XXI Escola Regional de Redes de Computadores (ERRC). DOI: 10.5753/errc.2024.4561

Artigo Completo/Full Paper

Optimizing Network Performance: Benchmarking of NVIDIA Bluefield-2 Offloading Capabilities

Arthur Vinícius Camargo • avccamargo@inf.ufrgs.br
Instituto de Informática • Universidade Federal do Rio Grande do Sul (INF/UFRGS)
Leandro Bertholdo • leandro.bertholdo@ufrgs.br
Instituto de Informática • Universidade Federal do Rio Grande do Sul (INF/UFRGS)
Lisandro Granville • granville@inf.ufrgs.br
Instituto de Informática • Universidade Federal do Rio Grande do Sul (INF/UFRGS)

ABSTRACT. The growing demand for higher Internet speeds is placing significant strain on general-purpose processors, particularly in handling network processing tasks. This trend has driven the shift toward the offloading of such tasks to specialized hardware. SmartNICs alleviate CPU load by efficiently managing network processing while providing programmability and customization. However, mastering the deployment and utilization of SmartNICs poses a significant challenge due to the steep learning curve involved. In this study, we test and evaluate the performance of NVIDIA's Bluefield-2 SmartNIC under a range of conditions, with the objective of achieving 100Gbps throughput on a single server. Our results demonstrate that specific libraries can enhance the processing of common traffic types, such as TCP and UDP, by up to tenfold, thereby enabling the system to reach 100Gbps line rate.

PALAVRAS-CHAVE: SmartNICs • Bluefield-2 • Deslocamento de tarefas de rede • Otimização de desempenho da rede
KEYWORDS: SmartNICs • Bluefield-2 • Network offloading • Network performance optimization

1 Introduction

As Moore's Law [1] and Dennard scaling [2] have reached their limits, addressing the challenges of increasing bandwidth and communication speed solely through enhanced processor capacity is no longer viable [3, 4]. The physical limitations of transistors and the stagnation in processing power growth have introduced significant obstacles, particularly in packet processing and managing network traffic flows. Today, the majority of data processed by computers originates from the Internet, driven by the proliferation of cloud services, content delivery networks (CDNs), and web-based solutions. Consequently, a large portion of CPU tasks is now devoted to infrastructure and networking. Furthermore, software-based solutions on Internet servers are increasingly struggling with throughput and latency, falling behind hardware-optimized counterparts. Companies providing fast and reliable online services face time constraints and must rely on proprietary, often opaque solutions to process packets efficiently. The outlook for the coming years remains uncertain, as port speeds continue to rise (e.g., innovations from Cisco and Huawei), while processor speeds show no signs of overcoming their current plateau.

To address these challenges, domain-specific processors have emerged, designed specifically for infrastructure tasks and routines. The primary concept is to offload tasks that CPUs do not excel at, similar to the successful integration of graphical processing units (GPUs) for specialized tasks. This approach has led to the evolution of an already essential piece of hardware in modern computers: the Network Interface Card (NIC). Originally responsible for handling layer 1 and layer 2 tasks, NICs have gradually offloaded more kernel responsibilities related to packet processing. The new generation NICs have been enhanced, incorporating hardware accelerators for executing common network tasks such as implementing TCP Offload Engine (TOE), checksum calculations, and other higher-level protocols within the network stack.

Recently, those cards evolved once more to contain a general purpose processing unit now called Data Processing Unit (DPU). That innovation brought the possibility of the NIC, now called SmartNIC, to manage their own accelerators and to spare the host CPU.

SmartNICs, or Smart Network Interface Cards, are typically constructed using FPGA (Field-Programmable Gate Array), ASIC (Application-Specific Integrated Circuit), or SoC (System-on-a-Chip) architectures. FPGA-based models offer greater programmability but tend to be more complex and costly, whereas ASIC models are simpler and more cost-effective but generally less efficient. SoC-based SmartNICs incorporate general-purpose processors,

Todos os autores são membros da SBC.

allowing them to manage infrastructure tasks [5].

