AMD SOLARFLARE X4

AMD SOLARFLARE™ X4 LOW LATENCY USING SDCI

WHITEPAPER | 2025

Authors: *Ashwin Thiagarajan, Sr.Manager; Paul Emberson, PMTS; Kieran Mansley, Fellow SW engineering; Greg Shippen, Fellow Silicon design engineering. Nikki Wearing, Sr SW Engineer*

Innovation in Ethernet network adapter technology has advanced to address a wide range of use cases, applications, and services required by diverse markets. The financial technology (or **Fintech**) market requires low latency, wire speed packet performance, deterministic packet jitter, and specialized multicast processing. These requirements are addressed with AMD Solarflare OpenOnload™ and network adapters, optimization in tuning of the CPU, BIOS and software. This paper describes how Smart Data Cache Injection (SDCI) enabled by AMD EPYC™ Gen 5 CPUs helps improve network latency and application performance.

- **Network latency** is the time spent by a packet to traverse from transmitter to receiver, reported in nanoseconds.

SMART DATA CACHE INJECTION (SDCI) - INTRODUCTION

Ethernet adapters generally use Direct Memory Access (DMA) to transfer the received data from the Ethernet interface to a host memory location pointed by the receive descriptors posted by the device driver. After the DMA, the NIC will generate an event to notify the host that the packet delivery has been completed. An application pinned on a specific CPU will then consume the data. On an AMD EPYC "Zen 5", a core issuing host memory read to CPU cache uses variable load-to-use latency which are greater than 14 clock cycles. SDCI is an enhancement in EPYC "Zen 5" CPUs that allows an IO device to route the ingress Ethernet traffic directly into the L2 cache of the CPU core that is executing the user application. With the SDCI enabled NIC, data is directly transferred to the L2 cache of the relevant CPU core that executes the user application, thus lowering the latency.

AMD has taken an end-to-end approach in reducing overall latency in real customer use cases. SDCI using PCIe® Transaction Layer Processing Hints (TPH) in the CPU along with latency optimized NIC and OpenOnload™ kernel bypass technology can reduce the receive latency of the packet. Onload, along with the implementation of Cut Through Programmable IO (CTPIO) in the NIC, addresses transmit latency. This holistic approach provides low end-to-end latency.

SOLARFLARE™ X4 NIC- SDCI PERFORMANCE

Latency performance is measured with **eflatency** and **sfnt-pingpong** benchmarking tools. Eflatency is implemented using ef_vi, a latency-optimized network layer L2 kernel bypass library for AMD Solarflare adapters. Sfnt-pingpong is written using the BSD sockets API and can use either UDP or TCP transports. It can be accelerated transparently using the Onload kernel bypass preload. Both benchmarks are ping-pong style applications which measure the time for a transmitted packet to be detected and responded to by a peer to produce a ½ round-trip delay. This white paper uses results from both benchmarks to show latency improvements with SDCI.

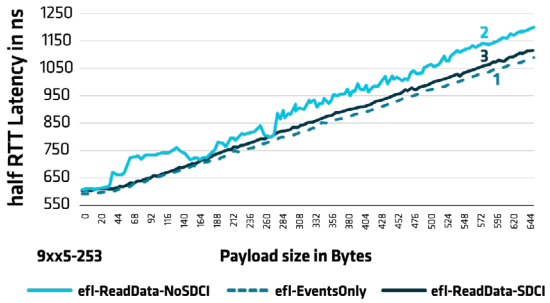
Performance – Eflatency

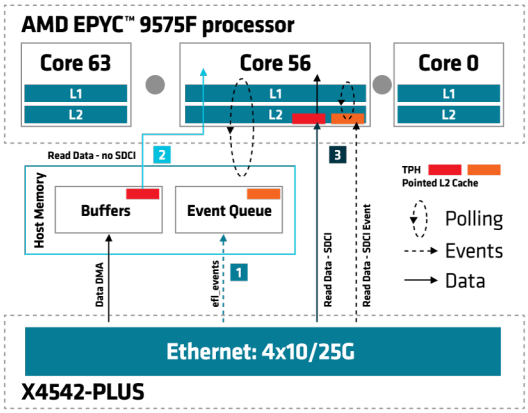
The eflatency graph represents latency reduction for 0 to 648B payload size, when Eflatency application measures the latency with and without SDCI. See end note 9xx5-253 for system config.

