.2.2, p 85, Aggregating Routing Information									
	Aggregating Routing Information - No tuple of type AS_SET with the same value SHALL appear more than once in the aggregated AS_PATH. An implementation may choose any algorithm which conforms to these rules. At a minimum a conformant implementation SHALL be able to perform the following algorithm that meets all of the above conditions:									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-38.11 **SHOULD**	RFC 4271, Sect. 9.2.2.2, p 86, Aggregating Routing Information,									
	Aggregating Routing Information If at least one of the routes to be aggregated has ATOMIC_AGGREGATE path attribute, then the aggregated route shall have this attribute as well.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-38.12 **MUST**	RFC 4271, Sect. 9.2.2.2, p 86, Aggregating Routing Information									
	Aggregating Routing Information Any AGGREGATOR attributes from the routes to be aggregated MUST NOT be included in the aggregated route. The BGP speaker per- forming the route aggregation MAY attach a new AGGREGATOR attribute (see Section 5.1.7).									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-39.1 **MUST**	RFC 4271, 9.3, p 86, Route Selection Criteria									
	Route Selection Criteria - If the local AS appears in the AS path of the new route being considered, then that new route can not be viewed as better than any other route (provided that the speaker is configured to accept such routes). If such a route were ever used, a routing loop could result.									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-40.1 **SHOULD**	RFC 4271, Sect. Appendix - F.1, p 91, Multiple Networks Per Message,									

	`Multiple Networks per Message`
	`The BGP protocol allows multiple address prefixes with the same`
	`Path attributes to be specified in one 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.1 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									

	`Revised Update Message Error Handling According To Draft`
	`If any attribute has Attribute Flags that conflict with the`
	`Attribute Type Code, then the error SHOULD be logged, and the`
	`Attribute Flags MUST be reset to the correct value.  The UPDATE`
	`message MUST continue to be processed.`
	`(This test checks for mandatory well-known attributes, Optional Bit`
	`and External Peer)`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.2 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									

	`Revised Update Message Error Handling According To Draft`
	`If any attribute has Attribute Flags that conflict with the`
	`Attribute Type Code, then the error SHOULD be logged, and the`
	`Attribute Flags MUST be reset to the correct value.  The UPDATE`
	`message MUST continue to be processed.`
	`(This test checks for mandatory well-known attributes, Optional Bit`
	`and Internal Peer)`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.3 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									

	`Revised Update Message Error Handling According To Draft`
	`If any attribute has Attribute Flags that conflict with the`
	`Attribute Type Code, then the error SHOULD be logged, and the`
	`Attribute Flags MUST be reset to the correct value.  The UPDATE`
	`message MUST continue to be processed.`
	`(This test checks for mandatory well-known attributes, Transitional Bit`
	`and External Peer)`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.4 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									

	`Revised Update Message Error Handling According To Draft`
	`If any attribute has Attribute Flags that conflict with the`
	`Attribute Type Code, then the error SHOULD be logged, and the`
	`Attribute Flags MUST be reset to the correct value.  The UPDATE`
	`message MUST continue to be processed.`
	`(This test checks for mandatory well-known attributes, Transitional Bit`
	`and Internal Peer)`									