In this paper, we analyze the DPU NVIDIA's Bluefield-2 SoC SmartNIC [6] and evaluate its ability to process various types of packets. Our contribution focuses on testing the Bluefield-2's offloading capabilities, programmability, and performance using a packet generator at 100Gbps line-rate speed. While the manufacturer provides some benchmarks, we offer complementary measurements by utilizing different packet types and protocols. We compare throughput with and without its features, and discuss our impressions of its programmability. Our findings reveal that when network processing occurs in the kernel, there is a drastic packet loss of 99.8%, indicating that most traffic is dropped.

This paper is organized as follows: In section 2, we review related work on SmartNICs, with a focus on performance-based analyses. In section 3, we outline our methodology and measurement environment for utilizing and programming the Bluefield-2. In section 5, we present our results on measuring its network processing capabilities and compare these results to other established solutions. In section 6, we discuss the outcomes of our experiments. Finally, in section 7, we provide our conclusions and final remarks.

2 Related works

The evolution of SmartNICS has been drive by the increasing demands of modern networks, being brought by the exponentially growth of its speed. Traditional NICs were initially designed to handle simple packet forwarding, relying on the OS kernel for most networking tasks. As the traffic surged, this model became inefficient. This limitation caused the development of offloaded NICs, that came with some of the tasks that the CPU struggled to deliver efficiently implemented in hardware, such as TCP offloading, checksum offloading and direct memory access (DMA). The need for even more advanced processing lead to the creating of SmartNICs. These specialized cards not only handle network tasks but also offload more complex functions like encryption, firewalling, and storage, providing a great relief to the CPU.

There are multiple types of SmartNICs that handle network tasks using various architectures and techniques. Kfoury *et al.* [7] provides a broad background of each technology and describe its architectures, development environments and advantages.

In Figure 1, we can see how the traditional, offloaded and Smart NICs differ in terms of components and complexity. While the traditional NIC were responsible for only implementing the physical and data layer. As there were a need to spare the CPU of some of the network duties, some accelerators begun to appear at the card, such as the TCP offload engine, given rise to what is called an Offload NIC. The increase of network connections and the need of more processing programability, gave space to the creation of the SmartNIC, which now contains their own CPU cores to manage the domain-specific accelerators and also implements a traffic manager that can provide Quality of service (QoS) functionalities and steers traffic to the other execution engines.

Kfoury *et al.* [7] also stated that offloading infrastructure tasks, typically handled by data centers, to SmartNICs can free up as much as 30% of processing capacity for user applications. Additionally, Cloud providers can be even more benefited from that offloading as their product involve renting CPU processing time [8].

Liu *et al.* [9] is the first who conduct tests using the BlueField-2, including saturating the line-rate of 100Gbps and using *stress-ng* [10] tool to stress test multiple components, such as the embedded CPU, memory as well as the kernel network stack. They also make some burst tests using *pktgen* [11], a kernel space tool that is capable of quickly inserting UDP packet in the IP network stack. Their findings shows that "offloading functions to the SmartNIC requires careful consideration about the resource usage and operations". Our approach utilizes more packet diversity and does not utilize that type of approach of generating data, which can stress other components and verify if there is some changes while utilizing different types of packets.

The emergence of programmable networks created opportunities for independent protocol customization, leading to the development of the P4 language[12]—a standard for programming network devices without requiring new hardware. Because of that, there is an effort for implementing that type of technology inside of the SmartNICs. The problem being that its optimizations rely on low-level program tuning, where P4 abstractions operate at one level above.

Xing *et al.* [13] implements an automated performance optimization framework for P4 programmable SmartNICs while evaluating its execution against not optimized P4 programs. However, their research focus on benchmark the Bluefield-2 only while utilizing the DPDK flow pipeline, which does not consider the more low-level hardware implementations that the SmartNIC can offer, such as compression and deep packet

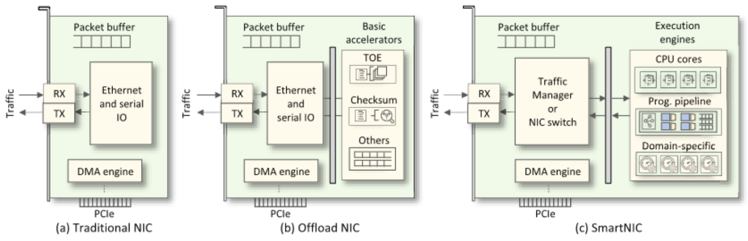

Figure 1. Evolution of SmartNICs, reproduced from Kfoury *et al.* [7]

inspection.

