

EdgeBalance: Model-Based Load Balancing for Network Edge Data Planes

Wei Zhang[†], Abhigyan Sharma^{*},
Timothy Wood[†]

†George Washington University

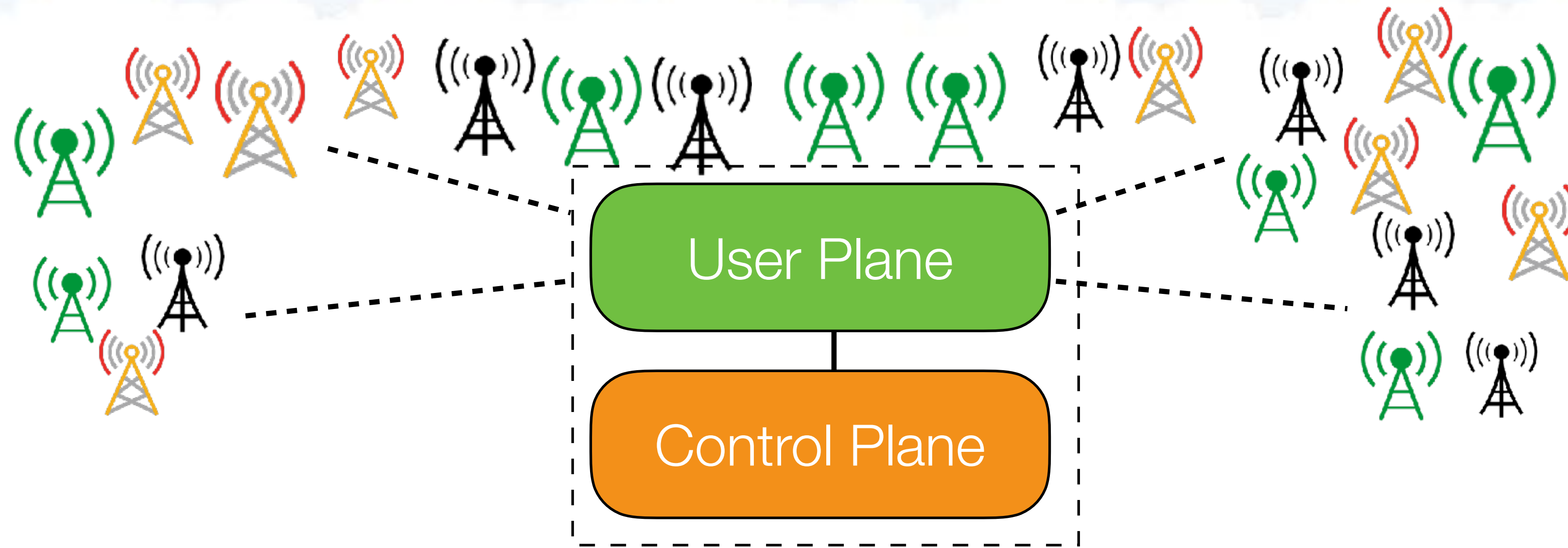
**AT&T Labs Research*

Outline

- **Background and Motivation**
- EdgeBalance Design
- Environment Setup and Evaluation
- Conclusion

