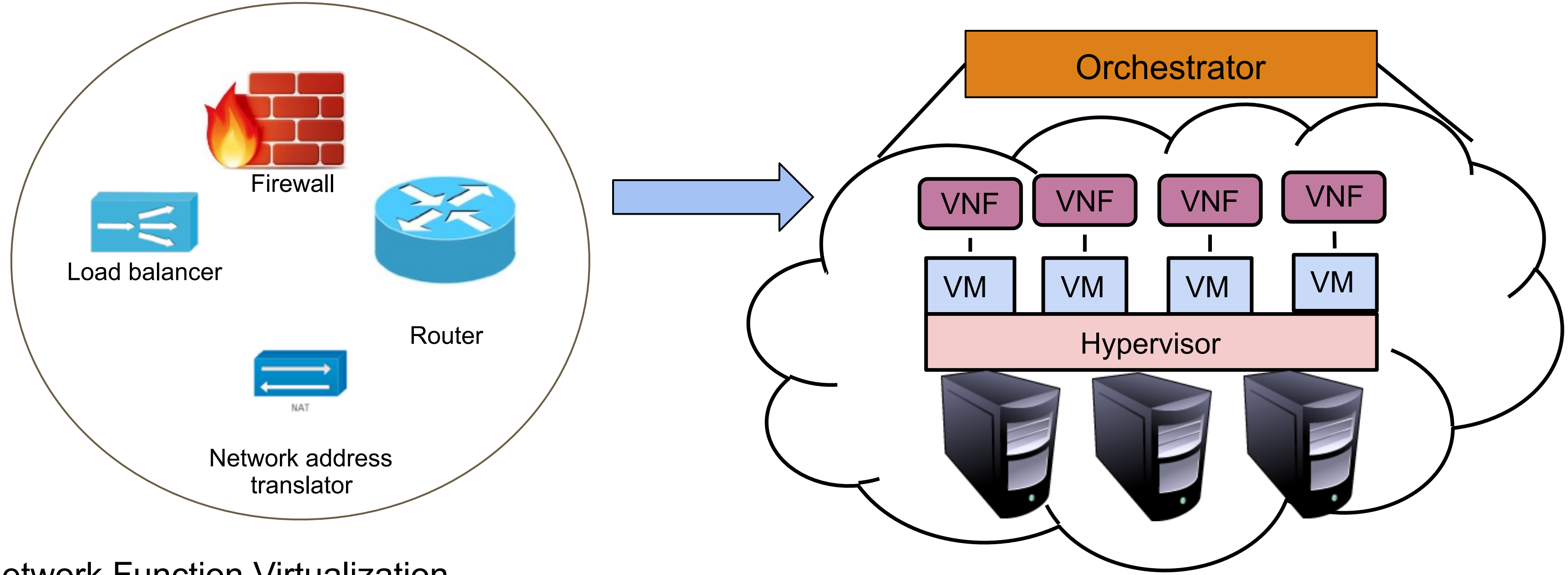
libVNF: building VNFs made easy

Priyanka Naik, Akash Kanase, Trishal Patel, Mythili Vutukuru

Dept. of Computer Science and Engineering
Indian Institute of Technology, Bombay

SoCC'18
11th October, 2018

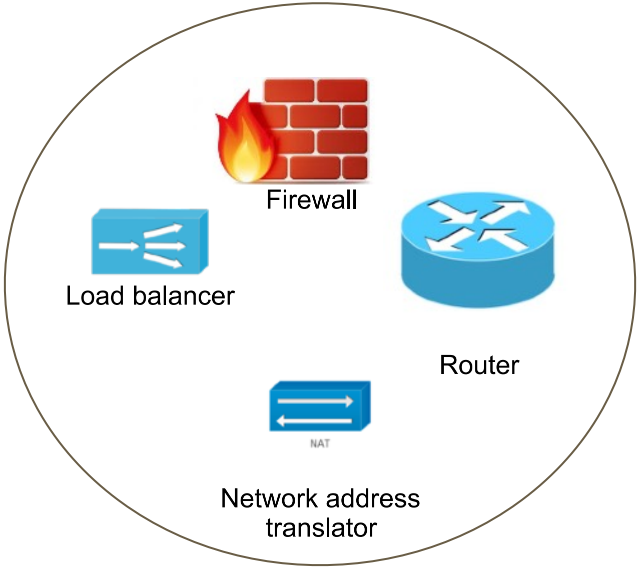
NFV ecosystem

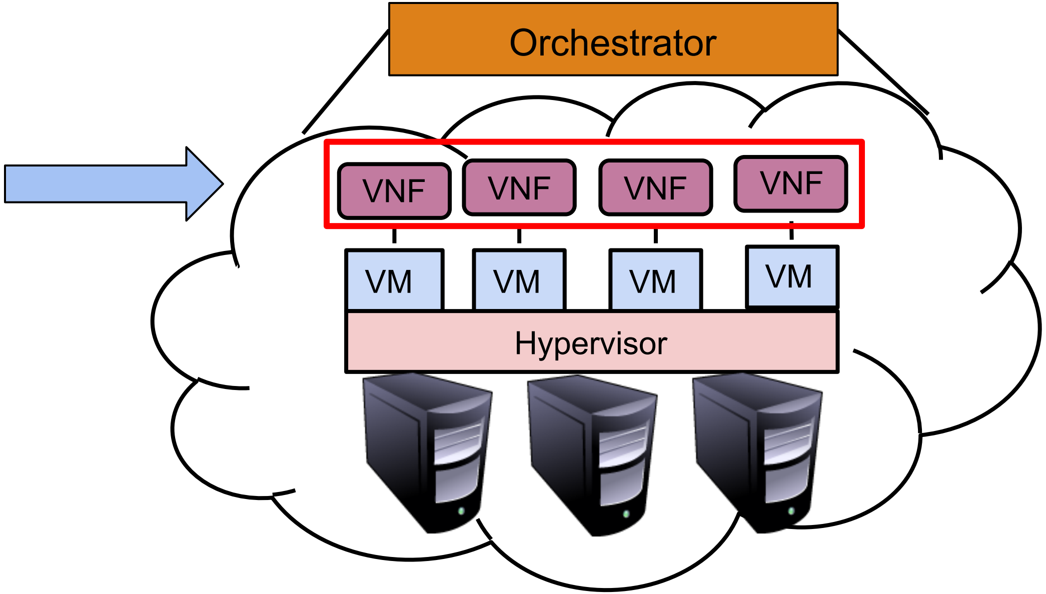


NFV: Network Function Virtualization
VNF: Virtual Network Function

NFV ecosystem

- Will they give good performance ?
- **Is it easy to build them?**





NFV: Network Function Virtualization
VNF: Virtual Network Function

How to build VNF?

VNF code developed by VNF developer

How to build VNF?

VNF code developed by VNF developer

38% EPC code → read/write packets

CORD Intel EPC: https://gerrit.opencord.org/ngic

How to build VNF?

VNF code developed by VNF developer

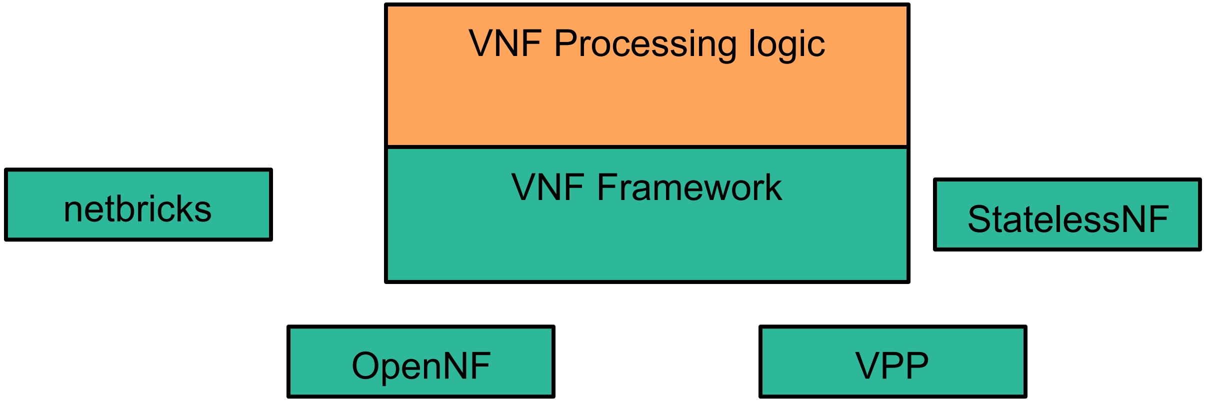
38% EPC code → read/write packets

VNF Processing logic

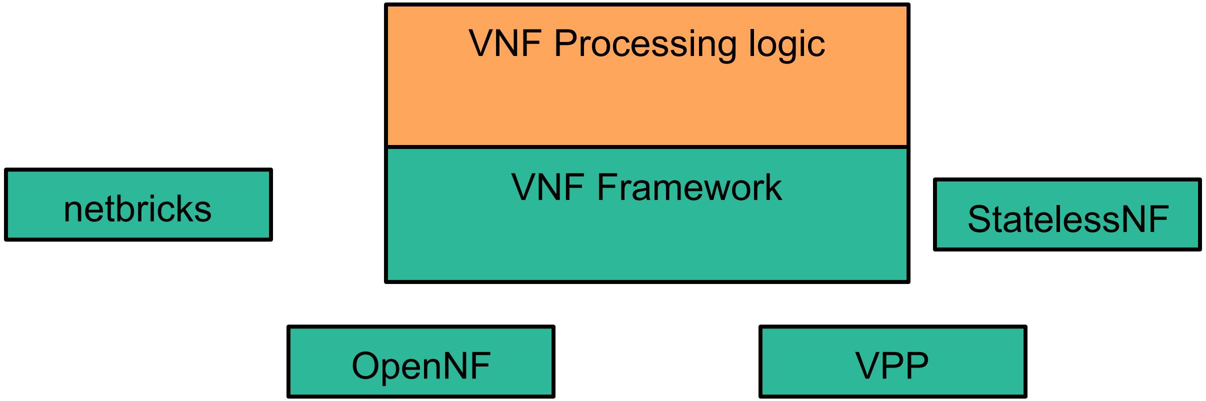
VNF Framework

CORD Intel EPC: https://gerrit.opencord.org/ngic

How to build VNF?



How to build VNF?



What is missing in these frameworks?

What is required from VNF frameworks?

- Requirement 1: Support for both L3 and Transport VNF
- Requirement 2: Flexibility of network stack
- Requirement 3: Support for distributed state management

What is required from VNF frameworks?

- **Requirement 1: Support for both L3 and Transport VNF**
- Requirement 2: Flexibility of network stack
- Requirement 3: Support for distributed state management

