RFC Compliance Test Report

BGP4 Results

www.OpenSourceRouting.org

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
Type	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR
Commit ID	1a664f5	3e71b5d	e61a754	3d7746c	b84ccd4	f731a65	bade23d	f30a732	f92f83b	dceb5f8
Commit Date	2017-03-08	2017-04-02	2017-04-04	2017-04-25	2017-05-16	2017-05-24	2017-06-02	2017-06-27	2017-07-01	2017-07-21
ANVL-BGP4-1.1	ANVL, setup verification									

MUST	`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: 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
ANVL-BGP4-1.2	ANVL, setup verification									

MUST	`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: 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
ANVL-BGP4-1.3	ANVL, setup verification									

MUST	`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: 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
ANVL-BGP4-1.4	ANVL, setup verification									

MUST	`ANVL, Setup Verification`
	`Router forwards new Update 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: 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
ANVL-BGP4-2.1	RFC4271, Sect. 4, p 11, Message Formats									

MUST	`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: 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
ANVL-BGP4-3.1	RFC4271, Sect.4.2, page 13, OPEN message format									

MUST	`OPEN Message Format`
	`After a TCP connection is established, the first message sent by each`
	`side is an 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-3.2 **MUST**	RFC4271, Sect.4.2, page 13, OPEN message format									

	`OPEN Message Format`
	`If the OPEN message is acceptable, a KEEPALIVE message`
	`confirming the OPEN is sent back.`									

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-3.3 **MUST**	NEGATIVE RFC4271, Sect. 4.2, p 13, 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: unpredict
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-3.4 **MUST**	RFC4271, Sect. 4.2, p 13, 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: 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
ANVL-BGP4-3.5 **MUST**	NEGATIVE RFC4271, Sect. 4.2, p 13, OPEN Message Format RFC4271, Sect. 6.2, p 32, 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: 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
ANVL-BGP4-3.6 **MUST**	NEGATIVE RFC4271, Sect. 4.2, p 14, 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: 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

<table>
<tr><th></th><th>Master
2017-03-07</th><th>Release
2.0</th><th>Master
2017-04-03</th><th>3.0-dev
2017-04-25</th><th>Master
2017-05-17</th><th>3.0-dev
2017-05-24</th><th>Master
2017-06-02</th><th>Master
2017-06-26</th><th>3.0-dev
2017-06-30</th><th>Master
2017-07-20</th></tr>
<tr><td rowspan="4">ANVL-BGP4-3.7

<b>MUST</b></td><td colspan="10">NEGATIVE
RFC4271, Sect. 4.2, p 14,
OPEN Message Format</td></tr>
<tr><td colspan="10">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)</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
<tr><td rowspan="4">ANVL-BGP4-4.1

<b>MAY</b></td><td colspan="10">RFC4271, Sect. 4.3, p 15,
UPDATE Message Format</td></tr>
<tr><td colspan="10">UPDATE Message Format
An UPDATE message MAY simultaneously advertise a feasible route and
withdraw multiple unfeasible routes from service.</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
<tr><td rowspan="4">ANVL-BGP4-4.2

<b>MUST</b></td><td colspan="10">RFC4271, Sect. 4.3, p 17,
UPDATE Message Format</td></tr>
<tr><td colspan="10">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)</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
<tr><td rowspan="4">ANVL-BGP4-4.3

<b>MUST</b></td><td colspan="10">RFC4271, Sect. 4.3, p 17,
UPDATE Message Format</td></tr>
<tr><td colspan="10">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)</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
<tr><td rowspan="4">ANVL-BGP4-4.4

<b>MUST</b></td><td colspan="10">RFC4271, Sect. 4.3, p 17,
UPDATE Message Format</td></tr>
<tr><td colspan="10">UPDATE Message Format
For well-known attributes, the Transitive bit must be set to 1.
(Note: Here we test with the path attribute type NEXT_HOP)</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
<tr><td rowspan="4">ANVL-BGP4-4.5

<b>MUST</b></td><td colspan="10">RFC4271, Sect. 4.3, p 17,
UPDATE Message Format</td></tr>
<tr><td colspan="10">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)</td></tr>
<tr><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td><td>Ubuntu 16.04: pass</td></tr>
<tr><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: untested</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td><td>FreeBSD 10.3: pass</td></tr>
</table>

