TECHNICAL PAPER: **June 2026**

Breaking the Bottleneck: A Deep Dive into Enhanced Datapath

Tech Paper Version 1.0.06-02

Table of Contents

Revision History 3

Intended Audience 4

Introduction to this Document 4

Architectural Prerequisites and Concepts 4

0 Modern Networking Infrastructure in ESXi 4

1 The Network Stack & Virtual Switch Modes 5

1.1 Standard Datapath (The “Slow Path”) 5

1.2 Enhanced Data Path – Standard (The “Fast Path”) 5

1.3 Enhanced Data Path – Dedicated (Poll Mode) 5

2 Strategic Hardware Integration and Offloads 6

2.1 NUMA Optimization 6

2.2 Virtual Device Requirements: VMXNET3 6

2.3 Hardware Offloads: TSO and LRO 6

2.4 Scaling with NetQueue and Receive Side Scaling (RSS) 6

2.5 NetQueue Optimization 7

2.6 Receive Side Scaling (RSS) 7

2.7 Guest vNIC RSS & vCPU Queue Scaling 7

2.8 Receive & Transmit (TX) Scaling and Advanced Configuration Reference 8

3 Enhanced Data Path Standard Core Technologies 9

3.1 Flow Cache and the EDP Fast Path 9

Flow Cache Mechanics 9

Flow Cache Management and Timeout Behavior 9

3.2 Thread Load Balancer 10

3.3 EDP Mbuf Framework 10

3.4 Capstone: The Synergistic Datapath 10
4 Performance Analysis and Operational Impact 12
4.1 Study I: Throughput Efficiency and Resource Reclamation 12
4.2 Study II: Operational Stability Under IPFIX Visibility 14
Conclusion 15
Looking Ahead: The Foundation for Next-Gen Acceleration 15
Appendix 16
Appendix A | Software Versions for EDP Standard in VMware Cloud Foundation 16
EDP in VCF 5.2.X Implementations 16
EDP in VCF 9.X Implementations 16
Appendix B | Driver Compatibility for EDP Standard 17
Verifying Hardware Readiness 17
EDP Standard Driver Minimums & Compatibility Status 17
Appendix C | Glossary 18
Appendix D | Flow Cache Action by Packet Types 19
About the Author and Contributors 21

Revision History

Date | Author	Summary of Revisions
June 2026	Initial release on VCF Blog.
May 2026	Rough cut shared on vcfcore.tech

Intended Audience

This document is intended for technology administrators and designers with an interest in the network performance capabilities of their VMware Cloud Foundation deployment. This document is for users wanting to understand the VMware network technologies that affect all connected workloads. Solution architects, technical sales engineers, field consultants, and advanced services specialists in the telecommunication industry may find this guide beneficial. While we provide foundational information related to vSphere networking, it helps if readers have a basic understanding of general ethernet networking, IP, VLANs and server hardware components.

Introduction to this Document

This paper presents the design and implementation of **Enhanced Data Path (EDP) Standard** — a high-performance virtual switch mode introduced with NSX. EDP Standard significantly enhances packet processing efficiency and CPU resource utilization with minimal configuration effort in VMware Cloud Foundation deployments.

EDP Standard leverages multiple advanced techniques, including NUMA-aware thread placement, dynamic resource management, compact packet data structures, flow-based caching, and alignment with modern multi-core and offload technologies.

While the paper includes foundational context where necessary, its core focus is to explain how EDP Standard works, why it improves performance, and how to implement it effectively using current best practices.

Architectural Prerequisites and Concepts

To understand the performance gains of EDP Standard, one must first evaluate the interaction between the ESXi hypervisor and modern networking hardware. This section establishes the technical baseline for packet processing, device abstraction, and the evolution of the virtual datapath.

0 Modern Networking Infrastructure in ESXi

The transition to high-bandwidth environments (100Gbps to 400Gbps) requires a fundamental shift in how the hypervisor interfaces with physical and virtual network layers. In an EDP-enabled environment, the relationship between the Physical Network Interface Card (pNIC) and the Virtual Network Interface Card (vNIC) is no longer just about connectivity – it's about path optimization.

- **pNIC Integration:** Modern pNICs provide the programmable hardware acceleration and offload technologies (Geneve, TSO, LRO) that EDP Standard orchestrates.
- **Virtual Device Alignment:**

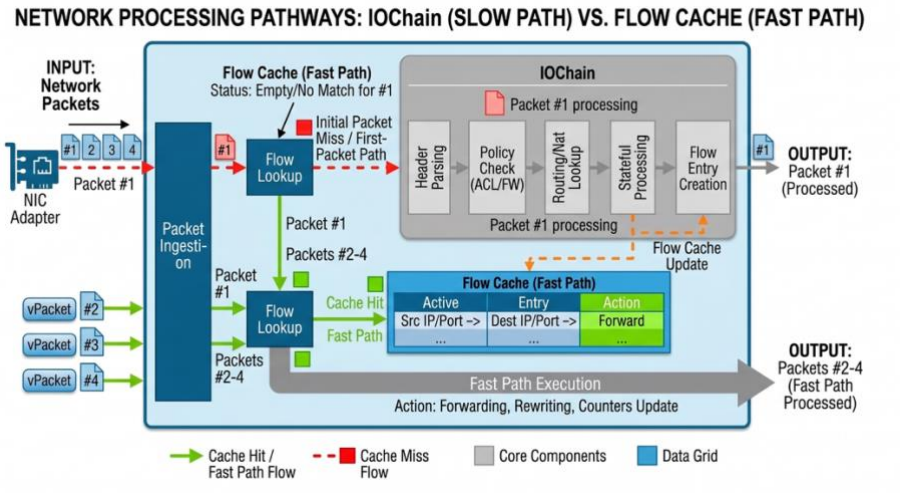
1 The Network Stack & Virtual Switch Modes

The network stack is the ESXi kernel subsystem responsible for all packet I/O. Under the VMware Cloud Foundation framework, the stack oeprates in two primary modes:

1.1 Standard Datapath (The "Slow Path")

The Standard mode is the heritage interrupt-stack, it uses the heritage IOChain framework for ESX networking. While it can be used reliably for generating EDP fast-path actions, it is not architected for the high-packet-rate demands on modern virtualized firewalls, load balancers or performance-dependent virtual workloads.