Luizelli *et al.* [14] provides a foundational insight into SmartNICs and a hands-on tutorial for the BlueField-2 giving instruction and examples of how to use NVIDIA's DOCA. Additionally, they present some concerns on the standardization of SmartNICS, which can hinder a wider adoption of the technology. There is also a problem about how to understand which section is supposed to be offloaded, and with which acceleration hardware, calling for a need of better development tools that supports those newer technology.

There are an abundant of works that offloads plenty of end host functions onto SmartNICS [15–23], which includes cloud functions, load-balancing, stateful TCP, high performance distributed transactions and others. Those works talks about how they utilized the SmartNICs to solve their specific problems and how well they performed related to more usual and already established solutions. Although those works wrap around a similar topic as covered in this paper, our objective is to create a way to measure packet processing and to compare it with how normal processing units does the same type of work.

3 Lab and Tools

In Figure 2, we show the setup of our lab. It was deployed using two BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56 interface cards. Each card includes an 8-core Arm87 processor, integrated BMC, PCIe Gen4 x16, 16GB on-board DDR4, and 1GbE OOB management. The P_0 and P_1 interfaces on each card are cross-connected using multi-mode optical fiber. The host system is powered by an AMD EPYC 7282 16-Core Processor running Ubuntu 22.04. In this setup, both Cards (SNIC-1 and SNIC-2) are connected within the same host.

We needed a software capable of generating traffic with different protocols and sizes, and a software that can create a great throughput in order to stress test if the card can keep up with the line rate. To do that, our

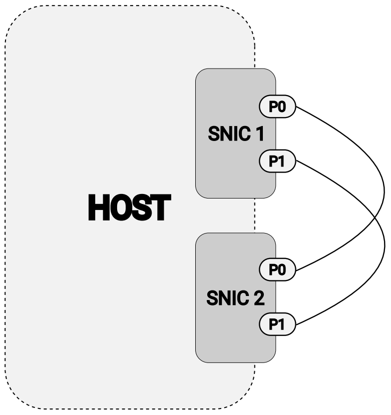

Figure 2. BlueField-2 Setup

solution was to use TRex [24], a realistic traffic generator that permits to emulate traffic from layer 3 up to layer 7 and to fully customize the packets. Additionally, TRex, uses DPDK, that permits it to scale up to 200Gbps using one server. As most of the traffic generators, it comes with a server and a client that needs to be installed into the borders of the measurement, and to establish a send speed that the client will try to send to the server.

In order to see how the main differences of processing the traffic in the CPU against offloading it into the SmartNIC off path, we needed to use the OvSwitch program that comes with the cards, in order to activate or not the Offload, that changes the path which the packets will travel and the devices that will process them.

4 Methodology

In our experiments, we changed the throughput values based on the percentage of the maximum line-rate possible (100Gbps) with 100%, 75%, 50% and 25%. Furthermore, our measurements were conducted for 10 seconds using 8 threads on the host, where the traffic generator server and client was running. To test the performance of the SmartNIC, four distinct packet sizes and types were utilized in the experiments:

- **UDP Small packet:** A very small packet with a

size of 64 bytes was used. This packet size is typical for testing scenarios that require minimal payloads.
- **IMIX:** A more balanced approach was taken with variable packet sizes ranging from 64 to 1500 bytes. IMIX (Internet Mix) represents a more realistic traffic profile by mixing various packet sizes to simulate real-world network conditions.
- **UDP Jumbo Packet:** This larger packet, also referred to as a jumbo frame, had a size of 9216 bytes. Testing with jumbo frames allows the evaluation of SmartNIC's performance when handling larger packets, which are typically used in high-throughput networks.
- **SYN attack:** A packet of size 64 bytes was used to simulate a TCP SYN attack, which targets connection stacks by overwhelming the system with connection requests. This test assesses the robustness of the SmartNIC in protecting against such attacks, and additionally uses a stateful transport protocol.

These tests provide a perspective on how the SmartNIC handles different types of network traffic, from small and large packets to simulated attack traffic.

