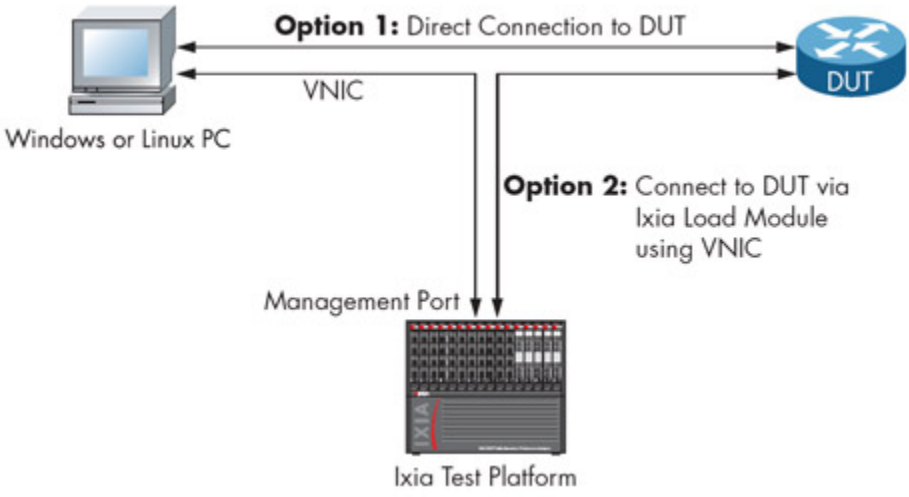
# IxANVL™ - Automated Network Validation Library (ANVL)

Ixia's IxANVL (Automated Network Validation Library or ANVL) 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.

IxANVL is able to run on minimal hardware, a PC with Linux or Windows operating system and an Ethernet card; however, it is particularly well suited to operate 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, IPv6, IP storage, RMON, VPN, MPLS, Voice over IP, Metro Ethernet and Multicast.




26601 W. Agoura Rd. Calabasas, CA 91302 Sales: 1.877.367.4942 / Int'l: 1.818.871.1800

## Test Coverage

| | IxANVL Test Suites | Target Protocols | Reference Specification | Test Case Count | Required Test Interfaces |
|---|---|---|---|---|---|
| **IP Test Suites** | IPv6 Core | IPv6 | RFC 2460, 2464 | 99 | 2 |
| | | IPv6CP | RFC 2472 | 17 | 1 |
| | | ICMPv6 | RFC 4443 | 38 | 2 |
| | IPv6 Advanced | NDP | RFC 2461 | 203 | 2 |
| | | Generic Packet Tunneling | RFC 2473 | 46 | 2 |
| | | AutoConfig | RFC 2462 | 28 | 2 |
| | | V6oV4 | RFC 4213, 2529, 3056, 3068 | 69 | 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, 2874 | 29 | 2 |
| | DHCPv6 | DHCP Client | RFC 3315 | 103 | 1 |
| | | DHCP Server | RFC 3315 | 141 | 2 |
| | IPv4 | IPv4 | RFC 791, parts of 1122, 1812 | 70 | 2 |
| | | ICMP | RFC 792 | 32 | 2 |
| | DHCPv4 | DHCP Client | RFC 2131 | 92 | 2 |
| | | DHCP Server | RFC 2131 | 74 | 2 |
| **Routing** | IP RIP | RIP | RFC 1058, 2453 | 50 | 2 |
| | | IPGW | RFC 1812, 1122 | 18 | 2 |
| | RIPng | RIPng | RFC 2080 | 63 | 2 |
| | OSPF Core | OSPF | RFC 1583, 2328 | 312 | 3 |
| | OSPF Extensions | Opaque LSA, NSSA, DB Overflow | RFC 2370, 3101, 1765 | 53 | 3 |