- **Operational Cost:** Packets traverse a linear series of modules for parsing, classification, and lookups.
- **Resource Allocation:** Uses PollWorlds for processing, which lack the dynamic rebalancing and NUMA-awareness found in EDP.



1.2 Enhanced Data Path – Standard (The "Fast Path")

EDP Standard is the recommended virtual switch mode for performance-critical workloads. It introduces a Fast Path that allows packets to bypass the IOChain once a flow is learned.

- **Efficiency:** Delivers 2.5X more efficiency (50-70% better performance) than a heavily tuned **Standard** stack. [1]
- **Key Innovation:** Replaces the legacy pktHandle format (256 byte) with the Mbuf-based framework (128 bytes), reducing the cache footprint by 50%.

1.3 Enhanced Data Path – Dedicated (Poll Mode)

Unlike the interrupt-based Standard and EDP Standard modes, **EDP Dedicated** utilizes a poll-mode mechanism.

- **Use Case:** Strictly for Telco/5G NFC workloads requiring dimensioning 100% core reservation.

[1] *As observed in large scale production deployment with one of our customers. Observed performance my vary.*

Constraint: Reserved cores are unavailable for general workloads, even when idle. EDP Dedicated requires rigorous upfront dimensioning, it is a less dynamic model and may not suit the 'bursty' requirements of general enterprise environments.

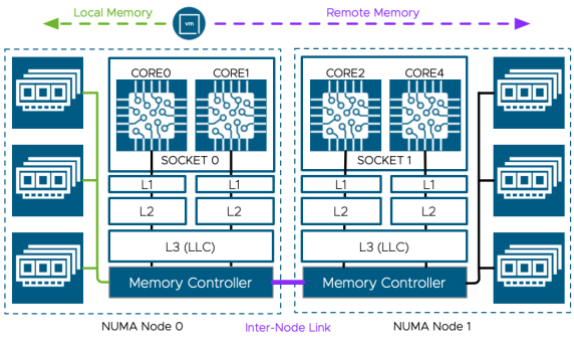
2 Strategic Hardware Integration and Offloads

To achieve the highest performance at 100Gbps and beyond, EDP Standard orchestrates several hardware-level technologies. EDP Standard natively orchestrates these features out-of-the-box, ensuring the datapath scales dynamically without manual tuning.

2.1 NUMA Optimization

EDP Standard is natively NUMA-aware, ensuring that pNIC operations, packet-forwarding threads (EnsNetWorlds), and Virtual CPUs prioritize NUMA alignment.

- **Performance Impact:** Accessing local NUMA memory provides the highest bandwidth and lowest latency.
- **Best Practice:** Maintain the ESXi NUMA scheduler at its default settings; manual node affinity is generally unnecessary as EDP Standard dynamically aligns threads to the correct node.



2.2 Virtual Device Requirements: VMXNET3

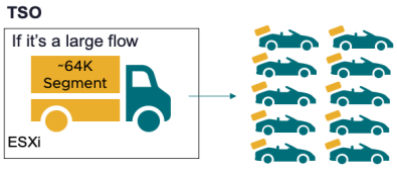
The performance innovations of EDP Standard are exclusive to the VMXNET3 para-virtualized adapter.

- **Constraint:** While legacy emulated adapters (E1000/E1000E) are compatible, they will not provide a performance improvement compared to the non-EDP Standard mode.
- **Recommendation:** Ensure all high-performance workloads are equipped with VMXNET3 to support advanced offloads like RSS, TSO, and LRO.

2.3 Hardware Offloads: TSO and LRO

EDP Standard automatically leverages TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) when supported by the underlying hardware.

- **Geneve Acceleration:** Hardware TSO is critical for encapsulated traffic, allowing the pNIC to handle segmentation and reducing CPU overhead by nearly 50%.
- **Optimization:** EDP Standard defaults to hardware-level offloads but will fall back to software-based processing with reduced efficiency if hardware support is not available.



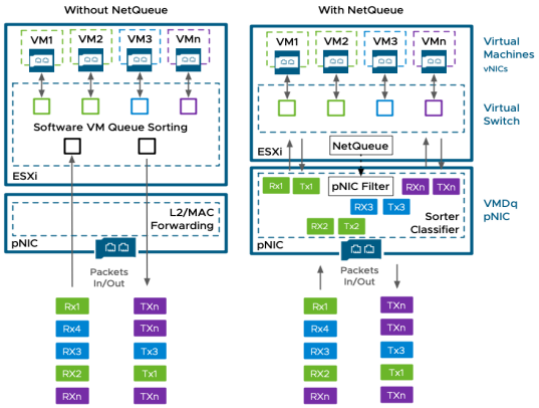
2.4 Scaling with NetQueue and Receive Side Scaling (RSS)

EDP Standard uses NetQueue and RSS to parallelize packet processing across multiple CPU cores, preventing single-core bottlenecks. The EDP Thread Load Balancer (TLB) maps RSS queues to packet-processing threads based on traffic load and thread utilization.

2.5 NetQueue Optimization

In legacy architectures, the virtual switch relies on a single vCPU to sort and distribute inbound packets, creating a severe bottleneck under heavy load. NetQueue mitigates this by moving the packet sorting function down to the physical NIC.

Using a programmable receive (Rx) filter, the pNIC Virtual Machine Device Queues (VMDq) classify inbound traffic by MAC address and VLAN tag, placing packets directly into the receive queue assigned to the destination VM. EDP Standard is fully NetQueue-aware; physical NIC queues are automatically assigned as individual NetQueues without requiring any manual configuration.



2.6 Receive Side Scaling (RSS)

While NetQueue maps traffic to specific VM queues, Receive Side Scaling (RSS) prevents individual CPU core saturation by distributing inbound traffic across multiple hardware receive queues. The pNIC uses a 5-tuple hash (source and destination IP, source and destination port, and protocol) to parallelize packet processing across multiple CPU cores.

Device RSS vs. NetQueue RSS

Device RSS and NetQueue are mutually exclusive technologies.

- Device RSS: The adapter hardware hashes and distributes packets across hardware queues entirely independently of the ESXi switch.
- NetQueue RSS: Uses NetQueue to place VM-bound traffic into designated pools of RSS queues managed by an RSS Engine.

For EDP Standard deployments requiring more performance than a single NetQueue can provide, Shared NetQueue RSS is the recommended architecture. It allows an RSS engine to be shared by multiple virtual machines assigned to it, balancing high throughput with resource efficiency. Dedicated NetQueue RSS, conversely, dedicates a full RSS engine to a single VM, which is not recommended in EDP Standard deployments due to severe resource cost and inability to guarantee performance during mobility events.