Support for L3 and transport VNFs

Layer 3 VNFs

Network address translator

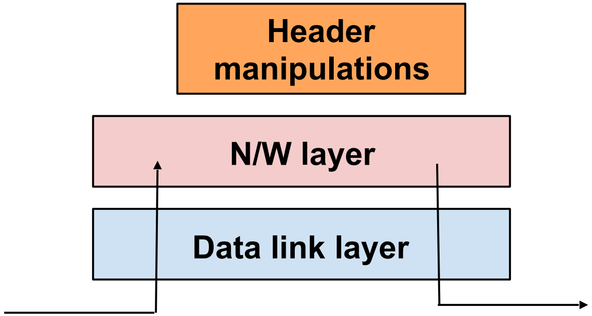
Layer 3 Load balancer

Support for L3 and transport VNFs

Layer 3 VNFs

Network address translator

Layer 3 Load balancer

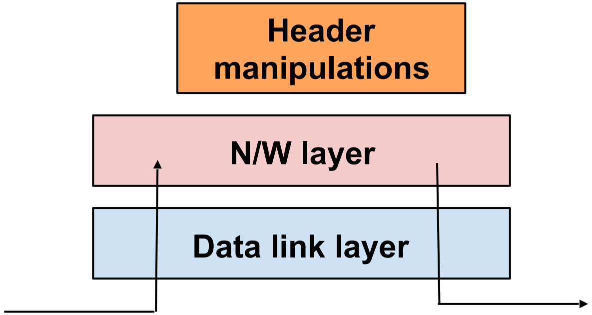


Support for L3 and transport VNFs

Layer 3 VNFs

Network address translator

Layer 3 Load balancer



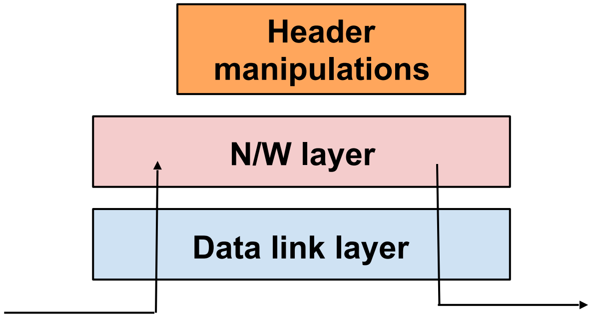
Frameworks: netbricks, YANFF

Support for L3 and transport VNFs

Layer 3 VNFs

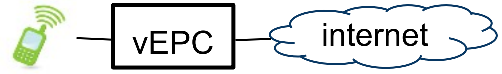
Network address translator

Layer 3 Load balancer



Frameworks: netbricks, YANFF

Transport Layer VNFs

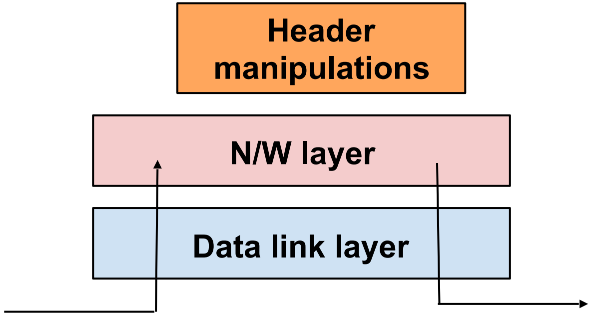


Support for L3 and transport VNFs

Layer 3 VNFs

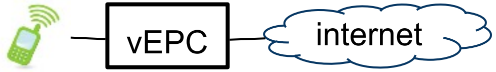
Network address translator

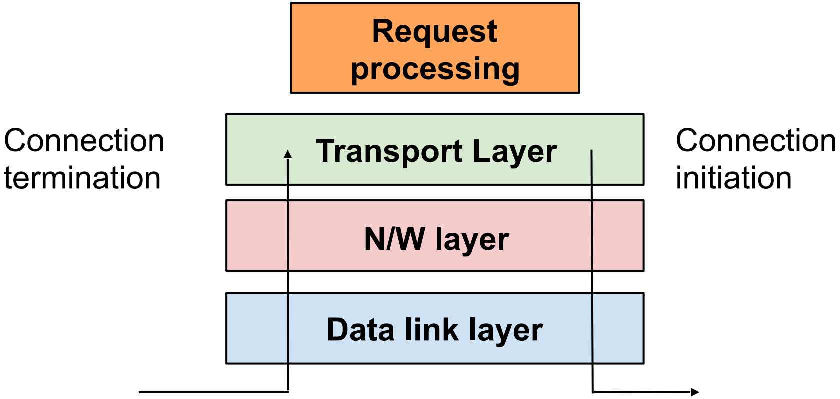
Layer 3 Load balancer



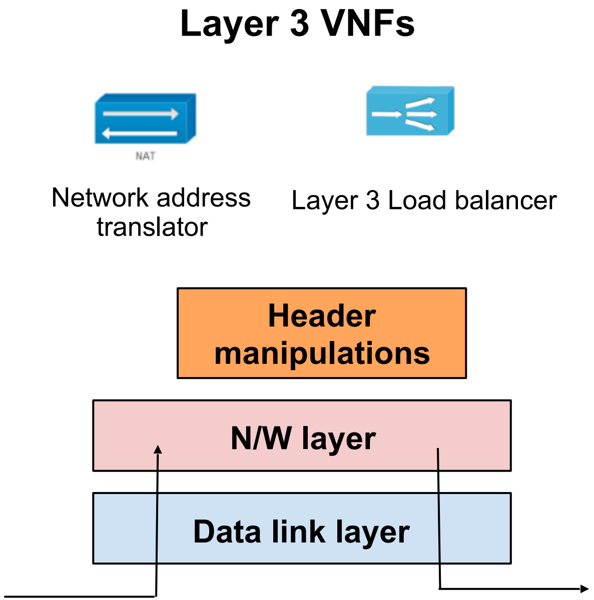
Frameworks: netbricks, YANFF

Transport Layer VNFs

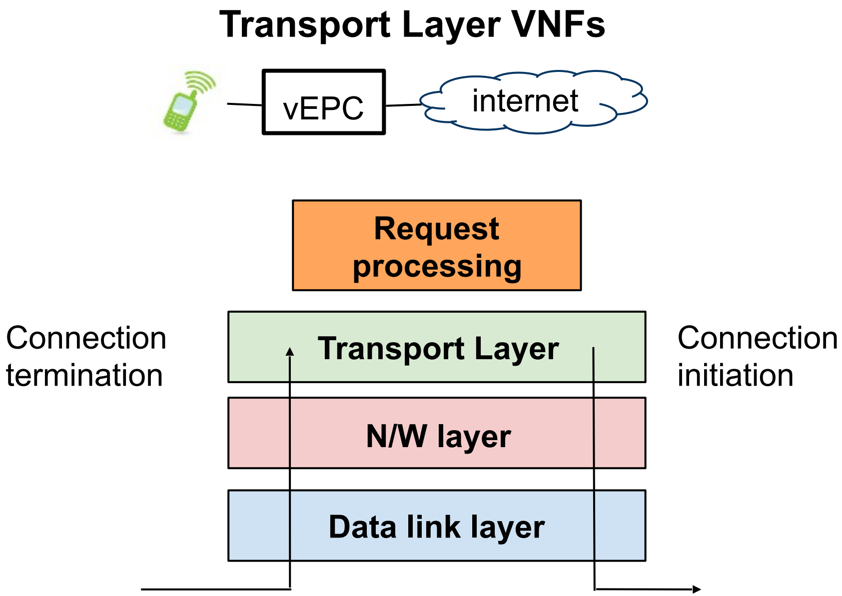




Support for L3 and transport VNFs



Frameworks: netbricks, YANFF

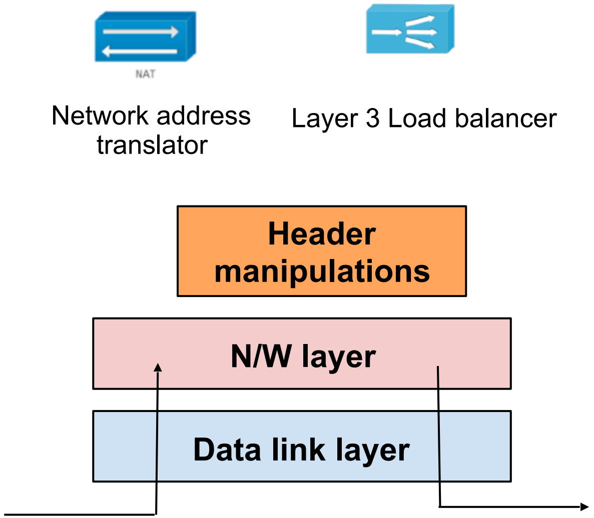


Frameworks: mTCP, TLDK

Netbricks: Taking the v out of nfv. In *Proc. of OSDI'16*
YANFF: *https://www.openhub.net/p/yanff*
mTCP: A highly scalable user-level tcp stack for multicore systems. In *Proc. of NSDI'14*
TLDK: *https://wiki.fd.io/view/TLDK*

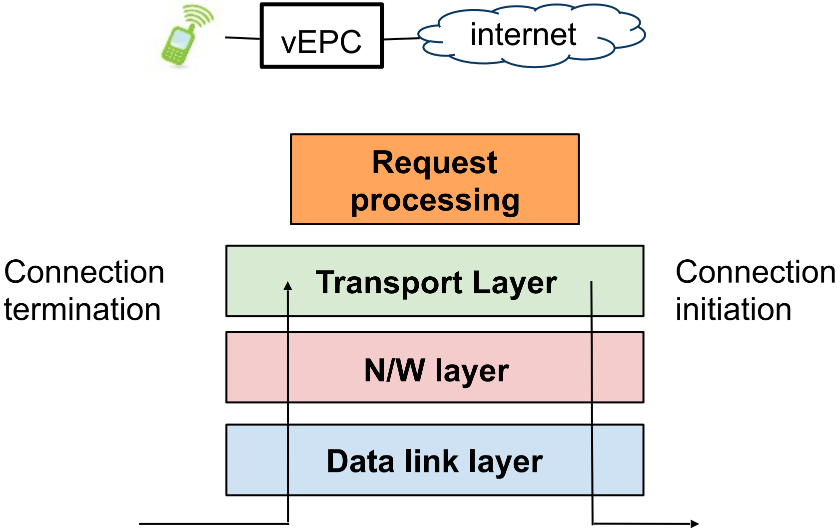
Support for L3 and transport VNFs

Layer 3 VNFs



Frameworks: netbricks, YANFF

Transport Layer VNFs



Frameworks: mTCP, TLDK

Are these frameworks enough?

Netbricks: Taking the v out of nfv. In *Proc. of OSDI'16*
YANFF: *https://www.openhub.net/p/yanff*
mTCP: A highly scalable user-level tcp stack for multicore systems. In *Proc. of NSDI'14*
TLDK: *https://wiki.fd.io/view/TLDK*

Event driven I/O

Existing transport-layer frameworks are event-driven

Event driven I/O

Existing transport-layer frameworks are event-driven

Pros:

Event driven I/O

Existing transport-layer frameworks are event-driven

Pros:

- Efficient for multi-core scalability

Event driven I/O

Existing transport-layer frameworks are event-driven

Pros:

- Efficient for multi-core scalability

Event driven I/O

Existing transport-layer frameworks are event-driven

Pros:

- Efficient for multi-core scalability

Cons:

Event driven I/O

Existing transport-layer frameworks are event-driven

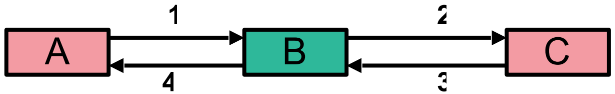
Pros:

- Efficient for multi-core scalability

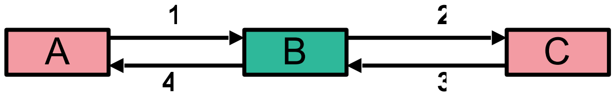
Cons:

- Needs explicit request state storage

Need to maintain request state



Need to maintain request state



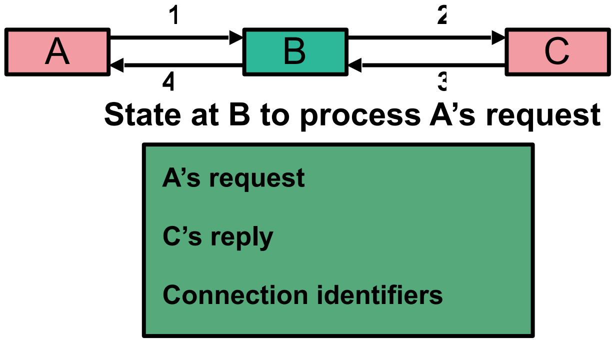
State at B to process A's request

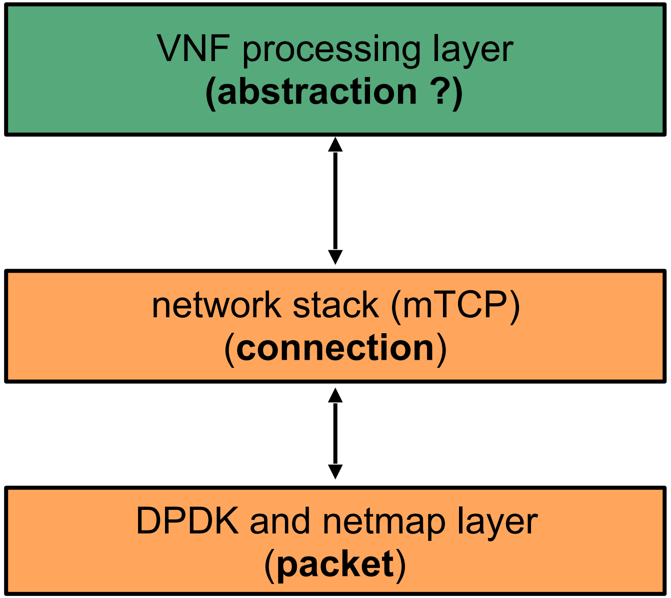
A's request

C's reply

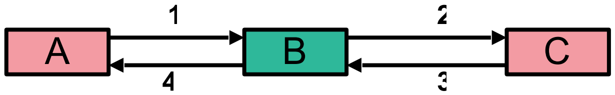
Connection identifiers

Need to maintain request state





Need to maintain request state

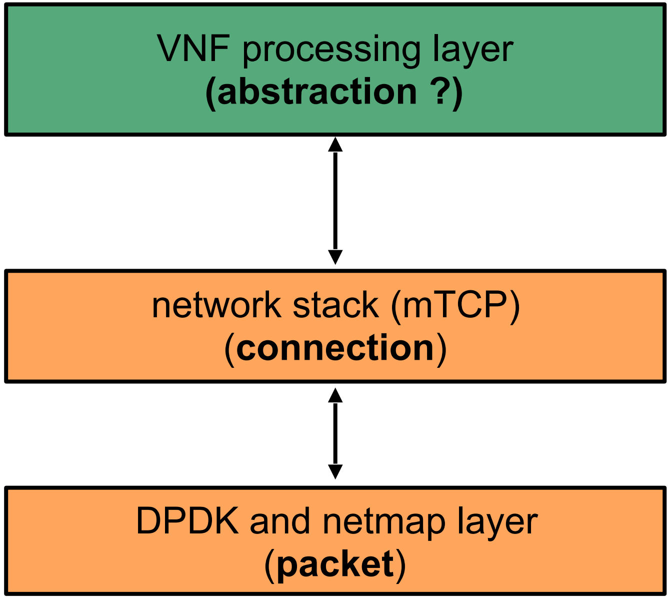


State at B to process A's request

A's request

C's reply

Connection identifiers



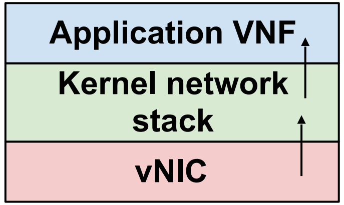
Existing frameworks do not provide this support

What is required from VNF frameworks?

