Meeting Latency and Jitter Demands of Beyond 5G Networking Era: **Are CNFs Up to the Challenge?**

Adil Bin Bhutto[1] Ryota Kawashima[2] Yuzo Taenaka[1]
Youki Kadobayashi[1]

[1]Nara Institute of Science and Technology, Japan
[2]Nagoya Institue of Technology, Japan

Contact: *Adil Bin Bhutto* <adil-b@ieee.org>

July 4th, 2024

Points

Goal

- Explore the **latency**, **jitter**, and **bandwidth** characteristics of **CNFs**.

Idea

- Focus on **CPU Power** and **Frequency Scaling** Configurations.

Result

- Insight for predictable **low latency/jitter** and **high throughput**.

Table of Contents

Compute Inter Connect for Beyond 5G

2 Introduction

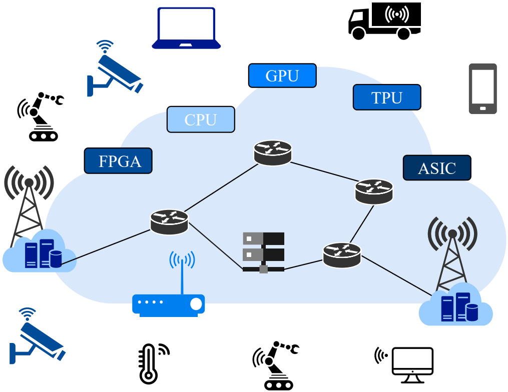


Figure: Compute-Inter-Connect platform with heterogeneous technologies.[1]

[1] Chafii, M., Bariah, L., Muhaidat, S., & Debbah, M. (2023). Twelve scientific challenges for 6G: Rethinking the foundations of communications theory. IEEE Communications Surveys & Tutorials, 25(2), 868-904.
Note: The figure depicted here is a modified version taken from this publication.

Evolving Network Functions

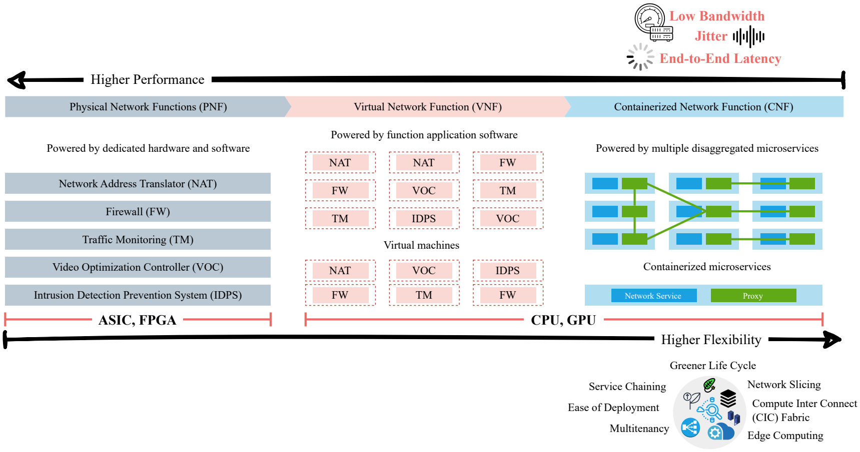


Figure: Evolving Network Functions

Why CNF?

1. **Agility**
   - Lightweight • Fast spin-off time
2. **Portability**
   - Open standards • Wide adoption
3. **Resource efficiency**
4. **Supporting ECO-System**

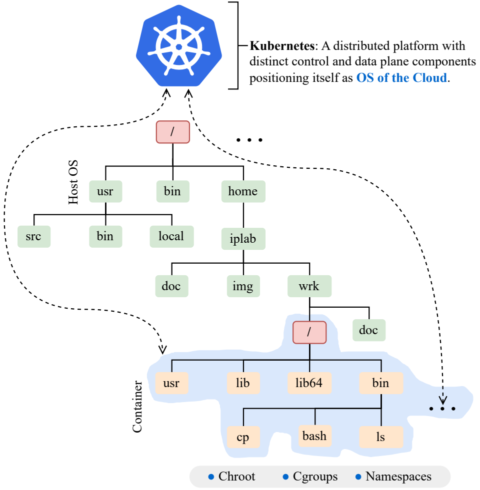


Figure: Container architecture and its orchestration.