2.7 Guest vNIC RSS & vCPU Queue Scaling

While the hypervisor manages pNIC scaling, the guest operating system must also be configured to process multiple queues. Guest vNIC RSS depends entirely on the addition of vCPUs and the availability of the RSS capability in the virtual machine guest OS and the VMXNET vNIC driver.

- **Linux Guests:** Multi-queue support is included in the VMXNET3 Linux device driver. Hardware Version 18 (VMXNET3 version 6) supports up to 32 queues.
- **Windows Guests:** RSS must be validated via the Get-NetAdapterRss cmdlet and enabled within the Device Manager UI properties for the VMXNET3 adapter.
- **Note on NSX Edges:** In EDP Standard deployments, vNIC RSS is configured by default for NSX Edges.

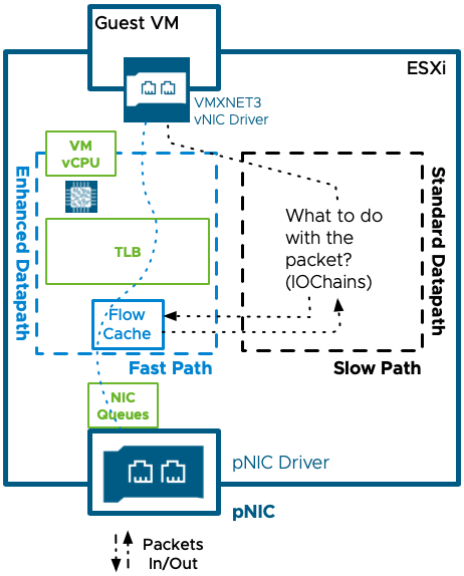
2.8 Receive & Transmit (TX) Scaling and Advanced Configuration Reference

By default, EDP Standard dynamically scales and re-balances packet processing threads, making the default TX settings optimal for most workloads on the host. However, for specialized workloads where packet processing for the single VM is identified as the bottleneck, administrators can manually define how networking threads are allocated for a VM's outbound traffic using the Multiple Contexts (ctxPerDev) setting.

The following advanced VMX parameters dictate how the VM interacts with the hypervisor's NetQueue RSS and Context scaling mechanisms:

Setting	VMX Parameter	Value	Architectural Impact
Shared NetQueue RSS	**ethernetX.pnicfeatures**	"4"	Scales queue availability for vNIC traffic using a Shared RSS engine.
Dedicated NetQueue RSS	**ethernetX.rssoffload**	True	Dedicates a full RSS Engine to a single VM. Generally not recommended (included for completeness).
VM-Level TX Scaling	**ethernetX.ctxPerDev**	2	Default. Uses one network TX thread to process the VM traffic.
vNIC-Level TX Scaling	**ethernetX.ctxPerDev**	1	Assigns a network TX thread per VM vNIC.
Queue-Level TX Scaling	**ethernetX.ctxPerDev**	3	Assigns a network TX thread per vNIC queue, up to 8. Multi-context per vNIC queues are automatically configured for VCF Networking Edge VMs

3 Enhanced Data Path Standard Core Technologies



The defining characteristic of Enhanced Data Path Standard is its ability to bypass legacy packet-processing bottlenecks. Once enabled, EDP Standard automatically synchronizes underlying hardware technologies to scale the datapath without exhausting CPU cycles. It achieves this through three tightly integrated architectural components: Flow Cache, the Thread Load Balancer (TLB), and the Mbuf data framework.

3.1 Flow Cache and the EDP Fast Path

The legacy Standard Datapath processes traffic via the **IOChain**, a serial execution environment where every packet undergoes discrete parsing, classification, and locking operations. In modern software-defined data centers—characterized by distributed routing, granular segment security, and pervasive telemetry—this serial overhead creates a performance ceiling that scales proportionally with packet volume. This 'Slow Path' architecture introduces compounding latency and excessive CPU cycles for high-bandwidth workloads.

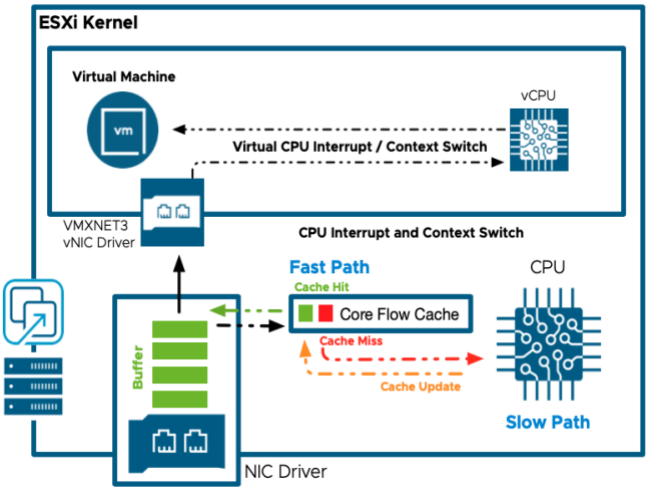
Flow Cache Mechanics

EDP Standard mitigates this overhead by monitoring initial packet behavior and recording the required handling instructions (actions) into a Flow Cache.

- **Cache Update** (The First Packet): When a new flow is initiated, the first packet must traverse the Slow Path. Once the IOChain resolved the necessary actions, the system stores those instructions as a Flow Cache entry.
- **Cache Hit** (The Fast Path): Subsequent packets matching the flow signature bypass the IOChain entirely. The system applies the cached actions via a single, rapid lookup, dramatically reducing latency and improving efficiency.

Flow Cache Management and Timeout Behavior

To prevent the cache from serving stale forwarding rules or overloading, EDP utilizes an automated eviction algorithm.



- **Randomized Timeouts:** Each entry is assigned a randomized expiration timer between 90 and 150 seconds. This staggering prevents simultaneous cache evictions and forces long-lived flows to periodically refresh and revalidate behavior. This is not a user-configurable setting.
- **Capacity Handling and Expansion:** Instead of a single global table, EDP allocates a dedicated flow cache table for each active packet-processing thread, with each table capable of storing up to 128K entries. If a massive volume of micro-burst traffic fills a specific thread's table, the system can dynamically expand overall cache capacity by spawning new processing threads. In the event that capacity is temporarily exhausted

across all active threads, new flows are routed exclusively through the Slow Path while the system waits for short-lived entries to seamlessly age out.

