One Primitive for all, all for one: Enabling Dynamic Datacenter Load Balancing

Antonio Marsico[1], Gianni Antichi[2], Theophilus Benson[3], Marco Savi[1], Andrew W. Moore[2]
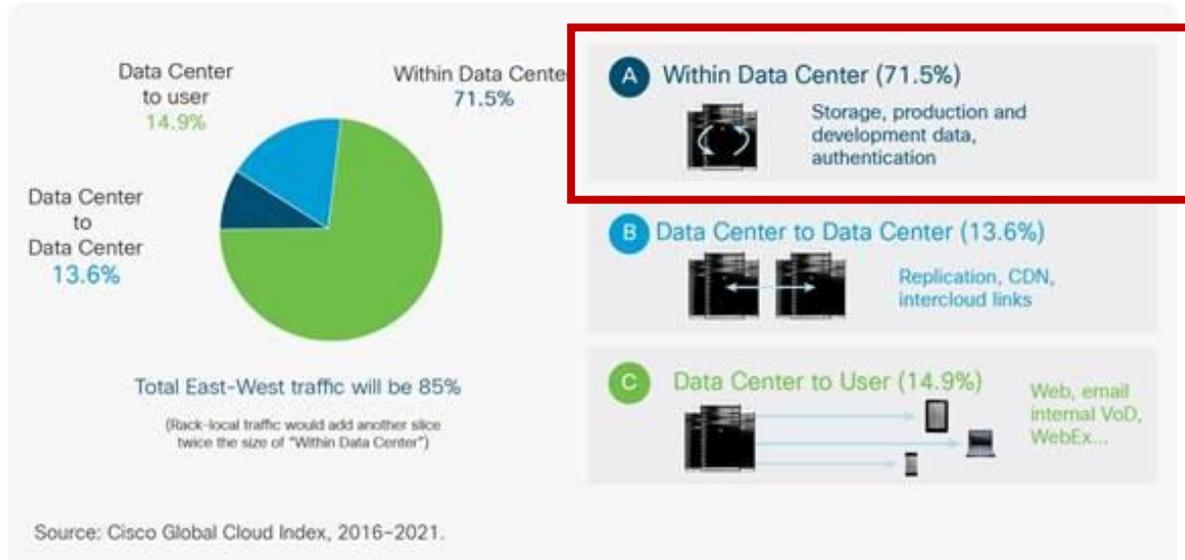
[1]FBK CREATE-NET, [2]University of Cambridge, [3]Brown University

Coseners 2018

July 6th 2018

Introduction: Today's Data Center traffic

- A ***growing*** number of ***applications*** generating a ***huge amount of traffic***

Data Center to user 14.9%

Within Data Center 71.5%

Data Center to Data Center 13.6%

A Within Data Center (71.5%)
Storage, production and development data, authentication

B Data Center to Data Center (13.6%)
Replication, CDN, intercloud links

C Data Center to User (14.9%)
Web, email internal VoD, WebEx...

Total East-West traffic will be 85%
(Rack-local traffic would add another slice twice the size of "Within Data Center")

Source: Cisco Global Cloud Index, 2016–2021.

