



# PAM: When Overloaded, Push Your Neighbor Aside!

Zili Meng\* Jun Bi\* Chen Sun\* Shuhe Wang\* Minhu Wang\* Hongxin Hu†

\*Tsinghua University

†Clemson University

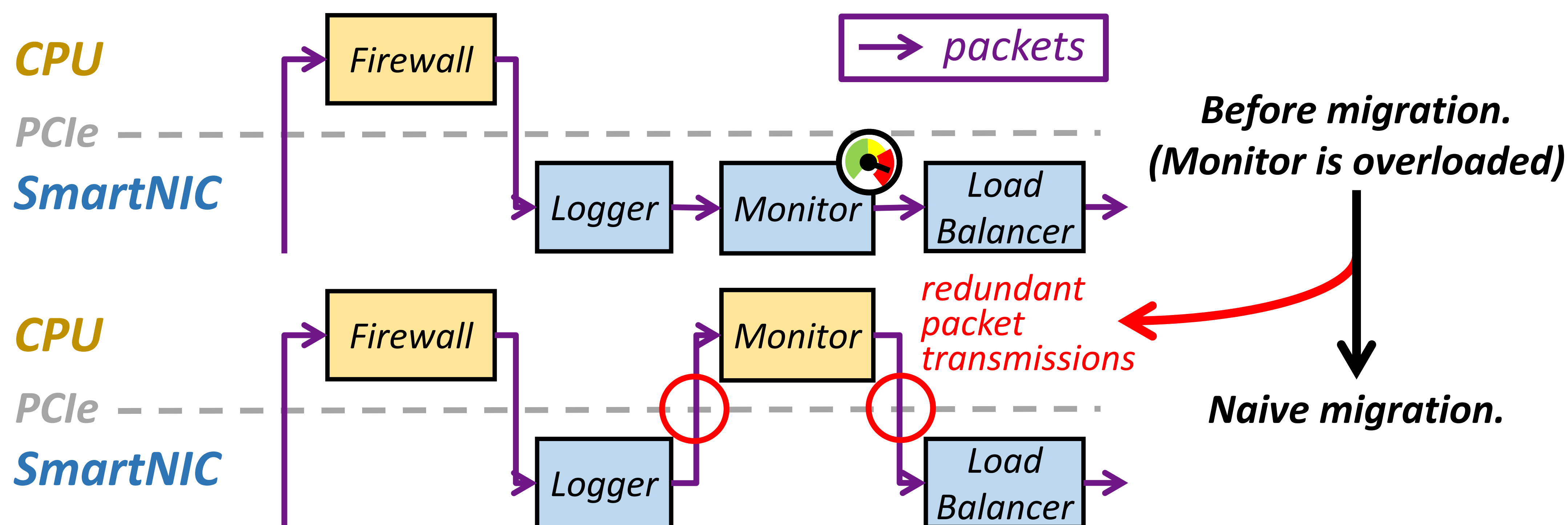


**Motivation:** Alleviate *single NF* hotspot at *service chain* scope.

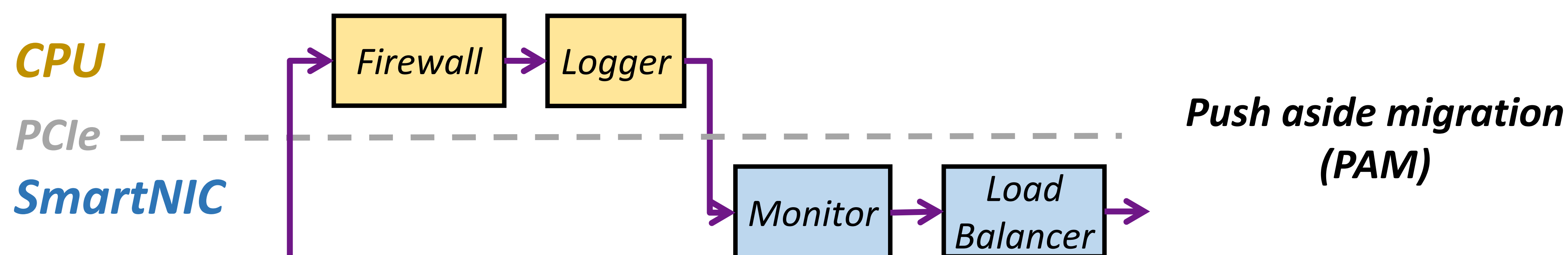
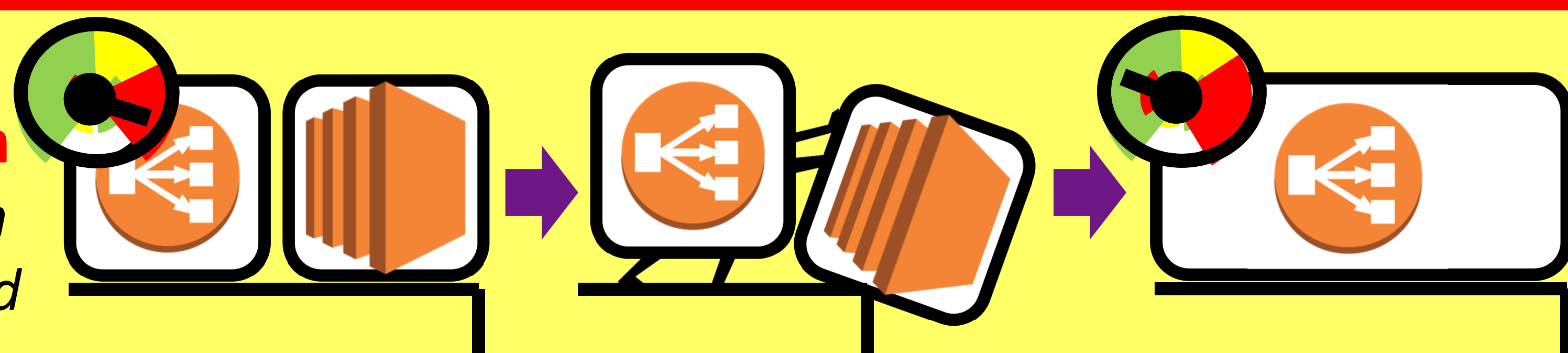
