SRv6LB: Leveraging IPv6, Segment Routing, and VPP for a Very Fast, Reliable, and Efficient Distributed DC Workload Balancing

Mark Townsley, Pierre Pfister, and Yoann Desmouceaux

Cisco + Ecole Polytechnique (Paris)

Kubecon: May 2, 2018

IPv6

IPv6

Segment Routing

+ VPP

= Workload Balancing

IPv6 in Kubernetes

- IPv4 Parity, no API Changes
- CNI 0.6.0 Bridge & Host-Local IPAM
- ip6tables & ipvs
- Kube-DNS & CoreDNS
- kubeadm

- Dual-Stack, parallel IPv4/IPv6
- Multiple IPs per pod
- Multiple IPs per service

- SRv6
- Istio IPv6
- Multiprefix Routing…

Rel 1.9 (Alpha)

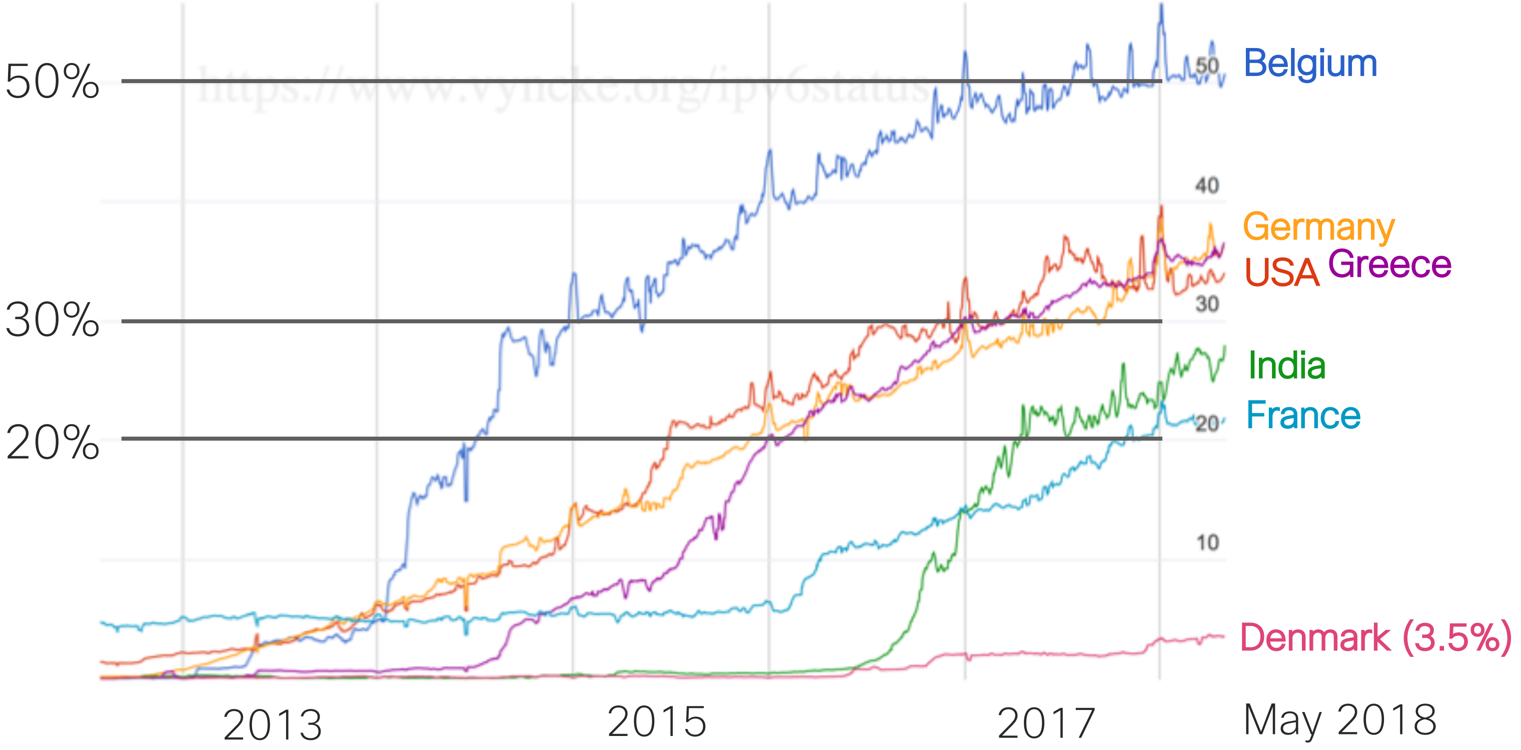
Rel 1.11 (Beta)

Rel 1.12 (targeting)

Planning and Preparing

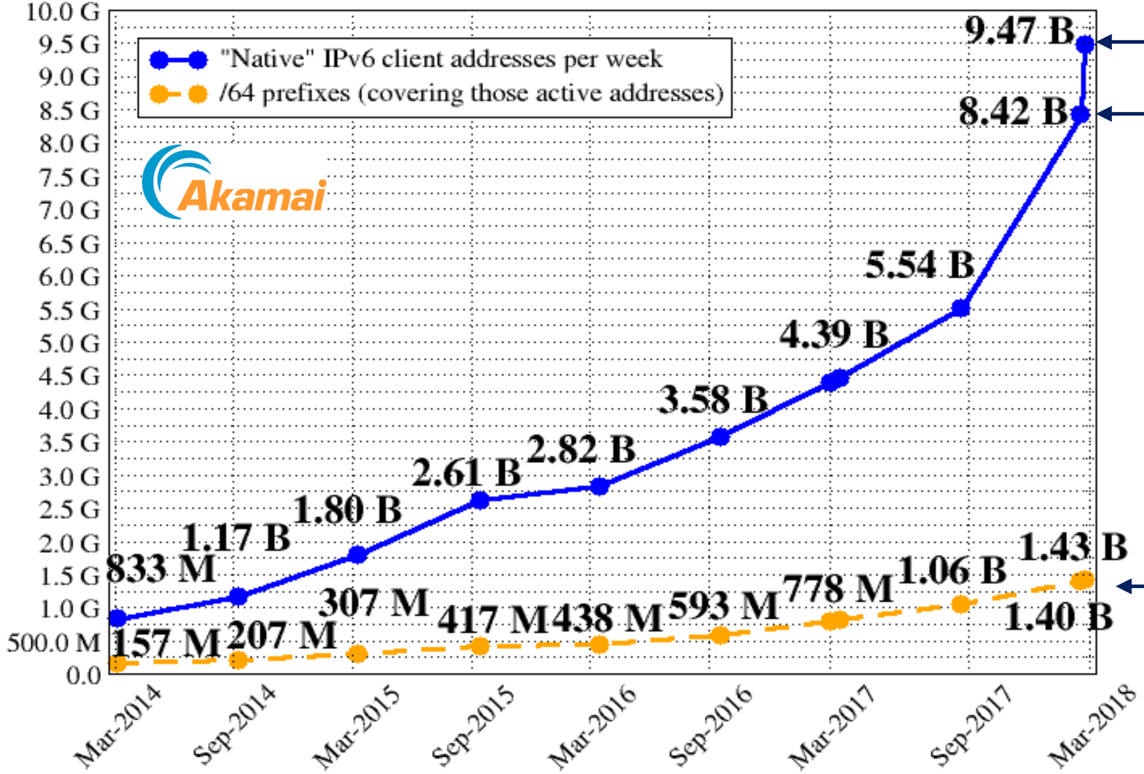
For more info, stop by the **Cisco Booth** and ask for **Dane Leblanc**, **Rob Pothier**, or **Paul Michali**

IPv6 on the Internet



https://www.vyncke.org/ipv6status/compare.php?metric=p&countries=be,us,de,in,gr,fr,dk
https://www.google.com/intl/en/ipv6/statistics.html#tab=per-country-ipv6-adoption&tab=per-country-ipv6-adoption

Unique IPv6 addresses seen by Akamai in one week



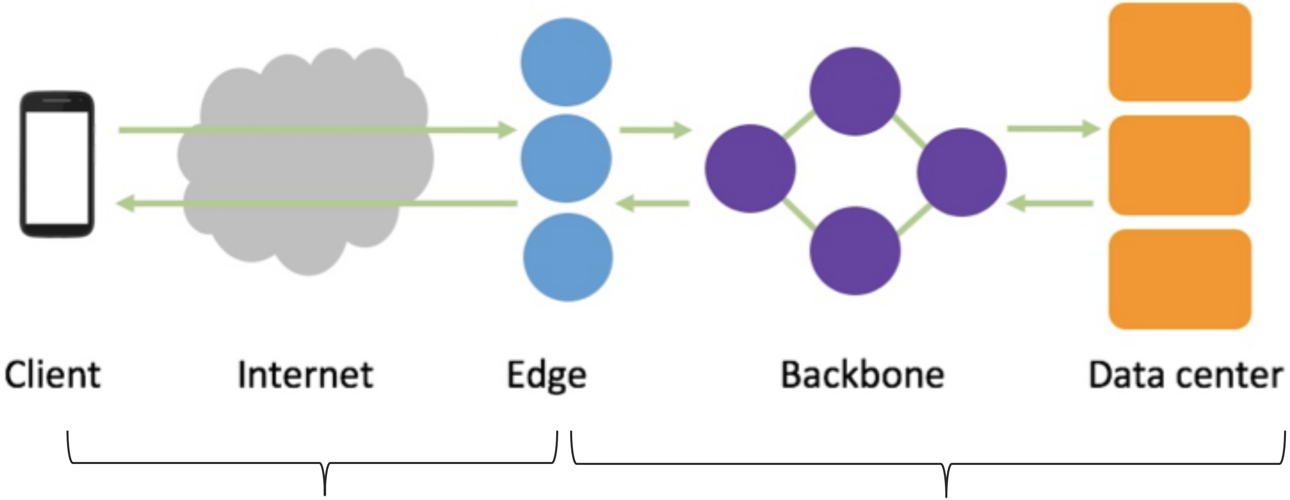
9.47 Billion (week of March 24)