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-4.6 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.7 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.8 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.9 **MUST**	RFC4271, Sect. 4.3, p 17, 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 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: 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
ANVL-BGP4-4.10 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.11 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-4.12 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.13 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.14 **MUST**	RFC4271, Sect. 4.3, p 17, 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: 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
ANVL-BGP4-4.15 **MUST**	RFC4271, Sect. 4.3, p 18, 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: 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
ANVL-BGP4-4.16 **MUST**	RFC4271, Sect. 4.3, p 19, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-4.17 **MUST**	RFC4271, Sect. 4.3, p 19, 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: 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
ANVL-BGP4-4.18 **MAY**	RFC4271, Sect.5.1.7 p.30, AGGREGATOR									

	`UPDATE Message Format`
	`A BGP speaker which performs route aggregation MAY add the AGGREGATOR`
	`attribute which SHALL contain its own AS number and IP 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: 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
ANVL-BGP4-5.1 **MUST**	RFC4271, Sect. 4.4, p 21, KEEPALIVE 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: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-6.1 **MUST**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.2 **MUST**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.3 **MUST**	RFC4271, Sect. 5, p 24, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-6.4 **MUST**	NEGATIVE RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.5 **MUST**	NEGATIVE RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.6 **MUST**	NEGATIVE RFC4271, Sect. 5, p 24, Path Attributes									

	`Path Attributes`
	`Once a BGP peer has updated any well-known attributes, it MUST pass`
	`these attributes to its peers in any updates it transmits.`
	`(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: 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
ANVL-BGP4-6.7 **SHOULD**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.8 **SHOULD**	RFC4271, Sect. 5, p 24, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-6.9 **SHOULD**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.10 **MUST**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.11 **MUST**	RFC4271, 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: 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
ANVL-BGP4-6.12 **MAY**	RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-6.14 **MUST**	NEGATIVE RFC4271, Sect. 5, p 24, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-6.15 **MUST**	NEGATIVE RFC4271, Sect. 5, p 24, 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: 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
ANVL-BGP4-7.1 **MUST**	RFC4271, Sect. 5.1.2, p 25, 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: 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
ANVL-BGP4-7.2 **MUST**	RFC4271, Sect. 5.1.2, p 25-26, 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: 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
ANVL-BGP4-7.3 **MUST**	RFC4271, Sect. 5.1.2, p 26, 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: 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
ANVL-BGP4-7.4 **MUST**	RFC4271, Sect. 5.1.2, p 26, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-7.5 **MUST**	RFC4271, Sect. 5.1.2, p 26, 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: 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
ANVL-BGP4-8.1 **MAY**	RFC4271, Sect5.1.3, p 26, NEXT_HOP									
	NEXT_HOP When sending a message to an internal peer, if the route is not locally originated the BGP speaker SHOULD NOT modify the NEXT_HOP attribute, unless it has been explicitly configured to announce its own IP address as 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: 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
ANVL-BGP4-8.2 **MAY**	RFC4271, Sect. 5.1.3, p 27, 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 for the NEXT_HOP attribute, provided that peer X shares a common subnet with this 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: 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
ANVL-BGP4-8.3 **SHOULD**	RFC4271, Sect. 5.1.3, p 27, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-8.4 **MUST**	NEGATIVE RFC4271, Sect 5.1.3, p 28, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-8.5 **MAY**	NEGATIVE RFC4271, Sect 5.1.3, p 28, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-9.1 **SHOULD**	RFC4271, Sect. 5.1.4, p 28, 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: 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
ANVL-BGP4-9.2 **MAY**	RFC4271, 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: 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
ANVL-BGP4-9.3 **MUST**	RFC4271, Sect. 5.1.4, p 28, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-9.4 **MUST**	RFC4271, Sect. 5.1.4, p 28-29, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-9.5 **MAY**	RFC4271, Sect. 5.1.4, p 29, 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 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: 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
ANVL-BGP4-10.1 **MUST**	RFC4271, Sect. 5.1.5, p 29, 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: 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
ANVL-BGP4-10.2 **MUST**	RFC4271, Sect. 5.1.5, p 29, 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: 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
ANVL-BGP4-10.3 **MUST**	RFC4271, Sect. 5.1.5, p 29, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-10.4	RFC4271, Sect. 5.1.5, p 29, LOCAL_PREF									
MUST	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: 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
ANVL-BGP4-10.5	RFC4271, Sect. 5.1.5, p 29, LOCAL_PREF									
MUST	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: 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
ANVL-BGP4-11.1	RFC4271, Sect. 5.1.6, p 30 ATOMIC_AGGREGATE									
SHOULD	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: 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
ANVL-BGP4-12.1	NEGATIVE RFC4271, Sect. 4.5, p 21, NOTIFICATION message format									
MUST	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: 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
ANVL-BGP4-12.2	NEGATIVE RFC4271, Sect. 6, p 30, BGP Error Handling									
MUST	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: 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
ANVL-BGP4-12.3	RFC4271, Sect. 6, p 30, BGP Error Handling									
MUST	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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-12.4 **MUST**	RFC4271, Sect. 6, p 30, BGP Error Handling									
	BGP Error Handling When "the BGP4 Connection is closed" then before the invalid routes are deleted from the system, it 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: 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
ANVL-BGP4-12.5 **MUST**	NEGATIVE RFC4271, Sect. 6, p 30, 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: 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
ANVL-BGP4-13.1 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, Message Header error handling									
	Message Header Error Handling If the Marker field of the message header is not as expected, 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: 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
ANVL-BGP4-13.2 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, Message Header error handling									
	Message Header Error Handling If the Length field of the message header is less than 19 or greater than 4096 then the Error Subcode MUST be set to Bad Message Length. The Data field MUST contain 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: 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
ANVL-BGP4-13.3 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, 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 MUST be set to Bad Message Length. The Data field MUST contain 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-13.4 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, 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 MUST be set to Bad Message Length. The Data field MUST contain 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: 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
ANVL-BGP4-13.5 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, 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 MUST be set to Bad Message Length. The Data field MUST contain 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: 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
ANVL-BGP4-13.6 **MUST**	NEGATIVE RFC4271, Sect. 6.1, p 31, Message Header error handling									
	Message Header Error Handling If the Type field of the message header is not recognized, then the Error Subcode MUST be set to Bad Message Type. The Data field MUST contain 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: 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
ANVL-BGP4-14.1 **MUST**	NEGATIVE RFC4271, Sect. 6.2, p 32, OPEN message error handling									
	Open Message Error Handling If the Autonomous System field of the OPEN message is unacceptable, then the Error Subcode MUST be 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: 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
ANVL-BGP4-14.2 **MAY**	NEGATIVE RFC4271, Sect. 6.2, p 32, OPEN message error handling									
	Open Message Error Handling If the Hold Time field of the OPEN message is Unacceptable, then the Error Subcode MUST be set to Unacceptable Hold Time. An implementation MAY reject any proposed Hold Time.									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-14.3 **MUST**	NEGATIVE RFC4271, 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 MUST be set to Bad BGP Identifier.`
	`Syntactic correctness means that the BGP Identifier field represents`
	`a valid unicast 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: 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
ANVL-BGP4-14.4 **MUST**	NEGATIVE RFC4271, 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: 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
ANVL-BGP4-15.1 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If the Withdrawn Routes Length or Total Attribute Length is too`
	`large (i.e., if Withdrawn Routes Length + Total Attribute Length + 23`
	`exceeds the message Length), then the Error Subcode MUST be set to`
	`Malformed Attribute List.`									