Offload NFs to SmartNIC: NPU-based (Netronome), FPGA-based (Azure)

NF migration between devices: UNO [SoCC'17]

Migrating overloaded NF may lead to performance degradation



**Key Novelty:**  
**Push Aside Migration**  
When overloaded, push your neighbor aside and occupy its resources.

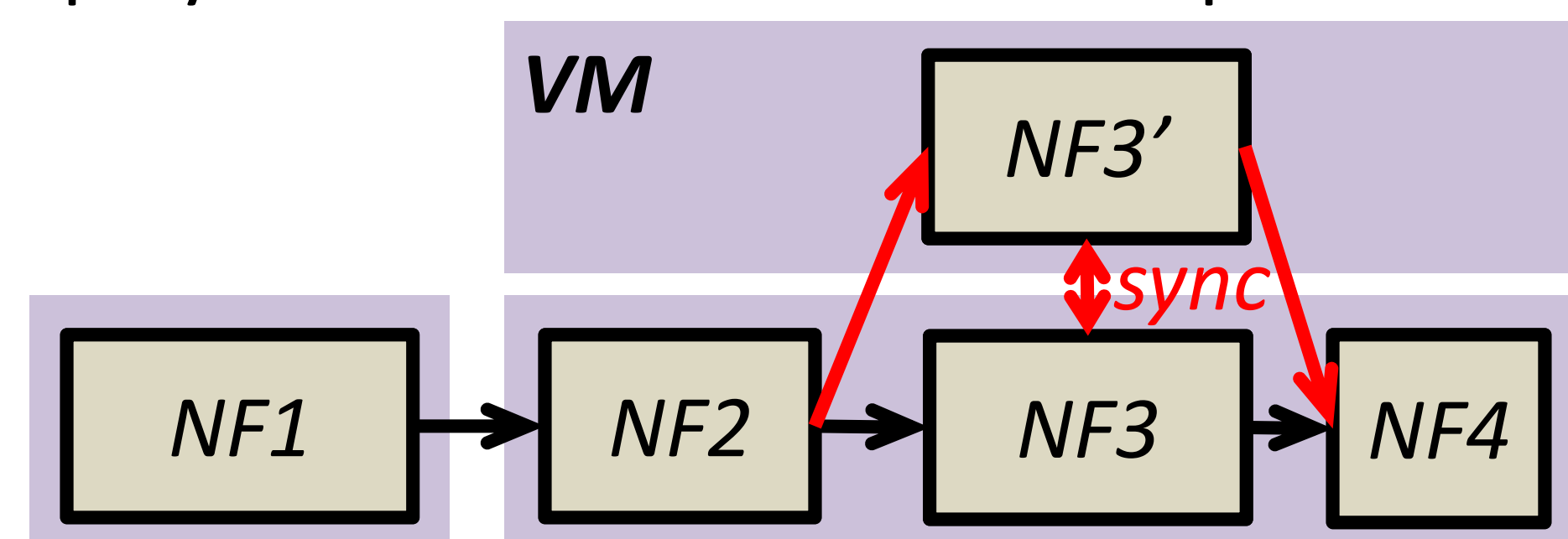


Redundant transmissions avoided & overload alleviated.