- Requirement 1: Support for both Layer 3 and Transport VNF
- **Requirement 2: Flexibility of network stack**
- Requirement 3: Support for distributed state management

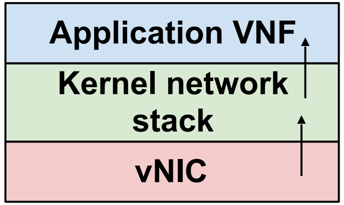
Flexibility of network stack

Kernel Stack

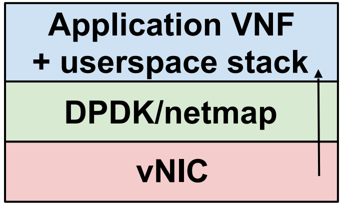


Flexibility of network stack

Kernel Stack

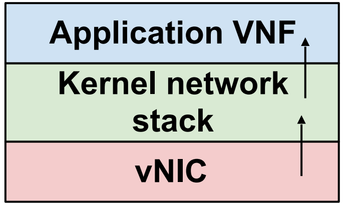


Kernel Bypass Stack

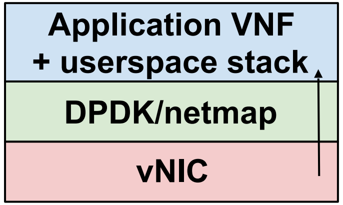


Flexibility of network stack

Kernel Stack



Kernel Bypass Stack

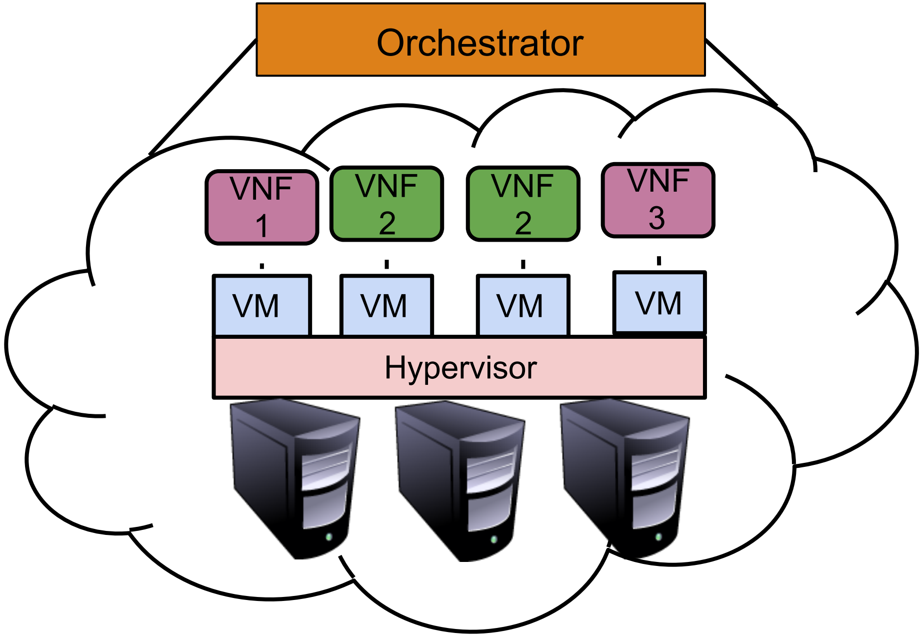


Easy switch between stacks

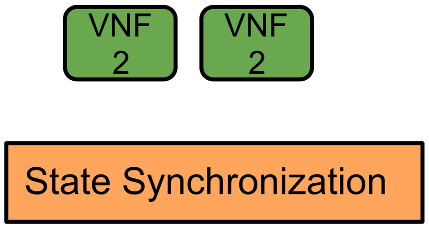
What is required from VNF frameworks?

- Requirement 1: Support for both L3 and Transport VNF
- Requirement 2: Flexibility of network stack
- **Requirement 3: Support for distributed state management**

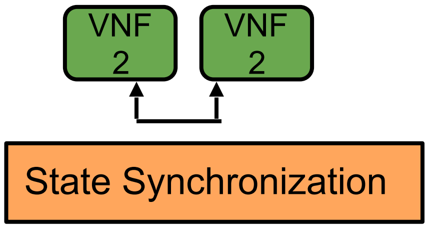
Support for distributed state management



Support for distributed state management

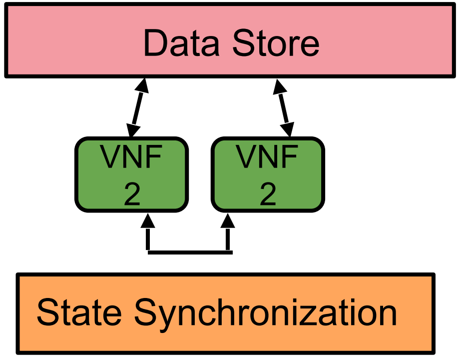


Support for distributed state management



State Migration
openNF,
split/merge

Support for distributed state management

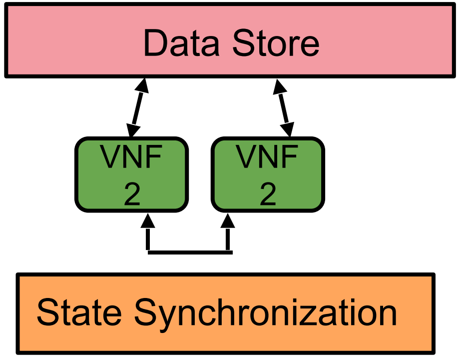


Remote store
statelessNF

State Migration
openNF,
split/merge

Stateless network functions: Breaking the tight coupling of state and processing. In *Proc. of NSDI'17*
Split/merge: System support for elastic execution in virtual middleboxes. In *Proc. of NSDI'13*
Opennf: Enabling innovation in network function control. In *Proc. of SIGCOMM'14*

Support for distributed state management



Remote store
statelessNF

State Migration
openNF,
split/merge

None of above support transport layer VNFs

Stateless network functions: Breaking the tight coupling of state and processing. In *Proc. of NSDI'17*
Split/merge: System support for elastic execution in virtual middleboxes. In *Proc. of NSDI'13*
Opennf: Enabling innovation in network function control. In *Proc. of SIGCOMM'14*

Summary of VNF Frameworks

Requirement/ Framework	netbricks	Flick	StatelessNF	Split-Merge/ OpenNF	libVNF
Layer 3 + App-layer support	no	yes	no	no	yes
Flexibility of network stack	no	no	no	no	yes
Distributed State Management	no	no	yes	yes	yes

Netbricks: Taking the v out of nfv. In *Proc. of OSDI'16*
Flick: Developing and running application-specific network services. In *Proc. of USENIX ATC'16*
Stateless network functions: Breaking the tight coupling of state and processing. In *Proc. of NSDI'17*
Split/merge: System support for elastic execution in virtual middleboxes. In *Proc. of NSDI'13*
Opennf: Enabling innovation in network function control. In *Proc. of SIGCOMM'14*

libVNF Design Goals

VNF processing logic

Handled by
VNF developer

R3 Distributed State Management

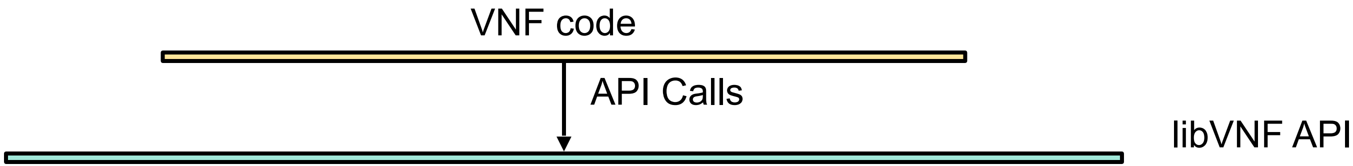
R2 Support for network and transport layer VNF