Network I/O Acceleration and Virtualization

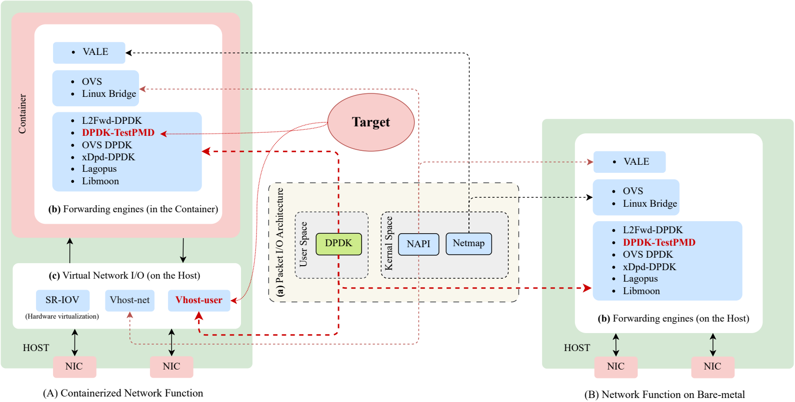


Figure: Performance acceleration and vNet I/O technologies.

Why DPDK?

→ Polling mode
→ User space driver
→ Core affinity
→ Optimized memory
→ Network virtualization
→ Cloud native acceleration

Table of Contents

Why Focus on CPU P&C States?

P-States (Performance States)

√ Operating **Frequency** Control √ **Voltage** Control

- Advantages
  - \+ Dynamic performance scaling
  - \+ Energy Efficiency
  - \+ Thermal management
- Disadvantages
  - – Potential performance impact
  - – Transition latency

C-States (Idle States)

√ **Turns off** Parts of **CPU** √ **Manage Power** Consumption

- Advantages
  - \+ Power savings
  - \+ Extended battery life
  - \+ Reduced heat generation
- Disadvantages
  - – Wake-up latency
  - – Power management complexity
  - – Potential performance penalty

CPU P&C State in CNF

Effect of P&C states on Container

- CPU behavior **impacts** container **performance**.
- Dynamic frequency scaling → Unpredictable process execution speed
- Idle CPUs take longer to wake up and run processes.

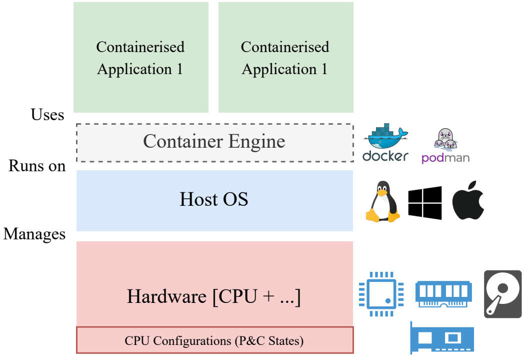


Figure: Relation between containerized application and CPU P&C states.

Table of Contents

Goal

Understand the **correlation** between **CNF performance** and **P&C Sate** of modern CPU.

Address the possibility of **using CNF** in **latency-sensitive applications**.

Approach

4 Our Study

Exhaustive Experiments and Analysis

- 10 Experiments + 152 Evaluations
- DPDK powered packet generation and forwarding CNF
- + Variable packet sized
  + Variable packet rates
  + ±PC
- Real H/W Devices

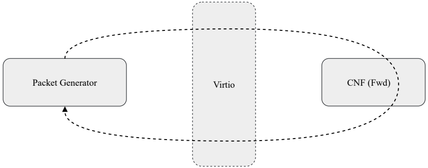


Figure: Abstract Testbed

Evaluation Design

$+$PC (P&C State Enabled)

√ CPU C States Support (enabled) √ Speed Step Technology (enabled) √ Turbo boost technology (enabled) √ Speed shift Technology (enabled) √ Thermal Velocity Boost Voltage Optimizations (enabled)

$-$PC (P&C State Disabled)

× CPU C States Support (disabled) × Speed Step Technology (disabled) × Turbo boost technology (disabled) × Speed shift Technology (disabled) × Thermal Velocity Boost Voltage Optimizations (disabled)

Hardware Virtualization ($+$PC / $-$PC)

× VT-d (disabled) × SR-IOV (disabled)

