International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Closing the Intent Gap: Policy-Driven Validation in Enterprise EVPN-VXLAN Data Center Architectures

Miteshkumar Jagdishbhai Patel

Abstract

Intent-Based Networking abstracts enterprise data center management into policy definitions that automation systems translate into fabric configurations, a compelling operational model that carries a specific and underexamined risk when deployed over EVPN-VXLAN architectures. The translation chain connecting expressed intent to protocol-layer enforcement spans multiple abstraction layers, each capable of introducing divergence between what operators defined and what the fabric actually does. This article terms this divergence the "intent gap" and argues that its existence in unvalidated IBN deployments constitutes a false confidence problem with direct operational and security consequences. A taxonomy of intent gap types specific to EVPN-VXLAN environments is developed, covering BGP EVPN control-plane, VXLAN data-plane, policy enforcement, and temporal gap categories, and a protocol-layer validation architecture is proposed to address each category. The article further presents a closed-loop enforcement framework that converts validated gap detection into autonomous remediation, reducing correction latency from human response time to automation execution time. The framework is illustrated throughout using Cisco ACI/APIC constructs as a concrete reference implementation, though the underlying validation architecture is intended to generalize to other EVPN-VXLAN controller platforms that expose equivalent control-plane and data-plane primitives. Deploying this validation architecture alongside IBN provisioning systems is argued to be a necessary condition for IBN to deliver the operational consistency its adoption is intended to provide.

Keywords: Intent-Based Networking, EVPN-VXLAN, Data Center Automation, Policy Validation, Closed-Loop Automation, Multi-Tenant Networking, Network-as-Code, SDN Controllers, Observability, Intent Gap

I. Introduction

A provisioning workflow that completes with all success indicators confirmed, controller status green, no error logs, deployment confirmed, while the network simultaneously violates the policy the workflow was deployed to enforce represents a specific and consequential failure mode that IBN literature has underexamined. In an EVPN-VXLAN fabric managed through a policy-based controller such as Cisco APIC, the translation chain between an operator's expressed intent, tenant A cannot communicate with tenant B, and its actual enforcement in fabric hardware spans multiple abstraction layers, each of which can introduce errors that the controller's status reporting cannot detect [1]. The intent gap that results is not visible in the management plane; it is visible only in the data plane, where traffic that should be blocked traverses an unintended path because a VXLAN Tunnel Endpoint (VTEP) misconfiguration was not caught at the translation layer.

EVPN-VXLAN architectures have become a widely adopted approach for enterprise data center fabrics because of their capacity to support multi-tenant segmentation, workload mobility, and scalable overlay networking within a unified control plane [7]. The same architectural properties that make EVPN-VXLAN operationally capable, the separation of control plane from data plane, the VRF-based tenant isolation model, and the VTEP-based endpoint reachability framework, create multiple independent layers at which intent expression and intent realisation can diverge. Deploying IBN over this architecture without continuous protocol-layer validation is, in effect, accepting that the automation layer will confirm

Independent Researcher, USA

success for operations that may have partially failed at layers it does not observe [5].

The operational risk this produces is specifically a false confidence problem. IBN deployments without validation create the operational appearance of automation-governed consistency while the underlying fabric operates in states that the automation never verified. An organisation that has deployed IBN for tenant provisioning and trusts its controller status reporting as evidence of policy compliance is in a worse operational position than an organisation that manages its fabric manually and knows that manual verification is required, because the former has systematically eliminated the manual verification steps that would have caught the gaps the automation missed [3], [4].

This article develops a validation-centric IBN framework specific to EVPN-VXLAN multi-tenant architectures. Cisco ACI/APIC terminology is used throughout as a concrete illustrative reference implementation because it provides the most extensively documented commercial EVPN-VXLAN controller model; the underlying gap categories and validation mechanisms are architectural rather than vendor-specific, and are intended to generalize to other controller platforms, Arista CloudVision, Juniper Apstra, and open Network-as-Code toolchains among them, that expose comparable BGP EVPN and VXLAN control primitives. The discussion proceeds through an analysis of the intent expression and translation stack, a taxonomy of intent gap types, a validation architecture that addresses each gap category, a closed-loop enforcement framework that converts detected gaps into automated remediation, and a discussion of the framework's own operational limitations. The contribution is not a new IBN paradigm but a specific architectural component, protocol-layer intent validation, that most IBN deployments currently lack and whose absence accounts for a significant fraction of the operational failures attributed to IBN immaturity [4].

