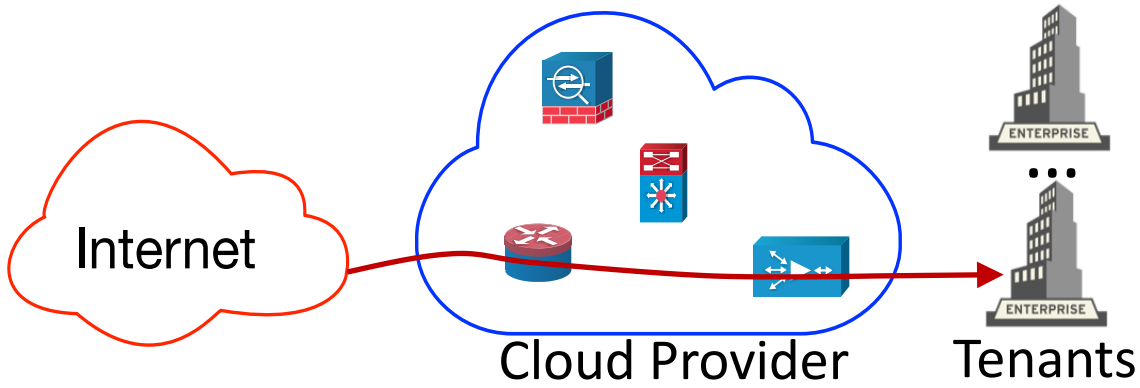
VNF Chain Allocation and Management at Data Center Scale



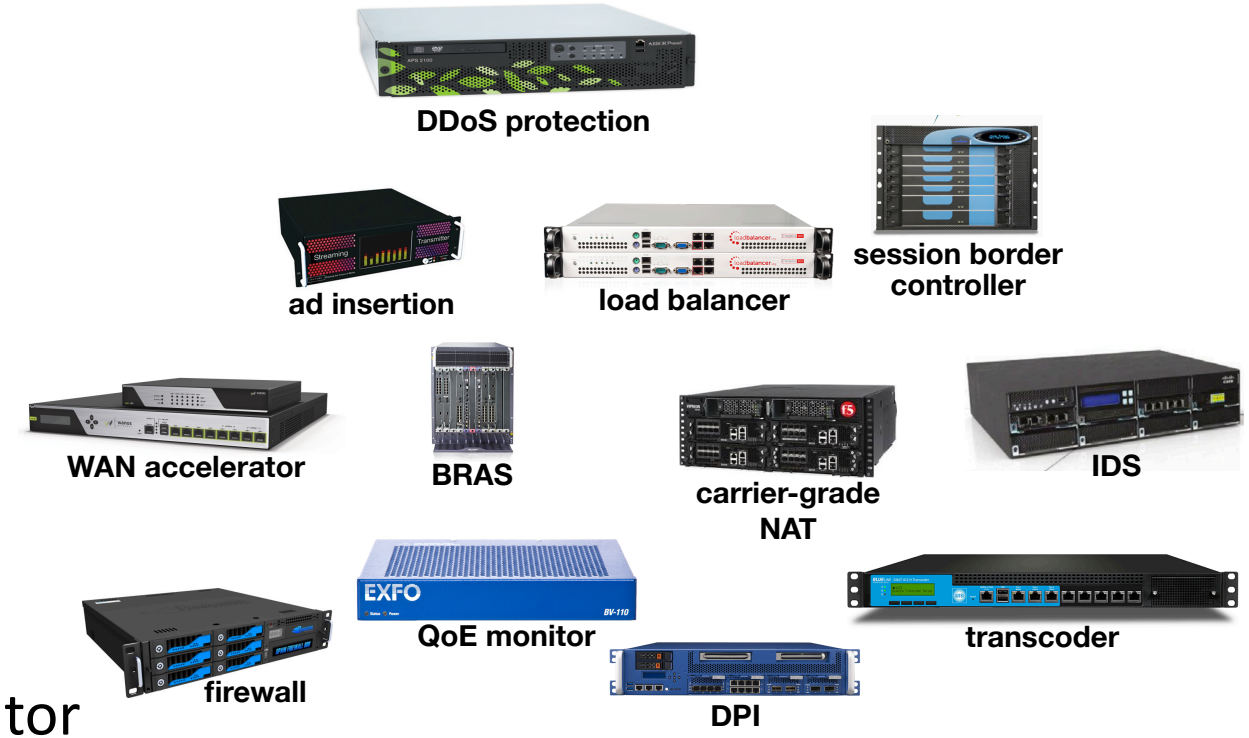
Nodir Kodirov, Sam Bayless, Fabian Ruffy,
Ivan Beschastnikh, Holger Hoos, Alan Hu

THE UNIVERSITY OF BRITISH COLUMBIA

Universiteit Leiden

Network Functions (NF) are useful and widespread

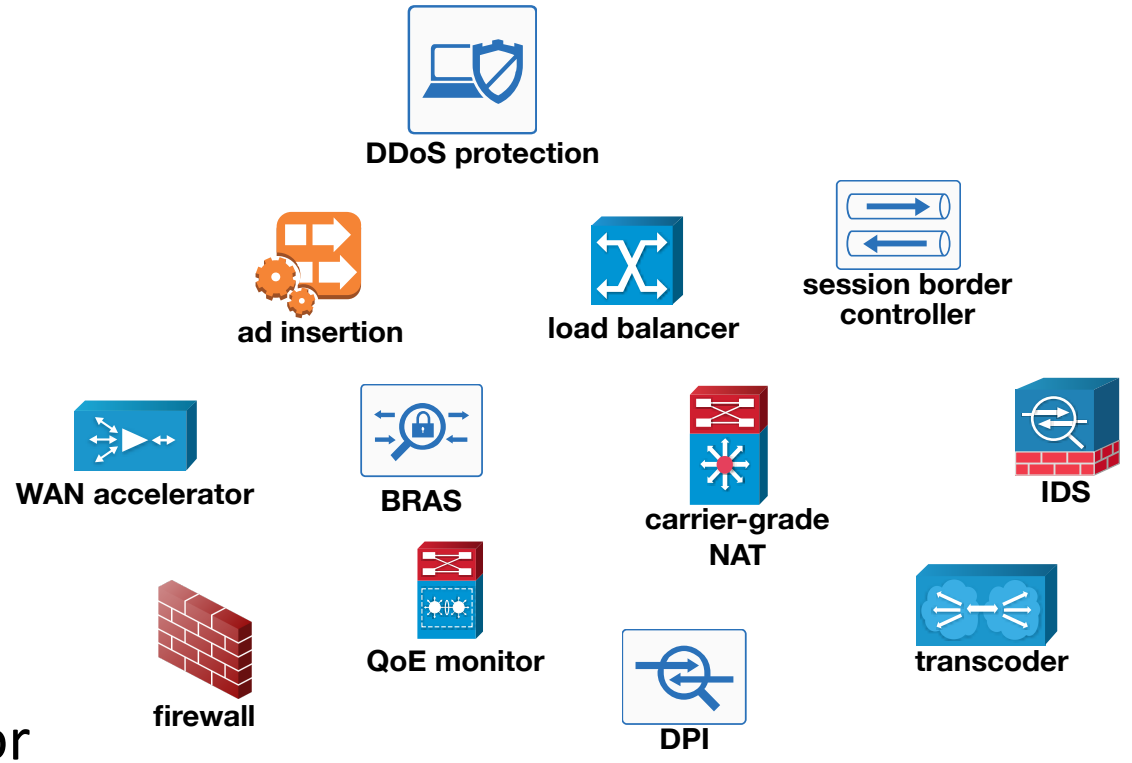
- Security
  - Firewall, DDoS protection, DPI
- Monitoring
  - QoE monitor, Network Stats
- Services
  - Ad insertion, Transcoder
- Network optimization
  - NAT, Load-balancer, WAN accelerator



Sherry et al. find # of middleboxes are ≈ to # of L2/L3 devices in enterprise

Network Functions (NF) are useful and widespread

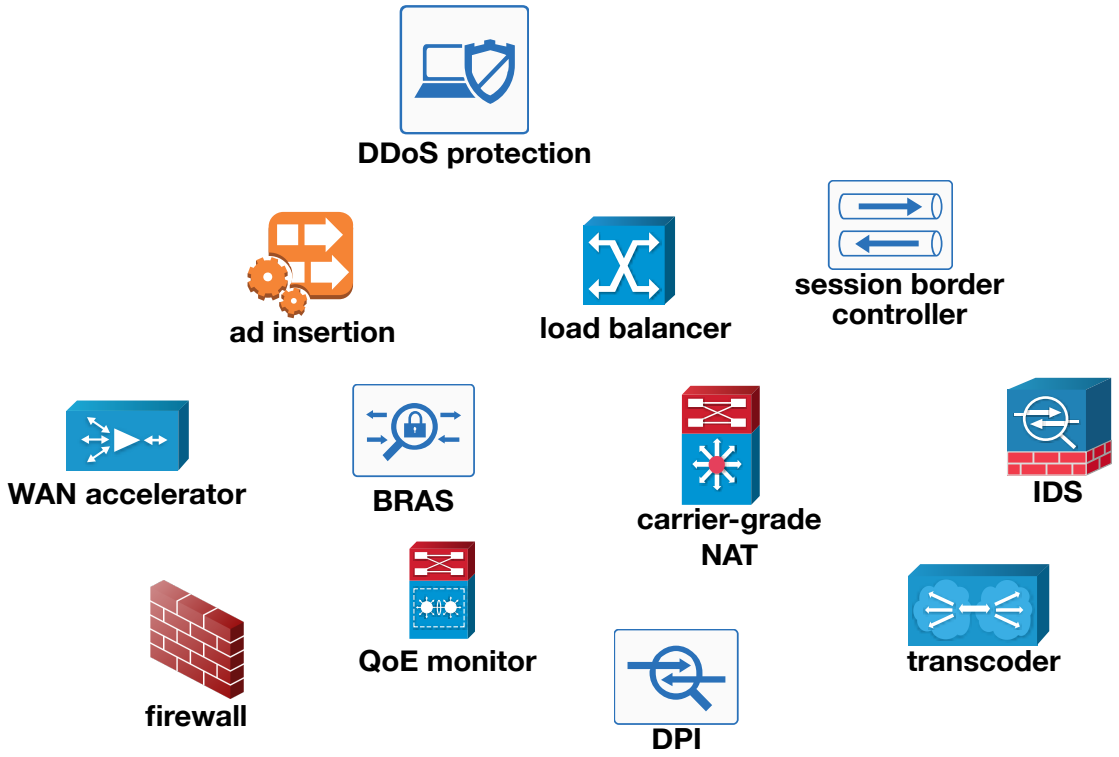
- Security
  - Firewall, DDoS protection, DPI
- Monitoring
  - QoE monitor, Network Stats
- Services
  - Ad insertion, Transcoder
- Network optimization
  - NAT, Load-balancer, WAN accelerator



Sherry et al. find # of middleboxes are ≈ to # of L2/L3 devices in enterprise

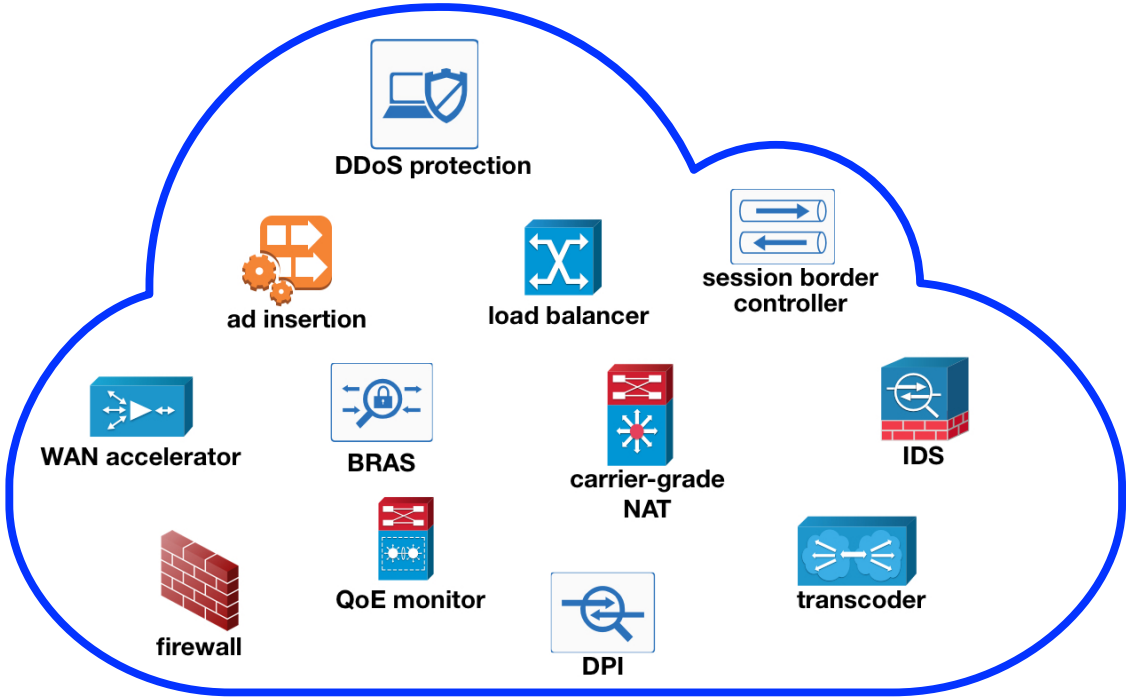
Benefits of Virtualized Network Functions (VNF)