8.42 Billion (week of March 17)

More unique /64s in a week than unique IPv4 addresses in a year

For more info, go see Dave Plonka's keynote at the Network Traffic Measurement and Analysis Conference in Vienna, June 26-29 2018

IPv6 @ Facebook



50% of US Mobile Traffic,
16% Total Traffic, is IPv6

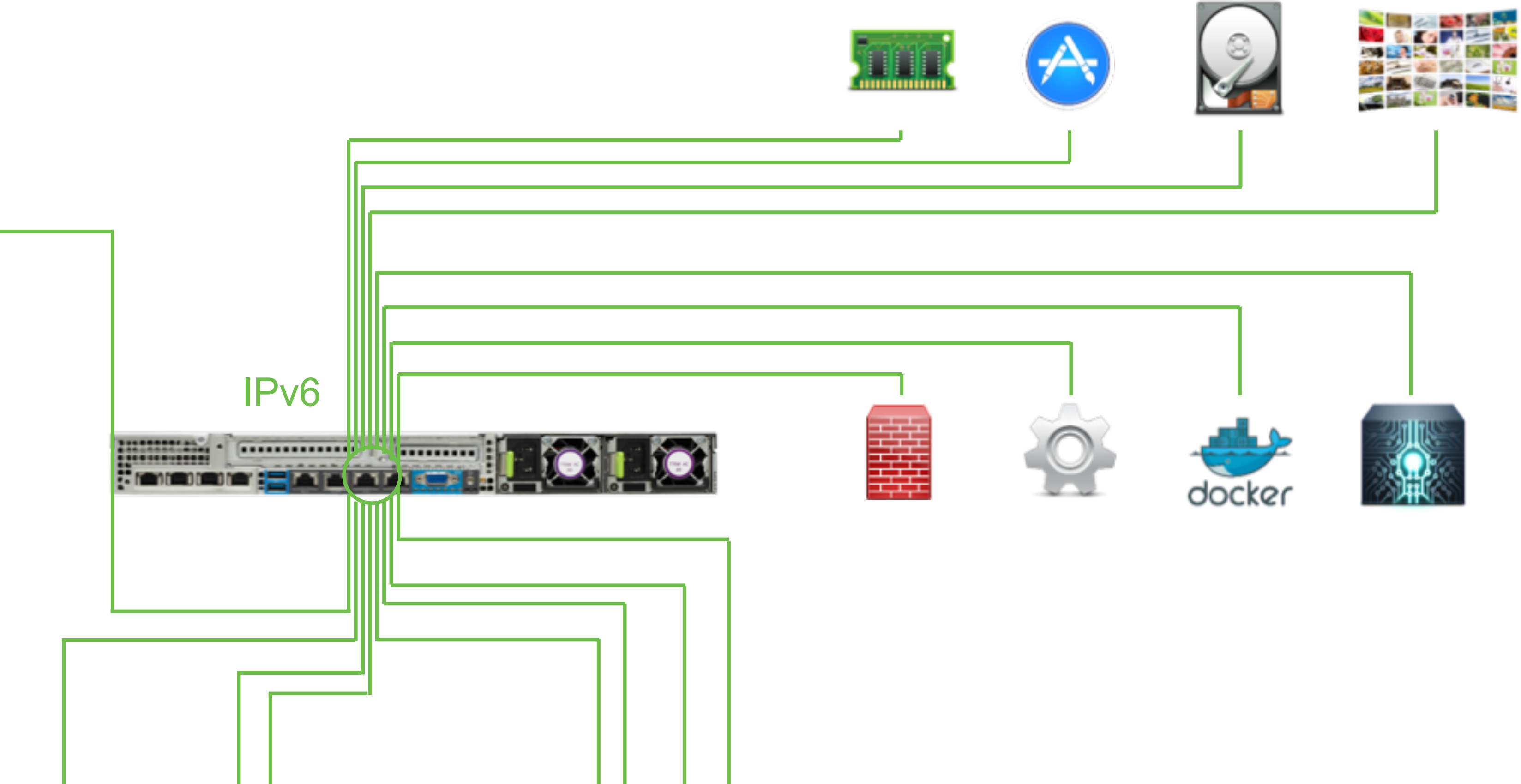
99.99% of traffic IPv6

Source: Dec 2017 UK IPv6 Council Presentation by Mikel Jimenez, Facebook Network Engineer

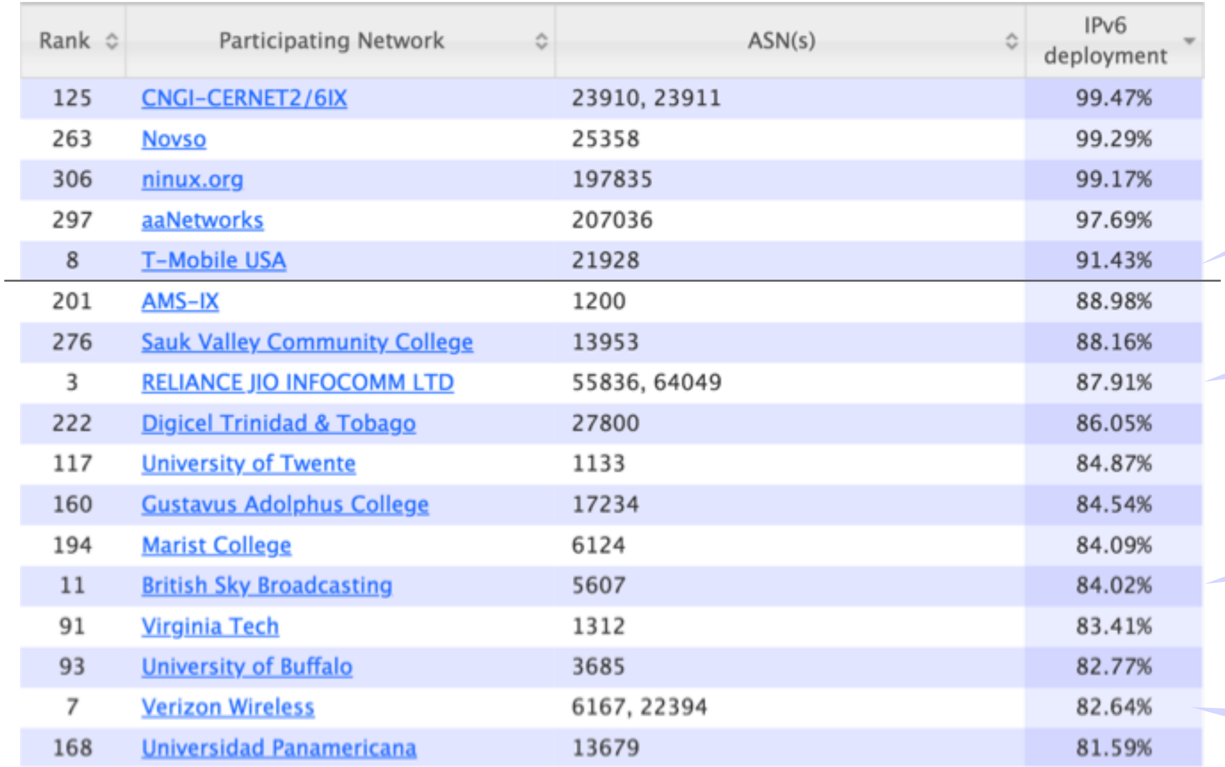
IPv6 Containers @ Facebook (!k8s)

- Every server gets a /64
- Unique IPv6 Address per *task*
  - Each task gets its own IPv6 /128
  - Each task gets the entire port space
  - No more port collisions (!!!)
  - Simpler scheduling and accounting
- /54 per Rack
- /44 per Cluster (/48 in edge)
- /37 DC Fabric
- No NATs!

IPv6 - Routing beyond the network interface



IPv6-Dominant Networks Today



Rank	Participating Network	ASN(s)	IPv6 deployment
125	CNGI-CERNET2/6IX	23910, 23911	99.47%
263	Novso	25358	99.29%
306	ninux.org	197835	99.17%
297	aaNetworks	207036	97.69%
8	T-Mobile USA	21928	91.43%
201	AMS-IX	1200	88.98%
276	Sauk Valley Community College	13953	88.16%
3	RELIANCE JIO INFOCOMM LTD	55836, 64049	87.91%
222	Digicel Trinidad & Tobago	27800	86.05%
117	University of Twente	1133	84.87%
160	Gustavus Adolphus College	17234	84.54%
194	Marist College	6124	84.09%
11	British Sky Broadcasting	5607	84.02%
91	Virginia Tech	1312	83.41%
93	University of Buffalo	3685	82.77%
7	Verizon Wireless	6167, 22394	82.64%
168	Universidad Panamericana	13679	81.59%

