ProActive Routing In Scalable Data Centers with PARIS

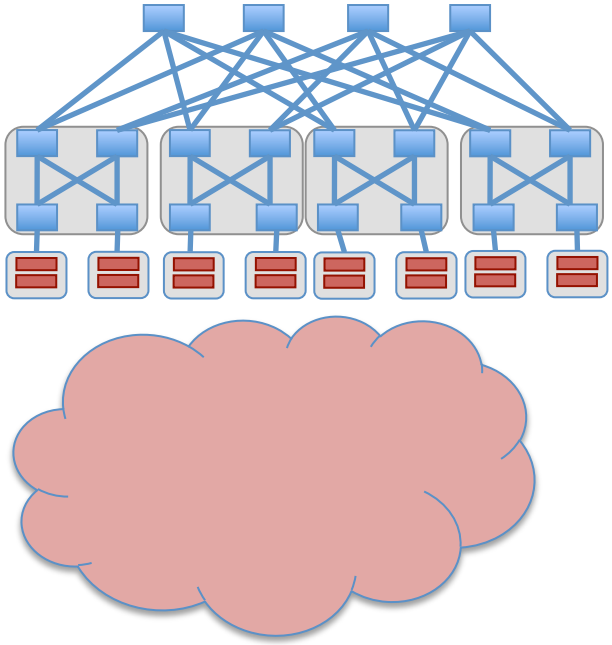
Theophilus Benson
Duke University

Joint work with Dushyant Arora[+] and Jennifer Rexford*

[+]Arista Networks
*Princeton University

Data Center Networks Must …

- Support diverse application
  - High throughput/low latency
  - Utilize multiple paths



- Scale to cloud size
  - 5-10 million VMs

- Support flexible resource utilization
  - Support seamless VM mobility

Evolution of Data Center Networks…

	Scalable	Seamless mobility	Multipath routing
Layer 2: Flat Addresses	−	+	−
Layer 3: Hierarchical Addresses	+	−	+
Overlays: VL2/Portland	−	+	+
PARIS	+	+	+

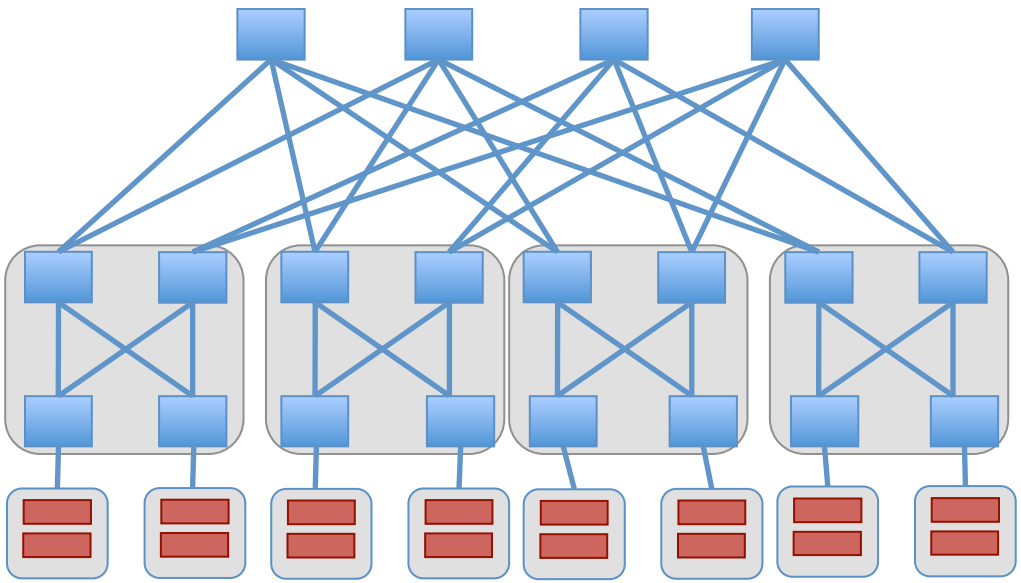
PARIS in a Nutshell…

- PARIS is a scalable and flexible flat layer 3 network fabric.
- PARIS hierarchically partitions addresses at the core
- PARIS runs on a data center of commodity switches

Outline

- Evolution of Data Center Networks
- PARIS Architecture
- Evaluation and Conclusion

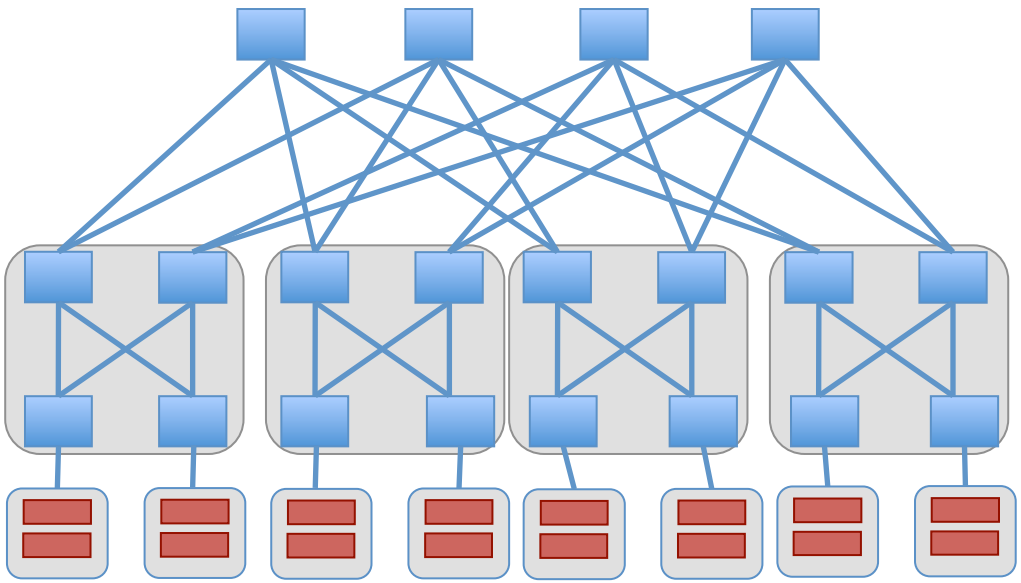
Evolution of Data Center Networks



Not scalable
Seamless mobility
No Multipath

- Flat layer 2: Spanning Tree
  - Uses flooding to discover location of hosts
    - Supports seamless VM migration
  - Traffic restricted to single network path

