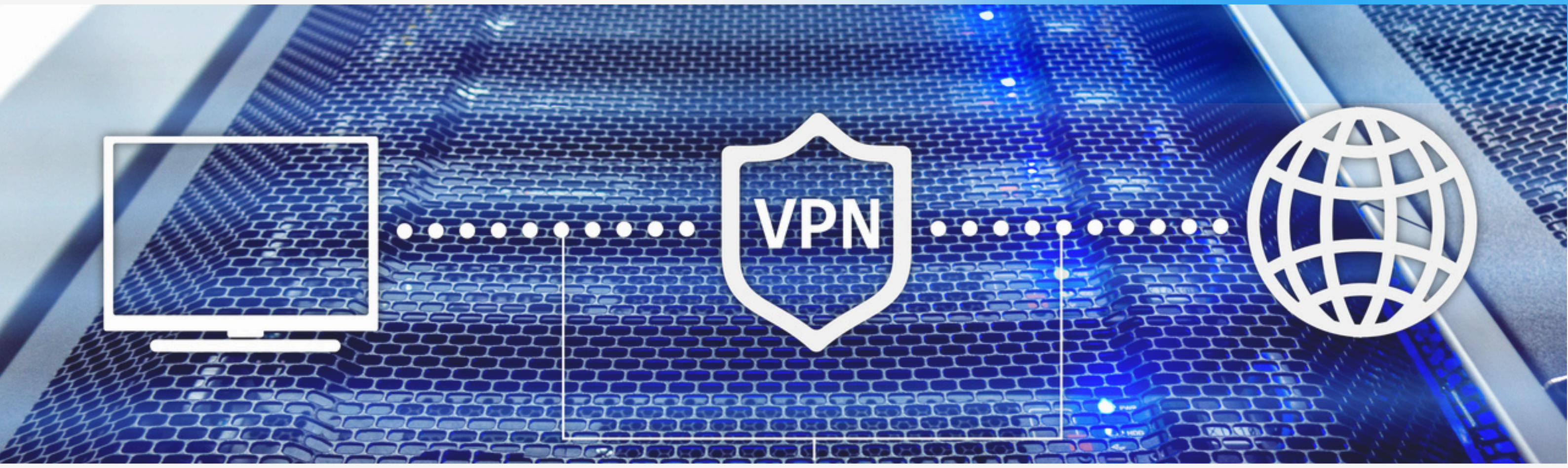
DPDK IPSec Security Gateway

Case-study



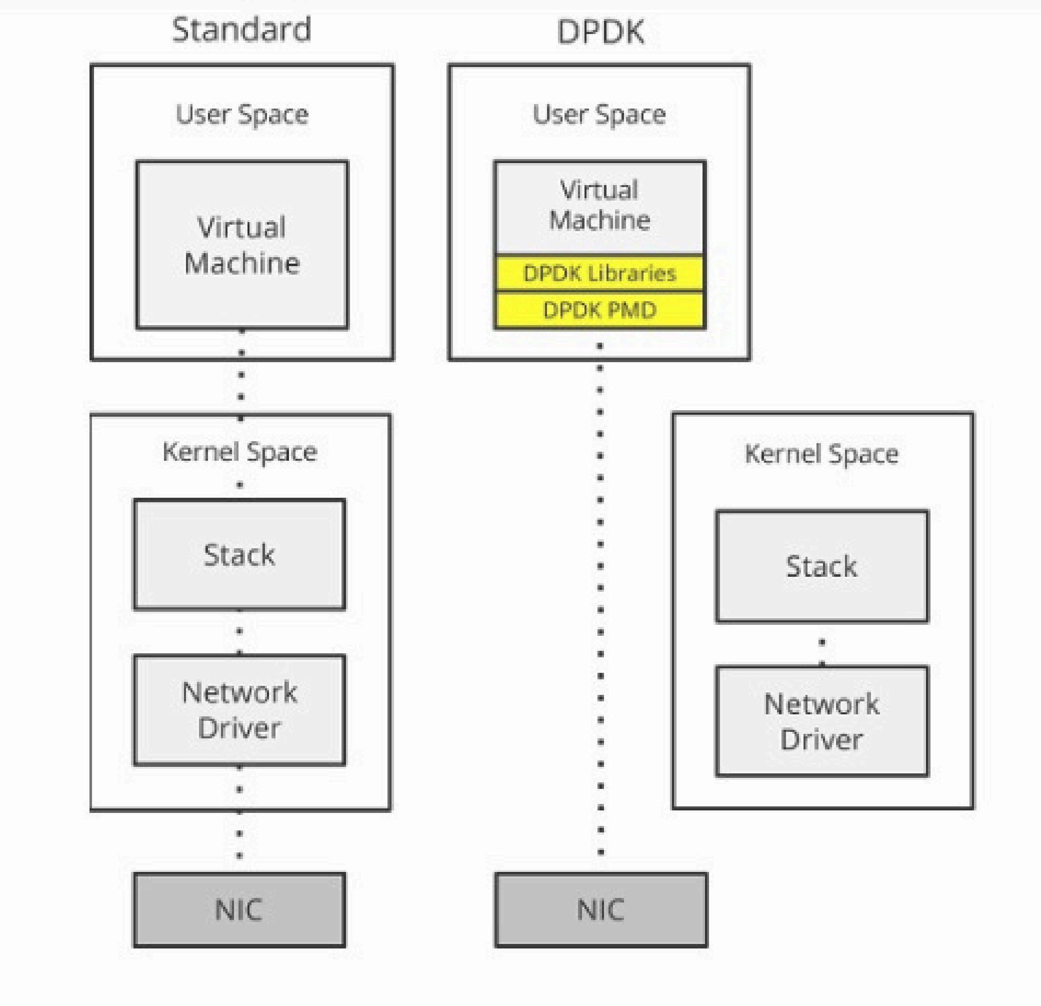
Introduction

Intoday’s digital age,maintaining network security without sacrificing speed is crucial. The **Data Plane Development Kit (DPDK)** offers a solution by enabling faster packet processing through the bypassing of traditional kernel-based methods. This capability is especially valuable in scenarios requiring high- speed data handling, such as firewalls, load balancers, and IPSec gateways. By leveraging