T-Mobile USA
70 Million Subscribers

Reliance JIO India
183 Million Subscribers

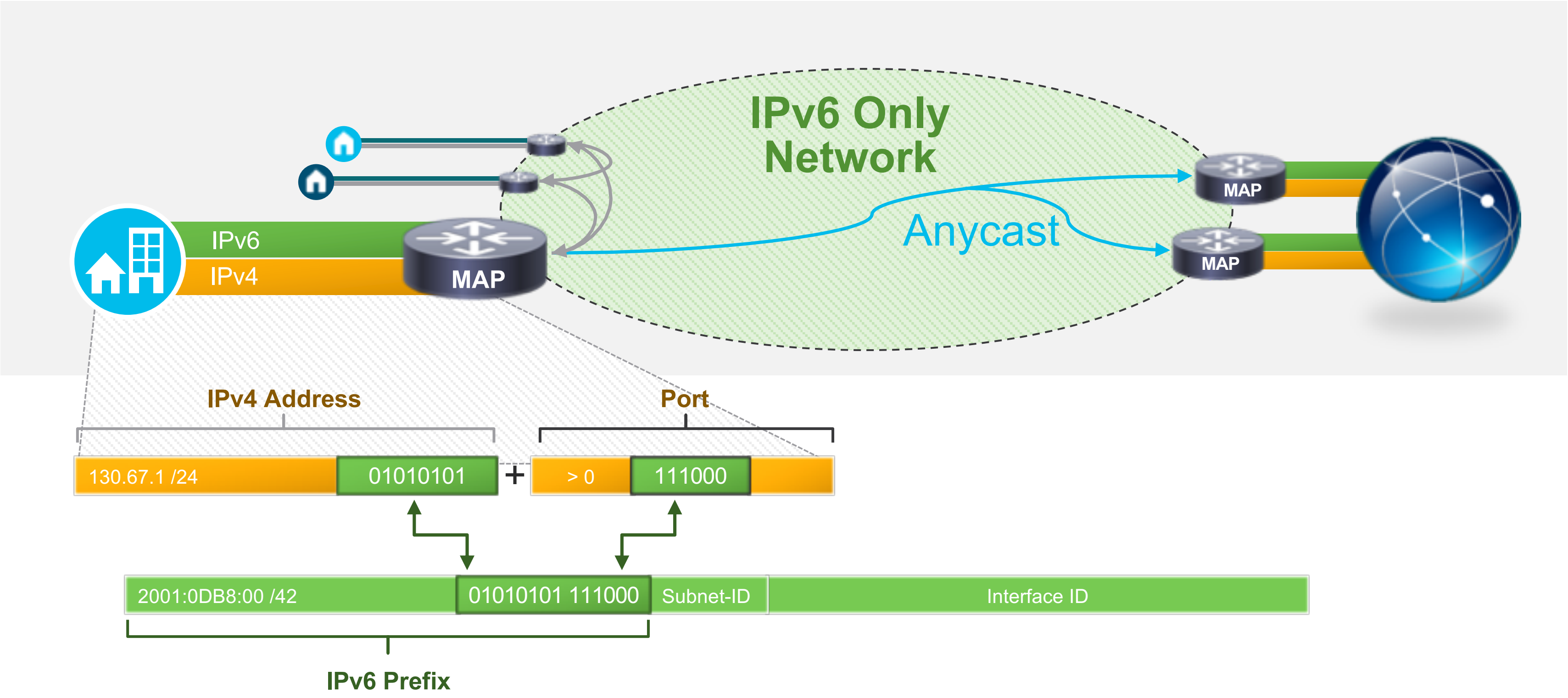
BSkyB (UK, Ireland...)
22.5 Million Subscribers

Verizon Wireless
150 Million Subscribers

http://www.worldipv6launch.org/measurements/

% Composite based on measurements from Google, Yahoo!, Facebook, Akamai, LinkedIn, APNIC

MAP: Routing IPv4 addresses and ports *inside* IPv6



https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/15-mt/nat-15-mt-book/iadnat-map-e.pdf

IPv6

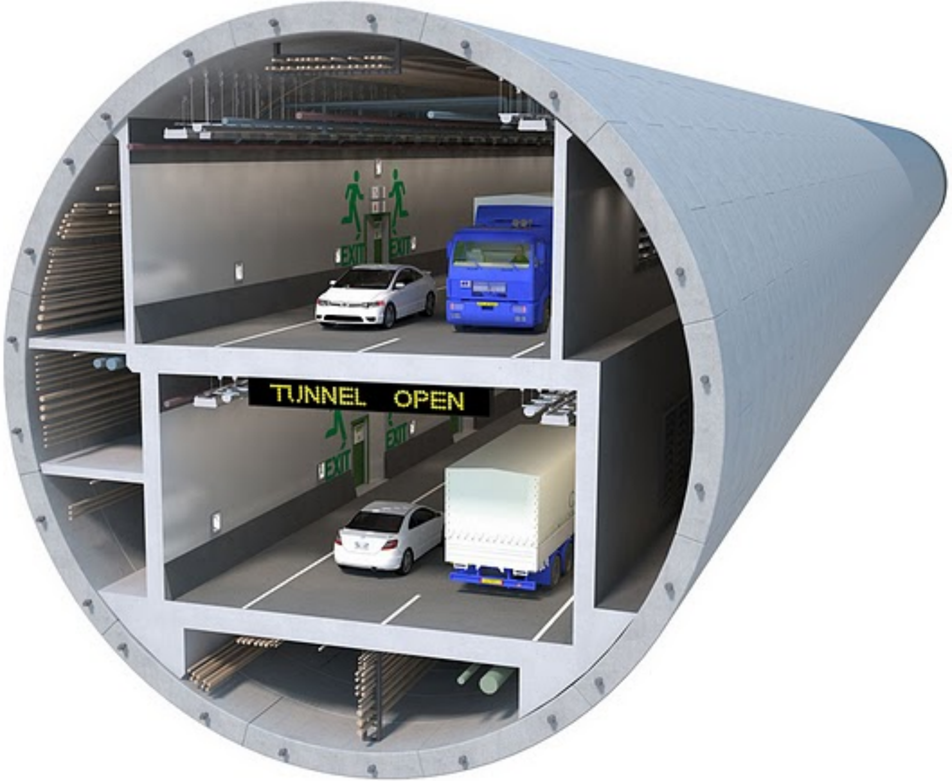
Segment Routing

+ VPP

= Workload Balancing

SR

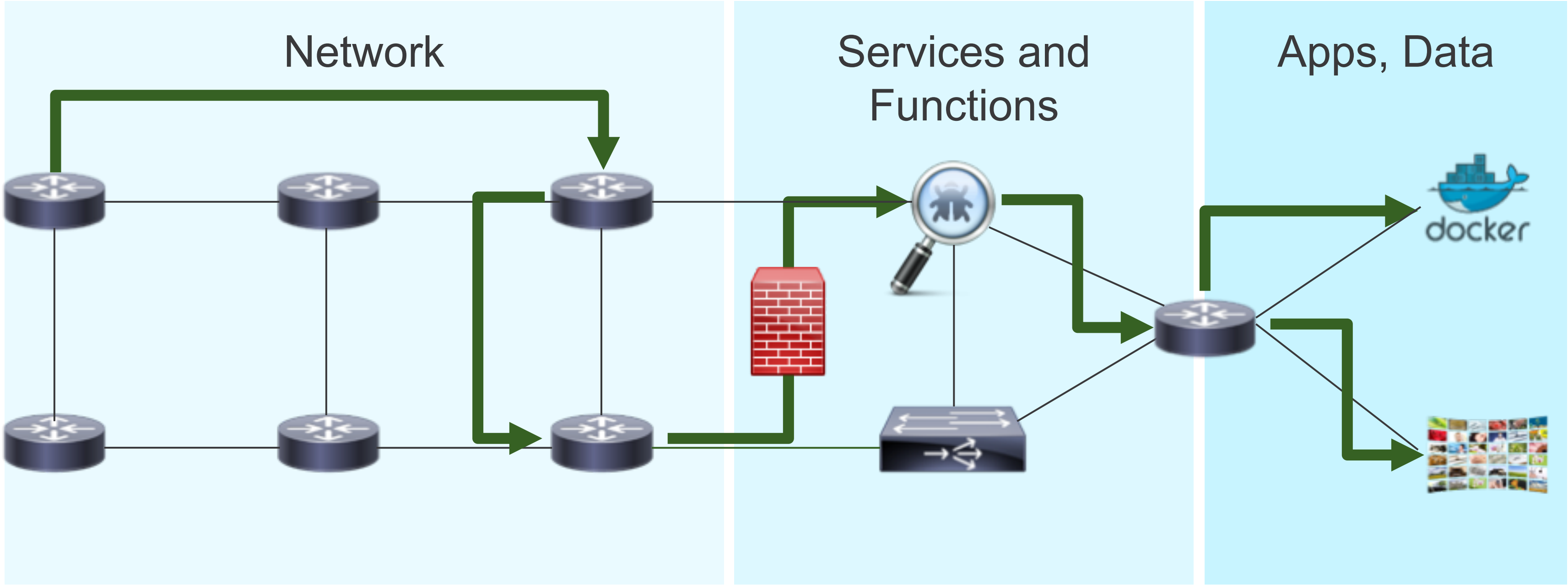
You’ve all heard of tunnels…



- GRE
- VxLAN
- L2TP, L2F, PPTP
- Geneve
- LISP
- GTP
- Mobile IP
- IPinIP
- 6rd, MAP-E
- …