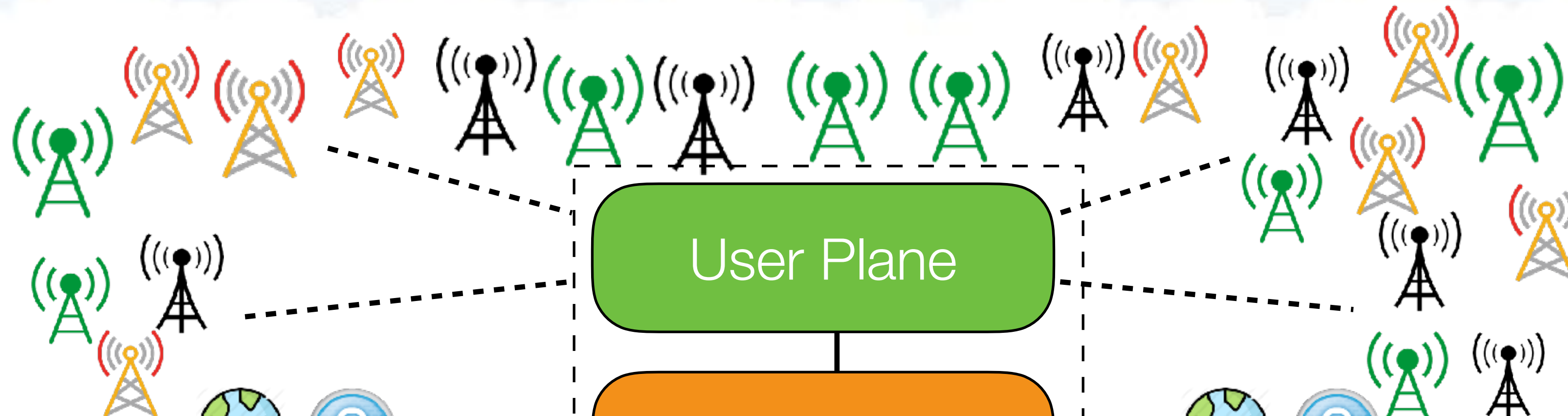
Network Edge Computing

Current

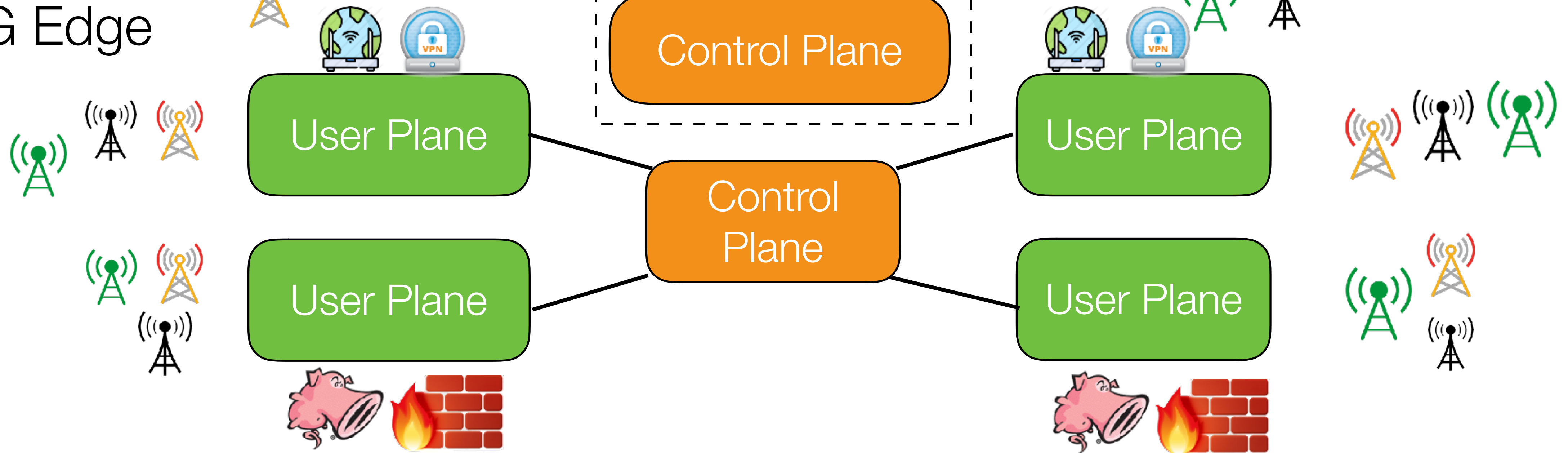


Network Edge Computing

Current

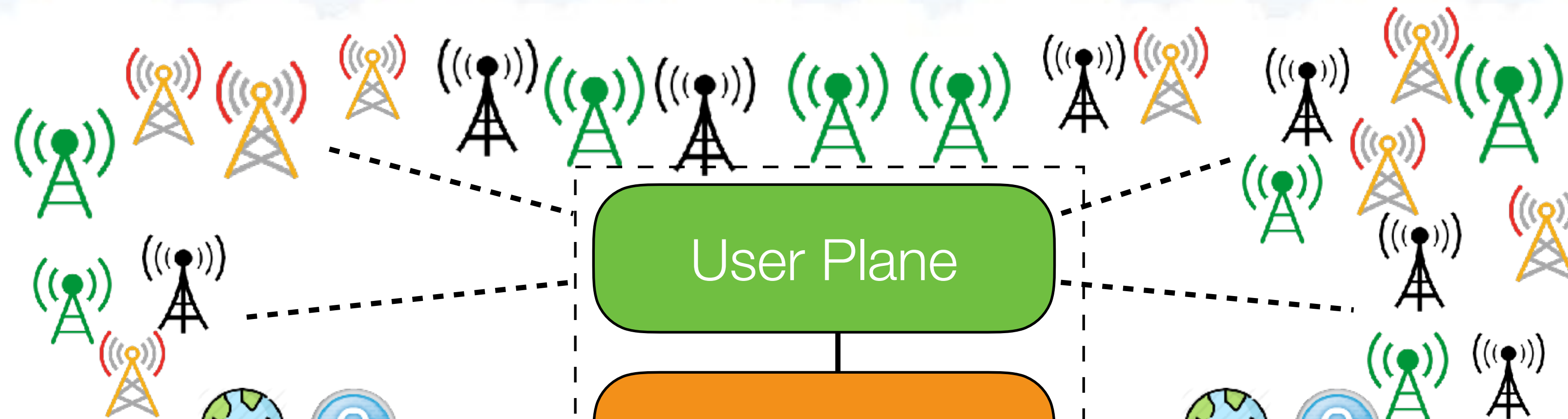


5G Edge

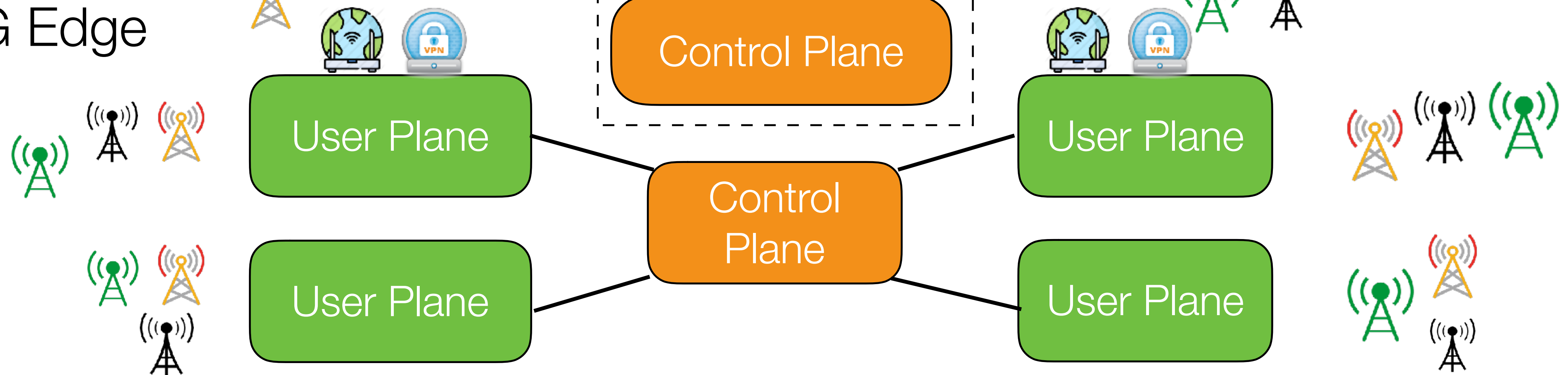


Network Edge Computing

Current



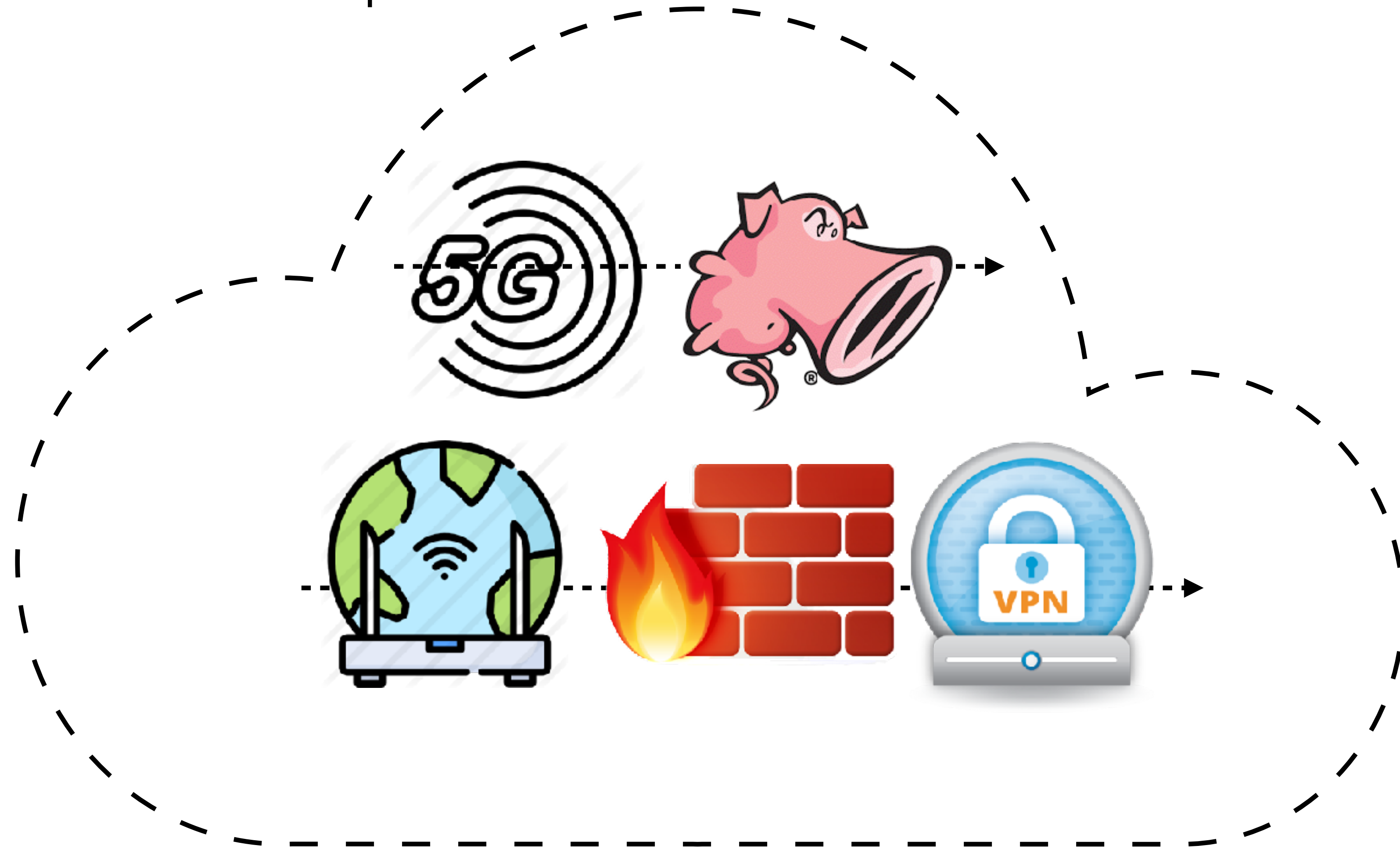
5G Edge



- **User plane more distributed**
- **More network services can be pushed into edge**