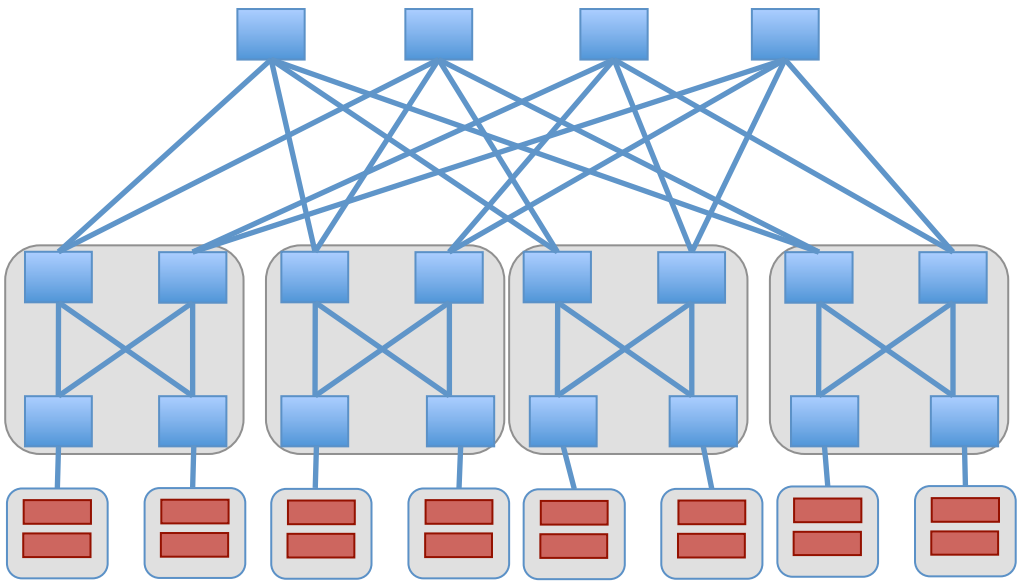
Evolution of Data Center Networks



Scalable
No seamless mobility
Multipath

- Layer 3:Hierarchical Addresses
  - Host locations are predefined
  - During VM mobility, IP-addresses change
  - Load balances over k shortest paths

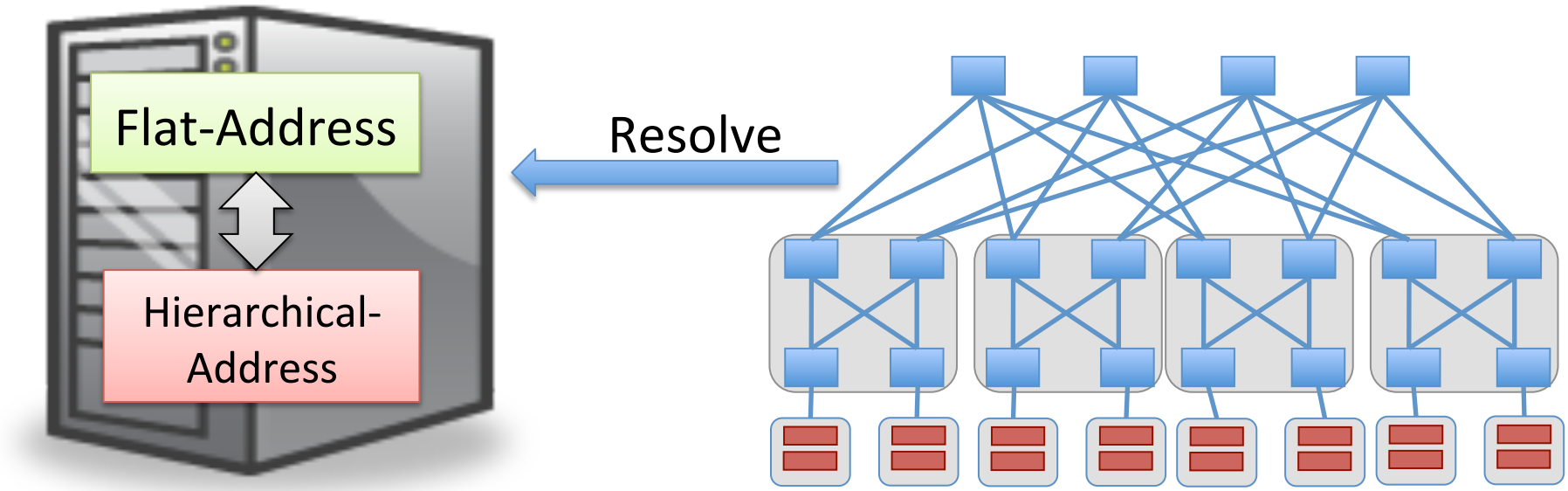
Evolution of Data Center Networks



Seamless mobility
Multipath
Not scalable

- Overlay solutions: Portland/VL2
  - Uses two addressing schemes:
    - hierarchical addresses: for routing traffic
    - flat addresses: for identifying VMs

Overheads introduced by Overlays Solutions…



- Address resolution infrastructure
  - Inflated flow startups times
- Switch CPU for encapsulation
- Switch storage for caching address resolutions

Evolution of Data Center Networks…

	Scalable	Seamless mobility	Multipath routing
Layer 2: Flat Addresses	−	+	−
Layer 3: Hierarchical Addresses	+	−	+
Overlays: VL2/Portland	−	+	+

Challenges..

Develop data center network that supports benefits of overlay routing while eliminating ..

- Overheads of caching and packet-encapsulation
- Overheads of address translation

ProActive Routing In Scalable

PARIS Architecture

Architectural Principles

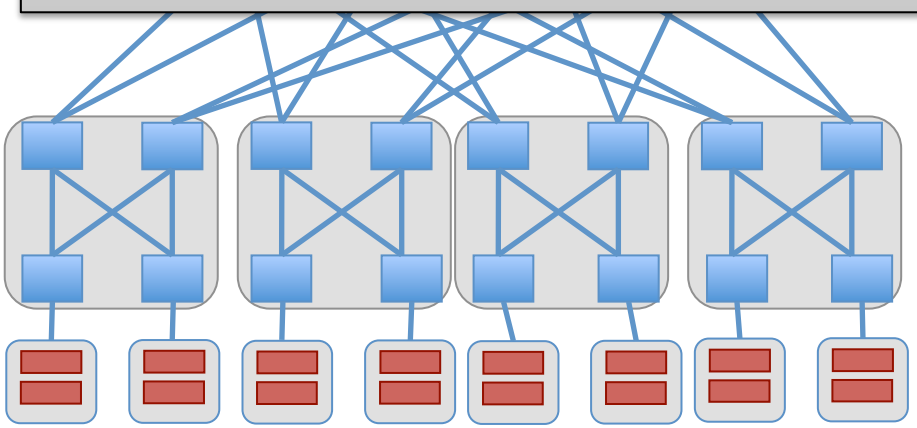
- Flat layer-three network
  - Allows for seamless VM mobility
- Proactive installation of forwarding state
  - Eliminates startup latency overheads
- Hierarchical partitioning of network state
  - Promotes scalability

