MiddleNet: A High-Performance, Lightweight, Unified NFV and Middlebox Framework

Ziteng Zeng, Leslie Monis, ***Shixiong Qi***, K. K. Ramakrishnan

University of California, Riverside

Networked Systems Group

UC RIVERSIDE

Emerging Trend of NFV and Middlebox

Softwarization of networks

Purpose-built appliances → *Softwarized functions*

NFs

firewall

carrier-grade NAT

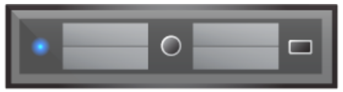
Middleboxes

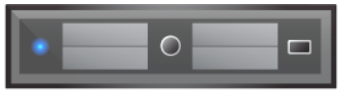
DPI

load balancer

NF1 | NF2 | Middleboxes

Virtualization Layer | Protocol stack





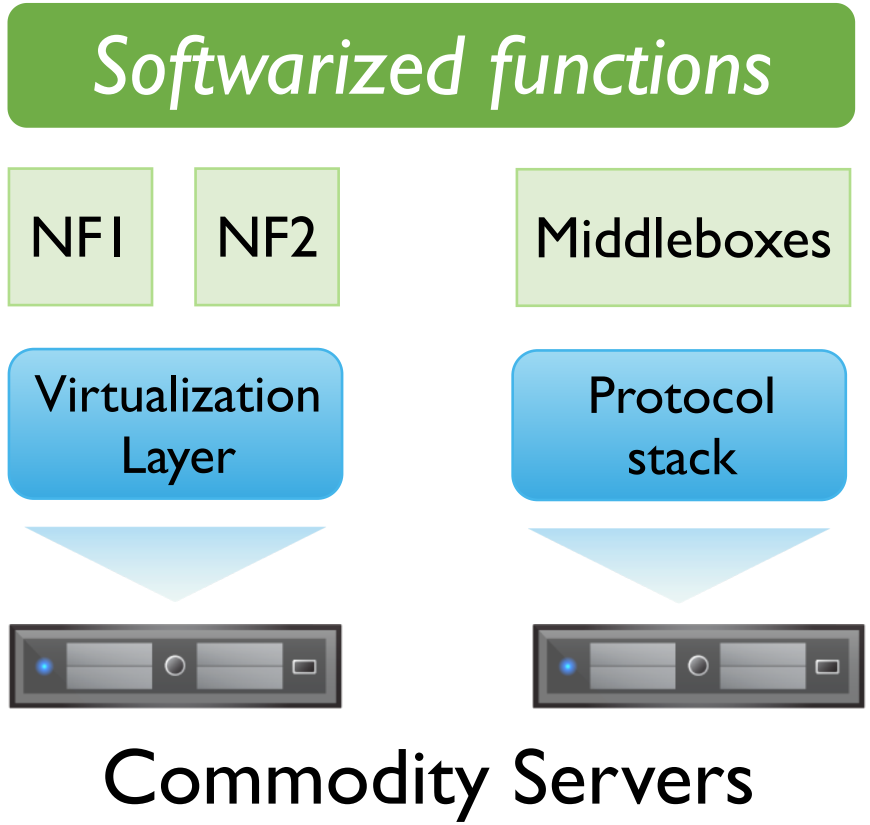
Commodity Servers

Networked Systems Group
UC RIVERSIDE

Emerging Trend of NFV and Middlebox

Distinction between the NFV and Middlebox

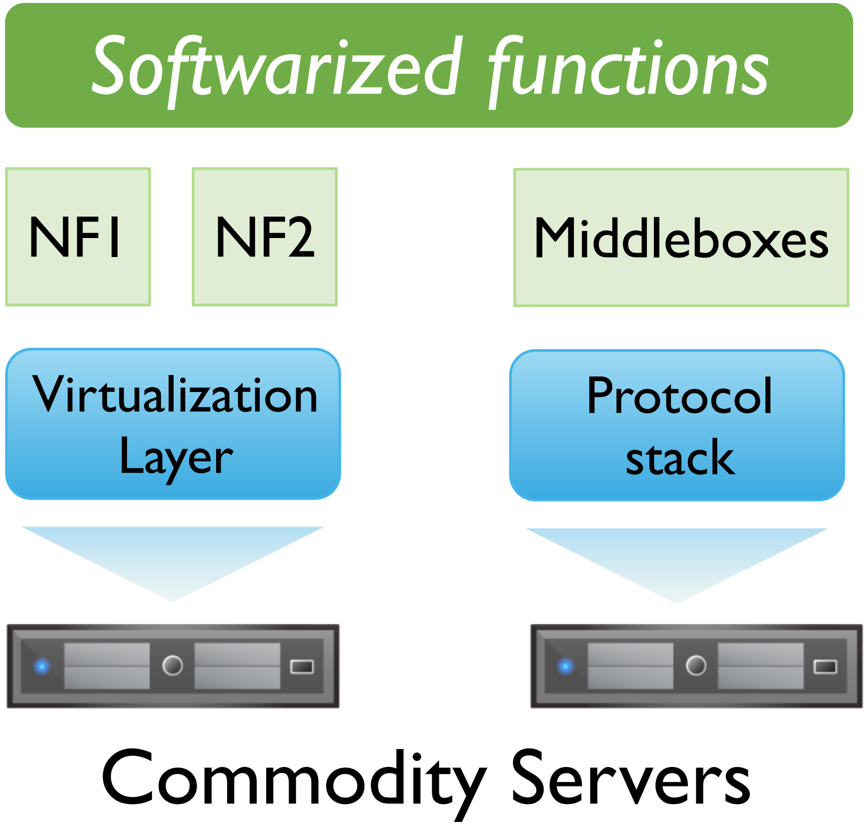
	NFV	Middlebox
Operating layer	L2/L3	L4/L7
Requirements	Full line rate	Full functionality
Dependencies	Kernel-bypass, zero-copy	Kernel protocol stack
Framework example	OpenNetVM, ClickOS, NetMap	mOS, microboxes, StackMap



Emerging Trend of NFV and Middlebox

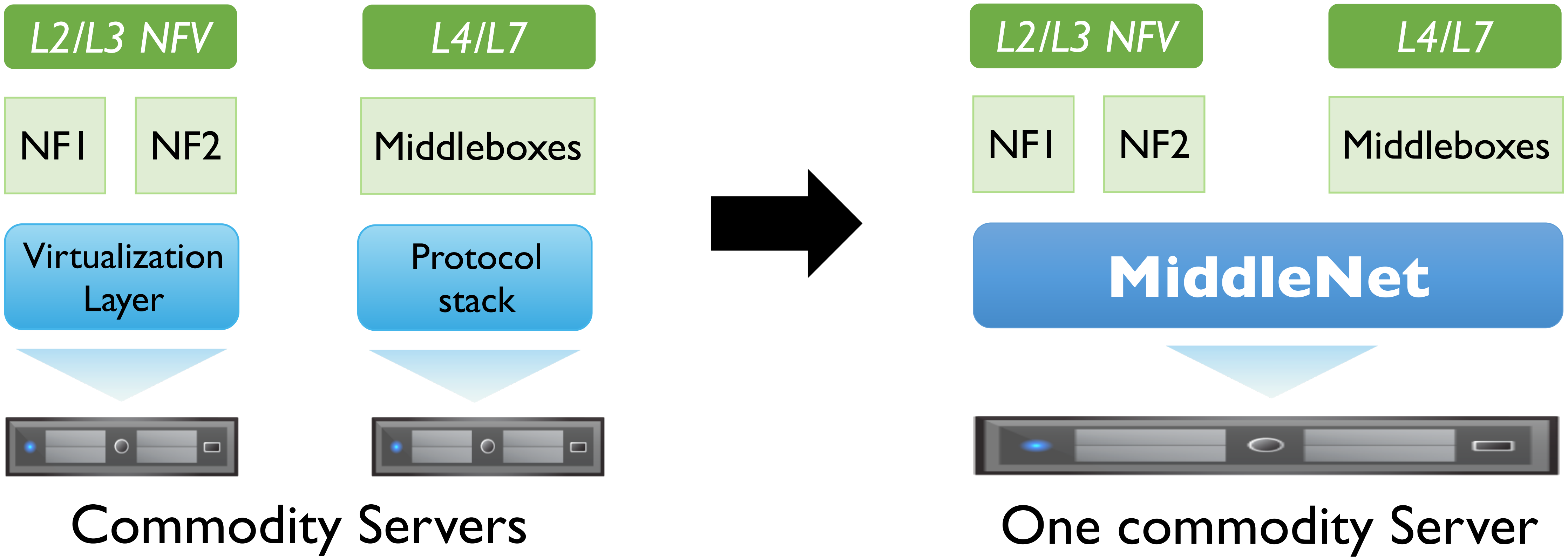
Distinction between the NFV and Middlebox

