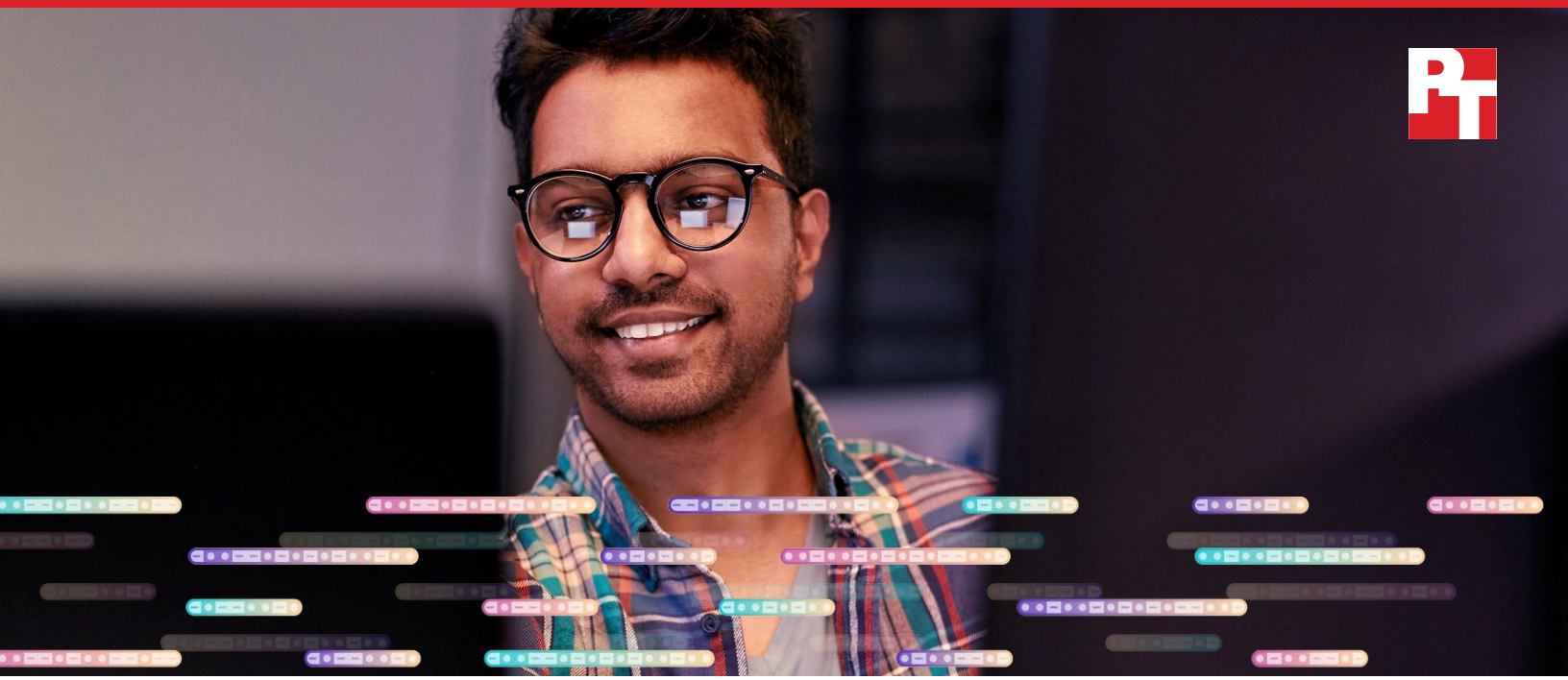


Achieve high throughput: A case study using a Pensando Distributed Services Card with P4 programmable software-defined networking pipeline

Comparing the Pensando DSC-200 and the NVIDIA Mellanox ConnectX-6 Dx with iperf3

Up to 13X the throughput in Gbps and packet rate*
with connection tracking on iperf3 tests

Up to 64% lower latency*
with connection tracking on sockperf tests

Organizations relying on cloud technologies face increasing demands for performance and scalability. To meet these demands, both processors and networking devices have evolved. With the goal of delivering networking, security, and storage in standard servers with extremely high levels of performance and flexibility, Pensando has introduced the P4 programmable architecture and Distributed Services Card (DSC), a software-defined networking (SDN) solution.

To quantify the potential advantages of the Pensando P4 programmable architecture for cloud service providers and organizations using private cloud solutions, we conducted a series of performance tests in two SDN environments: one based on the Pensando DSC-200 and the other based on the NVIDIA® Mellanox® ConnectX®-6 Dx SmartNIC (CX-6 Dx). We chose these two devices for this case study because they are they are the latest architectures available from each vendor. For example, the CX-6 Dx is the packet engine for NVIDIA's latest DPU, the Bluefield-2, and is the offload for the Arm subsystem that provides additional SDN and security performance.[1] Rather than using specialized packet generators, our testing used Linux user-level tools to generate and characterize network traffic for the servers' connection. This is a much closer approximation of the performance a company would experience in a data center or cloud environment.