3.2 Thread Load Balancer

EDP Standard utilizes specialized packet processing threads known as EnsNetWorlds. To prevent thread saturation and maintain performance, the Thread Load Balancer (TLB) dynamically orchestrates the mapping of these EnsNetWorlds to the corresponding vNIC and pNIC queues based on real-time traffic load.

Dynamic Scaling and Rebalancing Criteria

The TLB continuously evaluates thread utilization on a 4-second polling cycle. Rebalancing operations are triggered by topological changes, such as the addition or deletion of virtual machine vNICs, or when an active ENSNetWorld exceeds 70% utilization. When rebalancing is required, the TLB executes the following workflow:

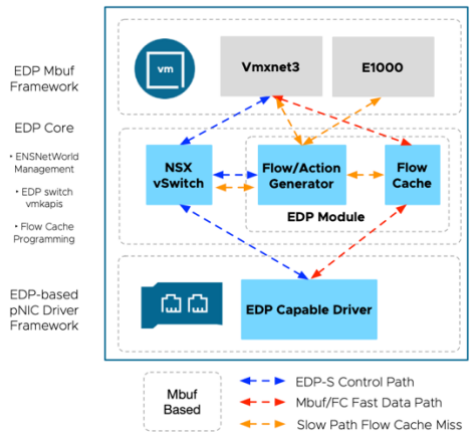
1. **Capacity Estimation:** Evaluates the current load on threads and the demand from datapath components.
2. **Bin-Packing Distribution:** Uses a bin-packing algorithm to seamlessly migrate high-load datapath elements onto low-load EnsNetWorlds.
3. **NUMA Enforcement:** Prioritizes vNIC-to-EnsNetWorld NUMA alignment to maintain maximum memory locality and performance. The TLB will only migrate processing to a remote NUMA node under extreme load conditions, selecting the least-loaded remote thread group.

3.3 EDP Mbuf Framework

To maximize the efficiency of the Fast Path, EDP relies on an optimized packet data structure. The legacy network stack utilizes a representation called 'pktHandle', which requires 256 bytes (four cache lines) of memory per packets.

EDP replaces this with the Mbuf Framework, a highly optimized data structure based on the open-source DPDK (Data Plane Development Kit) library.

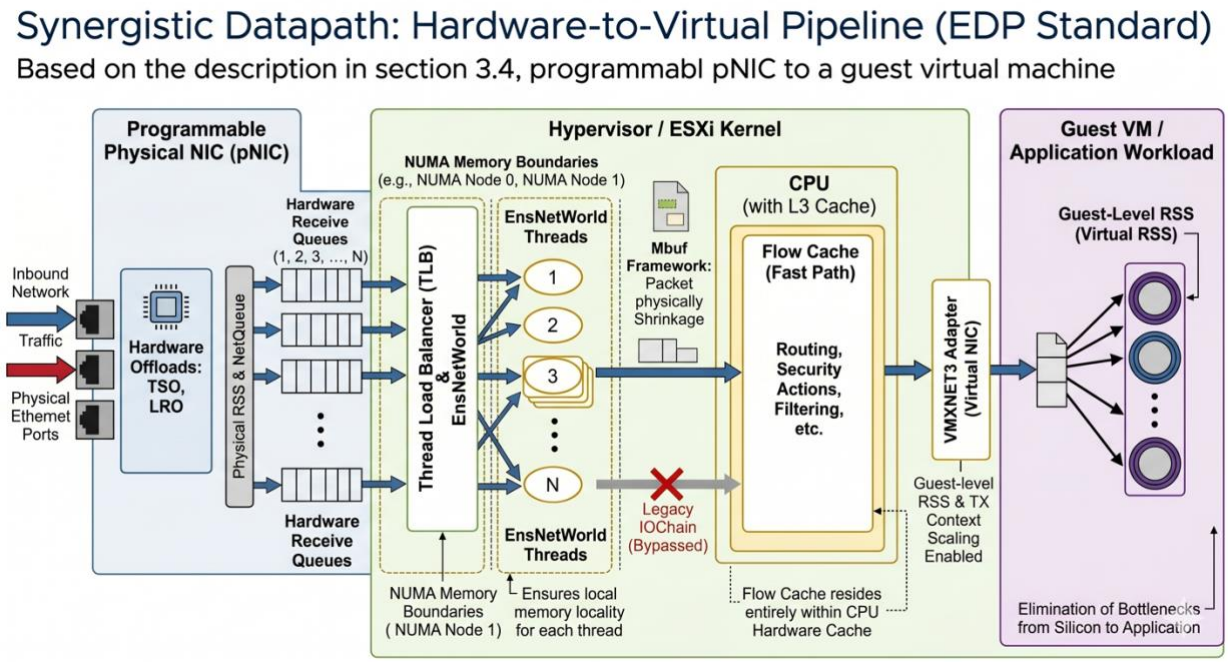
- **Cache Optimization:** Mbuf utilizes a highly efficient 128-byte representation, perfectly aligning with two hardware cache lines.
- **Architectural Impact:** By reducing the memory footprint by 50%, EDP allows significantly more packets to reside in the hardware cache. This low-latency memory access is crucial for optimal performance during packet initialization, allocation and deallocation.



3.4 Capstone: The Synergistic Datapath

The true performance of EDP Standard is realized through the **simultaneous orchestration** of the entire hardware-to-software pipeline. This synergistic flow commences at the physical layer, where programmable NICs execute hardware offloads (TSO/LRO) and leverage performant RSS engines to parallelize inbound streams across multiple hardware queues.

Upon entering the hypervisor, the **Thread Load Balancer (TLB)** dynamically maps these queues to **EnsNetWorld threads** while strictly enforcing NUMA memory boundaries to minimize cross-socket latency. Within these threads, the **Mbuf framework** compacts the packet data structure, allowing the **Flow Cache** to reside entirely within the CPU's L3 hardware cache. This architecture enables the system to bypass the legacy IOChain, applying complex routing and security policies at line rate. The resulting optimized stream is delivered to the guest via the **VMXNET3 adapter**, where guest-level RSS ensures the application can ingest the high-velocity throughput generated by the underlying infrastructure.