🙁 L2/L3 NFs and L4/L7 middleboxes continue to be handled by **distinct** platforms on **different** nodes.



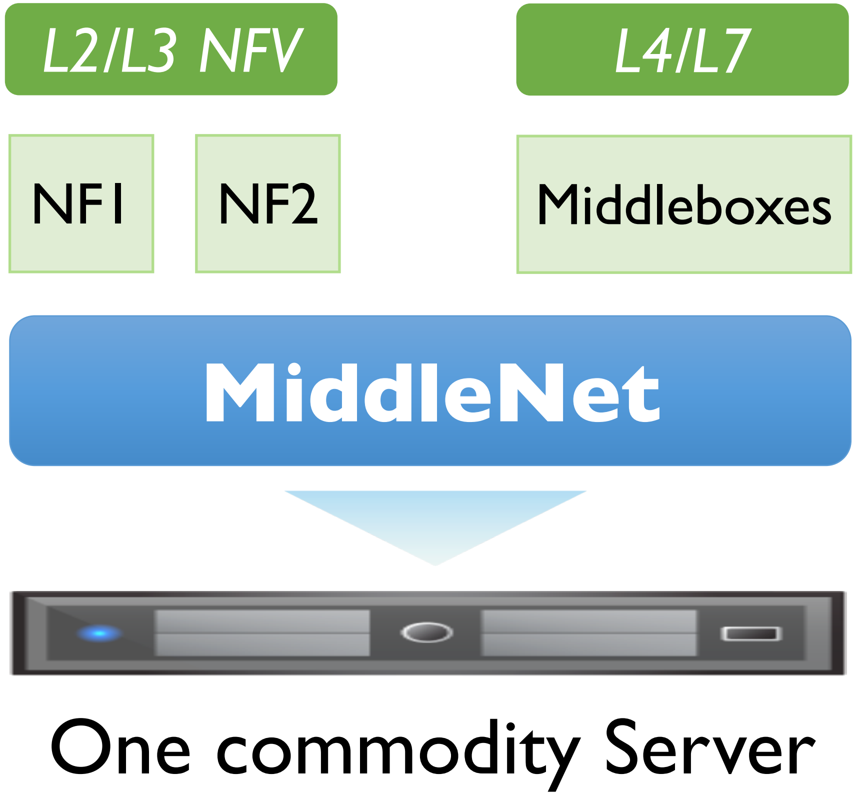
Networked Systems Group
UC RIVERSIDE

Unifying L2/L3 NFV and L4/L7 Middlebox



Networked Systems Group
UC RIVERSIDE

Unifying L2/L3 NFV and L4/L7 Middlebox



- What is the best way to build L2/L3 NFV?
- What is the best way to build L4/L7 Middleboxes?
- How to build a unified environment without performance loss?

L2/L3 NFV design: Options for MiddleNet

- Features of L2/L3 NFV
  - less emphasis on having a full-function protocol stack
  - **bump-in-the-wire** capability
- **Kernel-bypass & Zero-copy** packet delivery
  - *Option-1*: AF_XDP[1] and SKMSG[4] in **eBPF**
    - Naturally supported in Linux
    - Event-driven but has receive livelock[2] issue
  - *Option-2*: DPDK's PMD and RTE ring[3]
    - High Performance but Costly in Resources
    - Typically cannot use Kernel Protocol Stack

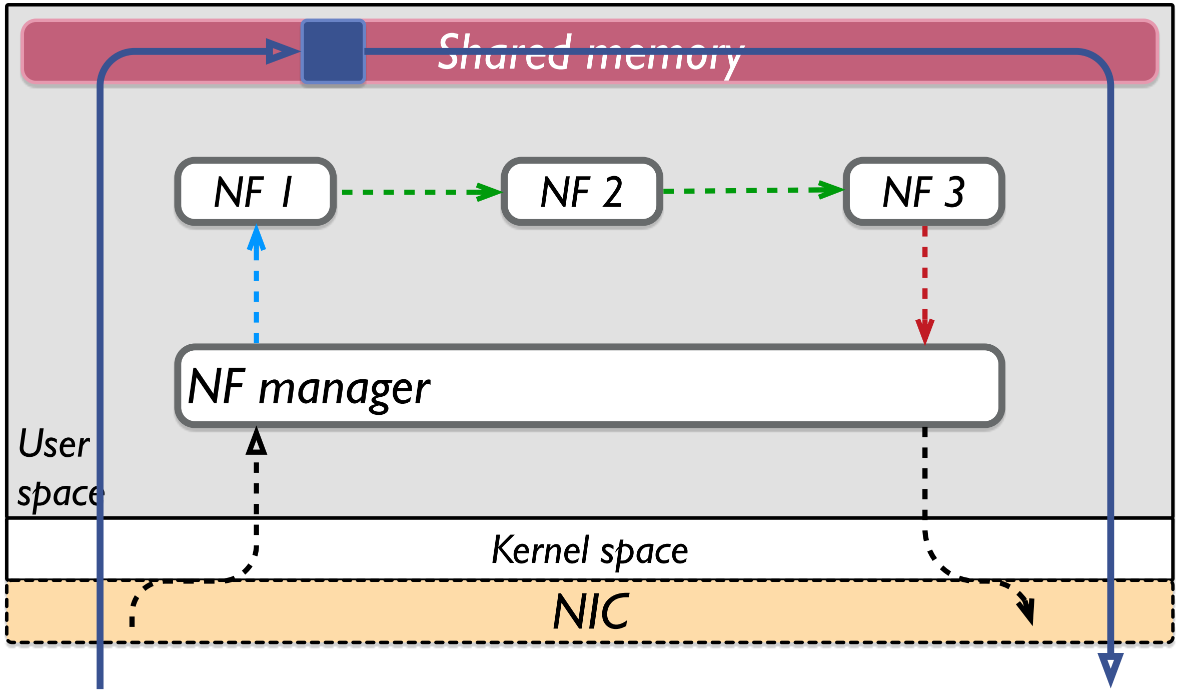
What should we use in **L2/L3** MiddleNet?