In our testing, the Pensando DSC-200 environment outperformed the CX-6 Dx one by achieving greater throughput—up to 13 times as much. It also delivered latency up to 64 percent lower. These findings make the Pensando DSC-200 a compelling choice for consumers and providers of cloud services.

*Pensando DSC environment vs. NVIDIA Mellanox CX-6 Dx environment

Meeting growing expectations from cloud requires innovation

As organizations across many industries shift all or part of their compute infrastructure from the data center to the cloud, they seek solutions that scale well and deliver strong performance and low latency. Networking is an important factor in achieving these goals. Cloud providers use virtualized instances running on multiple servers. They must enforce isolation between instances belonging to different businesses, and move data both among instances running in different locations and out to users everywhere.

The DSC-200 is Pensando's 2nd generation Programmable Distributed Services card base on the Elba ASIC.[2] It offers two QSFP28 network ports that support both PAM4 and NRZ and can be deployed at 2x100GE or 2x200GE or break out to lower port speeds. To compare the performance impact of the DSC-200 and another networking device—the NVIDIA Mellanox ConnectX-6 Dx—we conducted a test scenario that involved communication between end host devices running both products at 100G. Below, we draw on publicly available material to describe how the Pensando DSC-200 works

Pensando DSC-200 block diagram

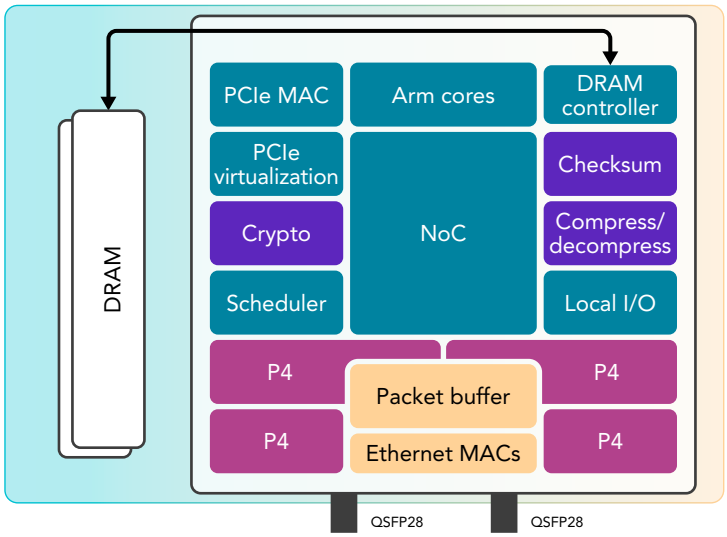


Figure 1: Pensando DSC-200 block diagram. Source: Principled Technologies, based on Figure 1 in Michael Galles and Francis Matus, "Pensando Distributed Services Architecture," IEEE Micro, 2021.[3]

Networking architecture involves two planes: the control plane, which determines how data moves within the network, and the data plane, on which data actually flows. Figure 1 shows the components of the Pensando DSC-200, which include P4 programmable application-specific processors. These carry out packet processing for the data plane. The Arm cores handle exception packets requiring particularly sophisticated data plane functions, and execute the control plane functions.

Cloud providers can program the P4-programmable data plane in the Distributed Services Card. Figure 2 shows the P4-programmable data pipeline, which network engineers can leverage to customize each layer of their infrastructure stack.

Note that the Pensando Distributed Services Platform supports a number of features, such as network security and storage acceleration.[4] We used the DSC as an inline networking switch with these product-specific capabilities: Advanced Observability and Advanced Networking.