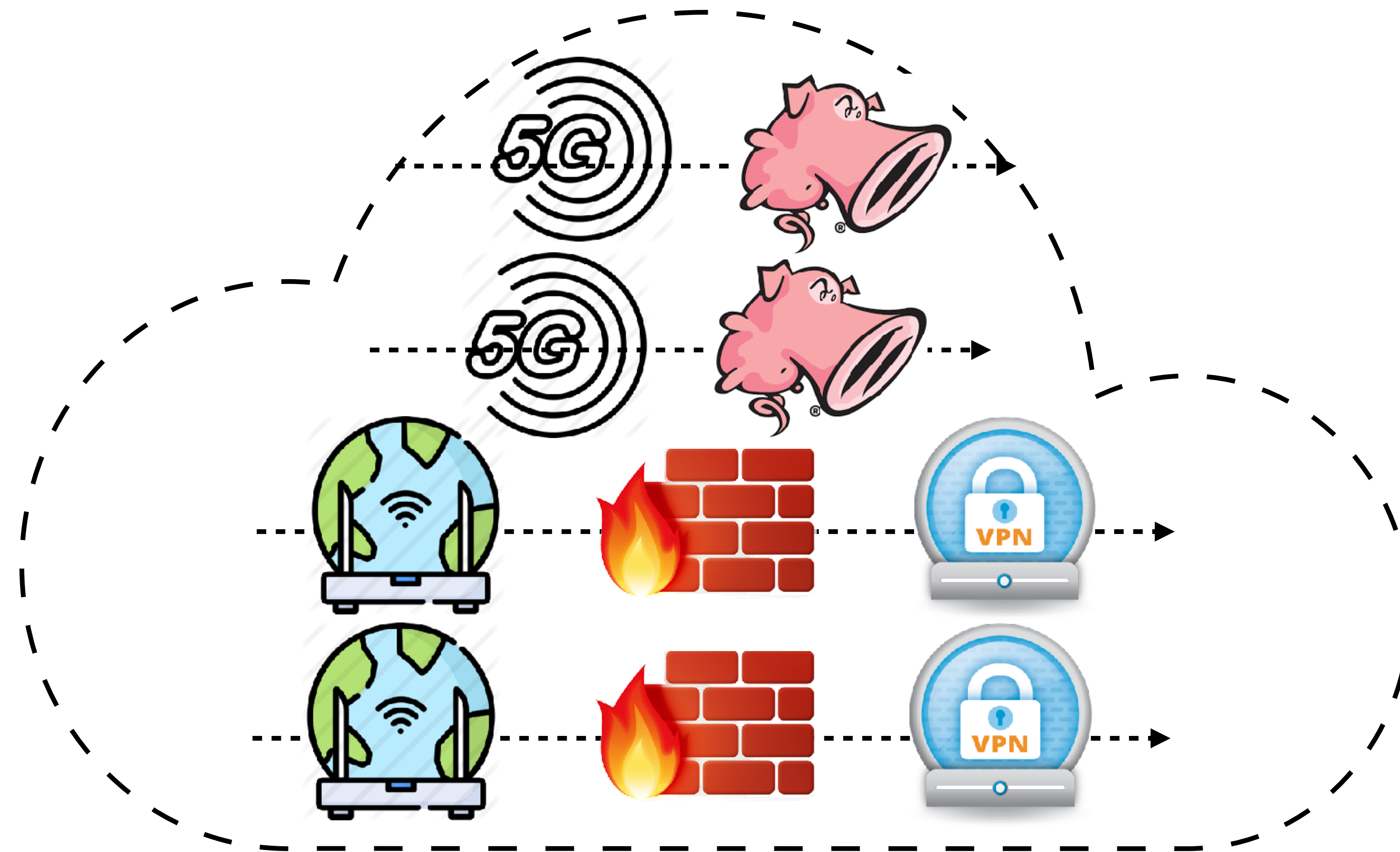
Edge Computing runs NFs

Network services comprise of chains



Edge Computing runs NFs

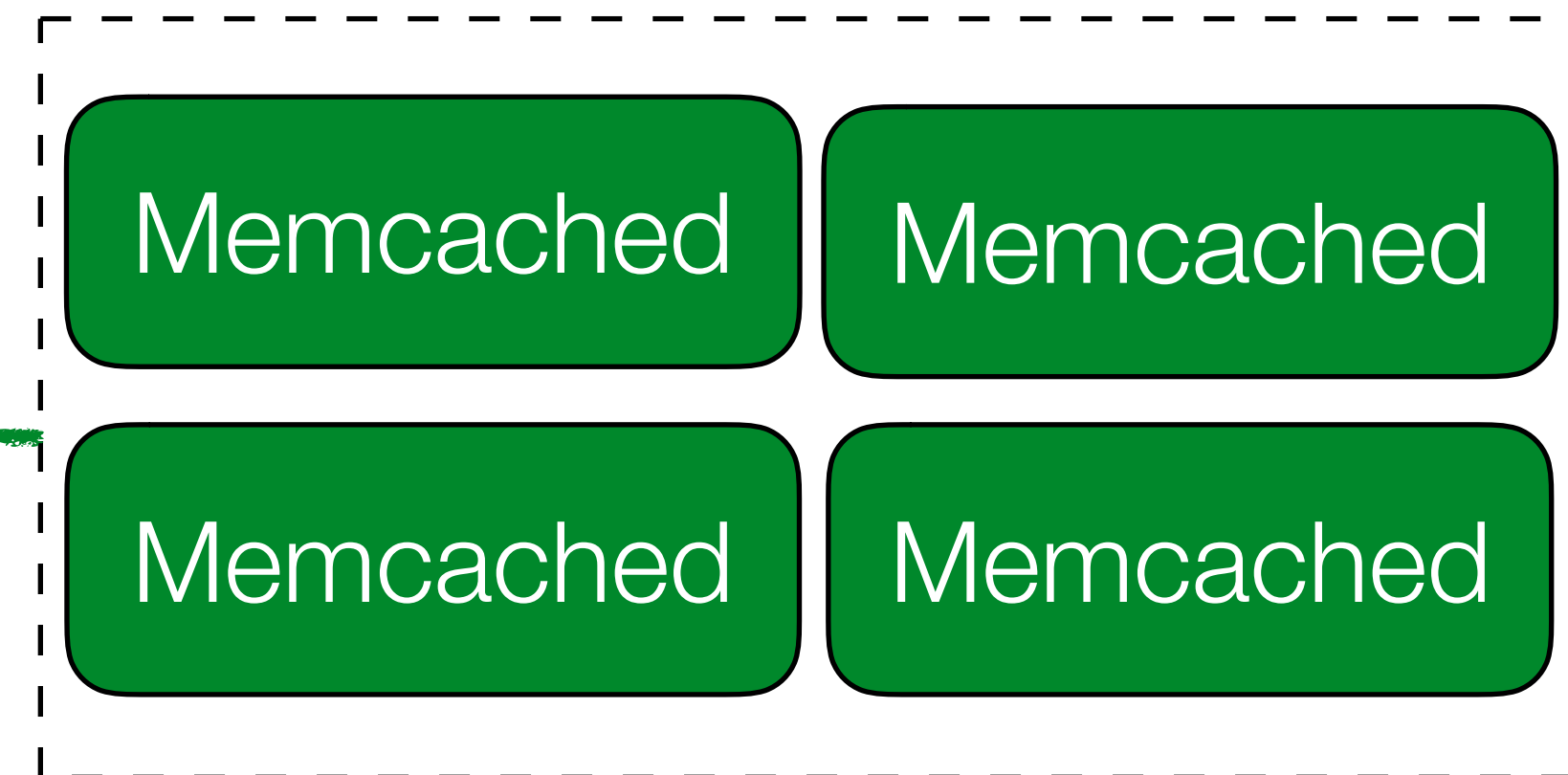
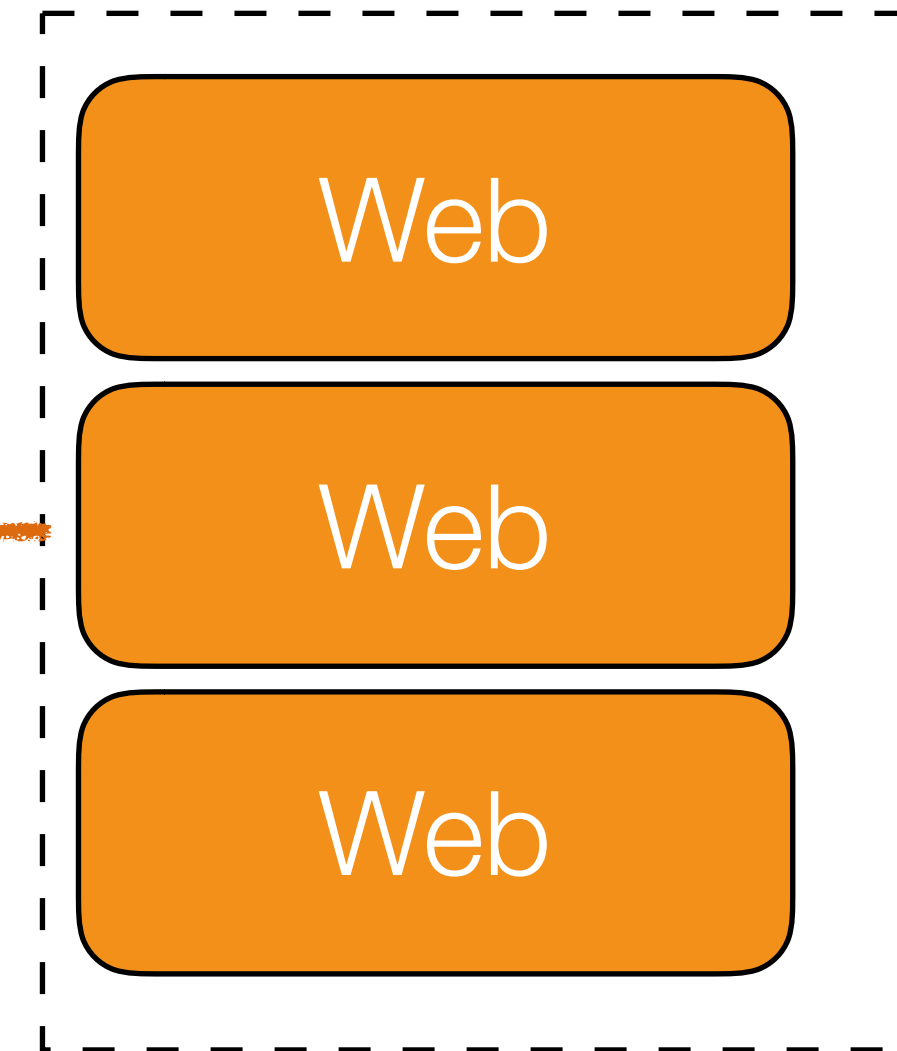
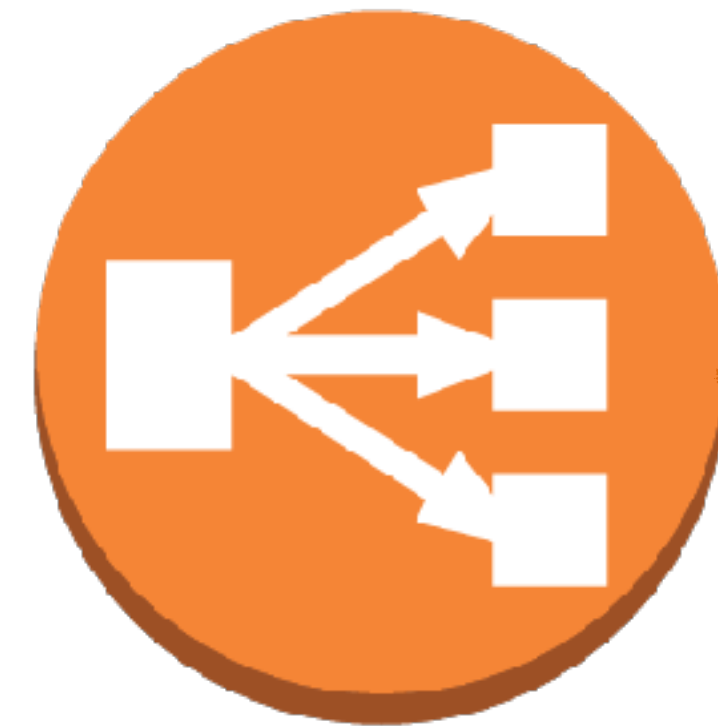
Replicated NF chains require a **load balancer**



Cloud Load Balancer

Homogeneous services, static policies

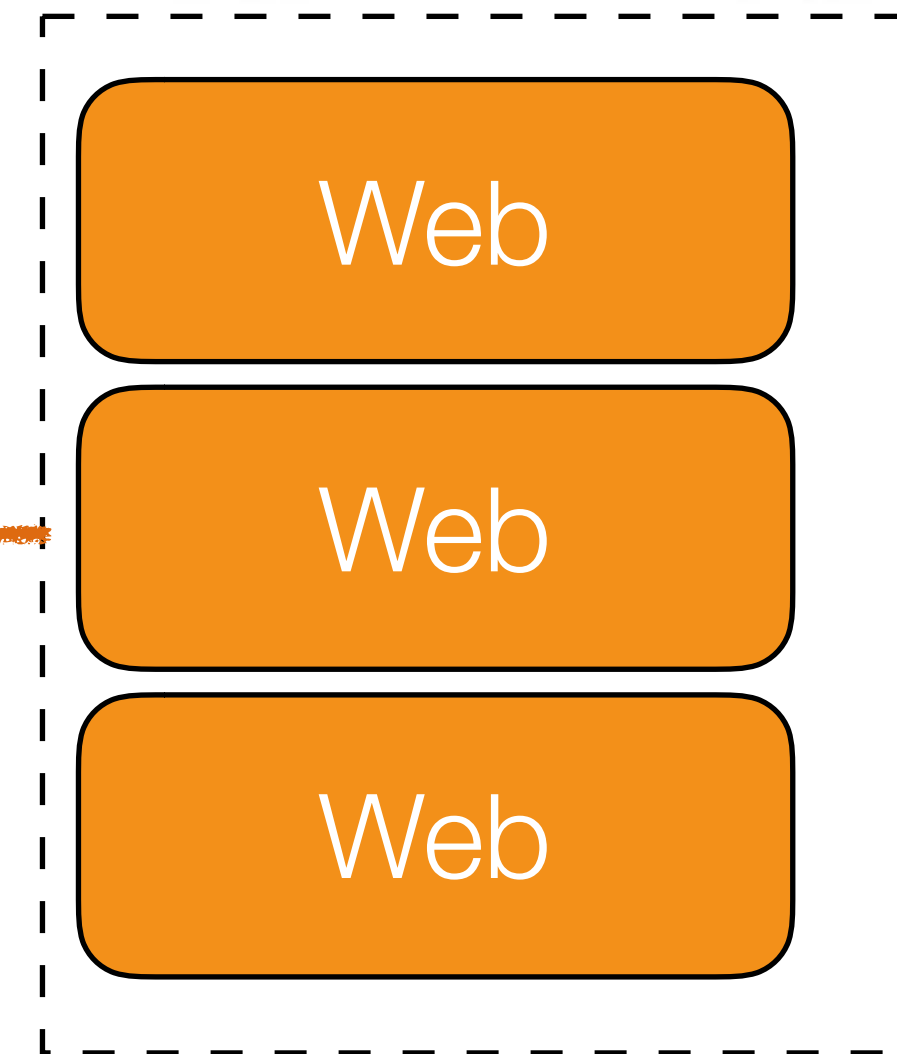
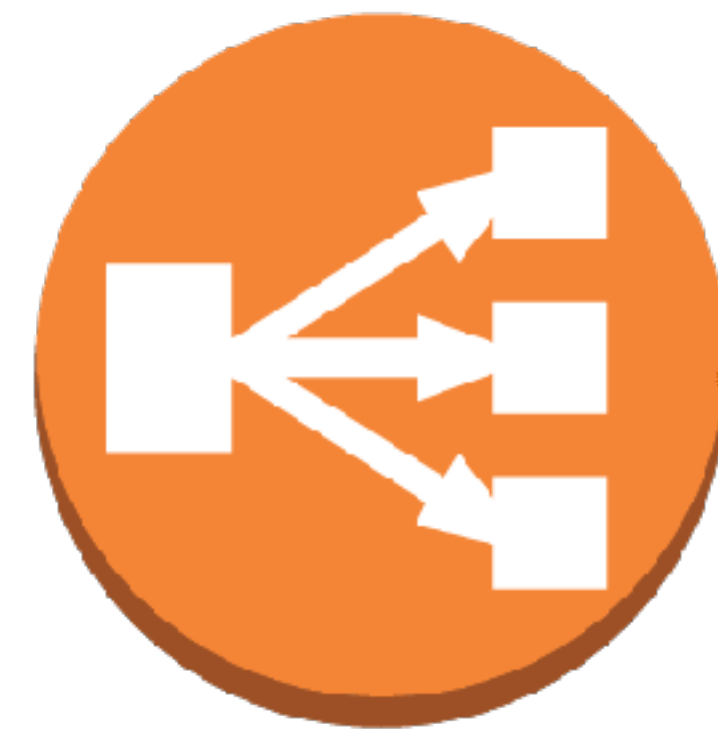
- Google Maglev
- Microsoft Ananta
- Facebook Mcrouter
- Twitter TwemProxy
-



