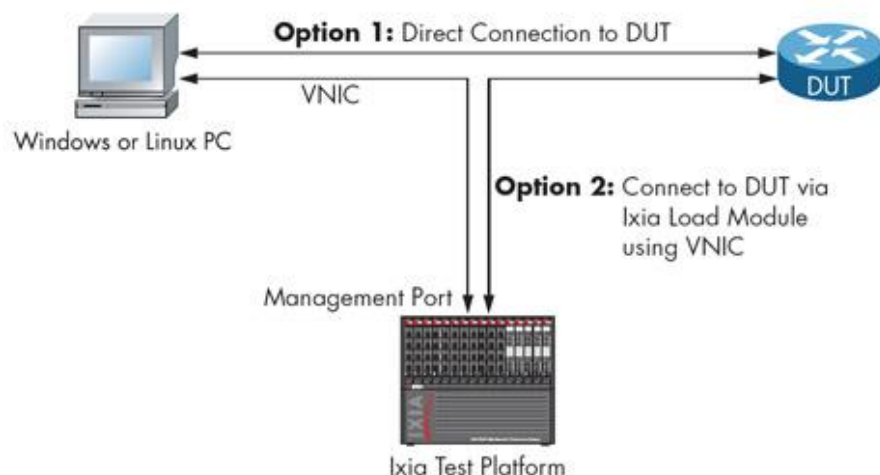


IxANVL™ – Automated Network Validation Library



Ixia's IxANVL (Automated Network Validation Library) is the industry standard for automated network/protocol validation. Developers and manufacturers of networking equipment and Internet devices rely on IxANVL to validate protocol compliance and interoperability. Many customers have chosen IxANVL for its ease-of-use, enhanced GUI, and flexible test automation capabilities. In addition, IxANVL offers a veritable universe of protocol libraries and utilities.

Though IxANVL is able to run on minimal hardware such as a PC with a Linux or Windows operating system and an Ethernet card, it is also well suited for operation on Ixia's powerful test and analysis platform via a VNIC (Virtual Network Interface Card) driver. This flexibility enables IxANVL to support all industry-standard test interfaces including 10/100/1G/10G Ethernet, ATM, serial, async, T1/E1, and POS. IxANVL provides conformance, negative, and regression testing on a vast selection of protocols including bridging, routing, PPP, TCP/IP, L4-7, IPv6, IP storage, IPsec, RMON, VPN, MPLS, voice over IP, Carrier Ethernet, and multicast.



IxANVL Benefits

Saves Time and Money

IxANVL allows vendors to verify device design during the product's entire life cycle. Problems can be identified earlier so as to prevent costly last-minute reworks. IxANVL emulates large, multi-node networks that previously were cost prohibitive — resulting in more efficient tests and quicker product release times.

Increases Confidence

IxANVL increases confidence in product quality by enabling extensive and thorough testing, performed automatically and without supervision. IxANVL's test results allow users to:

- Determine exactly where a device's protocol software does and does not meet the specification
- Observe how well the device handles traffic from non-complying network components
- Determine how new development effects existing code, via regression testing

Expands Easily

With a source code license, users can easily add new interface types, protocols, and/or test cases to their IxANVL system.

Supports More Protocols

IxANVL supports a comprehensive list of protocols, including unicast/multicast routing, bridging, IPv6, VPN, MPLS, PPP, TCP/IP, RMON, voice over IP, metro Ethernet, and IP storage.

Test Coverage

	IxANVL Test Suites	Target Protocols	Reference Specification	Test Case Count	Required Test Interfaces
IPv6 Test Suites	IPv6 Core	IPv6	RFC 2460, 2464	111	2
		IPv6CP	RFC 2472	17	1
		ICMPv6	RFC 4443	46	2
	IPv6 Advanced	NDP	RFC 4861	228	2
		Generic Packet Tunneling	RFC 2473	46	2
		AutoConfig	RFC 4862	37	2
		V6oV4	RFC 4213, 2529, 3056, 3068	66	2
		PMTU	RFC 1981	10	1
		IP Router Alert	RFC 2711	13	2
	Mobile IPv6	Home Agent	RFC 3775	159	2
		Correspondence Node	RFC 3775	153	1
		Mobile Node	RFC3775	95	2
	GRE	GRE	RFC 2890, 2784	29	2
	DHCPv6	DHCP Client	RFC 3315	103	1
		DHCP Server	RFC 3315	141	2
IPv4 Test Suites	IPv4	IPv4	RFC 791, parts of 1122, 1812	70	2
		ICMP	RFC 792, parts of 1812	32	2
	ARP	ARP	RFC 826 RFC 1027	61	2
	DHCPv4	DHCP Client	RFC 2131	90	2
		DHCP Server	RFC 2131	74	2

