CPU-to-FPGA network offload engine

Final Year Project Proposal

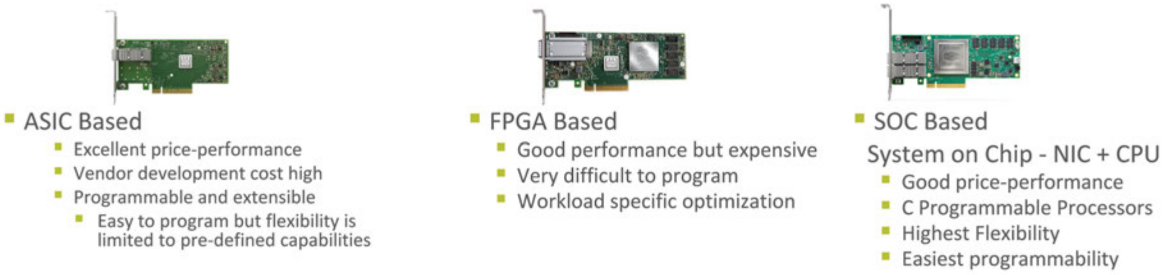
[redacted]

14 Oct, 2020

1 Project Context

Networking applications are becoming increasingly demanding due to the huge increase in the amount of data generated nowadays. More and more, these applications require higher bandwidth and lower latency to archive better and more responsive service. Hence, it is becoming a challenge not to overload the CPU with a network stack running at maximum bandwidth [1]. In other words, the CPU should be enabled to spend more of its resource dealing with actual data rather than running the network stacks.

DPU-based NICs (Data Processing Unit based Network Interface Cards) are designed to offload the traffic processing steps of the network stack from the CPU and the kernel [2]. Compared to simple NICs, DPU-based NIC goes beyond simple connectivity and implements network traffic processing on the board. Namely, network offloading NICs can be divided into 3 types, ASIC-based, FPGA-based and SOC-based, each with their own trade-offs [2]. As shown in Figure 1 below, an ASIC-Based NIC provides the best performance out of all 3, but the development cost is also the most expensive and the extensibility is the worst out of all 3. On the other side of the spectrum, SOC-based NIC requires the least development but the performance is incomparable with the other hardware-based solutions.



- ASIC Based
  - Excellent price-performance
  - Vendor development cost high
  - Programmable and extensible
    - Easy to program but flexibility is limited to pre-defined capabilities
- FPGA Based
  - Good performance but expensive
  - Very difficult to program
  - Workload specific optimization
- SOC Based
  System on Chip - NIC + CPU
  - Good price-performance
  - C Programmable Processors
  - Highest Flexibility
  - Easiest programmability

Figure 1: DPU-based NIC card comparison

Therefore, Field-Programmable Gate Arrays (FPGAs) has been proven to be a middle ground in this scenario. With their unique mix of configurable hardware logic, its NIC can deliver great network processing performance while keeping the development cost rather low. One can program an FPGA card simply by flashing a program on to the board, which is not possible compared to its ASIC counterparts [3].

Such that, the project aims to explore the possibility and usability of offloading lower layers of network connections to FPGAs. This includes building an offload engine that processes incoming network traffic on the board and exposing designed APIs for software to access processed application data.

2 Project Overview

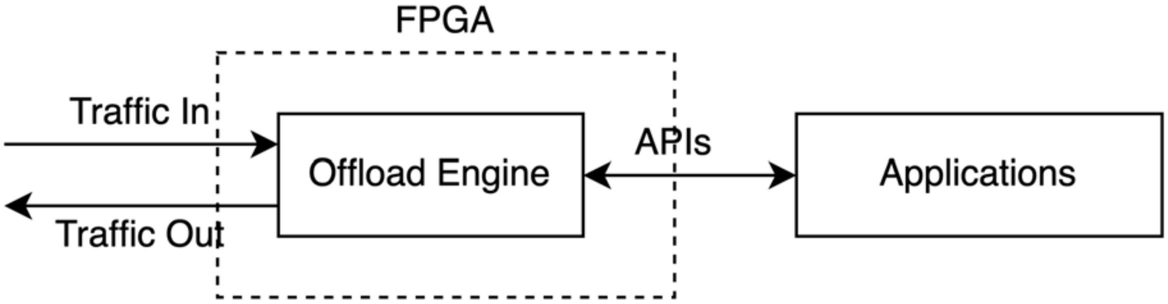


Figure 2: Design Overview

In this project, the main goal is to build an offload engine that handles Ethernet frames coming in and out of the node. The design of the project is shown above in Figure 2. A Xilinx Spartan 3E Starter Kit with an RJ45 connector would be used to handle the traffic as an offload engine. Applications running on host computer connecting to the board via USB (subject to change) would be able to receive and send processed data through the network via the software interfaces provided on the board.

3 Project Objectives

3.1 First Milestone

- To understand networking concepts, and implement senders and receivers of Ethernet packets via raw socket provided by the Linux Kernel
- To acquire knowledge used in FPGA programming, setup and understand FPGA development environment and write basic Verilog programs

3.2 Second Milestone - Receive Packets

- To implement offload engine in FPGA that would handle packet receiving from the network
- To provide API for software to read processed packets

3.3 Third Milestone - Send Packets

- To implement offload engine in FPGA that would handle packet sending to the network
- To provide API for software to send new packets

3.4 Stretch Goals

- To develop an integrity checking framework to check packet validity
- To implement offload engine to handle ICMP, IP packets

- To implement offload engine to handle UDP/IP packets
- To provide more comprehensive APIs for applications upwards of Layer 3 running on UDP/IP stack

References

[1] Ruining Chen and Guoao Sun. "A Survey of Kernel-Bypass Techniques in Network Stack". In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. CSAI '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 474–477. ISBN: 9781450366069. DOI: 10.1145/3297156.3297242. URL: https://doi.org/10.1145/3297156.3297242.

[2] Defining SmartNIC: What is a SmartNIC and How to Choose the Best One. May 2020. URL: https://blog.mellanox.com/2018/08/defining-smartnic/.

[3] I. Kuon and J. Rose. "Measuring the Gap Between FPGAs and ASICs". In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 26.2 (2007), pp. 203–215.