Intra-DC Load Balancing

Objective: *Increasing the utilization* of all the paths with equal costs between a SRC and a DST
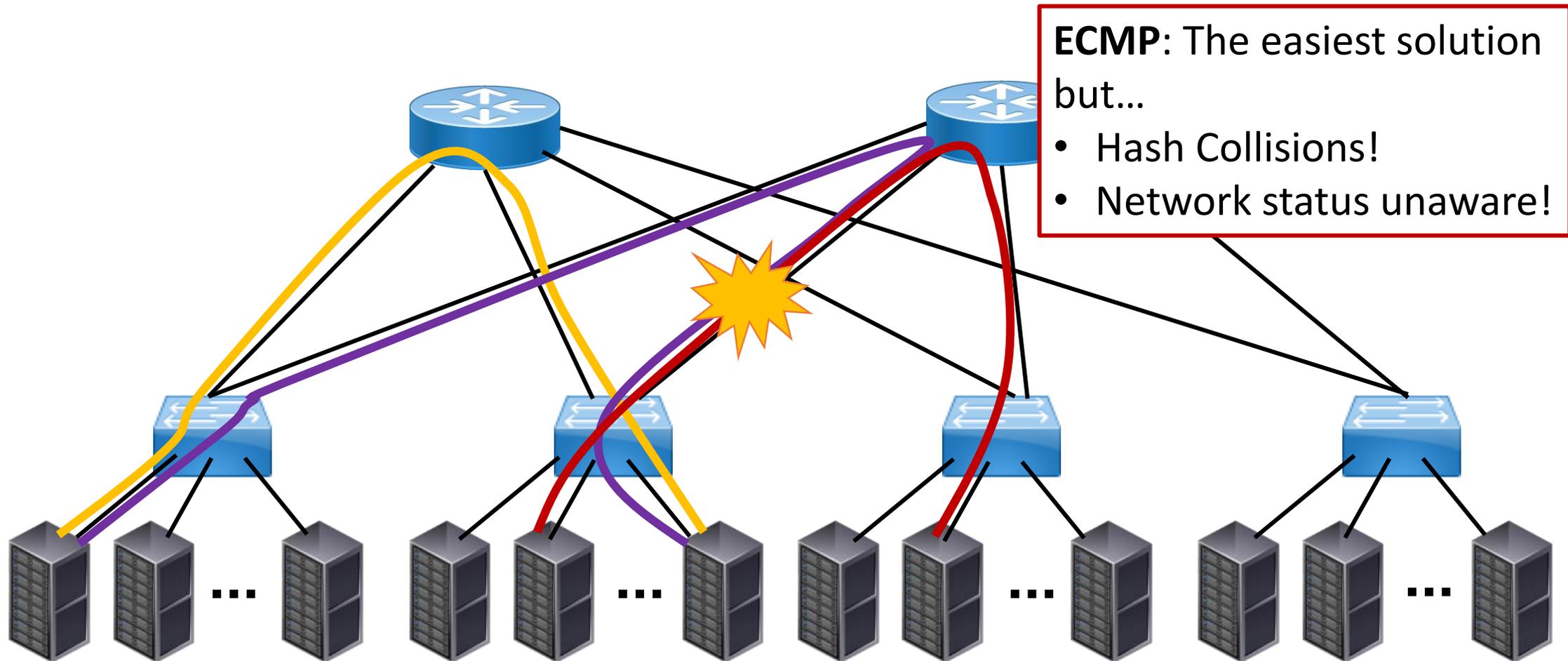
ECMP: The easiest solution but...
- Hash Collisions!
- Network status unaware!

Overcoming ECMP drawbacks

Load Balancing Solution	Traffic Granularity	Measurement Primitives	Designed for
Let it Flow	Flowlet	None	Asymmetric topologies
Hedera/MicroTE	Flow	Heavy hitters	Optimizing elephant flows
DRILL	Packet	Queue occupancy	High network load (> 80%)
Local Flow	Flow (selective splitting)	Link congestion	Symmetric topologies
Conga	Flowlet	Link congestion	Sym/Asym topologies
Hula	Flowlet	Link congestion	Sym/Asym topologies

Overcoming ECMP drawbacks

Load Balancing Solution	Traffic Granularity	Measurement Primitives	Designed for
Let it Flow	Flowlet	None	Asymmetric topologies
He		Heavy hitters	Optimizing elephant flows
DF		Queue occupancy	High network load (> 80%)
Lo	ng)	Link congestion	Symmetric topologies
Conga	Flowlet	Link congestion	Sym/Asym topologies
Hula	Flowlet	Link congestion	Sym/Asym topologies

> **Takeaway #1:**
> There *is no one measurement* primitive that is *superior* to the others

Example: HULA[1]