Routing	IP RIP	RIP	RFC 2453	53	2
		IPGW	RFC 1812, 1122	18	2
	RIPng	RIPng	RFC 2080	60	2
	OSPF Core	OSPF	RFC 1583, 2328	312	3
	OSPF Extensions	Opaque LSA, NSSA, DB Overflow, Stub Router Ext	RFC 2370, 3101, 1765, 3137	56	3
		OSPF TE	RFC 3630	54	2
	OSPFv3	OSPFv3	RFC 5340, parts of RFC 2328	328	3
	OSPF-GR	OSPFv2-GR	RFC 3623	56	2
	VRRP	VRRP	RFC 3768 RFC 5798	83	2
	BGP4 Core	BGP	RFC 4271, draft-ietf-idr-error-handling-01	217	3
	BGP4 Extensions	BGP-OSPF, Communities, Route Flap Damping, Route Reflection, Route Refresh, Confederations	RFC 1403, 1997, 2439, 2918, 4456, 5065, 1771, 4360, draft-ietf-idr-error-handling-01	150	3
	BGP Plus	BGP+ with IPv6	RFC 4271, 4760, 2545, draft-ietf-idr-error-handling-01	236	3
	OSPFv3-AF	Support of Address Families in OSPFv3	RFC 5838	37	2
	BGP 4-Byte AS	4-byte AS for BGP and BGPPlus	RFC 4893, draft-ietf-idr-error-handling-01	52	3
	ISIS	ISIS	RFC 1195, 3719, ISO/IEC 10589: 1992(E) RFC 5303 RFC 5309	237	2
	ISIS-TE	ISIS-TE	RFC 3784	31	1
	ISIS-MT	ISIS-MT	RFC 5120, RFC 5303 RFC 5309	105	3
	VRRPv6	VRRPv3 over IPv6	RFC 5798	77	2
	ISISv6	ISIS-v6	ISO/IEC 10589: 1992(E), RFC 3719, 1195, 5308, RFC 5303, RFC 5309	221	2

MPLS	MPLS	Label Encapsulation	RFC 3032	59	2
	RSVP-TE	RSVP-TE	RFC 3209, draft-ietf-mpls-rsvp-lsp-tunnel-07	87	3
	RSVP-TE	RSVP-TE P2MP	RFC 4875	48	3
	LDP	LDP	RFC 3036	329	3
	mLDP	mLDP P2MP	draft-ietf-mpls-ldp-p2mp-10	97	4
	LSP-Ping-Tr	LSP Ping and Traceroute	RFC 4379	128	2
	VCCV	Pseudo wire VCCV	RFC 5085	70	2
	L2VPN (PWE3)	PWE3-Control	RFC 4447	69	2
		PWE3-Encapsulation	RFC 4448, 4618, 4717, 4385, 4623	78	2
	VPLS	VPLS	RFC 4762	58	4
	VPLS-BGP	VPLS with BGP AD and signalling	RFC 4761	46	4
	L3 VPN	L3 VPN	RFC 4364	101	3
MPLS-TP	MPLS-TP-Y1731-CC-LD	MPLS-TP-Y1731-CC-LD	RFC 5586 (GACH), draft-bhh-mpls-tp-oam-y1731-06.txt, ITU-T-REC Y.1731-200605-I	85	1
	MPLS-TP-IETF-CC-CV-LD	MPLS-TP-IETF-CC-CV-LD	RFC 5586 (GACH), draft-ietf-mpls-loss-delay-01, draft-ietf-mpls-tp-on-demand-cv-02, draft-ietf-mpls-tp-cc-cv-rdi-03	210	1
	MPLS-TP-G.8031-APS-Y.1731	MPLS-TP-G.8031-APS-Y.1731	G.8031_Y.1342-2006-06	140	2