- 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

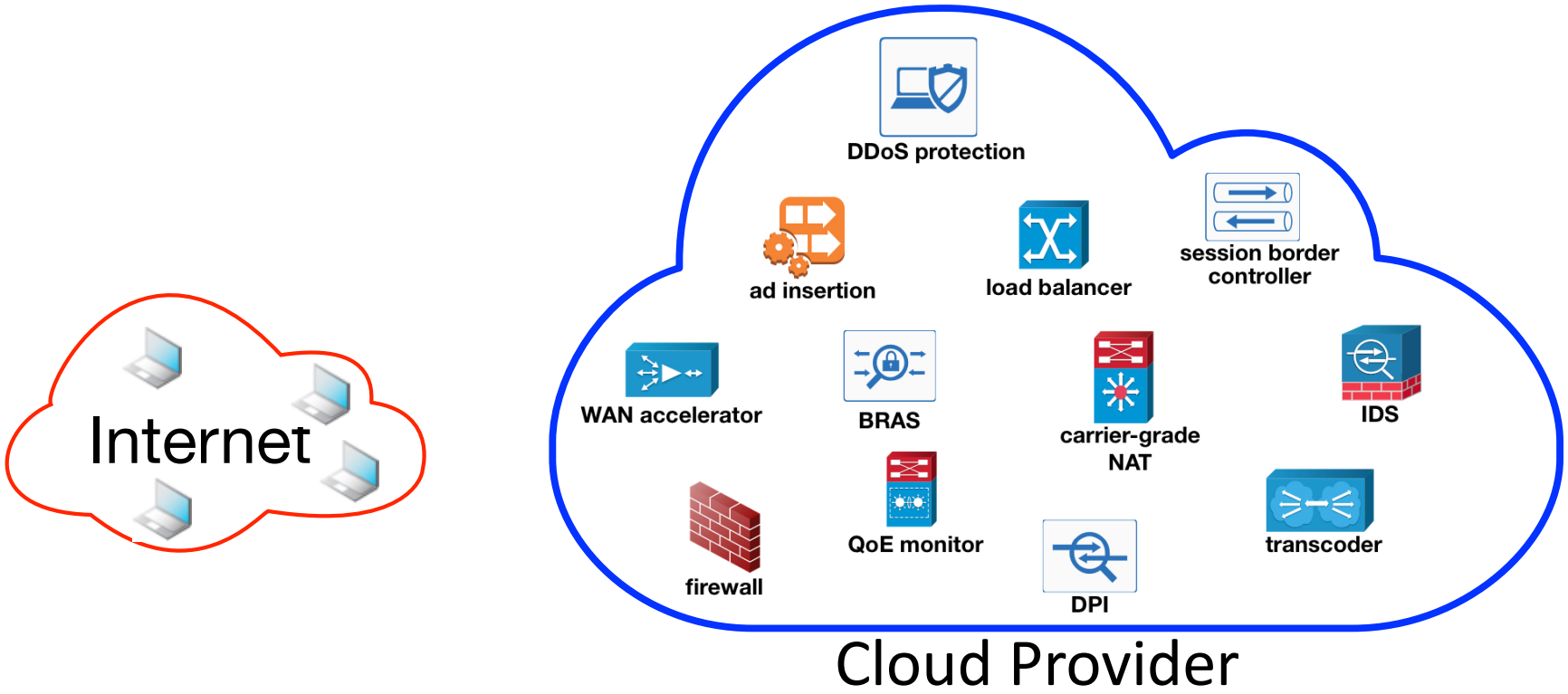


Sherry et al. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service, SIGCOMM'12
Rajagopalan et al., Split/Merge: System Support for Elastic Execution in Virtual Middleboxes, NSDI'13
Martins et al., ClickOS and the Art of Network Function Virtualization, NSDI'14