4 Performance Analysis and Operational Impact

The Enhanced Data Path (EDP) Standard delivers a superior performance profile by decoupling packet velocity from CPU overhead. The following data sets evaluate EDP Standard across two critical dimensions: raw throughput scaling and operational stability under deep-packet visibility (IPFIX).

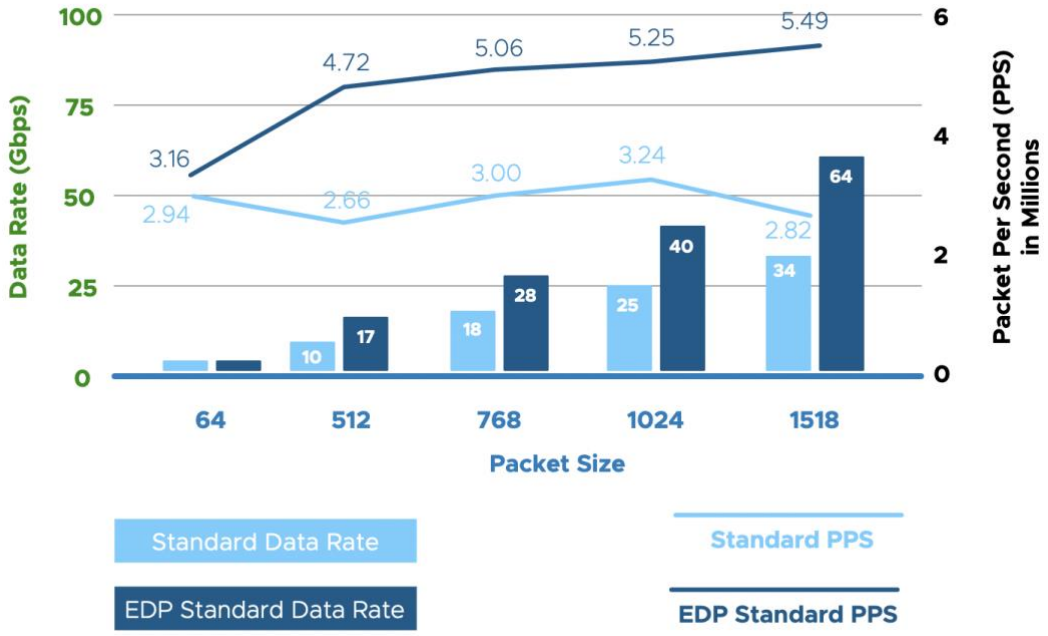
4.1 Study I: Throughput Efficiency and Resource Reclamation

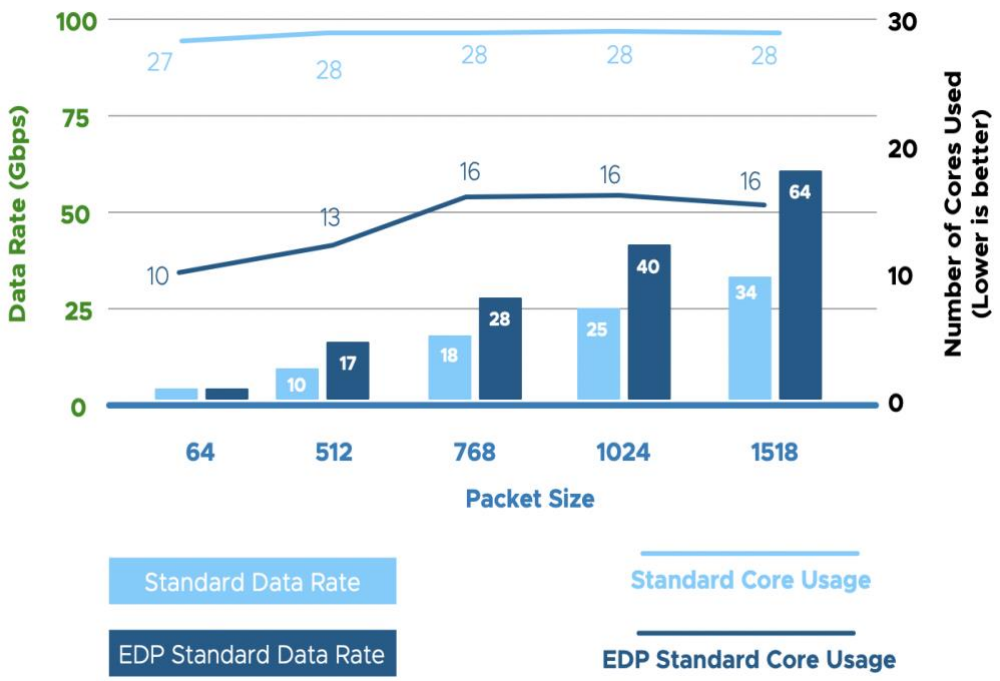
To establish a baseline, testing was conducted using **NSX 4.2 Virtual Edge X-Large** form factor on an **Intel® Xeon® Platinum 8480+** platform. The results confirm that EDP Standard removes the architectural throughput ceilings inherent in the legacy "Slow Path."

- **Throughput and Velocity:** Across standard datacenter packet sizes, EDP Standard delivers **50–70% higher performance**. For 1518-byte frames, throughput increases from 34 Gbps in Standard mode to **64 Gbps** with EDP.

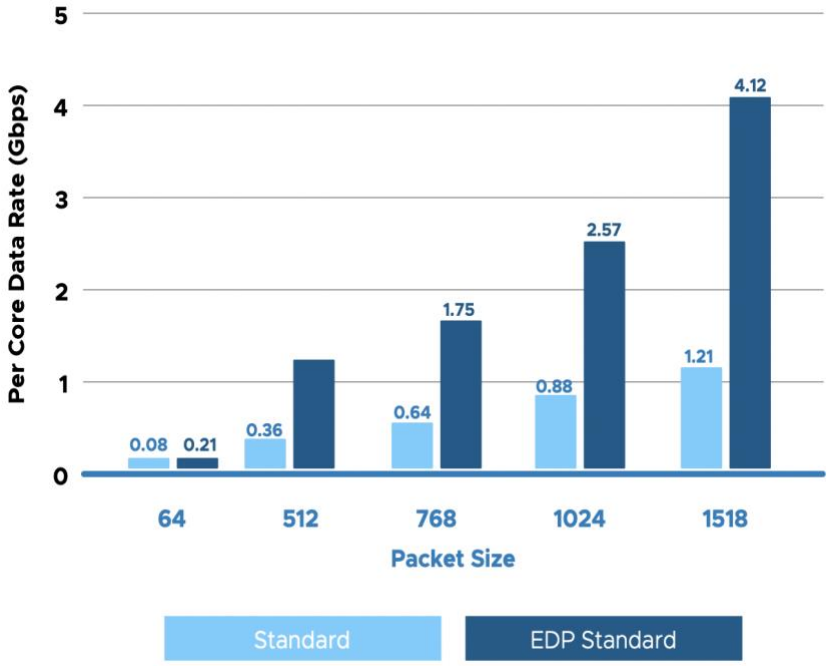
- **Decoupling Performance from Cost:** While the Standard stack requires up to **28 cores** to manage high traffic volumes, EDP Standard maintains a static, optimized footprint of only **16 cores**.