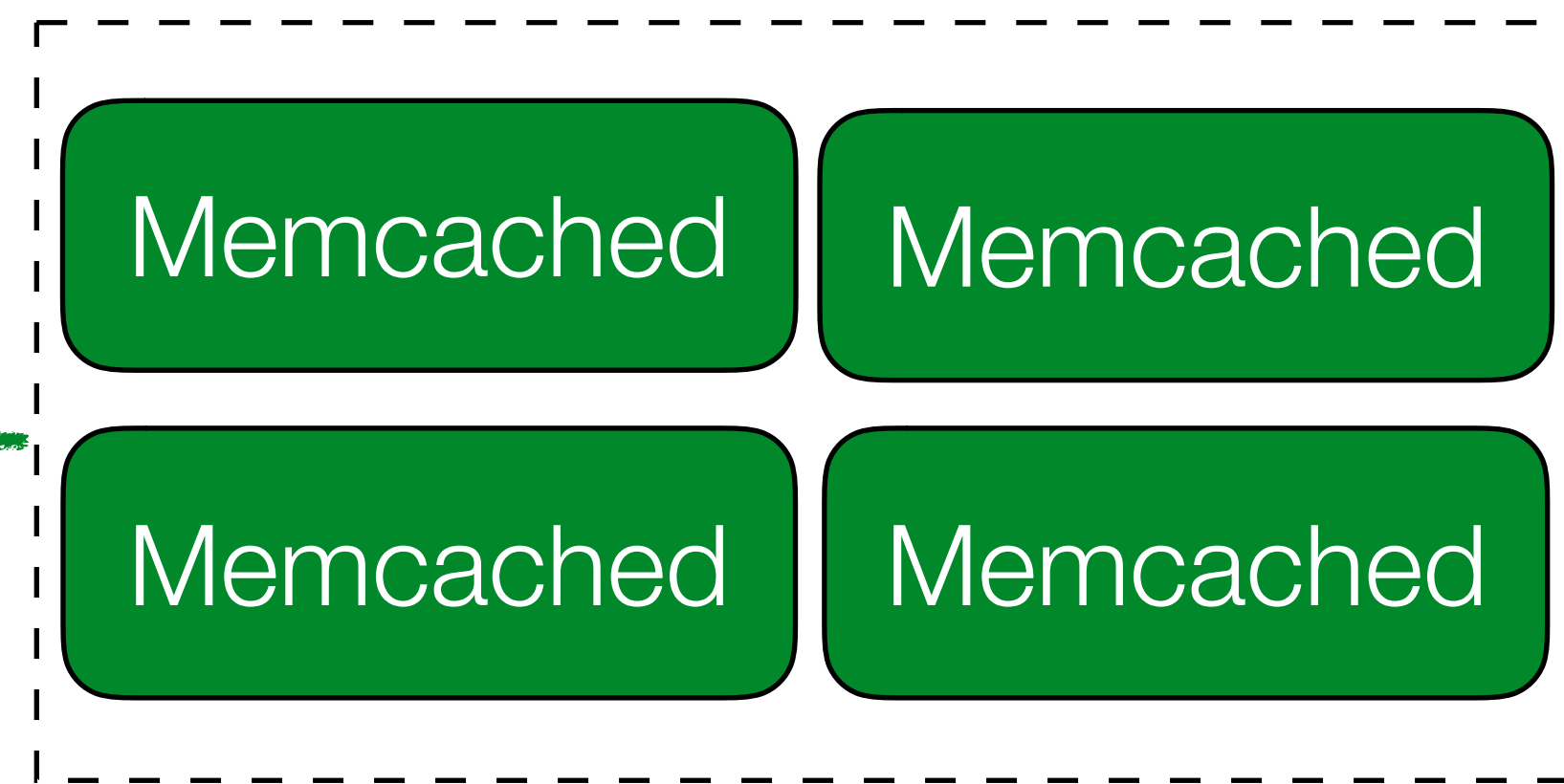
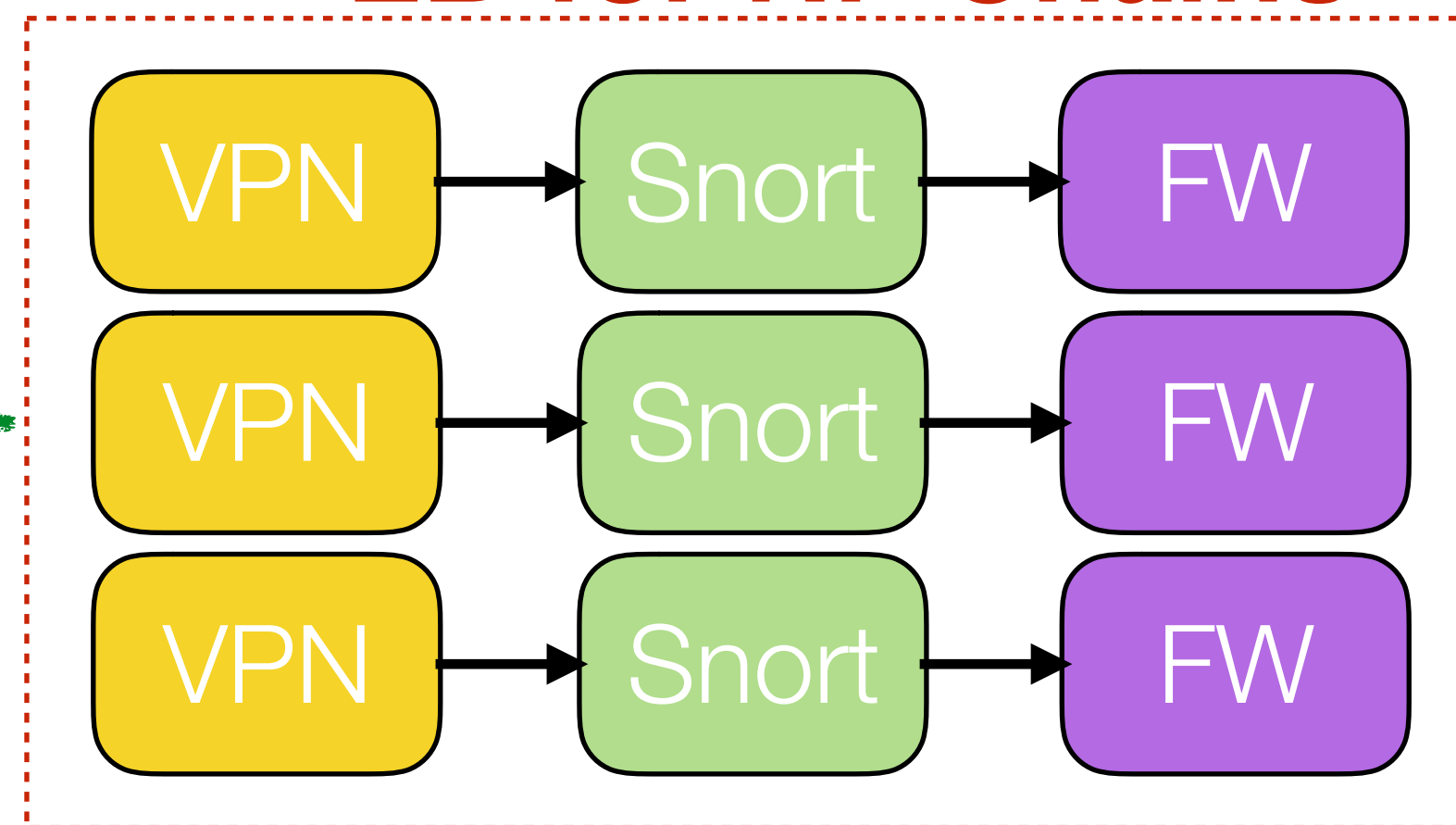
Cloud Load Balancer

Homogeneous services, static policies

- Google Maglev
- Microsoft Ananta
- Facebook Mcrouter
- Twitter TwemProxy

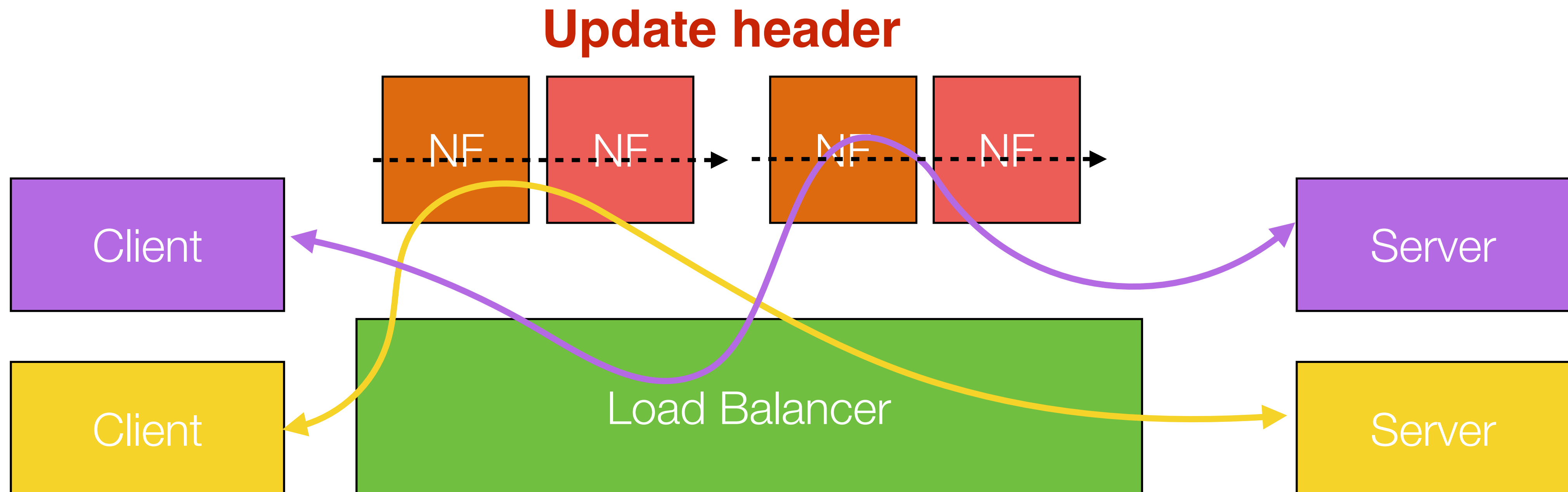


LB for NF Chains



Bidirectional Affinity

Forward and reverse flows are required to pass through the same set of network functions