| | | | | | |
|---|---|---|---|---|---|
| | | OSPF TE | RFC 3630 | 54 | 2 |
| | OSPFv3 | OSPFv3 | RFC 2740 | 325 | 3 |
| | VRRP | VRRP | RFC 3768 | 83 | 2 |
| | BGP4 Core | BGP | RFC 4271 | 187 | 3 |
| | BGP4 Extensions | BGP-OSPF, Communities, Route Flap Damping, Route Reflection, Route Refresh, Confederations | RFC 1403, 1997, 2439, 2796, 2918, 3065 | 118 | 3 |
| | BGP Plus | BGP+ with IPv6 | RFC 4271, draft-ietf-idr-bgp4-26, draft-ietf-idr-rfc2858-bis-06, RFC 2545 | 200 | 3 |
| | ISIS | ISIS | RFC 1195, ISO/IEC 10589: 1992(E) | 213 | 2 |
| | ISISv6 | ISIS-v6 | draft-ietf-isis-ipv6-05 | 205 | 2 |
| **MPLS** | MPLS | Label Encapsulation | RFC 3032 | 59 | 2 |
| | RSVP-TE | RSVP-TE | RFC 3209 | 90 | 3 |
| | LDP | LDP | RFC 3036 | 335 | 3 |
| | L2VPN (PWE3) | PWE3-Control | draft-ietf-pwe3-control-protocol-17 | 67 | 2 |
| | | PWE3-Encapsulation | draft-ietf-pwe3-hdlc-ppp-encap-mpls-05, draft-ietf-pwe3-atm-encap-09, draft-ietf-pwe3-ethernet-encap-10 | 70 | 2 |
| | VPLS | VPLS | draft-ietf-l2vpn-vpls-ldp-08 | 58 | 4 |

| | | | | | |
|---|---|---|---|---|---|
| | L3 VPN | L3 VPN | draft-ietf-l3vpn-rfc2547bis-03 | 101 | 3 |
| **Multicasting Test Suites** | IGMP | IGMPv2 | RFC 2236 | 49 | 2 |
| | | IGMPv3 | RFC 3376 | 155 | 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-09 | 324 | 3 |
| | PIMv6 | Sparse Mode | draft-ietf-pim-sm-v2-new-12 | 267 | 3 |
| | MLD | MLDv1 | RFC 2710 | 98 | 2 |
| | | MLDv2 | RFC 3810 | 203 | 2 |
| **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 | 48 | 1 |
| | TCP High Performance | Ext for High Performance, Selective Ack | RFC 1323, 2018, 793 | 48 | 1 |
| **UDP Test Suite** | UDP | UDP | RFC 728, 1122 | 35 | 1 |
| **Layer 4-7 Test Suite** | HTTP | HTTP Server | RFC 2616 | 345 | 1 |
| | Telnet | Telnet Client/Server | RFC 854 | 45 | 1 |
| | | | | | |
| **IP Storage Suites** | iSCSI | iSCSI Target | RFC 3720 | 211 | 1 |
| | | iSCSI Initiator | RFC 3720 | 205 | 1 |
| **VPN Test Suites** | IPSec AH | MD5, SHA | RFC 4301, 4302 | 61 | 2 |
| | IPSec ESP | MD5, SHA, DES, 3DES, | RFC 4301, 4303, 4304, 4305, | 76 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| | | Blowfish, AES | 4306 | | |
| | IPSec IKE | ISAKMP, IKE | RFC 2407, 2408, 2409 | 385 | 2 |
| | IPSec AH / IPv6 | MD5, SHA, IPSecv6 | RFC 4301, 4302 | 69 | 2 |
| | IPSec ESP / IPv6 | MD5, SHA, DES, 3DES, Blowfish, AES | RFC 4301, 4303, 2404, 2405, 2406 | 78 | 2 |
| | IPSec IKE / IPv6 | ISAKMP, IKE | RFC 2407, 2408, 2409 | 388 | 2 |
| | L2TPSec | L2TP | RFC 2661 | 105 | 2 |
| | L2TP | L2TP | RFC 2661 | 105 | 1 |
| | PPTP | PPTP | draft-ietf-pppext-pptp-02 | 55 | 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 |
| | | Multi-class Extension | RFC 2686 | 9 | 3 |
| **Metro Ethernet Service** | MEF9 | MEF9 | MEF1, MEF9, Iometrix Test Plan version 1.4 | 247 | 6 |
| | EtherCFM | Ethernet CFM | IEEE P802.1ag/D8.1 2007 | 238 | 3 |
| | EtherOAM | Ethernet OAM | IEEE 802.3-ah-2004 | 167 | 3 |
| **Bridging** | STP | 802.1d | IEEE Std. 802.1D-1998 | 53 | 3 |
| | RSTP | 802.1w | IEEE Std. 802.1D- | 127 | 4 |