Multicasting Test Suites	IGMP	IGMPv2	RFC 2236	49	2
		IGMPv3	RFC 3376	153	2
	DVMRP	DVMRP	draft-ietf-idmr-dvmrp-v3-07	66	3
	PIM	Dense Mode	draft-ietf-pim-dm-new-v2-04	162	3
		Sparse Mode, SSM	RFC 4601, draft-ietf-pim-sm-bsr-12	327	3
	PIMv6	Sparse Mode	draft-ietf-pim-sm-v2-new-12, draft-ietf-pim-sm-bsr-12	283	3
	MLD	MLDv1	RFC 2710	98	2
		MLDv2	RFC 3810	202	2
High Availability	BFD	BFD Base, BFD Generic, BFD-v4v6-1hop for OSPFv2/v3, ISIS and BGP BFD-MPLS	drafts draft-ietf-bfd-base-09.txt, draft-ietf-bfd-generic-05.txt, draft-ietf-bfd-v4v6-1hop-09.txt, draft-ietf-bfd-mpls-07.txt	178	3
TCP Test Suites (See Note 1)	TCP Core	TCP	RFC 793, 1122, 2460	179	2
	TCP Advanced	Slow Start, Congestion Control, PMTU Disc, MD5	RFC 2001, 2581, 1191, 2385, 2463, 1981	57	1
	TCP High Performance	Ext for High Performance, Selective Ack	RFC 1323, 2018	48	1
UDP Test Suite	UDP	UDP	RFC 768, 1122	35	1
Layer 4-7 Test Suite	HTTP	HTTP Server	RFC 2616	346	1
	SMB3-SERVER	SMB3-SERVER	[MS-SMB2] - v20121017 Server Message Block (SMB) Protocol Versions 2 and 3	100	1
	Telnet	Telnet Client/Server	RFC 854	43	1
IP Storage Suites	iSCSI	iSCSI Target	RFC 3720	210	1
		iSCSI Initiator	RFC 3720	205	1

VPN Test Suites	IPSec AH	MD5, SHA	RFC 4301, 4302	58	2
	IPSec ESP	MD5, SHA, DES, 3DES, Blowfish, AES	RFC 4301, 4303, 2403, 2404, 2405	72	2
	IPSec IKE	ISAKMP, IKE	RFC 2407, 2408, 2409	373	2
	IPSec AH / IPv6	MD5, SHA, IPSecv6	RFC 4301, 4302	66	2
	IPSec ESP / IPv6	MD5, SHA, DES, 3DES, Blowfish, AES	RFC 4301, 4303, 2403, 2404, 2405, 2406	74	2
	IPSec IKE / IPv6	ISAKMP, IKE	RFC 2407, 2408, 2409	384	2
	IKEV2	IKEV2, DES, 3DES, AES-128, 256, 192, MD5, SHA, DH-768, 1024, 1536, 2048, 3072	RFC 4306	189	2
	L2TP	L2TP	RFC 2661	105	1
	PPTP	PPTP	draft-ietf-pppext-pptp-02	55	1
	IKEV2-IPV6	IKEV2-IPV6	RFC 4306	204	2
Automotive Ethernet	AUTO-ETH-ARP	AUTO-ETH-ARP	RFC 826 AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	52	1
	AUTO-ETH-DHCP-SERVER	AUTO-ETH-DHCP-SERVER	RFC 2131 RFC 2132	74	1
	AUTO-ETH-DHCPV4-CLIENT	AUTO-ETH-DHCPV4-CLIENT	RFC 2131, RFC 2132, AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	90	1
	AUTO-ETH-DHCPV6-CLIENT	AUTO-ETH-DHCPV6-CLIENT	RFC 3315	104	1