II. Intent Expression and the EVPN-VXLAN Translation Stack

The distance between what an operator expresses as intent and what a network device ultimately executes is measured in translation steps, and each step introduces error probability. In a Cisco ACI deployment, an operator expressing the intent that a specific application endpoint group should have regulated access to a database Endpoint Group (EPG) triggers a translation sequence: the APIC controller interprets the policy model, assigns VXLAN Network Identifier (VNI) values, generates contract rules, determines which leaf switches are affected, calculates the VXLAN encapsulation requirements, and pushes device-specific NETCONF/YANG configurations to each relevant leaf [1]. At any point in this sequence, a translation error can produce a fabric state that differs from the expressed intent while the controller reports the operation as complete.

Network-as-Code (NaC) approaches to intent expression introduce a different but related translation challenge. When operators define EVPN-VXLAN tenant configurations in structured templates, Ansible playbooks, Cisco NSO service models, or Terraform configurations, the template parameters must map correctly to the specific EVPN control plane model: route distinguisher and route target assignments that correctly implement the VRF separation design, VNI assignments that do not conflict with existing allocations, and VTEP IP address configurations that are reachable from all relevant fabric peers [2]. Route-target misconfiguration is a particularly consequential translation error class: an incorrect RT import policy on a VRF causes that VRF to receive routes from tenants whose traffic it should not see, directly violating the isolation intent the provisioning workflow was designed to enforce [5].

SDN controller health and synchronisation state introduce a third translation failure category distinct from configuration error. A Cisco APIC controller cluster operating in a degraded state, with one node down, synchronisation lag between cluster members, or southbound connectivity issues with a subset of fabric leaves, may successfully translate policy intent into configuration objects while failing to push those configuration objects to all affected devices [6]. Monitoring controller operational health as an independent input to the intent validation system is a design requirement that the false confidence analysis makes clear.

The translation stack analysis establishes the architectural principle that governs the validation framework: intent verification must be performed at the layer where intent realisation can fail, not at the layer where intent is expressed. Controller status reporting operates at the intent expression and translation layers; protocol-layer validation operates at the intent realisation layer. Closing the intent gap requires validation coverage at both layers, with

particular emphasis on the EVPN control plane and VXLAN data plane where translation failures are most consequential and least visible [4].

III. Intent Gap Taxonomy in EVPN-VXLAN Multi-Tenant Fabrics

Control-plane intent gaps arise from discrepancies between the BGP EVPN state that the intent model expects and the BGP EVPN state that fabric devices actually maintain. A VRF provisioned to import routes from a specific route target should receive reachability information for all endpoints in the corresponding tenant; if the import policy is misconfigured on one or more leaves, those leaves will not install the tenant routes, producing an asymmetric reachability condition where traffic flows correctly from some fabric nodes and fails from others [1]. This asymmetry is particularly difficult to diagnose without protocol-layer validation because user-visible application failures are intermittent and topology-dependent rather than consistent.

Data-plane intent gaps involve discrepancies between the VXLAN encapsulation behavior that the intent model specifies and the encapsulation behavior that fabric hardware executes. VTEP IP address misconfigurations, VNI-to-VLAN mapping inconsistencies between fabric nodes, and MTU mismatches that cause VXLAN-encapsulated frames to be silently dropped represent the primary failure modes in this category [7]. Data-plane gaps are operationally distinctive because they affect traffic directly rather than routing state: a VNI misconfiguration does not prevent routes from being installed in the correct VRFs but causes traffic following those routes to be dropped or misforwarded at the encapsulation point.

Policy enforcement gaps occur when security group assignments, contract rules, or microsegmentation policies are correctly defined in the controller policy model but incompletely instantiated in fabric hardware. In Cisco ACI environments, EPG-to-EPG contracts are enforced in the hardware policy enforcement points of each leaf that has member endpoints from both EPGs; if the contract is not instantiated on all relevant leaves simultaneously, traffic between EPG members co-located on non-compliant leaves may bypass policy enforcement [5]. This partial enforcement condition produces the most serious security consequences of any intent gap category.

Temporal intent gaps arise not from provisioning errors but from state divergence occurring after correct provisioning. Fabric software upgrades that reset VTEP state, link failures that trigger Equal-Cost Multi-Path (ECMP) reconvergence to paths not anticipated by the intent model, and workload migrations that move endpoints to leaves where the destination VNI was not pre-provisioned all produce post-provisioning divergence between intent and fabric state [6]. Temporal gaps are the most challenging category for validation systems because they require the system to maintain an accurate model of intent at all times and compare it against current fabric state continuously.