| | | | | | |
|---|---|---|---|---|---|
| | | | 2004 | | |
| | EAPOL | 802.1x, MD5, TLS, TTLS | IEEE 802.1x-2004 | 74 | 3 |
| | MSTP | 802.1s | IEEE 802.1s, 802.1Q-2003 | 231 | 4 |
| | LLDP | LLDP | IEEE 802.1AB 2005 | 103 | 3 |
| | Mcast Snooping | IGMP/MLD Snooping | RFC 4541 | 41 | 3 |
| | VLAN | 802.1q, GMRP, GVRP | IEEE Std. 802.1Q-2003 | 155 | 4 |
| | LACP | 802.3ad | IEEE Std. 802.3-2002 Clause 43 | 119 | 4 |
| | QinQ | QinQ | IEEE 802.1ad 2005 | 135 | 2 |
| **Voice** | SIP | SIP Server | RFC 3261 | 297 | 2 |
| **RMON Test Suites** | RMON | Ethernet<br>General | RFC 1757<br>RFC 1757 | 116<br>372 | 1<br>1 |
| **Toolkits** | TCP Toolkit | | N/A | 4 | 1 |
| | SNMP Toolkit | Toolkit & sample tests only | N/A | 9 | 1 |

**Note 1**: TCP Test Suites require connection with Device Under Test from below and above the target TCP layer. Connection from below the TCP layer can be achieved by the traditional physical layer interface. Connection from above the TCP layer can only be done with a unique application called TCP Stub developed by Ixia. The TCP Stub is remotely controlled and managed by Ixia TCP Test Suites. The purpose of TCP Stub is to generate the necessary stimulus above the TCP layer required for testing. The TCP Stub is a portable C code bundled with TCP test suites. Customers are required to compile the TCP Stub onto their target systems.

## Benefits

### IxANVL Saves Time & Money

IxANVL allows vendors to verify the design during the entire product life cycle. Problems can be identified earlier in order to prevent costly last-minute reworks. IxANVL can emulate large, multi-

node networks that were previously cost prohibitive. The result is more efficient tests and quicker product release times.

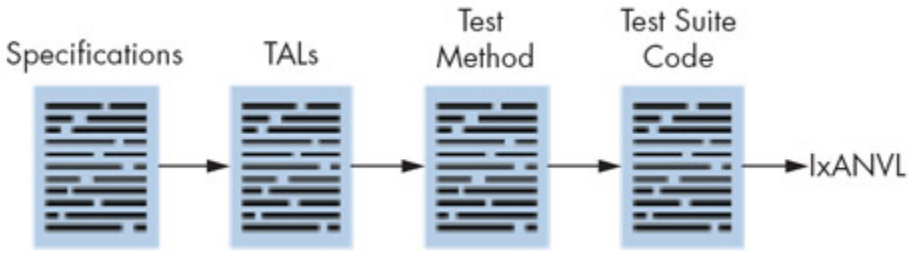
**IxANVL Increases Confidence**
IxANVL increases confidence in product quality by enabling extensive and thorough testing to be performed automatically, without supervision. IxANVL's test results allow users to:

- Determine exactly where a device's protocol software does and does not meet the specification
- Observe how well the device will handle traffic from non-complying network components
- Determine what effects new development has on existing code through regression testing

**IxANVL Expands Easily**
With a source code license, users can easily add new interface types, protocols, and/or tests to their IxANVL system.

**IxANVL 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 process to develop the test suite:

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

During the development process, continual verification with protocol standard authors or implementers is performed.