	AUTO-ETH-DHCPV6-SERVER	AUTO-ETH-DHCPV6-SERVER	RFC 3315	141	1
	AUTO-ETH-ICMP	AUTO-ETH-ICMP	RFC 792, RFC 1122, AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	21	1
	AUTO-ETH-ICMPV6	AUTO-ETH-ICMPV6	RFC 2463, RFC 4443	24	1
	AUTO-ETH-IP	AUTO-ETH-IP	RFC 791, RFC 1122, RFC 894, AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	29	1
	AUTO-ETH-IPV4-ACONF	AUTO-ETH-IPV4-ACONF	RFC 3927, AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	56	1
	AUTO-ETH-IPV6	AUTO-ETH-IPV6	RFC 2460 and RFC 2464	80	1
	AUTO-ETH-IPV6-AUTOCONFIG	AUTO-ETH-IPV6-AUTOCONFIG	RFC 4862	34	1
	AUTO-ETH-IPV6-MLD	AUTO-ETH-IPV6-MLD	RFC 2710	46	1
	AUTO-ETH-IPV6-NDP	AUTO-ETH-IPV6-NDP	RFC 4861	83	1
	AUTO-ETH-TCP-ADVANCED	AUTO-ETH-TCP-ADVANCED	RFC 793, RFC 2001, RFC 1191, RFC 2385, RFC 2463, RFC 1981, RFC 813, RFC 896 and AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	63	1
	AUTO-ETH-TCP-CORE	AUTO-ETH-TCP-CORE	RFC 793, RFC 1122, RFC 2460 and	177	1

	CORE		AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)		
	AUTO-ETH-UDP	AUTO-ETH-UDP	RFC 768 and AUTOSAR_SWS_Tcplp.pdf (AUTOSAR Specification of TCP/IP Stack V1.1.0 R4.1 Rev2)	35	1
PPP Test Suites	PPP	LCP, PPP, PPP in HDLC	RFC 1661, 1662	111	2
		Authentication (PAP, CHAP)	RFC 1334, 1994	37	1
	IPCP	IPCP	RFC 1332	19	2
	VJ	VJ Compression	RFC 1144	48	2
	PPPoE	PPP over Ethernet	RFC 2516	75	2
	Multilink PPP	MPPP	RFC1717, 1990	59	3
	Multilink PPP	Multi-class Extension	RFC 2686	9	3

Carrier Ethernet	MEF9	MEF9	MEF1, MEF9, Iometrix Test Plan version 1.4	247	6
	EtherCFM	Ethernet CFM	IEEE P802.1ag/D8.1 2007	246	3
	EtherOAM	Ethernet OAM	IEEE 802.3-ah-2004	166	3
	MEF OAM	MEF21 OAM	MEF 21 Abstract Test Suite for UNI Type 2	187	2
	Service OAM	Y.1731	ITU-T Y.1731 05/2006, IEEE P802.1ag/D8.1 June 8,2007	106	2
	Provider BB	PBB	IEEE 802.1ah D4.2 2005	55	2
	MEF Service OAM	MEF Service OAM	ATS for UNI Type 2 Part 3 - Service OAM	157	2
	CE2.0	CE2.0	CARRIER ETHERNET 2.0 TEST PLAN - PART 1: SERVICES ATTRIBUTES - Version 1.0 CARRIER ETHERNET 2.0 TEST PLAN - PART 2: TRAFFIC MANAGEMENT - Version 2.0	465	6
	MEF ELMI	MEF ELMI	D00063_004 ATS for UNI Type 2 Part 2 ELMI TC MEF 16	239	2
	G8031	G.8031 1:1 protection	Ethernet Automatic Protection Switching – ITU-T G8031/Y.1342	283	3
	G_8265_1	G_8265_1	IEEE 1588 Conformity Test Suite For Frequency Synchronization in Telecommunications Networks, Draft Version 2.0	348	1

Bridging	STP	802.1d	IEEE Std. 802.1D-1998	53	3
	RSTP	802.1w	IEEE Std. 802.1D-2004	126	4
	EAPOL	802.1x, MD5, TLS, TTLS	IEEE 802.1x-2004	83	3
	MSTP	802.1s	IEEE 802.1Q-2005	247	4
	LLDP	LLDP	IEEE 802.1AB 2005	104	3
	DCBX	DCBX	DCB Capability Exchange Protocol Specification (Rev 1.0), DCB Capability Exchange Protocol Base Specification (Rev 1.01)	92	1
	Mcast Snooping	IGMP/MLD Snooping	RFC 4541	42	3
	VLAN	802.1q, GMRP, GVRP	IEEE Std. 802.1Q-2005	161	4
	LACP	802.3ad	IEEE Std. 802.3-2005 Clause 43	118	4
	QinQ	QinQ	IEEE 802.1ad- 2005	127	2
	MVRP/MMRP	MVRP, MMRP	IEEE Std 802.1ak-2007 IEEE Std 802.1Q™-2005/Cor 1-2008	321	3
	OPENFLOW_1.0	OPENFLOW_1.0	OpenFlow Switch Specification Version 1.0.0 (Wire Protocol 0x01) and OpenFlow Switch Errata v1.0.1	194	5
	OPENFLOW1.3	OPENFLOW1.3	OpenFlow Switch Specification Version 1.3.2 (Wire Protocol 0x04)	341	5
VoIP	SIP	SIP Server	RFC 3261	297	2