- By design, **HULA** exploits *link congestion* information to make routing decisions
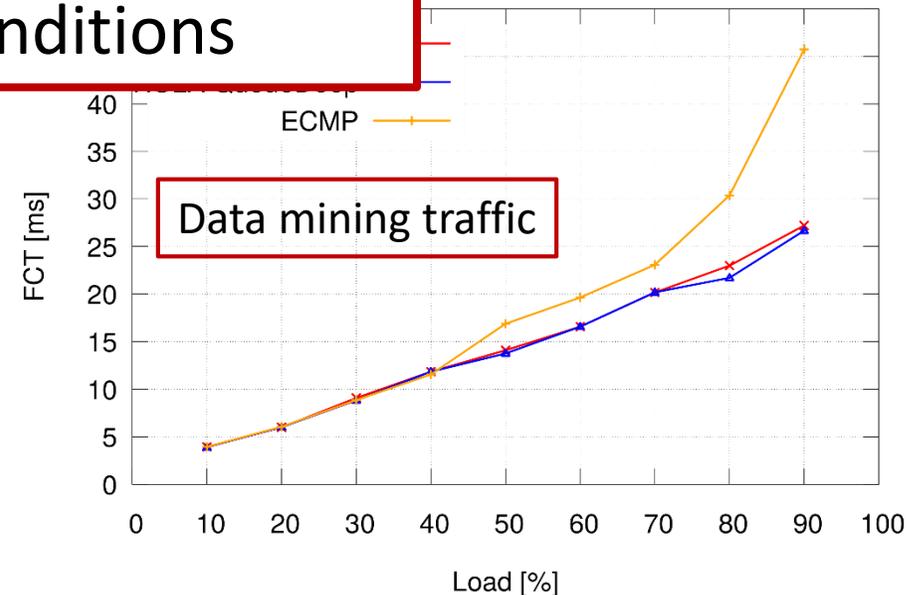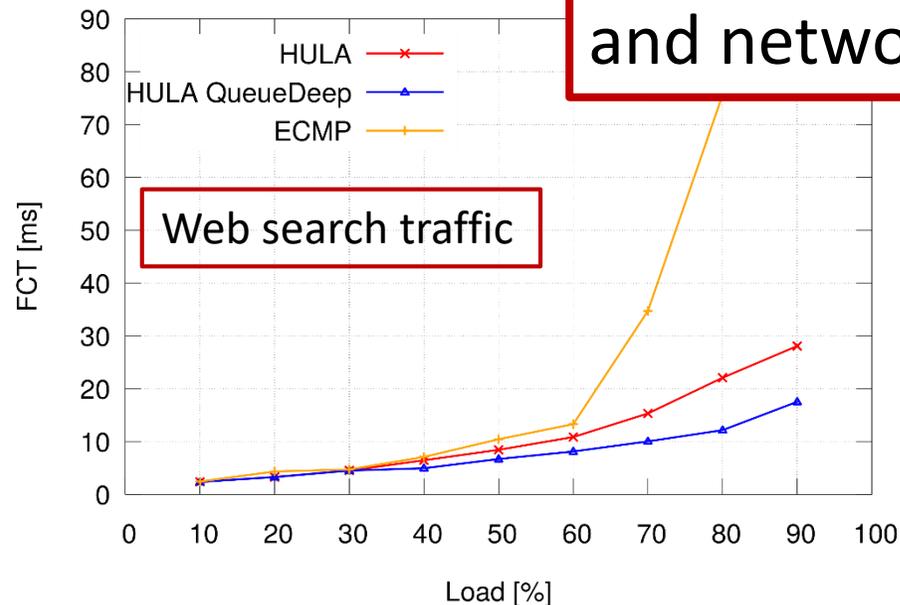
- What happens when we modify the measurement primitive?