Paris Architecture

Network Controller:
- Monitors network traffic
- Performs traffic engineering
- Tracks network topology

Overheads eliminated

- Pro-active rule installation → No start-up delay for switch rule installation
- No addresses indirection → No address resolution, encapsulation, caching
- /32 network addresses → No broadcast traffic; no ARP



Switches:
- Support ECMP
- Programmable devices

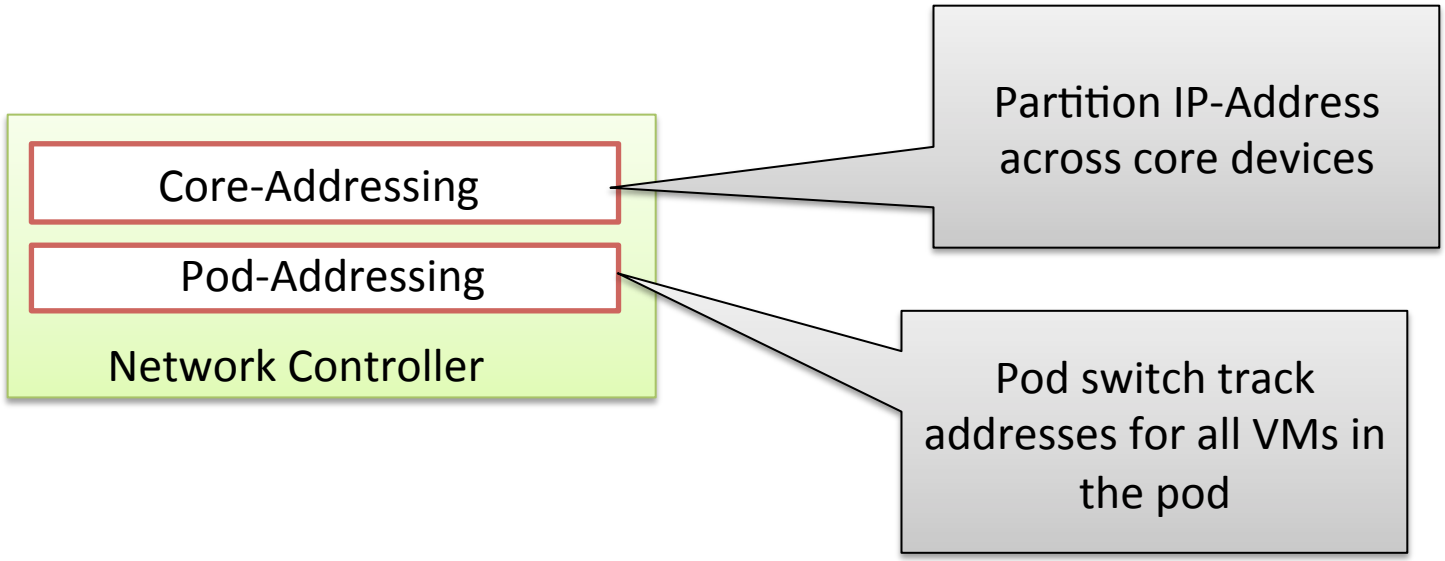
End-Hosts:
- /32 addresses
- Default GW: edge switch

Evolution of Data Center Networks…

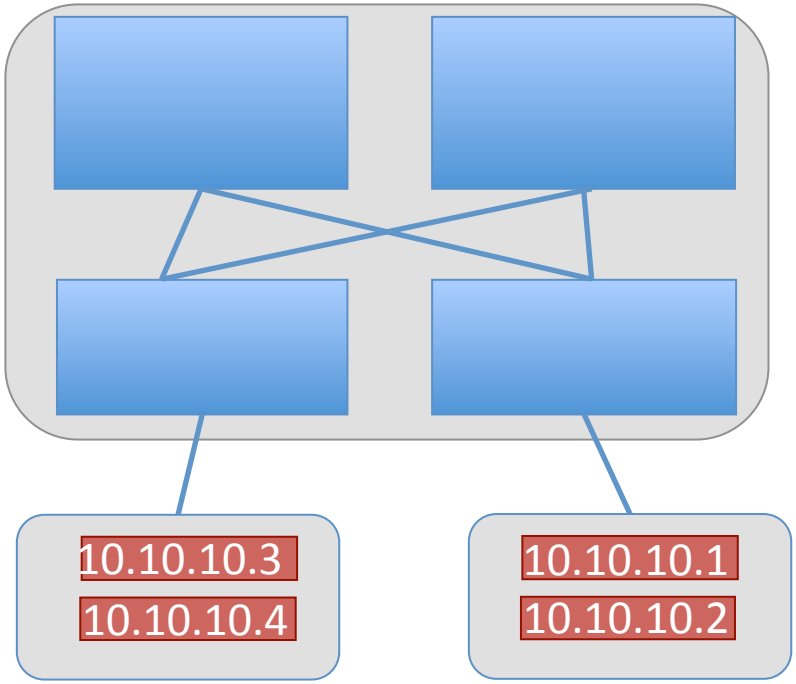
	Scalable	Seamless mobility	Multipath routing
Layer 2: Flat Addresses	−	+	−
Layer 3: Hierarchical Addresses	+	−	+
Overlays: VL2/Portland	−	+	+
PARIS	+	+	+

Paris Network Controller

- Switches have 1 million entries
  - But data center has 5-10 million VMs
  - Each pod has ~100K VMs

