PalC Disaggregated Packet Broker

Product Documentation

Purpose of This Document

To enable **technical evaluation**, **deployment, and operation** of the **PalC Disaggregated Packet Broker** in production data center environments.

This document answers:

- What problem the product solves
- How it is architected
- How it is deployed
- How it operates at scale
- What constraints and dependencies exist

1. Product Overview

Why Traditional Packet Visibility Breaks at Scale

Modern data centers generate predominantly **east–west traffic**, driven by:

- Microservices
- AI/ML workloads
- High-speed fabrics (100G → 800G)
- Distributed storage

Traditional TAP/SPAN-based visibility solutions fail because:

- They do not scale with fabric bandwidth
- They introduce bottlenecks at aggregation points
- They lack flexibility in multi-vendor environments.

PalC Disaggregated Packet Broker is designed to solve these limitations.

2. Architecture Overview

Disaggregated Design Principles

The PalC Packet Broker is built on:

- Whitebox switching platforms
- SONiC-based NOS
- Merchant silicon (Broadcom / Marvell).
- Software-defined traffic steering

This architecture enables:

- Linear scalability with fabric growth
- Vendor-independent hardware selection
- Software-controlled visibility pipelines
- Software-defined traffic steering

3. Functional Capabilities

Traffic Ingestion

- Line-rate ingestion at 100G / 400G / 800G
- Support for SPAN, ERSPAN, and fabric-level mirroring
- Lossless handling for bursty east–west flows

Traffic Processing

- Flow-based filtering
- Header slicing and masking
- Replication and load-balanced forwarding

Traffic Distribution

- Tool-aware forwarding
- Load balancing across monitoring and security tools
- Multi-tenant visibility separation

4. Deployment Models

Traffic Distribution

- Inline packet visibility in leaf–spine fabrics
- Out-of-band monitoring architectures
- AI fabric observability
- Security and compliance monitoring

Integration Points

- SONiC-based fabrics
- NetPro Suite for orchestration
- Prometheus / Grafana
- Third-party SIEM, NDR, IDS tools

5. Operational Considerations

Scalability

- Horizontal scale-out using additional Whitebox nodes
- No centralized bottlenecks

Observability

- Real-time telemetry for packet loss, latency, utilization
- Integration with centralized dashboards

Reliability

- Redundant paths
- Stateless forwarding design
- Graceful failure handling

6. Design Notes

This product assumes:

- Operators understand fabric-level mirroring
- Monitoring tools can handle high-speed traffic
- Packet visibility is treated as infrastructure, not an afterthought

This documentation intentionally highlights **real-world constraints**, not idealized scenarios.