Fiber Chanel over Ethernet (FCoE)	FIP	FCoE Initialization Protocol	FCoE Initialization Protocol, FIBRE CHANNEL BACKBONE-5, Rev 1.04, Prepared by the INCITS, T11/Project 1871-D	169	1
	FCoE FCF	FCoE FCF	FIBRE CHANNEL LINK SERVICES, Rev 2.11, Prepared by the INCITS, T11/Project 2103-D	56	1
RMON Test Suites	RMON	Ethernet General	RFC 1757 RFC 1757	116 372	1 1
Toolkits	TCP Toolkit		N/A	4	1
Toolkits	SNMP Toolkit	Toolkit & sample tests only	N/A	9	1

Note 1: TCP test suites require a connection with the device under test (DUT) from both below and above the targeted TCP layer. Connection from below the TCP layer is achieved via a traditional physical layer interface. Connection from above the TCP layer can only be achieved with a unique application called “TCP Stub,” developed by Ixia. The TCP Stub is controlled and managed remotely by Ixia TCP Test Suites. The purpose of the TCP Stub is to generate the necessary stimulus above the TCP layer required for testing. The TCP Stub is a portable C-code application bundled with TCP test suites. Customers are required to compile the TCP Stub onto their target systems.

IxANVL Benefits

Saves Time and Money

IxANVL allows vendors to verify device design during the product's entire life cycle. Problems can be identified earlier so as to prevent costly last-minute reworks. IxANVL emulates large, multi-node networks that previously were cost prohibitive — resulting in more efficient tests and quicker product release times.

Increases Confidence

IxANVL increases confidence in product quality by enabling extensive and thorough testing, performed automatically and without supervision. IxANVL's test results allow users to:

- Determine exactly where a device's protocol software does and does not meet the specification
- Observe how well the device handles traffic from non-complying network components
- Determine how new development effects existing code, via regression testing

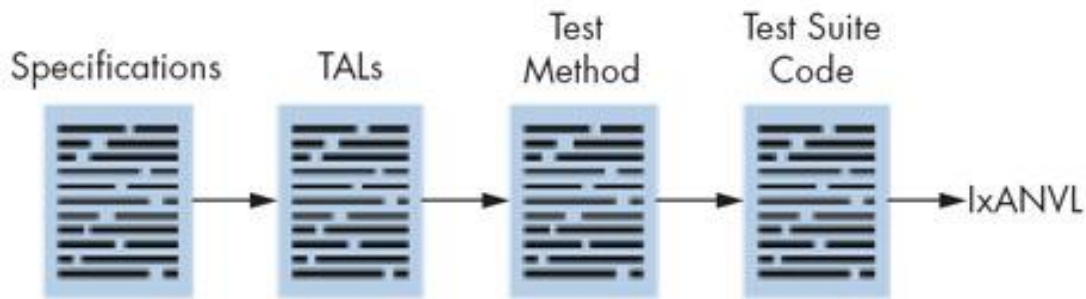
Expands Easily

With a source code license, users can easily add new interface types, protocols, and/or test cases to their IxANVL system.

Supports More Protocols

IxANVL supports a comprehensive list of protocols, including unicast/multicast routing, bridging, IPv6, VPN, MPLS, PPP, TCP/IP, RMON, voice over IP, metro Ethernet, and IP storage.

Test Methodology



IxANVL follows a rigorous test suite development process:

- Analyze a protocol specification line-by-line
- Develop a test assertion list (TAL), which is a list of testable statements
- Augment TALs with more negative tests
- Prioritize and group TALs for the test suite
- Develop a test method for each accepted test assertion

IxANVL performs continual verification of protocol standard authors or implementers during the development process.