- **Strategic ROI:** This represents a **40% reduction in CPU requirements**. By delivering **2.5x to 3.5x more throughput-per-core**, organizations can reclaim significant compute capacity for application workloads.





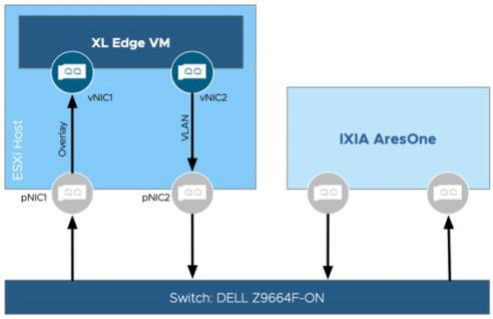
- **Normalized Core Efficiency:** When analyzing the **Per Core Data Rate (Gbps)**, EDP Standard demonstrates superior architectural scaling. For 1518-byte packets, EDP Standard achieves a processing rate of **4.12 Gbps per core**, compared to only **1.21 Gbps per core** for the Standard Datapath—rendering EDP Standard approximately **3.4 times more efficient** at the individual core level.



- **Test Topology**

  To ensure high-fidelity results, testing was conducted in a controlled environment using industry-standard traffic generation. This setup demonstrates the "out-of-the-box" performance of EDP Standard against a manually tuned heritage stack.

  The evaluation utilizes an **NSX Edge X-Large** form factor hosted on an ESXi 8.0 Update 3 node. Traffic is orchestrated via an **IXIA AresONE** generator connected through a high-density **DELL Z9664F-ON** 100G switch.



- **Methodology & Configurations:**

  The performance deltas were measured under a high-stress UDP profile designed to expose datapath bottlenecks.

  Traffic Profile: 2 million concurrent UDP flows with a strict 0.001% drop tolerance.

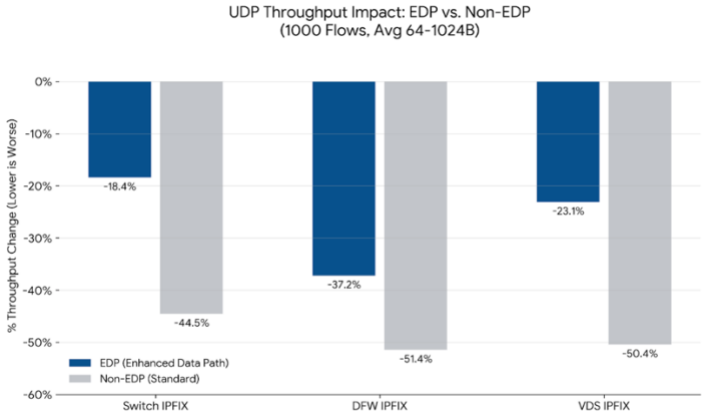
  Fabric Speed: Dual 100G ports utilizing an MTU of 1700 to reflect modern datacenter encapsulation requirements.

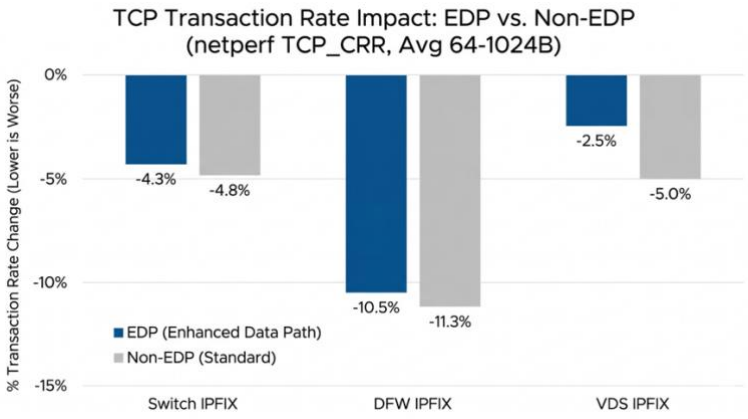
  Comparative Baseline: The Standard Datapath was manually tuned for optimal performance based on VMware best practices, whereas EDP Standard was tested with **zero manual tuning** (Out-of-the-box).

4.2 Study II: Operational Stability Under IPFIX Visibility

Visibility through **IPFIX** is critical for security and compliance, but it typically introduces a "performance tax" on the host. Recent evaluations of **VCF 9** architectures demonstrate that EDP Standard is a prerequisite for maintaining network integrity when deep monitoring is enabled.

- **Mitigating the "Performance Tax"**: In high-stress UDP scenarios (1,000 flows), heritage "Non-EDP" paths observe performance drops exceeding **50%** when DFW or VDS IPFIX is enabled. EDP Standard significantly mitigates this, restricting the impact to as low as **18.4%**.
- **Application Response Integrity**: Impact on **TCP Transaction Rates (TCP_CRR)** remains marginal (2.5% to 11.3%) with EDP enabled, ensuring snappy application response times even under heavy monitoring.
- **The VCF 9 Advantage**: Starting with **VCF 9.0 (Switch IPFIX)** and **9.1 (VDS IPFIX)**, a new **ConnTrack module** further improves the efficiency of flow tracking, making high-fidelity monitoring more accessible.





Conclusion

The transition to high-bandwidth environments requires the fundamental shift in hypervisor interfacing provided by the Enhanced Data Path. As demonstrated throughout this paper, EDP Standard is no longer merely an optional optimization but the foundational architecture for modern VMware Cloud Foundation deployments.

By leveraging the **2.5x efficiency gain** and the **Synergistic Datapath** pipeline, organizations can achieve a **40% reduction in CPU overhead** while maintaining the deep visibility required for security and compliance. This architecture ensures that as network demands scale, the compute resources required to manage them remain predictable and optimized, effectively eliminating bottlenecks from silicon to application.