[1] AF_XDP, https://www.kernel.org/doc/html/latest/networking/af xdp.html,.
[2] J. C. Mogul and K. K. Ramakrishnan, "Eliminating receive livelock in an interrupt-driven kernel," ACM Transactions on Computer Systems, 1997.
[3] W. Zhang et al, "Opennetvm: A platform for high performance network service chains," HotMiddlebox '16.
[4] "BPF_PROG_TYPE_SK_MSG", https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/assembly_understanding-the-ebpf-features-in-rhel_configuring-and-managing-networking

L2/L3 NFV design: Options for MiddleNet

Common design shared between DPDK and eBPF

- NF manager
  - *Mediate the packet delivery* ***to*** *and* ***from*** *the NIC*
  - ***Notify*** *the NF to process packets*
- Chained Functionality
  - *Functions are often* ***chained***
  - *Need high speed inter-function communication*
  - ***Zero-copy*** *packet delivery within the chain*
  - *Lock-free ownership transfer*
    - *Multiple readers,* ***single*** *writer*

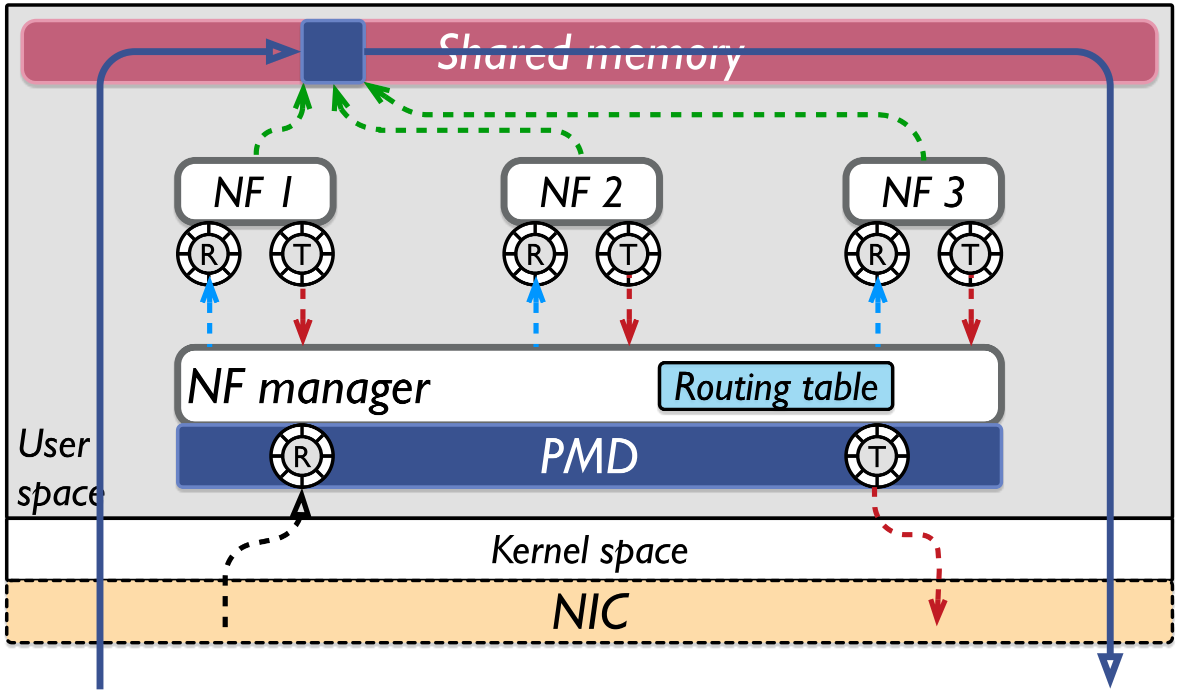


Networked Systems Group
UC RIVERSIDE

L2/L3 NFV design: Options for MiddleNet

DPDK-based solution

- Kernel bypass - DPDK
  - *Poll Mode Driver (PMD)*
    - Constantly poll the RX ring to retrieve arriving packets
- Messaging within the chain - DPDK
  - *DPDK's RTE rings (RX/TX)*
    - The NF **polls** its RX ring to retrieve arriving packets
- Great performance but occupies CPU cores
  - NFVnice[1] can help mitigate these overheads by sharing a CPU core across multiple NFs



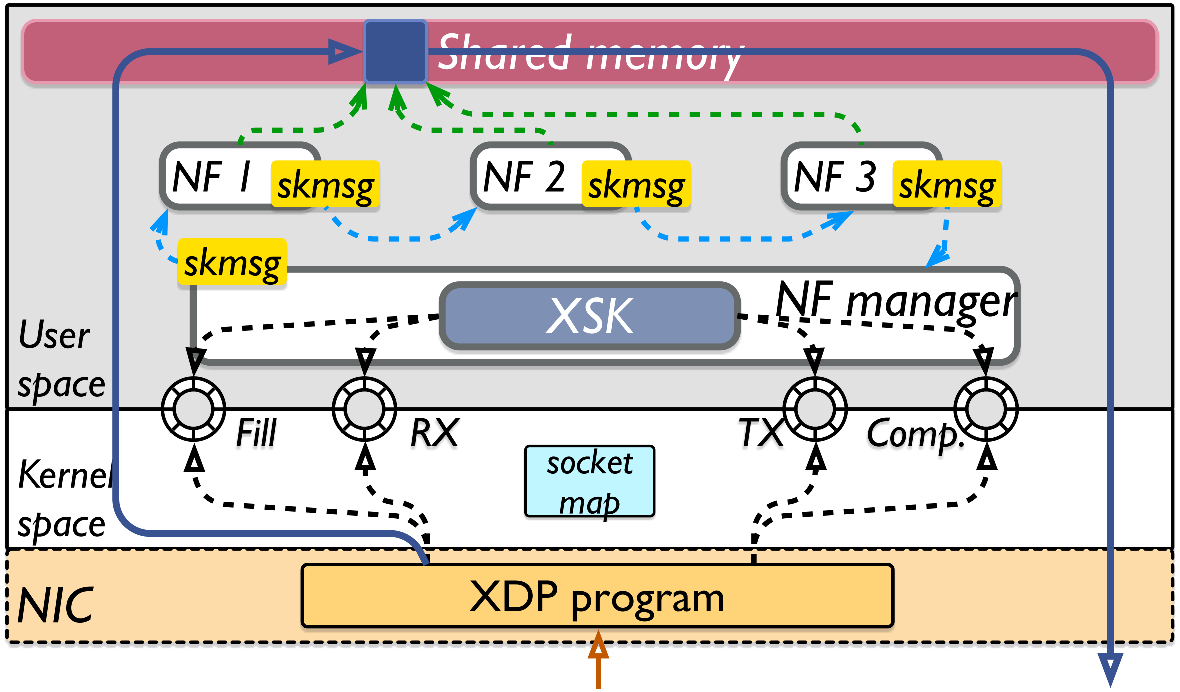
Networked Systems Group
UC RIVERSIDE

[1] S. G. Kulkarni et al, "Nfvnice: Dynamic backpressure and scheduling for nfv service chains," IEEE/ACM Transactions on Networking, 2020.