Web search traffic

Data mining traffic
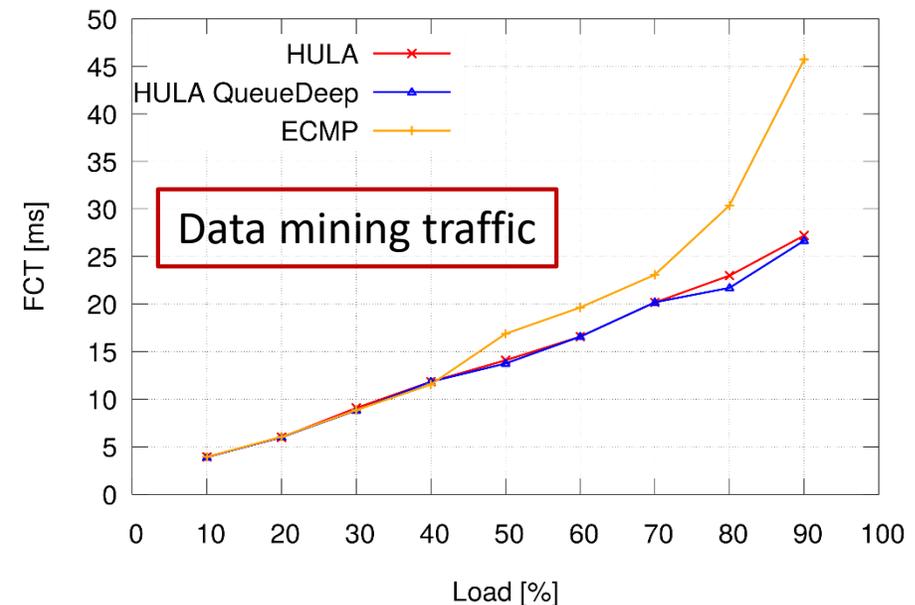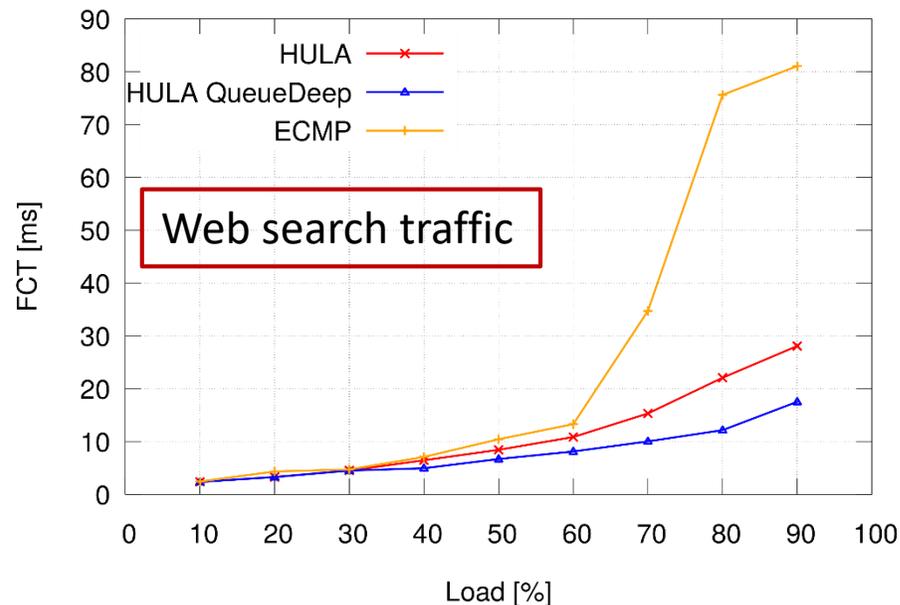
[1] N. Katta et al., "HULA: Scalable Load Balancing Using Programmable Data Planes", SORS 2016

Example: HULA[1]

- By design, **HULA** e̶̶ tion to make routing decisions

- What happens wh̶̶̶ nt primitive?

Takeaway #2:
The *choice* of measurement primitive *is predicated on* the application, topology, and network conditions

Web search traffic

Data mining traffic

[1] N. Katta et al., "HULA: Scalable Load Balancing Using Programmable Data Planes", SORS 2016