The taxonomy's operational value is that it maps validation requirements to gap categories precisely. Control-plane gaps require BGP EVPN state verification; data-plane gaps require active VTEP-pair probing; policy enforcement gaps require distributed contract instantiation verification; temporal gaps require continuous monitoring rather than periodic scanning [4]. A validation framework designed around this taxonomy can target its validation mechanisms at the specific failure modes most likely to occur in a given operational context.

TABLE I. Intent Gap Taxonomy: EVPN-VXLAN Failure Categories and Validation Requirements

Gap Category	Formation Mechanism	Fabric Manifestation	Detection Method Required
Control-plane	RT import/export misconfiguration, BGP session state errors	Asymmetric VRF route population	BGP EVPN RIB state comparison via telemetry
Data-plane	VTEP IP misconfiguration, VNI-VLAN mismatch, MTU errors	Silent traffic drop or misforwarding	Active VTEP-pair probing
Policy enforcement	Incomplete contract instantiation on leaf ASICs	Topology-dependent bypass of microsegmentation	Distributed contract instantiation audit

Temporal	Software upgrade, link failure, workload migration	Post-provisioning state divergence	Continuous telemetry-driven state comparison

IV. Validation Architecture: Closing the Intent Gap Through Protocol-Layer Verification

Protocol-layer intent validation's defining architectural characteristic is its independence from the controller reporting path. Where controller status confirms that a configuration push completed successfully from the management plane's perspective, protocol-layer validation confirms that the resulting operational state is correct from the data plane's perspective, that routes are present, that encapsulation is functioning, and that policies are enforced. These two confirmations address different questions, and only the second answers whether the fabric is doing what the intent model specifies [3].

BGP EVPN control-plane validation requires comparing the route tables and BGP session state of fabric devices against a reference intent model that specifies expected BGP peer relationships, route-target import/export policies, and the resulting VRF route populations for each tenant [1]. This reference model must be derived from the intent definition system, the NaC repository or IBN policy engine, rather than from the controller's internal state, because deriving it from the controller would circular-reference the same translation layer whose correctness the validation is designed to assess. Streaming telemetry exports of BGP EVPN Routing Information Base (RIB) state, for example via gNMI-based pipelines of the kind used in adjacent in-band telemetry work, can plausibly provide the device-side data for this comparison at sub-second granularity, though the specific gNMI/BGP EVPN RIB integration should be validated against vendor telemetry documentation before implementation [8].

Active VTEP-pair probing addresses data-plane validation requirements that passive telemetry monitoring cannot satisfy independently. Synthetic traffic injected between VTEP pairs with VXLAN encapsulation matching the specified VNI assignments tests the actual forwarding path that tenant traffic will follow, confirming that encapsulation and decapsulation are functioning correctly [10]. Drop counter monitoring on VTEP interfaces provides passive data-plane visibility into encapsulation failures, MTU violations, and forwarding anomalies that complement active probing results [9].

Building a multi-layer intent model that spans control-plane, data-plane, and policy enforcement layers simultaneously is the primary engineering investment that enables precise validation across all three layers [6]. The intent model must express not only high-level policies but the specific EVPN/VXLAN constructs that realise them, specific RT values, VNI assignments, VTEP IP addresses, and contract rules, so that each validation layer can compare fabric state against the specific expected values at its layer. This translation from high-level intent to validation-layer assertions is an engineering challenge that currently receives little attention in IBN system design, where most investment goes into the northbound intent expression layer rather than the southbound validation layer [4].

TABLE II. Protocol-Layer Validation Architecture: Coverage by Gap Category

Gap Category	Validation Mechanism	Data Source	Validation Frequency
Control-plane	BGP EVPN RIB state comparison	gNMI streaming telemetry	Continuous (sub-second)
Data-plane	Active VTEP-pair probing + drop counter monitoring	Synthetic traffic + telemetry	Per-tenant, interval-based
Policy enforcement	Distributed contract instantiation audit	Device management plane query	Event-triggered (post-provisioning)
Temporal	Continuous intent model vs fabric state delta comparison	Streaming telemetry + intent model	Continuous

V. Closed-Loop Intent Enforcement and Fabric Self-Correction

Detection latency between intent gap formation and operator awareness is the operational metric that determines whether validation-based IBN provides meaningful protection against fabric state violations [3]. Consider an illustrative, hypothetical scenario rather than a measured result: a validation system that detects an intent gap 30 seconds after formation but requires 20 minutes of human triage before any remediation begins would provide substantially less protection than its detection capability suggests in environments where application SLA violations can occur within seconds of fabric state divergence. The specific figures are used here only to make the detection-latency-versus-triage-latency distinction concrete; they are not derived from a measured deployment and should be validated against pilot or testbed data before being cited as an expected outcome.