IPv6 Segment Routing

One source address + a list of “way points” targeting a final destination



IPv6

Segment Routing

+ Vector Packet Processor

= Workload Balancing

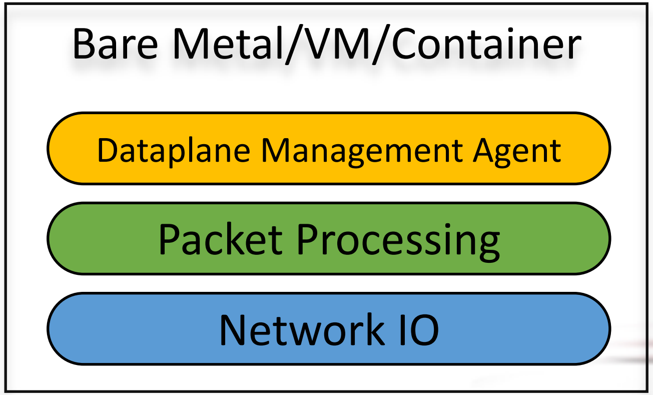
VPP

FD.io: VPP, The Universal Dataplane

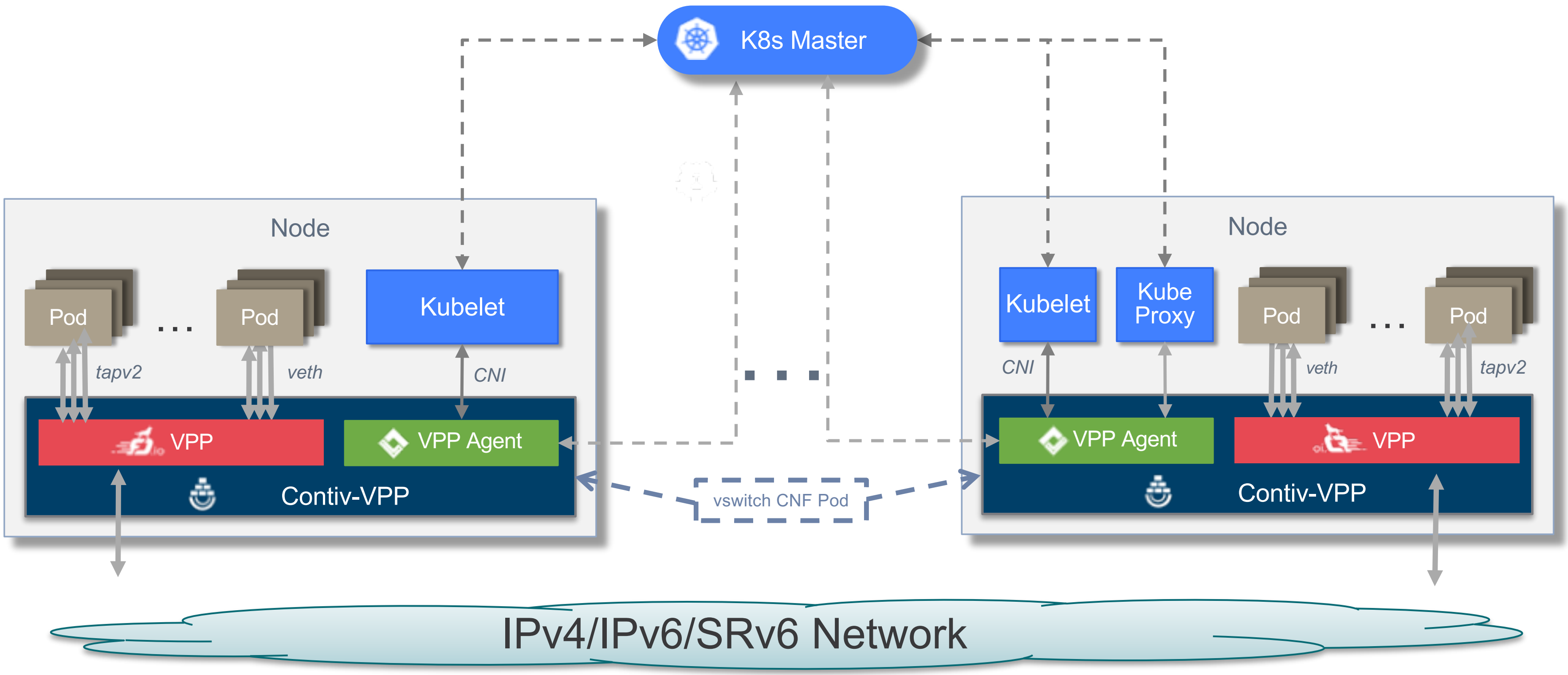
- Project at Linux Foundation
  - Multi-party
  - Multi-project
- Software Dataplane
  - High throughput
  - Low Latency
  - Feature Rich
  - Resource Efficient
  - Bare Metal/VM/Container
  - Multiplatform

intel ARM Power

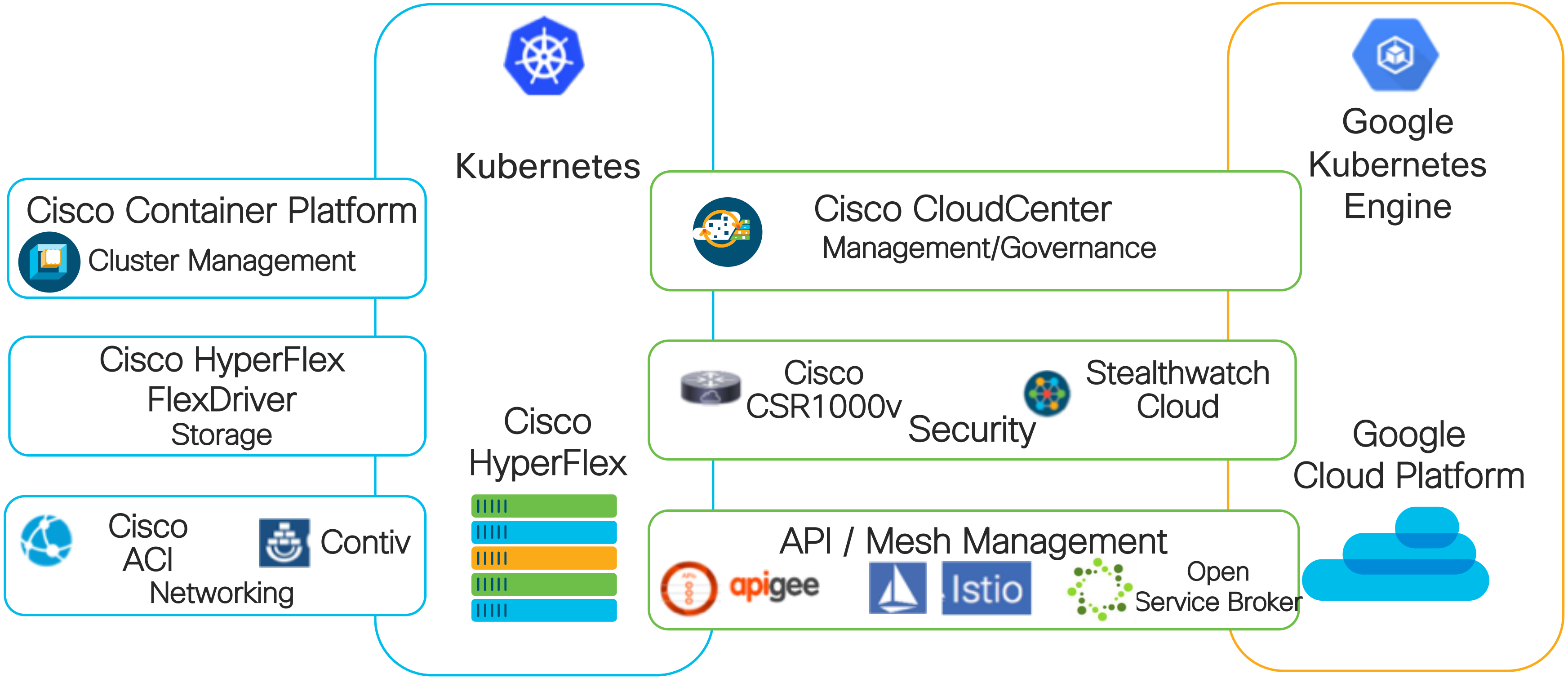
- Fd.io Scope:
  - **Network IO -** NIC/vNIC <-> cores/threads
  - **Packet Processing –** Classify/Transform/Prioritize/Forward/Terminate
  - **Dataplane Management Agents -** ControlPlane



Contiv-VPP: K8s Microservice Networking



Integrated solution: enterprise <-> cloud



Kubernetes

Cisco Container Platform
Cluster Management

Cisco HyperFlex
FlexDriver
Storage

Cisco
ACI
Networking

Contiv

Cisco
HyperFlex

Cisco CloudCenter
Management/Governance

Cisco
CSR1000v

Security

Stealthwatch
Cloud

API / Mesh Management

apigee

Istio

Open
Service Broker

Google
Kubernetes
Engine

Google
Cloud Platform

SRv6LB

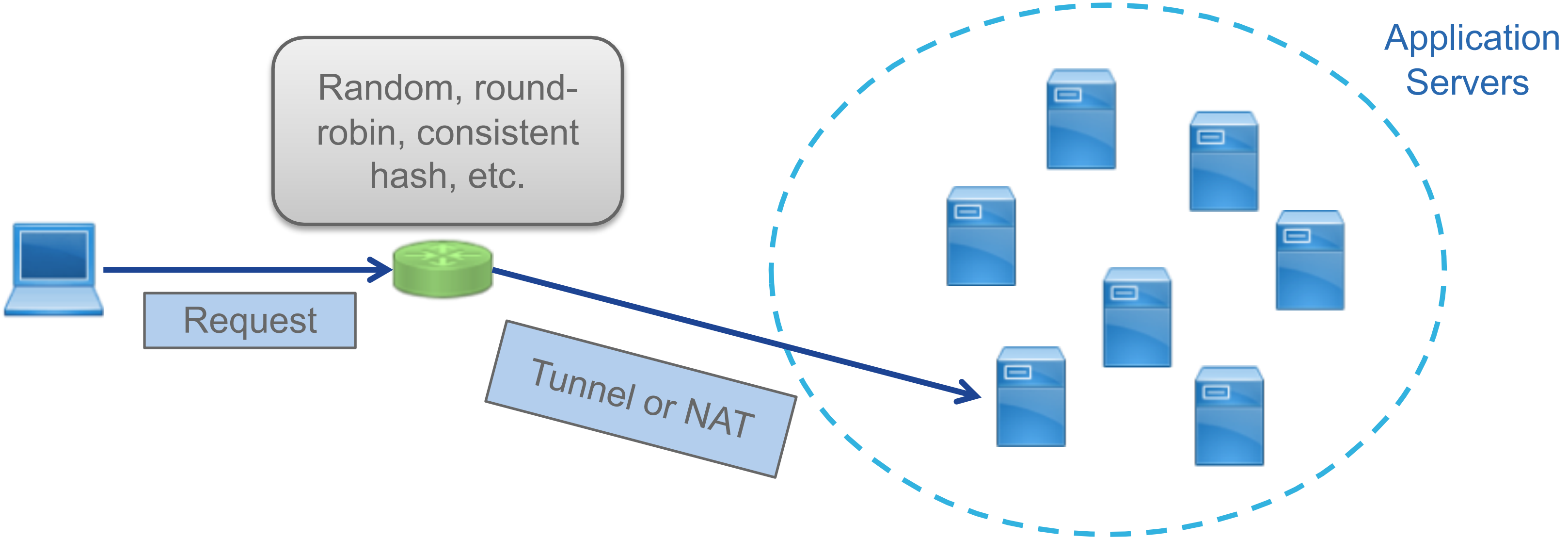
IPv6

Segment Routing

+ Vector Packet Processor

= Workload Balancing

L4 Load Balancing (w/o monitoring)