Programmability of Pensando DSC

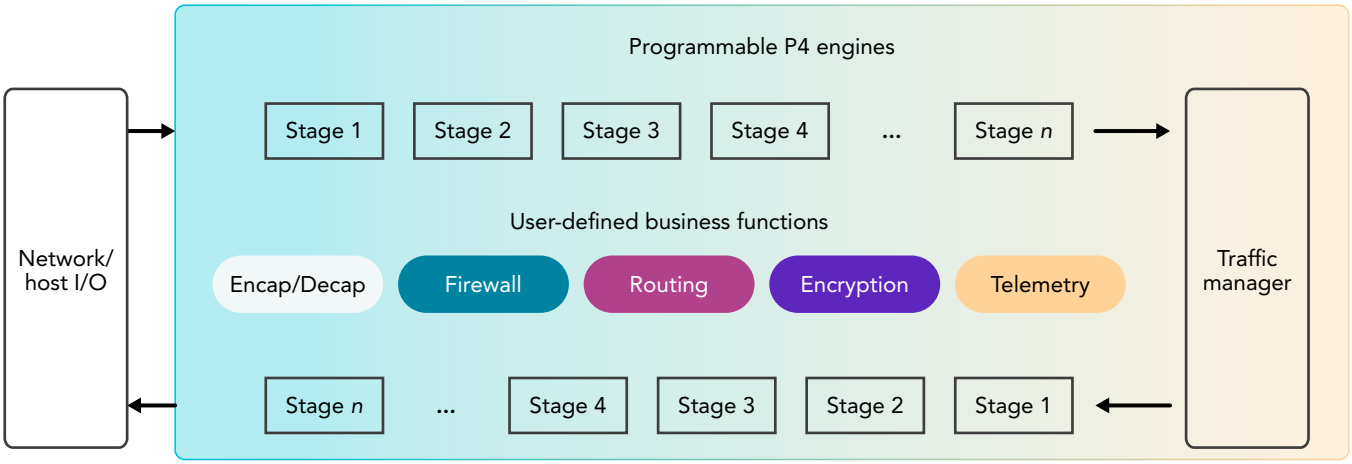


Figure 2: Programmability of the P4-programmable pipelines in the Pensando Distributed Services Card. Source: Principled Technologies, based on "Solution Brief: Distributed Services for Cloud Providers."[5]

About the Pensando Distributed Services Platform

The Pensando Distributed Services Platform supports containerized, virtualized, and bare-metal workloads in a cloud environment. The foundation of the platform is the Distributed Services Card, which Pensando describes as "a custom designed domain-specific programmable processor, providing highly optimized hardware for packet processing and offering a broad suite of software-defined networking, security, telemetry and storage services."[6]

According to Pensando, the DSC offers "high-performance, low-latency, low-jitter, and the highest scalability targeted for the largest cloud providers"[7] and its value lies in "not only the comprehensive number of services offered, but also in the ability to chain the services together in a programmable sequence, without loss of performance."[8]

Pensando architects present their platform as a best-of-both-worlds option, offering the versatility and programmability of an Arm-centric model along with the speed and power efficiency typical of ASICs.[9]

Learn more at https://pensando.io/platform/.

How we approached testing

In both public and private clouds, IT organizations often control communication by performing various match/actions on packets traveling between host devices. In our study, we looked at how well two networking environments— both using a VXLAN SDN overlay configuration—could move data chunks of varying sizes between applications running on two servers.

We compared the performance of the Pensando DSC-200-based environment using a P4 programmable SDN pipeline and the NVIDIA Mellanox ConnectX-6 Dx-based environment using Open vSwitch with hardware offload in a test scenario including the tasks in Table 1.

Table 1: The tasks in our test scenario.

SDN Operation	Benefit for a cloud environment
Parsing VXLAN tagged frames	Provide network isolation within a multi-tenant cloud environment
Matching a specific VXLAN ID	Identify a specific virtual cloud instance or cloud network
Matching the inner source-IP subnet for forwarding the packet	Provide security rules within a virtual cloud instance
Matching the inner destination-IP subnet	Look up the route for the next hop
Rewriting the inner destination-MAC	Route packets to next hop
Connection tracking	Provide layer-four (L4), stateful firewall capabilities
Metering	Prevent a single application or host from monopolizing bandwidth in shared environment

We used the iperf3 tool to measure throughput and the sockperf tool to measure latency. Figure 3 shows the testbed setup we used. For configuration information on the servers we used and a detailed test methodology, see the science behind the report.

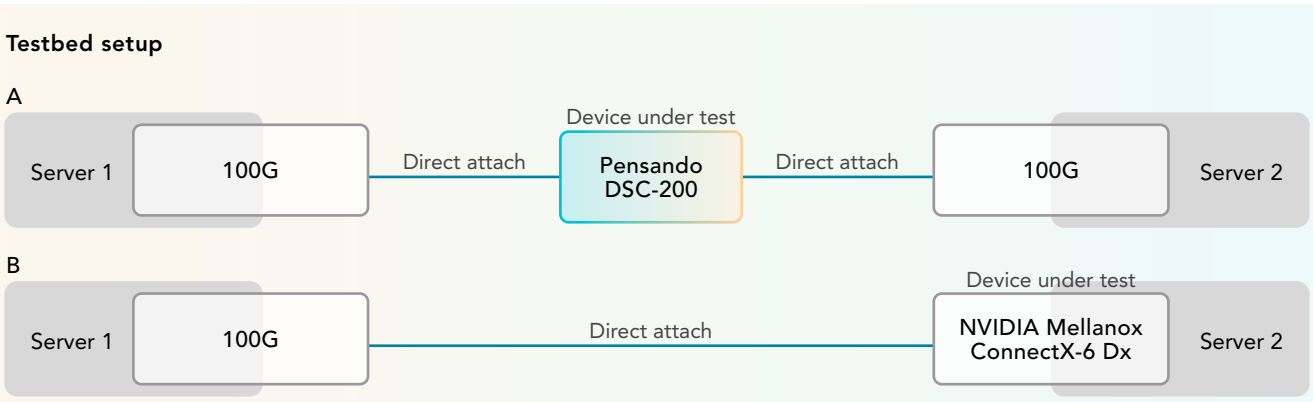


Figure 3: Our testbed setup included two HPE ProLiant DL380 servers with AMD EPYC™ 7320 processors, 256 GB of memory and 16 PCIe® Gen4 risers. Source: Principled Technologies.