5 Results

This section discourse about the results of our experiments on the packet processing of the Bluefield-2 SmartNIC. We found how each type of protocol change the offloading aspects of the SmartNIC and how it can affect the performance.

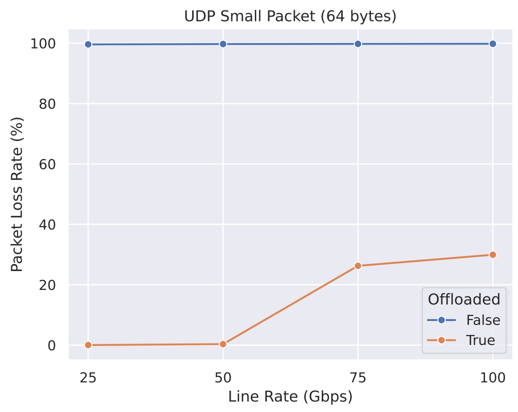

Figure 3. BlueField-2 Offload comparison using UDP small packets

We can observe for the majority of the time, when the offload is disabled, less than 1% of the packets are received by the host, which means that the processing queue is getting full and the processor is not able to handle all the traffic in time, generating a lot of packet loss. Meanwhile, the use of the offloaded version, makes it able to process most of the traffic, more than 50% at all the cases and without the use of the host CPU time.

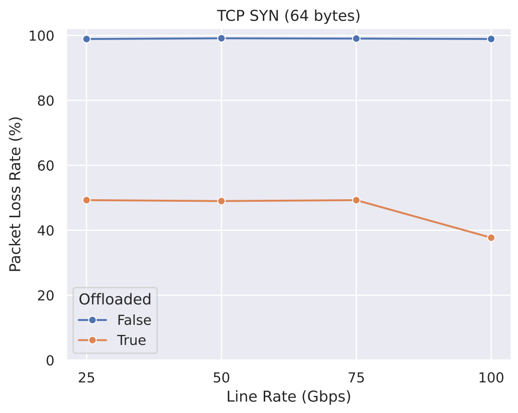

Figure 4. BlueField-2 Offload comparison using SYN packets

When handling very small TCP and UDP packets in figure 3 and 4, the card does shows a increased packet loss, wich 30% for UDP packets and 40% for TCP SYN attacks. That happens because of the overhead that processing a single packet creates. On the other hand, when handling a more optimistic UDP jumbo packet 5, where the size of the packet is way greater, the Bluefield-2 is able to have almost no packet loss when considering offloading.

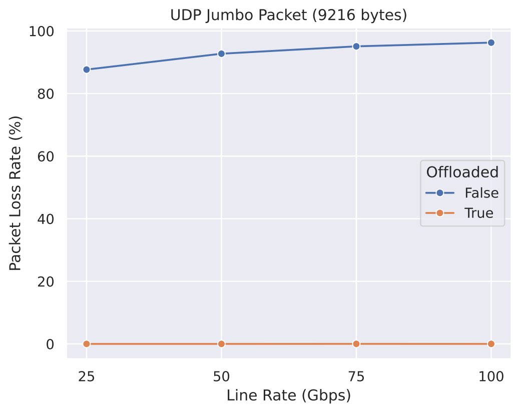

Figure 5. BlueField-2 Offload comparison using UDP jumbo packets

It is clear that offloading is necessary for being able to process network packets at a high speed proposed by the SmartNIC. That happens because handling that data inside the operating system kernel can pose some issues. First of all, most of the time, the path inside of the kernel relies on interruptions to be able to handle

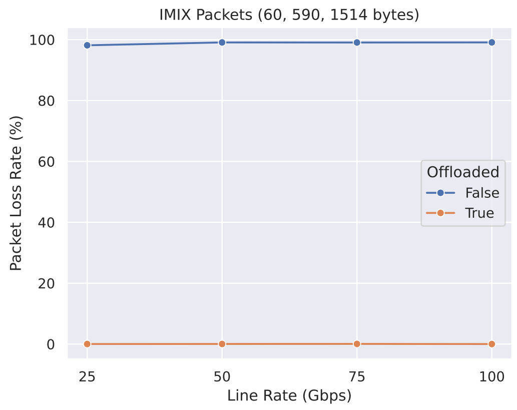

Figure 6. BlueField-2 Offload comparison using IMIX packets

multiple processes and threads. By doing that, a lot of overhead is generated for processing a great number of packets. Second of all, as a general purpose processor is not specifically suited for network processing, it lags behind more accelerated and hardware implemented approaches, as they are custom made for that purpose.