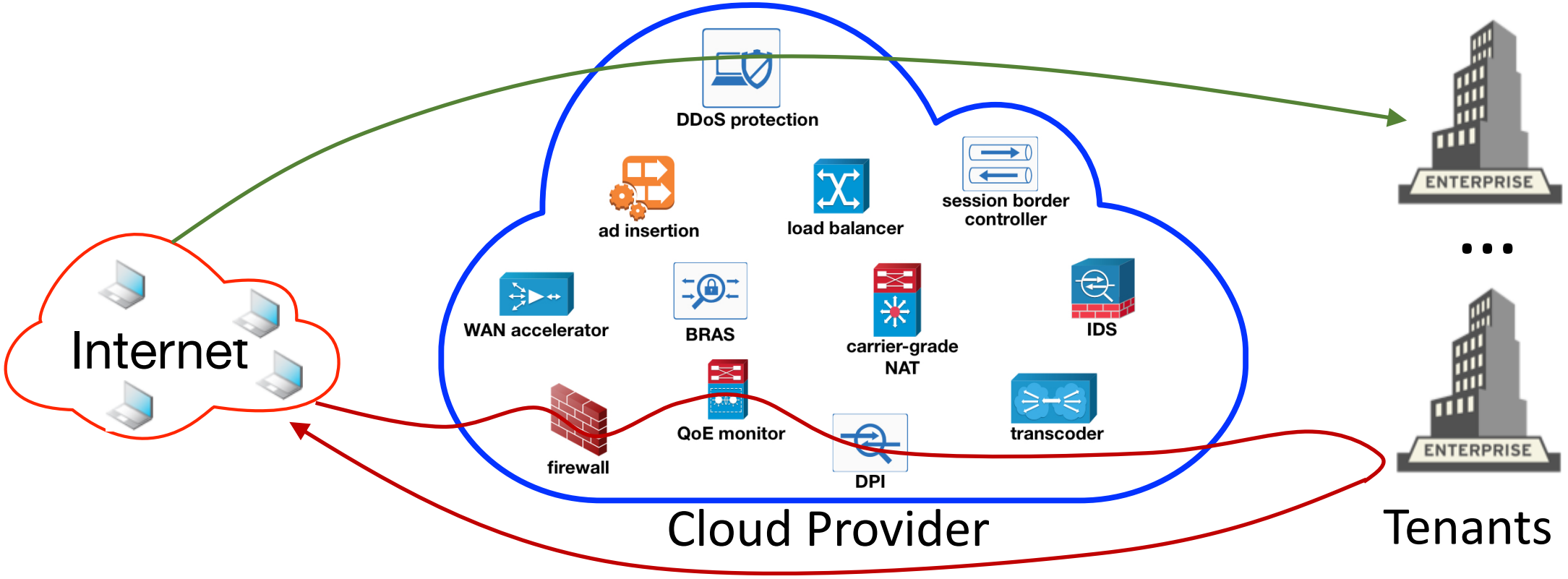
Outsourcing VNFs to the Cloud



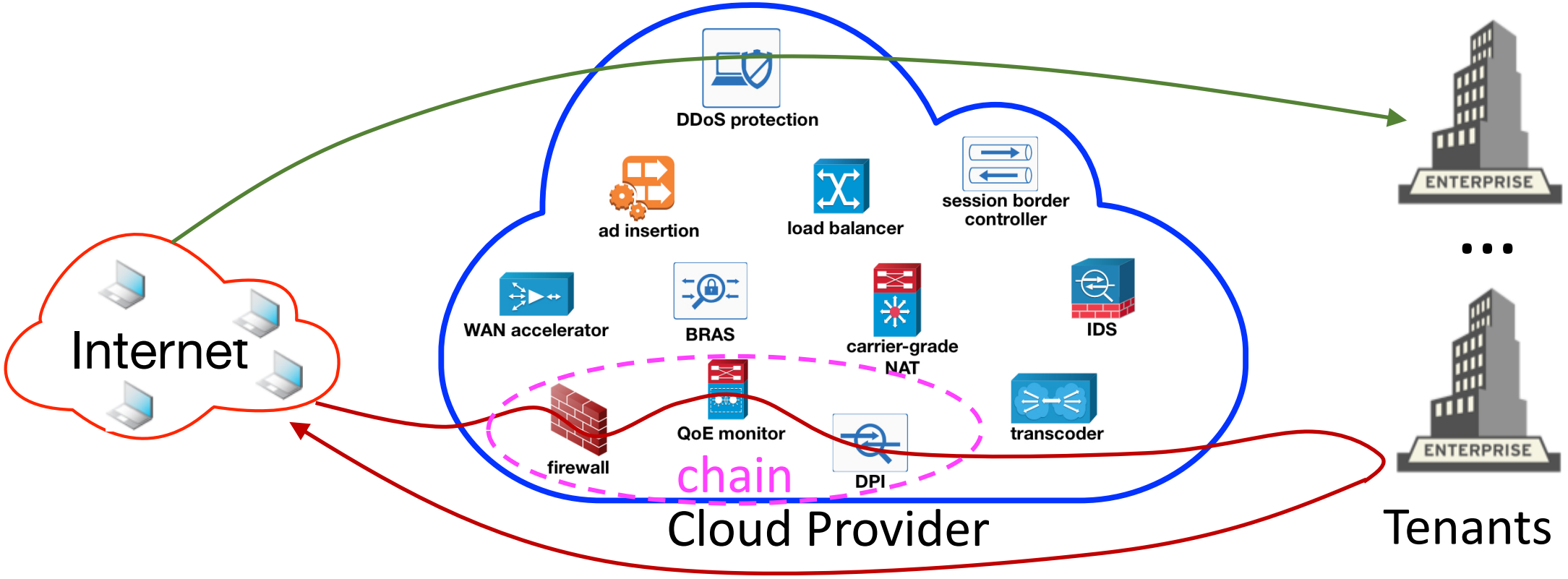
Outsourcing VNFs to the Cloud



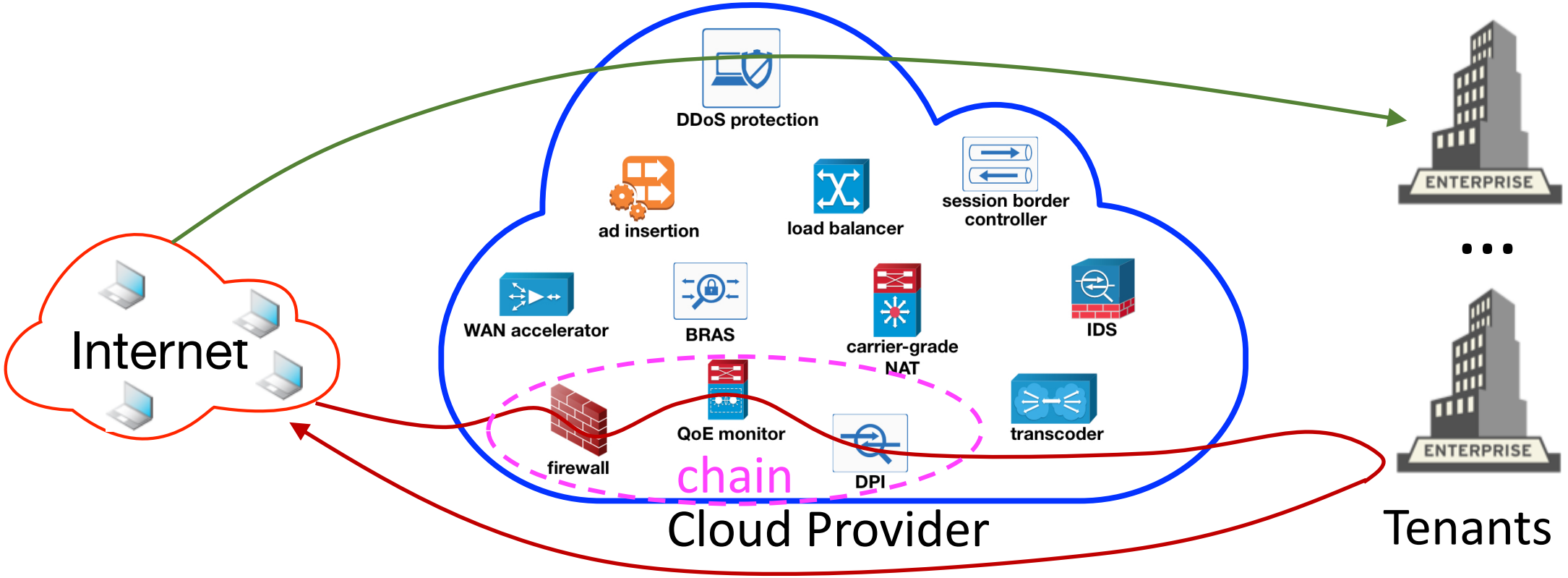
Outsourcing VNFs to the Cloud



Outsourcing VNF Chains to the Cloud



Outsourcing VNF Chains to the Cloud



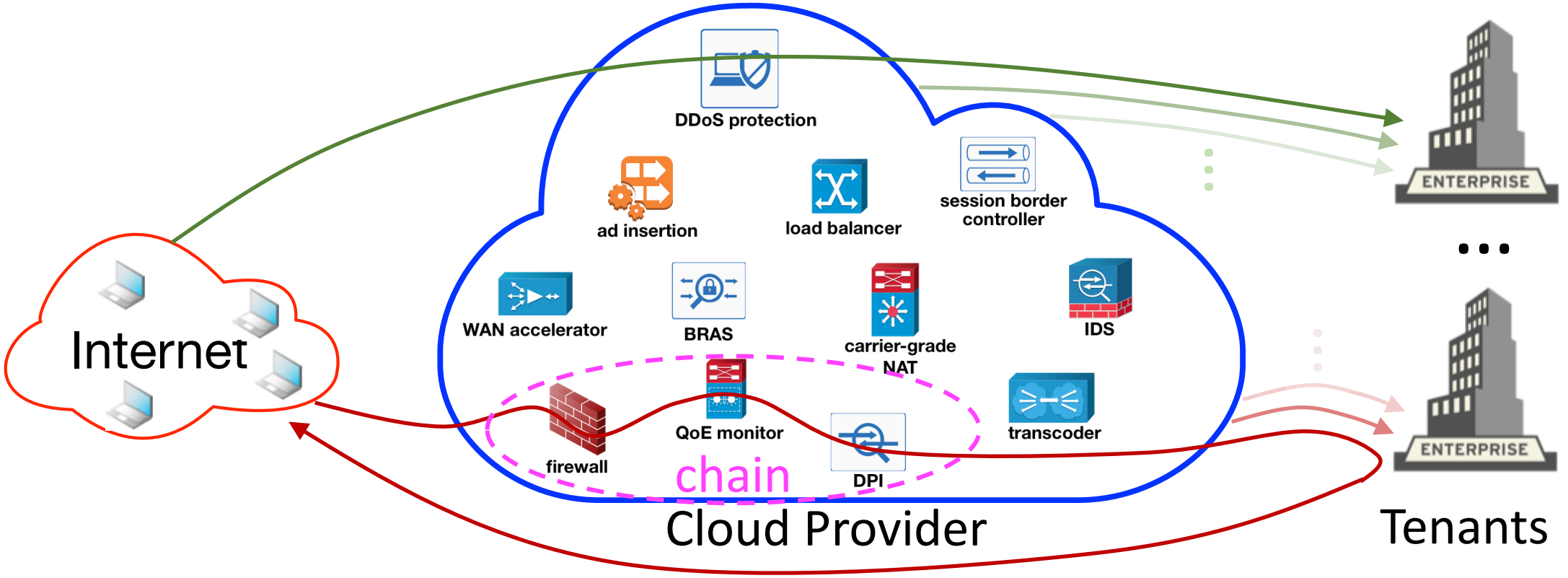
zscaler® The Cloud Security Company

CLOUDFLARE

Challenges of outsourcing VNF Chains

How can cloud providers achieve high data center utilization?

How can tenants allocate and manage their VNF chains?

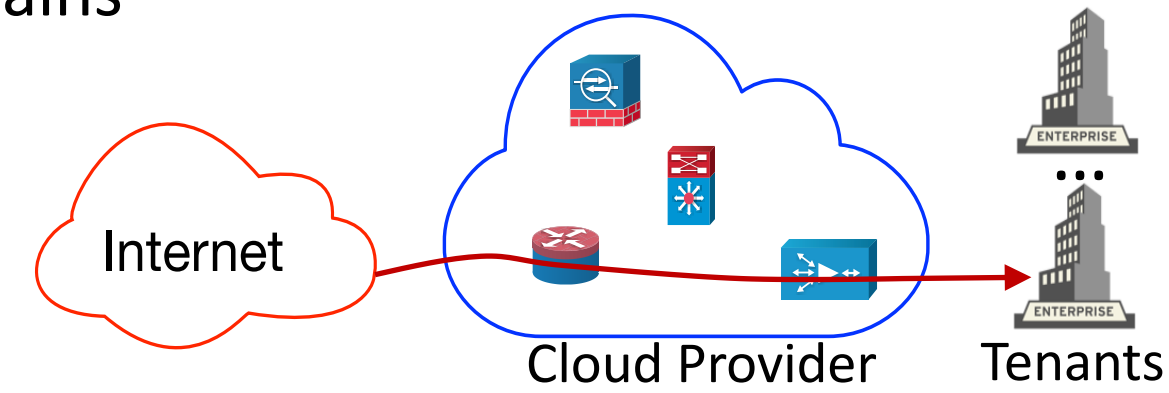


Our contributions: API and algorithm

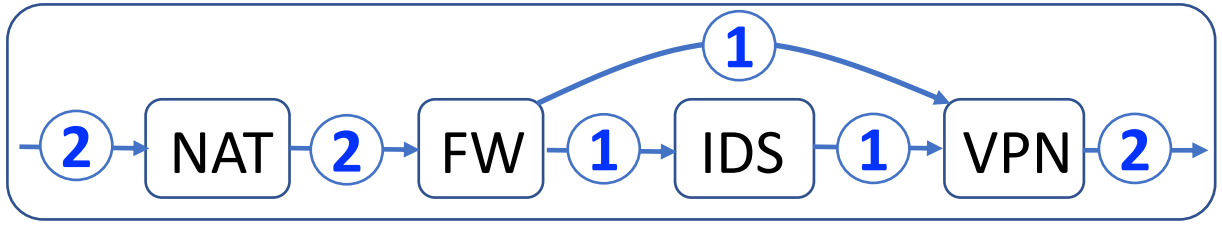
How can cloud providers achieve high data center utilization?

How can tenants allocate and manage their VNF chains?

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



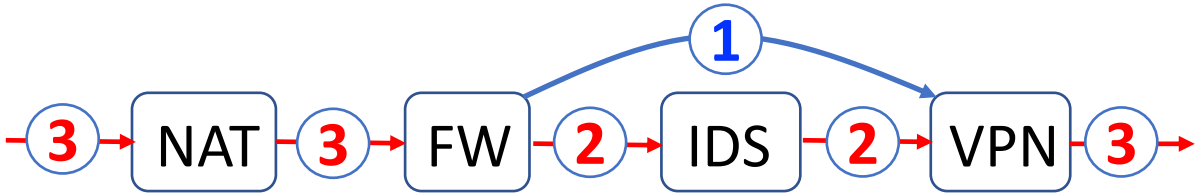
VNF Chain: six API with use-cases



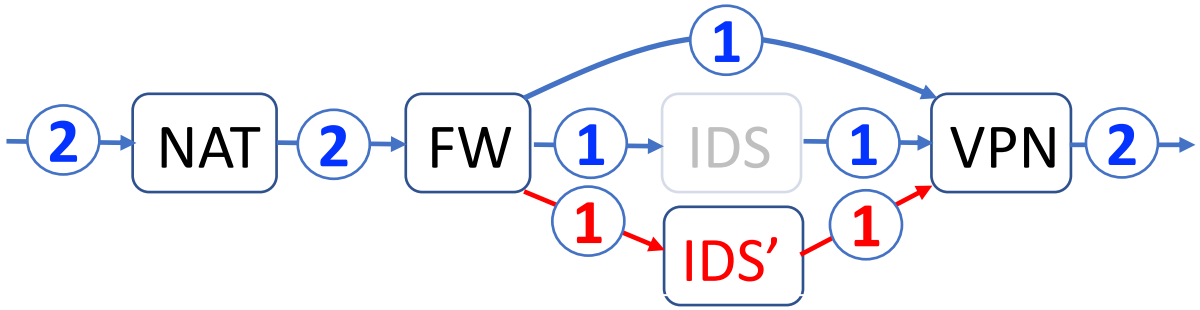
Initial chain

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

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

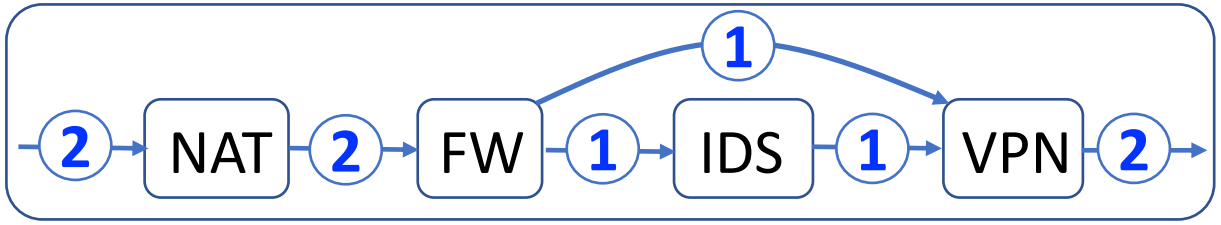


Chain scale-out



Element upgrade

VNF Chain: API is expressive



Initial chain

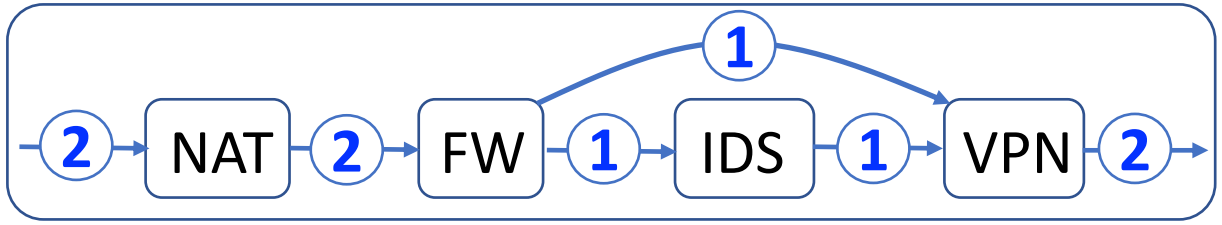
cid ← allocate-chain(C, bw)
add-link-bandwidth(a, b, bw, cid)
add-node(f, cid)

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

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

Chain scale-out Element upgrade Chain expand …

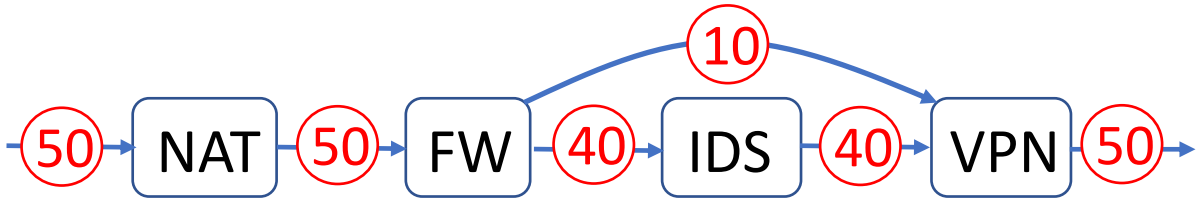
Scale-out beyond single physical resource capacity