Packet sizes

For some use cases, the size of the data chunk plays a critical role in ordinary performance. For example, block-level storage often transfers data in large bulk groups. When the server's kernel and/or application breaks these groups down into a packet size that the network supports, the entire bulk ends up being transmitted at a lower packet rate, and high throughput rates become possible. On the other hand, some applications send only small messages between servers, and are more likely to achieve fast, low-latency communication using small packet sizes.

NVIDIA provides two methods for accessing hardware offloading on Linux using its MLNX_OFED kernel modules and software. One, OVS-Kernel, uses networking functions in the Linux kernel, and the other, OVS-DPDK, uses the DPDK library for networking functions. Despite our following the NVIDIA documentation,[10] we were unable to effect hardware offloading for our test's OVS rules for tracking connections with either method.[11] The results in this report are from testing using the OVS-Kernel method. Because we could not offload connection tracking to the CX-6 Dx, the network performance results we present for the NVIDIA CX-6 Dx environment might be worse than they would be with hardware-offloaded connection tracking.

For our throughput tests, we controlled the size of the data chunks that would travel through the network. We adjusted the maximum packet (or Ethernet frame) size for the network link by setting the maximum transmission unit (MTU) in each server's NIC that served as the VXLAN target. For the latency test, we adjusted the application's parameters to set its data chunk size and adjusted the MTU to be greater than that size. These two definitions of packet size differ (by the size of the frame's non-IP headers), but are consistent throughout testing. We tested 10 data chunk sizes, with MTUs ranging from 96 to 9,000 bytes.

Connection tracking

In networking, connection tracking is when a device maintains information about connection status in memory tables. This connection tracking feature is a critical security element in public cloud networking.[12,13] It can also add value in the area of metrics.[14] Because it uses memory and processing cycles, connection tracking can impose a performance penalty in terms of throughput and latency. This penalty can be severe unless the SDN developer can offload the tracking to the device's application-specific processors rather than its standard processors. In a worst-case scenario, connection tracking falls to the server's processors.

In our testing of the Pensando environment, we were able to offload connection tracking to the Pensando DSC-200G card. In contrast, the NVIDIA Mellanox ConnectX-6 Dx with our Open vSwitch configuration did not offload connection tracking to the device's hardware. To quantify the performance penalty that a non-offloaded service imposed, we ran our scenario for the CX-6 Dx twice: once with connection tracking and once without.

VXLAN tunnel

We set up a VXLAN tunnel between the two servers. A VXLAN tunnel is a form of network encapsulation that cloud networks use to connect hosts in multiple locations in such a way that all hosts appear to share the same local area network (LAN). VXLAN tunnels can enforce network security by limiting which hosts can communicate through the tunnel and can even overcome limitations of network protocols. For example, they can remove MAC address conflicts for virtual hosts or allow more than 4,096 VLANs in an aggregated network.

What we found

Throughput

To measure the throughput potential of the two environments, we used multiple instances of the iperf3 tool to generate TCP network traffic. We measured both throughout in gigabits per second (Gbps) and packet rate (packets per second) using the operating system's interface to the CX6 counters. We tested 1, 4, 16, and 32 instances. In this section we present our findings for the 4- and 16-instance tests. Complete results are in the science behind the report.

Figures 4 and 5 show the throughput in Gbps and the packet rate, respectively, that the two environments achieved on the 4-instance test. Both charts show the Pensando solution, which provides connection tracking, and the CX-6 Dx environment without connection tracking achieving comparable performance. The performance of the CX-6 Dx environment with the non-offloaded connection tracking service was dramatically lower. The greatest difference between the Pensando DSC and the CX-6 Dx environment with connection tracking was with the 256-byte MTU limit, where the former achieved more than 8 times the throughput of the latter. (Please note that the data we present in this report represent the outputs from the scenarios we tested, and do not reflect the maximums either vendor advertises.)