Others

√ Variable packet sizes √ Variable packet rates
√ Simple l2 forwarding with mac swap of UDP packets

Testbed

4 Our Study

Table 1: Machine specification

Physical	Machine (Tester & DuT)
CPU	Intel® CoreTM i7-13700 5.20 GHz (16 cores w/o HT)
Memory	16 GB (DDR4-3200)
PCIe	PCI Express 4.0 [x16]
NIC	Intel XL710 (i40e)

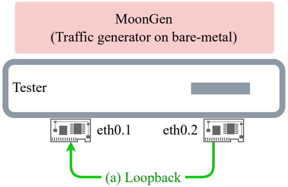


Figure: Testbed

Testbed

4 Our Study

Table 1: Machine specification

Physical	Machine (Tester & DuT)
CPU	Intel® CoreTM i7-13700 5.20 GHz (16 cores w/o HT)
Memory	16 GB (DDR4-3200)
PCIe	PCI Express 4.0 [x16]
NIC	Intel XL710 (i40e)

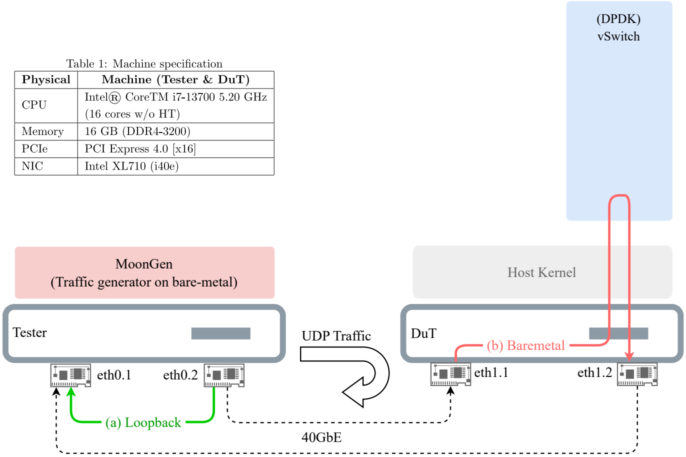


Figure: Testbed

Testbed
4 Our Study

Table 1: Machine specification

Physical	Machine (Tester & DuT)
CPU	Intel® CoreTM i7-13700 5.20 GHz (16 cores w/o HT)
Memory	16 GB (DDR4-3200)
PCIe	PCI Express 4.0 [x16]
NIC	Intel XL710 (i40e)

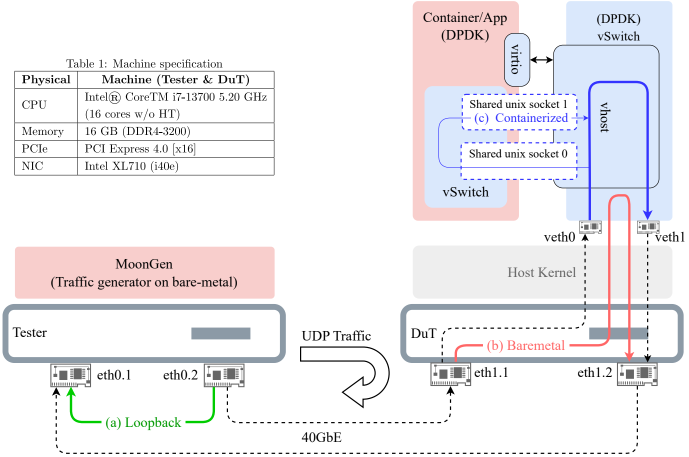
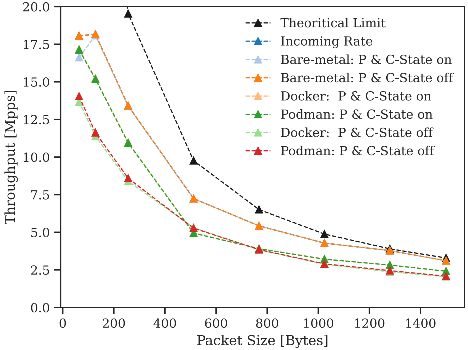