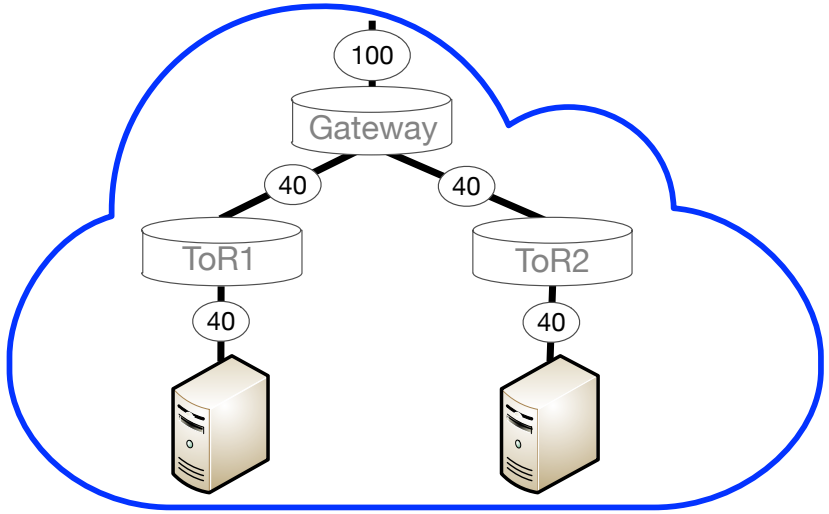
Initial chain

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)

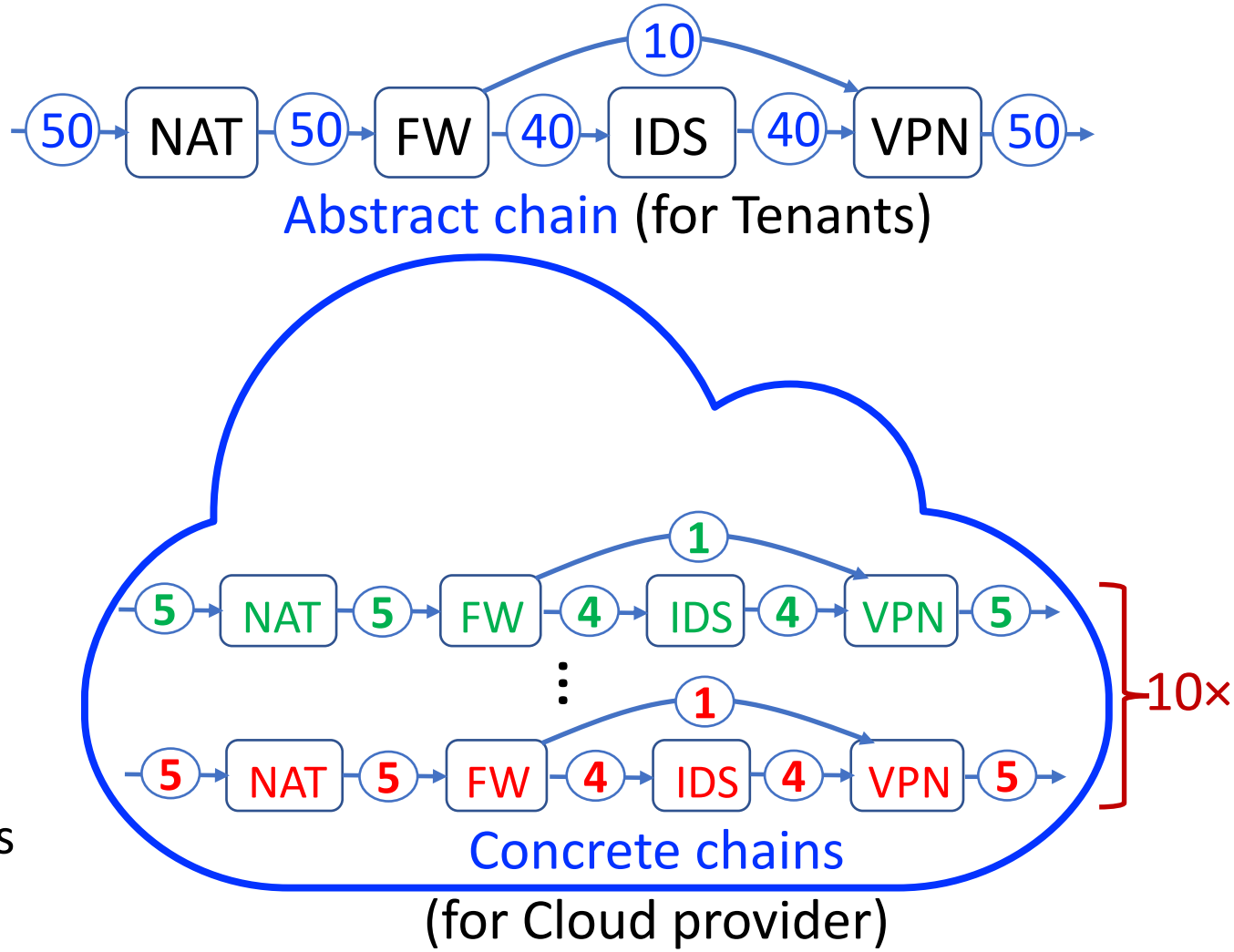


Chain scale-out



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
- Challenges
  - low-latency, packet loss, state synchronization, efficiency loss (see the paper and ANCS'18 poster)

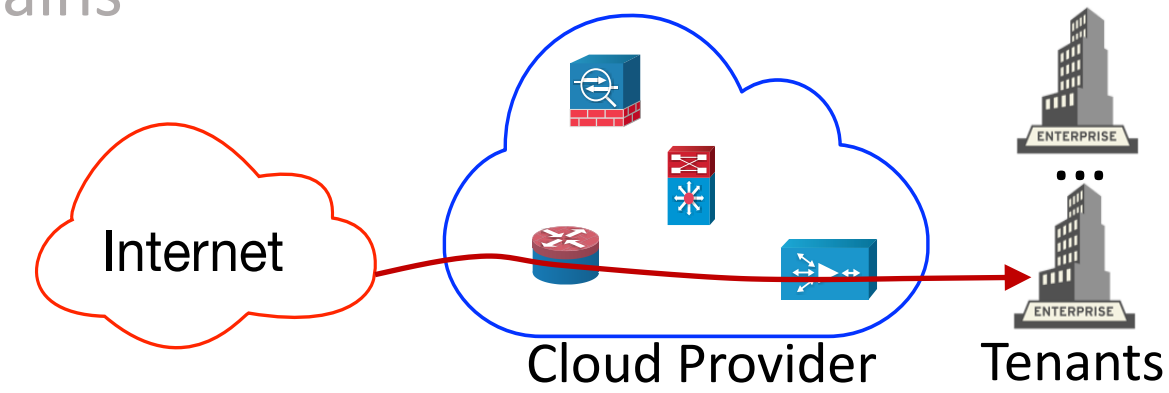


Our contributions: API and algorithm

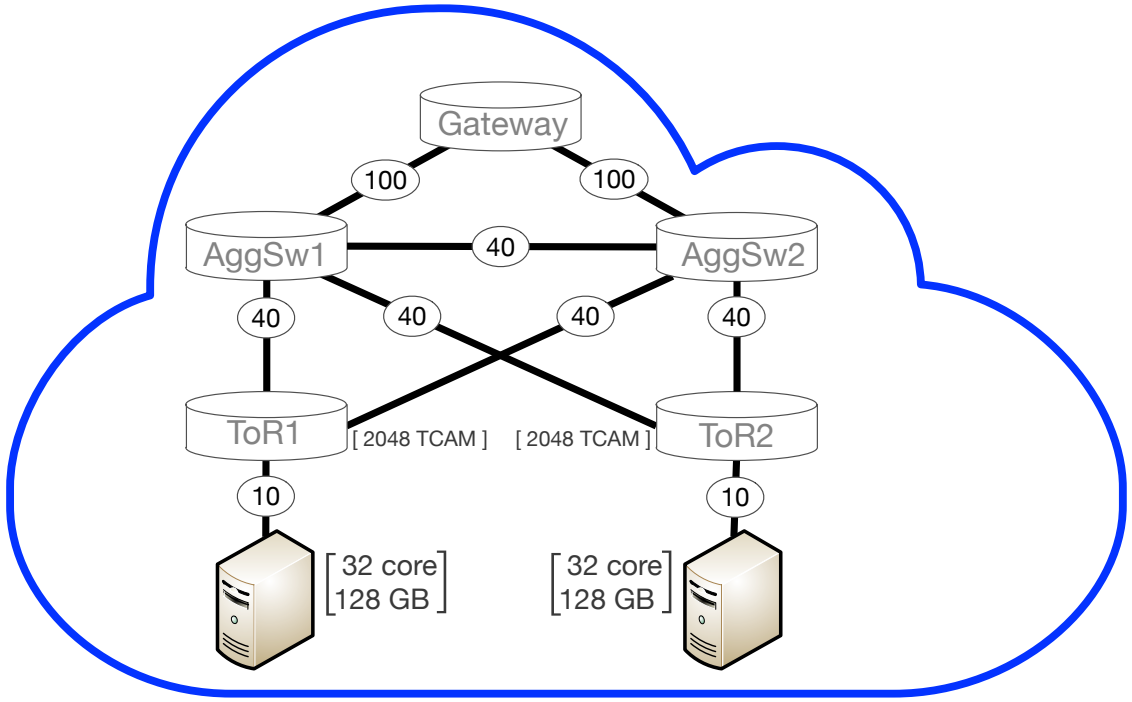
How can cloud providers achieve high data center utilization?

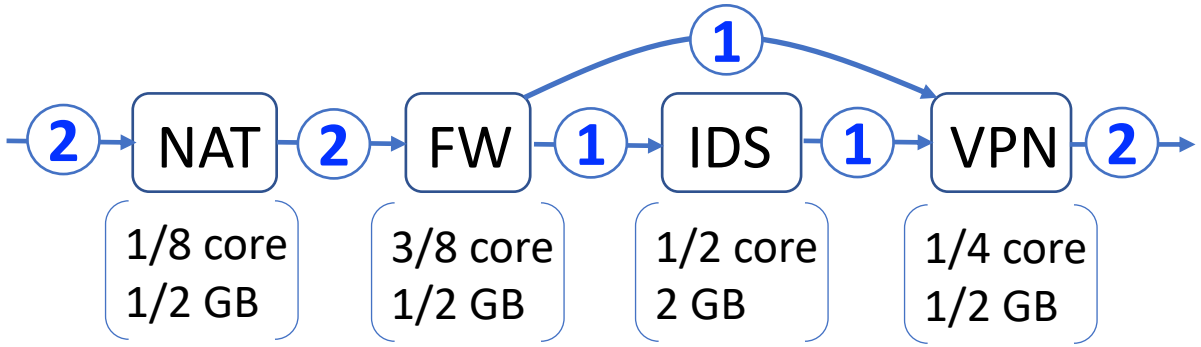
How can tenants allocate and manage their VNF chains?

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



Algorithm inputs: DC topology and chain

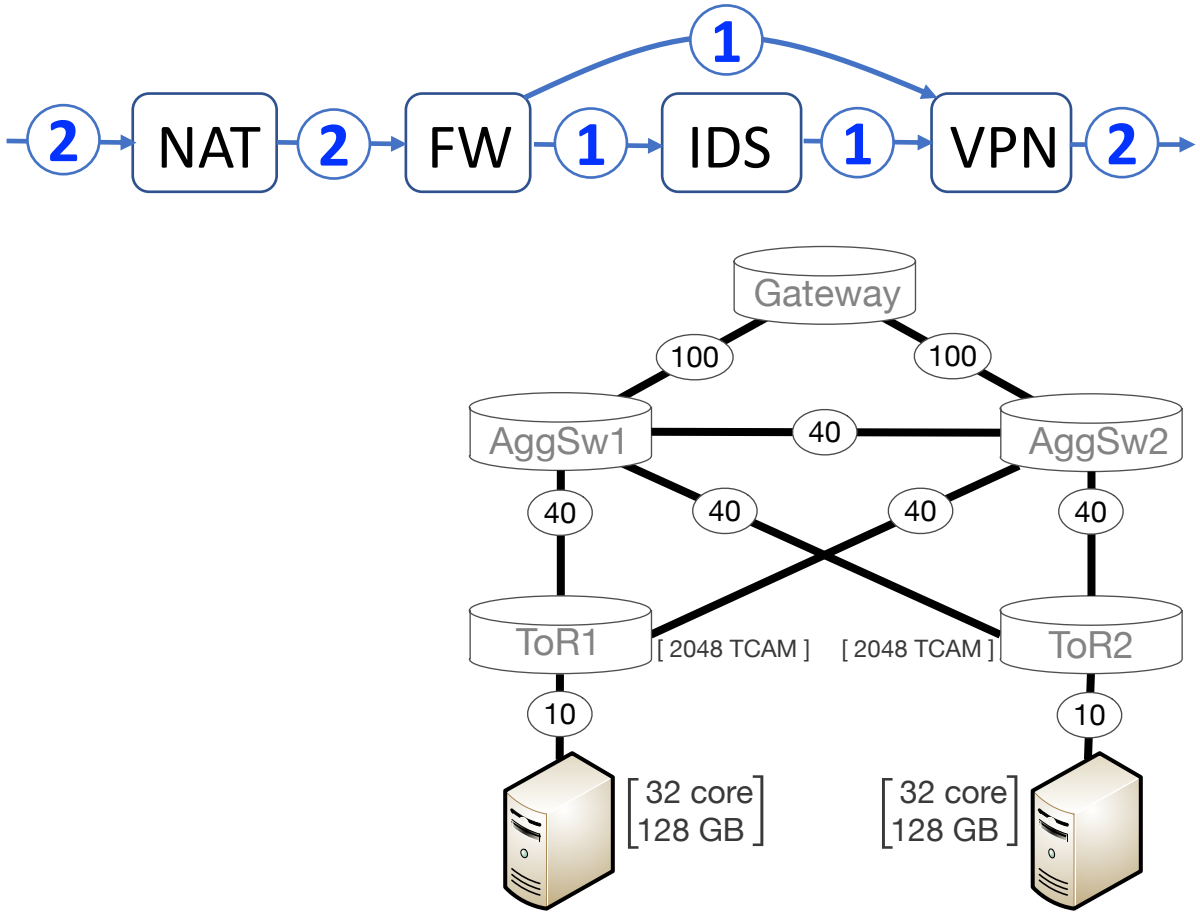




Expected resource consumption per Gbps of traffic
(see the paper for VNF profile generation)