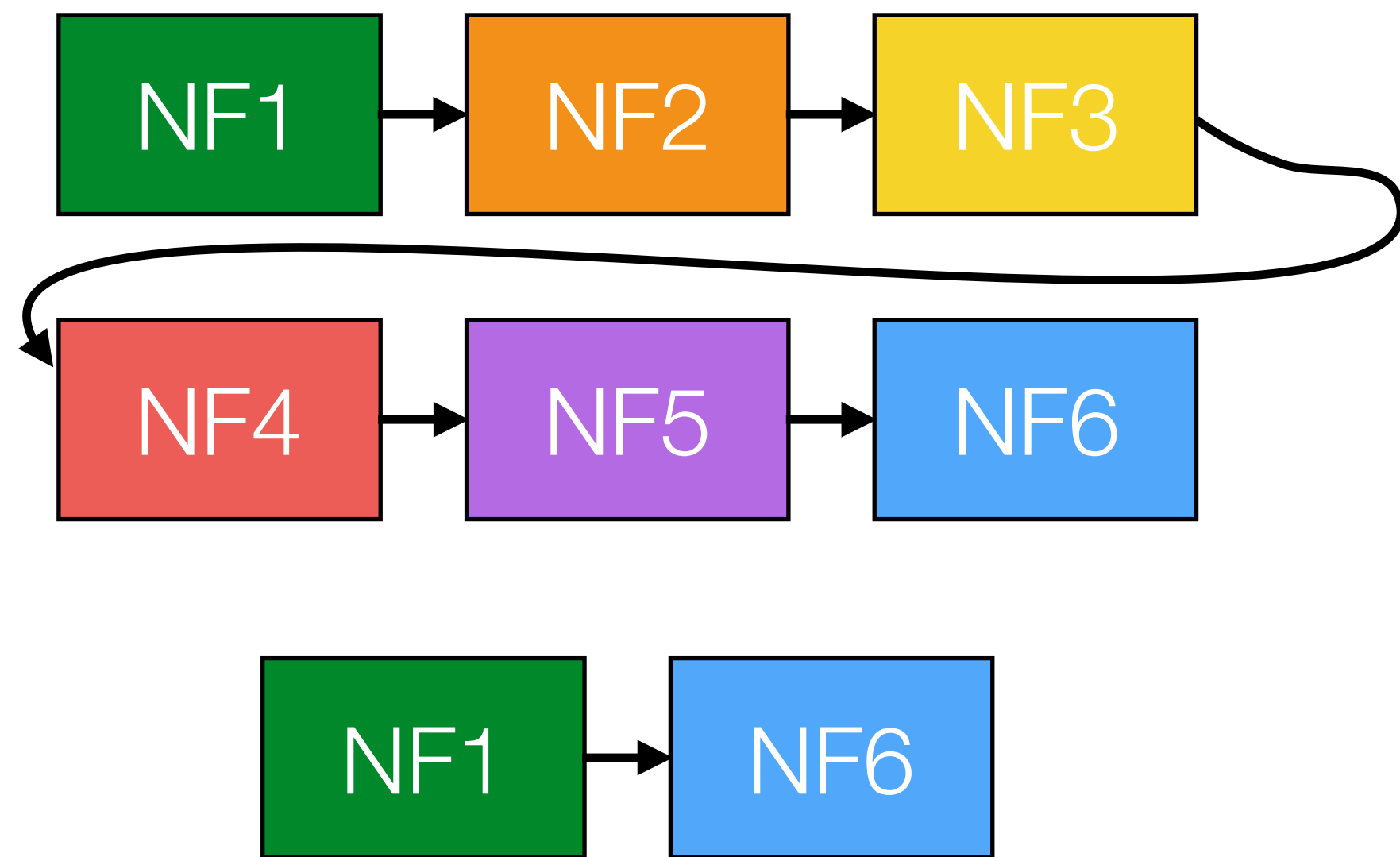


Static LB is not enough

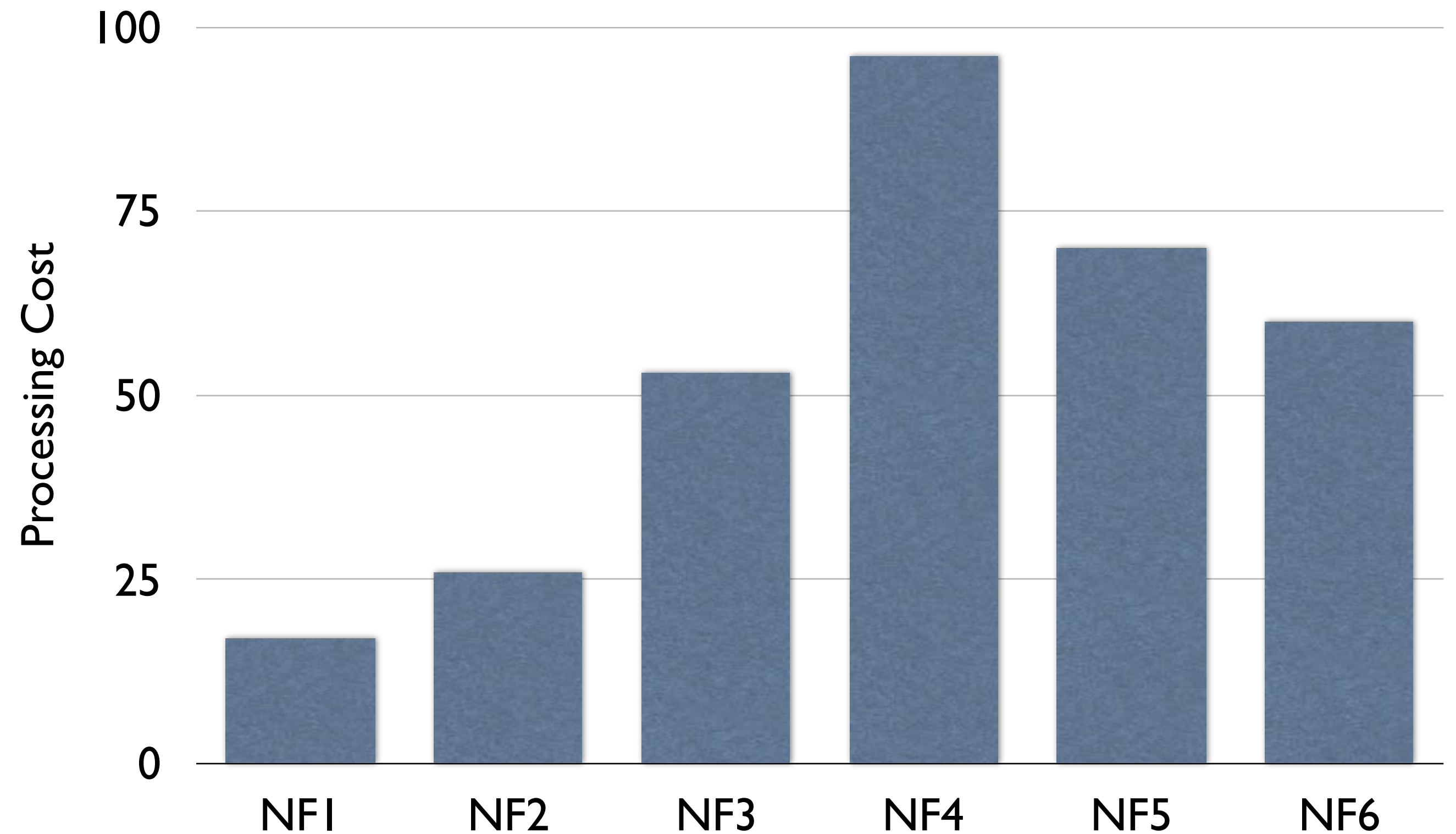
Dynamic server resources

Dynamic service chains

Dynamic workloads



Different Length of NF Chains



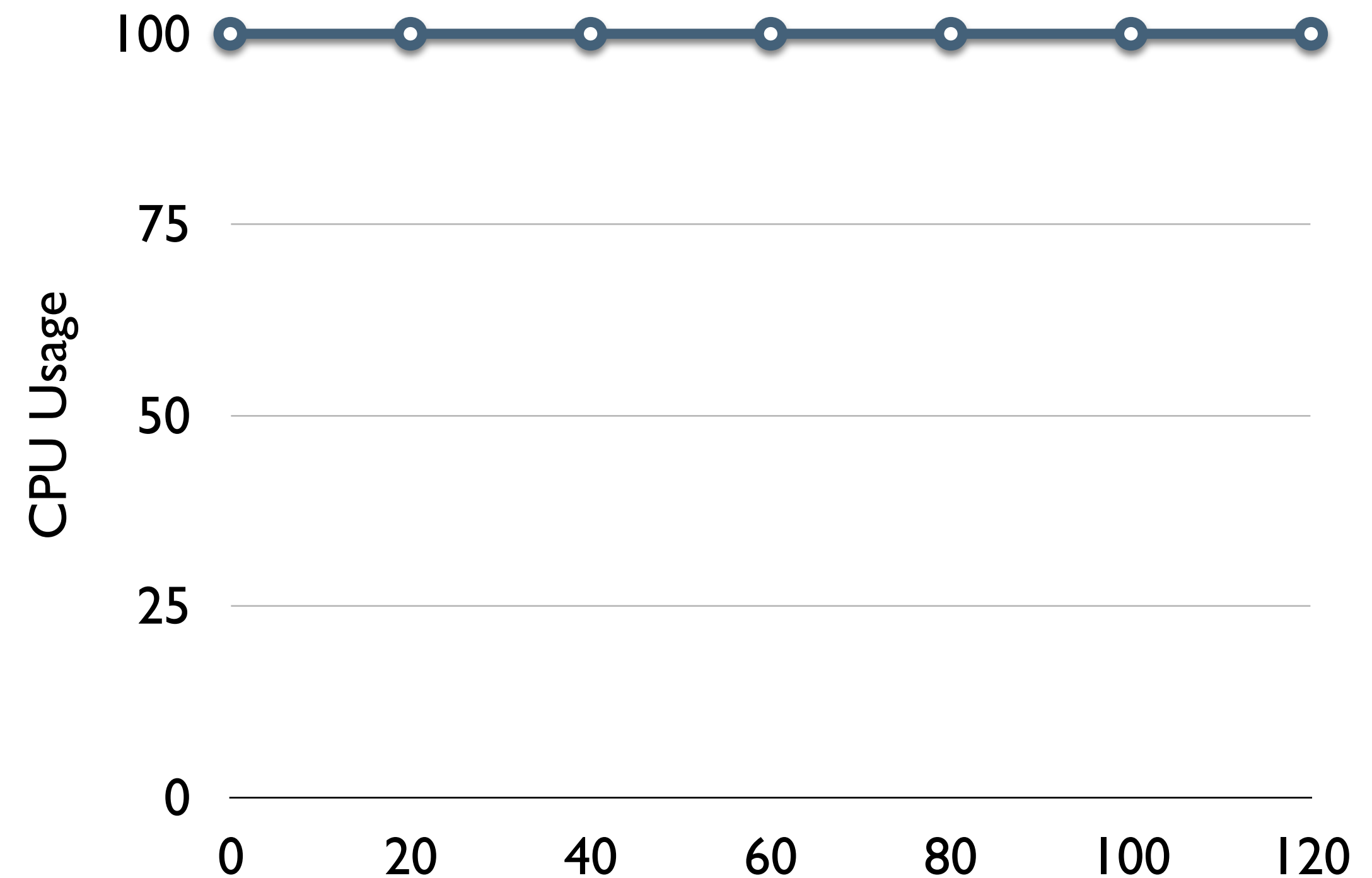
Different Processing Cost of NFs

Challenges for Dynamic LB

Fine-grained metrics are required for dynamic load balancing

Real time CPU monitoring is

- Impossible for poll mode NFs
- May have stale data
- Increases system overhead

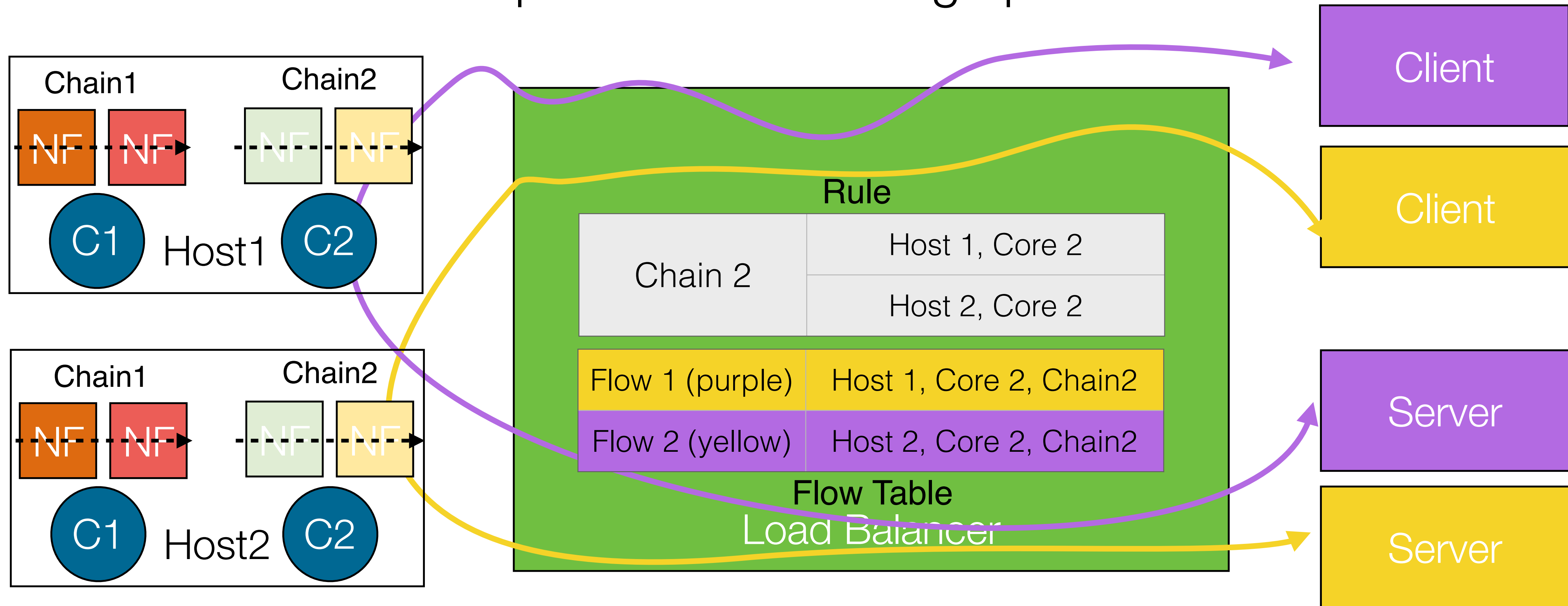


Outline

- Background and Motivation
- **EdgeBalance Design**
- Environment Setup and Evaluation
- Conclusion