One measurement does not fits all scenarios

- **Data planes** of data center devices must provide a **flexible** and **extensible** substrate to **support** various type of measurements
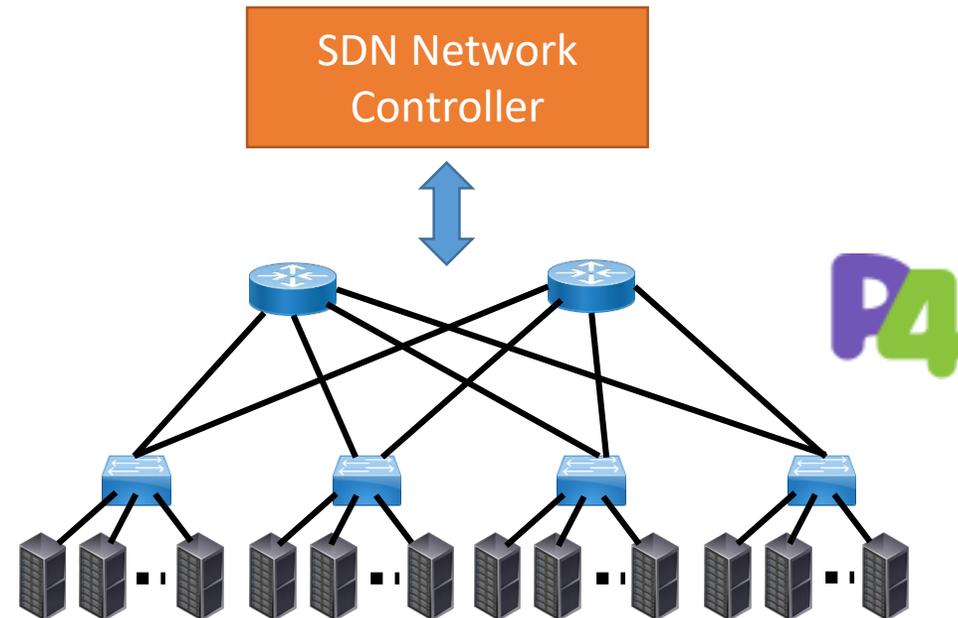
A possible solution – Interaction between Applications and Networks

- ***Applications/Hosts*** and ***Networks*** could ***cooperate together*** to ***improve*** the network performance
- Exploiting the best measurement primitive based on the application traffic
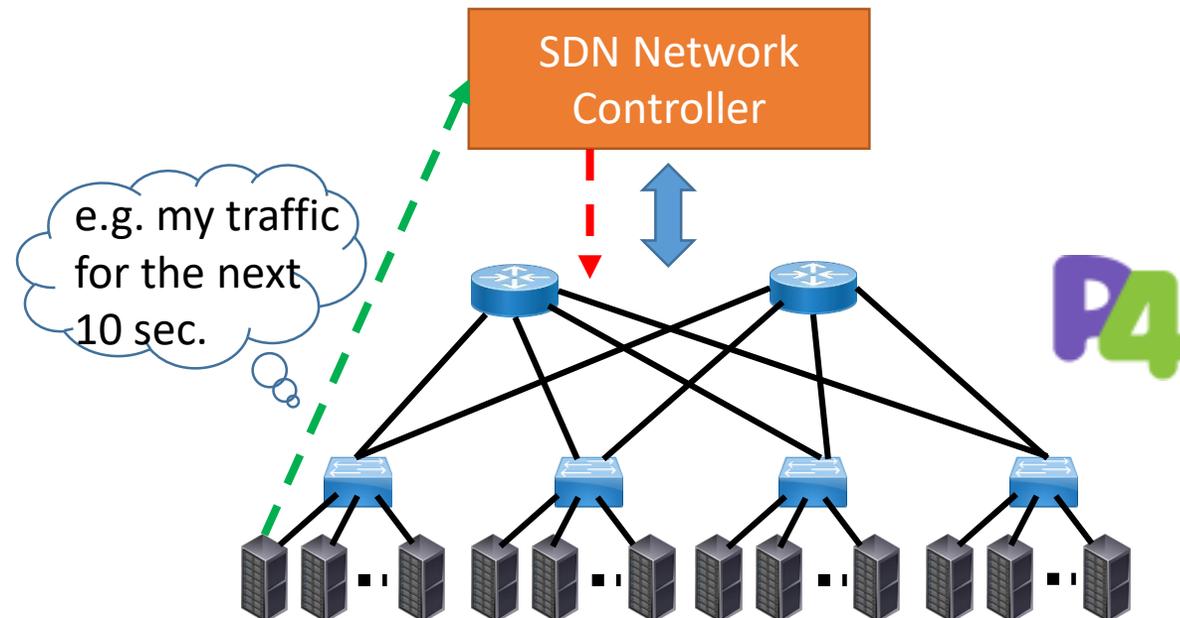
Food for thoughts

- **Challenges** for a **real implementation** (some examples):
    - P4 itself cannot generate probes to sense the network (e.g. HULA)
    - Communication between applications/hosts and controller
    - Modifying the data plane configuration
    - Handling a partial deployment scenario: not all the data center devices can be substitute at the same time!

Thank you for your kind attention!

amarsico@fbk.eu

@toto_120