Looking Ahead: The Foundation for Next-Gen Acceleration

Beyond its immediate performance benefits, EDP Standard serves as the essential architectural foundation for VCF 9.1's most advanced hardware features: **Uniform Passthrough (UPT)** and **Enhanced DirectPath I/O (EDPIO) for NVIDIA Accelerated NICs**. As modern replacements for traditional SR-IOV, these technologies offer direct hardware access to NVIDIA Connect-X and BlueField adapters for AI, Machine Learning, and HPC workloads. Crucially, by building upon the EDP Standard framework, these modern options maintain full compatibility with essential cluster features—including **vMotion, DRS, and High Availability (HA)**—ensuring administrators no longer have to compromise between raw passthrough performance and critical workload mobility.

With these performance benchmarks established, the following appendices provide the necessary software versions, driver compatibility minimums, and technical definitions required to initiate an EDP Standard deployment.

Appendix

Appendix A | Software Versions for EDP Standard in VMware Cloud Foundation

References to VCF X.X.X denote the integrated stack of vSphere (ESXi, vCenter) & NSX Networking, applicable to both formal VMware Cloud Foundation (VCF) Bill of Materials (BOM) and equivalent standalone component configurations.

EDP in VCF 5.2.X Implementations

In VCF 5.2.X environments, **EDP Standard** is an optional but highly recommended mode for environments requiring high-performance networking or long-term architectural futureproofing. Adhering to validated releases is critical due to the complex interplay between CPU scheduling, NIC drivers and firmware.

- **Recommended Baseline (April 2026):** vSphere 8.0 Update 3i & NSX 4.2.3.2**.**
- **Rationale:** This combination incorporates critical stability patches for Network I/O Control (NIOC) and optimizes the automated enablement scripts provided within NSX.
- **Legacy Baseline Note:** vSphere 8.0 Update 3e and NSX 4.2.2.1 represent the absolute minimum version requirements. This is suitable for dedicated NSX Edge clusters without NIOC but is not recommended for general-compute clusters due to later critical bug fixes.

EDP in VCF 9.X Implementations

In VCF 9.0.X, **EDP Standard** is the foundational network datapath architecture for the platform. The transition to this mode is now managed through a unified orchestration framework that reduces manual overhead and operational risk.

Deployment and Upgrade Behavior

- **Initial Provisioning:** EDP Standard is now the **default virtual switch mode** for all new VCF 9.X deployments, ensuring optimal CPU efficiency and throughput from day one.
- **In-Place Upgrades:** Upgrading from VCF 5.2.X to 9.X preserves the existing datapath state. If a cluster is running the legacy Standard stack, it remains on that stack post-upgrade unless explicitly transitioned.
- **Existing Configurations:** Clusters previously running EDP Standard in VCF 5.2.x will maintain that mode throughout the upgrade process.

Transition Orchestration: TNP and VCP

VCF 9.X introduces a more robust mechanism for datapath transitions, moving away from legacy CLI-based enablement scripts.

- **The Activation Trigger (TNP):** The NSX Transport Node Profile serves as the administrative trigger for the switch mode change. Modifying the TNP advanced settings to “Enhanced Datapath – Standard” communicates the architectural intent to the cluster.
- **The Orchestration Engine (VCP):** When vSphere Configuration Profiles (VCP) are enabled, VCF leverages this framework to execute the transition. VCP automates the heavy lifting by placing hosts into maintenance mode and applying the switch mode change sequentially across the cluster. This orchestrated approach ensures the datapath transition maintains the highest availability for virtual machines.
- Legacy Fallback: While manual enablement scripts are superseded by this automated workflow, they remain an option for environments where VCP is not yet active.

Appendix B | Driver Compatibility for EDP Standard

Verifying Hardware Readiness

By default, baseline hardware compatibility should always be determined using the **Broadcom Compatibility Guide (BCG) - IO Devices**. Before configuring your environment, verify the following in the guide:

- **Feature Verification:** The device must list the target ESXi version with **EDP Standard** under its supported features.

While the BCG is the source of truth for overall compatibility, certain driver versions harbor known bugs or performance penalties. For the specific drivers listed in the table below, you **must** use at least the listed minimum version to ensure stability.

EDP Standard Driver Minimums & Compatibility Status

NIC Vendor & Family	Driver Name	Minimum Version (Async/8.0U3)	EDP Status Notes / Known Issues Resolved
Broadcom NetXtreme-E *(BCM574xx series)*	bnxtnet	**233.0.31.0**	EDP Standard performance driver. Minimum version resolves an issue where Rx hang occurs.
Cisco VIC 14425, 14825 & 15XXX	nenic-ens	**1.0.18.0-1OEM**	EDP Standard performance driver. Minimum version resolves heap allocation failures. **Requires Geneve Offloading** to be enabled in the Cisco UCS Manager.
Intel E810 Series *(10/25/100GbE)*	icen	**2.2.2.0**	EDP Standard performance driver. Minimum version resolves heap allocation failure after many queue allocations, preventing traffic stops.
Intel 700 Series *(X710/XL710/XXV710)*	i40en	**2.11.1.0**	EDP Standard performance driver. Minimum version resolves ~10% packet drops occurring at defaultQ RSS / DQRSS.
Marvell FastLinQ *(QL41xxx / QL45xxx series)*	qedentv	**3.71.80.0**	EDP Standard performance driver. Minimum version resolves interrupt storm issues and prevents potential IOChain latency.
AMD Pensando Ionic	ionic_en	*Check BCG*	EDP Standard performance driver. No issues in 8.0U3, 9.0, 9.1
Mellanox/NVIDIA ConnectX	nmlx5_core	*Check BCG*	EDP Standard performance driver. No issues in 8.0U3, 9.0, 9.1

NIC Vendor & Family	Driver Name	Minimum Version (Async/8.0U3)	EDP Status Notes / Known Issues Resolved
Broadcom Legacy	elxnet, ntg3	N/A	Functions as EDP Standard compatible, but provides **no** high-performance gains.
Intel Legacy	igbn, ixgben, ne1000	N/A	Functions as EDP Standard compatible, but provides **no** high-performance gains.
Marvell	atlantic	N/A	Functions as EDP Standard compatible, but provides **no** high-performance gains.
AMD Solarflare	sfvmk	N/A	Functions as EDP Standard compatible, but provides **no** high-performance gains.
Cisco VIC 12XX, 13XX	nenic	N/A	Does not support EDP. Data Path Mode must be configured as **Standard**.
Marvell (HPE FlexFabric / NX2)	qfle3	N/A	Does not support EDP. Data Path Mode must be configured as **Standard**.