DPDK, organizations can achieve enhanced performance and efficiency in their security-focused network applications.

Problem Statement

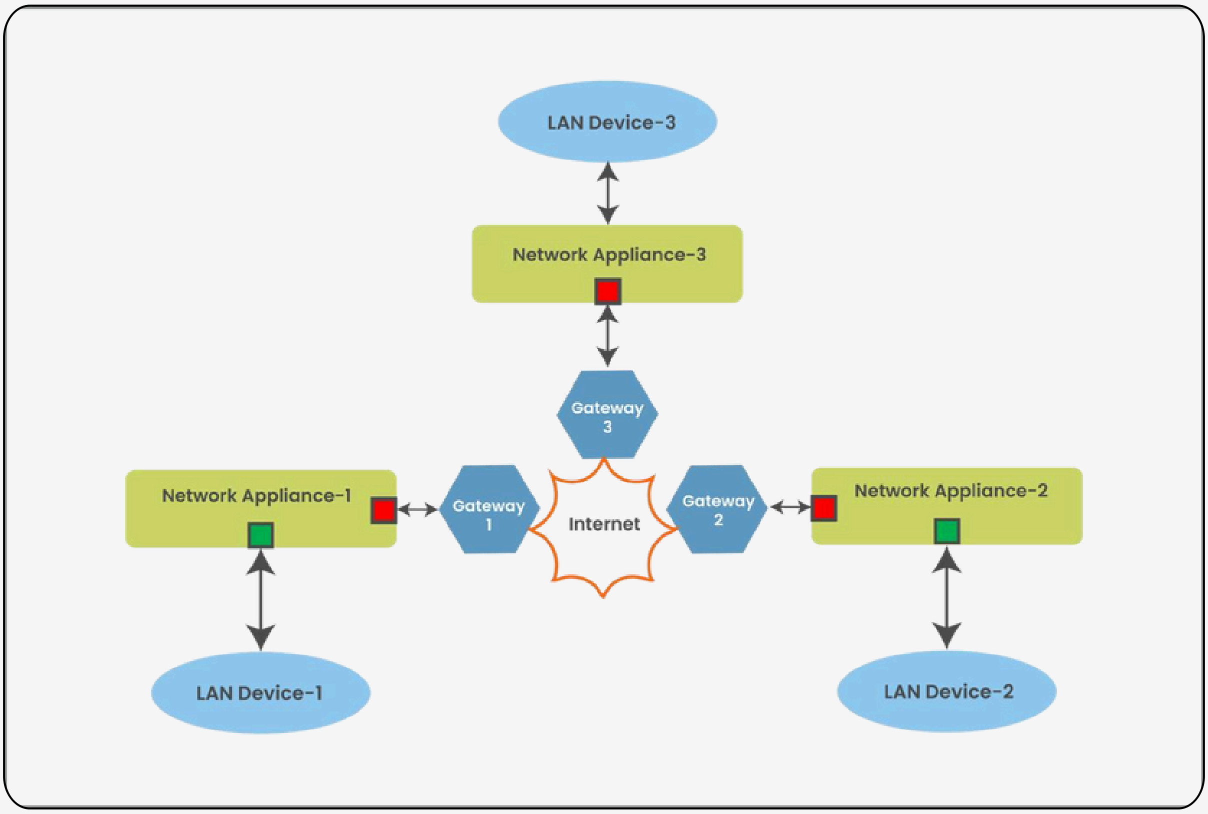
Ourclient, a key player inthe networking industry, was facing a significant challenge. Their existing IPSec gateway solution just couldn't keep up with the high traffic volume. The traditional packet processing method was too slow, leading to increased latency. They needed a solution that could handle more traf- fic without compromising on speed or security.