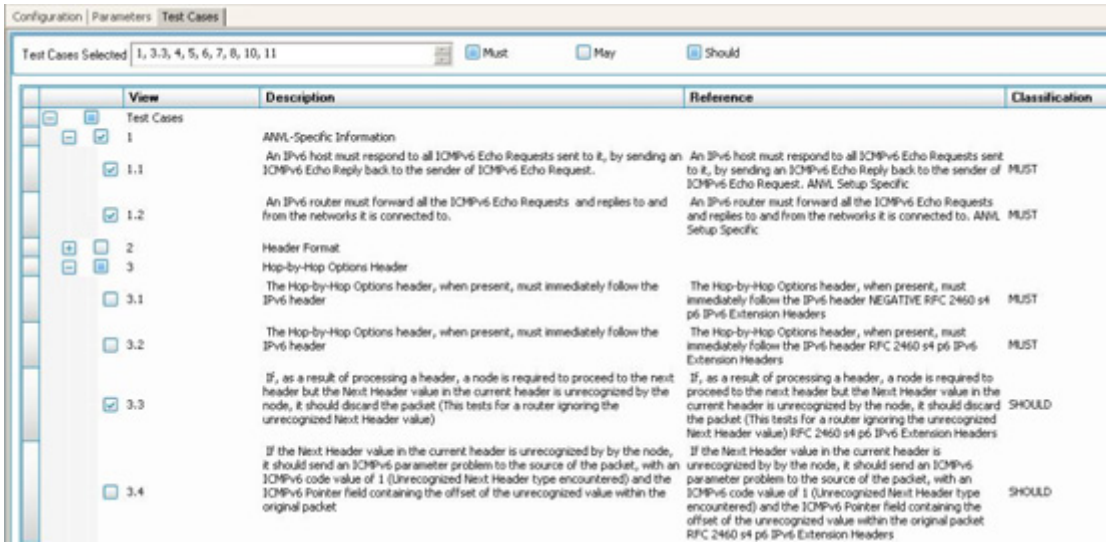
## Test Configuration

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 Device Under Test (DUT) via test interfaces. Up to four interfaces may be utilized, depending on the test configuration. IxANVL has the flexibility to emulate various system topologies to create virtually any test scenarios, for almost any DUT.

IxANVL offers the option of a command-line interface for test automation, and a user-friendly graphical user interface, which allows intuitive test execution management and detail reporting. A batcher runner is also available for scheduling a sequence of test runs for regression purposes.

## Test Execution

IxANVL classifies test cases into three categories: MUST, SHOULD, and MAY.



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 mode, the user types a command with options indicating which tests should be run and the desired level of output.

IxANVL then sends packets to the DUT based on what the test is designed to do, after which, IxANVL receives packets from the DUT, and compares them 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.

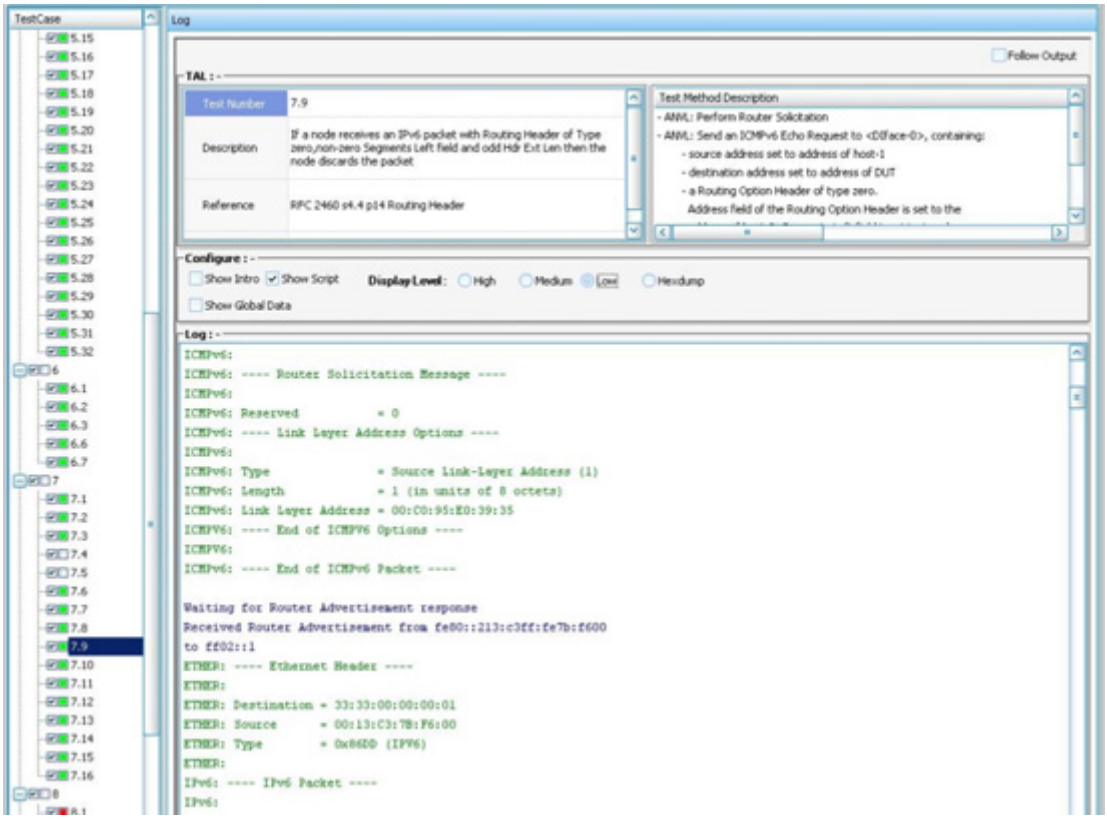
During the test, IxANVL logs the progress in real-time. After completion, IxANVL indicates whether the test passed or failed. IxANVL then repeats the process with the next test until all selected tests have been run.