## Discussions.

Other application scenarios:

Deploy service chains across multiple VMs.



## Future Work.

Difference of NF processing on both devices:

- Is it suitable for a certain NF to be offloaded?

Precise analysis on PCIe:

- With PCIe model (pcie-bench [SIGCOMM'18])

More types of SmartNIC:

- FPGA-based (ClickNP [SIGCOMM'16])

**Design:** Migrate elements with minimum migration cost.

How to achieve minimum migration cost? – Minimum number of NF to migrate

## Resource Analysis

### Resource Utilization

$\theta_i^C, \theta_i^S$ : throughput capacity of vNF  $i$  on CPU ( $C$ ) or SmartNIC ( $S$ ).

$\theta_{cur}$ : current (or estimated) throughput.

vNF $i$	$\theta_i^S$	$\theta_i^C$
Firewall	10Gbps	4Gbps
Logger	2Gbps	4Gbps
Monitor	3.2Gbps	10Gbps
Load Balancer	>10Gbps	4Gbps
Payload Analyzer	5Gbps	200Mbps

### Assumption

Resource utilization of a vNF increases linearly with its throughput.

$$r_i^S = \frac{\theta_{cur}}{\theta_i^S}, r_i^C = \frac{\theta_{cur}}{\theta_i^C}$$

### Deduction

The capacity  $\theta'$  of the chain  $E_1 \rightarrow E_2$ :

$$\frac{\theta'}{\theta_1^S} + \frac{\theta'}{\theta_2^S} = 1 \Rightarrow \theta' = \frac{\theta_1^S \theta_2^S}{\theta_1^S + \theta_2^S}$$

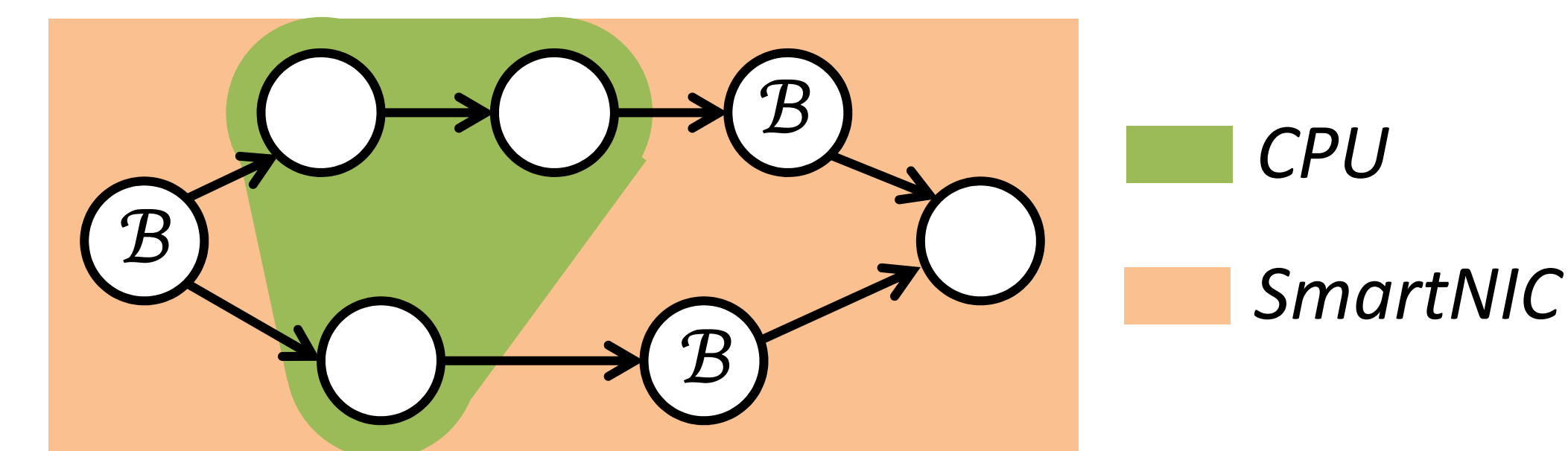
For "Payload Analyzer  $\rightarrow$  Monitor":

$$\theta'_{measure} = 1.8\text{Gbps} \approx \theta'_{theory} = 1.9\text{Gbps}$$

## Border NF Selection Algorithm

### Step 1: Border NF Identification

$\mathcal{B}$ : border elements on SmartNIC in a service chain.