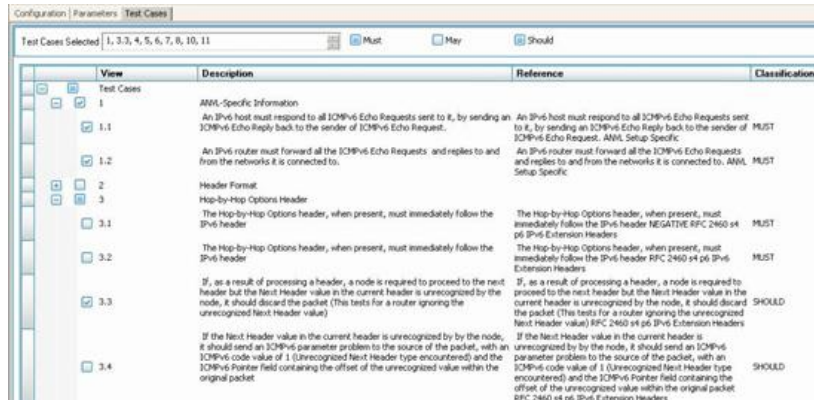
Test Configuration

The IxANVL test suite can run on a Linux or Windows PC with off-the-shelf network interface cards, or on Ixia's load modules through a virtual network interface card (VNIC) connection. The tester (PC) connects with the DUT via test interfaces. Up to four interfaces may be used, depending on the test configuration. IxANVL flexibly emulates various system topologies, and creates virtually any test scenarios for almost any DUT.

IxANVL offers both a command-line interface for test automation and a user-friendly graphical user interface, allowing intuitive test execution management and detail reporting. A batch runner is also available for scheduling regression test-run sequences.

Test Execution

IxANVL classifies test cases into three categories: MUST, SHOULD, and MAY. Tests can be selected and executed based on their categories or test topologies.



View	Description	Reference	Classification
1	ANA-Specific Information		
1.1	An IPv6 host must respond to all ICMPv6 Echo Requests sent to it, by sending an ICMPv6 Echo Reply back to the sender of ICMPv6 Echo Request.	An IPv6 host must respond to all ICMPv6 Echo Requests sent to it, by sending an ICMPv6 Echo Reply back to the sender of ICMPv6 Echo Request. ANA, Setup Specific	MUST
1.2	An IPv6 router must forward all the ICMPv6 Echo Requests and replies to and from the networks it is connected to.	An IPv6 router must forward all the ICMPv6 Echo Requests and replies to and from the networks it is connected to. ANA, Setup Specific	MUST
2	Header Format		
3	Hop-by-Hop Options Header		
3.1	The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header	The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header RFC 2460 §4 p6 IPv6 Extension Headers	MUST
3.2	The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header	The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header RFC 2460 §4 p6 IPv6 Extension Headers	MUST
3.3	If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node, it should discard the packet. (This tests for a router ignoring the unrecognized Next Header value)	If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node, it should discard the packet. (This tests for a router ignoring the unrecognized Next Header value) RFC 2460 §4 p6 IPv6 Extension Headers	SHOULD
3.4	If the Next Header value in the current header is unrecognized by the node, it should send an ICMPv6 parameter problem to the source of the packet, with an ICMPv6 code value of 1 (Unrecognized Next Header type encountered) and the ICMPv6 Pointer field containing the offset of the unrecognized value within the original packet	If the Next Header value in the current header is unrecognized by the node, it should send an ICMPv6 parameter problem to the source of the packet, with an ICMPv6 code value of 1 (Unrecognized Next Header type encountered) and the ICMPv6 Pointer field containing the offset of the unrecognized value within the original packet RFC 2460 §4 p6 IPv6 Extension Headers	SHOULD

The IxANVL test can be run using two options —GUI or command line input. In GUI mode, the user selects which test suite and test cases to run. In command line mode, the user types a command with options indicating which tests should run and the desired output level.

In the test, IxANVL sends packets to the DUT based on the test designed, and compares the received DUT packets to what was expected. After receiving these packets, IxANVL reacts according to the returned information— it may continue the test, stop the test, log an error message, or a host of other functions.