L2/L3 NFV design: Options for MiddleNet

eBPF-based solution

- Kernel bypass - eBPF
  - *AF_XDP socket (XSK)*
    - interact with the kernel to handle RX and TX from/to the NIC
    - Triggered by XDP program in the NIC
- Message channel - eBPF
  - *eBPF's socket message (SKMSG)*
    - eBPF's socket map for routing
    - Packet desc. delivery done by SKMSG
- *Event-driven and load-proportional*
  - But we need to overcome receive livelock issue

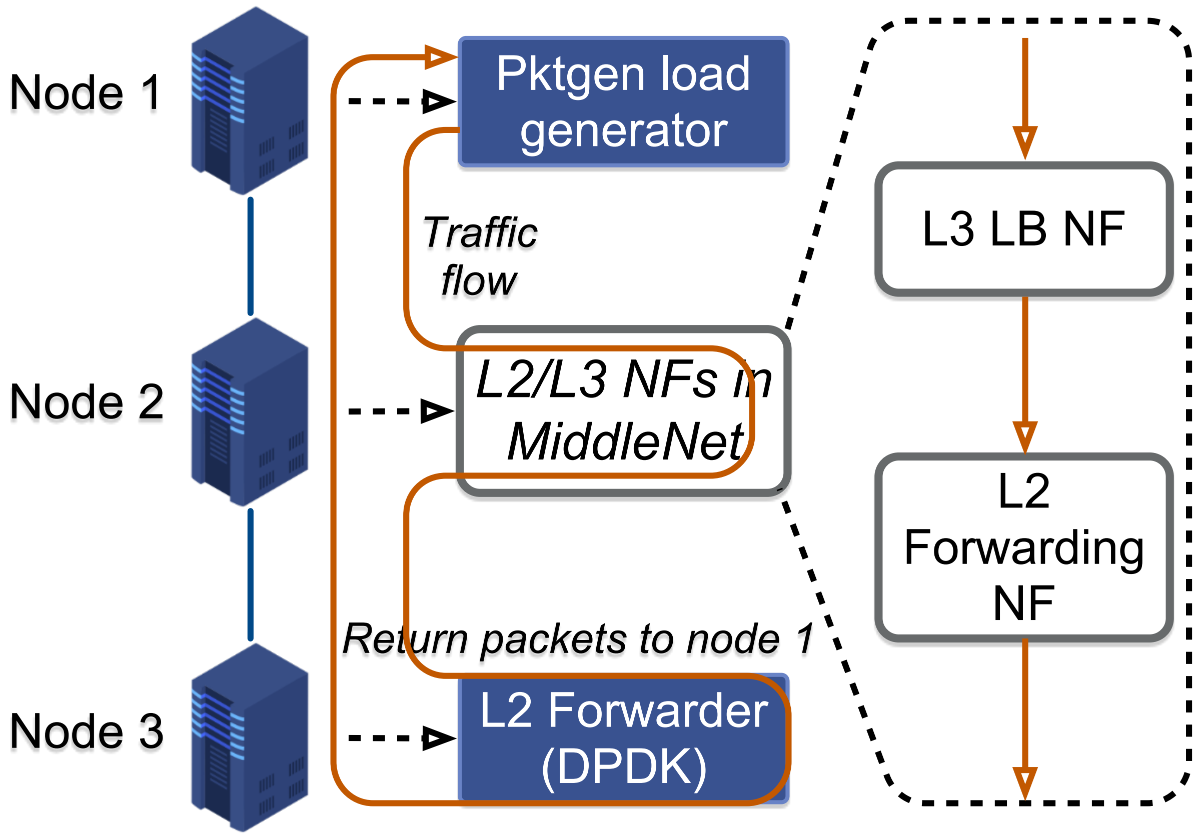


Networked Systems Group
UC RIVERSIDE

L2/L3 NFV design: Options for MiddleNet

Performance Evaluation of Alternatives

- 1st node
  - Pktgen load generator
- 2nd node (MiddleNet)
  - L3 LB function
    - Updates the IP address
  - L2 forwarding function
    - Updates the MAC address
- 3rd node
  - return the packets back to the 1st node

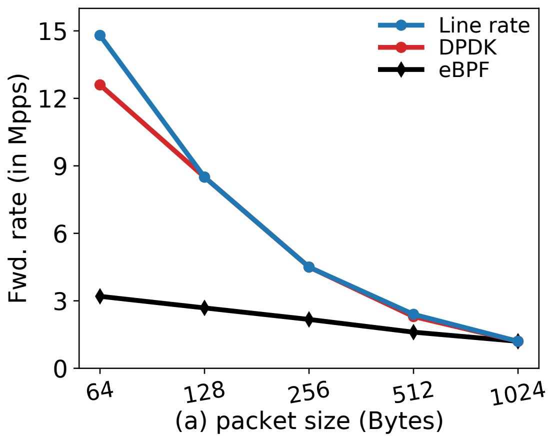


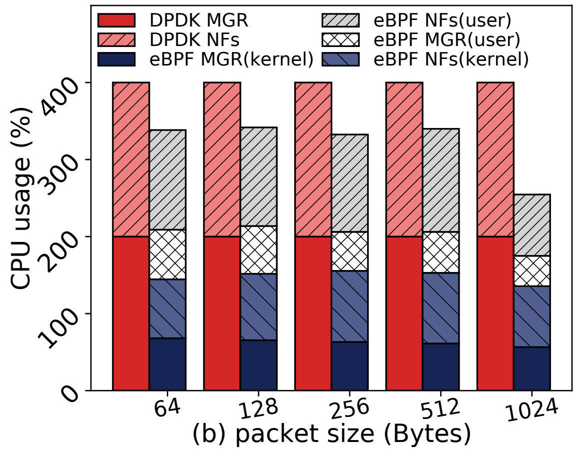
- NFS Cloudlab Server
  - 40-core CPU
  - 192 GB memory
  - 10 Gbps NIC