Appendix C | Glossary

ENSNetWorld (Threads): Specialized packet processing threads utilized by the EDP Standard. The Thread Load Balancer dynamically maps these to network queues based on real-time traffic load.

Fast Path: A processing route that allows packets to bypass the legacy IOChain once a flow's handling instructions are learned.

Flow Cache: A mechanism that monitors initial packet behavior and records the necessary handling instructions into a cache. This allows subsequent matching packets to bypass the IOChain entirely via a single lookup.

LRO (Large Receive Offload): A hardware-level offload technology that EDP Standard automatically leverages when supported by the underlying physical hardware.

Mbuf (Mbuf Framework): An optimized 128-byte data structure based on the DPDK library. It replaces the legacy format to shrink the packet payload and reduce the memory footprint by 50%.

NUMA (Non-Uniform Memory Access): A memory architecture where accessing local memory provides the highest bandwidth and lowest latency. EDP Standard natively aligns processing threads to local nodes.

pNIC (Physical Network Interface Card): The hardware adapter providing programmable hardware acceleration and offloads like TSO and LRO.

PollWorld: The processing module used by the legacy Standard network stack, which lacks the dynamic rebalancing and NUMA-awareness found in EDP.

RSS (Receive Side Scaling): A hardware feature that prevents individual CPU core saturation by distributing inbound traffic across multiple hardware receive queues. It uses a 5-tuple hash to parallelize packet processing.

Slow Path (Standard Datapath): The legacy IOChain where packets must traverse a linear series of parsing, classification, and locking operations.

TLB (Thread Load Balancer): An intelligent bridge that dynamically maps hardware queues to EnsNetWorld threads based on utilization while strictly enforcing NUMA memory boundaries.

TSO (TCP Segmentation Offload): A hardware-level acceleration feature that allows the pNIC to handle packet segmentation. It reduces CPU overhead by nearly 50%, especially for encapsulated traffic.

vNIC (Virtual Network Interface Card): The virtual network adapter assigned to a guest VM. The performance innovations of EDP Standard are exclusive to the VMXNET3 para-virtualized vNIC adapter.

vSwitch (Virtual Switch): The ESXi kernel subsystem responsible for all packet I/O. It operates in primary modes such as the Standard Datapath or Enhanced Data Path.

Appendix D | Flow Cache Action by Packet Types

Type of packets	Forwarding path	Comment
Unicast	Flow Cache	
Multicast	Flow Cache	
Fragmented IPv4 or IPv6 packets	Slow path	
IPv6 packets with options header	Flow Cache*	* Largely fast path with some exceptions.
ARP	Slow path	Multiple modules like VDL2/VDR and switch security perform multiple operations, like processing for ARP suppression.
ND	Slow path	Multiple modules like VDL2/VDR and switch security perform multiple operations, like processing for ND suppression.
DHCP	Slow path	Switch security needs to inspect those packets.
IGMP	Slow path	Flow cache only supports unicast and vSwitch performs snooping on IGMP traffic

BFD	Slow path	
Traceflow	Slow path	Flows need to be analyzed by various modules and so go through slow path
Broadcast packets	Slow path	Flow cache only handles unicast traffic
Unknown Unicast packets	Slow path	Flow cache only handles unicast traffic
Multicast packets	Flow cache*	* Up to 64 replications
Monitoring: ERSPAN IPFIX	Flow cache	
Mirrored non-ERSPAN packets	Slow path	Flow cache only handles unicast traffic
Promiscuous mode and sink ports	Slow path	Flow cache only handles unicast traffic
LLDP	Slow path	Link local packet that does not hit the fc-fast-path IO Chain and always take slow-path
LACP	Slow path	
TCP Segmentation Offload (TSO) Large Receive Offload (LRO) TCP Checksum Offload	Flow cache	
VDS Functions (Geneve Encapsulation/Decapsulation MAC/VLAN Lookups)	Flow cache	
NSX Distributed Router Functions (e.g. Ipv4 and Ipv6 Lookups, TTL Decrement)	Flow cache	
NSX Distributed Firewall (e.g. L4 stateful flow inspection)	Flow cache	
Valid IPv4 packets not matching previous rows	Flow cache	
Valid IPv6 unicast TCP, UDP or ICMP packet	Flow cache	

About the Author and Contributors

Gabe Rosas is a Product Manager at Broadcom within the VMware Cloud Foundation (VCF) division, focusing on **Enhanced Data Path (EDP)**, Network Performance, and Hardware Offloads. During his tenure at VMware, he spearheaded early software-defined cloud initiatives, notably leading migration and performance for **VMware HCX**. A U.S. Navy veteran and father of six, he loves sharing insights on his blog at **vcfcore.tech**.

Contributors

Samuel Kommu currently works at Broadcom, in the VMware Cloud Foundation division, focusing on profiling network and compute resource usage of various applications and their performance, relevant bench-marking approaches and design recommendations.

Jin Heo is a Staff Engineer at Broadcom and a UIUC Ph.D. specializing in NSX network data plane development and performance optimization. An original member of the Enhanced Data Path (EDP) team, he leads technical projects in Enterprise & Telco-grade virtual switching, DPDK acceleration, and NIC offloads.

Ankur Sharma is a Datapath Engineer at Broadcom (VCF division) with nearly a decade of experience in **distributed systems** and high-performance networking. He **focuses** on network virtualization, SmartNIC integration, and architectural offloads like UPT and EDPIO. Ankur holds degrees from NIT Prayagraj and Carnegie Mellon University.

Jerome Catrouillet is a Product Management lead in the VCF division at Broadcom. In the VCF Networking team, Jerome leads NSX & vSphere Networking product strategy as well as NSX Federation. Jerome has 20 years of experience across networking architecture for cloud, datacenter, and service provider in France and United States.

Ken Guo serves as a Product Manager on the VCF Networking team. He leads Network Operations and Tooling. He brings more than 18 years of expertise in computer networking across engineering and product roles.

Copyright © 2026 Broadcom. All rights reserved.

The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Item No: vmw-bc-wp-tech-temp-uslet-word-2026 1/24