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-15.2 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If any recognized attribute has Attribute Flags that conflict with the`
	`Attribute Type Code, then the Error Subcode MUST be set to Attribute`
	`Flags Error. The Data field MUST contain the erroneous attribute (type,`
	`length and value).`
	`(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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.3 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value).`
	`(This test checks for mandatory well-known attributes, Optional 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: 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
ANVL-BGP4-15.4 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value).`
	`(Note : This test checks for mandatory well-known attributes, Transitive 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: 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
ANVL-BGP4-15.5 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value).`
	`(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: 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
ANVL-BGP4-15.6 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Error Data field MUST contain the erroneous attribute (type, length and value).`
	`(Note : This test checks for mandatory well-known attributes, Partial 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.7 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (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: 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
ANVL-BGP4-15.8 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute. (type, length and value). (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: 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
ANVL-BGP4-15.9 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling.									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (This test checks for MULTI_EXIT_DISC (optional non-transitive) 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: 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
ANVL-BGP4-15.10 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (Note : This test checks for MULTI_EXIT_DISC (optional non-transitive) 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.11 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (Note : This test checks for ATOMIC_AGGREGATE (well-known discretionary) attribute, and 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: 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
ANVL-BGP4-15.12 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (This test checks for ATOMIC_AGGREGATE (well-known discretionary) attribute, and 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: 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
ANVL-BGP4-15.13 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (This test checks for ATOMIC_AGGREGATE (well-known discretionary) attribute, 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: 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
ANVL-BGP4-15.14 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode MUST be set to Attribute Flags Error. The Data field MUST contain the erroneous attribute (type, length and value). (This test checks for AGGREGATOR (optional transitive) attribute, and 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.15 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Error Data field MUST contain the erroneous attribute (type, length and value). (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: 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
ANVL-BGP4-15.16 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Error Data field MUST contain the erroneous attribute (type, length and value). (Note: This test checks by sending incorrect length for NEXT_HOP 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: 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
ANVL-BGP4-15.17 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Data field MUST contain the erroneous attribute (type, length and value). (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: 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
ANVL-BGP4-15.18 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Data field MUST contain the erroneous attribute (type, length and value). (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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.19 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Error Data field MUST contain the erroneous attribute (type, length and value). (This test checks by sending incorrect length for ATOMIC_AGGREGATE 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: 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
ANVL-BGP4-15.20 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									
	Update Message Error Handling If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode MUST be set to Attribute Length Error. The Data field MUST contain the erroneous attribute (type, length and value). (This test checks by sending incorrect length for AGGREGATOR 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: 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
ANVL-BGP4-15.21 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 33, UPDATE message error handling									
	Update Message Error Handling If any of the well-known mandatory attributes are not present, then the Error Subcode MUST be set to Missing Well-known Attribute. The Data field MUST contain the Attribute Type Code of the missing, well-known attribute. (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: 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
ANVL-BGP4-15.22 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 33, UPDATE message error handling									
	Update Message Error Handling If any of the mandatory well-known attributes are not present, then the Error Subcode MUST be set to Missing Well-known Attribute. The Data field MUST contain the Attribute Type Code of the missing well-known attribute. (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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.23 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 33, UPDATE message error handling									