R1 Flexibility of network stack

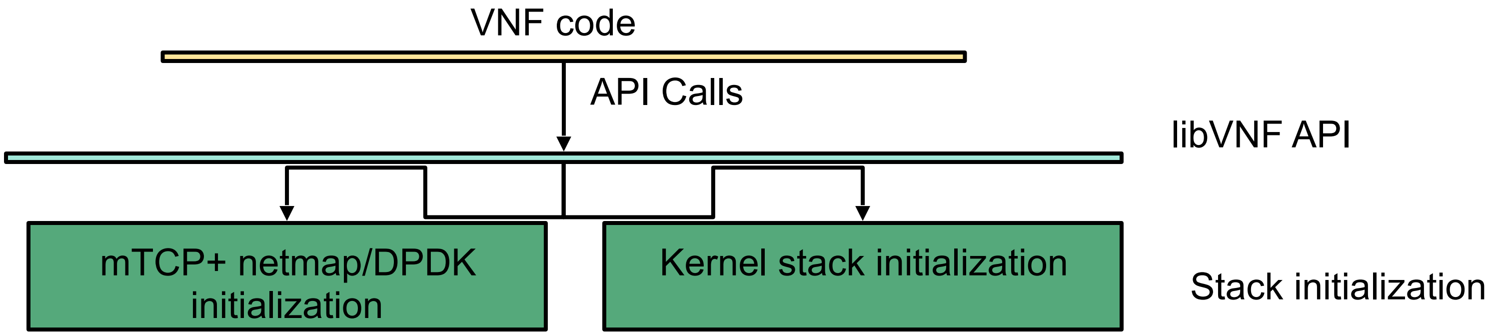
Handled by
libVNF

R: Requirement

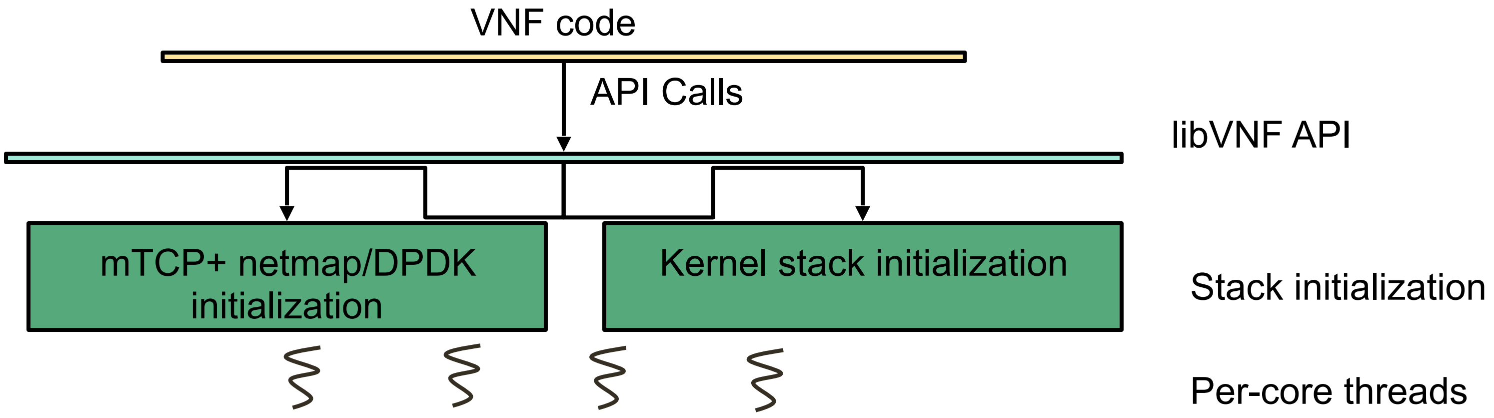
libVNF overview



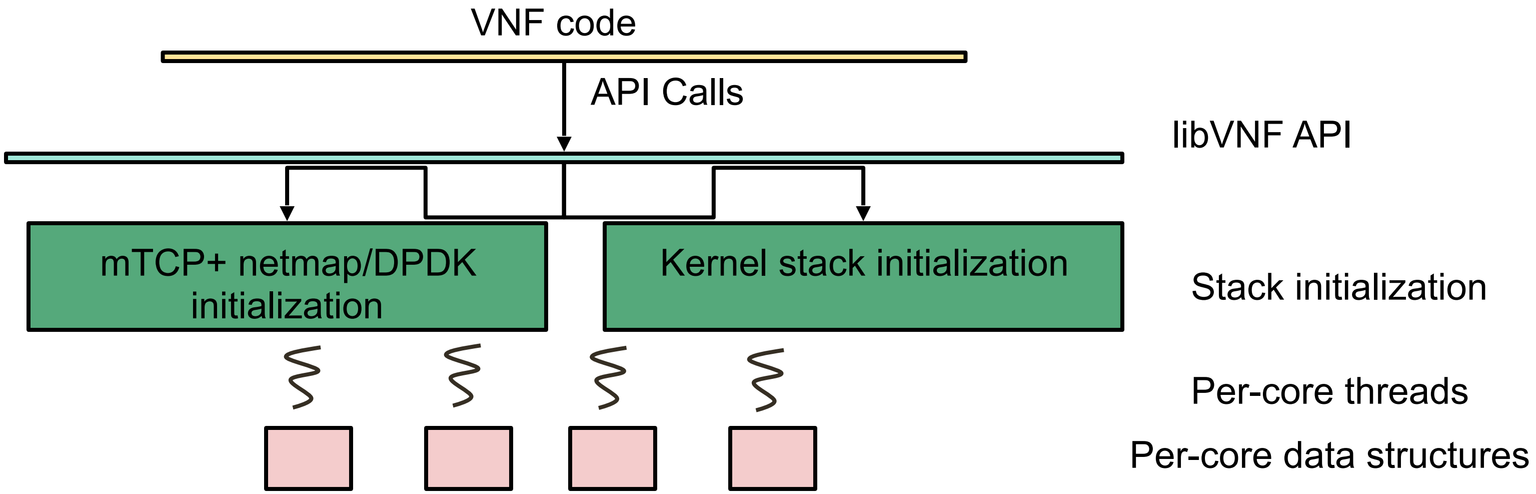
libVNF overview



libVNF overview



libVNF overview



- **Lock-free**
- **Cache optimized**

libVNF API

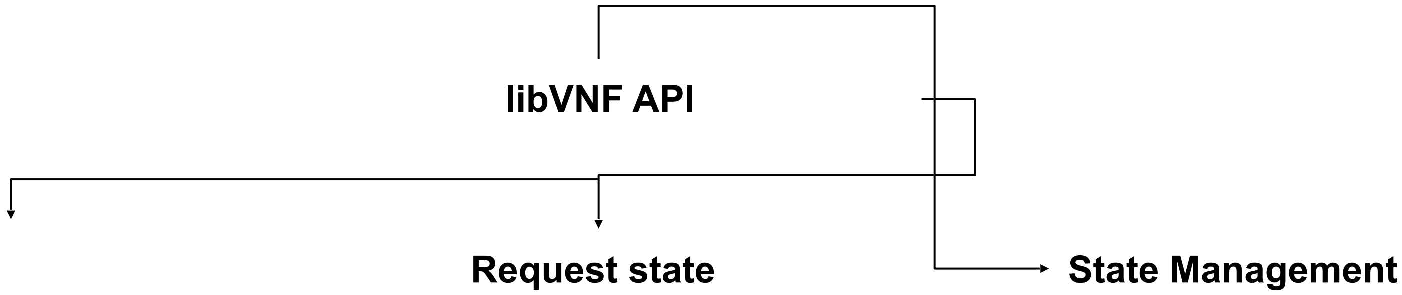
libVNF API

Communication

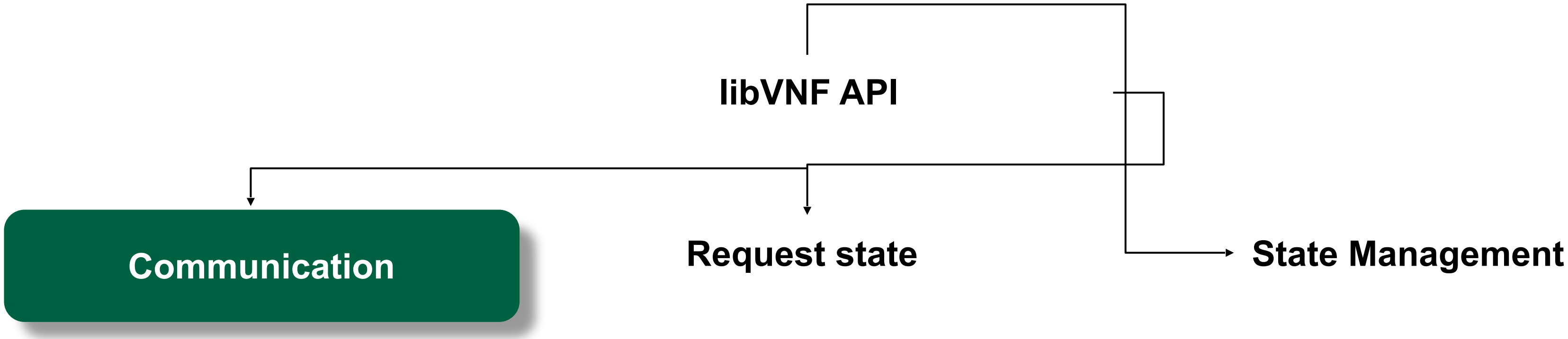
Request state

State Management

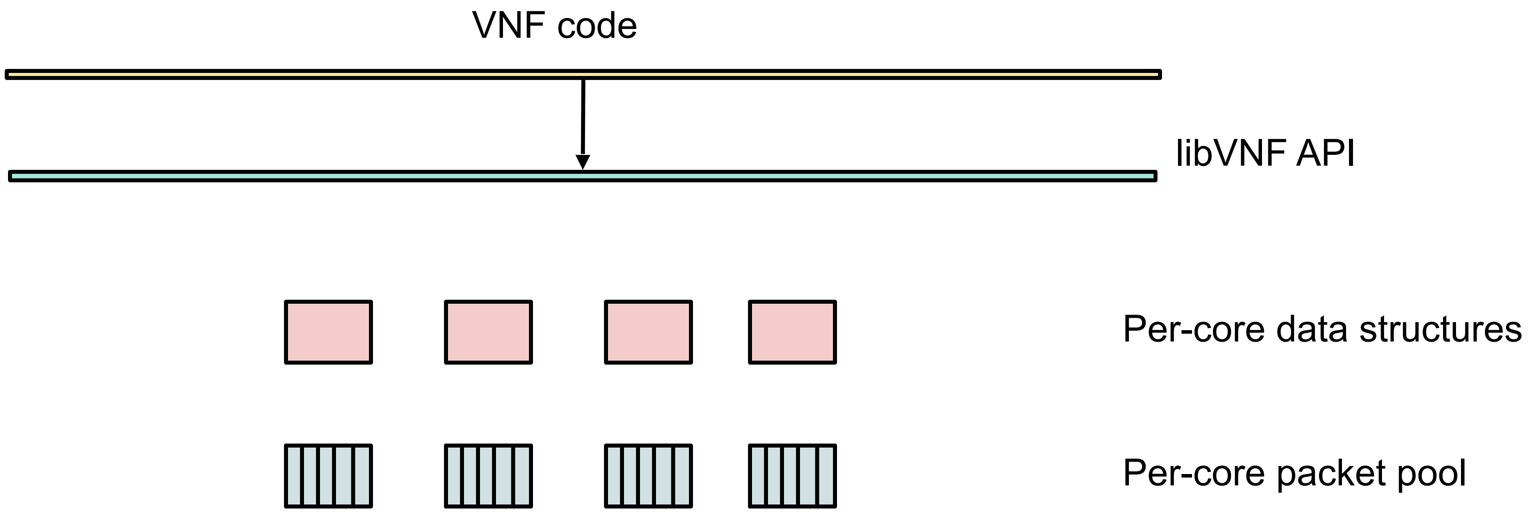
libVNF API



libVNF API

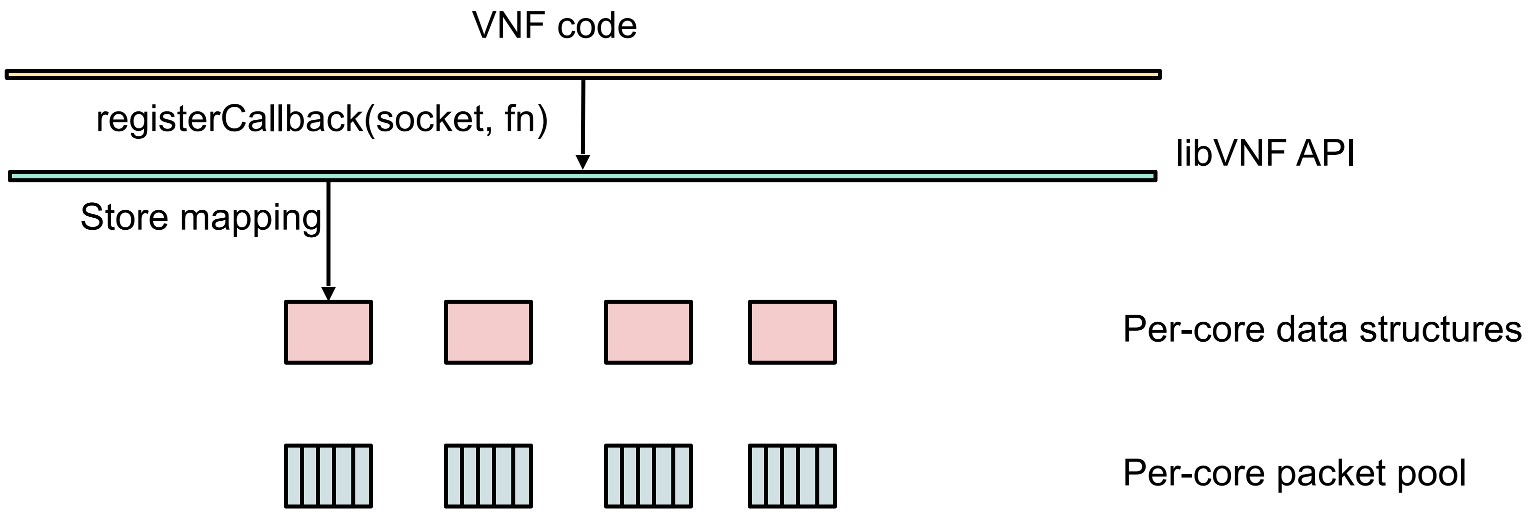


Communication API



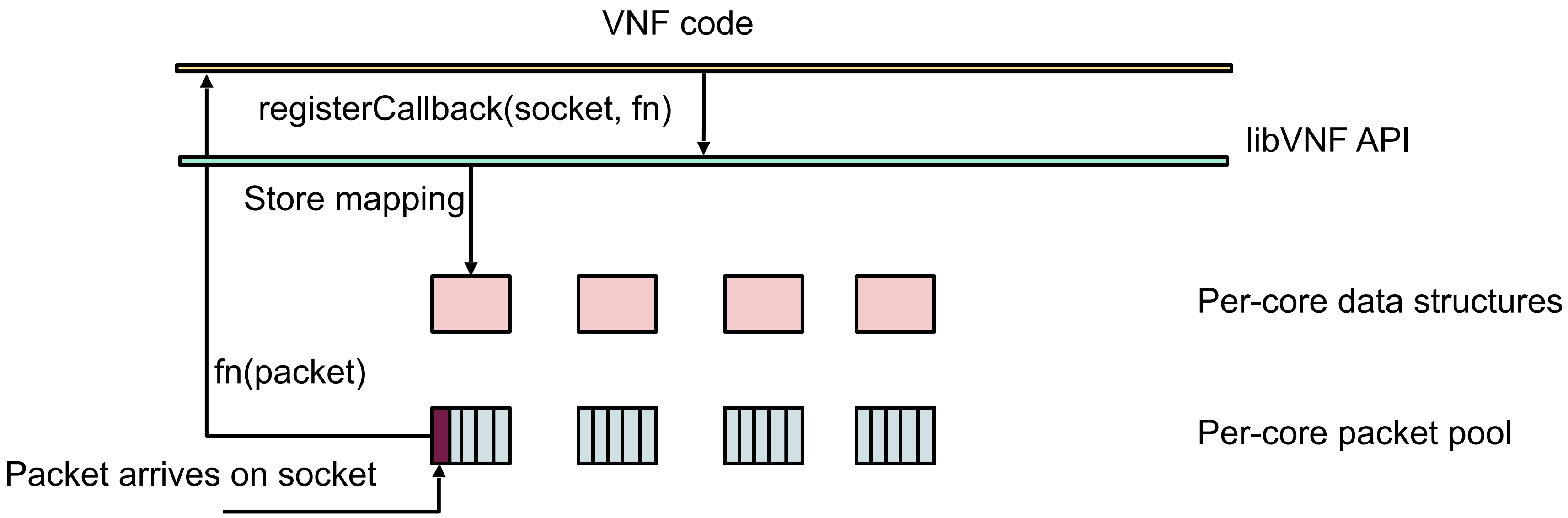
Pre-allocated memory pools (Per-core packet pools)