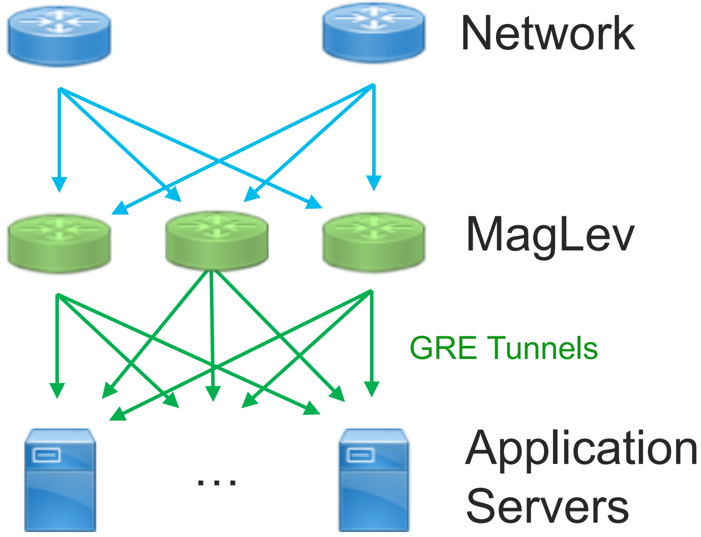
Google’s Maglev* is a very nice example of this kind of Load Balancer.

Self-described as “”Embarrassingly Distributed”

*Maglev: A Fast and Reliable Software Network Load Balancer https://www.usenix.org/node/194967 - March 2016

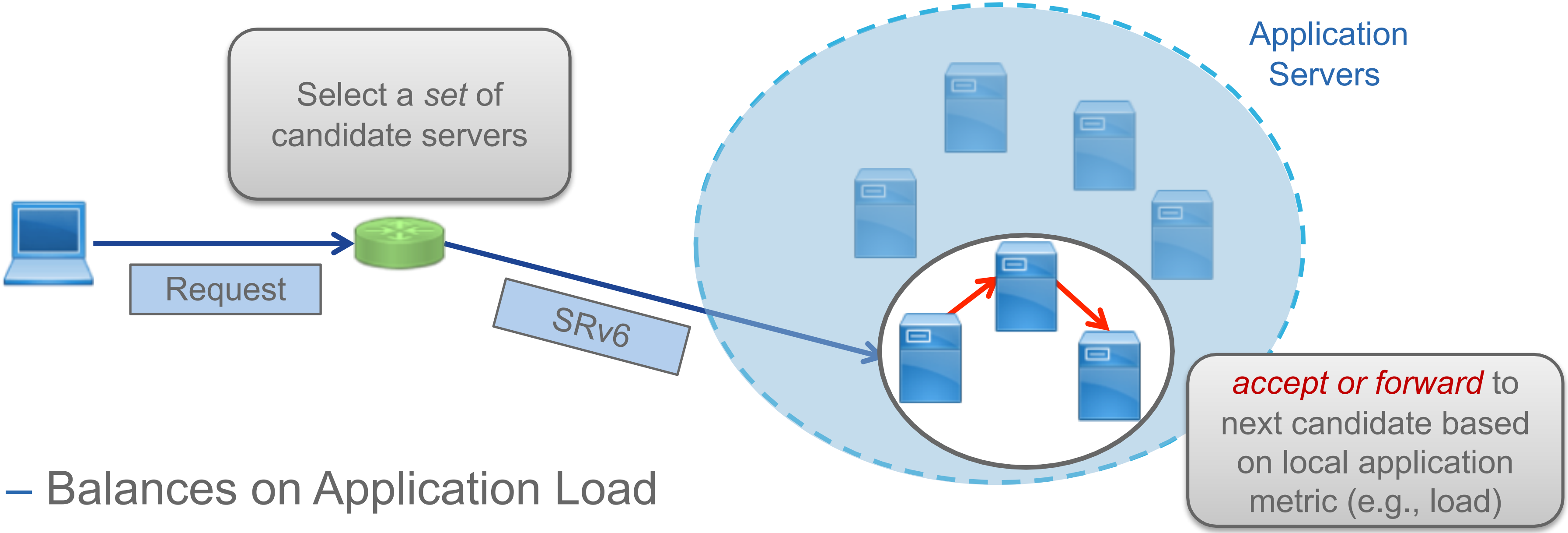
Maglev* – Google's L4 load Balancer

- Network (per-path)
  - Per-path ECMP (Equal Cost Multipath)
- MagLev (per-flow)
  - Pseudo Random consistent hashing
  - Flow Table stickiness
  - Unaware of application load
- Application Servers
  - Terminates GRE Tunnels for upstream traffic from MagLev
  - Return traffic routed directly (DSR)



*Maglev: A Fast and Reliable Software Network Load Balancer https://www.usenix.org/node/194967 - March 2016

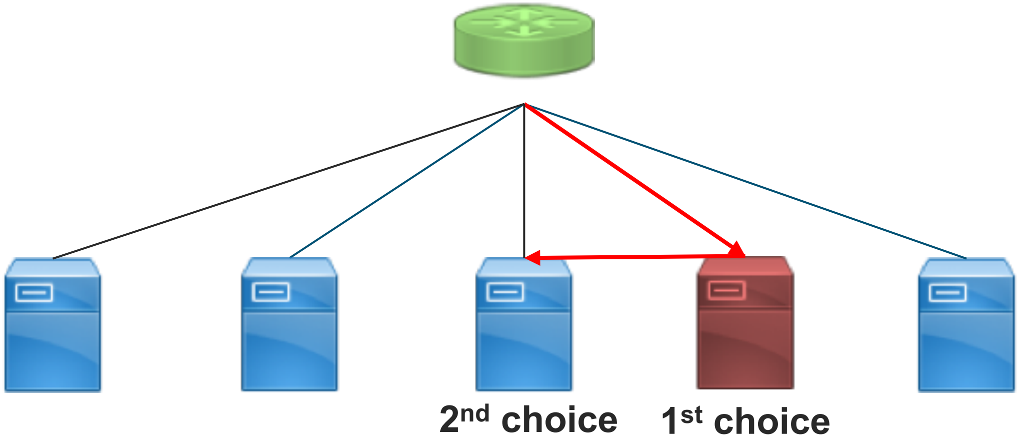
SRv6LB: “Built-in” Load Balancing


Select a *set* of candidate servers
Request
SRv6
Application Servers
accept or forward to next candidate based on local application metric (e.g., load)

- Balances on Application Load
- Without Application Monitoring
- Can use any application metric (# threads, CPU %, queue depth…)

Power of 2 Choices

- The Power of 2 Choices* shows that moving from a single random choice to two random choices can be very powerful

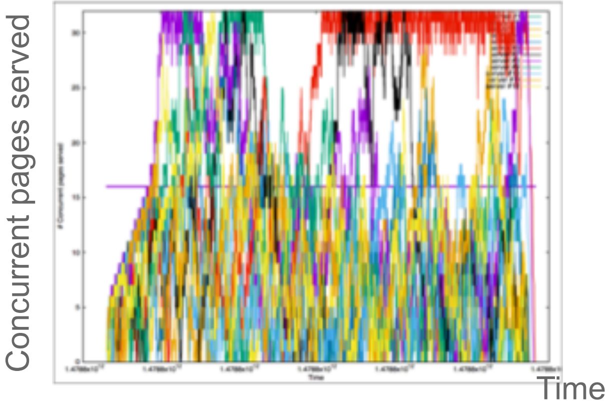


*M. Mitzenmacher, “The power of two choices in randomized load balancing,” IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 10, pp. 1094–1104, 2001.