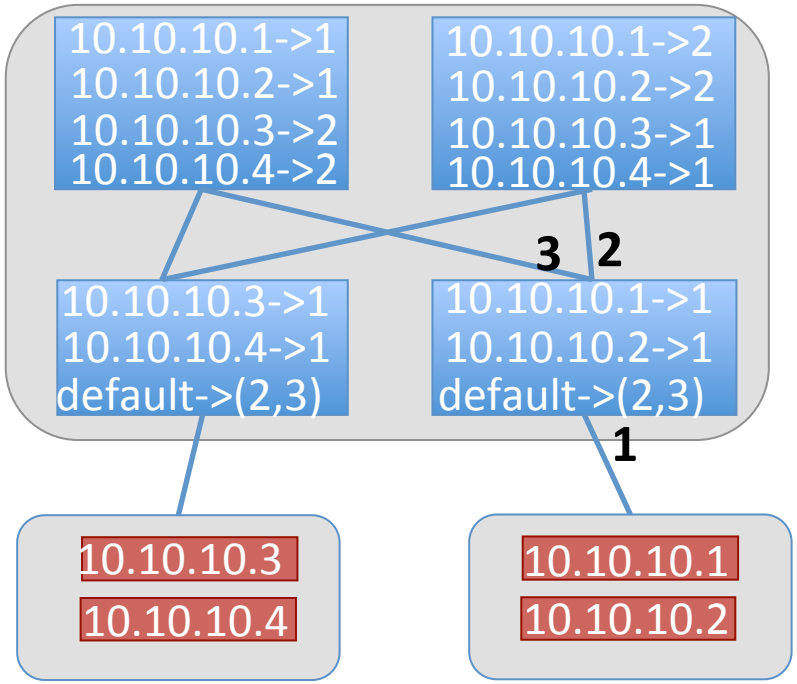


Pod-Addressing Module



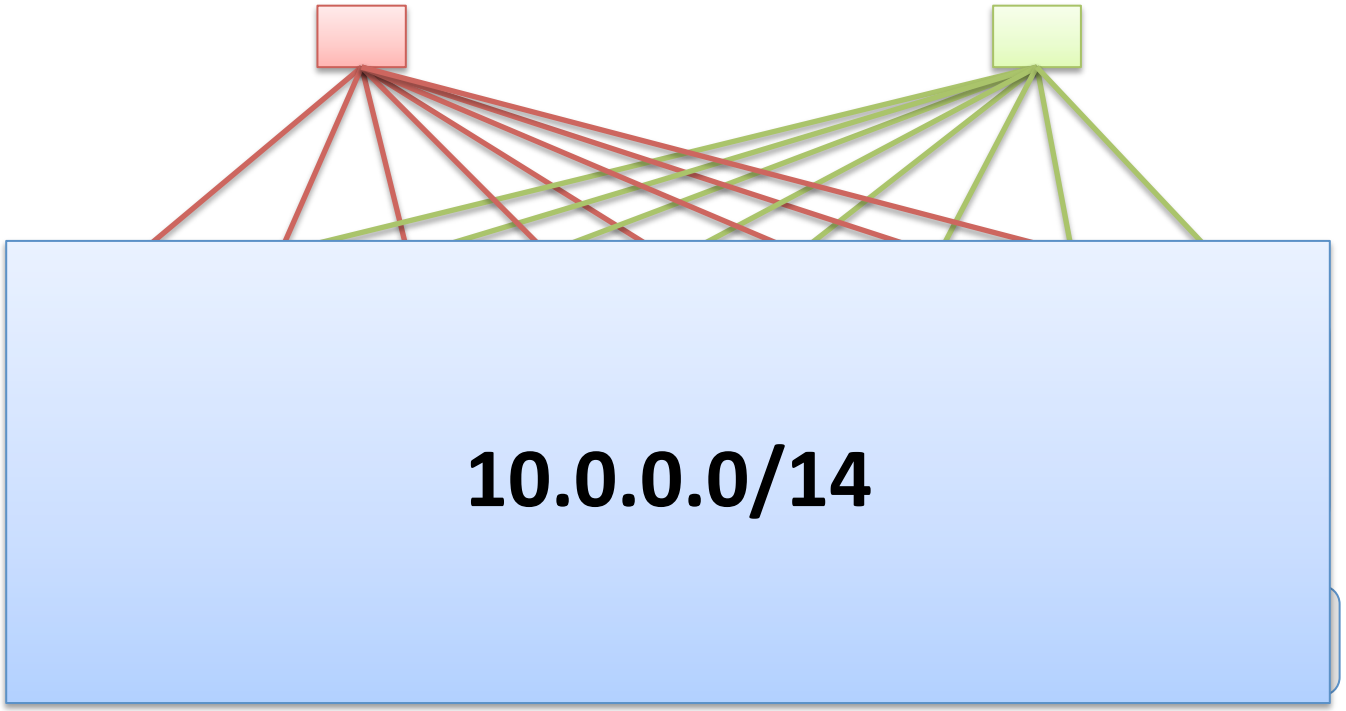
- Edge & aggregation addressing scheme

Pod-Addressing Module



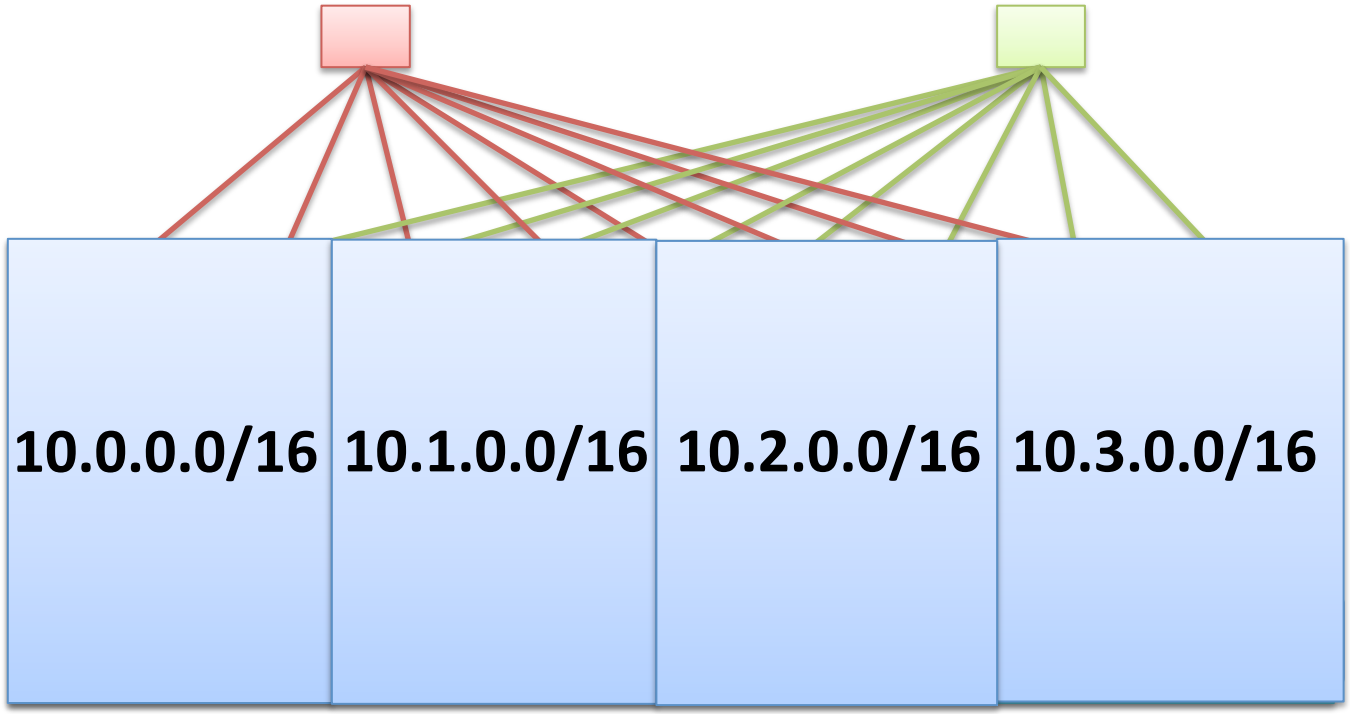
- Edge & aggregation addressing scheme
  - Edge: stores address for all connected end-hosts
  - Agg: stores addresses for all end-hosts in pod