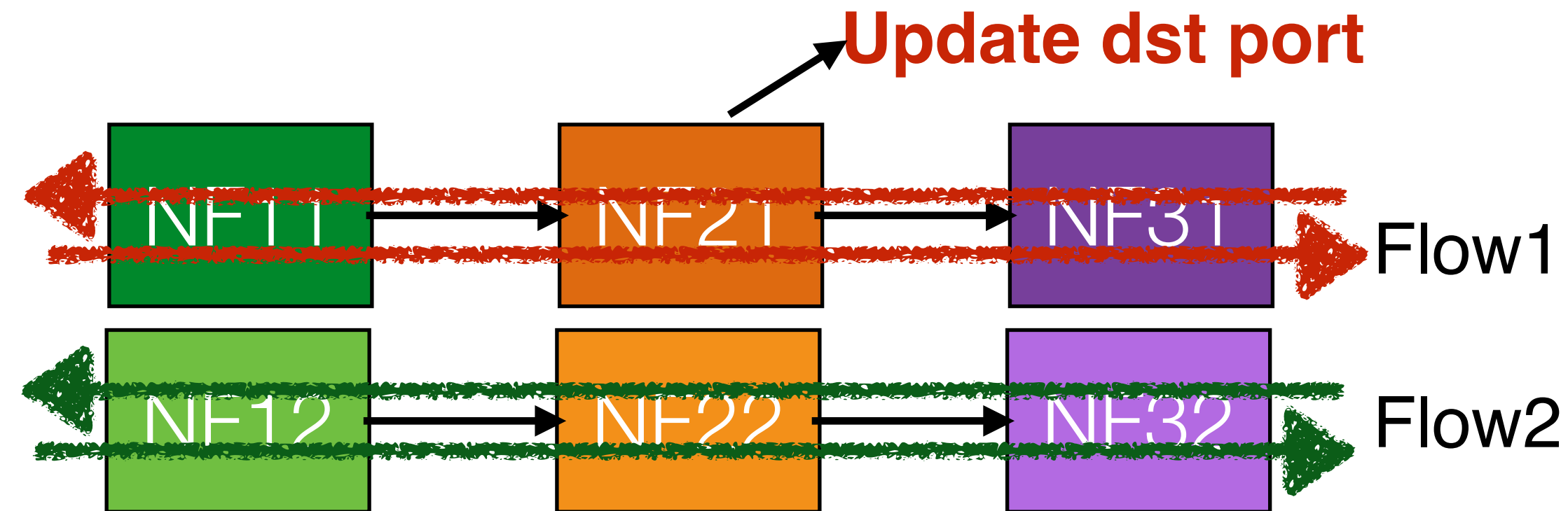
EdgeBalance

1. Bidirectional affinity
2. Accurate CPU prediction
3. Dynamic load balancing
4. High performance



Bidirectional Affinity

Add forward flow and reverse flow into forward table and reverse table



5 Tuple	NF Chains
<srcIP1, dstIP1, udp, srcPort1, dstPort1 >	NF11->NF21->NF31
<srcIP2, dstIP2, tcp, srcPort2, dstPort2>	NF12->NF22->NF32

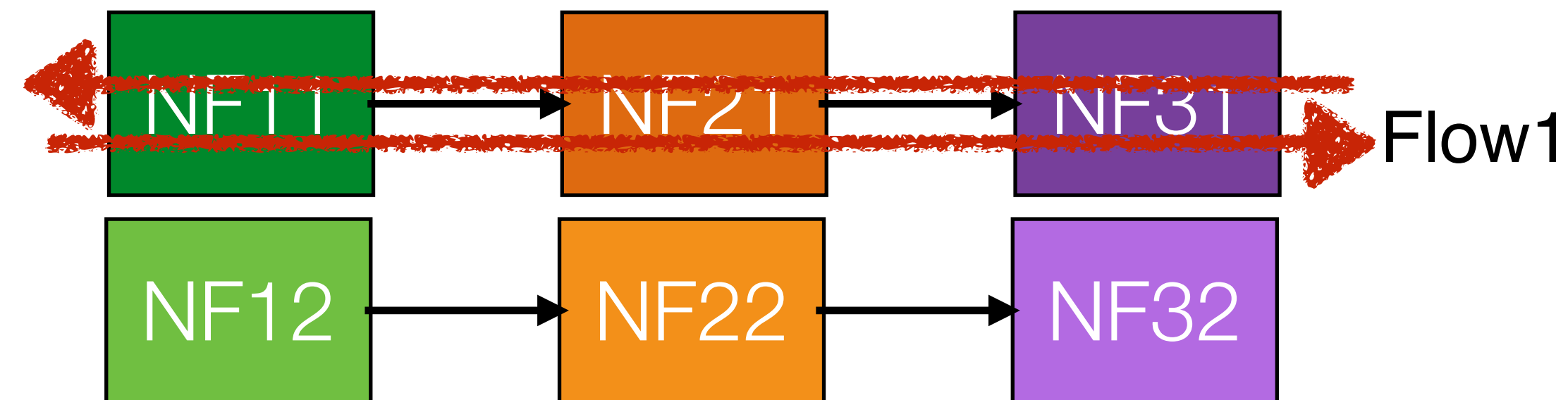
Forward Table

5 Tuple	NF Chains
<dstIP1, srcIP1, udp, dstPort1 , srcPort1>	NF11->NF21->NF31
<dstIP2, srcIP2, tcp, dstPort2, srcPort2>	NF12->NF22->NF32

Reverse Table

Bidirectional Affinity

Remove forward flow and reverse flow from forward table and reverse table



5 Tuple	NF Chains
<srcIP , dstIP , udp, srcPort , dstPort >	NF11->NF21->NF31

Forward Table

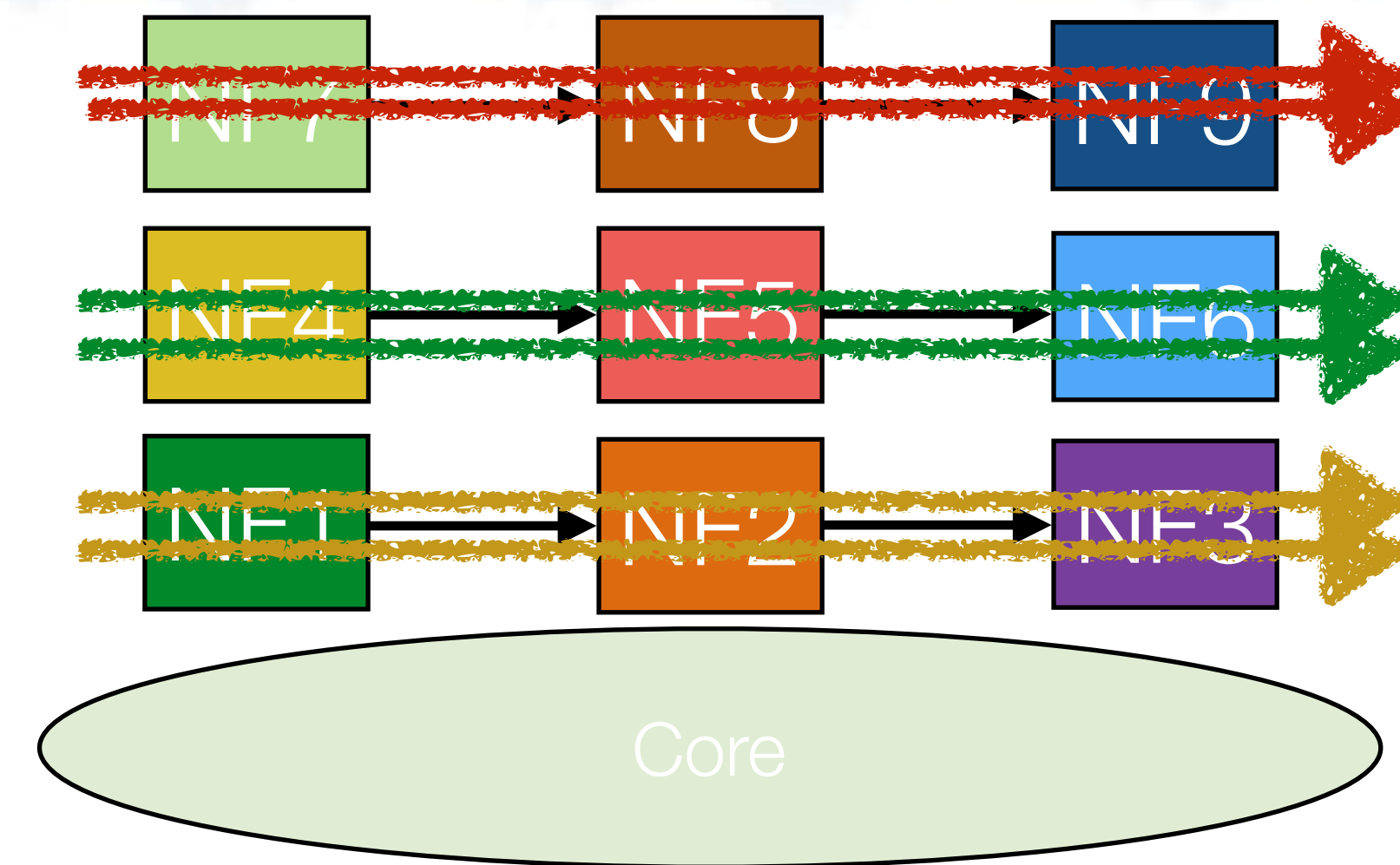
5 Tuple	NF Chains
<dstIP , srcIP , udp, dstPort , srcPort >	NF11->NF21->NF31

Reverse Table

Accurate CPU Prediction

Model to predict CPU usage

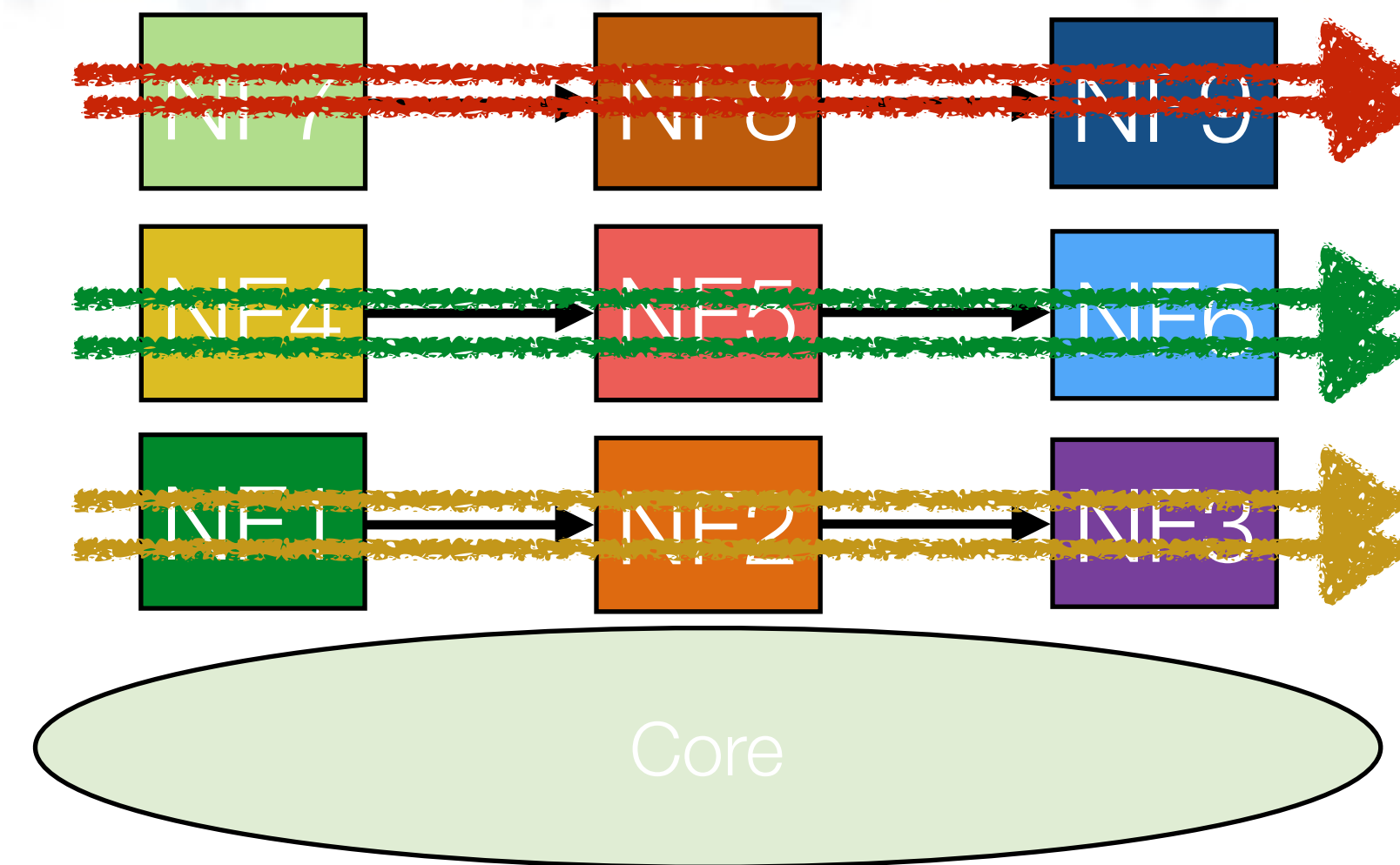
- Chain processing cost
- The number of flows
- Flow rate