Fairer balancing across servers

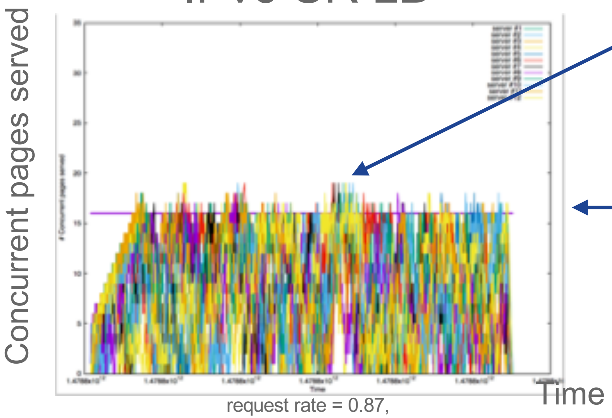
“unlucky” requests

Maglev



IPv6 SR LB

These requests got “unlucky twice”

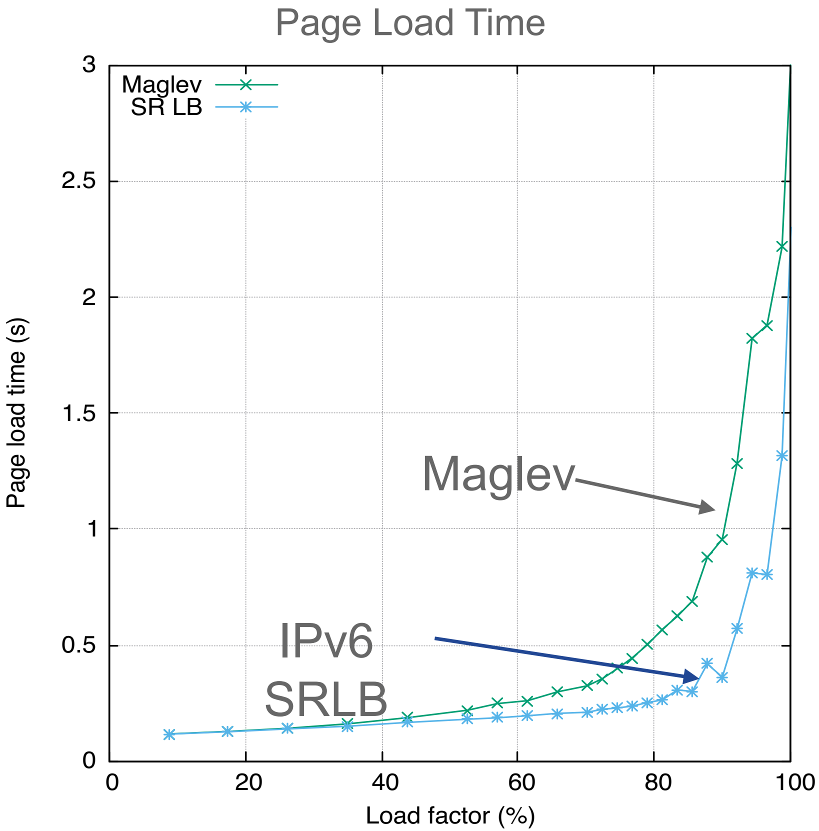


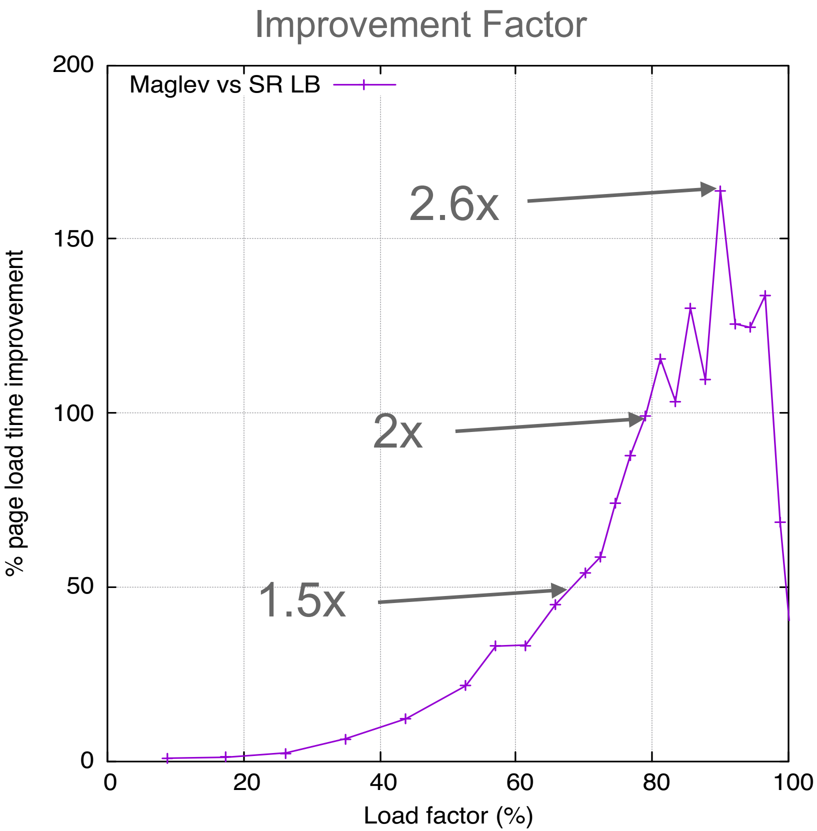
request rate = 0.87,
policy = (X<=16)

Target Busy Threshold

- Maglev: a server can get overloaded (purple, green and red lines)
- SRv6LB: better distributes the same number of queries between all servers

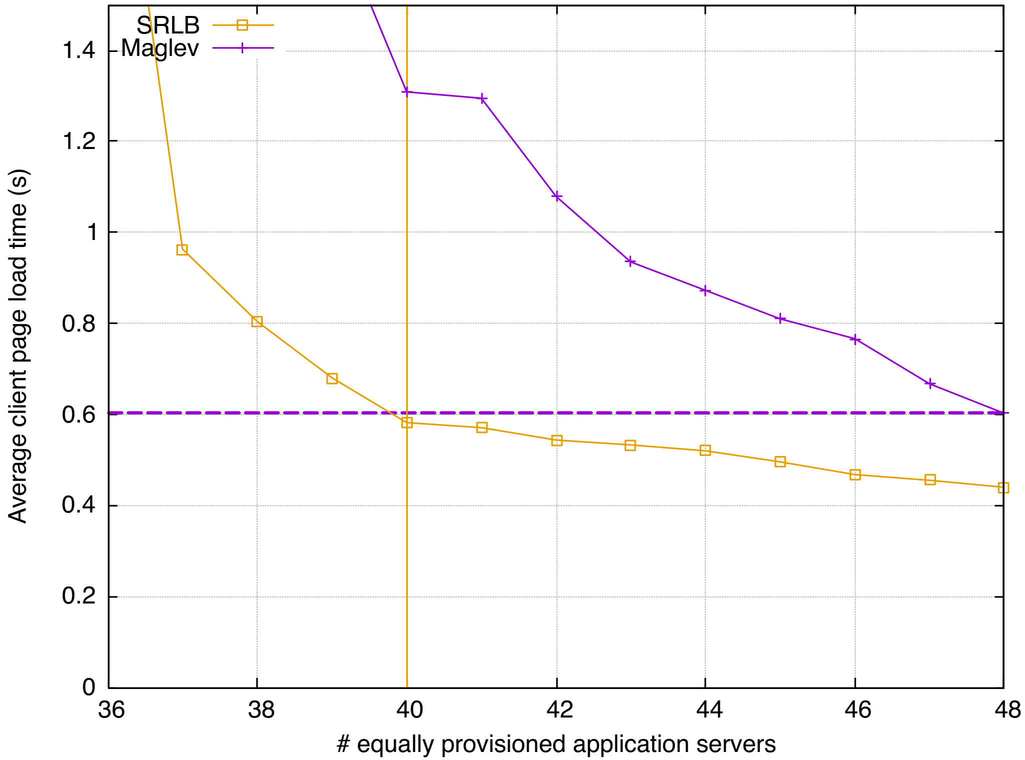
Improved page-load for a given set of servers





20000 requests, X=4

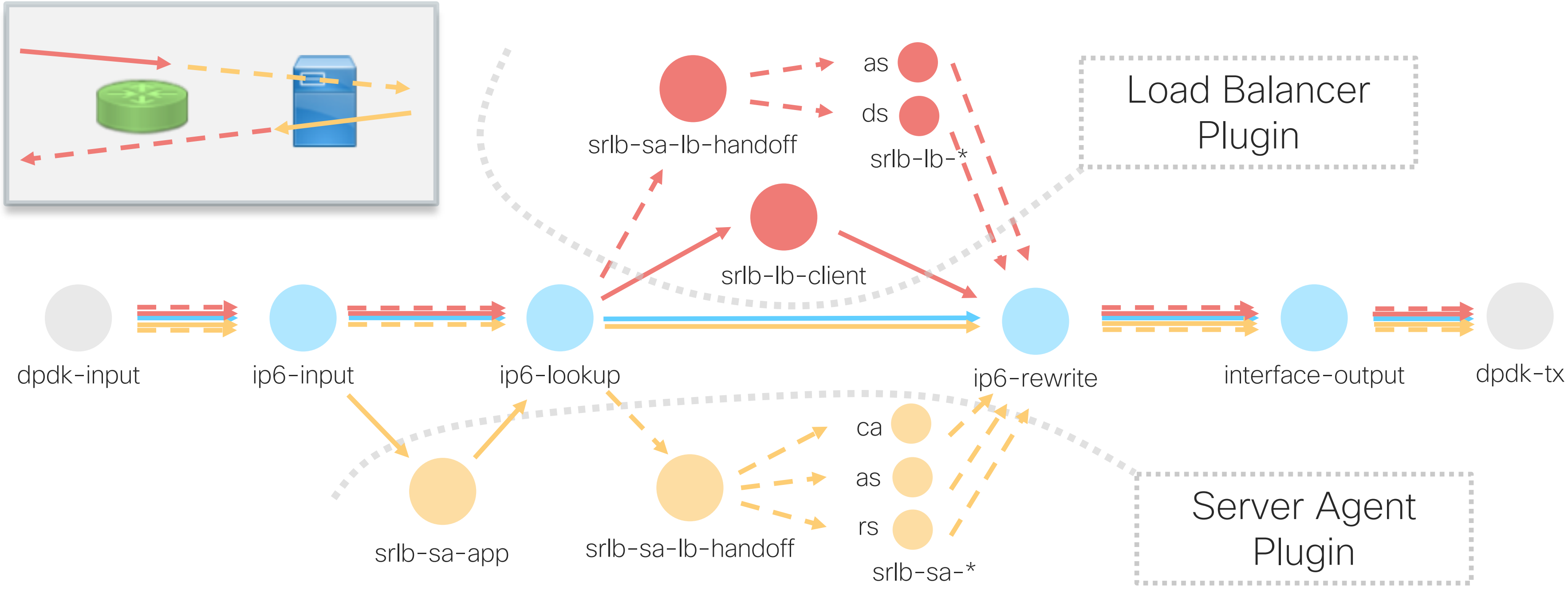
Fewer server instances for a given target SLA



For a given request rate from clients, SRLB with 40 server instances (one VM per CPU core) yields the same average page load time for clients as Maglev with 48 server instances.