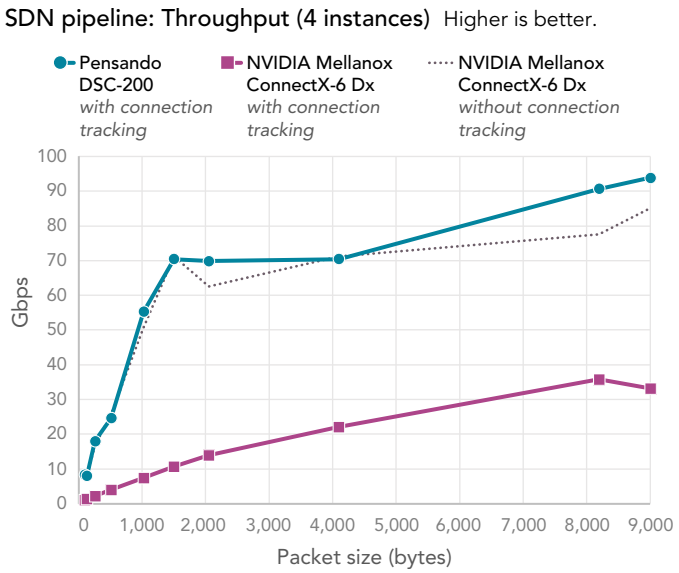


Figure 4: SDN pipeline throughput in Gbps for the two test environments with 4 instances. Higher is better. Source: Principled Technologies.

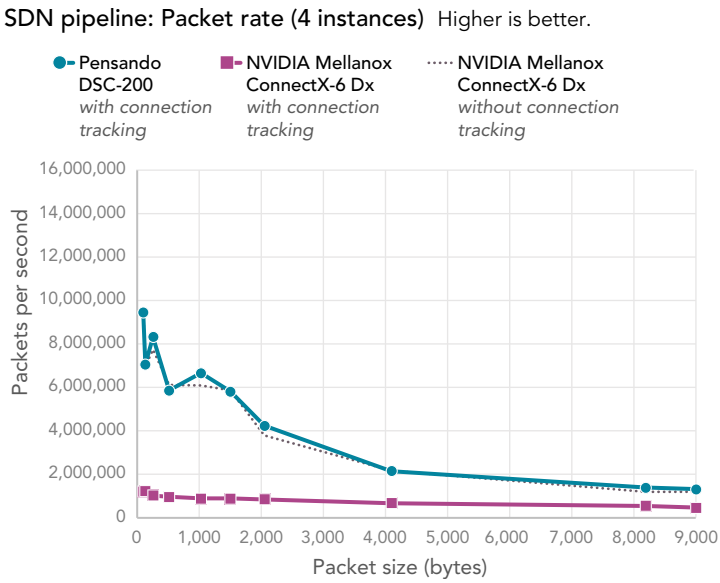
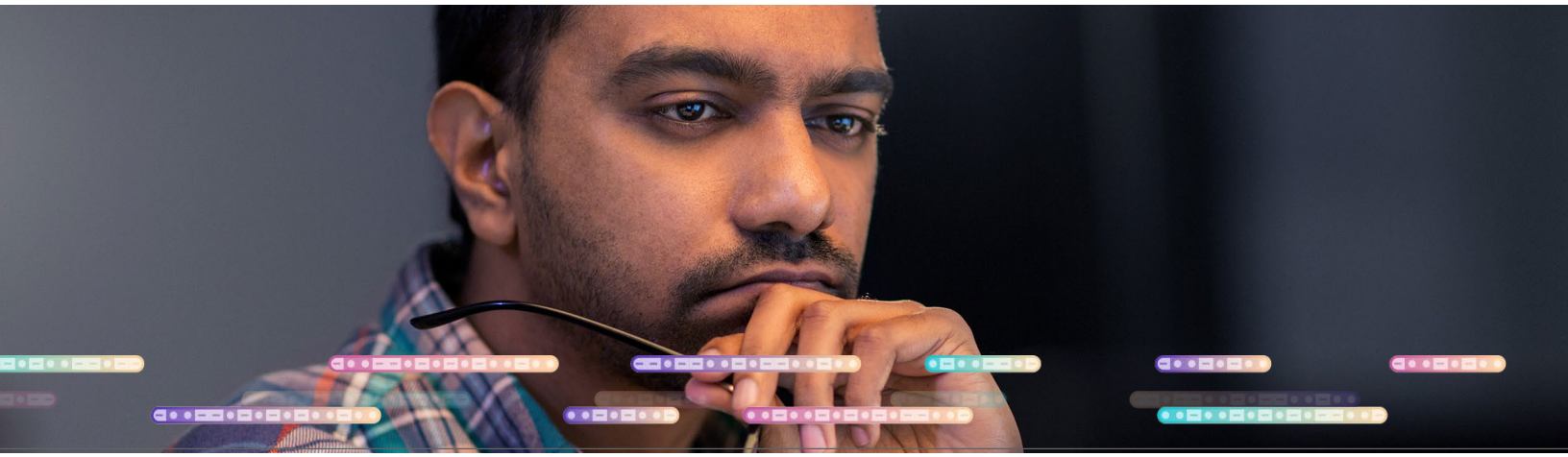


Figure 5: SDN pipeline packet rate for the two test environments with 4 instances. Higher is better. Source: Principled Technologies.