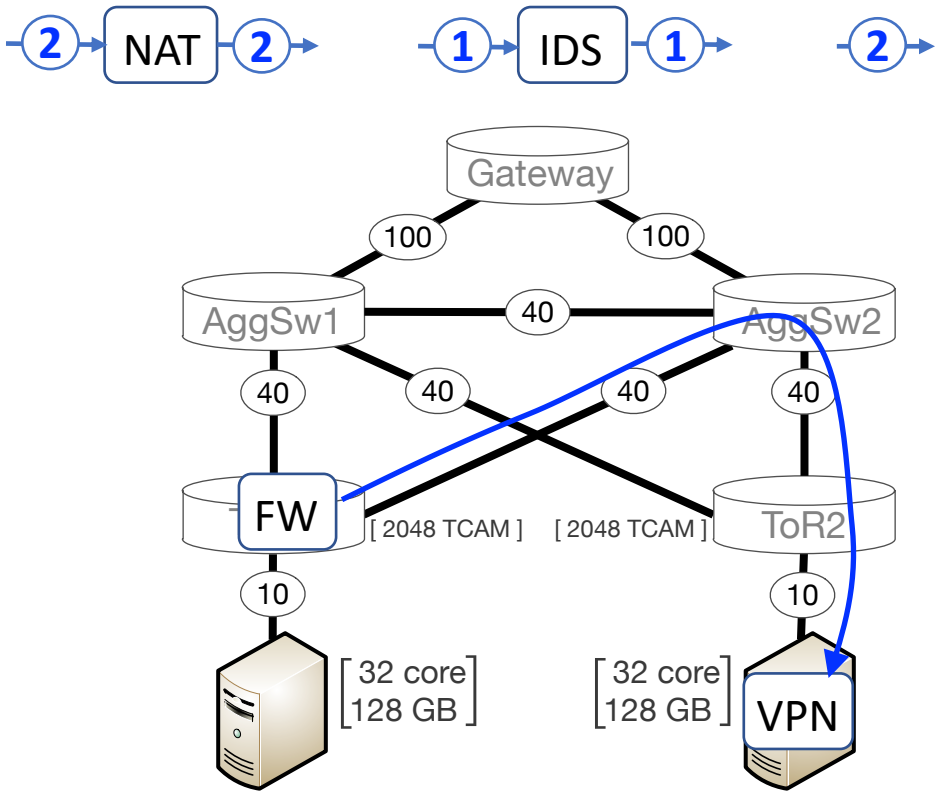
Palkar et al., E2: A Framework for NFV Applications, SOSP'15
Naik et al., NFVPerf: Online performance monitoring and bottleneck detection for NFV, IEEE NFV-SDN 2016.
Nam et al., Probius: Automated Approach for VNF and Service Chain Analysis in Software-Defined NFV, SOSR'18

Algorithms for Chain Allocation and Management



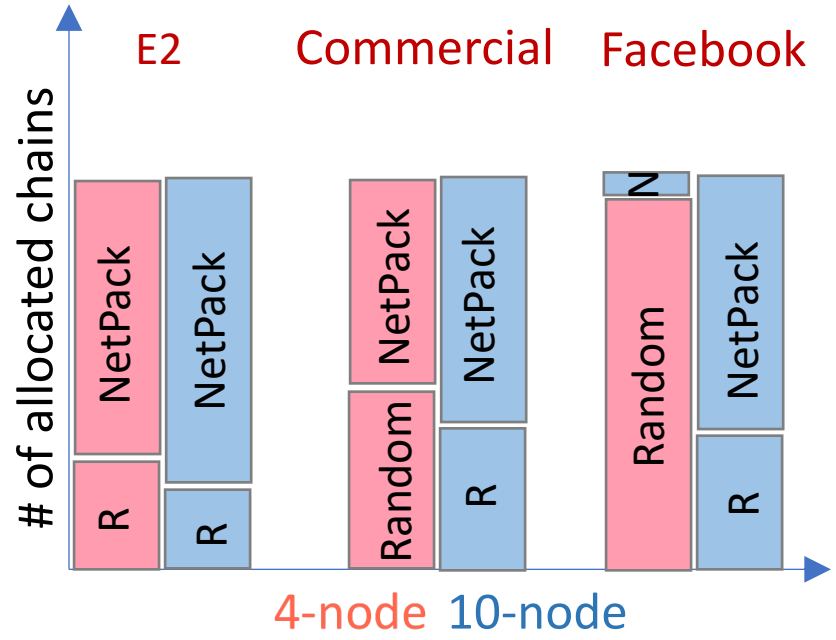
Algorithms for Chain Allocation and Management

- 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

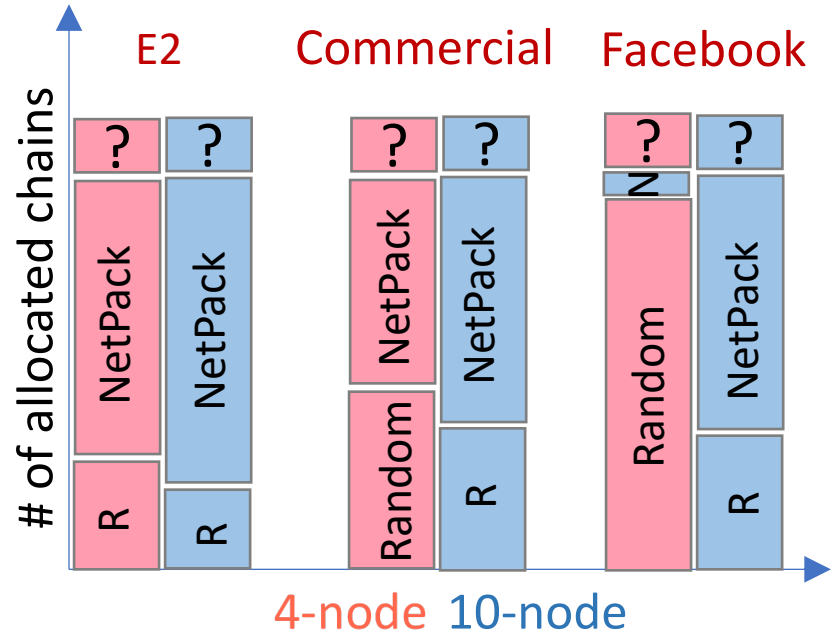
- Random (baseline)
  - Consider NFs and servers/switches in random order
  - Attempt the above step *n* times (e.g., n=100)
  - Choose the shortest path between chain NFs
- NetPack: Random + 3 simple heuristics
  - Consider the chain NFs in a topological order
  - Re-use the same server when allocating consecutive NFs
  - Gradually increase the network scope: rack, cluster, etc.



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

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



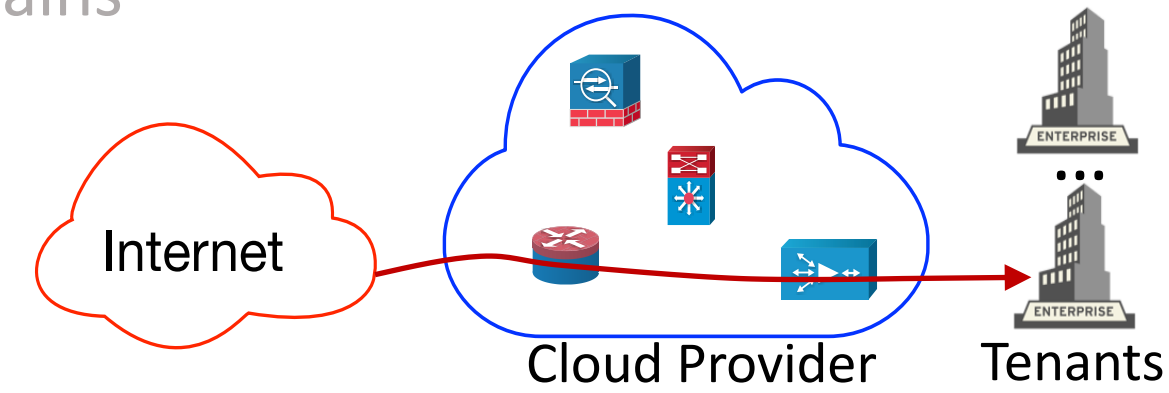
Palkar et al., E2: A Framework for NFV Applications, SOSP'15
Bayless et al., SAT Modulo Monotonic Theories, AAAI'15

Our contributions: API and algorithm

How can cloud providers achieve high data center utilization?

How can tenants allocate and manage their VNF chains?

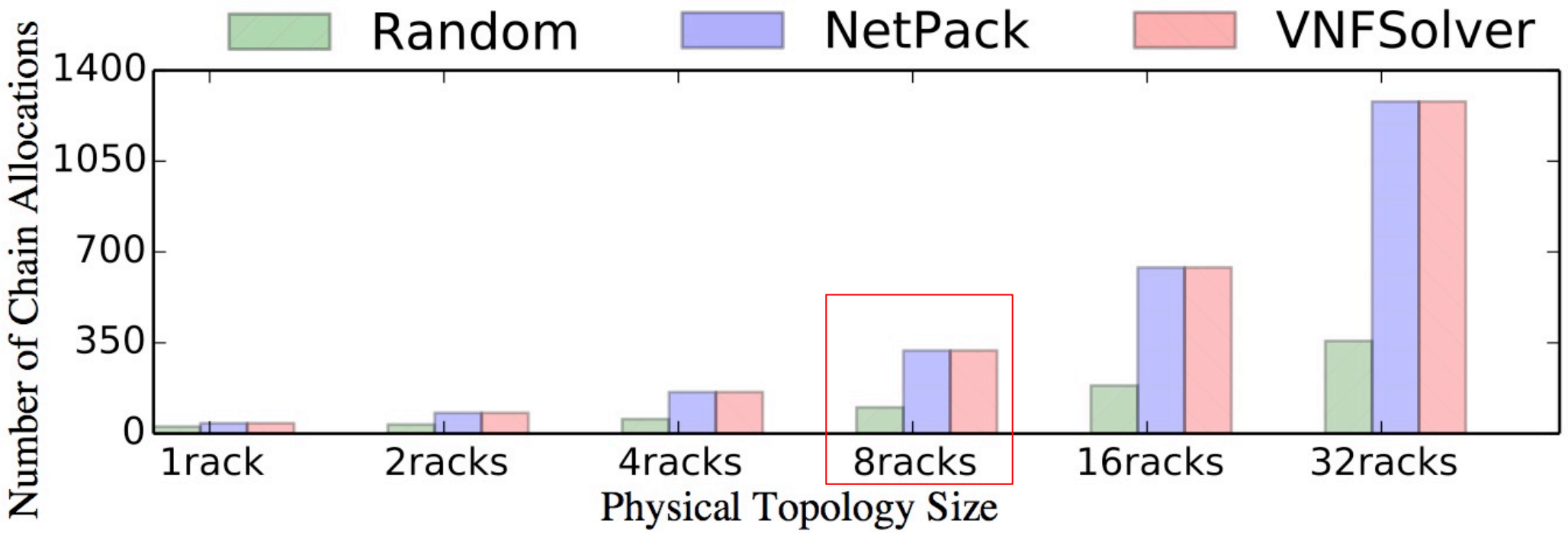
- API to allocate and manage VNF chains
- Three algorithms
  - implement the API, and
  - achieve high data center utilization
- Evaluation
  - simulate: in data center scale with 1000+ servers
  - Daisy: emulate chain management at rack-scale