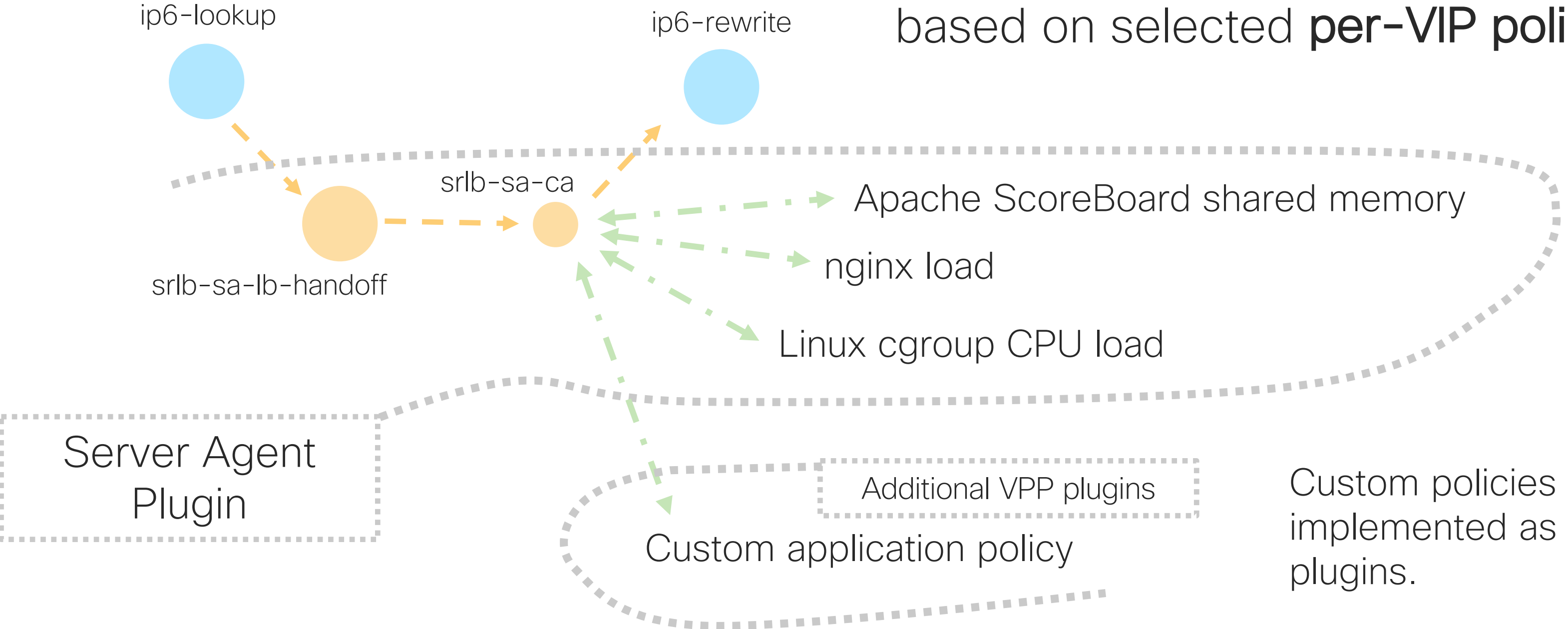
SRv6LB Implementation in FD.io's VPP

- VPP is a DPDK based fast Virtual Router
- SRv6LB implemented as 2 plugins



Application-specific connection acceptation policy.

When SRv6LB « Connect if Available » function is invoked, accept decision is based on selected **per-VIP policy**.



