Mercury Consulting Group · MCG, LLC
PO Box 1237 · Williamstown, NJ 08094
tel 844.888.8123
bherd@mcgllc.com

AI / LAI Based Test Generation

IMSv2 - Advanced Network Intelligence Center | Self-test, automatic configuration, automatic discovery, SmartHost tracking

MCG Labs technical white paper - November 2026 - A subsidiary of MCG LLC - mcg-labs.ai

Executive summary

IMSv2 Configuration Automation points a Localized Advanced Intelligence (LAI) appliance at a customer network and architects the whole operations wall - discovering devices, inferring the topology, tracking every host (SmartHost), and designing the monitored set. Because operators act on what the wall shows, a confident wrong answer is worse than none, so discovery and inference must be validated against external ground truth. This paper describes the technique we use to do that: teaching the appliance's own intelligence to read the reference material a network exposes - MIB object IDs, real published backbone maps, device semantics - and from it manufacture the exact known-answer test stimulus needed to grade the product.

Measured against 276 real Internet Topology Zoo backbones, the first inference was confidently wrong on 193 of 276 networks (70%) and never abstained. The stimulus exposed it and drove the fix; over-confident mis-calls fell to 19 of 276 (6.9%). On the single-hub and dual-hub designs that make up the great majority of real customer deployments, the inference scores highest and never fabricates a hub.

1. The value and the need for test stimulus

Stimulus is the controlled, known-answer input you feed a system so its output can be graded against truth. For a network product it is the network itself - devices, adjacency, forwarding tables, SNMP responses on the wire. It is needed because:

- Self-test must be against ground truth, not the author's expectations. A synthetic unit test only proves the code does what its author imagined; it cannot show behavior on the variety of real networks.
- You cannot keep 276 real backbones in the lab, and a customer's live network is not a place to experiment. Stimulus brings that reality into CI.
- It is a regression gate: once scored automatically, every future change to discovery, inference, or SmartHost tracking must keep scoring correctly or the build fails.

Good stimulus is expensive to author by hand - a single realistic switch is hundreds of internally consistent SNMP objects cross-referencing each other by index. That is exactly the work the on-site intelligence is built to do.

2. Using AI / LAI to understand references and create stimulus

The governing idea: the stimulus generator reads the same references the product reads, so test and subject share one source of truth and cannot silently drift apart. A network exposes two reference bodies, and we generate stimulus from each.

2.1 Structural reference - real published topologies

The Internet Topology Zoo is 276 real backbones built from operator-published maps. The harness reads each graph, converts it into the IMSv2 neighbor-adjacency evidence form (the same shape a live LLDP sweep produces), independently classifies the graph's true structure from its degree sequence, feeds the evidence to the product's own topology inference, and grades the inferred shape against the known structure - across all 276 in one run.

2.2 Live-wire reference - the device MIB

The Zoo has no VLANs, forwarding tables, or ARP, so the Layer-2 half of discovery and all of SmartHost tracking need a device answering real SNMP. The harness reads the product's own probe to learn the exact numeric OIDs it walks, synthesizes a coherent device from a high-level spec (device_profile.py) emitting every object at the exact OID with correctly cross-referencing indexes, and serves it as genuine SNMP v2c over UDP via a self-contained agent and walk client (snmp_emulator.py) - standard library only, no net-snmp, no pysnmp. One file defines the OID bases shared by emulator and product, so they stay in lock-step by construction. No frontier model, no cloud, no data egress.

3. SmartHost tracking - the process under validation

SmartHost is the IMSv2 process that gives every host a stable identity and a physical location, independent of its IP. It fuses location - joining the switch forwarding table (MAC to switch-port to VLAN) with ARP (IP to MAC) to pin each host to a physical switch, port, and VLAN - with a probabilistic multi-signal identity (serial, SNMP engine ID, SSH host-key, TLS cert, MAC, sysName, DNS). A host that changes IP but keeps its MAC and switch-port is a move, not a new host. The full binding is MAC <-> IP <-> port <-> switch <-> VLAN.