Figures 6 and 7 show the throughput in Gbps and the packet rate, respectively, that the two environments achieved on the 16-instance test. The packet rate results are very similar to those we saw with 4 instances: The Pensando environment with connection tracking achieved comparable packet rates to the CX-6 Dx environment without connection tracking and the packet rates of the CX-6 Dx environment with the non-offloaded connection tracking service were dramatically lower.

The picture differs slightly for throughput, with the Pensando environment achieving a clear advantage over even the CX-6 Dx environment without connection tracking—up to 46 percent greater throughput at 8,192 MTU—and much greater throughput than the CX-6 Dx environment with connection tracking at all MTU limits. The greatest differences were with the 96- and 256-byte MTU limits, where the Pensando DSC environment achieved 13 times the throughput of the CX-6 Dx environment.

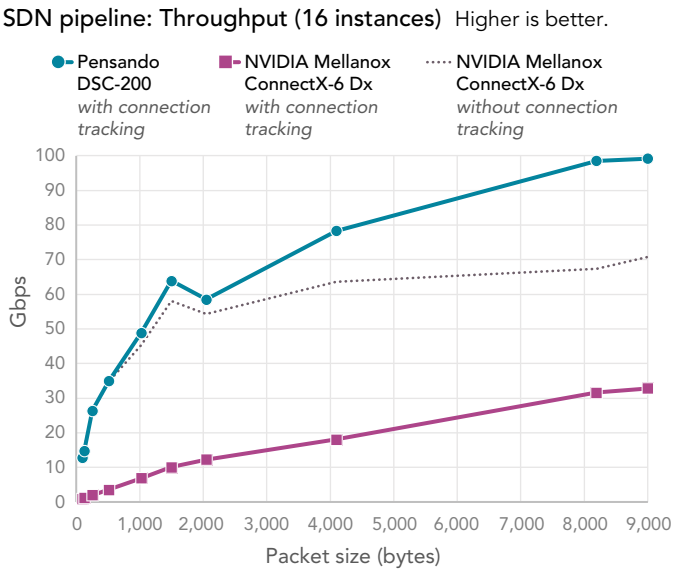


Figure 6: SDN pipeline throughput in Gbps for the two test environments with 16 instances. Higher is better. Source: Principled Technologies.

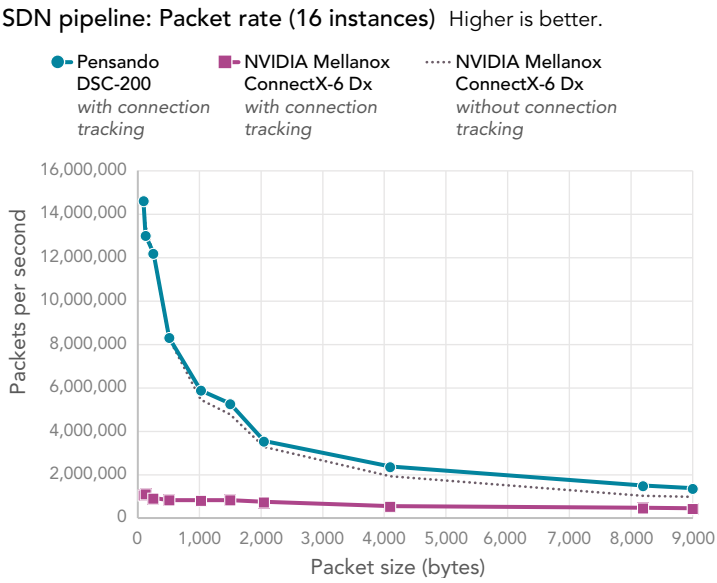


Figure 7: SDN pipeline packet rate for the two test environments with 16 instances. Higher is better. Source: Principled Technologies.

About the test tool: iperf3

iperf3 is a tool that lets users measure the maximum achievable throughput on IP networks. According to its documentation on the iperf3 website, "It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6). For each test it reports the bandwidth, loss, and other parameters."[15]

Latency for a single instance

To measure the latency the two environments delivered while performing this scenario, we used the sockperf tool. As Figure 8 shows, latency with the Pensando DSC environment was the lowest at all packet sizes, with a gradual increase as packet size increased. Without connection tracking, the CX-6 Dx achieved latency only slightly worse than that of the Pensando environment at each packet size. However, with the non-offloaded connection tracking service, the latency of the CX-6 Dx environment increased considerably at all packet sizes, particularly at the largest (9,000 bytes). Here, the latency of the Pensando environment was less than half that of the CX-6 Dx environment with connection tracking—64 percent lower.