The remediation action library for EVPN-VXLAN gap types maps directly to the gap taxonomy: controller re-push addresses configuration translation failures where the intended configuration was not correctly delivered to the device; BGP session reset procedures address control-plane state corruption where the session is maintaining incorrect state; VNI re-provisioning addresses data-plane gaps where VTEP configuration is inconsistent with the intent model [4]. Action selection logic must be precise: applying a BGP session reset to a device experiencing a data-plane gap will disrupt traffic unnecessarily without addressing the actual failure.

SDN controller health monitoring must be integrated into the closed-loop enforcement framework as a distinct validation input. Controller operational state, cluster consensus health, southbound device connectivity, policy model synchronisation lag, affects the reliability of controller-driven remediation actions [6]. A controller experiencing synchronisation issues may acknowledge a re-push request while failing to propagate the configuration to all affected fabric nodes, producing a remediation action that monitoring records as successful while the intent gap persists.

The closed-loop enforcement framework's operational contribution is converting IBN from a provisioning tool, where intent is expressed and manually verified occasionally, into an operational framework where intent is expressed and continuously enforced. Continuous enforcement means that intent gaps do not accumulate between validation cycles; they are detected and remediated within the correction latency of the closed-loop system, which is bounded by automation execution time rather than human response time [11].

TABLE III. Closed-Loop Remediation Action Library: EVPN-VXLAN Gap-to-Action Mapping

Gap Category	Remediation Action	Execution Pre-Condition	Post-Action Validation
Control-plane (RT misconfiguration)	Controller policy re-push for affected VRF	Controller health confirmed healthy	BGP RIB state re-verification
Control-plane (session corruption)	BGP session targeted reset	Identify specific affected session	Peer adjacency and route population check
Data-plane (VTEP misconfiguration)	VTEP configuration re-application	Controller southbound connectivity confirmed	Active VTEP-pair probing post-apply
Policy enforcement (partial instantiation)	Contract targeted re-instantiation on affected leaves	Identify specific non-compliant leaves	Distributed contract instantiation audit
Temporal (post-provisioning drift)	Automated re-sync of VNI/VTEP/contract state for affected endpoints (e.g., post-migration, post-upgrade)	Delta confirmed against continuous telemetry model	Continuous intent model vs. fabric-state delta re-check

VI. Limitations and Open Challenges

The validation architecture proposed here carries operational costs and unresolved risks that a deployment decision should weigh explicitly.

Continuous sub-second comparison of BGP EVPN RIB state against a reference intent model, as described for control-plane and temporal gap categories, imposes a recurring computational and telemetry-bandwidth load that scales with fabric size and tenant count; at large multi-tenant scale this overhead has not been characterised in this article and would need to be measured against the specific streaming telemetry infrastructure in use before continuous validation is deployed fabric-wide.

Active VTEP-pair probing introduces a second-order risk that the framework does not fully resolve: synthetic traffic injected to test the data plane consumes fabric resources and, if probing volume or frequency is not carefully bounded, can itself contend with production traffic on the paths it is testing. A production deployment would need rate limiting, scheduling that avoids peak utilisation windows, and probe traffic that is clearly distinguishable from tenant traffic for billing and troubleshooting purposes.

A more fundamental limitation concerns the reference intent model itself. The validation architecture in Section IV treats the Network-as-Code repository or IBN policy engine as ground truth and validates fabric state against it; the framework does not address what happens when the intent model is itself incorrect, for example when a route-target assignment was wrong at the point of authoring, or when the intent model has drifted from an approved change without detection. Validating fabric state against an incorrect intent model would confirm compliance with the wrong policy rather than closing a genuine intent gap. Establishing intent-model correctness, through change-approval workflows, version control, and independent policy review, is a prerequisite that this article assumes rather than solves.

Closed-loop automated remediation also carries a false-positive risk that the article's own caution about action-selection precision does not fully address. A validation system that misclassifies a transient condition, for instance a brief BGP convergence event following a legitimate topology change, as a control-plane intent gap could trigger an automated BGP session reset that disrupts traffic unnecessarily. Guarding against this class of error would require the remediation layer to incorporate confirmation thresholds, such as requiring a gap to persist across multiple consecutive validation cycles before triggering action, together with staged rollout of automated remediation actions, starting in an alert-only mode before automated execution is enabled, and rollback procedures that can reverse a remediation action if post-action validation shows the action did not resolve the underlying condition. None of these safeguards is elaborated in the closed-loop design presented in Section V, and each represents a design decision that would need to be resolved before autonomous remediation is enabled in a production fabric.