	`Update Message Error Handling`
	`If any of the mandatory well-known attributes are not recognized,`
	`then the Error Subcode MUST be set to Unrecognized Well-known Attribute.`
	`The Data field MUST contain the unrecognized attribute (type, length and`
	`value).`									

	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: 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
ANVL-BGP4-15.24 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 32, UPDATE message error handling									

	`Update Message Error Handling`
	`If the ORIGIN attribute has an undefined value, then the Error`
	`Subcode MUST be set to Invalid Origin Attribute. The Data field`
	`MUST contain the unrecognized attribute (type, length and 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: 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
ANVL-BGP4-15.25 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 33, UPDATE message error handling									

	`Update Message Error Handling`
	`If the NEXT_HOP attribute field is syntactically incorrect,`
	`then the Error Subcode MUST be set to Invalid NEXT_HOP Attribute.`
	`The Data field MUST contain the incorrect attribute`
	`(type, length and 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: 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
ANVL-BGP4-15.26 **MUST**	NEGATIVE RFC4271, Sect.6.3, page 33, UPDATE message error handling.									

	`Update Message Error Handling`
	`If the NEXT_HOP attribute is semantically incorrect, the error`
	`SHOULD be logged, and the the route SHOULD be ignored.  In this`
	`case, a NOTIFICATION message SHOULD not be sent.`									

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-15.27 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 33, UPDATE message error handling									

	`Update Message Error Handling`
	`If the AS_PATH attribute is syntactically incorrect, then the Error`
	`Subcode MUST be set to Malformed 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-15.28 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 34, UPDATE message error handling									
	Update Message Error Handling If an optional attribute is recognized, then the value of this attribute MUST be checked. If an error is detected, the attribute MUST be discarded, and the Error Subcode MUST be set to Optional Attribute Error. The Data field MUST contain the attribute (type, length and value). (This test checks for AGGREGATOR 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: 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
ANVL-BGP4-15.29 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 35, UPDATE message error handling									
	Update Message Error Handling If any attribute appears more than once in the UPDATE message, then the Error Subcode MUST be set to Malformed Attribute List. (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: 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
ANVL-BGP4-15.30 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 34, UPDATE message error handling									
	Update Message Error Handling If any attribute appears more than once in the UPDATE message, then the Error Subcode MUST be set to Malformed Attribute List. (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: 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
ANVL-BGP4-15.31 **MUST**	NEGATIVE RFC4271, Sect. 6.3, p 34, UPDATE message error handling									
	Update Message Error Handling The NLRI field in the UPDATE message is checked for syntactic validity. If the field is syntactically incorrect, then the Error Subcode MUST be set to Invalid Network 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: 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
ANVL-BGP4-15.32 **MUST**	RFC4271, Sect. 6.3, p 34, UPDATE message error handling									
	Update Message Error Handling An UPDATE message that contains correct path attributes, but no NLRI, SHALL be treated as a valid UPDATE message. (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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-16.1 **SHOULD**	NEGATIVE RFC4271, Sect. 6.4, p 34, NOTIFICATION message error handling									