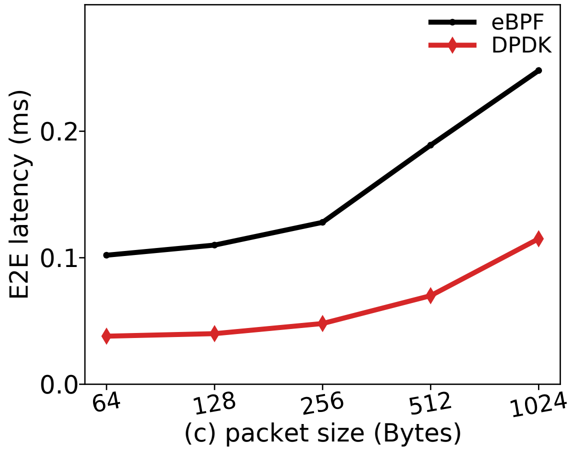
Networked Systems Group
UC RIVERSIDE

L2/L3 NFV design: Options for MiddleNet

Performance Evaluation



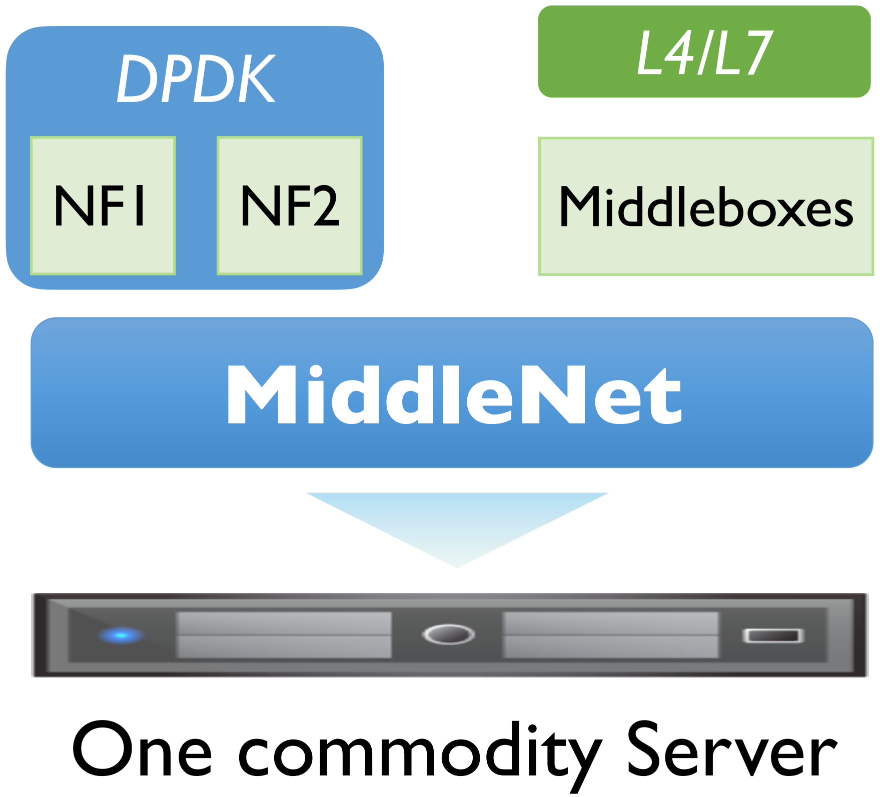




- MLFR(Maximum Loss Free Rate)
  - DPDK: Achieves almost line rate for different packet sizes.
  - eBPF: Far less than DPDK
- CPU usage at MLFR
  - DPDK: Constant high CPU usage
  - eBPF: Most CPU time spent in **kernel** to handle interrupts
- End-to-end latency
  - DPDK achieves 2× improvement compared to eBPF

Networked Systems Group
UC RIVERSIDE

Unifying L2/L3 NFV and L4/L7 Middlebox

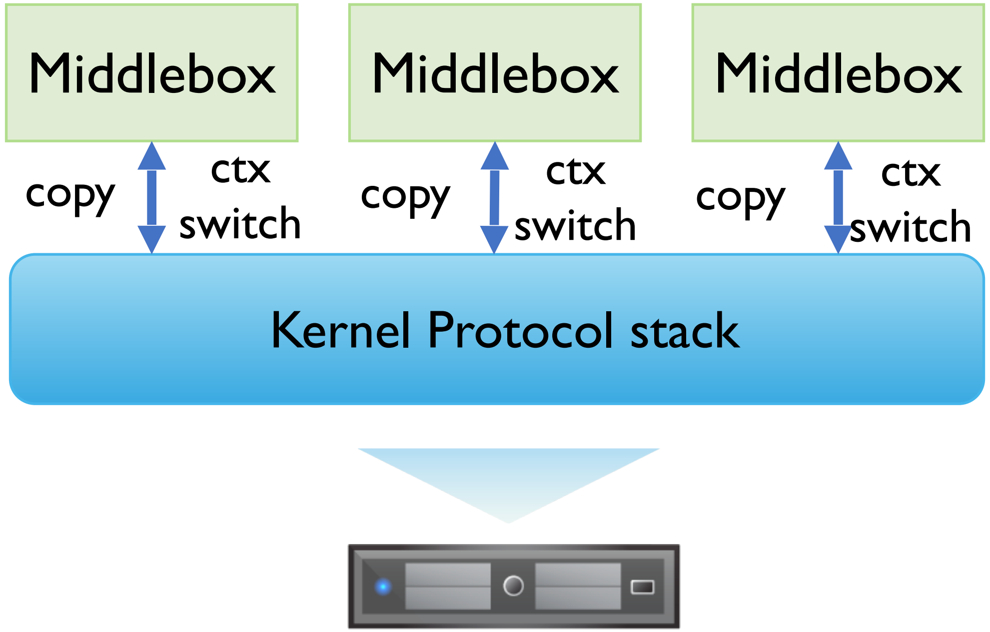


- What is the best way to build L2/L3 NFV?
  - ***DPDK***
- What is the best way to build L4/L7 Middleboxes?
- How to build a unified environment without performance loss?

Networked Systems Group
UC RIVERSIDE

L4/L7 Middlebox design in MiddleNet

- Features of L4/L7 Middlebox
  - depend on a full-function protocol stack
- User-space protocol stack
  - Combined with kernel-bypass
    - High performance
  - mTCP[1], Microboxes[2]
    - Protocol support is still not complete
- Kernel protocol stack
  - full-function, robust and proven
  - but incurs **data copy** & **context switch**



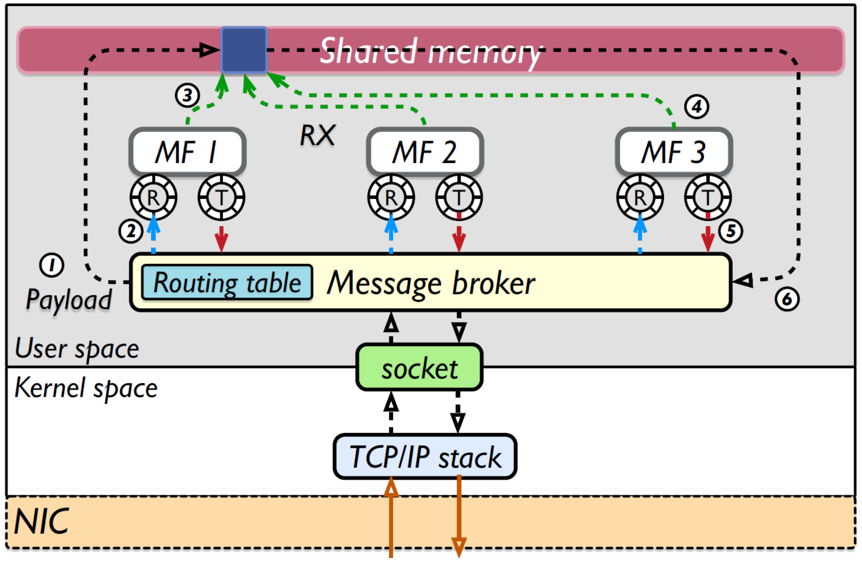
Overheads accumulate with a chain of middleboxes

L4/L7 Middlebox design in MiddleNet

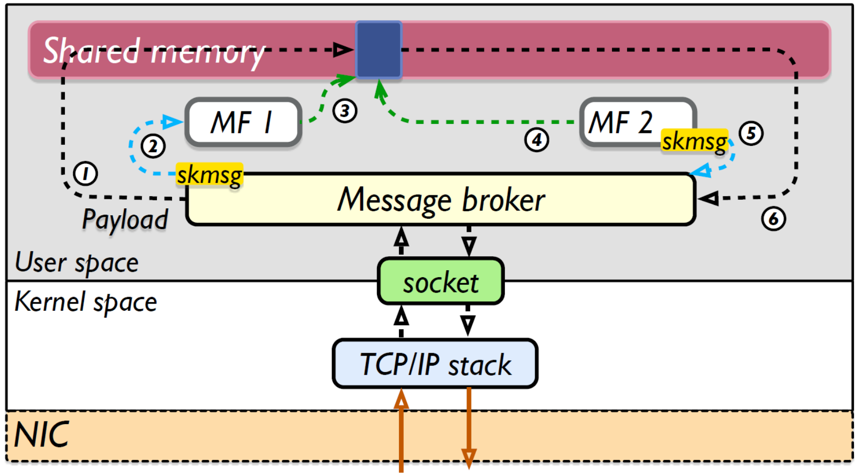
Optimization on a chain of middleboxes

- #1: Consolidate kernel stack processing
  - **One** data copy & context switch - whether for DPDK or eBPF alternative
- #2: Zero-copy function chain communication
  - *Just like the L2/L3 NFV design option*
- Adaptive batching for SKMSG
  - *Read multiple (up to a limit) packet descriptors available in the socket buffer at once*
    - Reduce the total number of interrupts through batching
    - Mitigate overload behavior
- Designs: DPDK, eBPF, monolithic kernel (NGINX) baseline