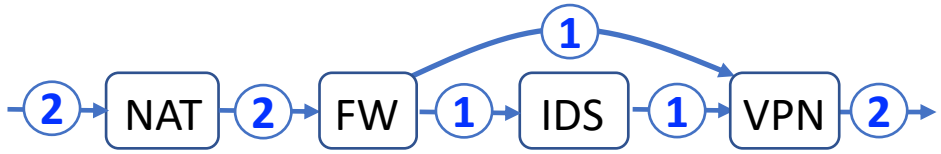


Evaluation: Objectives

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

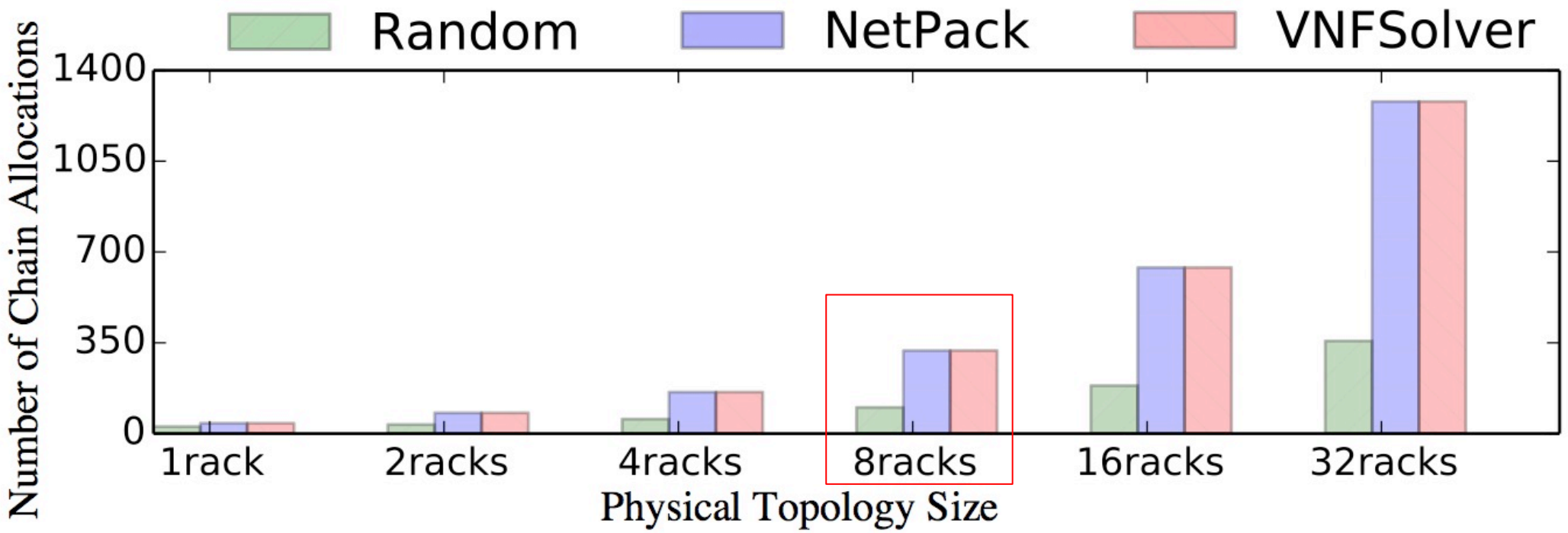
Data center utilization evaluation





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

Data center 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

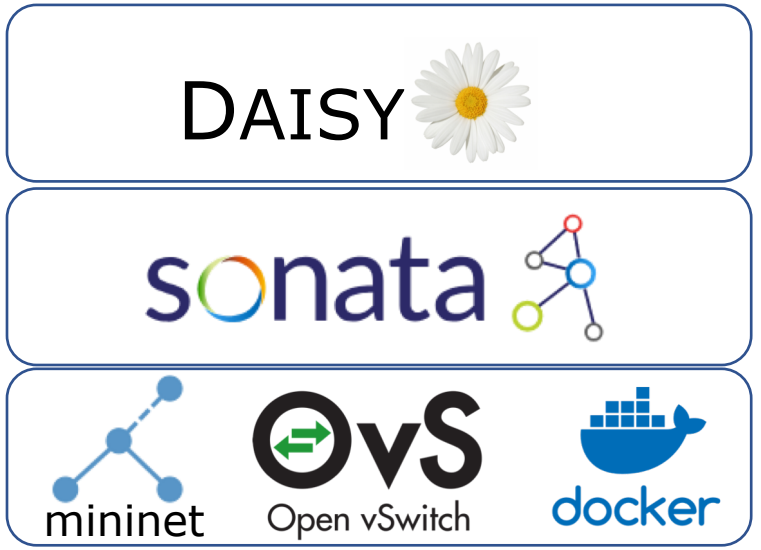
NetPack Utilization and Speed

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)

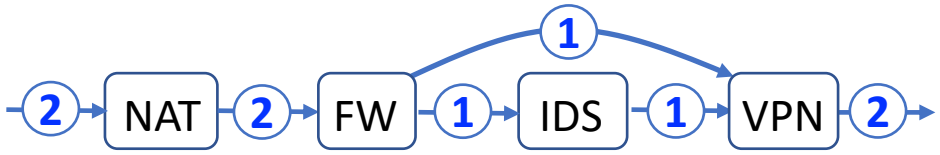
Feasibility check: can API be implemented?

- 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

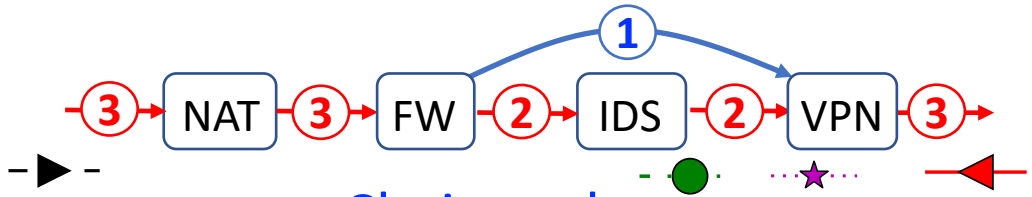


Peuster et al., Sonata NFV SDK, github.com/sonata-nfv/son-emu, 2017

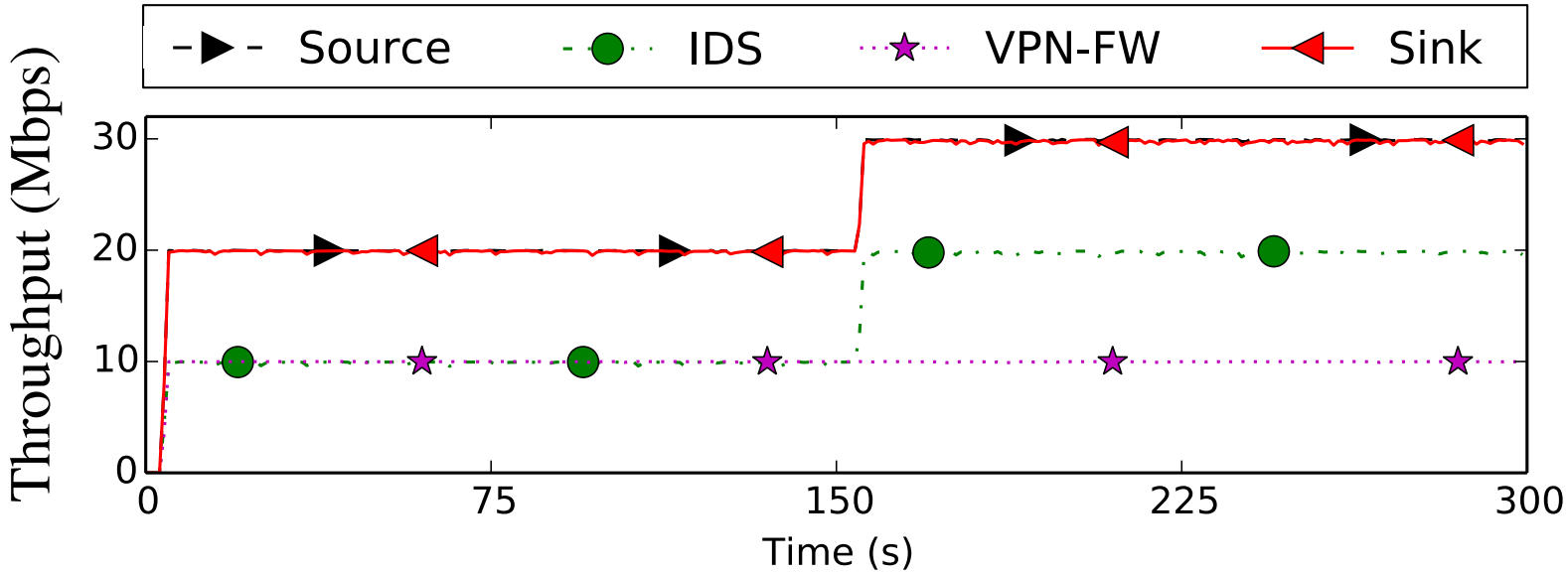
VNF Chain use-cases are feasible with narrow API



Initial chain



Chain scale-out

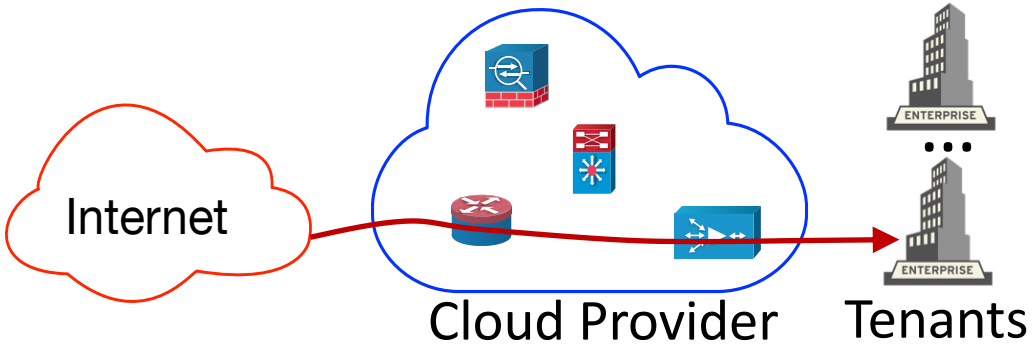


Daisy implements scale-out with no packet drops.

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.

Contributions



- 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

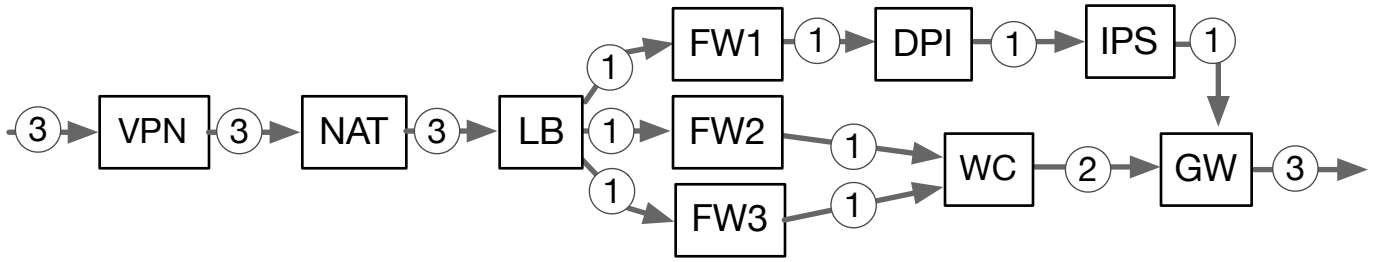
How can tenants allocate and manage their VNF chains?

How can cloud providers achieve high data center utilization?

Thank you!