DPDK (Data Plane Development Kit) is a high-performance framework under the Linux Foundation, designed for fast packet processing by leveraging various user-space libraries and drivers. Initially developed by Intel for x86 CPUs, DPDK now supports multiple CPU architectures, including IBM POWER and ARM. The framework achieves its performance primarily through Fast-Path (Kernel bypass) and Poll Mode Driver (PMD) techniques. Fast-Path creates a direct path from the NIC to the application within user space, bypassing the kernel to eliminate context switching and associated overheads. The PMD dedicates a CPU core to constantly poll the NIC for new packets, avoiding the delays of interrupt-driven processing. Additionally, DPDK optimizes performance with Buffer Managers for efficient network buffer handling, non-uniform memory access (NUMA) awareness to minimize costly memory operations, and Huge-pages to improve physical-to-virtual memory mapping efficiency. Overall comparison of the work-flow between 'Standard' and 'DPDK data-path' can be depicted as below:



The objective of this project was to enhance throughput by offloading IPSec functionality from the kernel to a DPDK-based application. In this setup, the ports connected to tunnel endpoints are referred to as unprotected ports, while those connected to LAN devices are designated as protected ports. These ports are unbound from their default kernel drivers and re-bound to DPDK-compatible drivers. Once bound, traffic bypasses the kernel, routing directly to the DPDK application. However, control plane packets still require processing by the kernel, which is facilitated through Kernel Network Interface (KNI) modules integrated into the DPDK IPSec-gateway application. Virtual Ethernet (vEth) interfaces are established for both protected and unprotected ports. Packet classification within the application ensures that only the necessary packets are forwarded to the kernel, while others are directed to their intended destinations. Outgoing packets may be tunneled or non-tunneled, depending on the rules defined within the application.

The below is the typical topology that was used for the implementation:



🟥 Unprotected/WAN Port + KNI LAN

🟩 Protected/LAN Port + KNI

Solution

We introduced DPDK into their system to offload the heavy lifting from the kernel, allowing for quicker packet processing and higher throughput.

Here's what we did:

- Got the System Ready: We started by setting up DPDK on all the necessary nodes. This included configuring the AES-NI Multi-Buffer Crypto Poll Mode Driver to speed up cryptographic operations and integrating Intel's Quick Assist Technology (QAT) to enhance secure transaction processing.
- Reconfigured the Network Interfaces: Next, we unbound the network interfaces from their default kernel drivers and re-bound them to DPDK-compatible ones. This change allowed DPDK to take over the packet processing directly from the network interface cards (NICs).
- Set Up the Application: We modified the existing IPSec gateway application to include a Kernel NIC Interface (KNI) module. This tweak allowed control plane packets to still be processed by the kernel, while the data plane packets were handled directly by DPDK. We also set up the application to classify and forward packets according to specific security rules.
- Tested and Fine-Tuned: Finally, we put the new setup through rigorous testing using different traffic patterns. The results were impressive; the system could handle significantly more encrypted sessions without any drop in performance.

Conclusion

By integrating DPDK into the IPSec gateway, the overall solution not just had an increased throughput but also provided a scalable way to manage high volumes of encrypted traffic efficiently. This case study demonstrates the power of DPDK in boosting the performance of security-focused network applications.