### Step 2: Migration NF Selection

Select the vNF with minimum capacity to alleviate overload with minimum vNF migration:

$$b_0 = \underset{b \in \mathcal{B}}{\operatorname{argmin}} \theta_b^S$$

### Step 3: Overload Alleviation Check

( $\mathcal{C}_1$ ): Migration will not cause new hot spots on CPU.

$$\sum_{i \in \{NFs \text{ on } C\}} \frac{\theta_{cur}}{\theta_i^C} + \frac{\theta_{cur}}{\theta_{b_0}^C} < 1$$

Remove  $b_0$  from  $\mathcal{B}$ . Migrate  $b_0$  if ( $\mathcal{C}_1$ ) is satisfied.

Otherwise go back to **Step 2**.

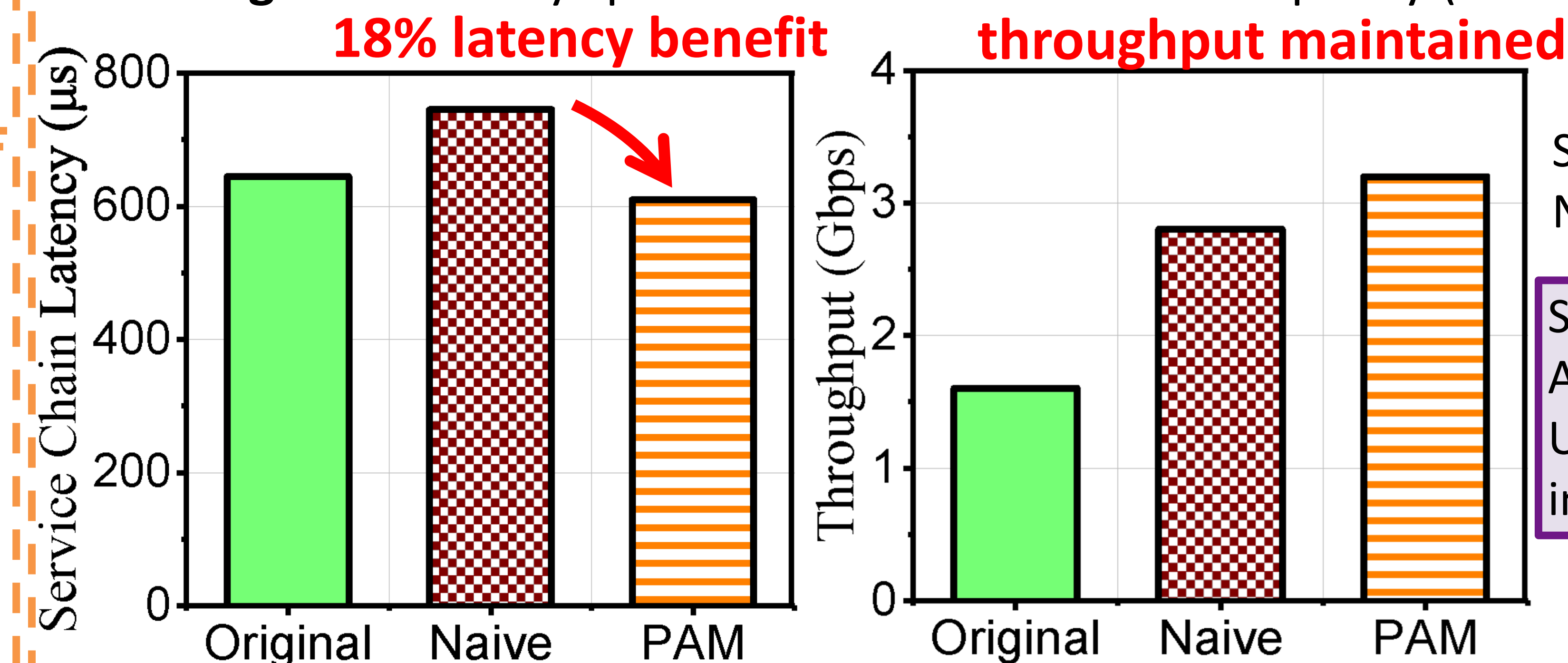
( $\mathcal{C}_2$ ): The overload on SmartNIC should be alleviated.

$$\sum_{i \in \{NFs \text{ on } S, i \neq b_0\}} \frac{\theta_{cur}}{\theta_i^S} < 1$$

Algorithm terminates if ( $\mathcal{C}_2$ ) is satisfied. Otherwise go back to **Step 2**.

**Evaluation:** PAM v.s. naive migration.

Naive migration: always pick the vNF with minimum capacity (overloaded vNF).



Slightly improved since NFs perform differently.

SmartNIC: Netronome Agilio CX 2x10GbE  
Use the service chain in motivation.