DPDK:



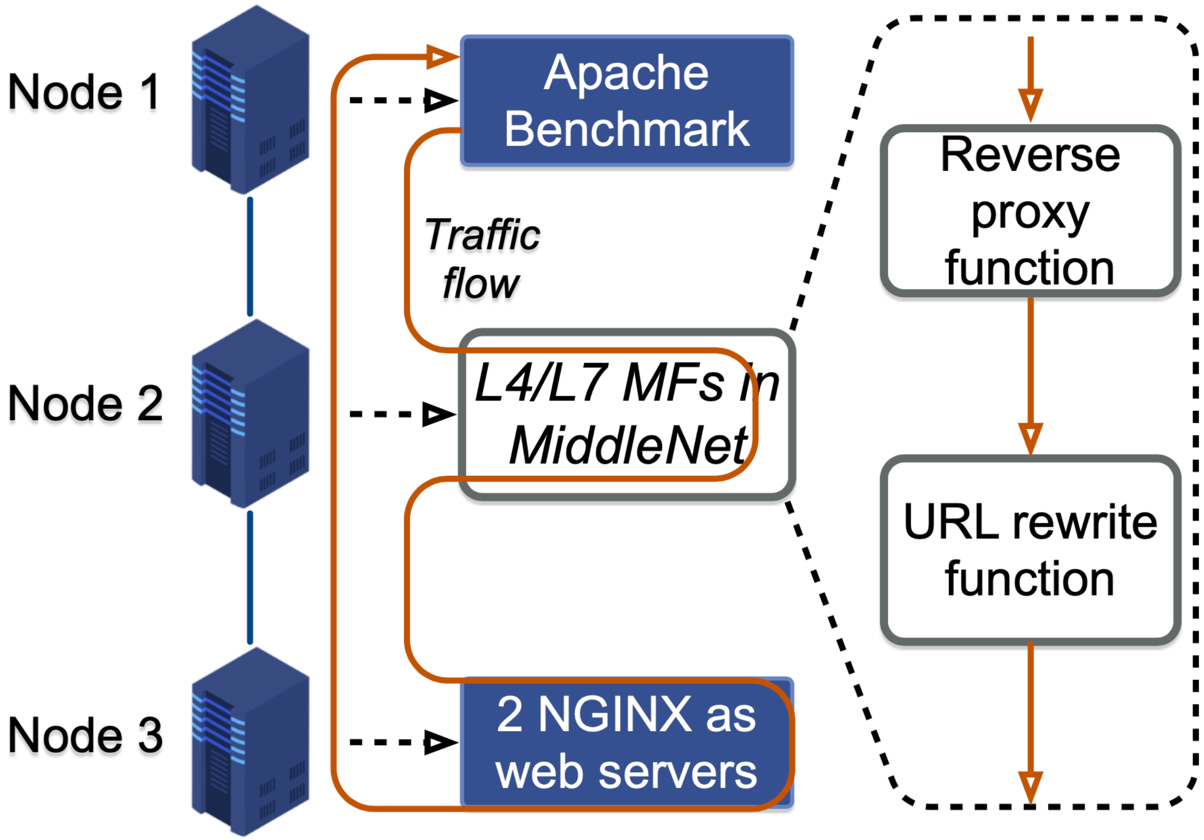
eBPF:



L4/L7 Middlebox design in MiddleNet

Performance Evaluation of Alternatives

- 1st node
  - Apache Benchmark
- 2nd node (L4/L7 MiddleNet)
  - Reverse proxy function
    - Balances the load between the 2 NGINX web server backends on 3rd node
  - URL rewrite function
    - Helps to perform redirection for static websites
- 3rd node
  - 2 NGINX web servers

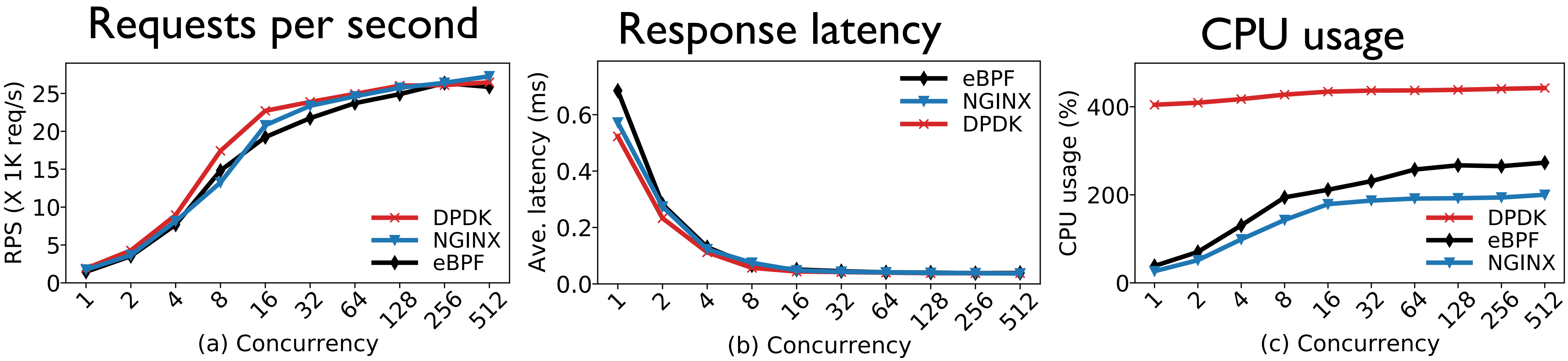


- NFS Cloudlab Server
  - 40-core CPU
  - 192 GB memory
  - 10 Gbps NIC

Networked Systems Group
UC RIVERSIDE

L4/L7 Middlebox design in MiddleNet

Performance Evaluation of Alternatives



#1: At light loads, DPDK has the **lowest** response latency and **higher** RPS, but the CPU usage is **high**