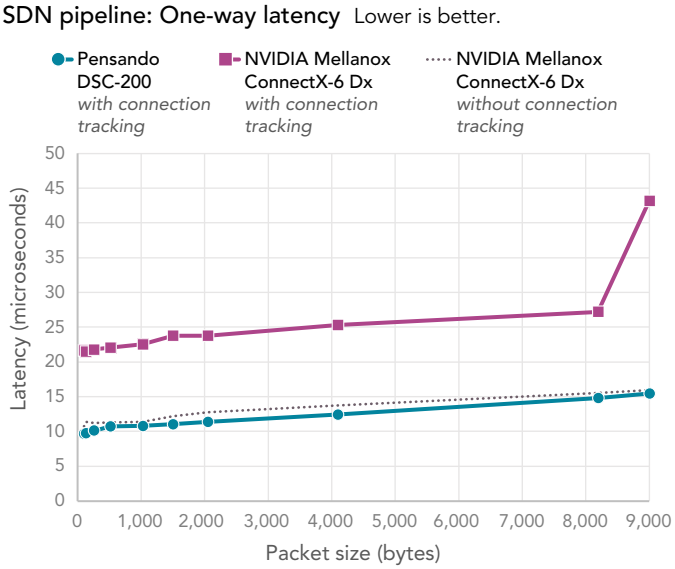


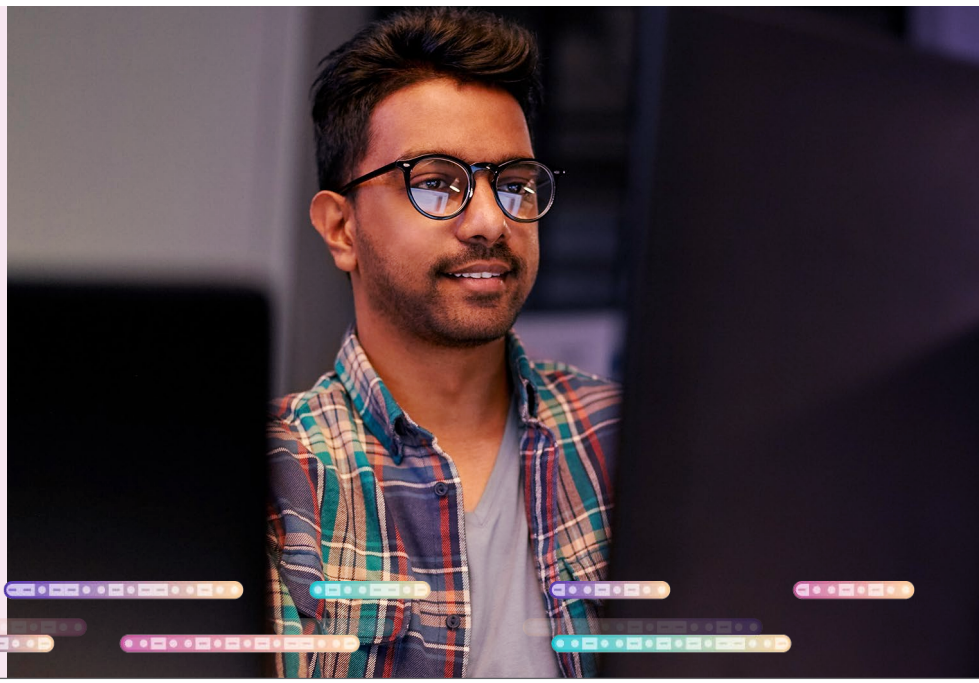
Figure 8: SDN pipeline one-way latency for the two test environments with a single instance. Lower is better. Source: Principled Technologies.

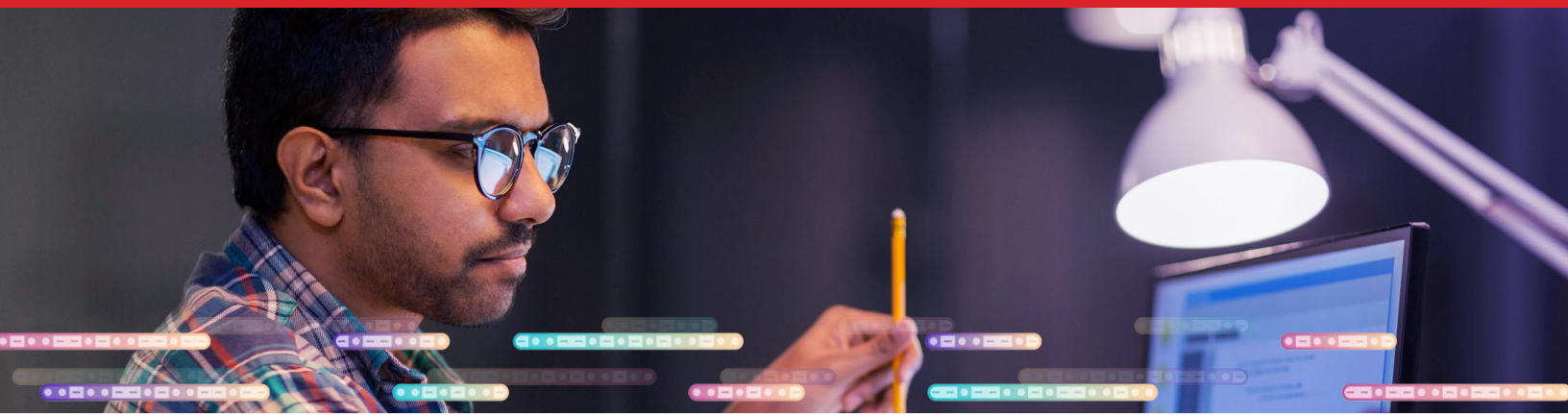
The performance of host-based SDN devices depends on the host's third-party libraries and modules