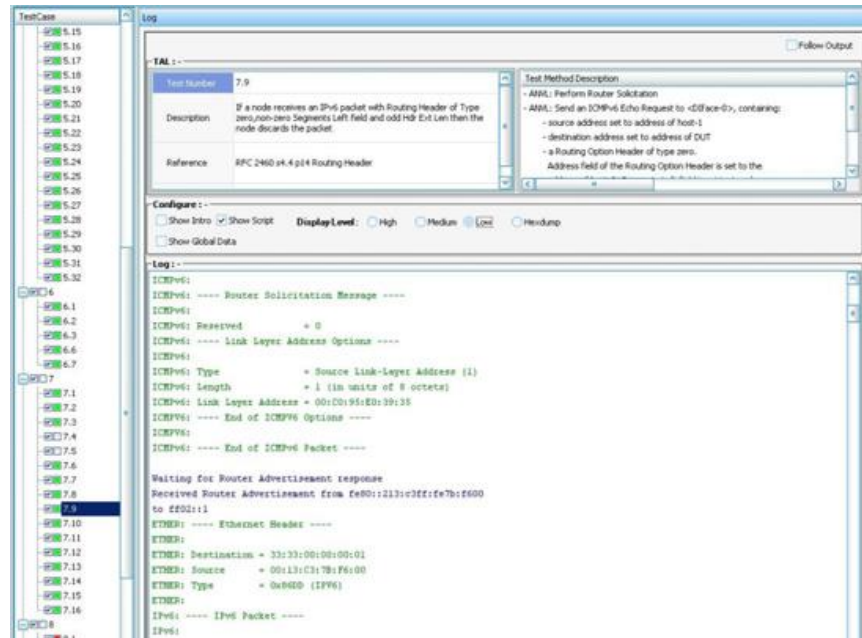
During the test, IxANVL logs the progress in real-time. After completion, IxANVL indicates whether the test passed or failed. IxANVL then repeats the process with the next test until all selected tests have been run.

Test Results

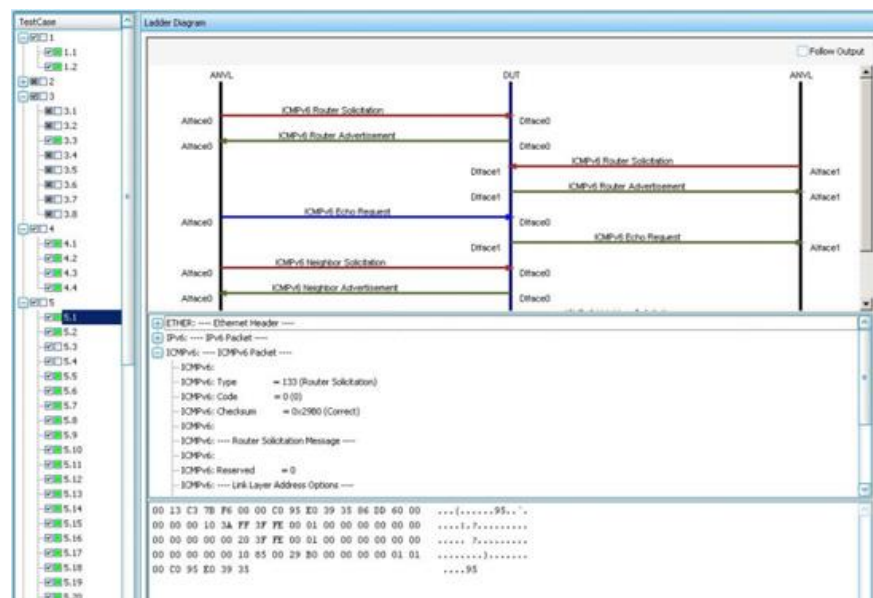
Users can specify four levels of test outputs:

- High level—basic pass/fail
- Medium level—pass/fail and test event status
- Low level—comprehensive report with packet decode
- Hexdump— detail report with hexdump of every packet exchanged between tester and DUT

IxANVL results include detailed trace outputs with a description of the test methodology for side by side reference.



In addition to log outputs, IxANVL provides a timing diagram that represents the relationship of the test packets exchanged between IxANVL and DUT.



IxANVL provides comprehensive packet-by-packet analysis for every test case.

The screenshot shows the IxANVL Packet Decoder interface. On the left, a tree view shows the packet list with filters for Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol. The main window displays a table of packets with columns: Frame, Time Stamp, Interface, Source, Destination, and Type. Below the table, the details of the selected packet (Frame 2) are shown, including Ethernet II header, IPv4 header, and ICMPv4 packet details.