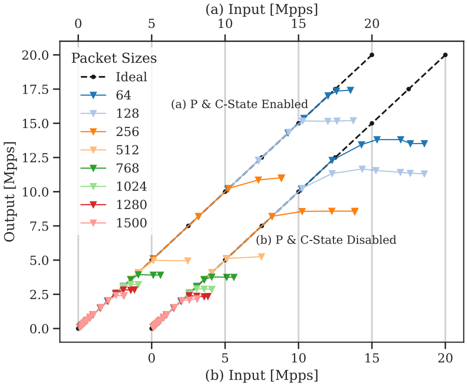
Figure: Testbed

Table of Contents

Throughput

5 Results





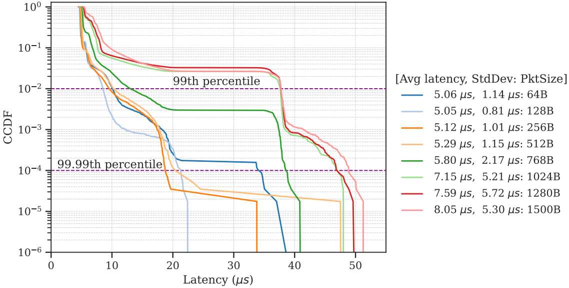
64-800 → BNF(+PC) ≈ BNF(-PC) > CNF(+PC) > CNF(-PC)
800-1500 → BNF(+PC) ≈ BNF(-PC) > CNF(-PC) ≈ CNF(+PC)
→ BNF(±PC) ≈ LIMIT(40GbE)

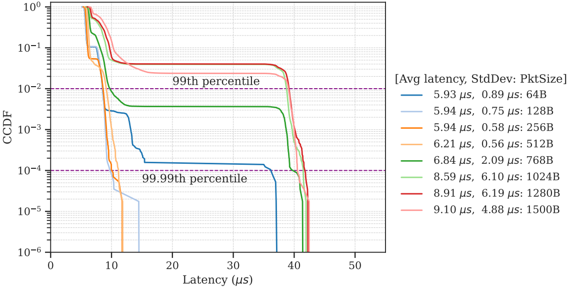
*BNF: Baremetal Network Function

64-800 → CNF(+PC) > CNF(-PC)
→ Saturation: **High Packet Rate**

800-1500 → CNF(-PC) ≈ CNF(+PC)
→ Saturation: **High Data Rate**

Latency and Jitter: CNF at 100 Kpps

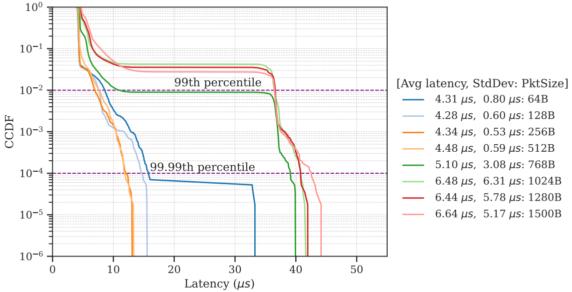


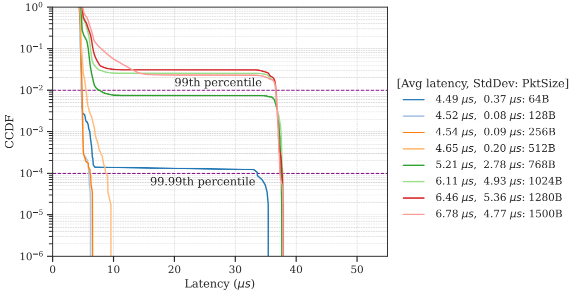


+PC : Latency (↓) Jitter (↑)

−PC : Latency (↑) Jitter (↓)

Latency and Jitter: BNF at 100 Kpps



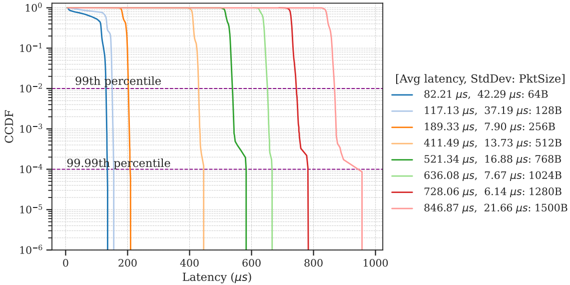


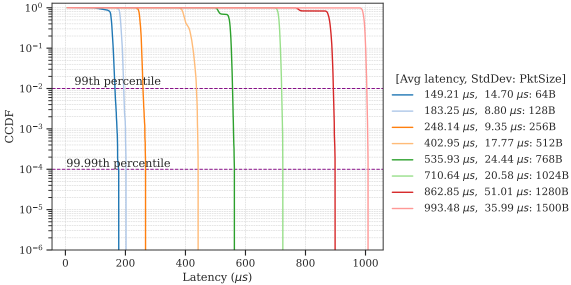
+PC / −PC : **Latency (≃)** **Jitter (≃)**