The DSC works independently of the servers creating the traffic. In contrast, some of the features the NVIDIA CX-6 Dx supports cannot work independently of the servers creating the traffic. The networked application, the Open vSwitch implementation, and/or the Linux kernel must explicitly support the NVIDIA feature. It is possible that a different Open vSwitch implementation, such as one that uses DPDK, could permit more offloading to the CX-6 Dx, which would improve throughput and latency through the VXLAN tunnel. However, to get this performance boost with DPDK, it would be necessary to rewrite the applications generating tunnel traffic to use DPDK. Our testing uses off-the-shelf, non-DPDK applications, and the DSC performed well.

About the test tool: sockperf

sockperf is a network benchmarking utility over socket API. According to its documentation on GitHub, it was "designed for testing performance (latency and throughput) of high-performance systems. It covers most of the socket API calls and options."[16]





Conclusion

Many options are available to cloud providers trying to meet customer need for ever-greater levels of performance and scalability. In our hands-on cloud workload tests in two SDN implementations, the environment based on the Pensando DSC-200 outperformed the environment based on the NVIDIA Mellanox ConnectX-6 Dx SmartNIC in terms of both packet rate throughput and total throughput in Gbps. It also achieved much lower latency. These findings demonstrate that the Pensando architecture holds strong potential for organizations that depend on cloud technologies.

1. BlueField Software v2.0.1.10841 Documentation, Functional Diagram, accessed January 7, 2022, https://docs.nvidia.com/networking/display/BlueFieldSWv20110841/Functional+Diagram.
2. Michael Galles and Francis Matus, "Pensando Distributed Services Architecture," IEEE Micro, 2021, accessed November 23, 2021,https://ieeexplore.ieee.org/document/9352483.
3. Michael Galles and Francis Matus, "Pensando Distributed Services Architecture."
4. "Solution Brief: Distributed Services for Cloud Providers," accessed November 23, 2021, https://pensando.io/wp-content/uploads/2020/03/Pensando-Distributed-Services-for-Cloud-Providers.pdf.
5. "Solution Brief: Distributed Services for Cloud Providers."
6. "Solution Brief: Distributed Services for Cloud Providers."
7. "Solution Brief: Distributed Services for Cloud Providers."
8. "Solution Brief: Distributed Services for Cloud Providers."
9. Section III of M. Baldi, D. Crupnicoff, S. Gai, "Programmable Dataplane Architecture for Distributed Services at the Network Edge," 7th IEEE International Conference on Software Defined Systems (SDS2020), Paris, France. July 2020, doi: 10.1109/SDS49854.2020.9143973.
10. We used MLNX_OFED_LINUX-5.4-1.0.3.0-ubuntu18.04-x86_64 for both methods, and dpdk-20.11.3 for OVS-DPDK.
11. "NVIDIA MLNX_OFED Documentation Rev 5.4-1.0.3.0," published July 5, 2021, accessed on September 30, 2021, https://docs.nvidia.com/networking/display/MLNXENv541030/NVIDIA+MLNX_EN+Documentation+Rev+5.4-1.0.3.0.
12. Google Cloud Platform, Monitoring connections, accessed November 24, 2021, https://cloud.google.com/network-connectivity/docs/interconnect/how-to/monitoring.
13. AWS, Security group connection tracking, accessed November 24, 2021, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-connection-tracking.html.
14. NVIDIA, Accelerating Connection Tracking to Turbo-Charge Stateful Security, accessed November 24, 2021, https://developer.nvidia.com/blog/accelerating-connection-tracking-to-turbo-charge-stateful-security/.
15. "iPerf - The ultimate speed test tool for TCP, UDP and SCTP," accessed November 23, 2021, https://iperf.fr.
16. GitHub repository for sockperf, accessed November 23, 2021, https://github.com/Mellanox/sockperf.

Read the science behind this report at https://facts.pt/6IrEIJu ►

Principled Technologies®

Facts matter.®

This project was commissioned by Pensando.