Core Addressing-Modules



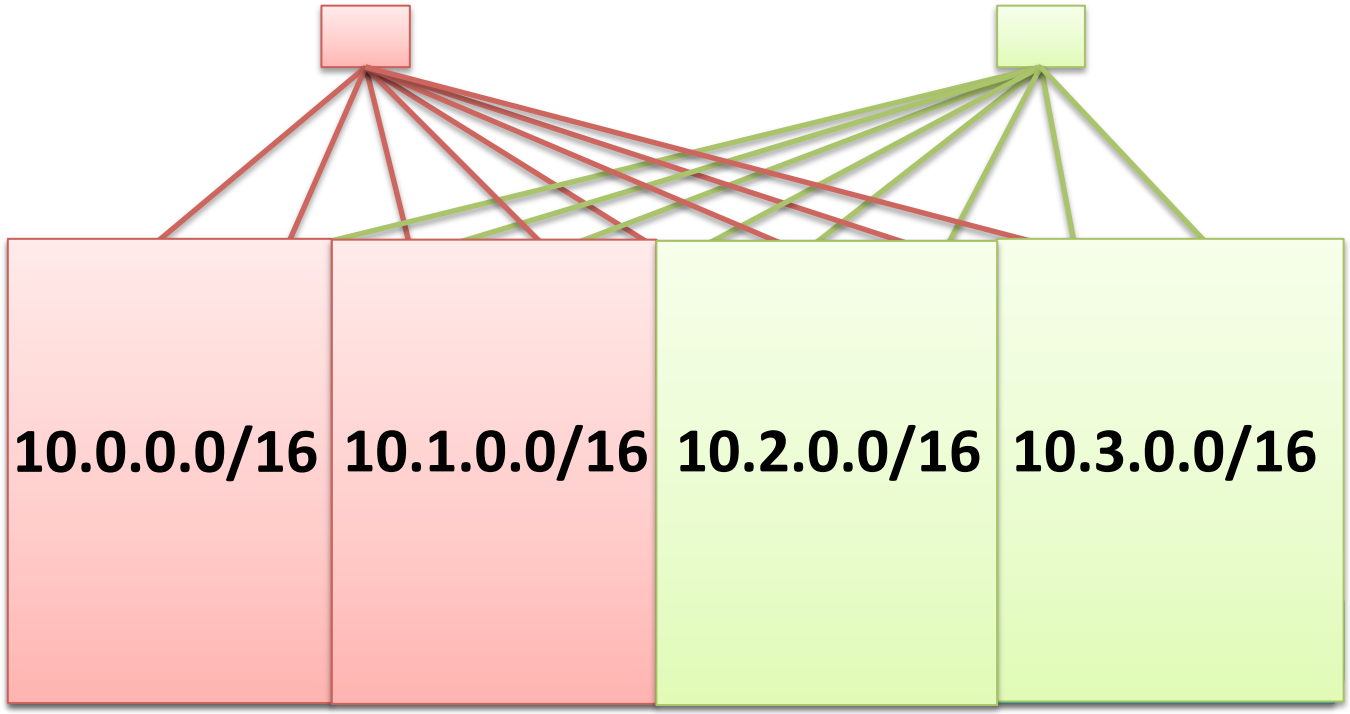
- Partitions the IP-space into virtual-prefix