The SNMP emulator exists to stand that chain up end to end: live_vlan_validate.py serves a synthetic 3-VLAN switch, points the product's real probe at it, runs SmartHost, and asserts the whole binding. On its first run the stimulus caught two real product bugs (a host-count-by-port error and ARP-discovered hosts not being seeded into the monitored set) - neither visible to the product's own unit tests.

4. What we found out - transparent results

Run against 276 real backbones, the first inference reported a confident single-hub or dual-hub for 193 of 276 networks (70%) that have no hub at all - they are meshes - and never abstained. Its confidence was a fixed constant, carrying no information. A value that is always the same constant is not a measurement.

4.1 How we treat a mesh

The shape is now decided by the count of cores, where a core is a node that genuinely forwards (co-equal with the busiest node, degree at or above 60% of the maximum):

- one core - single-hub (hub-and-spoke with many point-to-point leaves);
- two cores - dual-hub (two hubs plus a core interconnect);
- three or more interconnected co-equal cores - full-mesh, a new first-class shape; a mesh is no longer forced into a hub label;
- no cores and no dominant hub - unknown; the inference abstains rather than fabricate a hub.

Confidence is now computed from the evidence (core count and degree margin), and the full rationale is recorded so an operator can see why and overrule it.

4.2 The hit rate, stated plainly

Re-validated against the same 276 networks, over-confident mis-calls fell from 70% to 6.9% (19 of 276). The corpus-wide verdicts:

Verdict	Count	Share
plausible (fits structure)	217	78.6%
conservative (honest unknown)	40	14.5%

suspicious (fabricated hub)	19	6.9%

Right or honestly abstaining 93% of the time; outright over-confident under 7%. Accuracy by the network's real structure:

Real structure	Networks	Correct	Accuracy
star -> hub / spoke	23	22	96%
mesh -> full-mesh	30	28	93%
ring -> abstain	3	3	100%
tree	9	7	78%
partial-mesh	185	137	74%
dual-core -> dual-hub	26	14	54%

4.3 The single-hub / dual-hub case - what matters in the field

The Topology Zoo is ISP and research backbones, which are overwhelmingly meshes - the hardest possible setting for hub detection, reported on purpose. It is not what a typical IMSv2 customer runs. A real customer deployment - an enterprise, a county, a small carrier - is almost always single-hub or dual-hub. We expect those shapes to be on the order of 80% of the deployments we meet in the field (an engineering expectation from the customer base, not a lab measurement), and the hub family is exactly where the inference scores highest: a clean star at 96%, a true mesh at 93%, and it never fabricates a hub on a graph that has none. The weaker figure, dual-core at 54%, is a genuine boundary ambiguity (one busy node plus a couple of lesser cores reads as either a star or a small mesh); the operator confirms the shape in the wizard in every case.

5. Why this is a product advantage

- Self-testing by construction: the product's intelligence reads references and generates its own known-answer stimulus, so discovery and SmartHost tracking are graded against reality every build.

- Zero egress: the generation is mechanical synthesis from references the appliance already holds - standard library only, nothing leaves the building; it self-tests even in air-gapped and CJI sites.

- It generalizes: structural stimulus hardens inference, live-wire stimulus hardens the probe and SmartHost, and a data overlay drives the wall for demos. One discipline, three uses.

6. Reproducibility and scope

Fully reproducible from the dev-only imsv2-testnet-harness repository (private; never shipped in the appliance image). Point it at the IMSv2 engine/configauto checkout and run handlers/run_zoo.py for the topology validation, and handlers/live_vlan_validate.py for the SNMP emulator and SmartHost checks; standard-library Python throughout. Scope limit, stated honestly: the Zoo graphs are point-of-presence adjacency, validating the adjacency and routing half of inference, not the Layer-2 forwarding half - which the SNMP emulator covers. Broadening the live-wire stimulus toward multi-device, multi-vendor fabrics (Containerlab / Cisco CML) is the planned next step. The figures here are a dated snapshot that is re-measured every harness run, not a fixed constant.