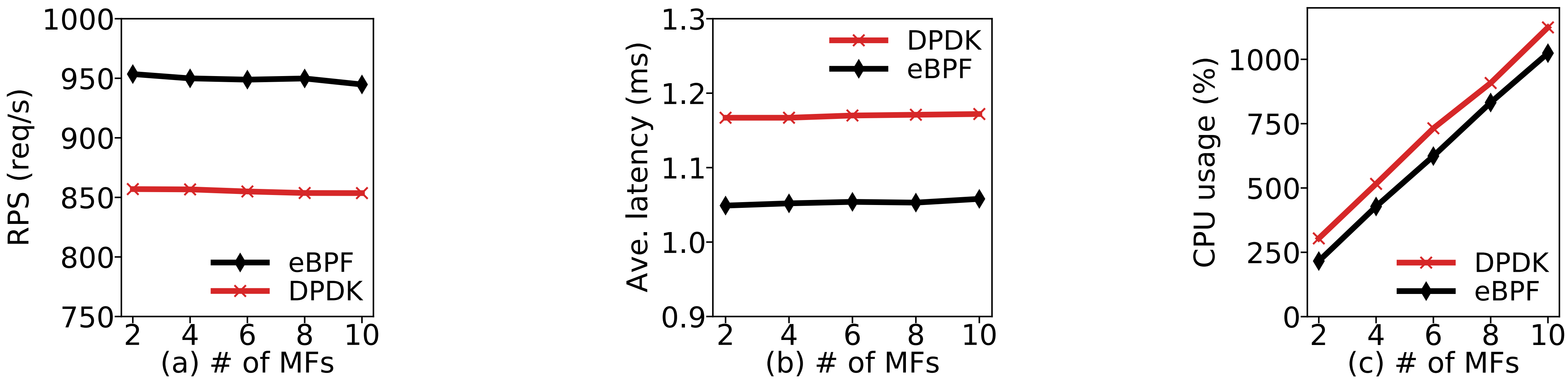
#2: At heavy loads, eBPF's adaptive batching takes effect: **performance close** to others

#3: MiddleNet has more flexibility and resiliency than NGINX's monolithic implementation

L4/L7 Middlebox design in MiddleNet

* Each MF runs a prime number generation function based on the sieve-of-Atkin algorithm

Performance Evaluation with CPU-intensive middleboxes (MFs)

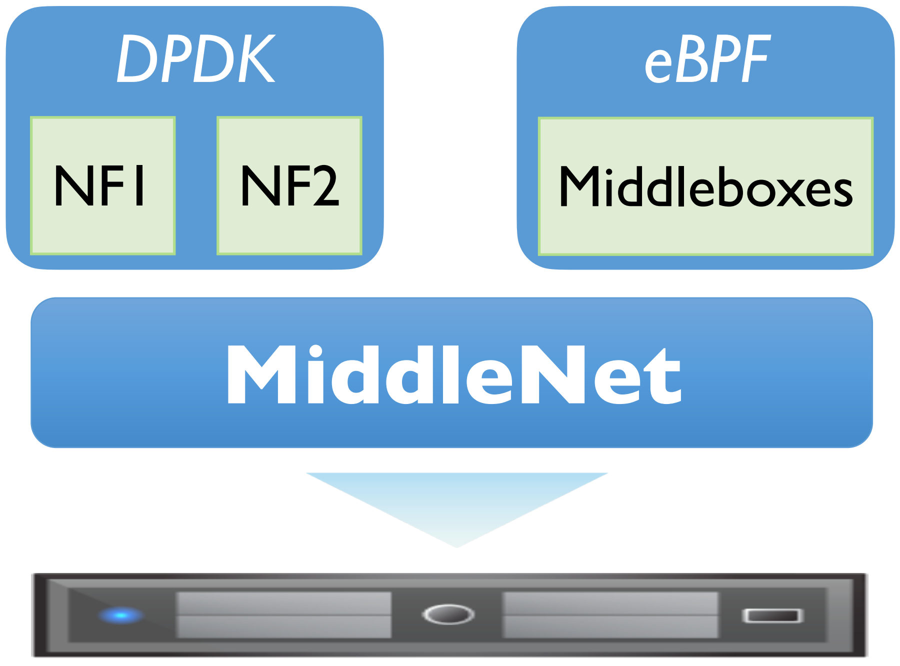


#1: Both DPDK and eBPF show **good scalability** as the chain scales

#2: eBPF has **better** performance and **less** CPU usage with CPU-intensive middleboxes

#3: DPDK's polling **contends** with CPU-intensive middleboxes

Unifying L2/L3 NFV and L4/L7 Middlebox



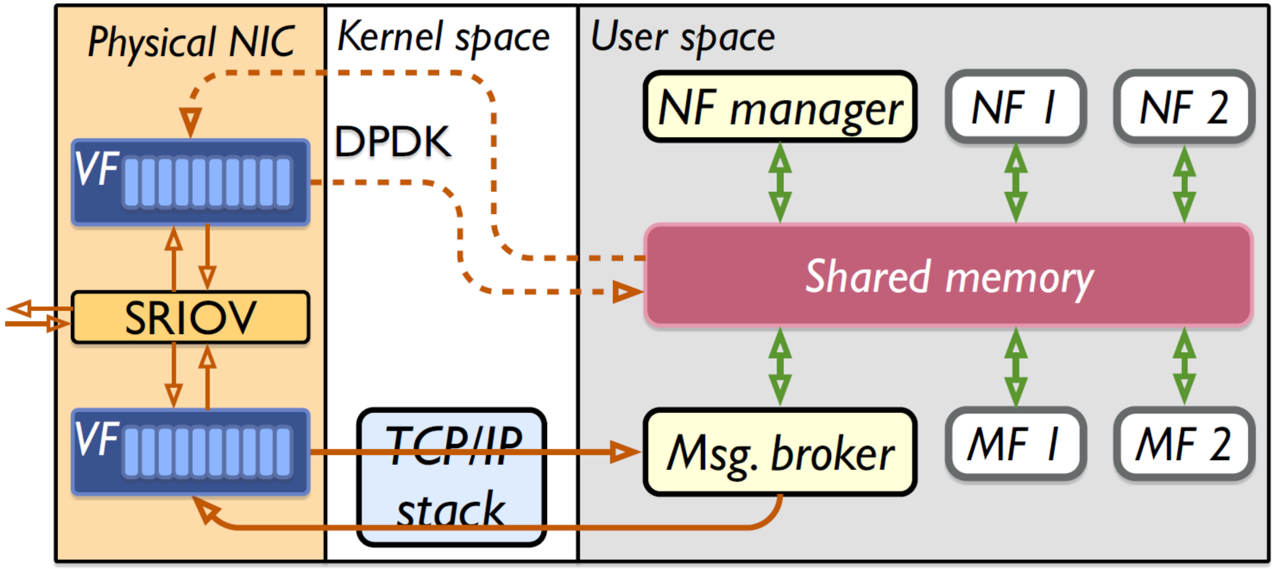
One commodity Server

- What is the best way to build L2/L3 NFV?
  - ***DPDK***
- What is the best way to build L4/L7 Middleboxes?
  - ***eBPF***
- How to build a unified environment without performance loss?

Networked Systems Group
UC RIVERSIDE

A Unified Design Based on SR-IOV

- Virtual Functions (VFs) on the NIC
  - Share NIC among VFs
  - VF has direct acess to physical resources
  - Separate VFs for L2/L3 and L4/L7 MiddleNet
    - Dedicted queue for each VF
- Flow Bifurcation mechanism[1]
  - Available on SR-IOV NIC
  - Splitting the traffic within the NIC
  - State-dependent flow processing
    - Packet classification based on IP 5 tuples (source/destination IPs, source/destination ports, protocol)