Conclusion

The intent gap problem in IBN-managed EVPN-VXLAN fabrics arises directly from the translation stack depth that separates high-level policy expression from protocol-layer enforcement. Each translation step introduces the possibility of divergence between expressed intent and operational fabric state. Current IBN deployments that rely on controller status reporting as validation evidence are operating at the top of this stack, blind to the failure modes that occur at its lower layers.

The intent gap taxonomy developed in this article makes gap formation predictable: control-plane gaps concentrate at BGP EVPN route-target policies and peer session state; data-plane gaps concentrate at VTEP configuration consistency and VNI mapping; policy enforcement gaps concentrate at distributed contract instantiation across leaf hardware; temporal gaps arise from post-provisioning state divergence driven by operational events. Predictable gap formation enables targeted validation, verification mechanisms designed specifically for the failure modes most likely to occur.

Protocol-layer validation operating below the controller abstraction layer is the architectural component that current IBN deployments systematically lack. Deploying control-plane BGP EVPN state verification, active VTEP-pair data-plane probing, and distributed policy enforcement consistency checking alongside the IBN provisioning system closes the visibility gap between expressed intent and operational fabric state, eliminating the false confidence that characterises unvalidated IBN deployments.

Closed-loop enforcement, built on this validation foundation, converts intent validation from a monitoring function into an operational control system, one that detects gaps and closes them autonomously within bounded correction latency, maintaining continuous alignment between the intent model and the fabric it governs. Enterprise data centers that deploy IBN with this validation

architecture achieve the operational promise that IBN's proponents articulate; those that deploy IBN without it achieve an operationally credible appearance of automation without its substance.

The limitations discussed in Section VI point to a concrete research agenda rather than closing the topic: standardising validation APIs across controller vendors so that the architecture proposed here is not tied to a single platform, developing intent-model correctness and drift-detection mechanisms that verify the reference model rather than only the fabric state against it, and characterising the scaling behaviour of continuous sub-second telemetry comparison in large multi-tenant fabrics with quantitative testbed data. Each of these questions must be resolved before protocol-layer validation moves from an architectural proposal to a broadly deployable operational practice.

Conflict of Interest Statement

The author declares no conflict of interest. No funding was received for this work.

References

[1] A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, 2023. Available: https://ieeexplore.ieee.org/document/9925251

[2] E. Coronado et al., "Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks," IEEE Communications Surveys & Tutorials, vol. 24, no. 4, pp. 2535–2578, 2022. Available: https://ieeexplore.ieee.org/document/9913206

[3] P. Almasan et al., "Network Digital Twin: Context, Enabling Technologies, and Opportunities," IEEE Communications Magazine, vol. 60, no. 11, pp. 22–27, 2022. Available: https://ieeexplore.ieee.org/document/9795043

[4] M. Gharbaoui et al., "Assurance and Conflict Detection in Intent-Based Networking: A Comprehensive Survey and Insights on Standards and Open-Source Tools," IEEE Transactions on Network and Service Management, 2026. Available: https://ieeexplore.ieee.org/document/11334180

[5] Yu-Cheng Yang et al., "SDN-Enabled EVPN-VXLAN With P4 Accelerated User Plane," 25th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2025. Available: https://ieeexplore.ieee.org/document/11181298

[6] Xiaoang Zheng and Aris Leivadeas, "Network Assurance in Intent-Based Networking Data Centers with Machine Learning Techniques," 17th International Conference on Network and Service Management (CNSM), 2021. Available: https://ieeexplore.ieee.org/document/9615580

[7] Talvinder Singh et al., "VXLAN and EVPN for data center network transformation," 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017. Available: https://ieeexplore.ieee.org/document/8203947

[8] D. Kong et al., "Toward Security-Enhanced In-Band Network Telemetry in Programmable Networks," IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 137–155, 2024. Available: https://ieeexplore.ieee.org/document/10764753

[9] W. Zhong and X. Zhao, "Research on DCI Streaming Telemetry System Based on gNMI," 4th International Conference on Information Science, Parallel and Distributed Systems (ISPDS), 2023. Available: https://ieeexplore.ieee.org/document/10235592

[10] P. Lechowicz et al., "Trade-Offs in Implementing Unsupervised Anomaly Detection with TAPI-Based Streaming Telemetry," IEEE 25th International Conference on High Performance Switching and Routing (HPSR), 2024. Available: https://ieeexplore.ieee.org/document/10635922

[11] Vignesh Karunakaran et al., "TAPI-Based Telemetry Streaming in Multi-Domain Optical Transport Network," Optical Fiber Communications Conference and Exhibition (OFC), 2024. Available: https://ieeexplore.ieee.org/document/10526531