BNF → Latency < CNF → Latency

BNF → Jitter < CNF → Jitter

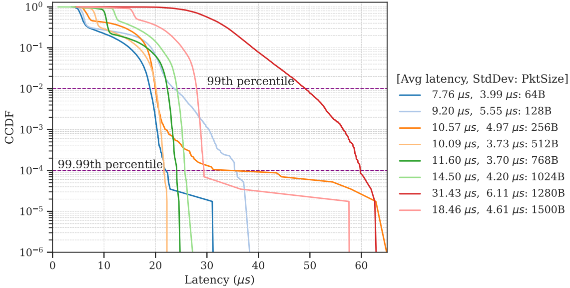
Latency and Jitter: CNF at Max Incoming Rate

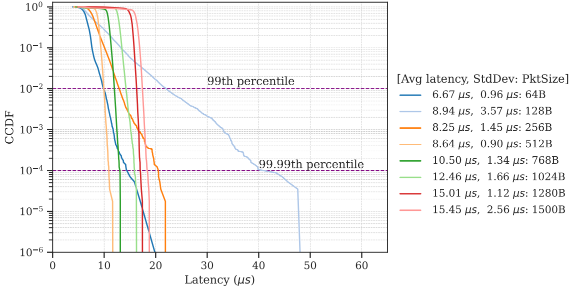




+PC / −PC : Latency(Unusable ↑) Jitter(Unusable ↑)

Latency and Jitter: BNF at Max Incoming Rate





+PC : Latency (↑) Jitter (↑)

−PC : Latency (↓) Jitter (↓)

Findings

1. The **default implementation** of CNF is **not suitable** for latency-sensitive NF.
2. **DPDK** shines in baremetal and is **capable** of **high throughput** and **low latency** with **predictable jitter**.
3. Introduction of vNet I/O in CNFs $\rightarrow$ Might be the Culprit for poor performance

Table of Contents

Conclusion and Future Work

- Found CNFs are more prone to system settings than baremental NF.
- Needs **further study** towards the **explainability** of performance variations.

Future Direction

- **Explain** the observed **performance disparity** in CNFs through further analysis and instrumentation.
- Develop a better and **improved** version of **vNet I/O**.

CPU Probing using RD-TSC [Ongoing]

- **Q2.** Why does CNF suffer from poor performance compared to bare-metal NF?
  - **RO2.1.** To determine the network component in the CNF architecture causing the bottleneck.

- Need to **measure** the **performance** of networking **components** of **Virtio** and find the performance bottleneck with a low overhead method like reading TSC counter of CPU (*RD_TSC*).
- Propose a solution to **address** the **bottleneck**.
- Conduct **combinational experiments** to narrow down the effect of P&C states further.

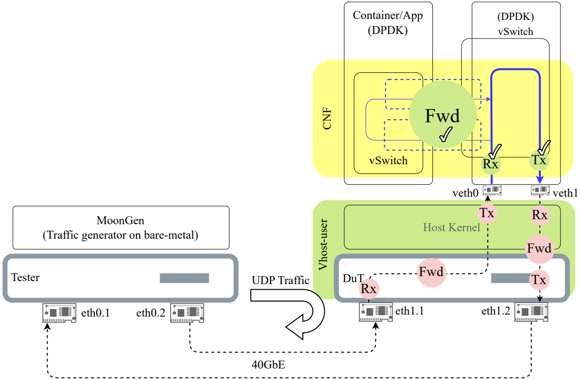


Table of Contents

Understanding CNF Perfromance

Key Points

- $+$ **P&C-State** $\Rightarrow$ Throughput $\uparrow$ Jitter $\uparrow$
- **Low Latency/Jitter Application** ✓
  $\sim$ Packet Size $\times$ Traffic Rate $\sim$ CPU Configuration.