Backup Slides

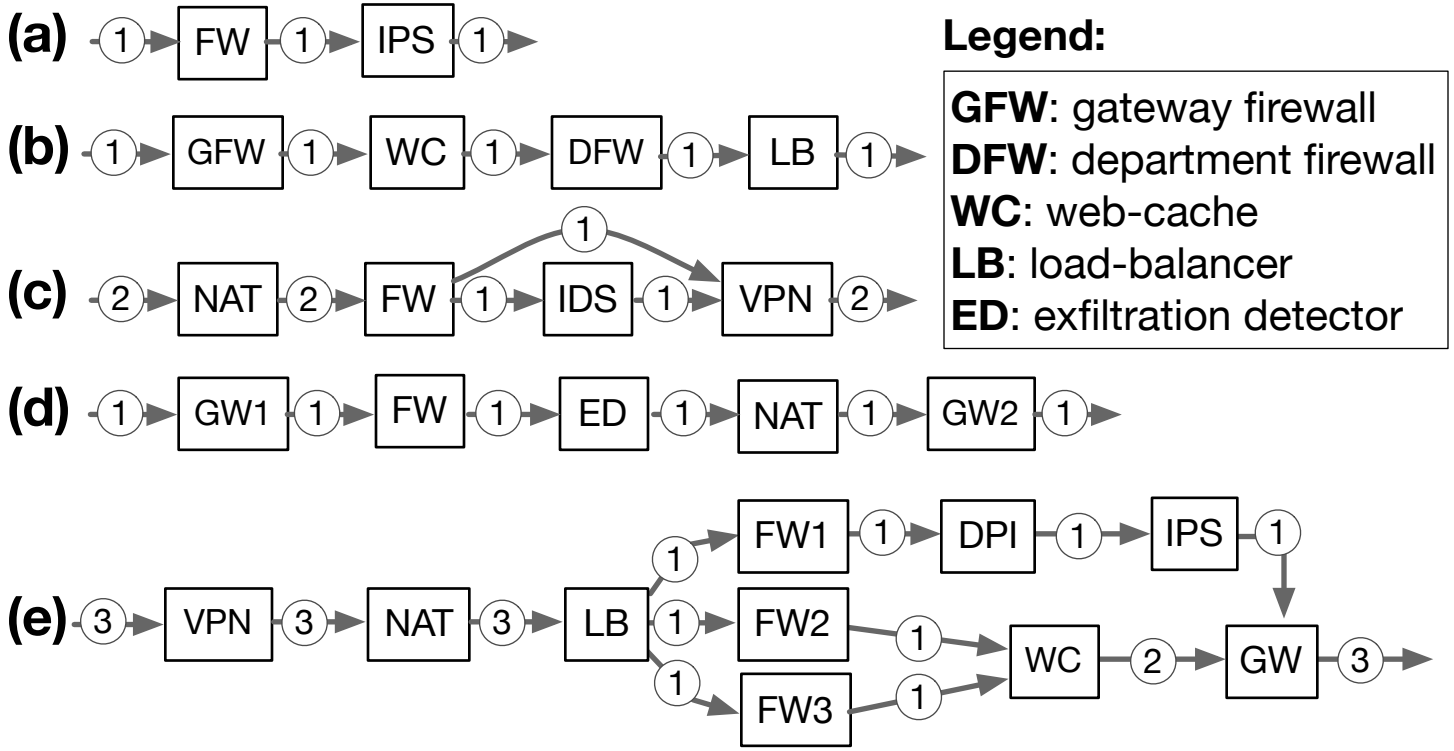
Topological sort of VNF chain



An example using Kahn's algorithm
(VPN, NAT, LB, FW3, FW1, FW2, WC, DPI, IPS, GW)

A. B. Kahn, Topological sorting of large networks, Communications of the ACM, 1962

VNF Chains we consider



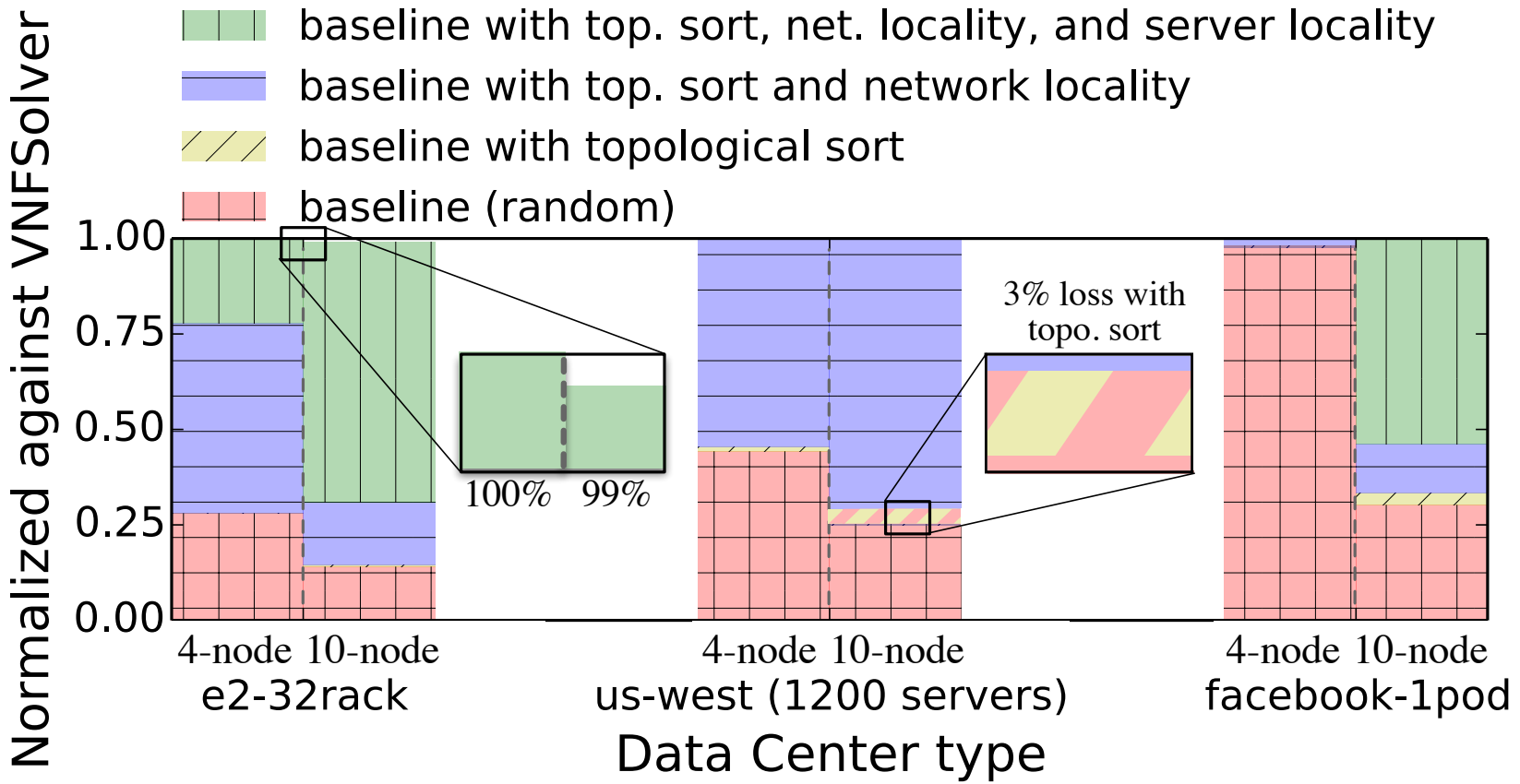
Chains (a) and (b) are from OpenBox, (c) and (e) are from E2, and (d) is from Embark.

Bremler-Barr et al., OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions, SIGCOMM'16

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

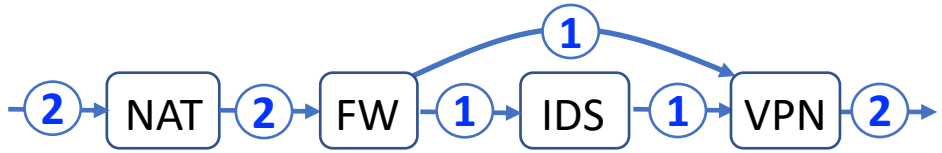
Chang et al., Embark: Securely Outsourcing Middleboxes to the Cloud, NSDI'16

NetPack: Contribution of each Optimization



VNF Chain use-cases are feasible with narrow API

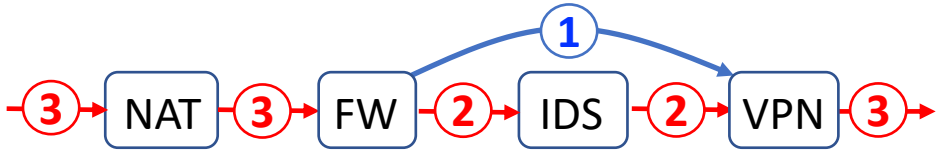
add-link-bandwidth()



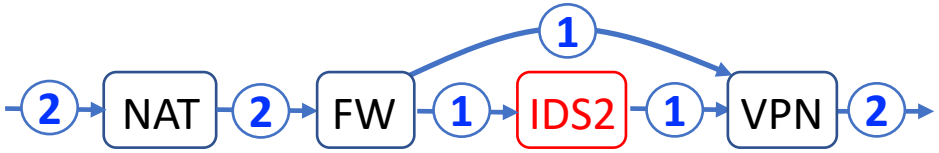
add-node()

remove-link-bandwidth()

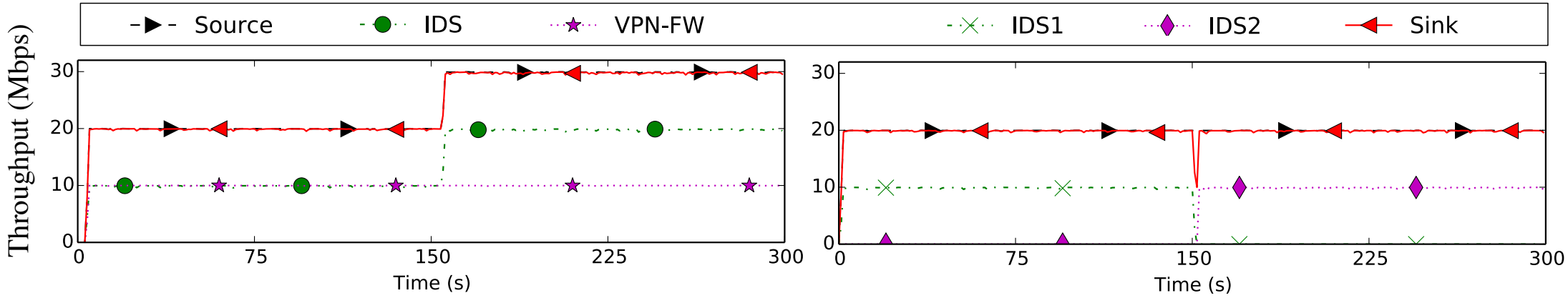
remove-node()