Evaluation configuration

Operating system - Ubuntu 25.04 with kernel 6.14.x; CPU - AMD EPYC 9575F (64 Core); Graphs- 50% median values.

X4542: 10G:ef_vi app: Events vs SDCI





Eflatency: 55ns (6.1%) average reduction in latency for 0 to 648B payload

A SDCI supported NIC, with eflatency that consumes data, produced 55ns (6.1%) lower latency compared to eflatency that consumes data placed on host memory.

Once NIC transfers the data to host memory, the eflatency user application receives a notification from the NIC.

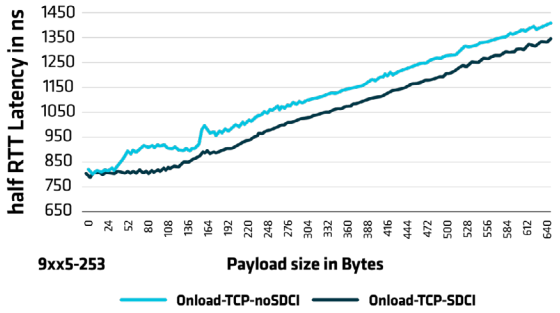
1. The efl-EventsOnly represents the packet latency where the eflatency application will not read the payload data from host memory (default configuration)
2. The efl-ReadData-NoSDCI represents the packet latency where the application consumes the data residing in the host memory, after receiving events from the NIC.
3. The efl-ReadData-SDCI represents packet latency where the data is placed in L2 cache using SDCI for the application to consume data.

With SDCI enabled, the NIC provides TPH for the CPU to post the data directly to L2 cache of a targeted core where the user application is running.

When the user application uses a SDCI enabled NIC with 5th Gen EPYC CPUs, the NIC will write TPH steering tags in PCIe TLPs so the CPU can steer ethernet data and events directly to the CPU L2 cache. TPH is supported on X2, X4, and Onload libraries which detect application cores and configure NIC steering tags used by 5th Gen EPYC CPUs. The user application affinized to the CPU consumes the data directly from the L2 cache, thus achieving low latency. SDCI enabled NIC achieves lower latency by reading data from L2 cache instead of host memory.

Performance – Onload

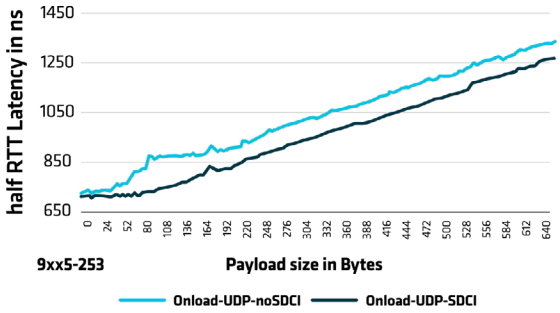
X4542: Onload - TCP: SDCI vs noSDCI



Onload-TCP: 67ns (6.1%) average reduction in latency for 0 to 648B payload

With Onload user networking stack, the TCP data is delivered to the sfnt-pingpong benchmarking tool. The Onload network stack inspects the packet header and copies data to the application. With SDCI enabled, sfnt-pingpong TCP processing observed 67ns (6.1%) average reduced latency compared to data placed in host memory.