Core Addressing-Modules

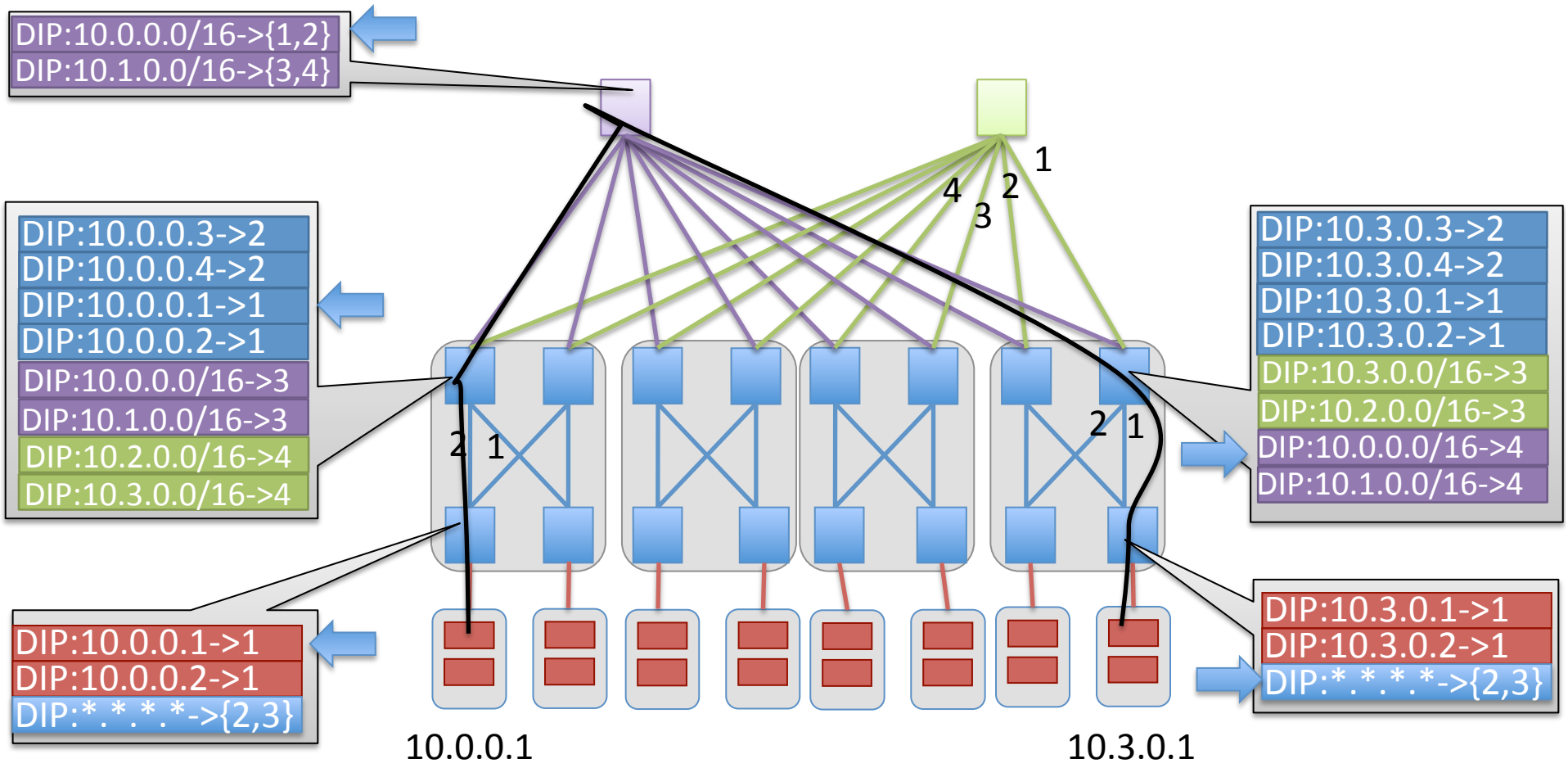


- Partitions the IP-space into virtual-prefix
- Each core is an Appointed prefix switch (APS)

Core Addressing-Modules

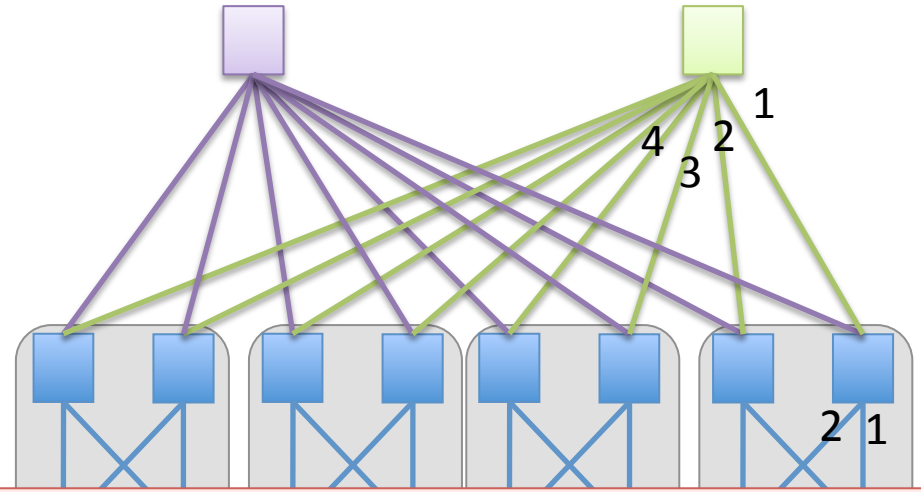


- Partitions the IP-space into virtual-prefix
- Each core is an Appointed prefix switch (APS)
  - Tracks all address in a virtual-prefix


DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}
DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:10.0.0.0/16->3
DIP:10.1.0.0/16->3
DIP:10.2.0.0/16->4
DIP:10.3.0.0/16->4
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}
1
2
3
4
2
1
2
1
DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:10.3.0.0/16->3
DIP:10.2.0.0/16->3
DIP:10.0.0.0/16->4
DIP:10.1.0.0/16->4
DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}
10.0.0.1
10.3.0.1

DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}

DIP:10.3.0.0/16->{1,2}
DIP:10.2.0.0/16->{3,4}



DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:10.0.0.0/16->3
DIP:10.1.0.0/16->3
DIP:10.2.0.0/16->4
DIP:10.3.0.0/16->4

DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:10.3.0.0/16->3
DIP:10.2.0.0/16->3
DIP:10.0.0.0/16->4
DIP:10.1.0.0/16->4

DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}

DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}

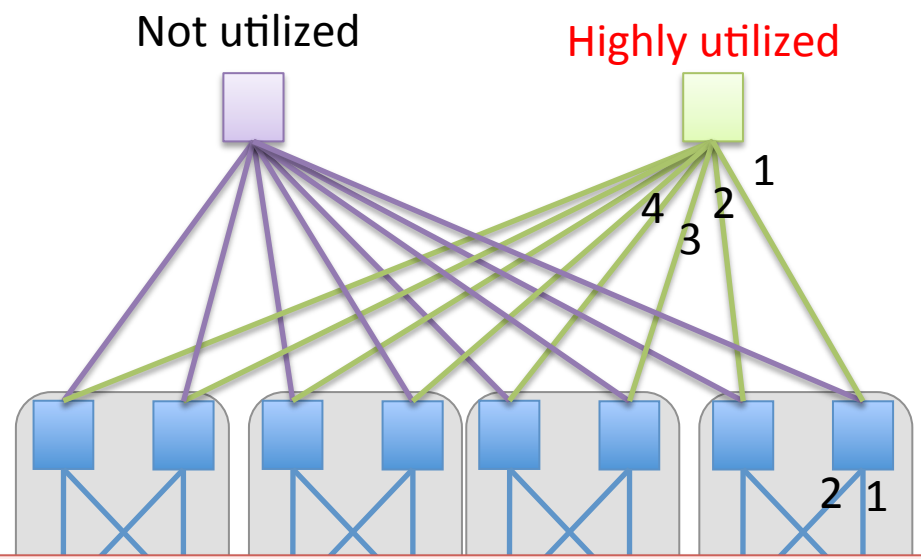
Limitations

- **No Load balancing between the core nodes**
- **Multi-path in core is not utilized!**

DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}

DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:10.0.0.0/16->3
DIP:10.1.0.0/16->3
DIP:10.2.0.0/16->4
DIP:10.3.0.0/16->4

DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}



DIP:10.3.0.0/16->{1,2}
DIP:10.2.0.0/16->{3,4}

DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:10.3.0.0/16->3
DIP:10.2.0.0/16->3
DIP:10.0.0.0/16->4
DIP:10.1.0.0/16->4