Although there is a need to use the offloading to be able to fully utilize the SmartNIC potential, which means that it does not provide a CPU level of programability at line-rate. There are still some customization that the NVIDIA development kit permits and that were not explored in this article. It includes some C APIs that enable programmers and administrators to implement programs that will use the proper accelerators, like direct memory access, hardware implementations of compression algorithms, and security based approaches that comply with zero trust standards.

6 Discussion and Findings

Our results show that offloading the network packet processing can greatly up to 70% on small packets (64 bytes) and up to 90% on bigger packets (9216 bytes), increase throughput and avoid possible packet losses in the way. On the other hand, not offloading network tasks to BlueField-2 can lead to almost no throughput and reduced scalability. This negatively impacts overall performance, especially in high-traffic, complex network environments.

Liu *et al.* [9] performed experiments aimed at evaluating the available processing headroom on the SmartNIC's processors while the card is fully engaged in transmitting network data at maximum speed. Unlike our approach, they introduced varying delays between packet bursts to simulate a scenario where computations are applied to the network stream. That way, their experiments provide a more optimistic scenario, where they were still able to process 1 GBps not using the offloading capabilities and with a packet size of 128 bytes and a burst of 25 packets. In comparison, our results shows an rate of only 0.042 GBps while not using the offloading. That means that steering the traffic to the ARM processor of the Bluefield-2 is not enough for packet processing.

Unlike traditional NICs, the BlueField-2 offers extensive customization and programmability within its network processing stack. This enables efficient traffic steering to the DPU, routing through multiple accelerators, and even packet inspection—both headers and content. All of this is achieved while offloading intensive processing tasks from the host CPU, optimizing performance and freeing up resources. However, this complexity means that developers often find a steep learning curve, and there is a scarcity of comprehensive training resources and tutorials. That fact is that programming those cards often requires specialized knowledge of networking, their specific (DPU) and it's proprietary SDK **DOCA**, which can be seen as a vendor lock-in, as makes it hard for new users to fully leverage the hardware's capabilities and creates a challenge to migrate to other SmartNICs and adopt new technologies. Moreover, Bluefield-2 emphasizes ready-to-use offloading functions for networking, diving into the hardware at a very granular level isn't as easy and straightforward as with programmable platforms like P4 [13].

To conclude, we can say that the SmartNICs represents a big change on the actual network cards interface, and that it has the potential of being a great inclusion for increasing throughput at the same that that it gives flexibility and programmability. But it is safe to say, that it is not possible to process at line-rate speeds while running a fully custom code without using their offloading capabilities and exploring their accelerators.

7 Conclusion

The reduction in processing power scaling necessitates the development of new hardware solutions to compensate for this limitation. SmartNICs emerge as a promising solution, enabling the fast and programmable processing of infrastructure tasks, while reducing the load in the main host system. Despite some existing challenges in adoption and integration, SmartNICs hold significant potential to become a crucial component in modern networking infrastructure. Their ability to offload tasks and enhance performance in scalable and flexible ways makes them a key technology for addressing future network demands.

Declarations

Availability of data and additional materials

Datasets and code may be made available upon request.

References

1 Moore, G. E. *et al.* Progress in digital integrated electronics. *In*: WASHINGTON, DC. ELECTRON devices meeting. 1975. v. 21, p. 11–13.

2 Dennard, R. H. *et al.* Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of solid-state circuits*, IEEE, v. 9, n. 5, p. 256–268, 1974.

3 Powell, J. R. The Quantum Limit to Moore's Law. *Proceedings of the IEEE*, v. 96, n. 8, p. 1247–1248, 2008. DOI: 10.1109/JPROC.2008.925411.

4 Prati, E. *et al.* From the Quantum Moore's Law toward Silicon Based Universal Quantum Computing. *In*: 2017 IEEE International Conference on Rebooting Computing (ICRC). 2017. P. 1–4. DOI: 10.1109/ICRC.2017.8123662.