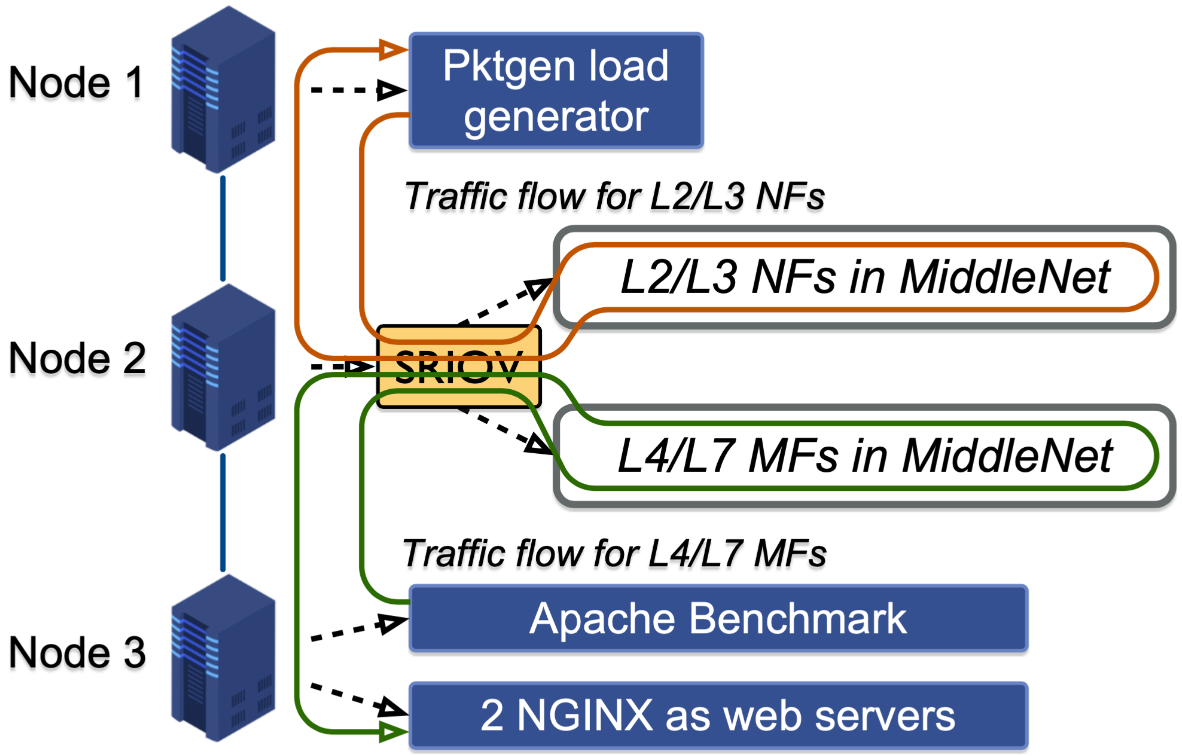
Networked Systems Group
UC RIVERSIDE

[1] DPDK Project, "Flow Bifurcation How-to Guide," https://doc.dpdk.org/guides-19.02/howto/flow bifurcation.html

A Unified Design Based on SR-IOV

Performance Evaluation

- 1st node (mainly for L2/L3)
  - Pktgen load generator
    - sending rate is kept at MLFR
- 2nd node
  - L4/L7 MiddleNet (eBPF)
  - L2/L3 MiddleNet (DPDK)
- 3rd node (mainly for L4/L7)
  - Apache Benchmark (concurrency: 256)
  - 2 NGINX web servers

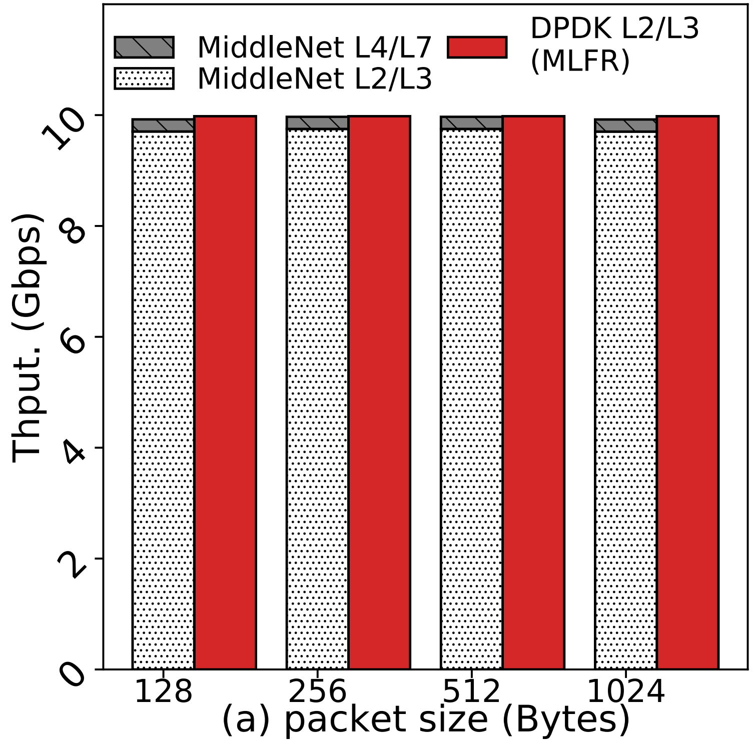


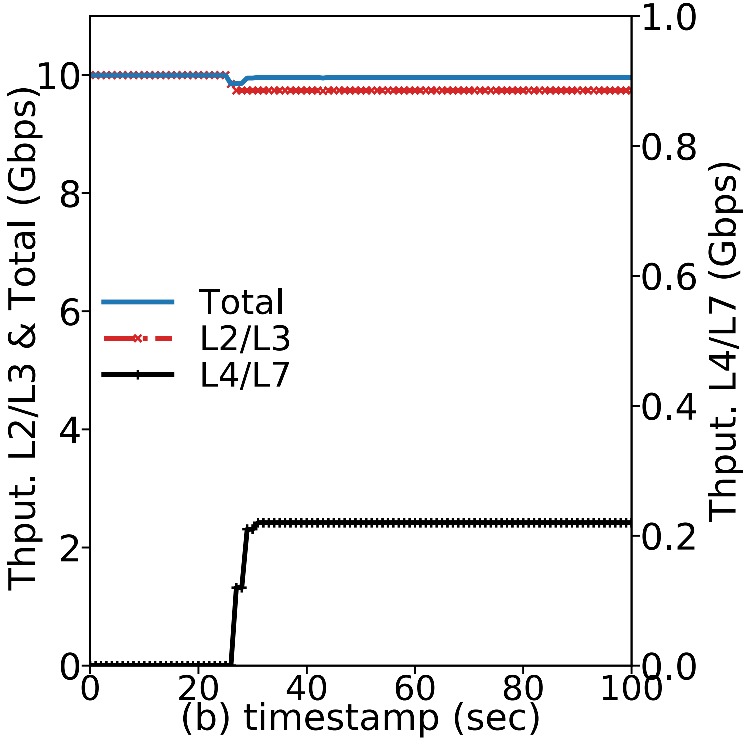
- NFS Cloudlab Server
  - 40-core CPU
  - 192 GB memory
  - 10 Gbps NIC

Networked Systems Group
UC RIVERSIDE

A Unified Design Based on SR-IOV

Performance Evaluation





- The aggregate throughput is close to line rate
  - **negligible** performance loss with SR-IOV

Networked Systems Group UC RIVERSIDE

Conclusion

- MiddleNet Unifies L2/L3 NFV with DPDK & L4/L7 middleboxes with eBPF
  - Best of each world: DPDK's high performance & eBPF's resource efficiency
- MiddleNet leverages *shared memory* to support high performance
  - High performance, full function L2/L3 and L4/L7 function chains
- MiddleNet creates a unified environment with SR-IOV
  - negligible performance loss
  - Support both L2/L3 NFV and L4/L7 middlebox on the same node

Networked Systems Group
UC RIVERSIDE