Apache ScoreBoard shared memory

nginx load

Linux cgroup CPU load

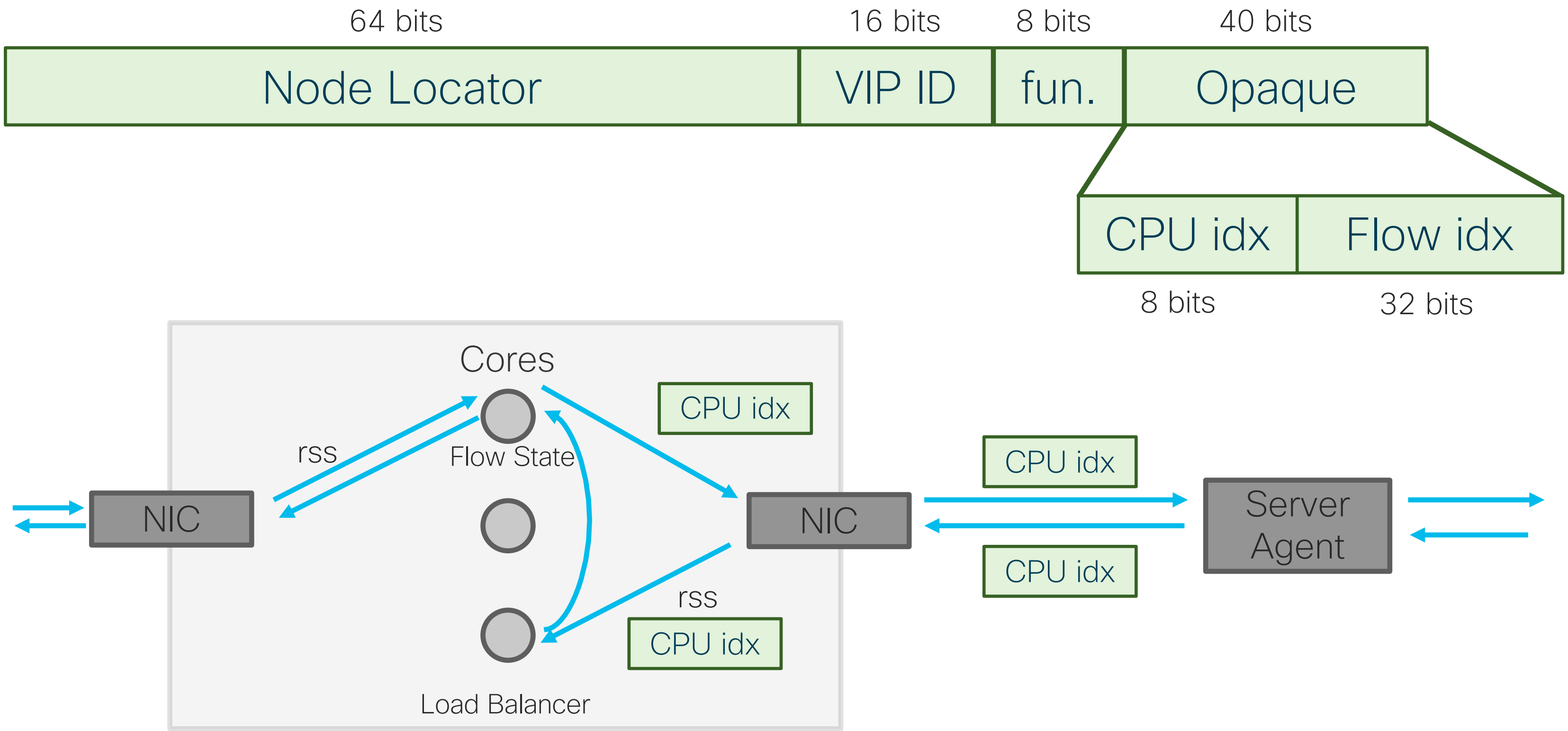
Server Agent Plugin

Additional VPP plugins

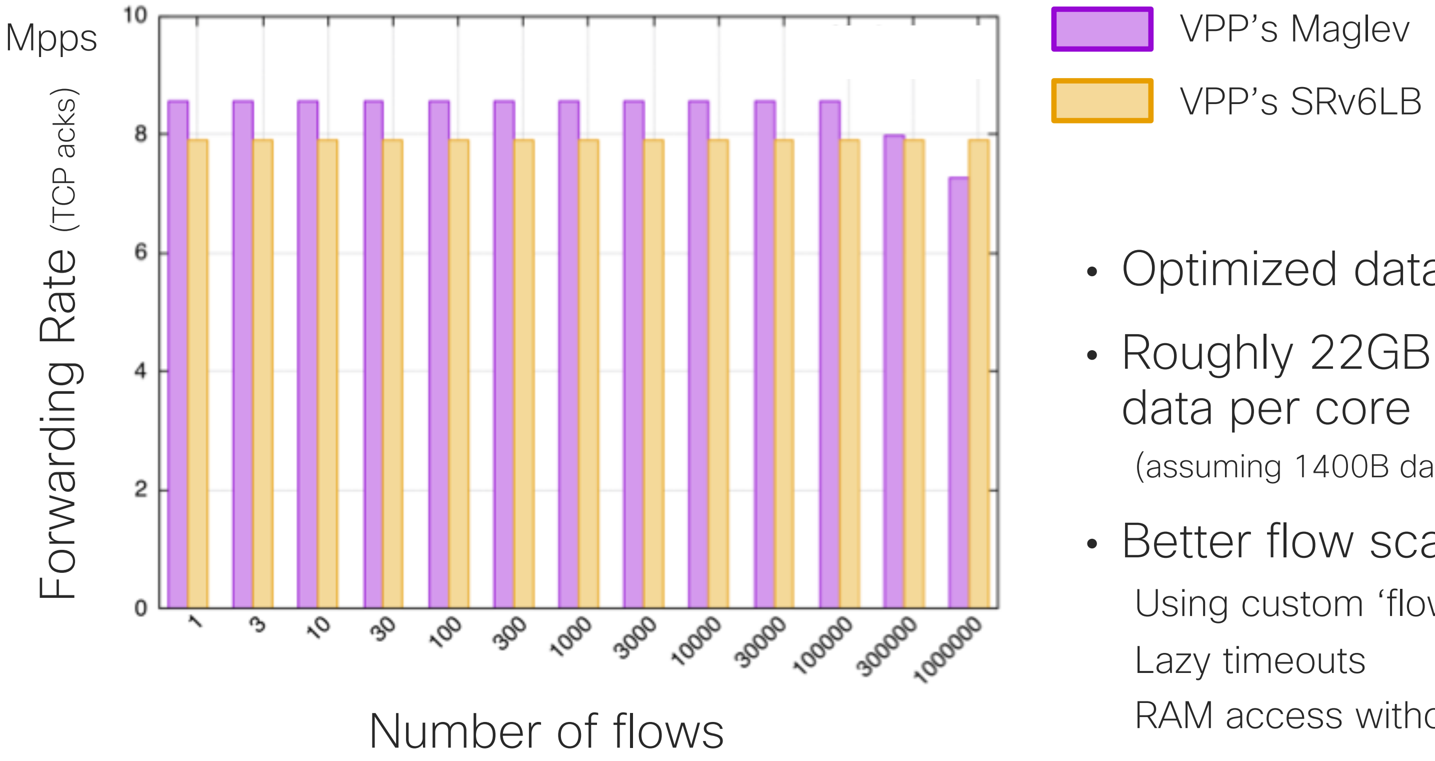
Custom application policy

Custom policies may be implemented as additional VPP plugins.

IPv6 used for CPU and flow steering



One million flows with VPP on a single core



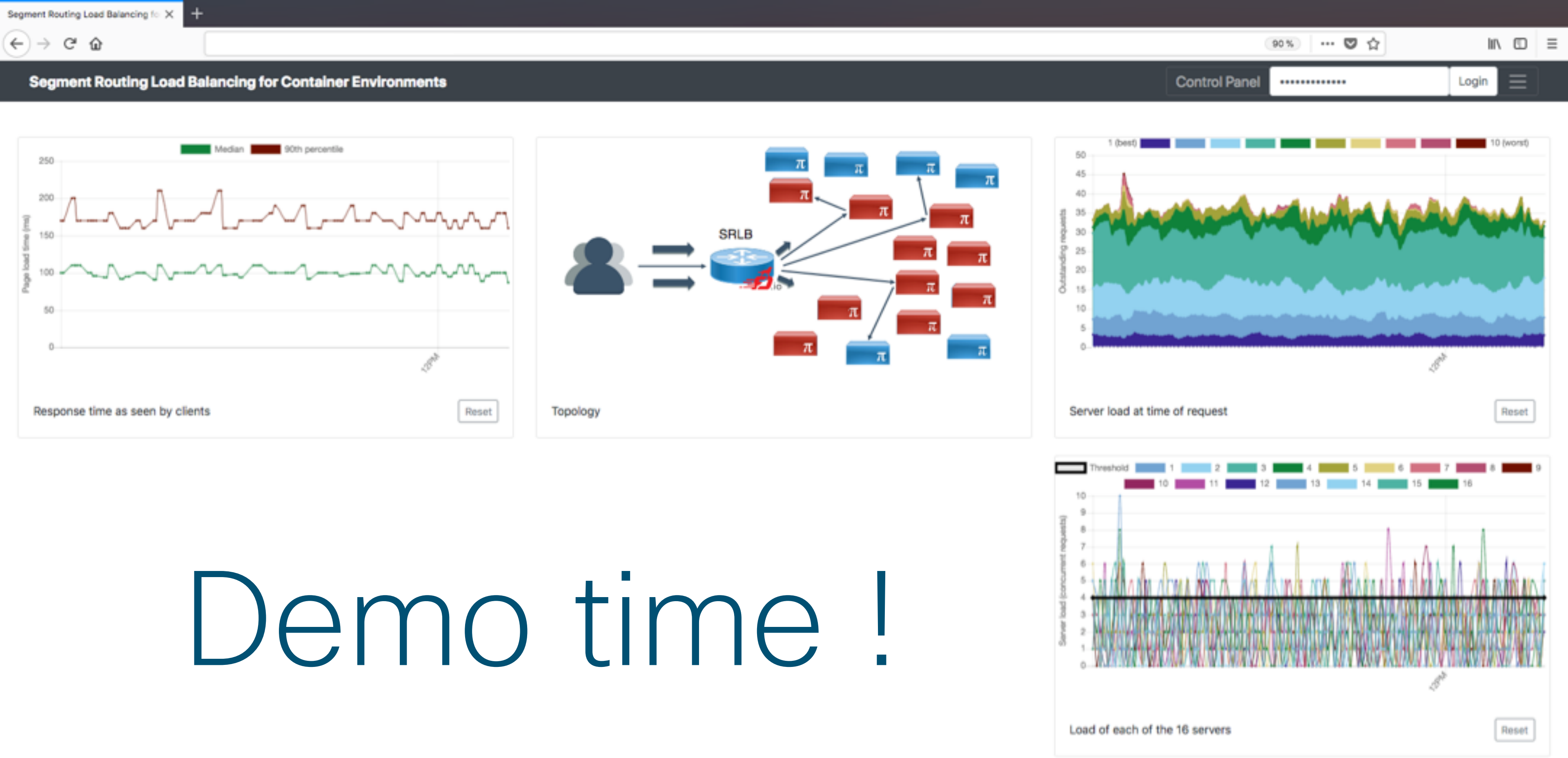
- Optimized data-path
- Roughly 22GBps downstream data per core

  (assuming 1400B data packets)
- Better flow scalability

  Using custom ‘flowhash’ table

  Lazy timeouts

  RAM access without perf. degradation.



Segment Routing Load Balancing for Container Environments

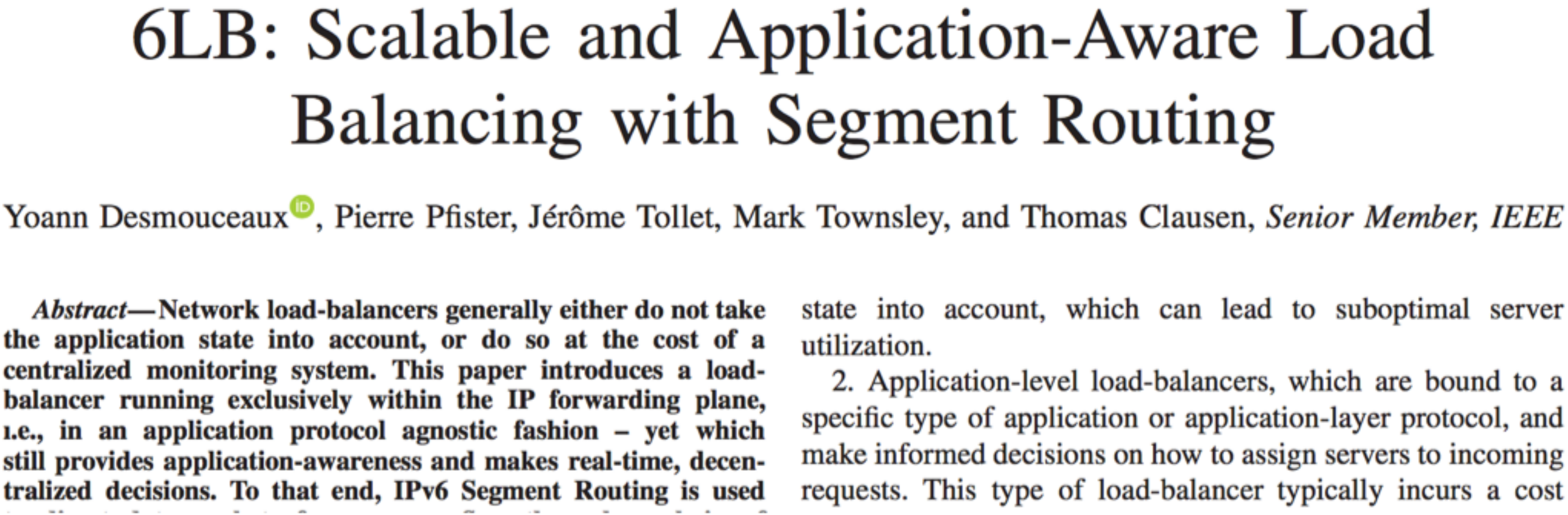
Response time as seen by clients

Topology

Server load at time of request

Load of each of the 16 servers

Demo time !



6LB: Scalable and Application-Aware Load Balancing with Segment Routing

Yoann Desmouceaux, Pierre Pfister, Jérôme Tollet, Mark Townsley, and Thomas Clausen, *Senior Member, IEEE*

***Abstract*—Network load-balancers generally either do not take the application state into account, or do so at the cost of a centralized monitoring system. This paper introduces a load-balancer running exclusively within the IP forwarding plane, i.e., in an application protocol agnostic fashion – yet which still provides application-awareness and makes real-time, decentralized decisions. To that end, IPv6 Segment Routing is used**

state into account, which can lead to suboptimal server utilization.

2. Application-level load-balancers, which are bound to a specific type of application or application-layer protocol, and make informed decisions on how to assign servers to incoming requests. This type of load-balancer typically incurs a cost

Published in:
IEEE/ACM Transactions on Networking
(Volume: 26, Issue: 2, April 2018)

Research Gate link (no paywall)
http://cs.co/6LB-Paper