Accurate CPU Prediction

Model to predict CPU usage

- Chain processing cost
- The number of flows
- Flow_rate



<hostID, coreID, chainID>	Stats (bits, packets)
<host I, core I, chain I >	xx, yy

$$Predict_CPU_{ijk} = \sum_{k=1}^n Cost_{ijk} * Rate_{ijk}$$

i: host *I*

j: core *j*

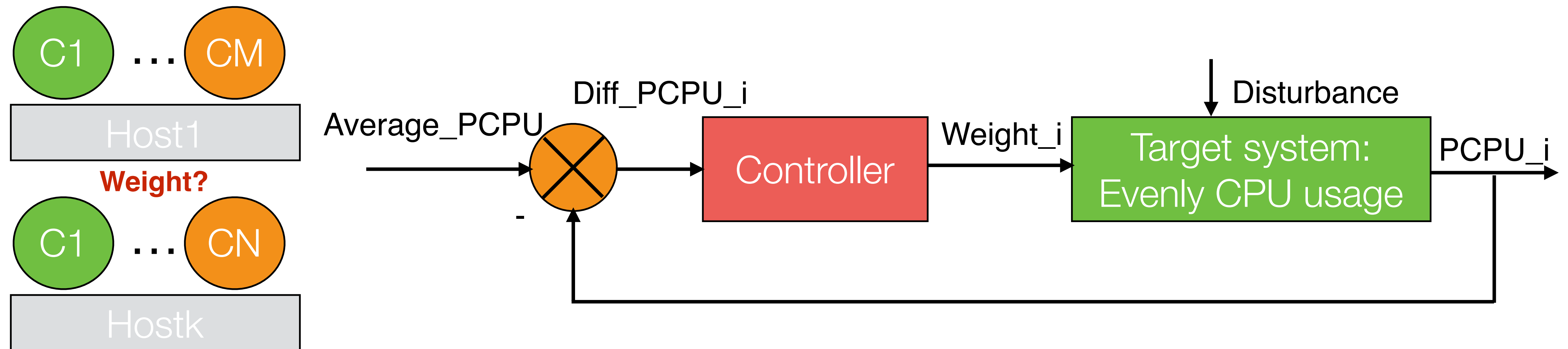
k: number of chains

Dynamic Load Balancing

Balance the load across the hosts and cores

PID controller for LB weight setting

- Seeks to equalize load on hosts and cores
- Uses local information for feedback



High Performance

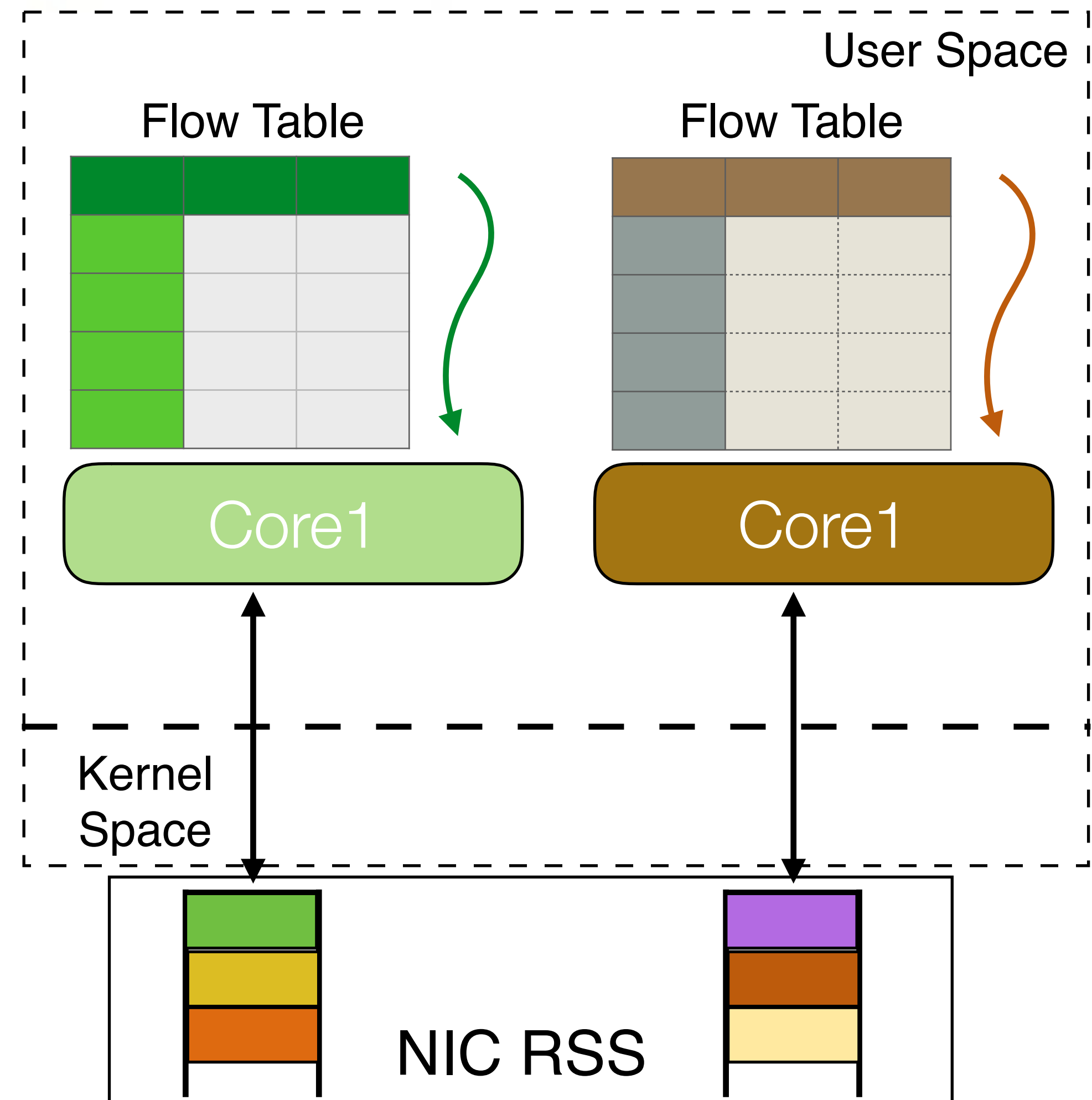
DPDK high network IO

Multi-threads

- RSS dispatch flows to cores

Lockless flow table

- Per-core flow table
- Cleanup on the same thread



Outline

- Background and Motivation
- EdgeBalance Design
- **Environment Setup and Evaluation**
- Conclusion