Communication API



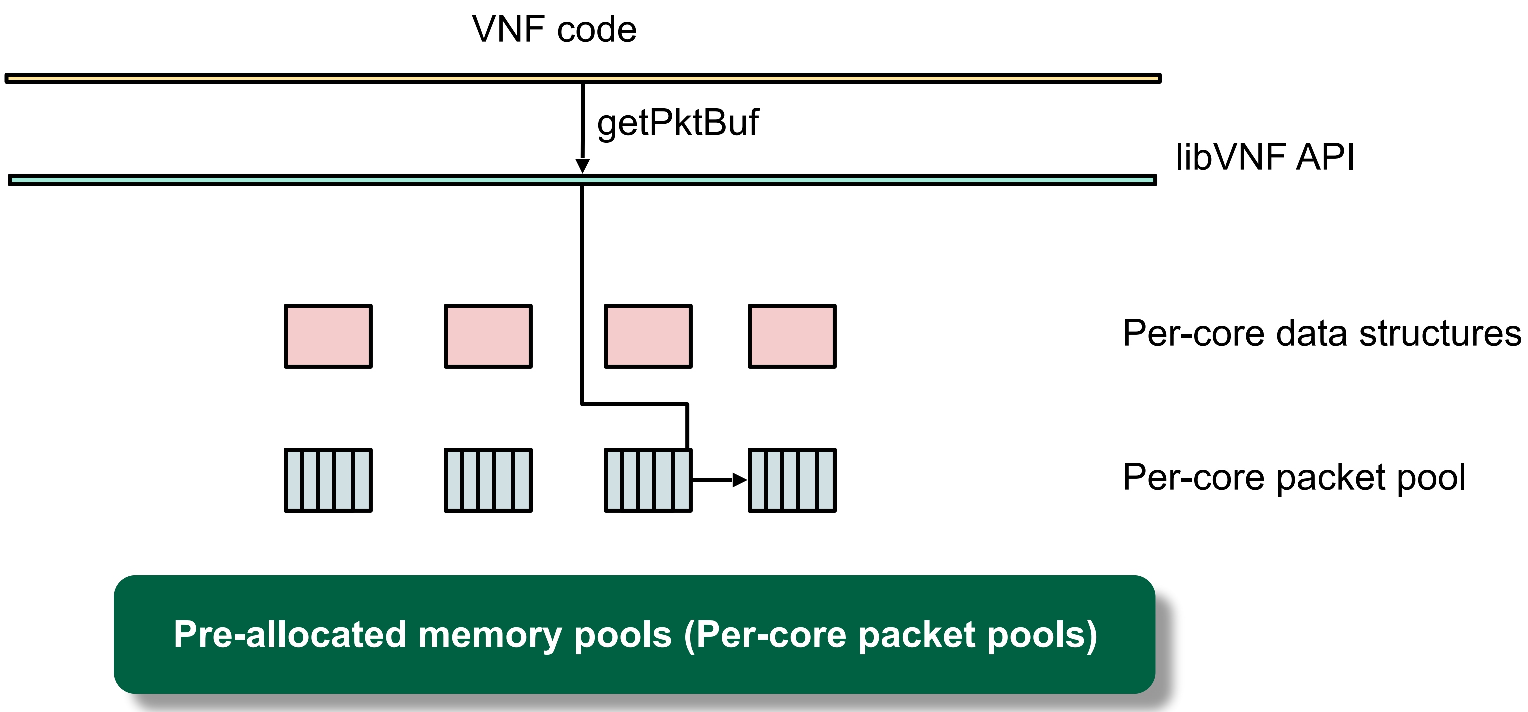
Pre-allocated memory pools (Per-core packet pools)

Communication API

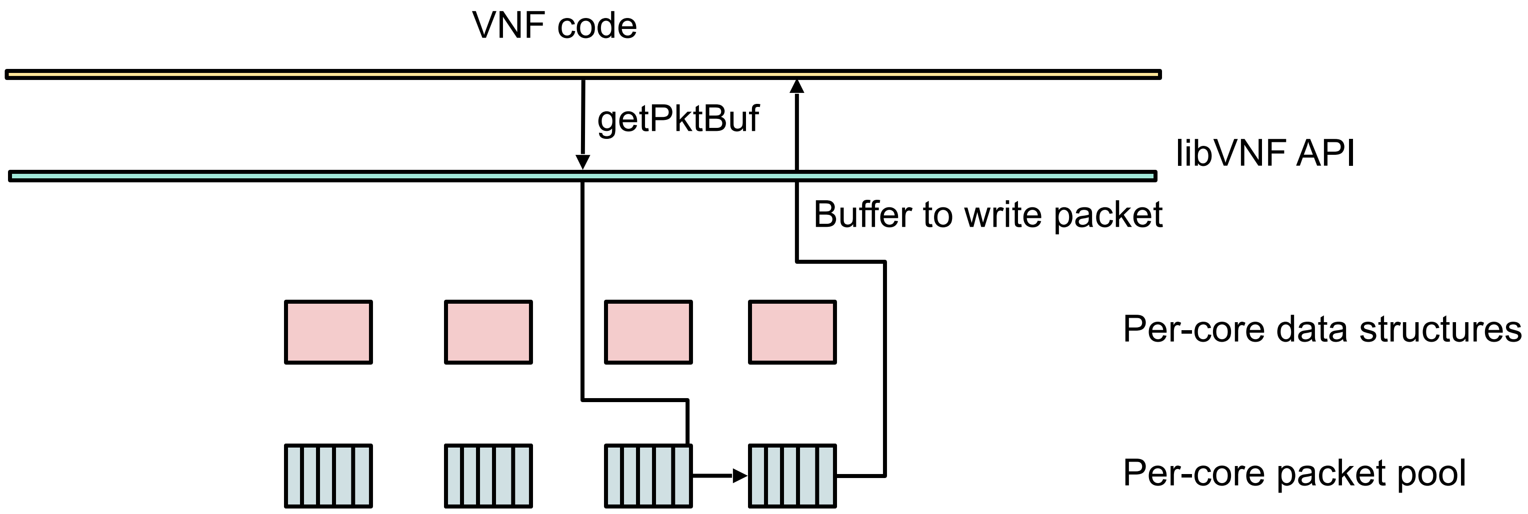


Pre-allocated memory pools (Per-core packet pools)

Communication API



Communication API



Pre-allocated memory pools (Per-core packet pools)

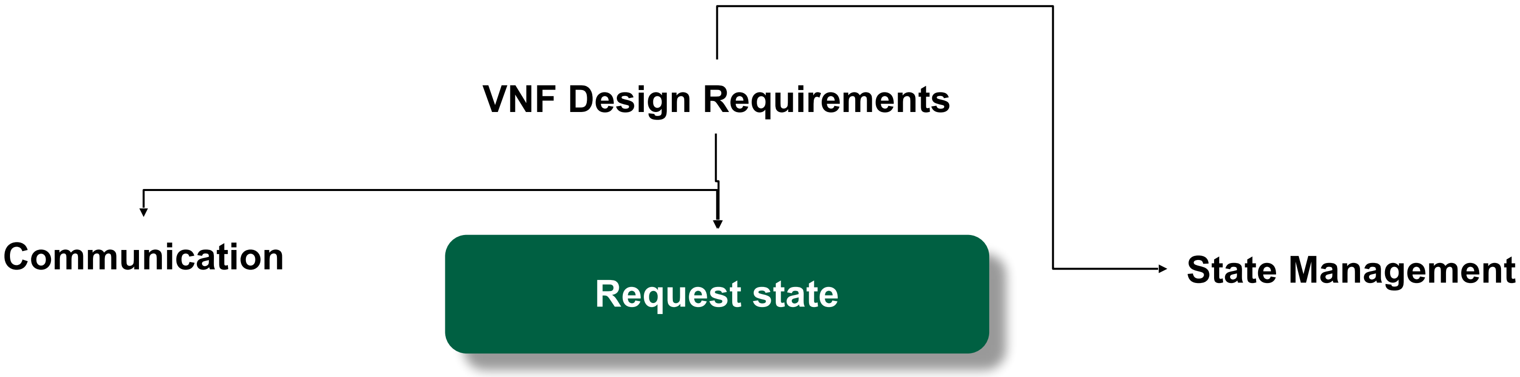
libVNF API

VNF Design Requirements

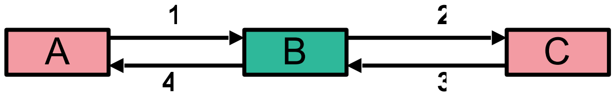
Communication

State Management

libVNF API


VNF Design Requirements
Communication
Request state
State Management

Need for request state

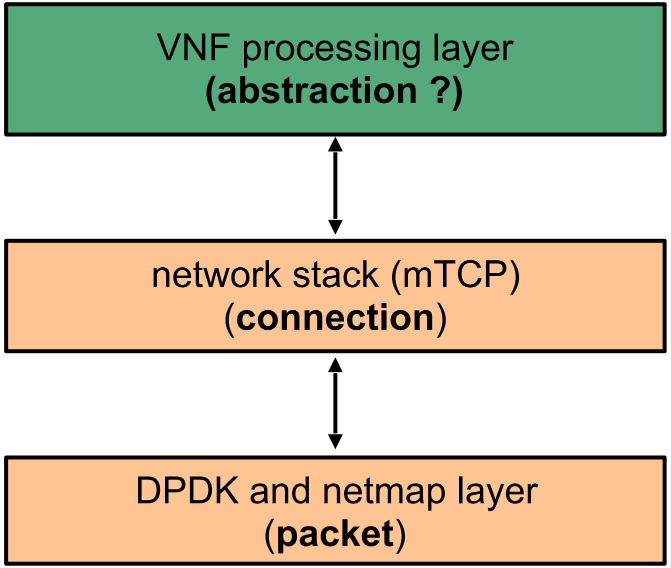


State at B to process A's request

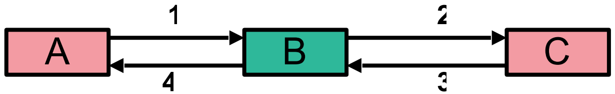
A's request

C's reply

Connection identifiers



Need for request state



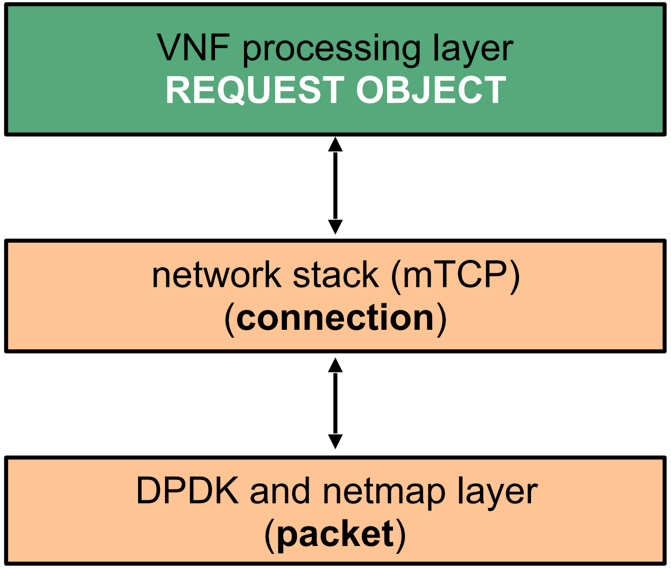
State at B to process A's request

A's request

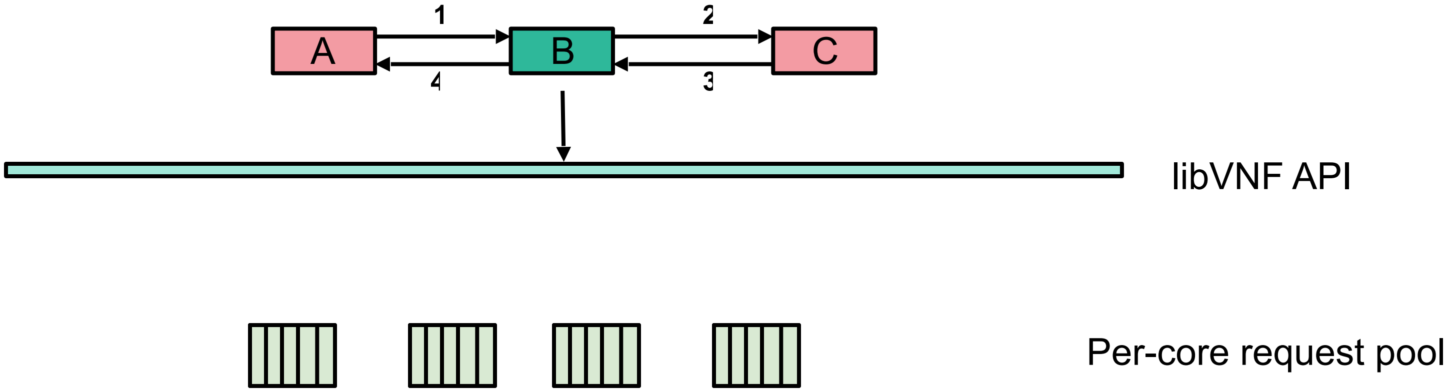
C's reply

Connection identifiers

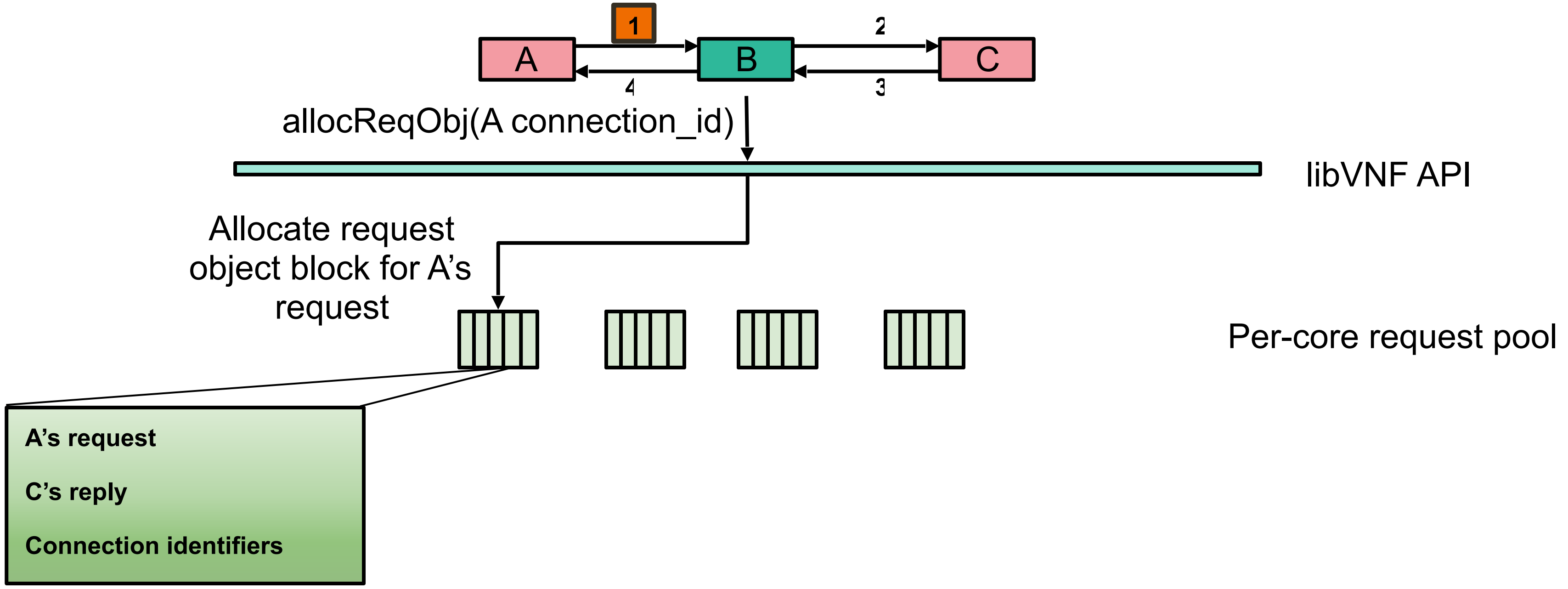
Request object



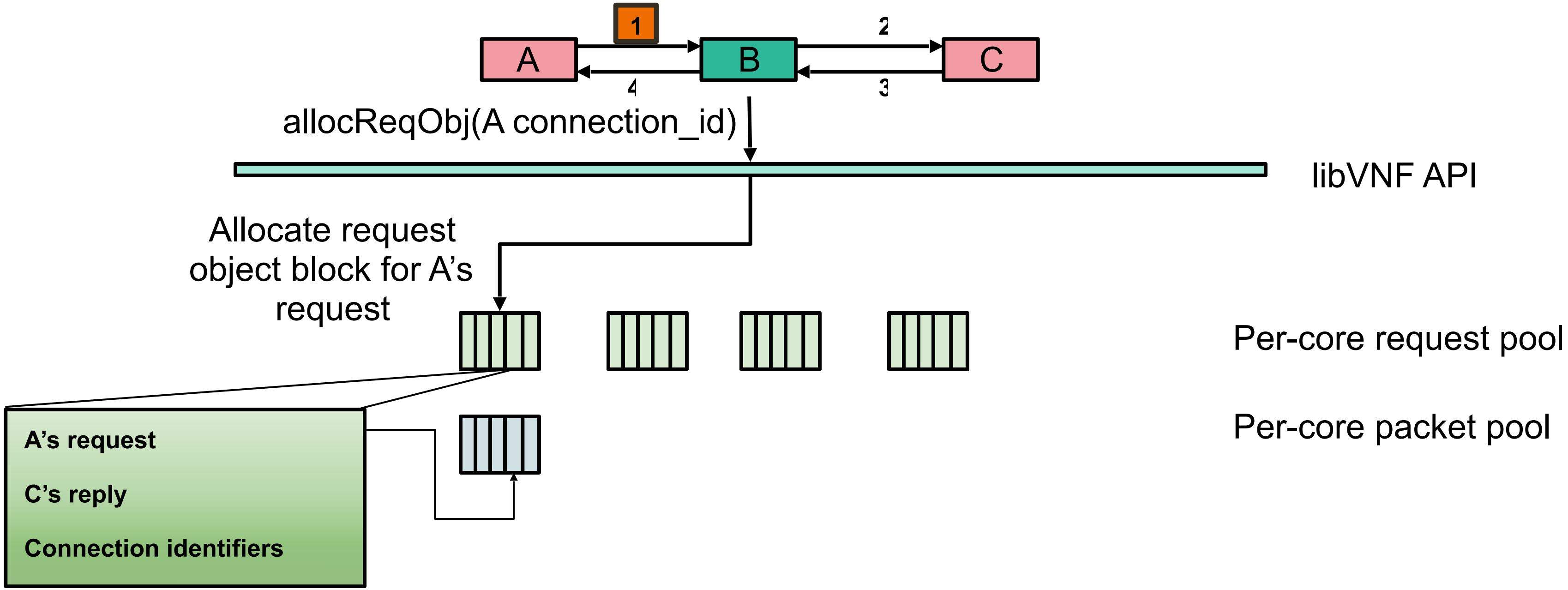
Request Object API



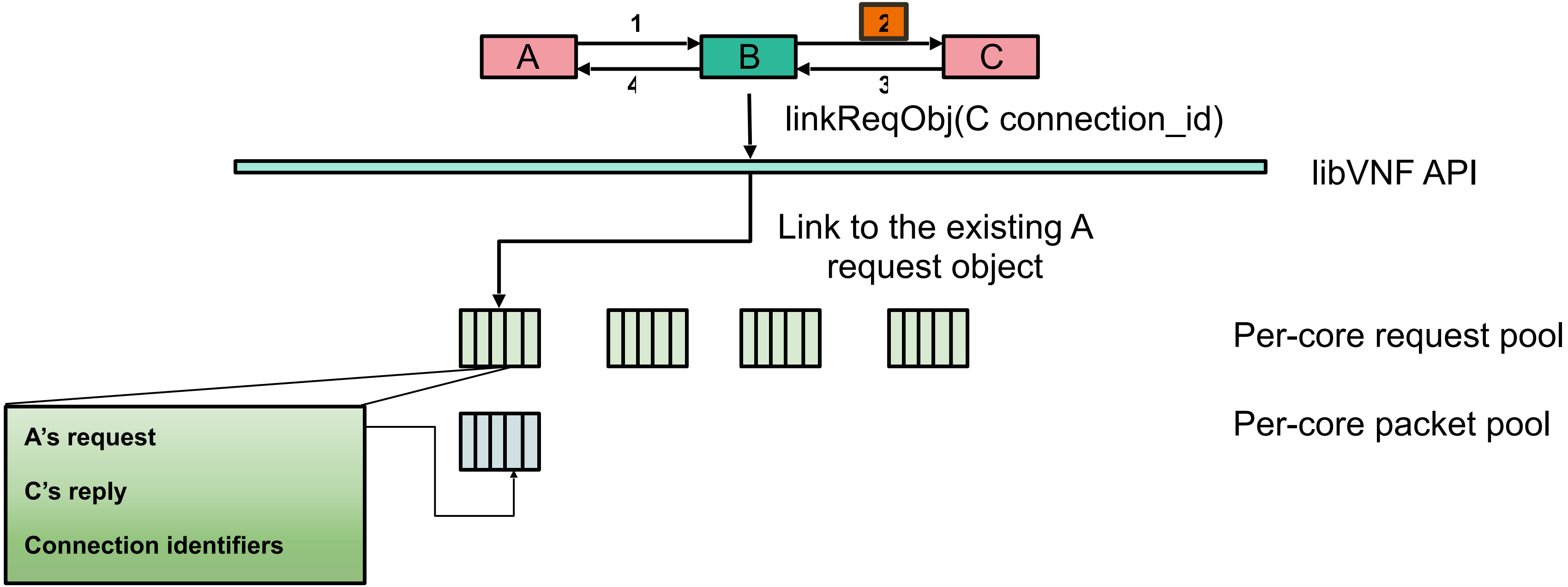
Request Object API



Request Object API



Request Object API



libVNF API

VNF Design Requirements

Communication

Request state

libVNF API

VNF Design Requirements

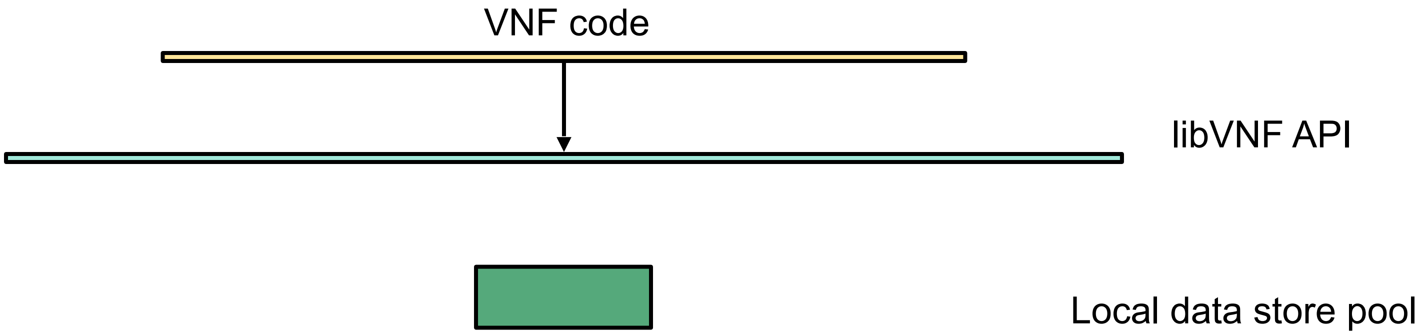
Communication

Request state

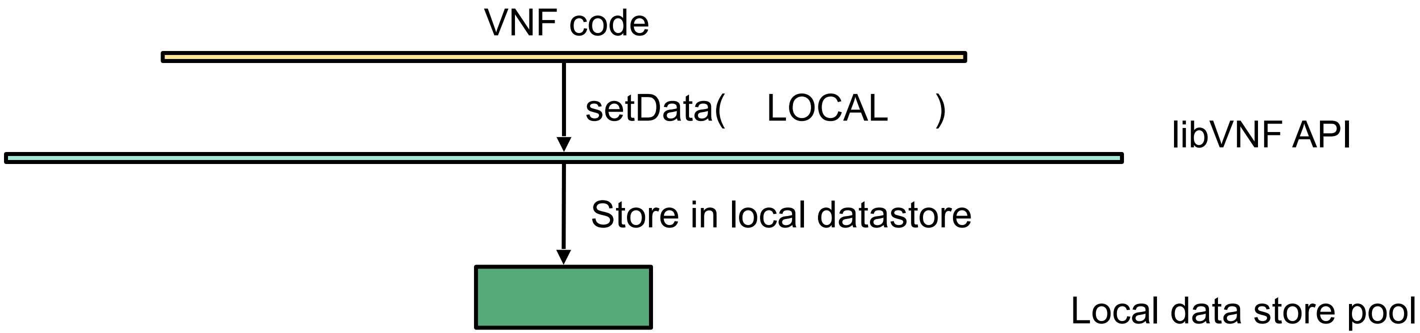
State Management

State Management API

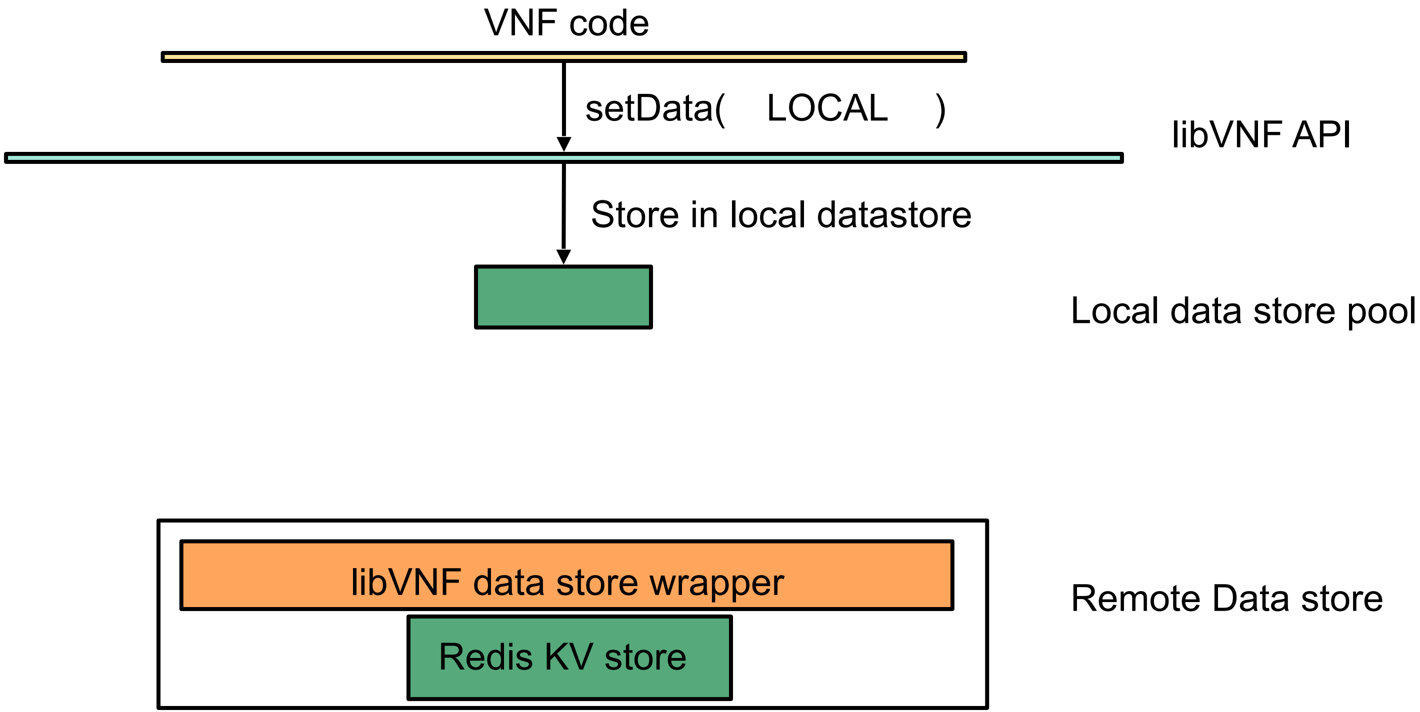
State Management API



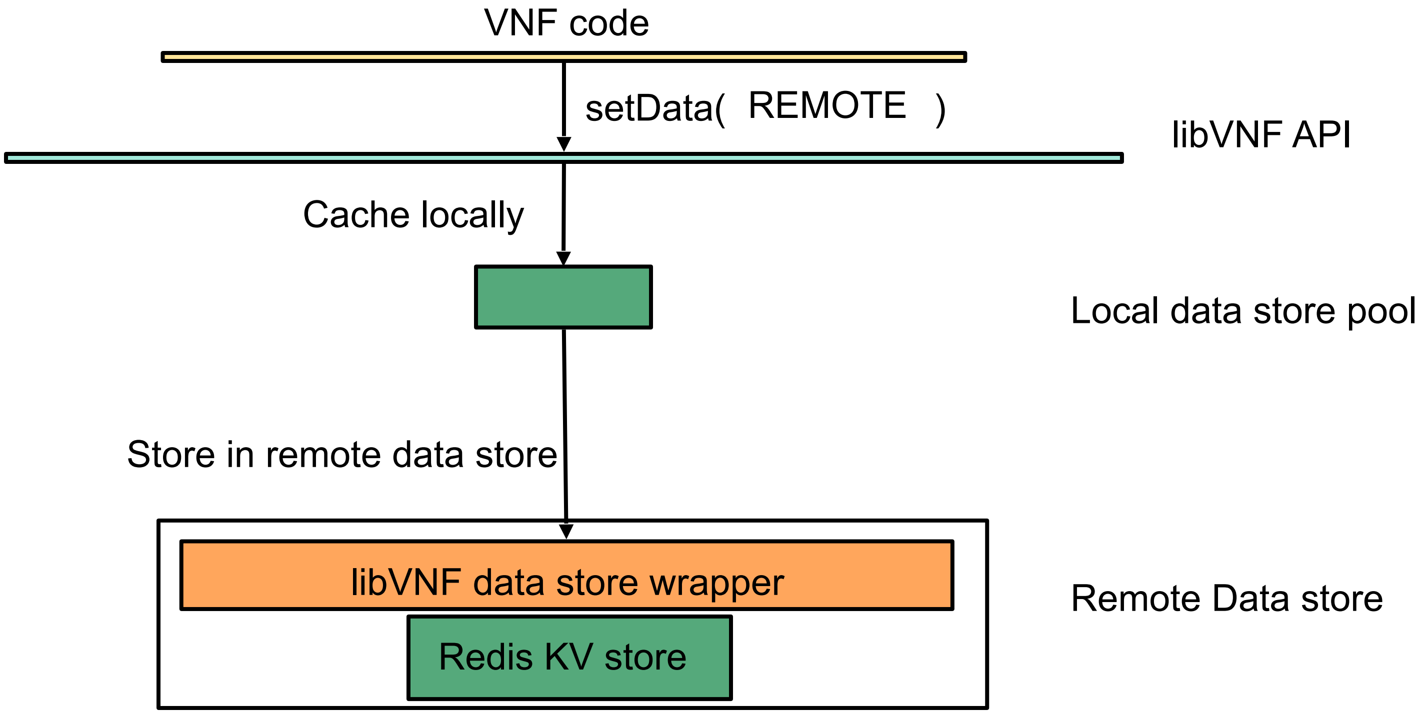
State Management API



State Management API



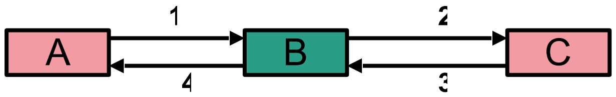
State Management API



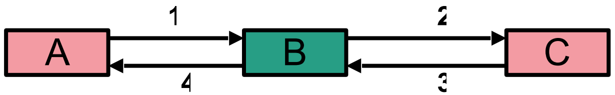
Evaluation

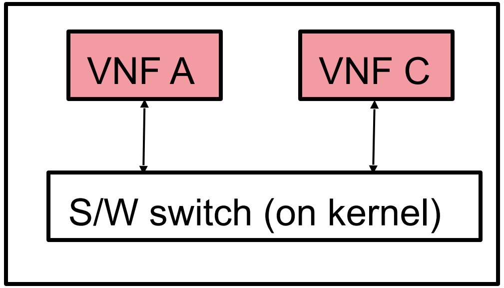
- Overhead of libVNF
- Scalability with cores
- Benefits of libVNF

Setup

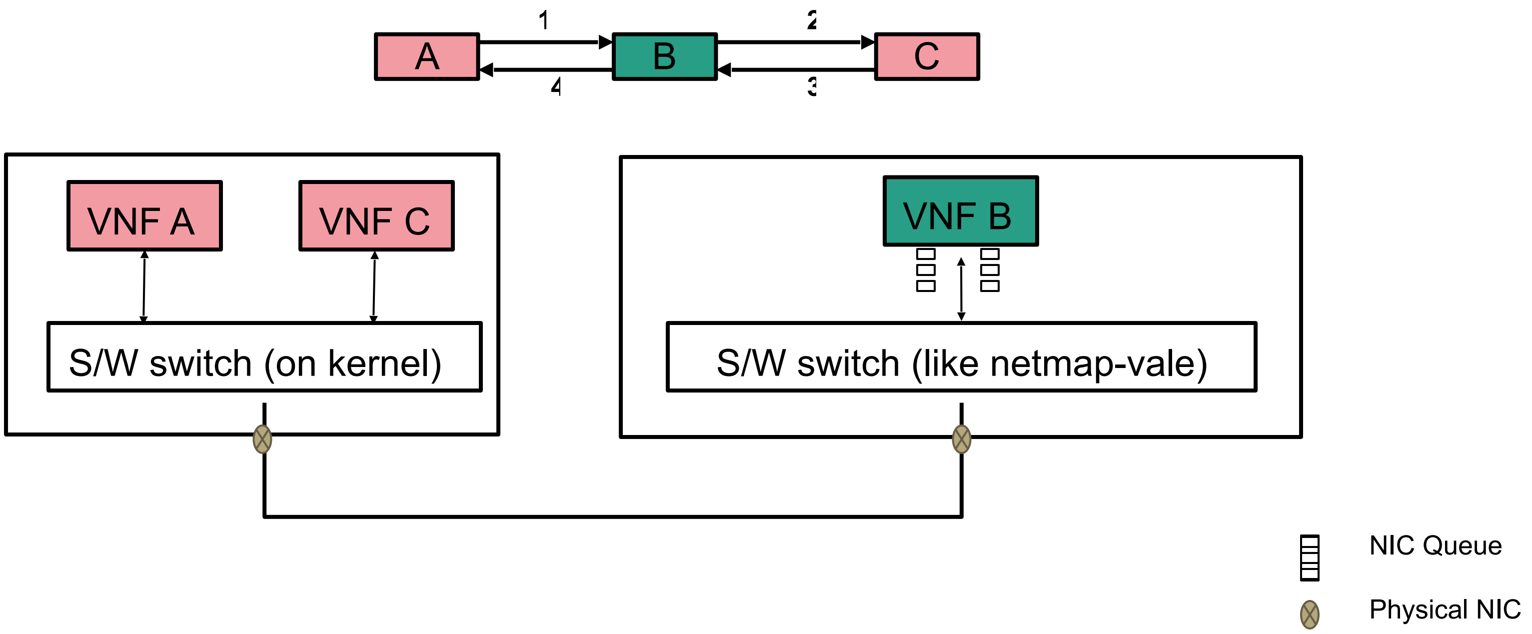


Setup



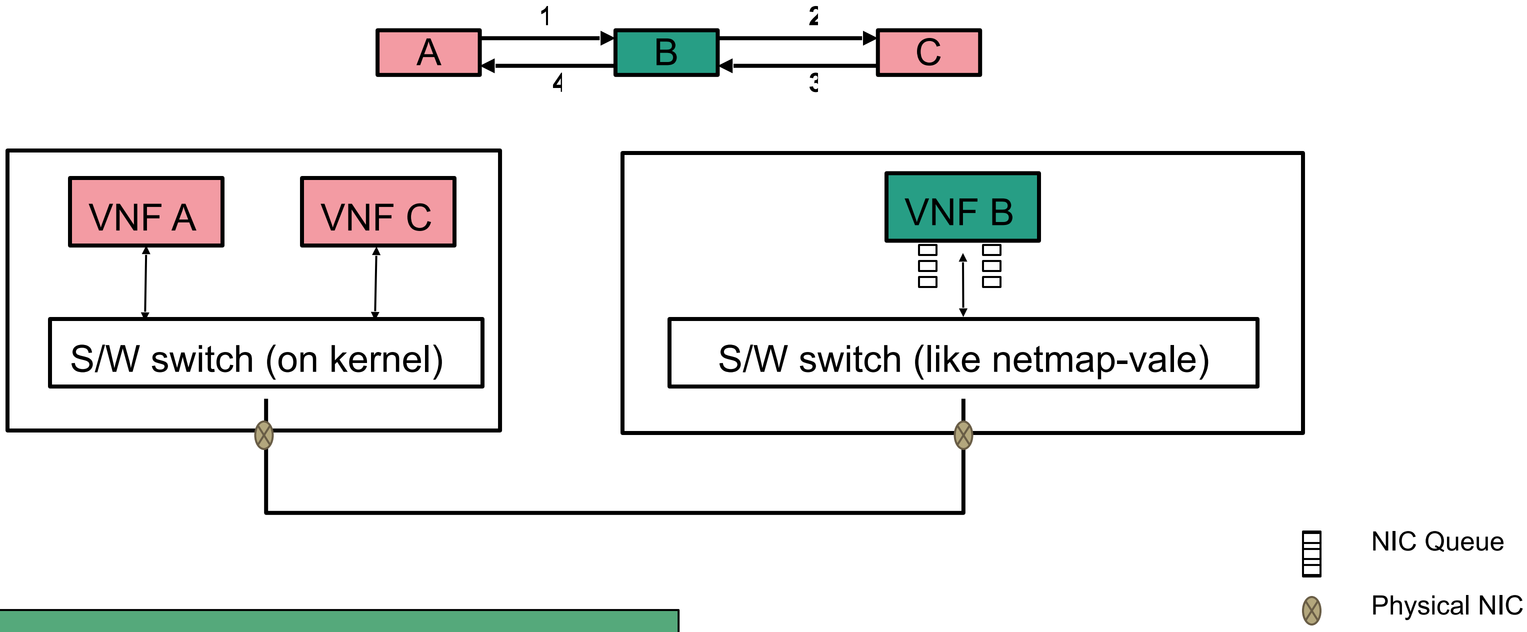


Setup



1
2
A
B
C
4
3
VNF A
VNF C
S/W switch (on kernel)
VNF B
S/W switch (like netmap-vale)
NIC Queue
Physical NIC

Setup

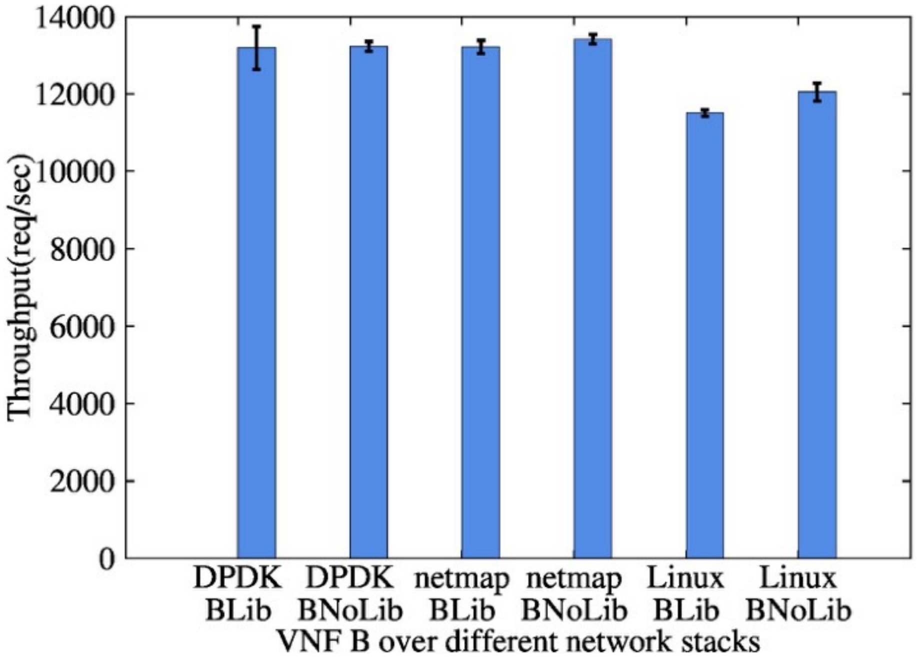


VNF A, C: 4 core, 4GB RAM
VNF B: 4 GB RAM, cores varied

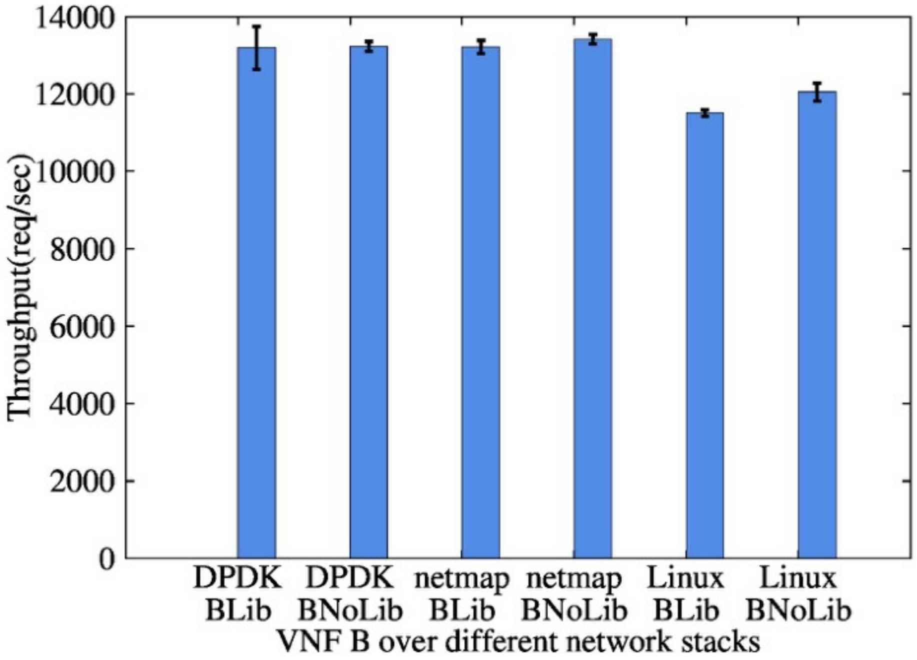
Evaluation

- **Overhead of libVNF**
- Scalability with cores
- Benefits of libVNF

Overhead check



Overhead check



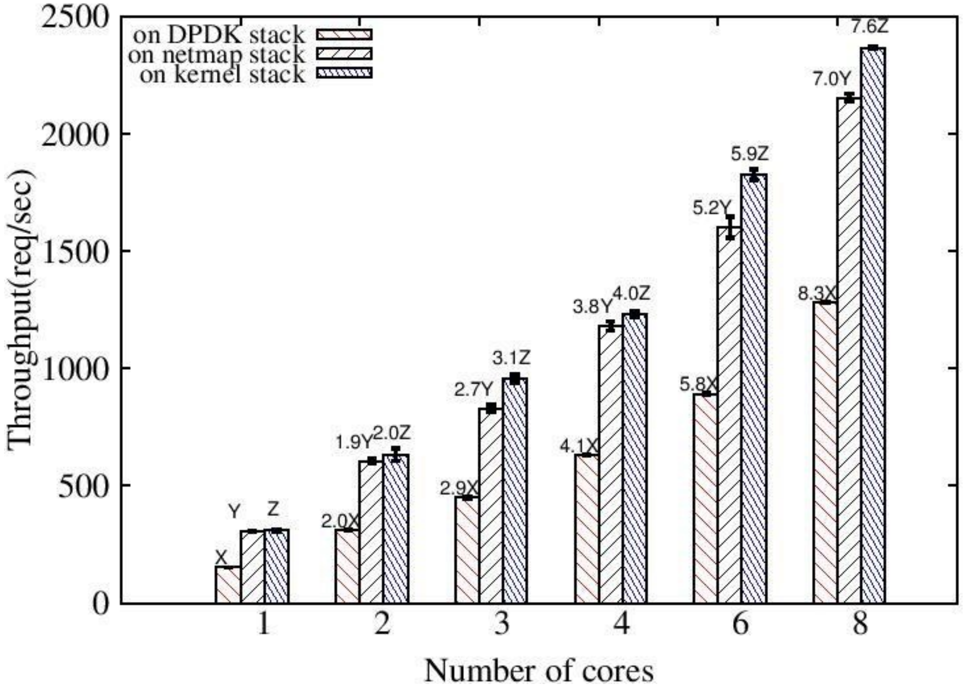
<5% overhead of libVNF

DPDK~ netmap performance

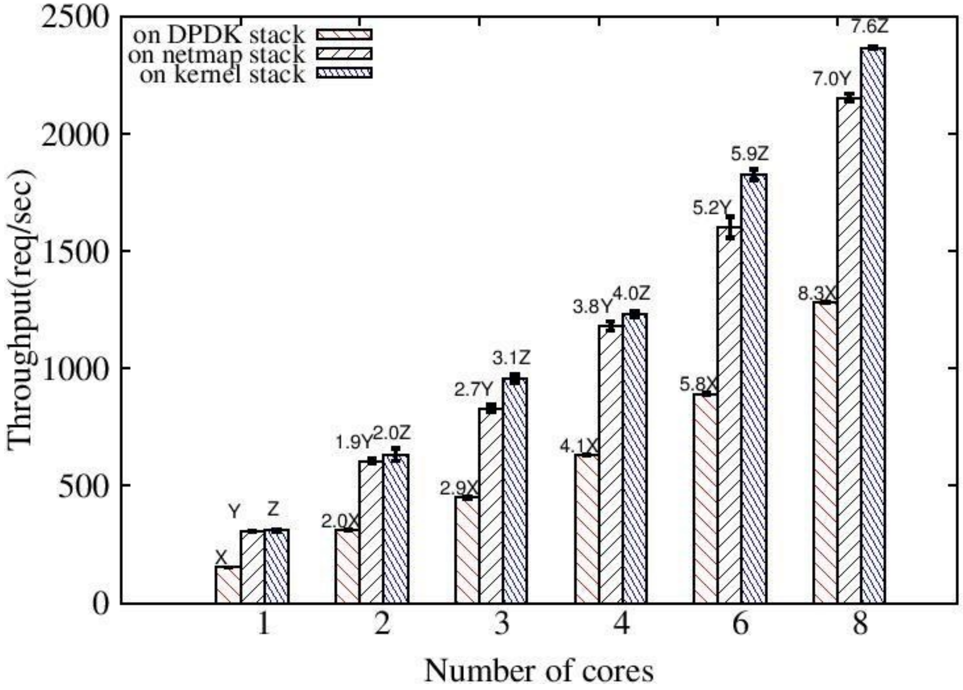
Evaluation

- Overhead of libVNF
- **Scalability with cores**
- Benefits of libVNF

Core scalability



Core scalability



scales linearly with cores

Evaluation

- Overhead of libVNF
- Scalability with cores
- **Benefits of libVNF**

Building VNFs

VNF	Performance Overhead of libVNF	LoC Saved
IMS (IP Multimedia Subsystem)	3.4%	42%
EPC (LTE-Evolved Packet Core)	5.5%	38%
Layer 3 Load Balancer	14%	52%

Building VNFs

VNF	Performance Overhead of libVNF	LoC Saved
IMS (IP Multimedia Subsystem)	3.4%	42%
EPC (LTE-Evolved Packet Core)	5.5%	38%
Layer 3 Load Balancer	14%	52%

Low overhead in app-layer VNF
Higher overhead in L3 VNF

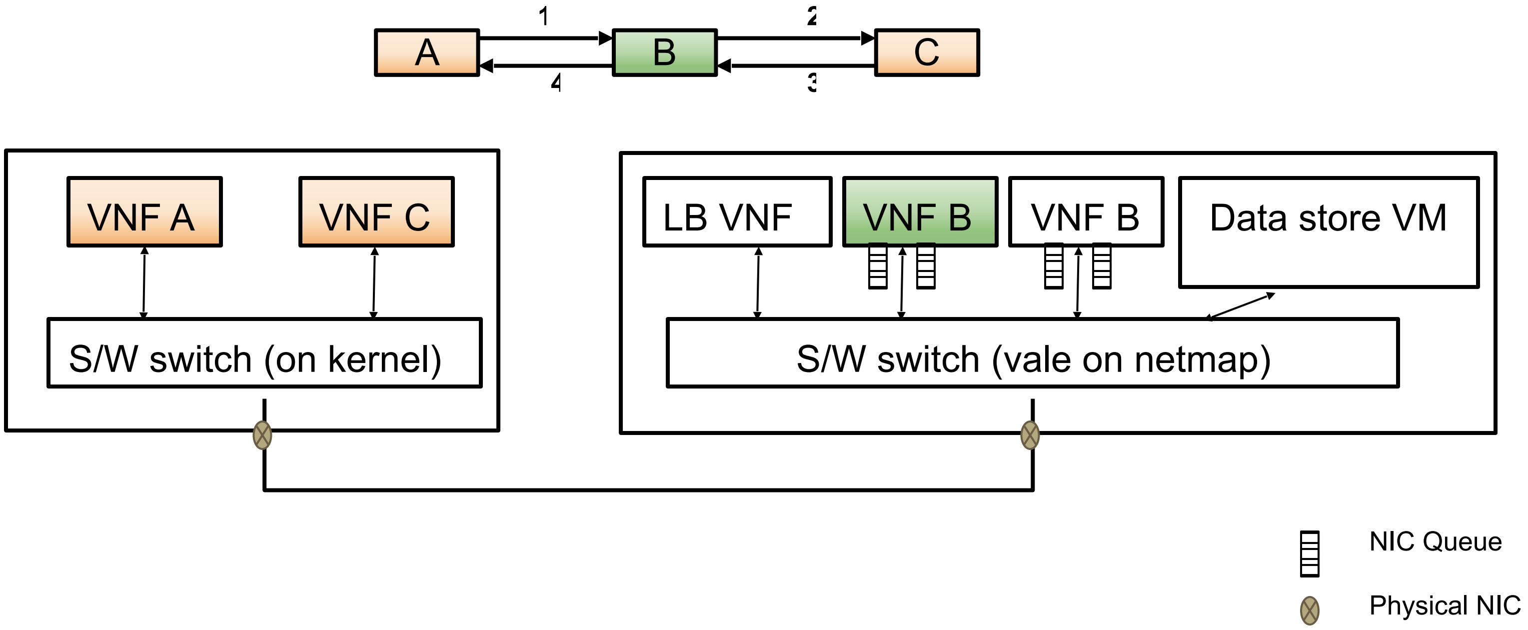
Summary

- Library to ease building of VNFs
- Expressive to build L3 and App-layer VNF
- Supports multiple network stacks
- Low performance overhead

https://github.com/networkedsystemsIITB/libVNF
ppnaik@cse.iitb.ac.in

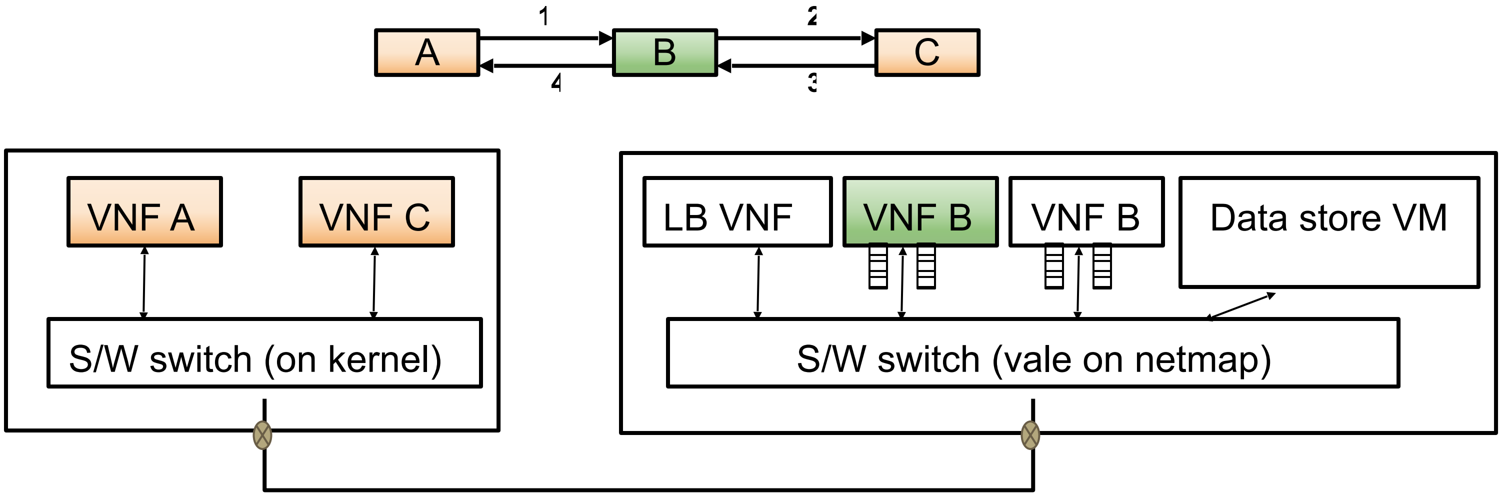
Thank You

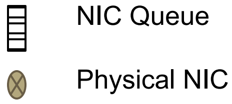
Setup



1
2
A
B
C
4
3
VNF A
VNF C
S/W switch (on kernel)
LB VNF
VNF B
VNF B
Data store VM
S/W switch (vale on netmap)
NIC Queue
Physical NIC

Setup





VNF A, C: 4 core, 4GB RAM
VNF B: 4 GB RAM, cores varied
Data Store VM: 6 core, 16GB RAM
LB: 1 core, 4GB RAM