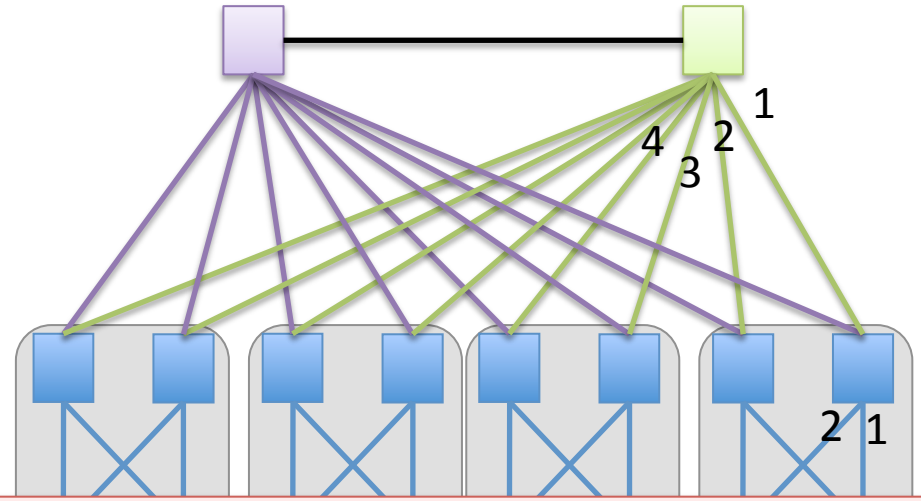
DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}

Limitations

- **No Load balancing between the core nodes**
- **Multi-path in core is not utilized!**

DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}

DIP:10.3.0.0/16->{1,2}
DIP:10.2.0.0/16->{3,4}



DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1

DIP:10.0.0.0/14->{3,4}

DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.1->1
DIP:10.3.0.2->1

DIP:10.0.0.0/14->{3,4}

High-BW PARIS:

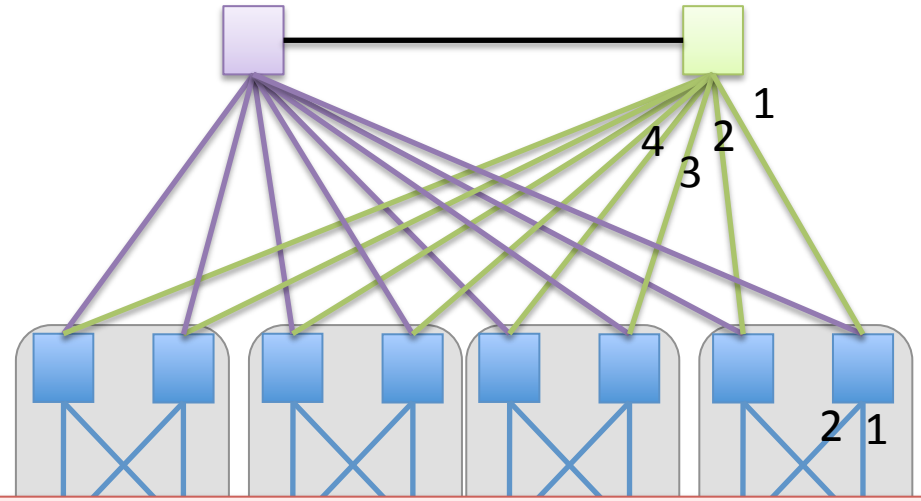
- **Connect core nodes in a mesh**
- **Change rules at aggregation to load balance across core nodes**
- **Use Valiant Load-balancing in the core**

DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}

DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}

DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}

DIP:10.3.0.0/16->{1,2}
DIP:10.2.0.0/16->{3,4}



DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1

DIP:10.0.0.0/14->{3,4}

DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.1->1
DIP:10.3.0.2->1

DIP:10.0.0.0/14->{3,4}

High-BW PARIS:

- **Connect core nodes in a mesh**
- **Change rules at aggregation to load balance across core nodes**
- **Use Valiant Load-balancing in the core**

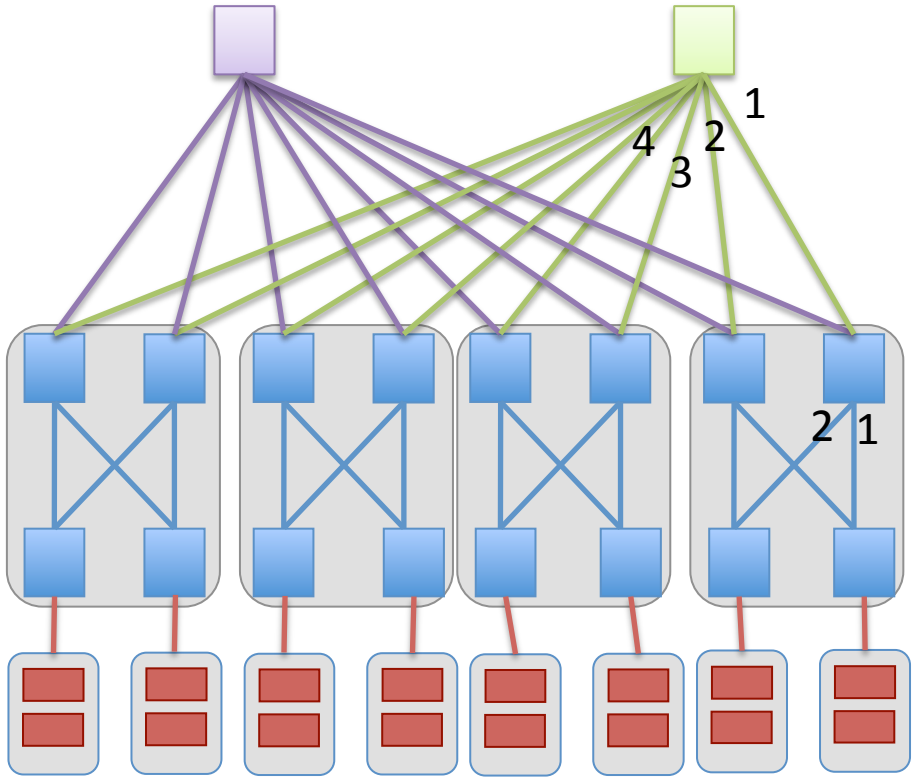
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}

DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}

DIP:10.0.0.0/16->{1,2}
DIP:10.1.0.0/16->{3,4}

DIP:10.3.0.1->1
DIP:10.0.0.3->2
DIP:10.0.0.4->2
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:10.0.0.0/16->3
DIP:10.1.0.0/16->3
DIP:10.2.0.0/16->4
DIP:10.3.0.0/16->4

DIP:10.3.0.1->1
DIP:10.0.0.1->1
DIP:10.0.0.2->1
DIP:*.*.*.*->{2,3}



DIP:10.3.0.1->{7,8}
DIP:10.3.0.0/16->{1,2}
DIP:10.2.0.0/16->{3,4}

DIP:10.3.0.3->2
DIP:10.3.0.4->2
DIP:10.3.0.2->1
DIP:10.3.0.0/16->3
DIP:10.2.0.0/16->3
DIP:10.0.0.0/16->4
DIP:10.1.0.0/16->4

DIP:10.3.0.1->1
DIP:10.3.0.2->1
DIP:*.*.*.*->{2,3}

Evaluation

Evaluation

- How does PARIS scale to large data centers?
- Does PARIS ensure good performance?
- How does PARIS perform under failures?
- How quickly does PARIS react to VM migration?