## Test Results

Users can specify four different levels of test outputs:

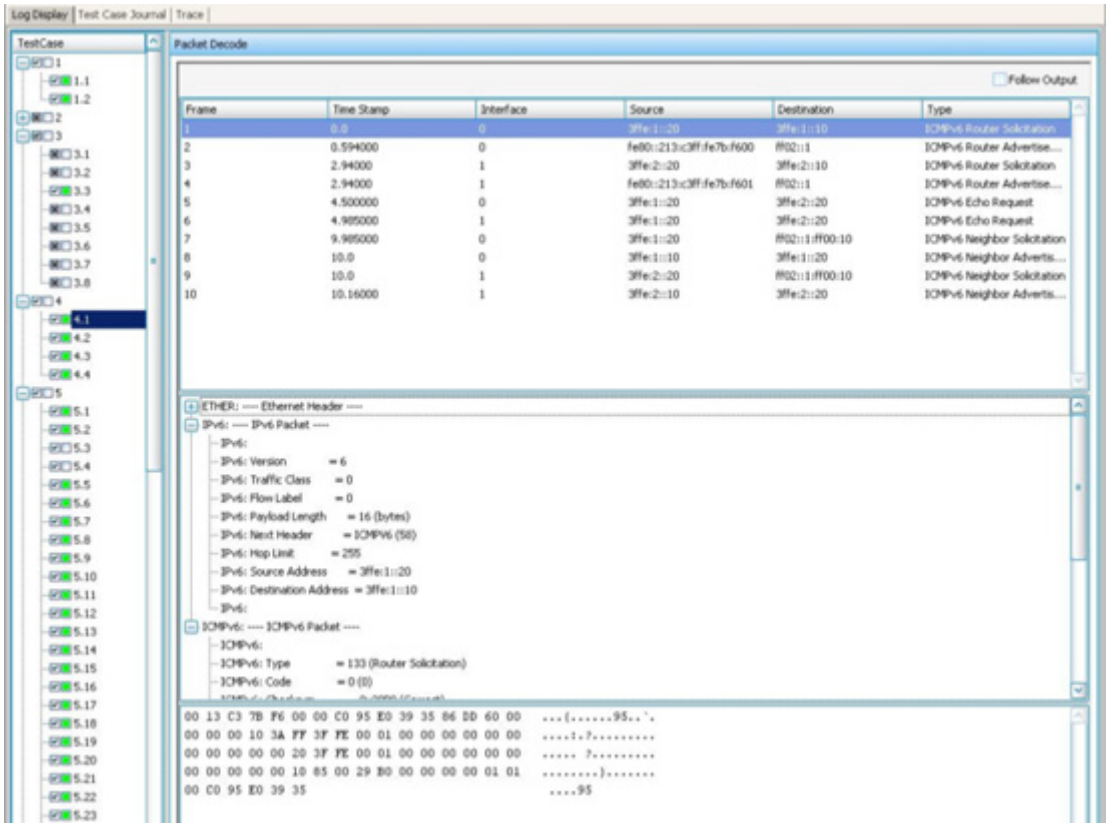
- High level - basic pass/fail
- Medium level - pass/fail and test event status
- Low level - comprehensive report with packet decode
- Hexdump - detail report with hexdump of every packet exchanged between tester and DUT

IxANVL results include detailed trace outputs with description of 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 cases.




All IxANVL tests are logged for post analysis.

## Platform

IxANVL workstation supports the following configuration:

- Redhat Linux 9.0 with kernel 2.4.20-8 or 2.4.20-6, Redhat Enterpise 4.0 with kernel 2.6.9-11 or 2.6.22.0.2.EL.
- Microsoft Windows XP Professional or Windows 2000 Professional (US English versions)
- 1.5G Hz Pentium CPU or faster (32 bit system only)
- 1G MB 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). Ixia VNIC is an interface driver residing on a Linux workstation and Ixia chassis that 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
- Ethernet TXS 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 9.0 with kernel 2.4.20-8 or 2.4.20-6, Redhat Enterprise 4.0 with kernel 2.6.9-11.EL or 2.6.22.0.2.EL
- Server (Ixia Chassis): 5.00 SP3 (or higher), or 5.10 SP1 (or higer)

Each conformance test suite supports different set of test interfaces. Please contact Ixia for applicable test interfaces for the test suite of your interest.

## Product Ordering Information

The following items are required to complete an IxANVL test system:

**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