	`Notification Message Error Handling`
	`If a peer sends a NOTIFICATION message, and the receiver of the message`
	`detects an an error in that message, ....`
	`Any such error (e.g., an unrecognized Error Code or Error Subcode)`
	`SHOULD be noticed, logged locally, and brought to the attention of the`
	`administration of the 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: 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
ANVL-BGP4-17.1 **MUST**	NEGATIVE RFC4271, Sect. 6.5, p 34, OPEN Message Format									

	`Hold Timer Error Handling`
	`If a system does not receive successive KEEPALIVE and/or UPDATE`
	`and/or NOTIFICATION messages within the period specified in the`
	`Hold Time field of the OPEN message, then the NOTIFICATION`
	`message with Hold Timer Expired Error Code is sent and the BGP`
	`connection is 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: 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
ANVL-BGP4-18.1 **MAY**	RFC4271, Sect. 6.7, p 35, 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 BGP 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: 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
ANVL-BGP4-18.2 **MUST**	NEGATIVE RFC4271, Sect. 6.7, p 35, 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: 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
ANVL-BGP4-18.3 **MUST**	NEGATIVE RFC4271, Sect. 6.7, p 35, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-18.4 **MUST**	NEGATIVE RFC4271, Sect. 6.7, p 35, 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: 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
ANVL-BGP4-19.1 **MUST**	RFC4271, Sect. 6.8, p 36, 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 BGP 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: 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
ANVL-BGP4-19.2 **MUST**	RFC4271, Sect. 6.8, p 36, 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 (the one associated with the newly`
	`received OPEN message), 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: 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
ANVL-BGP4-19.3 **MUST**	RFC4271, Sect. 6.8, p 36, 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: 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
ANVL-BGP4-19.4 **MUST**	RFC4271, Sect. 6.8, p 36, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-19.5 **MUST**	RFC4271, Sect. 6.8, p 36, 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: 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
ANVL-BGP4-19.6 **MUST**	RFC4271, Sect. 6.8, p 36, 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: unpredict	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: unpredict
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-20.1 **MUST**	NEGATIVE RFC4271, Sect. 6.2, p 31, OPEN message error handling RFC4271, Sect. 7, p 36, BGP Version Negotiation									
	BGP Version Negotiation If the version number contained in the Version field of the received OPEN message is not supported, then the Error Subcode MUST be set to Unsupported Version Number. The Data field is a a 2-octet unsigned integer, which indicates the largest, locally supported version number less than the version the remote BGP peer bid (as indicated in the received OPEN message) If an open attempt fails with an Error Code OPEN Message Error, and an Error Subcode Unsupported Version Number....... 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: 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
ANVL-BGP4-21.1 **MUST**	RFC4271, Sect. 8.2.2, p 53, 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: 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
ANVL-BGP4-21.2 **MUST**	RFC4271, Sect. 8.2.2, p 53, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-21.3 **MAY**	RFC4271, Sect. 8.2.2, p 59, 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 a remote BGP peer									
	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: unpredict	Ubuntu 16.04: FAIL	Ubuntu 16.04: unpredict
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict
ANVL-BGP4-21.4 **MUST**	RFC4271, Sect. 8.2.2, p 63, 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: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-21.5 **MUST**	NEGATIVE RFC4271, Sect. 8.2.2, p 64, 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: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-21.6 **MUST**	RFC4271, Sect. 8.2.2, p 64, 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: 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
ANVL-BGP4-21.7 **MAY**	RFC4271, Sect. 8.2.2, p 64, 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: 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: FAIL