Evaluation

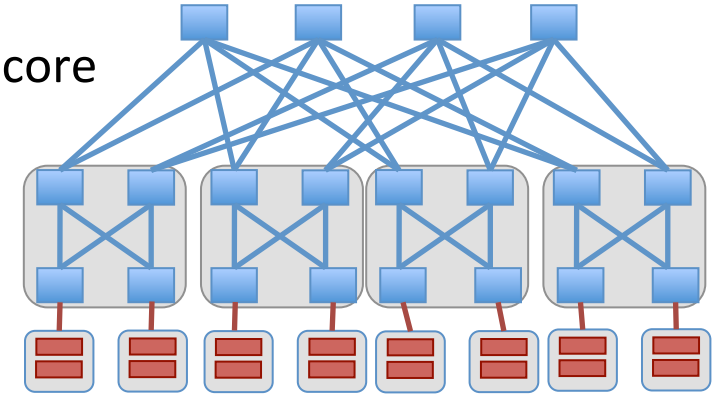
- How does PARIS scale to large data centers?
- Does PARIS ensure good performance?
- How does PARIS perform under failures?
- How quickly does PARIS react to VM migration?

TestBed

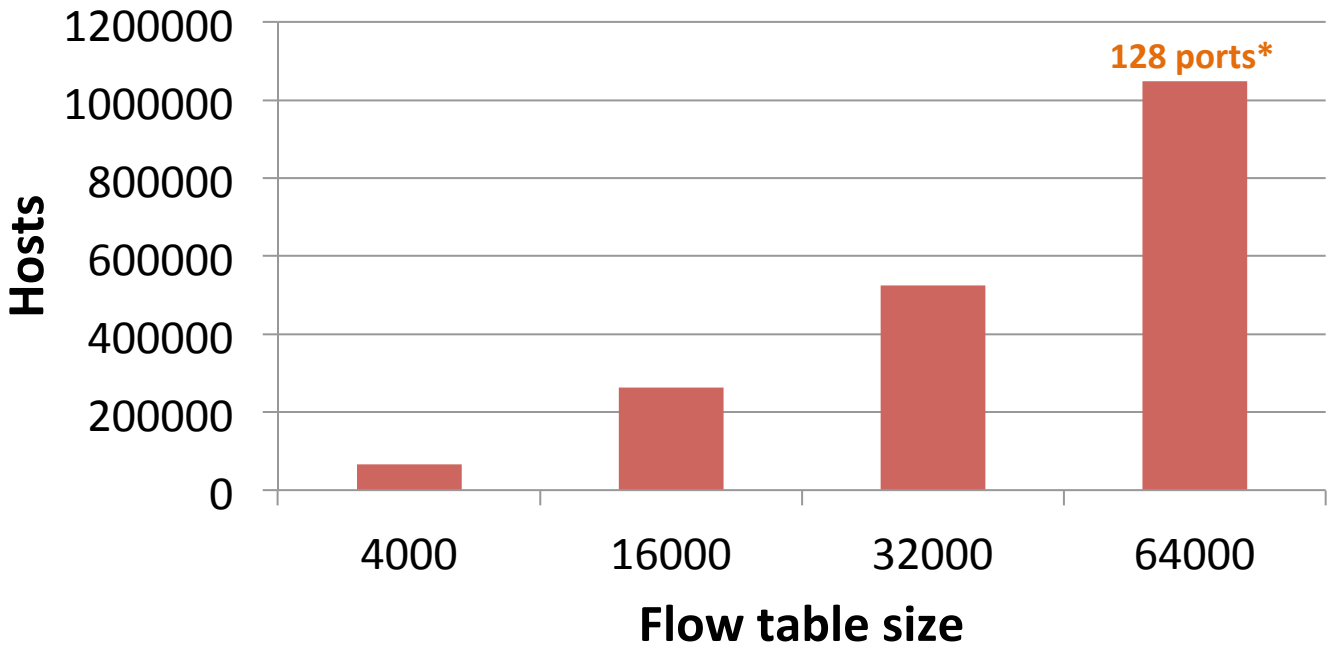
- Emulate data center topology using Mininet
  - Generate traffic using IPerf
  - Random traffic traffic matrix

- Implemented PARIS on NOX

- Data center topology
  - 32 hosts, 16 edge, 8 aggregation, and 4 core
  - No over-subscription
  - – Link capacity:
    - Server Uplinks: 1Mbps
    - Switch-Switch: 10Mbps



Scaling to Large Data Centers



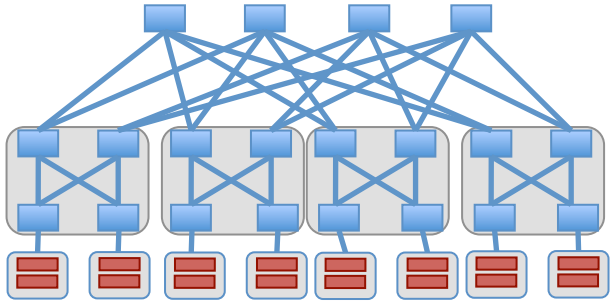
- NoviFlow has developed switches with 1 million entries [1].

[1] NoviFlow. 1248 Datasheet. http://bit.ly/1baQd0A.

Does PARIS Ensure Good Performance?

- How low is latency?
  - Recall: random traffic matrix.

Communication Pattern	Latency
Inter-pod	61us
Intra-pod	106us



Summary

- PARIS achieves scalability and flexibility
  - Flat layer 3 network
  - Pre-positioning forwarding state in switches
  - Using topological knowledge to partition forwarding state
- Our evaluations show that PARIS is practical!
  - Scales to large data-centers
  - Can be implemented using existing commodity devices

Questions