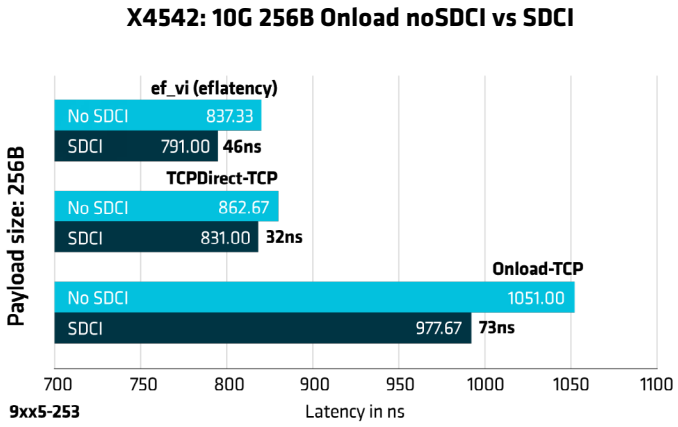
X4542: Onload-UDP: noSDCI vs SDCI



Onload-UDP: 76ns (7.4%) average reduction in latency for 0 to 648B

With Onload user networking stack, the UDP data is delivered to the sfnt-pingpong benchmarking tool. The Onload network stack inspects the packet header and copies data to the application. With SDCI enabled, sfnt-pingpong UDP processing observed 76ns (7.4%) average reduced latency compared to data placed in host memory.

256B Latency improvement with SDCI



256B payload size

- **Eflatency with SDCI:** 46ns (5.5%) average lower latency
- **TCPDirect-TCP with SDCI:** 31.67ns 3.6%) average lower latency
- **Onload-TCP with SDCI:** 73.33ns (7 %) average lower latency

SDCI REQUIREMENTS

AMD provided Fintech solutions with SDCI:

- AMD 5th Gen EPYC CPUs supporting SDCI
- AMD Solarflare Network adapters supporting TPH
- Linux® driver for the NIC that supports SDCI
- AMD 5th Gen EPYC BIOS that enables SDCI

AMD FINTECH LATENCY REDUCTION – A NOVELL APPROACH

Achieving a low latency requires a comprehensive system-level approach, starting from bare metal hardware, system configuration, network optimization and finally pinning the application to the desired CPU core. The following documents provide overall guidance on various resources available for customers when planning for low latency with AMD EPYC 9005 CPUs and AMD Solarflare Ethernet Adapters.

AMD EPYC™ 9005 BIOS & Workload Tuning guide

Low latency tuning for AMD EPYC powered servers

AMD Solarflare Ethernet Adapters

OpenOnload resources

AMD EPYC 9005 Low latency config

PRODUCT CONTACTS

The X2, X3 and X4 series adapters can be purchased through AMD authorized distributors and channel partners.
Please contact **AMD sales** for more information.

Servers designed with 5th generation AMD EPYC processors are available through OEMs and ODMs.
Please contact **AMD sales** for more information.

END NOTES

9xx5-253: SDCI improvements: AMD Internal Testing as of 9/15/2025. Workload config: Onload benchmarking tool measuring Latency for payload ranging from 0B to 648B in ns. System Config: 2 Supermicro systems consist of 1 AMD EPYC 9575F (64C, 12x 32GB DDR5-6400, Max speed of 5GHz, Ubuntu 25.04 with kernel 6.14-23, Custom BIOS version 5.35 (pre-release), OS configured with AMDLL profile and BIOS configured for low latency) with AMD Solarflare X4542 (bundle 1.0.3.2 and bootloader 1.0.17.0). Release Candidate version of Onload is used (https://github.com/Xilinx-CNS/onload/commit/0395821bb0685a3019018b3a9b45c4f30915146f). Driver version 6.6.1.1005 AMD Low Latency Tuning guide available at https://www.amd.com/content/dam/amd/en/documents/epyc-technical-docs/white-papers/58649_amd-epyc-tg... 256B latency comparison with and without SDCI: eflatency: 46ns (5.53%) lower latency with SDCI; TCPDirect-TCP 32ns (3.6%) lower latency with SDCI; Onload TCP 73ns (6.98%) lower latency with SDCI. Onload UDP 80ns (8.24%) lower latency with SDCI ; 0 to 648B latency comparison with and without SDCI: eflatency: 55ns (6.1%) lower latency with SDCI; TCPDirect-TCP 44ns (4.66%) lower latency with SDCI; Onload TCP 67ns (6.09%) lower latency with SDCI; Onload UDP 76ns (7.4%) lower latency with SDCI;

DISCLAIMERS

COPYRIGHT NOTICE