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-21.8 **MUST**	RFC4271, Sect. 8.2.2, p 65, 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: 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
ANVL-BGP4-21.9 **MUST**	RFC4271, Sect. 8.2.2, p 67, 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: 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
ANVL-BGP4-21.10 **MUST**	RFC4271, Sect. 8.2.2, p 67, 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: FAIL
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: unpredict
ANVL-BGP4-21.11 **MUST**	RFC4271, Sect. 8.2.2, p 67, 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: 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
ANVL-BGP4-21.12 **MUST**	RFC4271, Sect. 8.2.2, p 71, 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: 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
ANVL-BGP4-21.13 **MUST**	RFC4271, Sect. 8.2.2, p 72, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-21.14 **MUST**	NEGATIVE RFC4271, Sect. 8.2.2, p 74, 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: 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
ANVL-BGP4-22.1 **MAY**	NEGATIVE RFC4271, Sect. 9, p 75, 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 establised)									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-22.2 **MAY**	NEGATIVE RFC4271, Sect. 9, p 75, 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: 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
ANVL-BGP4-22.3 **MUST**	RFC4271, Sect.9 p.75, UPDATE Message Handling									
	Update Message Handling If the UPDATE message contains a non-empty WITHDRAWN ROUTES field, the previously advertised routes whose destinations (expressed as IP prefixes) are contained in this field SHALL be removed from the Adj-RIB-In.									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-23.1 **MUST**	RFC4271, Sect.9.1, page 76, Decision Process									
	Calculation of Degree of Preference Phase 1 is responsible for calculating the degree of preference for each route received from 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-23.2 **MUST**	RFC 4271, Sect.9.1.1, p.77, Phase 1: Calculation of Degree of Preference									

	`Calculation of Degree of Preference`
	`If the route is learned from an internal peer, the value of LOCAL_PREF`
	`attribute shall be taken as the degree of preference.`									

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-24.1 **SHOULD**	NEGATIVE RFC4271, Sect. 9.1.2, p 78 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: 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
ANVL-BGP4-24.2 **MUST**	RFC4271, Sect. 9.1.2, p 79 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: 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
ANVL-BGP4-24.3 **MUST**	RFC4271, Sect. 9.1.2, p 78, 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: 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
ANVL-BGP4-24.4 **MUST**	RFC4271, Sect. 9.1.2, p 78, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-24.5 **SHOULD**	RFC4271, Sect. 9.1.2, p 79, 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: 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
ANVL-BGP4-24.6 **MUST**	RFC4271, Sect.9.1.2 p.78, Phase 2: Route Selection									
	Phase 2: Route Selection If the NEXT_HOP attribute of a BGP route depicts an address that is not resolvable, or it would become unresolvable if the route was installed in the routing table the BGP route MUST 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: 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
ANVL-BGP4-25.1 **MUST**	NEGATIVE RFC4271, Sect. 9.1.2.1, p 79, Route Resolvability Condition RFC4271, Sect. 9.1.2.1, p 79-80, Route Resolvability Condition RFC4271, Sect. 9.1.2, p 79, Phase 2: Route Selection									
	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 indirectly) 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 current contents of the Routing Table (an example of such routes would be mutually recursive routes). AND Unresolved routes SHALL be removed from the Loc-RIB and 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: 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
ANVL-BGP4-26.1 **MUST**	RFC4271, Sect. 9.1.2.2, p 80, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-26.2 **MUST**	RFC4271, Sect. 9.1.2.2, p 80, 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: 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
ANVL-BGP4-26.3 **MUST**	RFC4271, Sect. 9.1.2.2, p 81, Breaking Ties (Phase 2)									
	Breaking Ties (Phase 2) Remove from consideration routes with less-preferred MULTI_EXIT_DISC 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: 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
ANVL-BGP4-26.4 **MUST**	RFC4271, Sect. 9.1.2.2, p 80, Breaking Ties (Phase 2)									
	Breaking Ties (Phase 2) MULTI_EXIT_DISC is only comparable between routes learned from the same neighboring AS. (This test checks the case when two routes are received from two different ASs, having different MULTI_EXIT_DISC values)									
	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: 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
ANVL-BGP4-26.5 **MUST**	RFC4271, Sect. 9.1.2.2, p 80 Breaking Ties (Phase 2)									
	Breaking Ties (Phase 2) MULTI_EXIT_DISC is only comparable between routes learned from the same neighboring AS. (This test checks the case when two routes are received from same AS, having different MULTI_EXIT_DISC values)									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-26.6 **MUST**	RFC4271, Sect. 9.1.2.2, p 80, 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: 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
ANVL-BGP4-26.7 **MUST**	RFC4271, Sect. 9.1.2.2, p 82, 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: 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