	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.5 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for mandatory well-known attributes, Partial Bit and External Peer)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.6 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (This test checks for mandatory well-known attributes, Partial Bit and Internal Peer)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.7 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for MULTI_EXIT_DISC (optional non-transitive) attribute and for Optional Bit)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.8 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for MULTI_EXIT_DISC (optional non-transitive) attribute and for transitive Bit)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.9 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for MULTI_EXIT_DISC (optional non-transitive) attribute and for Partial Bit)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.10 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for ATOMIC AGGREGATE (well known discretionary) attribute and for Optional Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.11 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for ATOMIC AGGREGATE (well known discretionary) attribute and for Transitive Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.12 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for ATOMIC AGGREGATE (well known discretionary) attribute and for Partial Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.13 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged, and the Attribute Flags MUST be reset to the correct value. The UPDATE message MUST continue to be processed. (NOTE:This test only checks for Processing This test checks for AGGREGATOR (optional transitive) attribute and for Optional Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.14 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (Note: This test checks by sending incorrect length for ORIGIN attribute)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.15 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (Note: This test checks by sending incorrect length for MULTI_EXIT_DISC attribute)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.16 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (Note: This test checks by sending incorrect length for LOCAL_PREF attribute)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.17 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "attribute discard" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ATOMIC_AGGREGATE and AGGREGATOR. (Note: This test checks by sending incorrect length for ATOMIC_AGGREGATE attribute)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.18 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (This test checks for well-known mandatory attributes missing.For IBGP)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.19 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (This test checks for well-known mandatory attributes missing.For EBGP)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.20 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (NOTE:ORIGIN attribute has an undefined value)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.21 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft The approach of "treat-as-withdraw" MUST be used for the error handling of the cases described in Section 6.3 of [RFC4271] that specify a session reset and involve any of the following attributes: ORIGIN, AS_PATH, NEXT_HOP, MULTI_EXIT_DISC, and LOCAL_PREF. (NOTE:AS_PATH attribute is syntactically incorrect)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.22 **MUST**	draft-ietf-idr-error-handling-01.txt Section 5.1 Page 6 " AGGREGATOR"									
	Revised Update Message Error Handling According To Draft The AGGREGATOR attribute SHALL be considered malformed if any of the following applies: Its length is not 6 (when the "4-octet AS number capability" is not advertised to, or not received from the peer [RFC4893]). Its length is not 8 (when the "4-octet AS number capability" is both advertised to, and received from the peer). An UPDATE message with a malformed AGGREGATOR attribute SHALL be handled using the approach of "attribute discard". NOTE:In this test "length is not 6"									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.23 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If an attribute appears more than once in an UPDATE message, then all the occurrences of the attribute other than the first one SHALL be discarded and the UPDATE message continue to be processed. (This test checks for EBGP)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.24 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft If an attribute appears more than once in an UPDATE message, then all the occurrences of the attribute other than the first one SHALL be discarded and the UPDATE message continue to be processed. (This test checks for IBGP)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.25 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft When multiple malformed attributes exist in an UPDATE message, if the same approach (either "treat-as-withdraw" or "attribute discard") is specified for the handling of these malformed attributes, then the specified approach MUST be used. Otherwise "treat-as-withdraw" MUST be used. (NOTE:ORIGIN and AS_PATH attribute field malformed and Same approach specified for both the malformed attributes i.e "treat as withdraw")									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-41.26 **MUST**	draft-ietf-idr-error-handling-01.txt Section 2 Page 4 " Revision to Base Specification"									
	Revised Update Message Error Handling According To Draft When multiple malformed attributes exist in an UPDATE message, if the same approach (either "treat-as-withdraw" or "attribute discard") is specified for the handling of these malformed attributes, then the specified approach MUST be used. Otherwise "treat-as-withdraw" MUST be used. (NOTE:ORIGIN, AS_PATH and AGGREGATOR attribute field malformed and Same approach not specified for all the malformed attributes i.e "treat as withdraw")									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.27 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 4 Page 5 "Operational Considerations"									
	Revised Update Message Error Handling According To Draft When a malformed attribute is indeed detected over an IBGP session, we RECOMMEND that routes with the malformed attribute be identified and traced back to the ingress router in the network where the routes were sourced or received externally, and then a filter be applied on the ingress router to prevent the routes from being sourced or received. This will help maintain routing consistency in the network. (NOTE:ORIGIN, AS_PATH attribute field malformed Checking for filter applied or not on ingress router over an IBGP session to prevent route for which malformed attribute received earlier)									
	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	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: untested	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL	FreeBSD 10.3: FAIL
ANVL-BGPPLUS-41.28 **MUST**	draft-ietf-idr-error-handling-01.txt Section 3 Page 5 "Parsing of NLRI Fields" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft To facilitate the determination of the NLRI field in an UPDATE with a malformed attribute, the MP_REACH or MP_UNREACH attribute (if present) SHOULD be encoded as the very first path attribute in an UPDATE as recommended by [RFC4760bis]. An implementation, however, MUST still be prepared to receive these fields in any position. (NOTE:ANVL checks if DUT receive these field in any position MP_REACH_NLRI attribute encoded as last path attribute in the UPDATE 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-41.29 **MUST**	draft-ietf-idr-error-handling-01.txt Section 3 Page 5 "Parsing of NLRI Fields" UPDATE message error handling									
	Revised Update Message Error Handling According To Draft To facilitate the determination of the NLRI field in an UPDATE with a malformed attribute, the MP_REACH or MP_UNREACH attribute (if present) SHOULD be encoded as the very first path attribute in an UPDATE as recommended by [RFC4760bis]. An implementation, however, MUST still be prepared to receive these fields in any position. (NOTE:ANVL checks if DUT receive these field in any position MP_UNREACH_NLRI attribute encoded as last path attribute in the UPDATE 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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-42.1 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged. (NOTE:Error Log Checking) (This test checks for mandatory well-known attributes, Optional Bit and External Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-42.2 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged. (NOTE:Error Log Checking) (This test checks for mandatory well-known attributes, Optional Bit and External Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-42.3 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged. (NOTE:Error Log Checking) (Note : This test checks for mandatory well-known attributes, Transitive Bit and Internal Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-42.4 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged. (NOTE:Error Log Checking) (Note : This test checks for mandatory well-known attributes, Partial Bit and Internal Peer)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Release 2.0	3.0-dev 2017-04-25	Master 2017-05-17	3.0-dev 2017-05-24	Master 2017-06-02	Master 2017-06-26	3.0-dev 2017-06-30	Master 2017-07-20	Release 3.0-rc1	Master 2017-08-16
ANVL-BGPPLUS-42.5 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged (NOTE:Error Log Checking) (Note : This test checks for MULTI_EXIT_DISC (optional non-transitive) attribute and for Optional Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-42.6 **SHOULD**	draft-ietf-idr-error-handling-01.txt Section 2 Page 3 " Revision to Base Specification" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check If any attribute has Attribute Flags that conflict with the Attribute Type Code, then the error SHOULD be logged (NOTE:Error Log Checking) (Note : This test checks for ATOMIC_AGGREGATE (Well known discretionary) attribute and for Optional Bit)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGPPLUS-42.7 **MUST**	draft-ietf-idr-error-handling-01.txt Section 4 Page 6 "Operational Considerations" UPDATE message error handling									
	Update Message Error Handling According To New Draft Atrribute Flag error log check Because of these potential issues, a BGP speaker MUST provide debugging facilities to permit issues caused by a malformed attribute to be diagnosed. At a minimum, such facilities MUST include logging an error listing the NLRI involved, and containing the entire malformed UPDATE message when such an attribute is detected. (Note : This test checks sending Wrong Attribute flags conflicting with Attribute type Code for well-known madatory attribute, and error lists NLRI involved)									
	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	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass