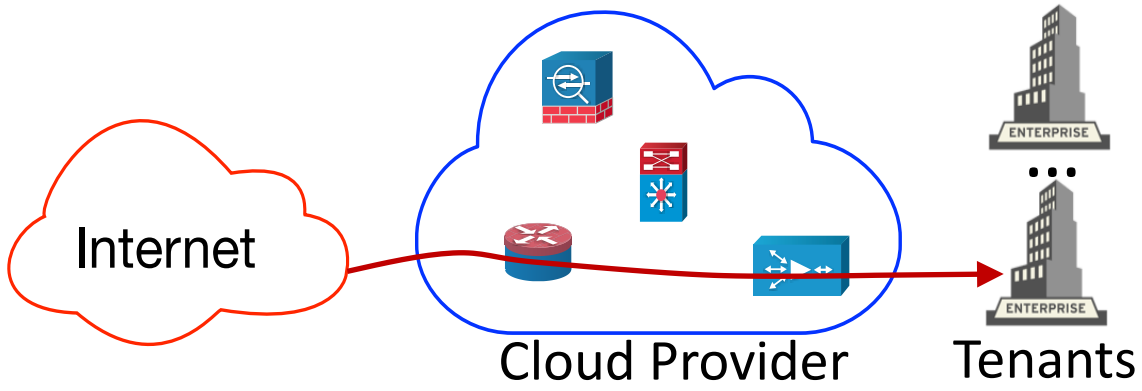
VNF Chain Allocation and Management at Datacenter Scale



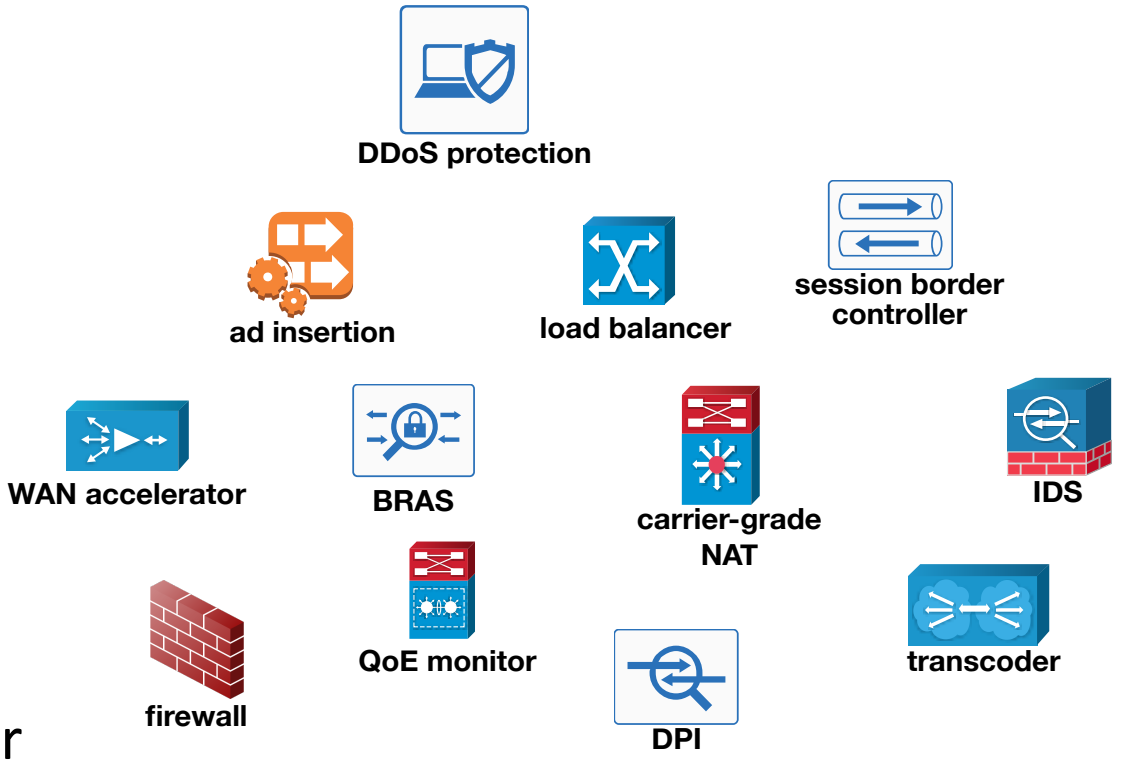
Nodir Kodirov
Sam Bayless, Fabian Ruffy, Swati Goswami
Ivan Beschastnikh, Holger Hoos, Alan Hu, Margo Seltzer

THE UNIVERSITY OF BRITISH COLUMBIA

Universiteit Leiden

Network Functions (NF) are useful and widespread

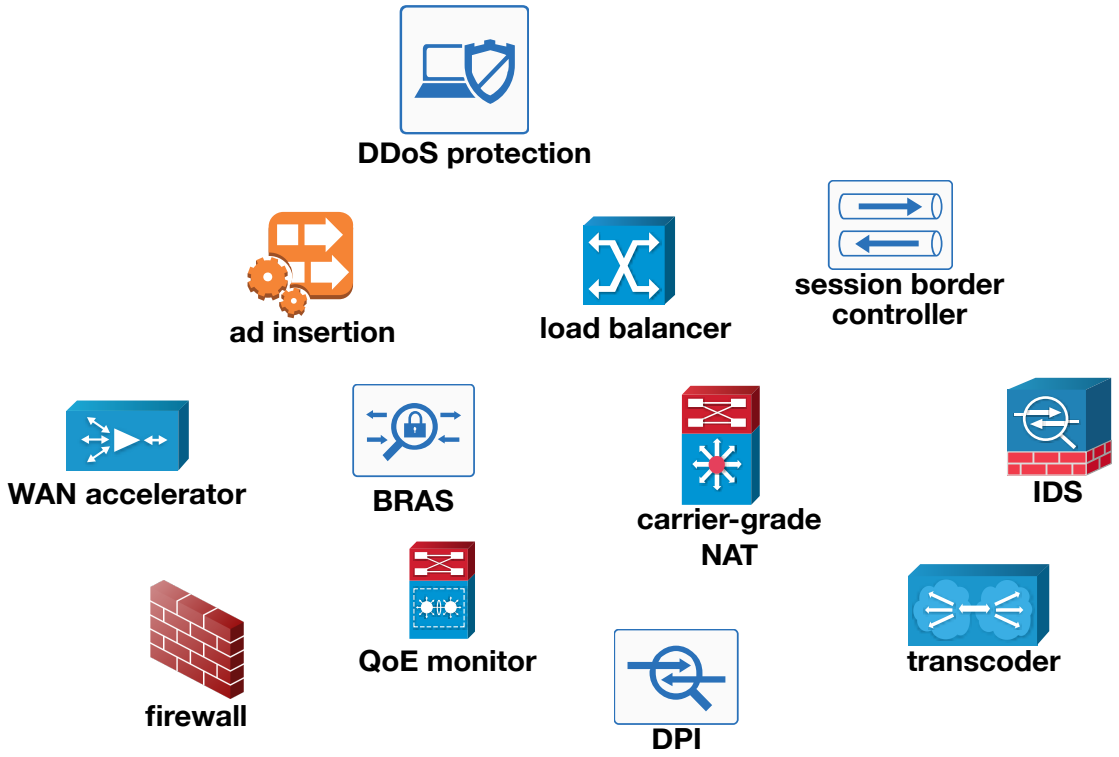
- Security
  - Firewall, DDoS protection, DPI
- Monitoring
  - QoE monitor, Network Stats
- Services
  - Ad insertion, Transcoder
- Network optimization
  - NAT, Load-balancer, WAN accelerator



middleboxes ≈ # L2/L3 devices [Sherry et al. SIGCOMM'12]

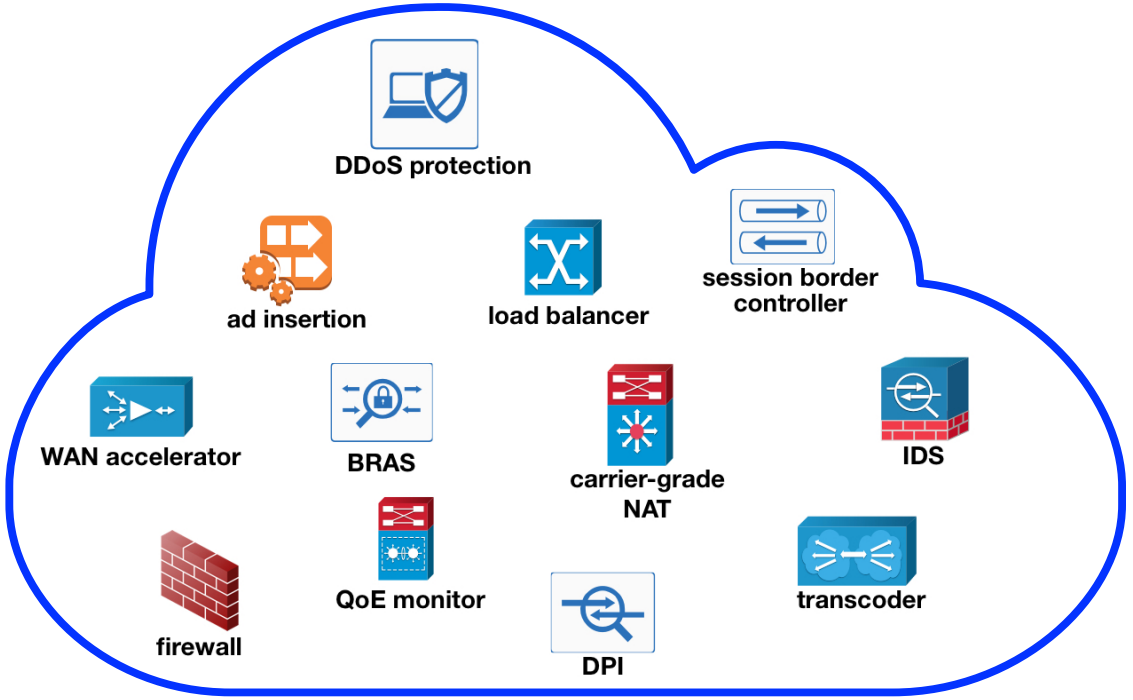
Benefits of Virtualized Network Functions (VNF)

- Elasticity
  - Quick scale up and down NFs
- Fast upgrades
  - No need to wait for new hardware
- Quick configuration, recovery
  - Failover to the backup NF instance
- Outsourcing

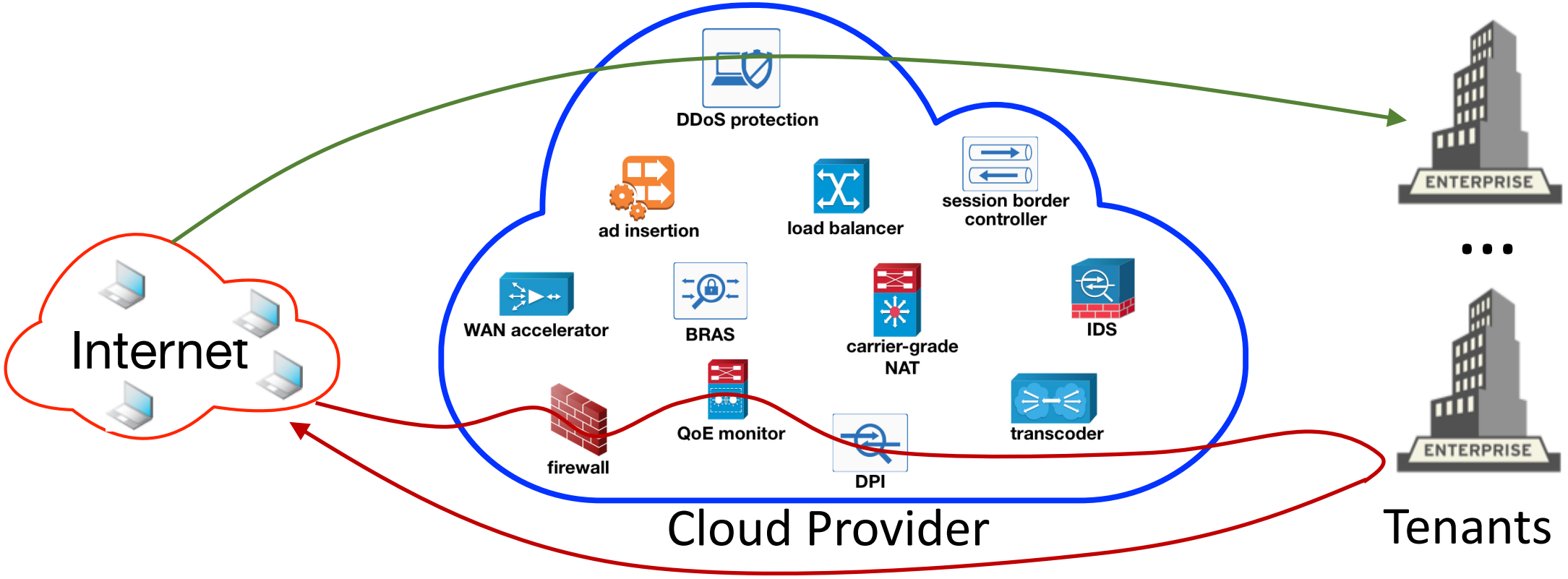


Sherry et al. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service, SIGCOMM'12
Rajagopalan et al., Split/Merge: System Support for Elastic Execution in Virtual Middleboxes, NSDI'13
Martins et al., ClickOS and the Art of Network Function Virtualization, NSDI'14

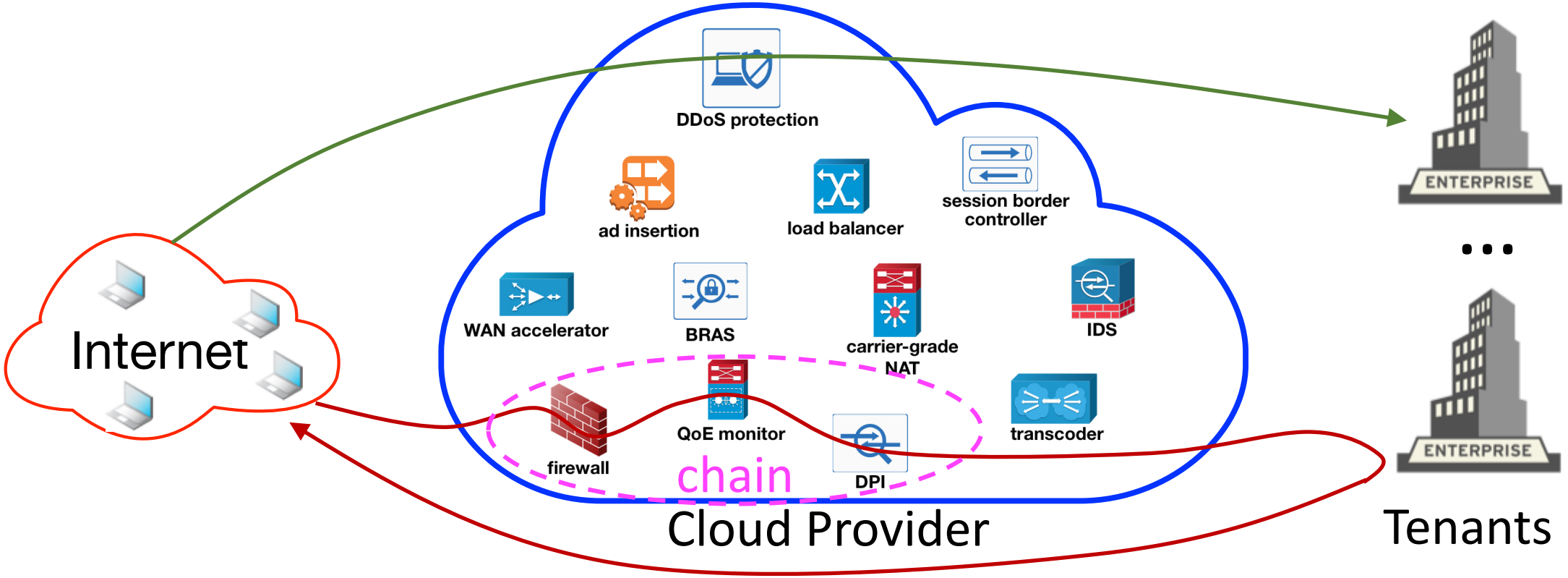
Outsourcing VNFs to the Cloud



Outsourcing VNFs to the Cloud



Outsourcing VNF Chains to the Cloud



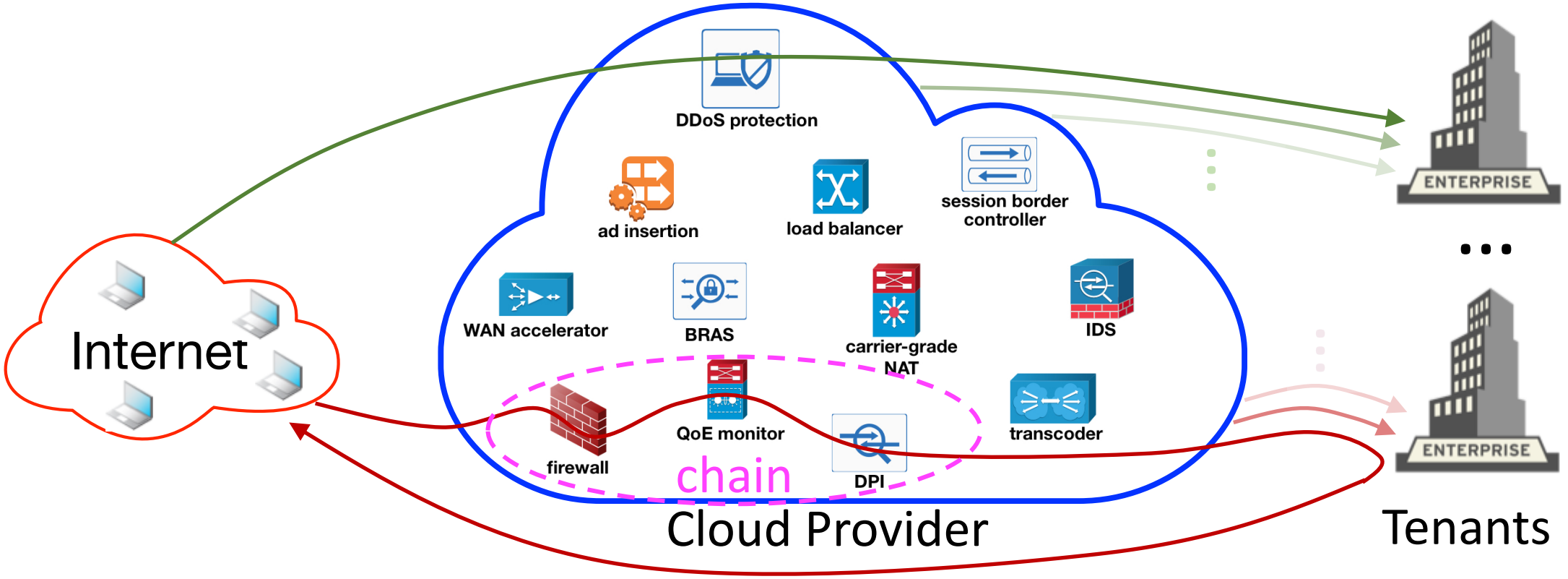
zscaler® The Cloud Security Company

CLOUDFLARE

Challenges of outsourcing VNF Chains

How can tenants allocate and manage their VNF chains?

How can cloud providers achieve high datacenter utilization?

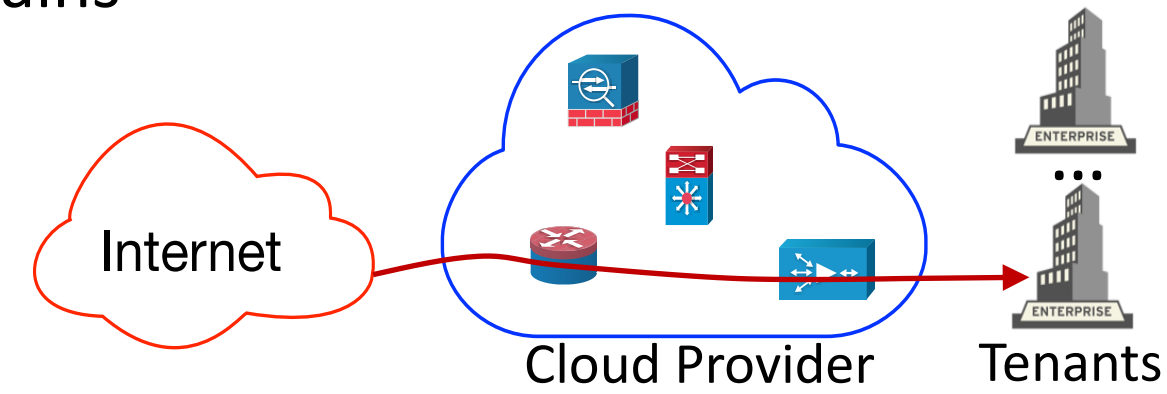


Our contributions: API and algorithm

How can tenants allocate and manage their VNF chains?

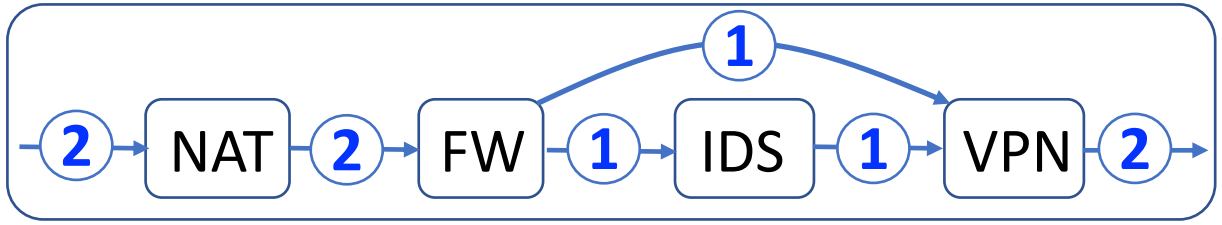
How can cloud providers achieve high datacenter utilization?

- API to allocate and manage VNF chains
- Three algorithms
  - implement the API, and
  - achieve high datacenter utilization
- Evaluation
  - simulate: in datacenter scale with 1000+ servers
  - Daisy: emulate chain management at rack-scale
- Ongoing work: chain abstraction



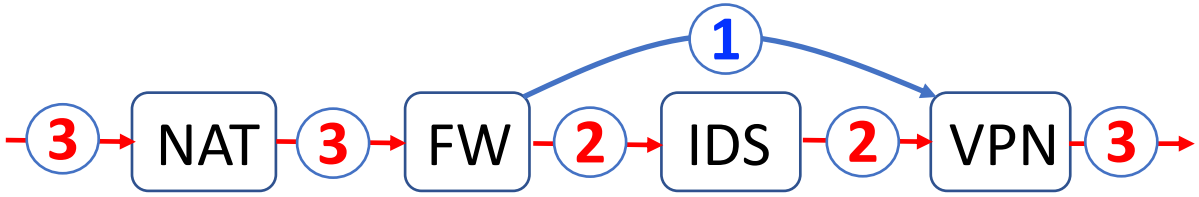
N. Kodirov et al, VNF Chain Allocation and Management at Data Center Scale, ANCS 2018

VNF Chain: six API with use-cases



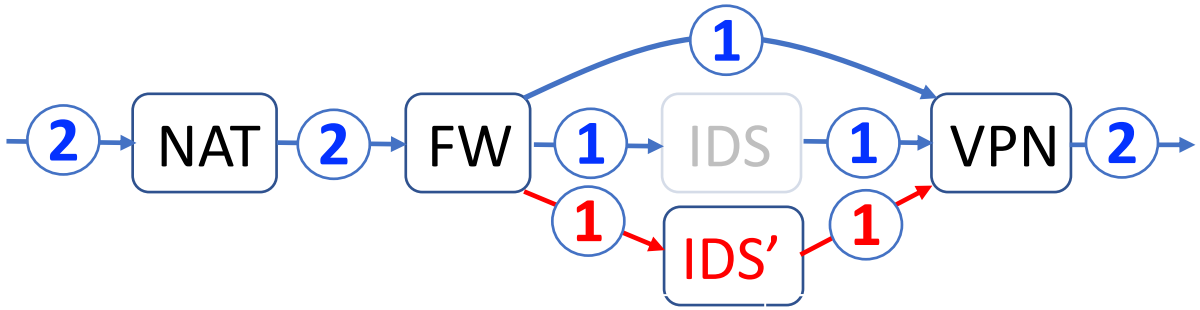
Initial chain

cid ← allocate-chain(C, bw)
add-link-bandwidth(a, b, bw, cid)
add-node(f, cid)



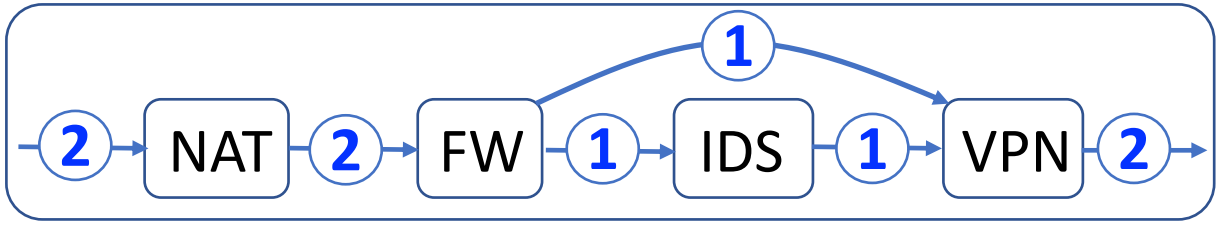
Chain scale-out

remove-link-bandwidth(a, b, bw, cid)
remove-node(f, cid)
remove-e2e-bandwidth(cid, bw)



Element upgrade

VNF Chain: six API with use-cases



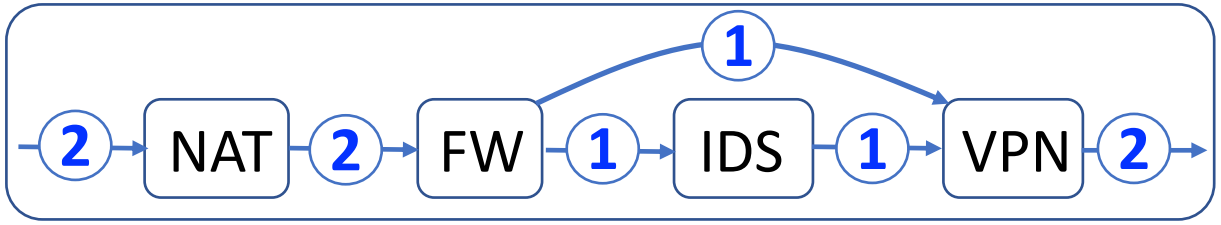
Initial chain

A graph can be transformed arbitrarily by manipulating individual nodes and edges.

Chain scale-out Element upgrade Chain expand …

N. Kodirov et al, VNF Chain Allocation and Management at Data Center Scale, ANCS 2018

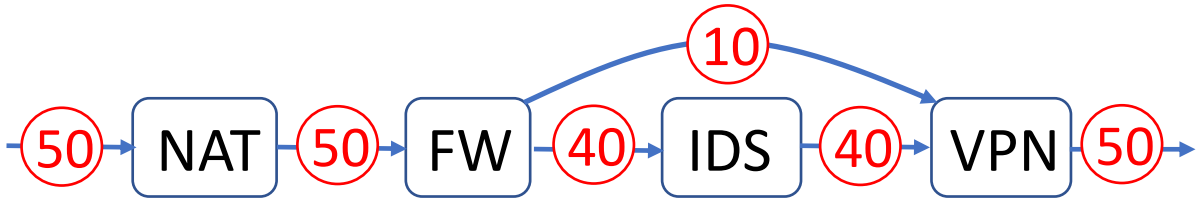
Scale-out beyond single physical resource capacity



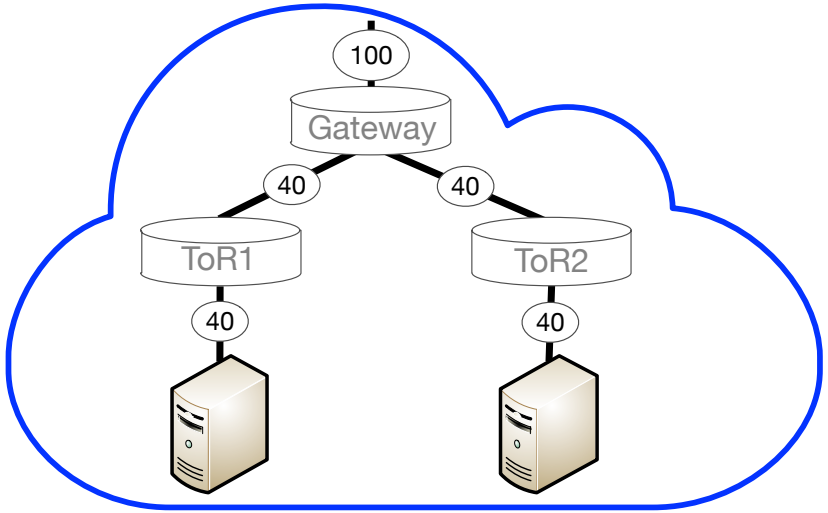
Initial chain

cid ← allocate-chain(C, bw)
add-link-bandwidth(a, b, bw, cid)
add-node(f, cid)

remove-link-bandwidth(a, b, bw, cid)
remove-node(f, cid)
remove-e2e-bandwidth(cid, bw)

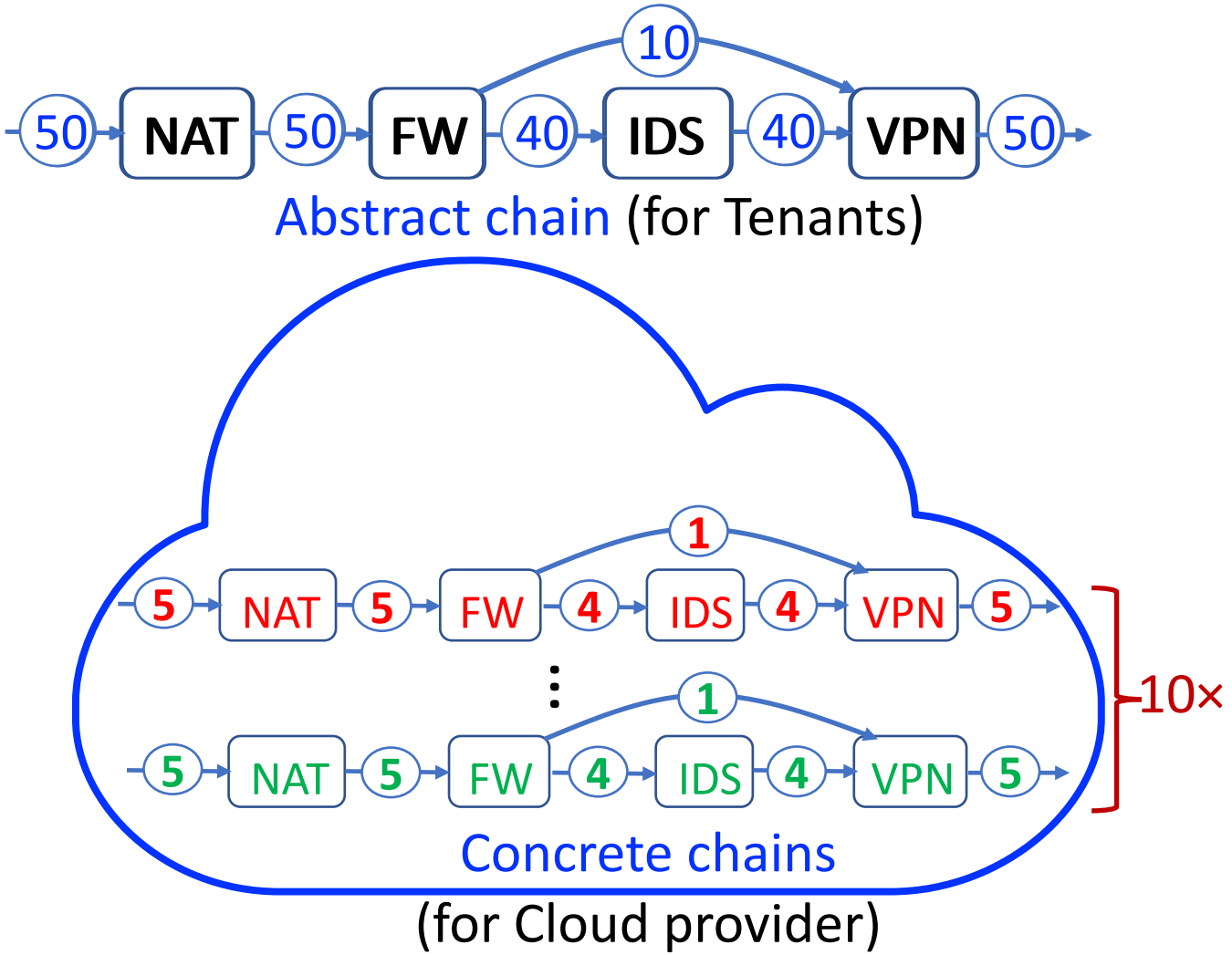


Chain scale-out



Chain Abstraction: Abstract-Concrete VNF Chains

- Abstract VNF chain
  - what tenant requires to allocate and operates on
- Concrete VNF chain
  - cloud provider's implementation of the abstract chain
- Chains abstraction advantages
  - facilitates high DC utilization
  - improves SLA guarantees

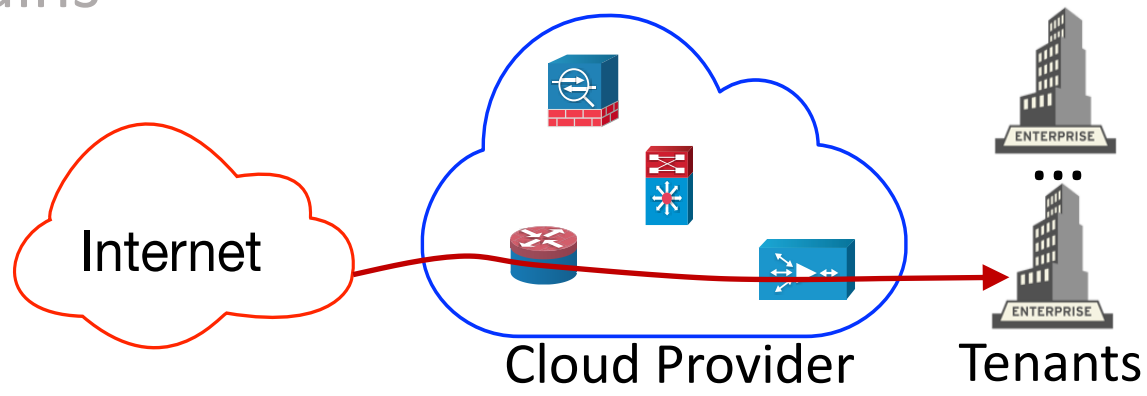


Our contributions: API and algorithm

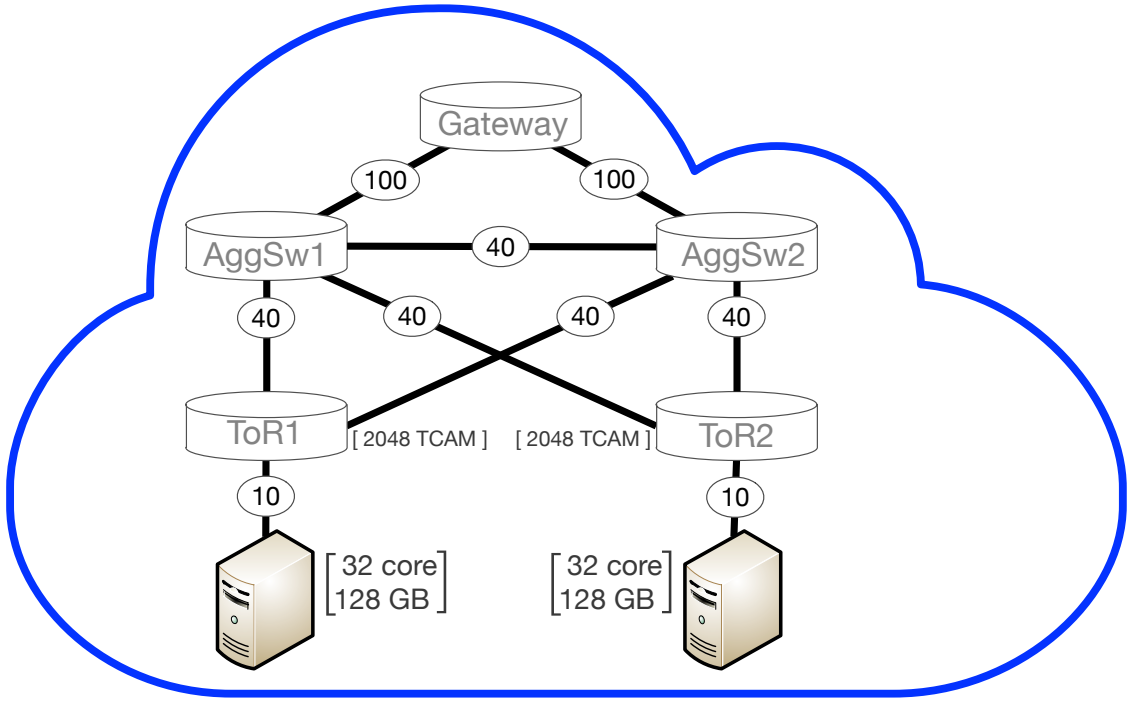
How can cloud providers achieve high datacenter utilization?

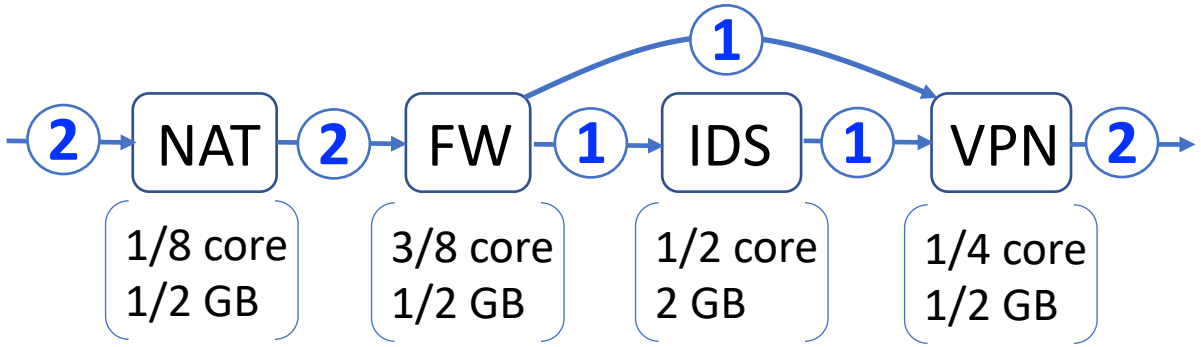
How can tenants allocate and manage their VNF chains?

- API to allocate and manage VNF chains
- Three algorithms
  - implement the API, and
  - achieve high datacenter utilization
- Evaluation
  - simulate: in datacenter scale with 1000+ servers
  - Daisy: emulate chain management at rack-scale
- Ongoing work: chain abstraction



Algorithm inputs: DC topology and chain

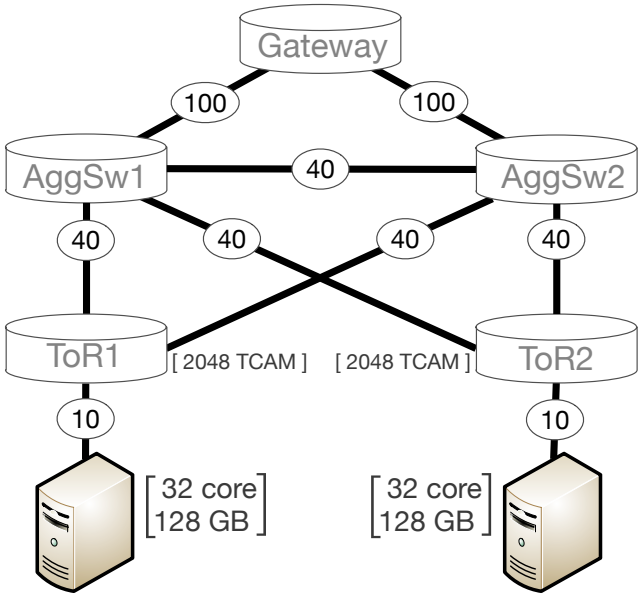


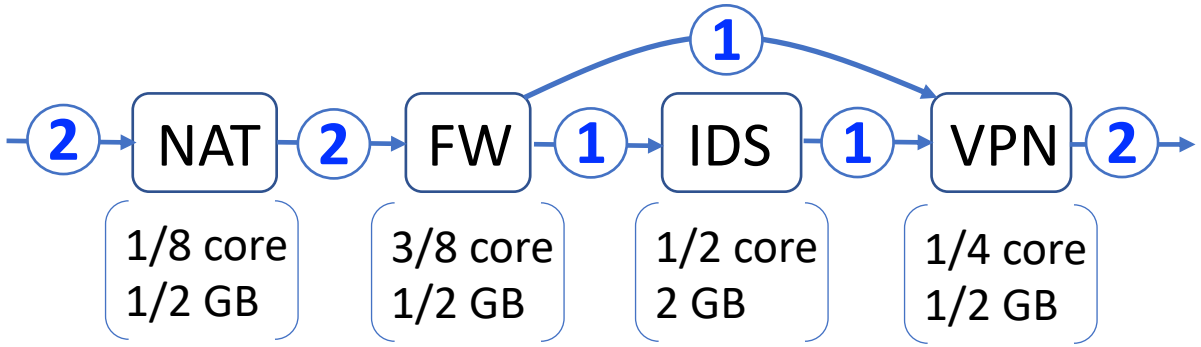


Expected resource consumption per Gbps of traffic
(see the paper for VNF profile generation)

Palkar et al., E2: A Framework for NFV Applications, SOSP'15
Naik et al., NFVPerf: Online performance monitoring and bottleneck detection for NFV, IEEE NFV-SDN 2016.
Nam et al., Probius: Automated Approach for VNF and Service Chain Analysis in Software-Defined NFV, SOSR'18

Algorithms for Chain Allocation and Management

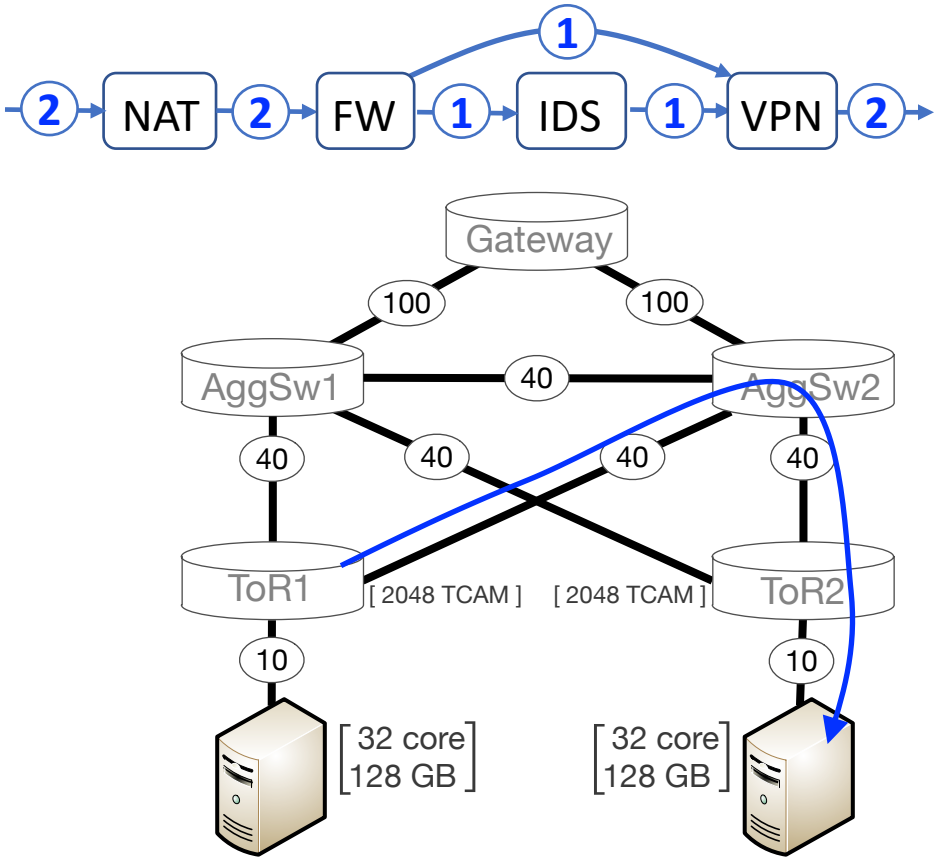




Expected resource consumption per Gbps of traffic
(see the paper for VNF profile generation)

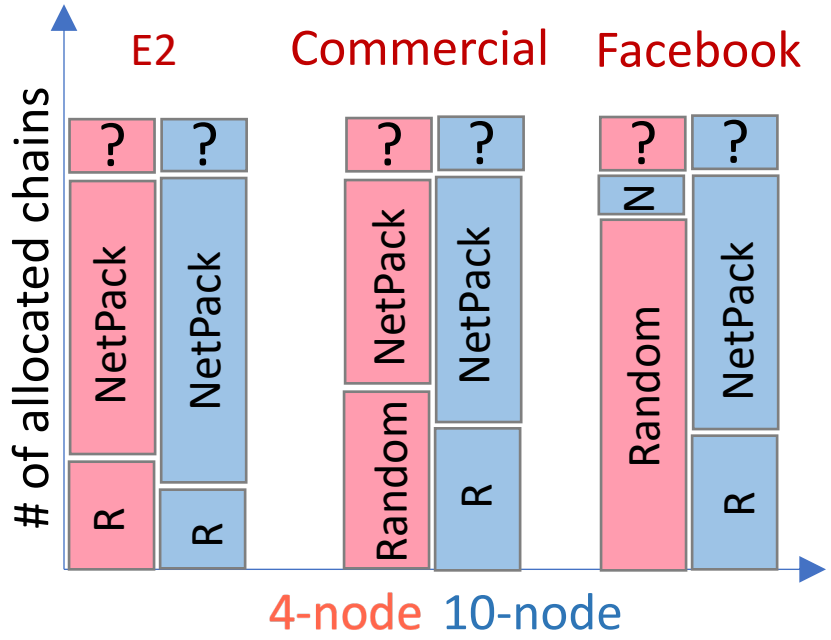
Algorithms for Chain Allocation and Management

- Random (baseline)
  - Consider NFs and servers/switches in random order
  - Attempt the above step *n* times (e.g., n=100)
  - Choose the shortest path between chain NFs



Algorithms for Chain Allocation and Management

- Random (baseline)
  - Consider NFs and servers/switches in random order
  - Attempt the above step *n* times (e.g., n=100)
  - Choose the shortest path between chain NFs
- NetPack: Random + 3 simple heuristics
  - Consider the chain NFs in a topological order
  - Re-use the same server when allocating consecutive NFs
  - Gradually increase the network scope: rack, cluster, etc.
- VNFSolver: how optimal is NetPack?
  - Constraint-solver based chain allocation algorithm
  - Slow, but complete: finds a solution when one exists



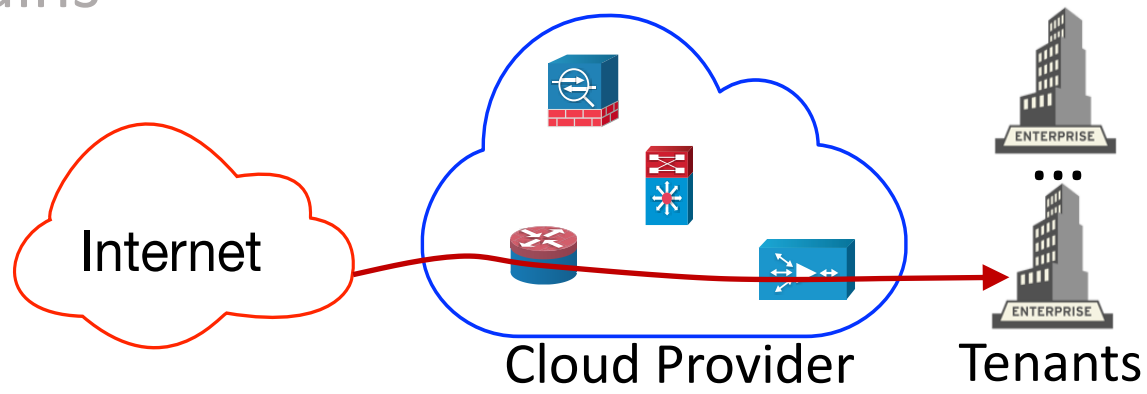
Palkar et al., E2: A Framework for NFV Applications, SOSP'15
Bayless et al., SAT Modulo Monotonic Theories, AAAI'15

Our contributions: API and algorithm

How can cloud providers achieve high datacenter utilization?

How can tenants allocate and manage their VNF chains?

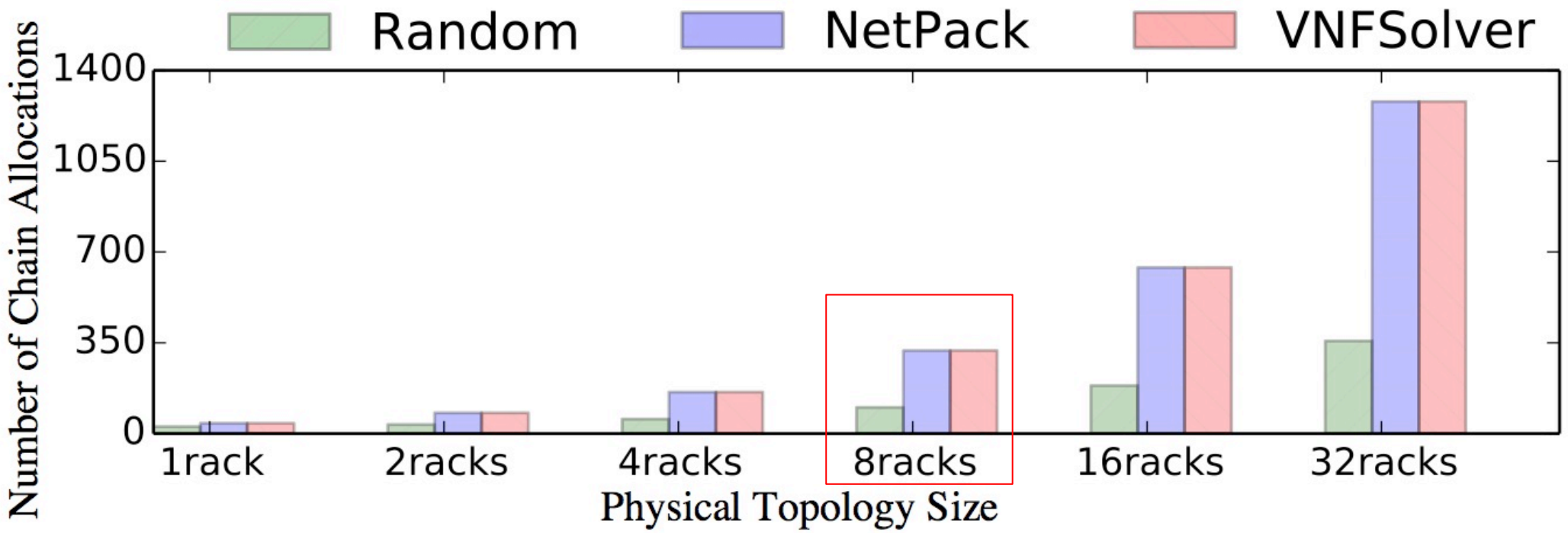
- API to allocate and manage VNF chains
- Three algorithms
  - implement the API, and
  - achieve high datacenter utilization
- Evaluation
  - simulate: in datacenter scale with 1000+ servers
  - Daisy: emulate chain management at rack-scale
- Ongoing work: chain abstraction

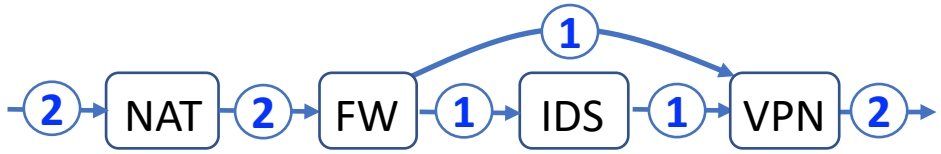


Evaluation: Objectives

- How good is the datacenter utilization?
  - Evaluate Random, NetPack, and VNFSolver
  - Consider three different datacenter topologies
  - Use five different VNF chains with varying length (2-10)
- How fast is chain allocation?
  - Measure time it takes to saturate the datacenter
- Does API reliably implement the use-cases?
  - Prototype scale-out and chain upgrade in Daisy
  - Use two different racks, two sources of packet traces

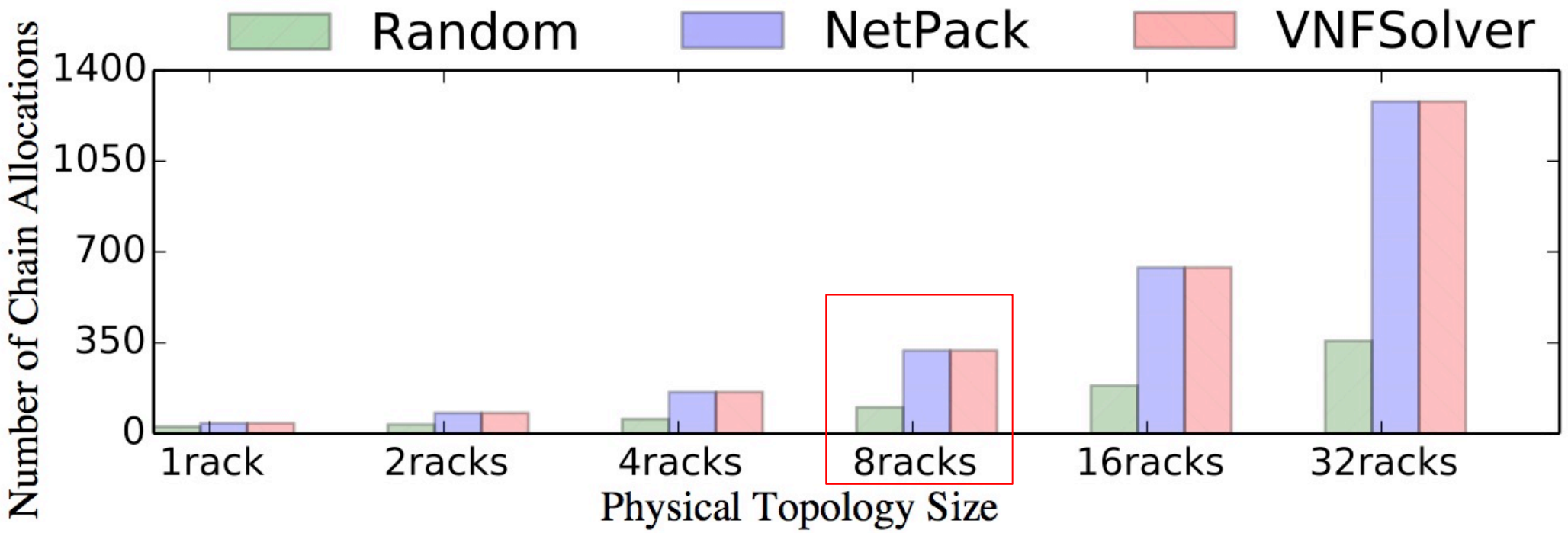
Datacenter utilization evaluation





Palkar et al., E2: A Framework for NFV Applications, SOSP'15

Datacenter utilization evaluation



NetPack achieves at least 96% of VNFSolver allocations.
Chain allocation time: Random ≲ NetPack ≪ VNFSolver.

Palkar et al., E2: A Framework for NFV Applications, SOSP'15

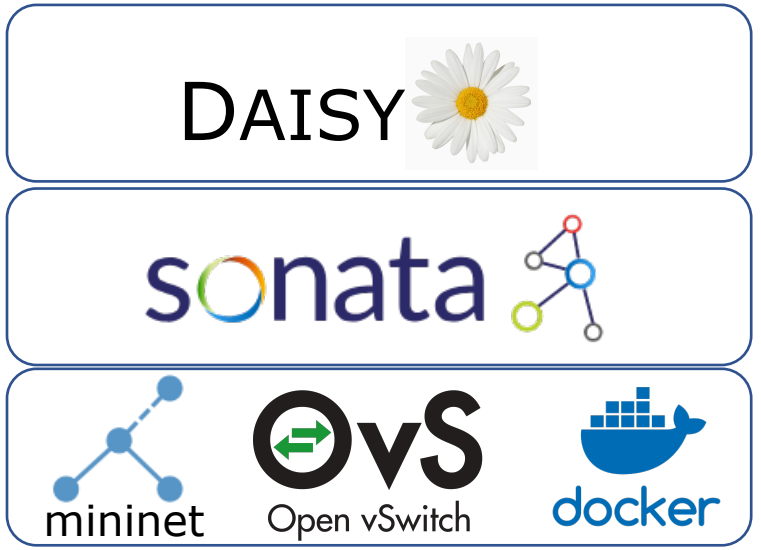
NetPack Utilization and Speed

NetPack achieves at least 96% of VNFSolver allocations while being 82x faster than VNFSolver (optimal) and only up to 54% slower (per chain) than Random (baseline) on average.

Qualitatively similar results with Facebook and Commercial DC topologies with chains of up to 10 nodes.
(see the paper for details)

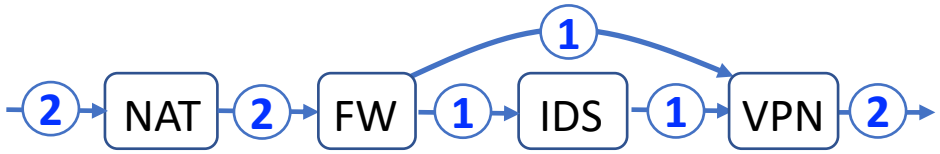
Feasibility check: does API work?

- Daisy builds on Sonata framework
  - Mininet to build DC topology
  - OVS for switches, and Dockers for NFs
- Runs on a single Azure VM
  - 64 cores, 432 GB RAM
- Emulates use-cases and chain arrivals
  - scale-out and upgrade use-cases
  - continuous arrival of tenant chains in rack-scale

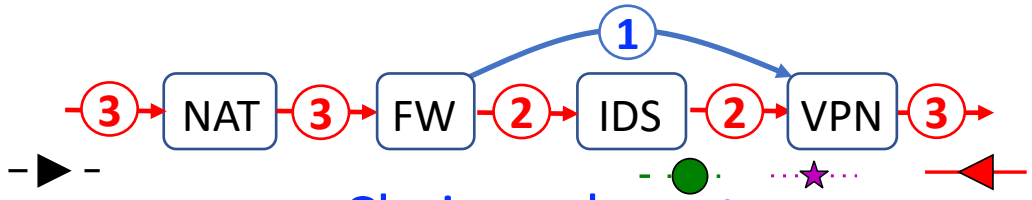


Peuster et al., Sonata NFV SDK, github.com/sonata-nfv/son-emu, 2017

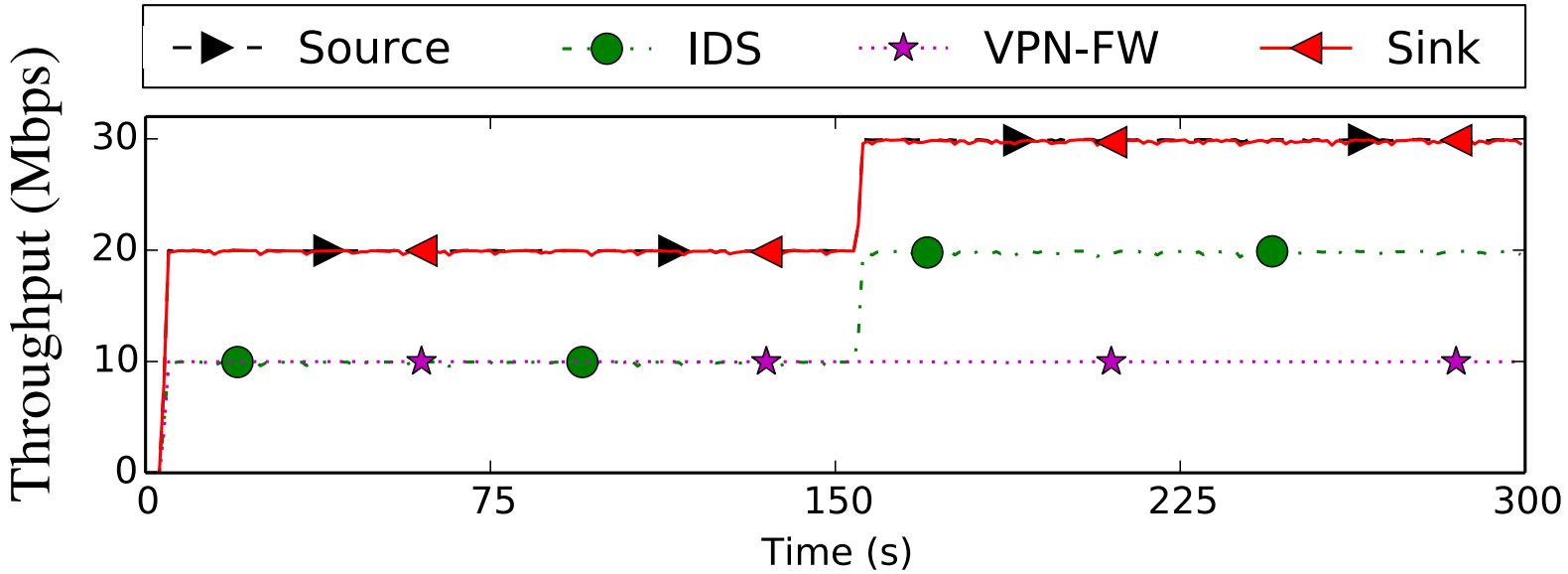
VNF Chain scale-out



Initial chain

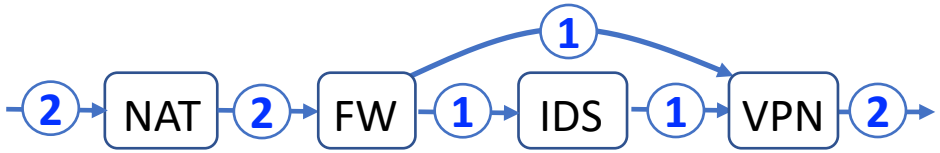


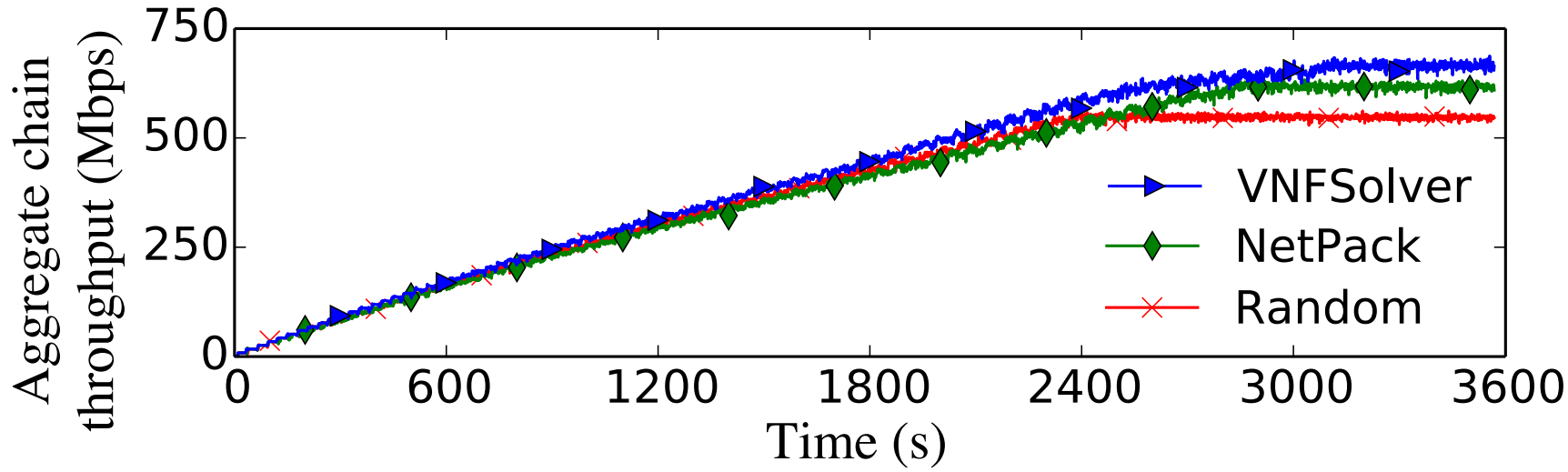
Chain scale-out



Daisy implements scale-out with no packet drops.

Daisy: emulate continuous chain arrival



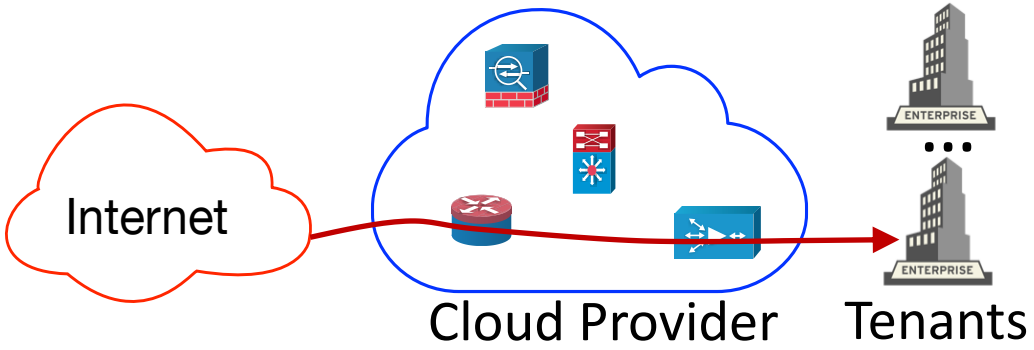


VNFSolver allocated 75 chains (687 Mbps)
NetPack allocated 67 chains (633 Mbps)
Random allocated 61 chains (561 Mbps)
(throughput with iperf generated packets is precise)

Daisy Contributions

Daisy implements scale-out with no packet drops and element upgrade with 1s packet drop at most.
We also emulated continuous chain arrival case where different tenants make chain allocation requests one-by-one.

Snapshot of complete work



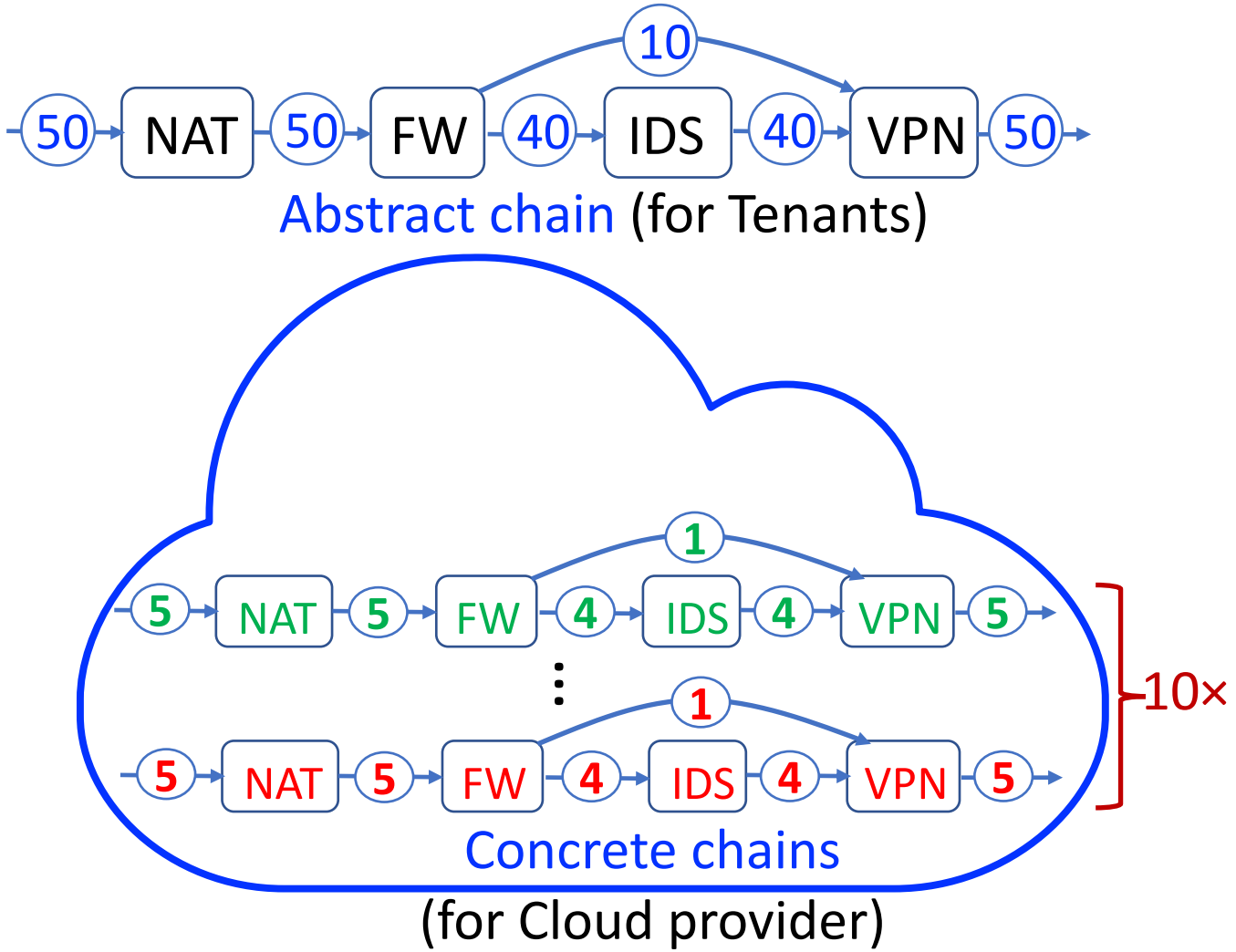
- API with six primitives
  - Implements wide-range of chain operations
  - Chain abstraction facilitates full DC utilization
- NetPack algorithm
  - Handles DC-scale allocation with 1000+ servers
  - Achieves at least 96% allocations of VNFSolver (optimal) while being 82x faster on average
- Daisy prototype
  - Demonstrates feasibility of API and algorithms
- Ongoing work: chain abstraction

How can tenants allocate and manage their VNF chains?

How can cloud providers achieve high data center utilization?

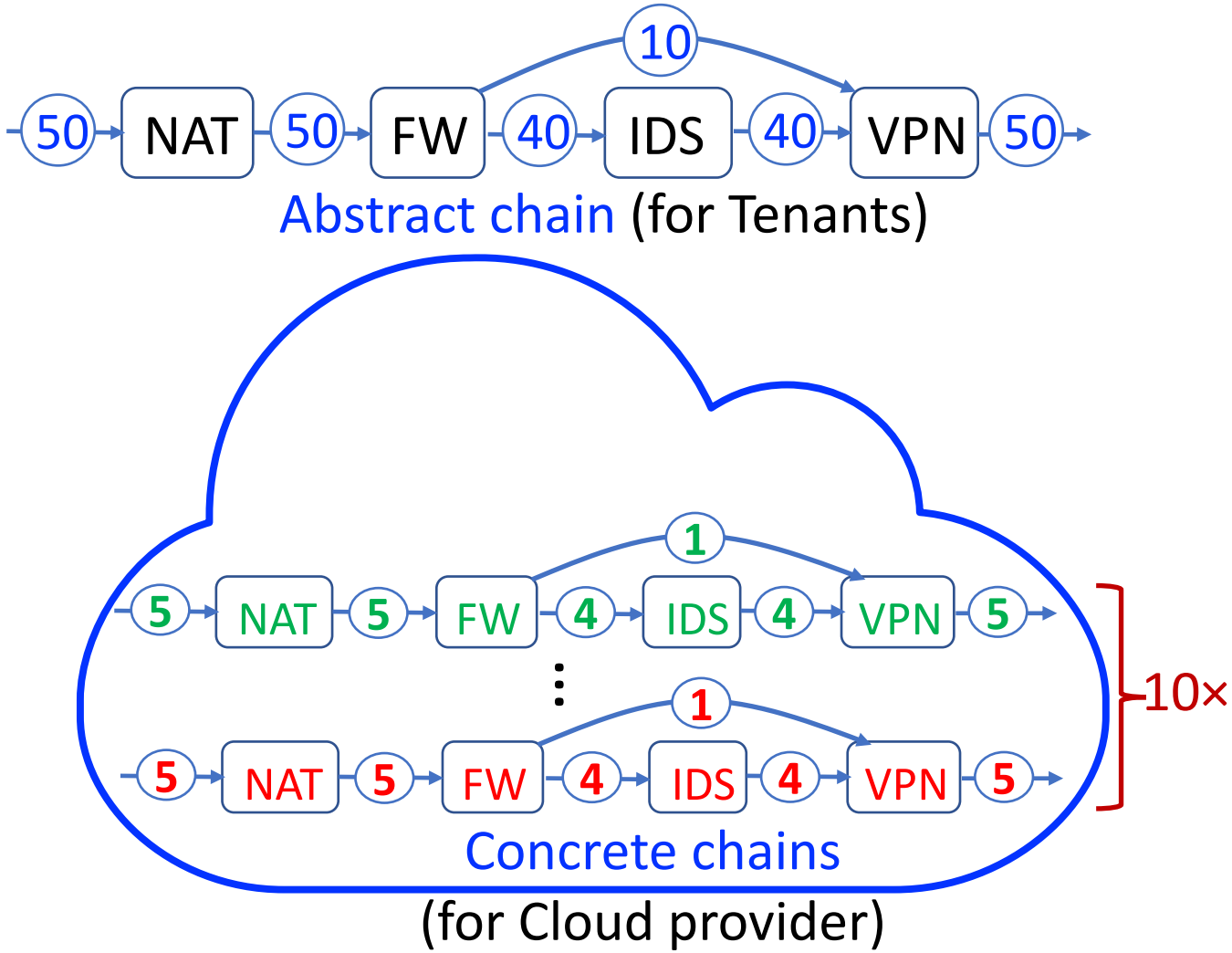
Chain abstraction challenges

- Abstract VNF chain
  - what tenant requires to allocate and operates on
- Concrete VNF chain
  - cloud provider's implementation of the abstract chain
- Chains abstraction advantages
  - facilitates high DC utilization
  - improves SLA guarantees
- Challenges
  - low-latency, packet loss, state synchronization, efficiency loss, hotspots

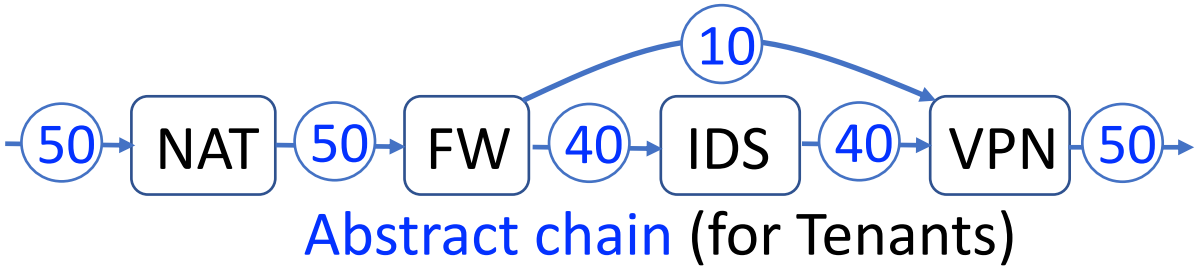


Kodirov et al., VNF Chain Abstraction for Cloud Service Providers, ANCS'18 poster

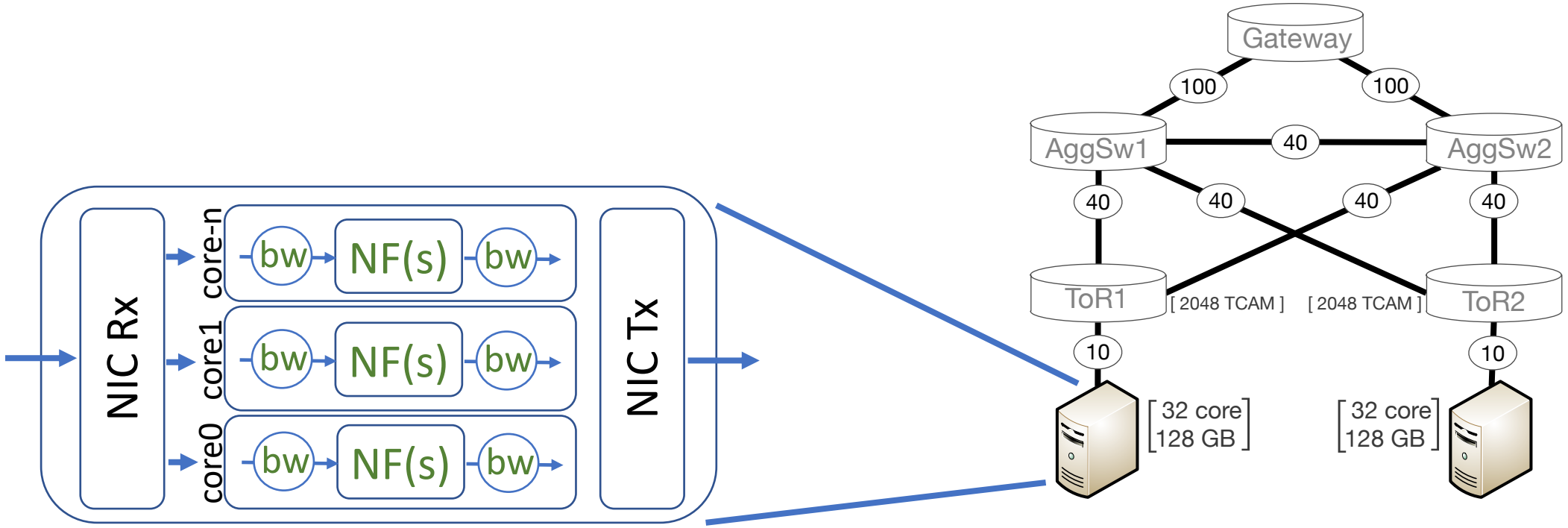
Problem: Hotspots



Problem: Hotspots



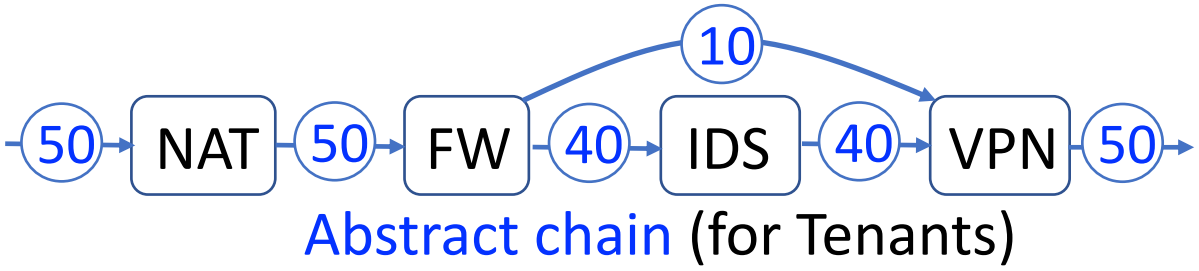
Abstract chain (for Tenants)

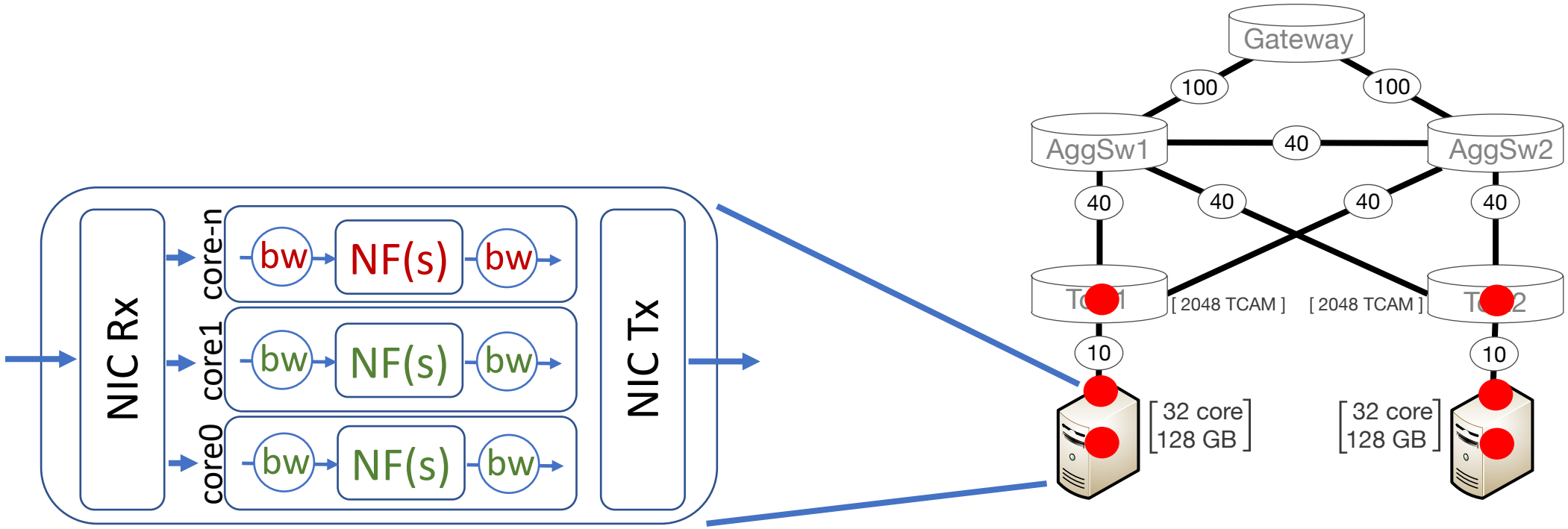


Problem: Hotspots

- Hotspots in different layers:

  CPU cores, NIC ports, ToR switch ports, etc.

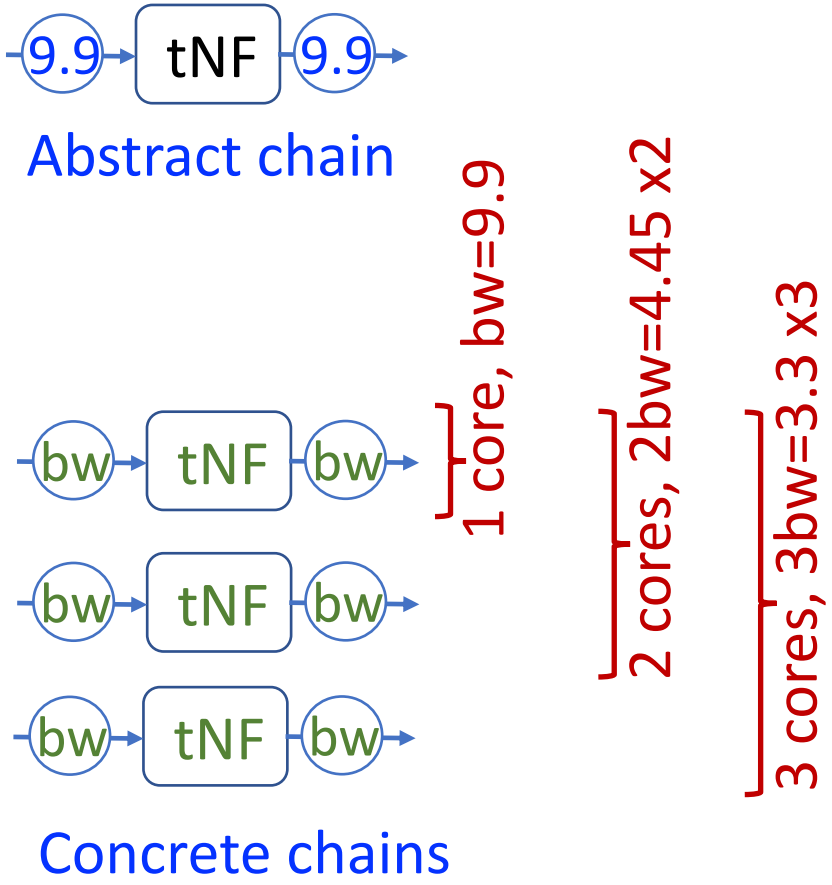




Sadok et al., A Case for Spraying Packets in Software Middleboxes, HotNets'18

Microbenchmarks with up to 3 chains

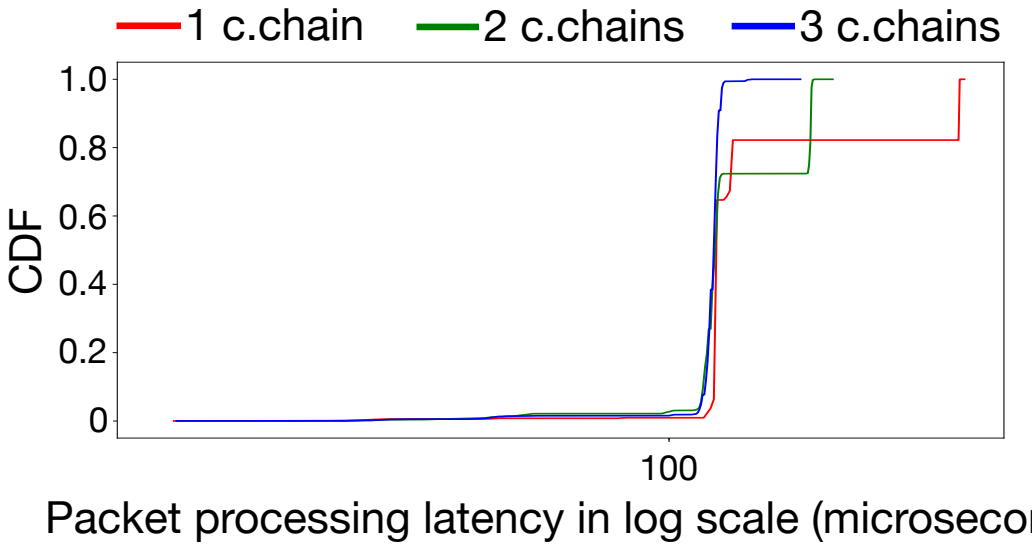
- tNF: tunable NF
  - computes N prime numbers, per packet (<50K in our experiments)
  - Throughput: 0.83 Mpps (9900 Mbps with 1500 byte packet)
- NF runs on OpenNetVM
- Varied the number of concrete chains from one to three
- Each concrete chain processes a separate flow (5-tuple)



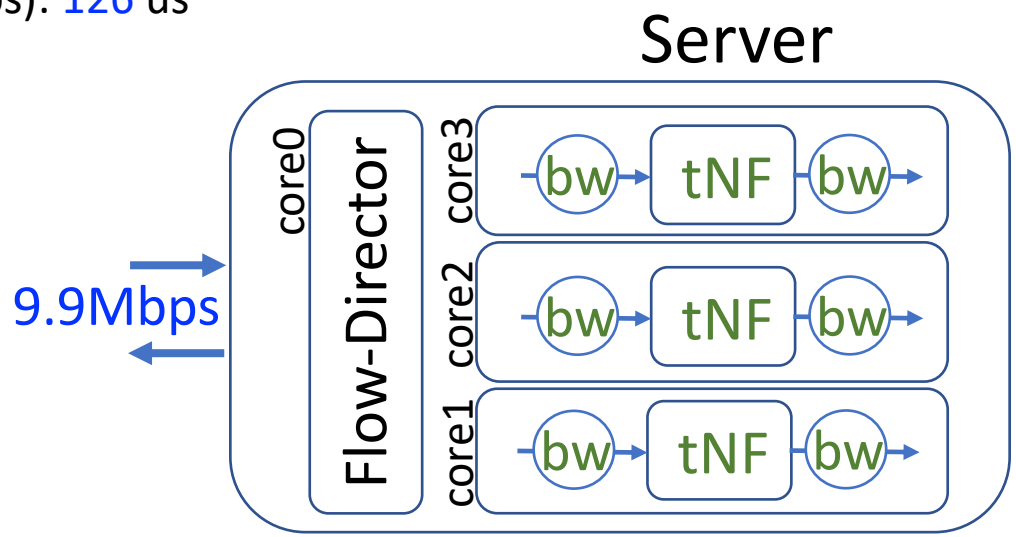
OpenNetVM, github.com/sdnfv/openNetVM, 2016

Microbenchmarks with up to 3 chains

- Tail latencies (99 percentile) in microseconds
  - 1 concrete chain (uses 1 core for 9.9Gbps): 336 us
  - 2 concrete chains (uses 2 cores for 9.9Gbps): 182 us
  - 3 concrete chains (uses 3 cores for 9.9Gbps): 126 us



Abstract chain

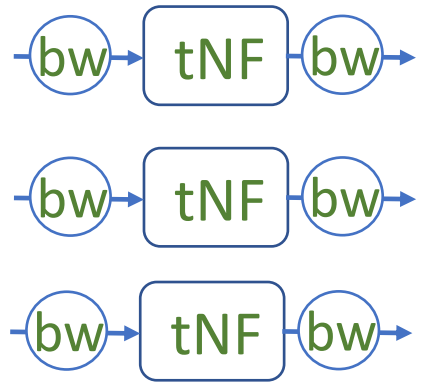


Concrete chains

Observations and Ongoing Work

- Latency grows proportional to the core load
- Need CPU load-aware chain-splitting mechanism
  - N flows to 1 core (N-to-1) for mice flows
  - 1-to-N for elephant flows
- Splitting should happen in multiple levels
  - CPU cores, NIC ports, ToR ports
- Need to support wide-range of NFs
  - Bounded tail latencies are particularly challenging for stateful NFs, such as DPI

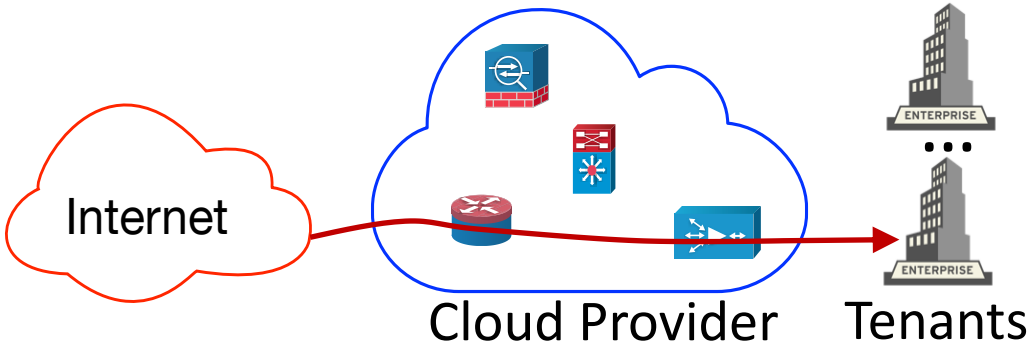
Abstract chain



Concrete chains

Sadok et al., A Case for Spraying Packets in Software Middleboxes, HotNets'18
Khalid and Akella, Correctness and Performance for Stateful Chained Network Functions, NSDI'19

Conclusion



- API with six primitives
  - Implements wide-range of chain operations
  - Chain abstraction facilitates full DC utilization
- NetPack algorithm
  - Handles DC-scale allocation with 1000+ servers
  - Achieves at least 96% allocations of VNFSolver (optimal) while being 82x faster on average
- Ongoing work: chain abstraction
  - Need load-aware chain-splitting mechanism

How can tenants allocate and manage their VNF chains?

How can cloud providers achieve high data center utilization?

Thank you!