5 Huawei. *How do DPUs and Co-Processors for Data Processing Work?* Dec. 2023. https://forum.huawei.com/enterprise/en/how-do-dpus-and-co-processors-for-data-processing-work/thread/741347071123931136-667213860488228864. (Accessed on 09/19/2024).

6 Bhalgat, A. *Choosing the Best SmartNIC | NVIDIA Technical Blog*. Sept. 2021. (Accessed on 09/19/2024). Available from: https://developer.nvidia.com/blog/choosing-the-best-dpu-based-smartnic/.

7 Kfoury, E. F. *et al.* A Comprehensive Survey on SmartNICs: Architectures, Development Models, Applications, and Research Directions. *IEEE Access*, v. 12, p. 107297–107336, 2024. DOI: 10.1109/ACCESS.2024.3437203.

8 Lin, W. *et al.* SuperNIC: An FPGA-Based, Cloud-Oriented SmartNIC. *In*: PROCEEDINGS of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays. 2024. P. 130–141.

9 Liu, J. *et al. Performance Characteristics of the BlueField-2 SmartNIC*. 2021. https://arxiv.org/abs/2105.06619. arXiv: 2105.06619 [cs.NI].

10 King, C. I. *Stress next generation*. June 2015. https://github.com/ColinIanKing/stress-ng. (Accessed on 10/07/2024).

11 community, T. kernel development. *HOWTO for the linux packet generator*. https://www.kernel.org/doc/html/latest/networking/pktgen.html. (Accessed on 10/07/2024).

12 Bosshart, P. *et al.* P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, ACM New York, NY, USA, v. 44, n. 3, p. 87–95, 2014.

13 Xing, J. *et al.* Unleashing SmartNIC Packet Processing Performance in P4. *In*: PROCEEDINGS of the ACM SIGCOMM 2023 Conference. New York, NY, USA: Association for Computing Machinery, 2023. (ACM SIGCOMM '23), p. 1028–1042. ISBN 9798400702365. DOI: 10.1145/3603269.3604882.

14 Luizelli, M. C. *et al.* SmartNICs: The Next Leap in Networking. *Simposio Brasileiro de Redes de Computadores*, 2024.

15 Cui, T. *et al.* Offloading load balancers onto smartnics. *In*: PROCEEDINGS of the 12th ACM SIGOPS Asia-Pacific Workshop on Systems. 2021. P. 56–62.

16 Eran, H. *et al.* {NICA}: An infrastructure for inline acceleration of network applications. *In*: 2019 USENIX Annual Technical Conference (USENIX ATC 19). 2019. P. 345–362.

17 Firestone, D. *et al.* Azure accelerated networking:{SmartNICs} in the public cloud. *In*: 15TH USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). 2018. P. 51–66.

18 Li, J. *et al.* {AlNiCo}:{SmartNIC-accelerated} contention-aware request scheduling for transaction processing. *In*: 2022 USENIX Annual Technical Conference (USENIX ATC 22). 2022. P. 951–966.

19 Liu, M. *et al.* Offloading distributed applications onto smartnics using ipipe. *In*: PROCEEDINGS of the ACM Special Interest Group on Data Communication. 2019. P. 318–333.

20 Min, J. *et al.* Gimbal: enabling multi-tenant storage disaggregation on SmartNIC JBOFs. *In*: PROCEEDINGS of the 2021 ACM SIGCOMM 2021 Conference. 2021. P. 106–122.

21 Moon, Y. *et al.* {AccelTCP}: Accelerating network applications with stateful {TCP} offloading. *In*: 17TH USENIX Symposium on Networked Systems Design and Implementation (NSDI 20). 2020. P. 77–92.

22 Qiu, Y. *et al.* Automated smartnic offloading insights for network functions. *In*: PROCEEDINGS of the ACM SIGOPS 28th Symposium on Operating Systems Principles. 2021. P. 772–787.

23 Schuh, H. N. *et al.* Xenic: SmartNIC-accelerated distributed transactions. *In*: PROCEEDINGS of the ACM SIGOPS 28th Symposium on Operating Systems Principles. 2021. P. 740–755.

24 Team, T. *TRex Realistic Traffic Generator*. 2024. https://trex-tgn.cisco.com/. (Accessed on 09/24/2024).