	Master	Release	Master	3.0-dev	Master	3.0-dev	Master	Master	3.0-dev	Master
	2017-03-07	2.0	2017-04-03	2017-04-25	2017-05-17	2017-05-24	2017-06-02	2017-06-26	2017-06-30	2017-07-20
ANVL-BGP4-26.8 **MUST**	RFC4271, Sect. 9.1.2.2, p 82, Breaking Ties (Phase 2)									
	Breaking Ties (Phase 2) e) Remove from consideration any routes with less-preferred interior cost. The interior cost of a route is determined by calculating 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: 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
ANVL-BGP4-26.9 **MUST**	RFC4271, Sect. 9.1.2.2, p 82, 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: 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
ANVL-BGP4-26.10 **MUST**	RFC4271, Sect. 9.1.2.2, p 82, 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: 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
ANVL-BGP4-27.1 **SHOULD**	RFC4271, Sect. 9.1.4, p 83, 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: 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
ANVL-BGP4-27.2 **MUST**	RFC4271, Sect. 9.1.4, p 83-84, Overlapping Routes									
	Overlapping Routes If both a less and a more specific route are accepted, then the Decision Process MUST install, in Loc-RIB, either both the less and the more specific routes or aggregate the two routes and install, in Loc-RIB, the aggregated route, provided that both routes have the same value of the NEXT_HOP 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-28.1 **MUST**	RFC4271, Sect. 9.2, p 84, 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: 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
ANVL-BGP4-29.1 **MUST**	RFC4271, Sect. 9.2.1.1, p 85, 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 MinRouteAdvertisementIntervalTimer.									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-30.1 **MUST**	RFC4271, Sect. 9.2.1.1, p 85 Frequency of Route Advertisement RFC4271, Sect. 10, p 90 BGP Timers									
	Frequency of Route Origination The parameter MinRouteAdvertisementIntervalTimer determines the minimum amount of time that must elapse between an advertisement and/or withdrawal of routes to a particular destination by a BGP speaker to a peer. The suggested default value for the MinRouteAdvertisementIntervalTimer- Timer is 30 seconds 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: 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
ANVL-BGP4-30.2 **MUST**	RFC4271, Sect. 9.2.1.2, p 85 Frequency of Route Origination RFC4271, Sect. 10, p 90 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 15 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: 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
ANVL-BGP4-31.1 **SHOULD**	RFC4271, Sect. 9.2.2.2, p 87, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-31.2 **SHOULD**	RFC4271, Sect. 9.2.2.2, p 87, 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: 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
ANVL-BGP4-31.3 **MAY**	RFC4271, Sect.9.2.2.2, p 87 Aggregating Routing Information									
	Aggregating Routing Information Path attributes that have different type codes can not be aggregated together. (Here we test that the DUT has aggregated two routes having the same type code and all the mandatory attributes are present)									
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: untested	FreeBSD 10.3: pass	FreeBSD 10.3: pass	FreeBSD 10.3: pass
ANVL-BGP4-31.4 **MUST**	RFC4271, 9.2.2.2, p 87, 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: 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
ANVL-BGP4-31.5 **MUST**	RFC4271, Sect. 9.2.2.2, p 87, 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: 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
ANVL-BGP4-31.6 **MUST**	RFC4271, Sect. 9.2.2.2, p 87, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-31.7 **MUST**	RFC4271, Sect. 9.2.2.2, p 87, 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: 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
ANVL-BGP4-31.8 **MUST**	RFC4271, Sect. 9.2.2.2, p 88, 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: 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
ANVL-BGP4-31.9 **MUST**	RFC4271, Sect. 9.2.2.2, p 88, 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: 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
ANVL-BGP4-31.10 **MUST**	RFC4271, Sect. 9.2.2.2, p 88, 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: 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
ANVL-BGP4-31.11 **MUST**	NEGATIVE RFC4271, Sect. 9.2.2.2, p 88, 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: 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

	Master 2017-03-07	Release 2.0	Master 2017-04-03	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
ANVL-BGP4-31.12 **SHOULD**	RFC4271, Sect. 9.2.2.2, p 89, 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: 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
ANVL-BGP4-31.13 **MUST**	RFC4271, Sect. 9.2.2.2, p 89, 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 performing 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: 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
ANVL-BGP4-32.1 **MUST**	RFC4271, 9.3, p 89, 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: 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
ANVL-BGP4-33.1 **SHOULD**	RFC4271, Sect. Appendix - F.1, p 95, 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: 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