Frame	Time Stamp	Interface	Source	Destination	Type
1	0.00000	0	fffe:1:10	fffe:1:10	ICMPv4 Router Solicitation...
2	0.00000	0	fffe:1:10	fffe:1:10	ICMPv4 Router Solicitation...
3	2.94000	1	fffe:2:10	fffe:2:10	ICMPv4 Router Solicitation...
4	2.94000	1	fffe:2:10	fffe:2:10	ICMPv4 Router Solicitation...
5	4.00000	0	fffe:1:10	fffe:2:10	ICMPv4 Echo Request
6	4.00000	1	fffe:1:10	fffe:2:10	ICMPv4 Echo Request
7	9.00000	0	fffe:1:10	fffe:1:10	ICMPv4 Neighbor Solicitation...
8	10.0	0	fffe:1:10	fffe:1:10	ICMPv4 Neighbor Solicitation...
9	10.0	1	fffe:2:10	fffe:1:10	ICMPv4 Neighbor Solicitation...
10	10.00000	1	fffe:2:10	fffe:2:10	ICMPv4 Neighbor Solicitation...

Packet Details (Frame 2):

- Ethernet II: Ethernet Header
 - IPv4: IPv4 Packet
 - IPv4: Version = 4
 - IPv4: Traffic Class = 0
 - IPv4: Flow Label = 0
 - IPv4: Payload Length = 16 (bytes)
 - IPv4: Next Header = ICMPv4 (58)
 - IPv4: Hop Limit = 255
 - IPv4: Source Address = fffe:1:10
 - IPv4: Destination Address = fffe:1:10
 - ICMPv4: ICMPv4 Packet
 - ICMPv4: Type = 133 (Router Solicitation)
 - ICMPv4: Code = 0 (0)

All IxANVL tests are logged for post analysis.

Platform

- An IxANVL workstation supports the following configuration: CentOS 5.3 (kernel 2.6.18-128.el5xen); CentOS 5.9 (32 bit) with kernel 2.6.18-348.el5; RHEL5.9 (32 bit) with kernel 2.6.18-348.el5
- Microsoft Windows 2003 Server (32 bit), Windows XP Professional (32 bit), Windows 2008 Server Enterprise SP2 (32-bit), Windows 7 Enterprise (32 bit), Windows 7 Professional (32 bit), Windows 7 Professional (64 bit), Windows 2008 Server R2 standard (64 bit) (US English versions)
- 2 GHz Pentium CPU or faster 2 GB RAM
- 512 MB Free Disk Space

Supported Interfaces

IxANVL supports a wide range of network interface cards that directly attach to a Linux or Windows PC:

- Ethernet 10/100
- Gigabit Ethernet
- Async serial
- Sync serial
- T1/E1
- PPPoE

IxANVL also supports Ixia's virtual network interface card (VNIC, an interface driver that resides on a Linux workstation and Ixia chassis, and allows the IxANVL test suites to access Ixia's load modules.

Ixia VNIC supports the following types of Ixia load modules (per-port CPU required):

- 10 Gigabit Ethernet including NGY family
- Ethernet family (10/100/1000 Mbps)
- Packet over SONET OC3/12/48/192
- ATM OC-3/12

VNIC requires the following software:

- Client (IxANVL Workstation): Redhat Enterprise 4.0 with kernel 2.6.9-11.EL or 2.6.9-22.EL, Redhat Enterprise 5.0 with kernel 2.6.18-53.el5, Centos 5.3 with kernel 2.6.18-128.el5xen, Windows 2003 Server (32 bit), Windows XP Professional (32 bit), Windows 2008 Server Enterprise SP2 (32-bit), Windows 7 Enterprise (32 bit), Windows 7 Professional (32 bit), Windows 7 Professional (64 bit), Windows 2008 Server R2 standard (64 bit) (US English versions)
- Server (Ixia Chassis): 6.40GA or 6.50 EA (or higher)

Each conformance test suite supports different sets of test interfaces. Please contact Ixia for applicable test interfaces for test suites of interest.

Product Ordering Information

924-00x-10xx

IxANVL Framework license

924-040-91xx

IPv6 Framework Upgrade if IPv6 test is needed

924-030-xxx

Interface Support Software for each individual test interface. This is the custom interface driver needed to run IxANVL test suite

924-xxx-xxx

Individual IxANVL test suite

This material is for informational purposes only and subject to change without notice. It describes Ixia's present plans to develop and make available to its customers certain products, features, and functionality. Ixia is only obligated to provide those deliverables specifically included in a written agreement between Ixia and the customer.