Chain scale-out

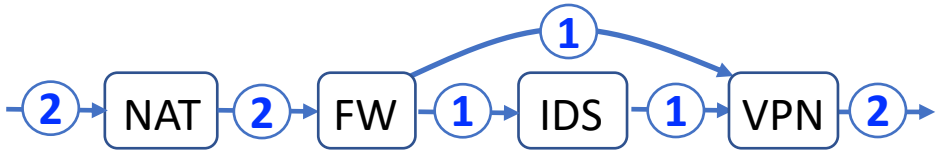


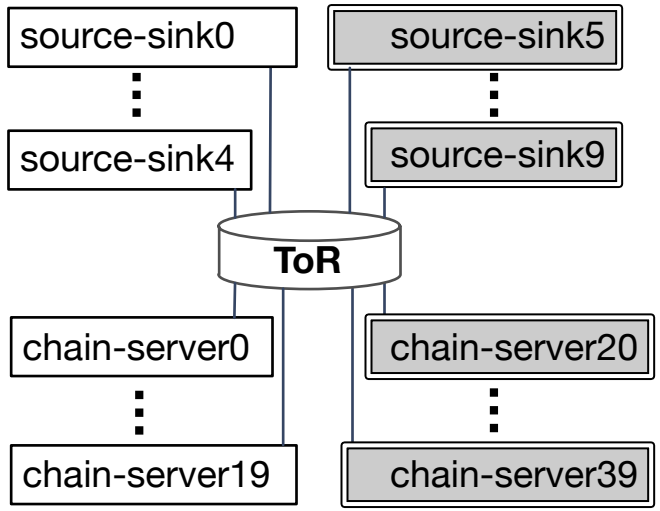
Element upgrade

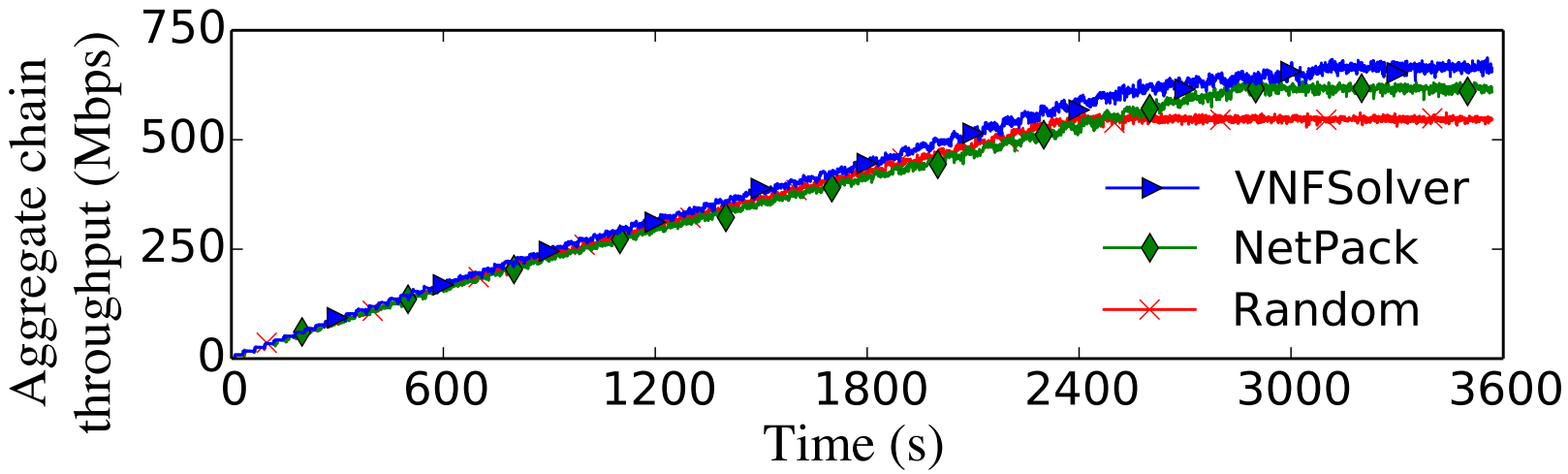


Daisy implements scale-out with no packet drops and element upgrade with 1s packet drop at most (not shown).

Daisy: continuous chain arrival

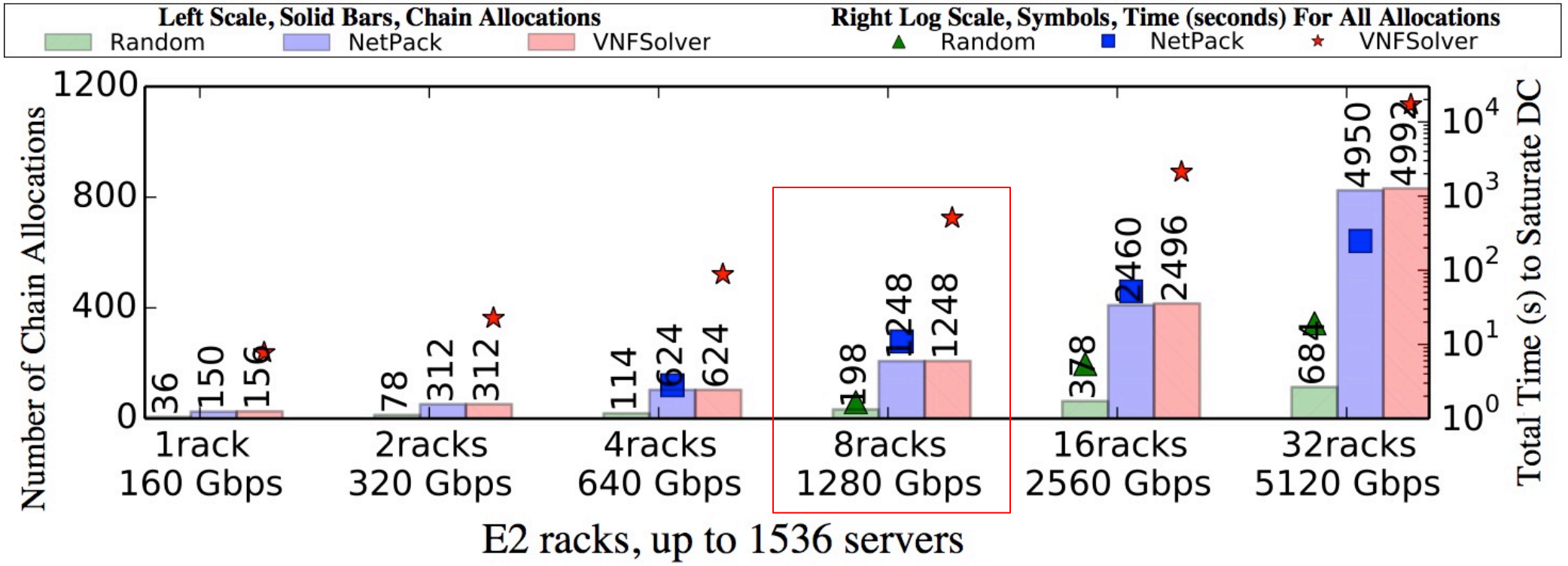






VNFSolver allocated 75 concrete chains (687 Mbps)
NetPack allocated 67 concrete chains (633 Mbps)
Random allocated 61 concrete chains (561 Mbps)
(throughput with iperf generated packets is precise)

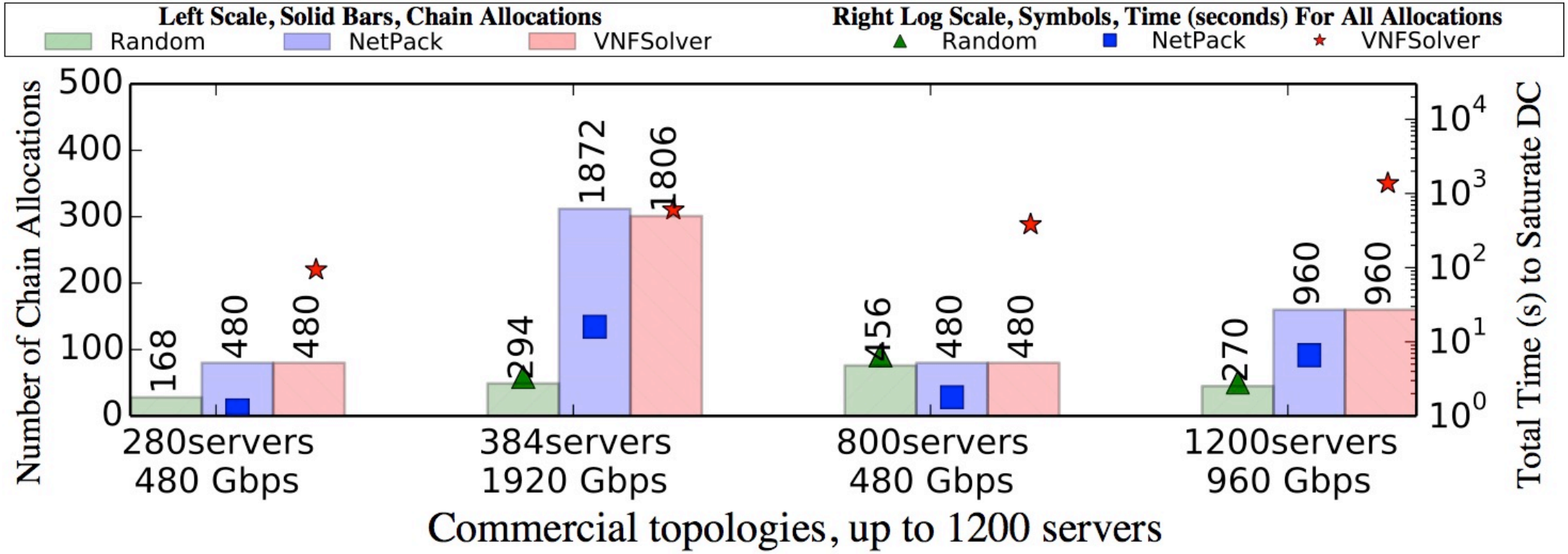
Utilization and Speed on E2 racks



NetPack achieves at least 96% of VNFSolver allocations while being 82x faster than VNFSolver on average.

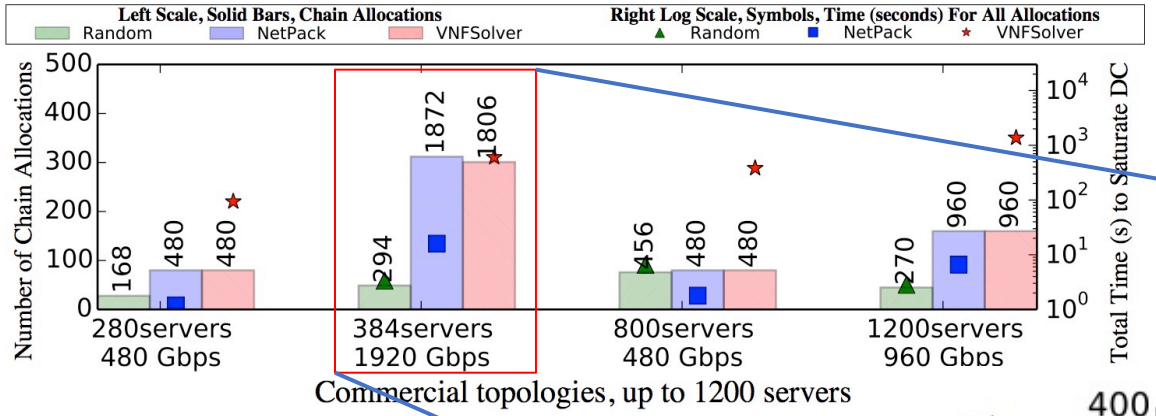
Palkar et al., E2: A Framework for NFV Applications, SOSP'15

Utilization and Speed on Commercial Topologies

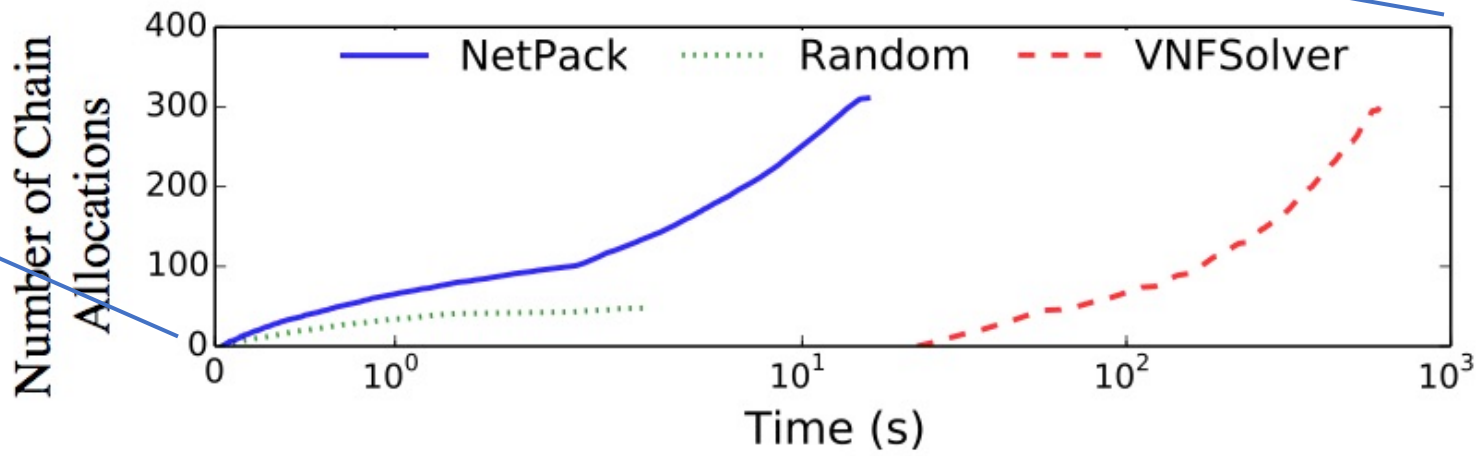


Gives qualitatively similar results, but also reveals a corner case for VNFSolver (-3.65%).

A corner case for VNFSolver



Variance across 10 runs:
- Random < 10.4%
- NetPack < 0.7%
- VNFSolver < 3.7%



Gives qualitatively similar results, but also reveals a corner case for VNFSolver (-3.65%).