Experiment Setup

Cloumlab c220g5 specs

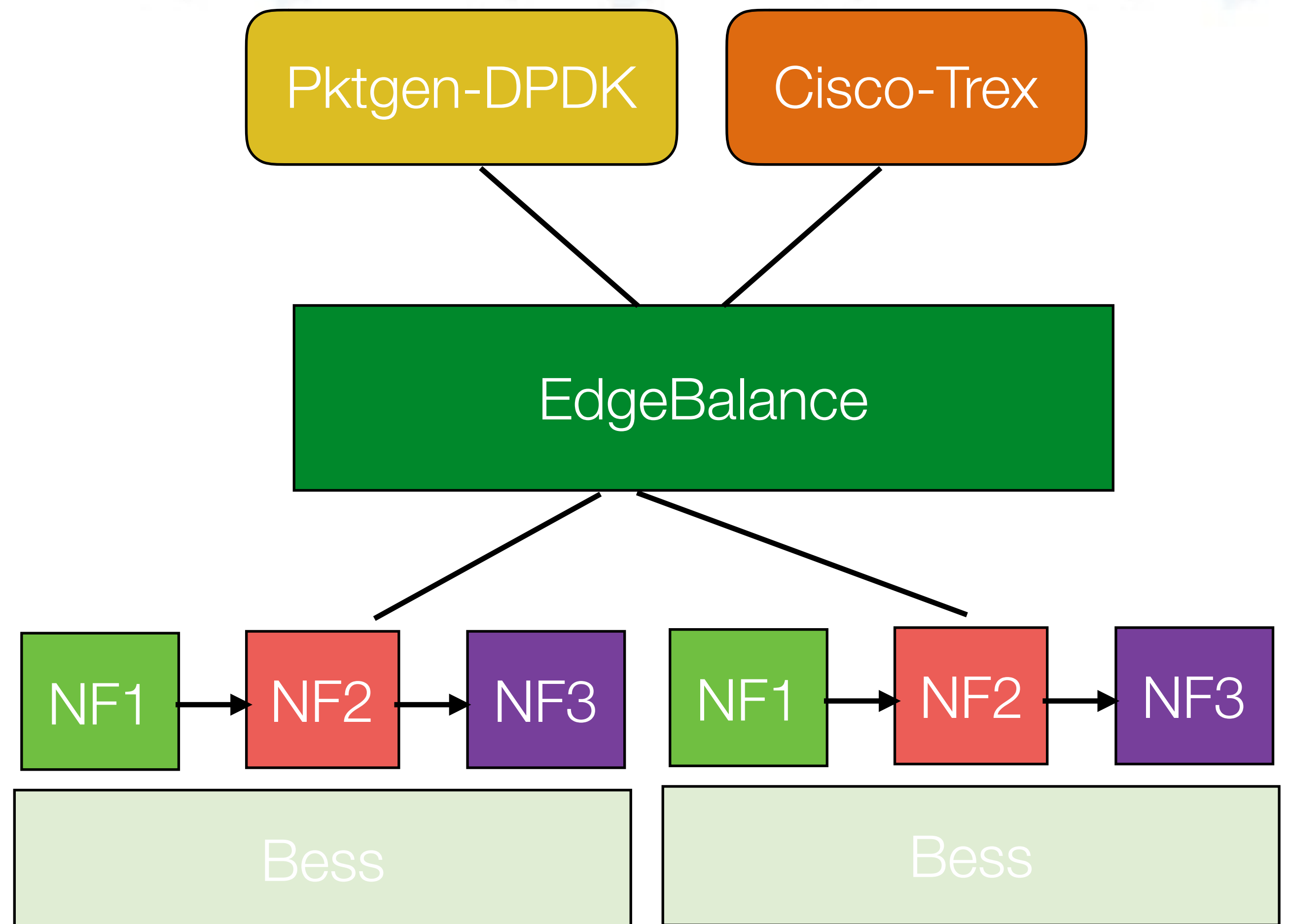
DPDK 18.02

Ubuntu 14.04

Packet generator

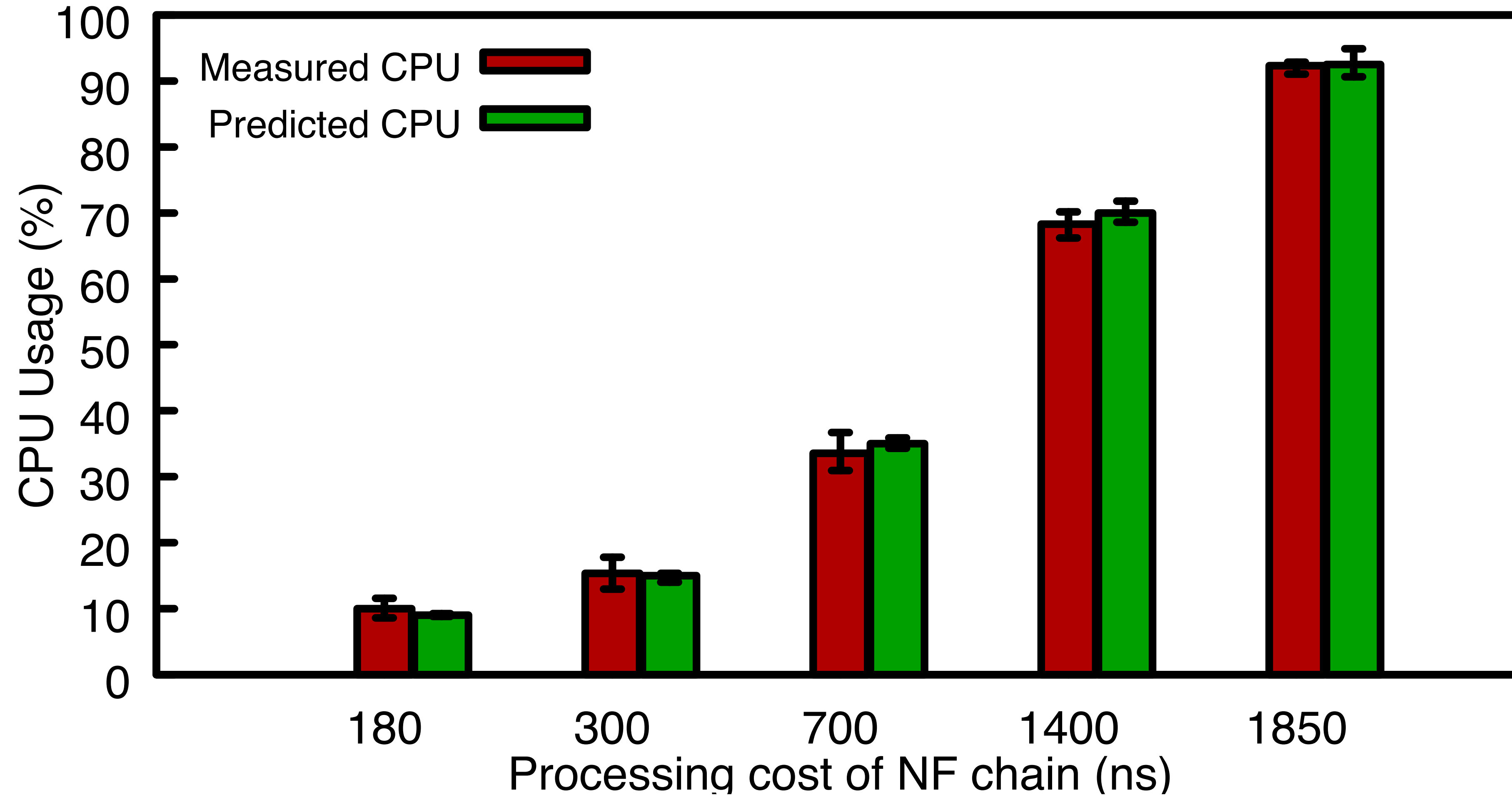
- Pktgen-DPDK
- Cisco Trex

Bess runs NFs



Evaluation: Prediction Accuracy

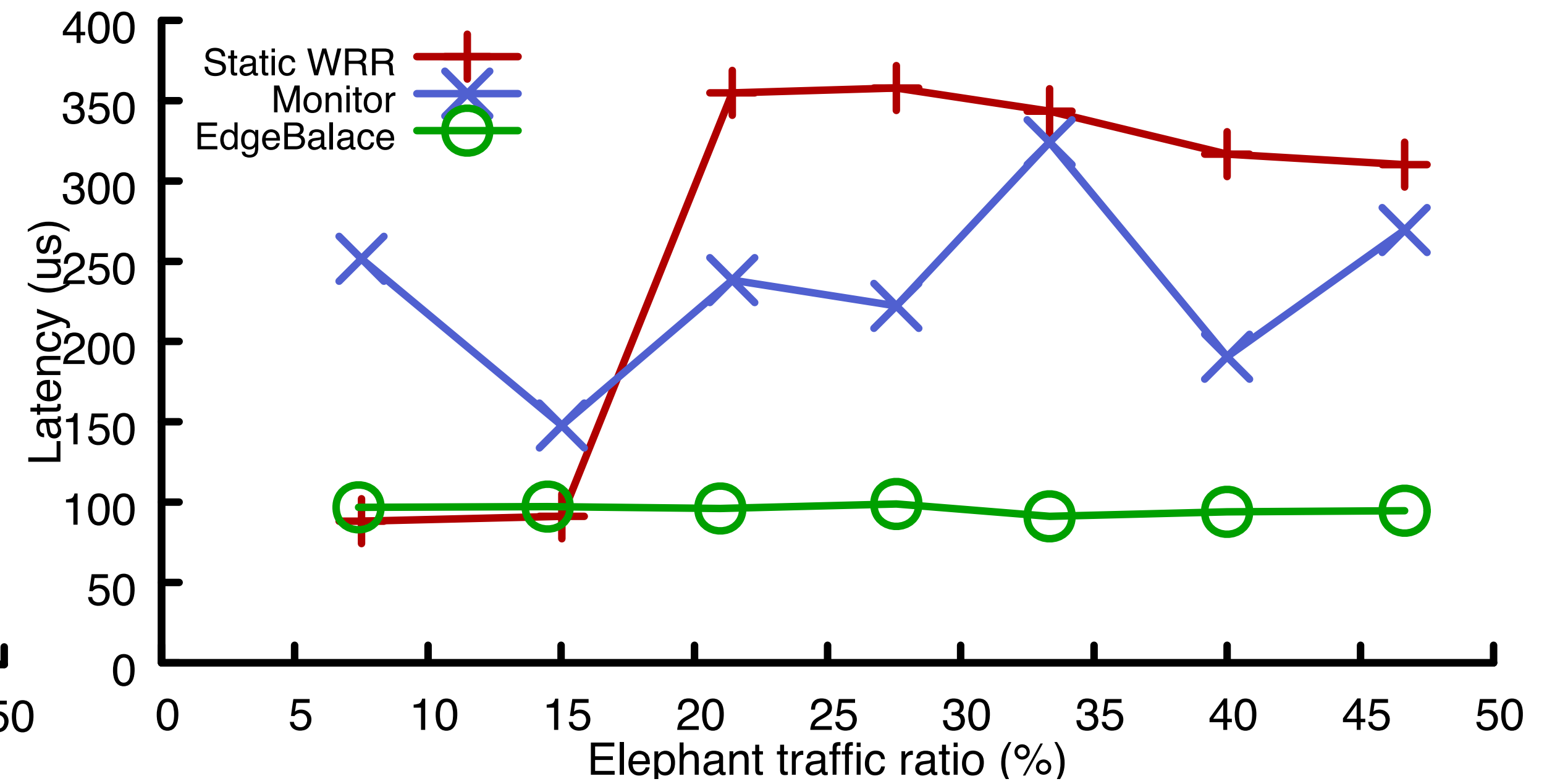
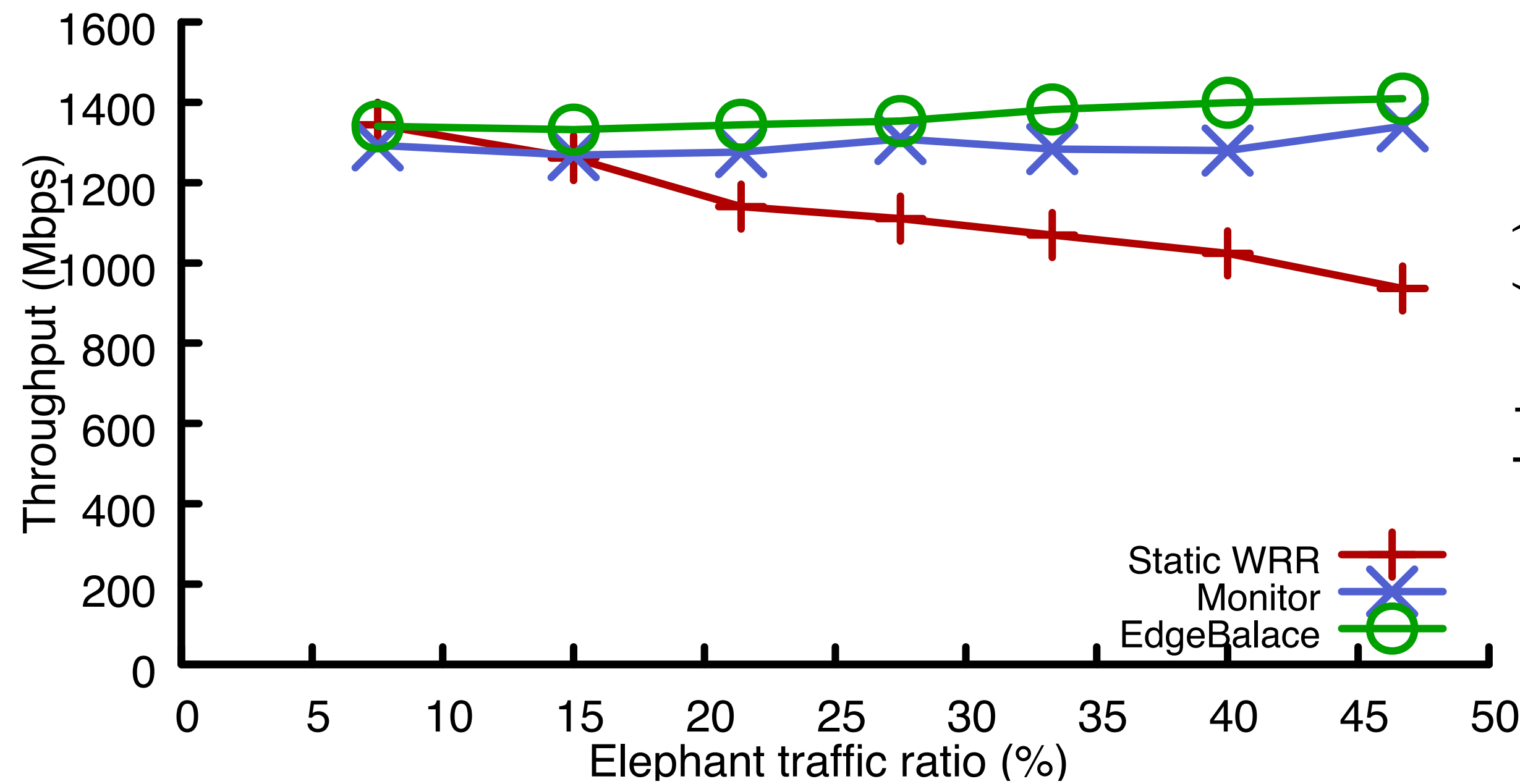
Accuracy of model-based CPU use prediction



Evaluation: Outperform

A workload with a mix of elephant and mice flows

EdgeBalance outperforms static and monitor-based LBs



Conclusions

EdgeBalance aims to balance the load for network service chains at edge

Network edge has unique challenges

- ➔ Limited resources
- ➔ Heterogeneity
- ➔ Affinity due to stateful network functions

Bidirectional flow affinity

- ➔ Forward flow and reverse flow go through the same set of NFs

Dynamic load balance

- ➔ Accurately predict CPU load
- ➔ Dynamically adjust allocation weight with PID controller to balance CPU load across hosts

wei.zhang.gb@gmail.com