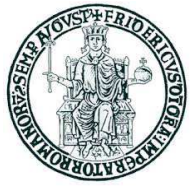
Cloud and Datacenter Networking

Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione DIETI
Laurea Magistrale in Ingegneria Informatica

Prof. Roberto Canonico

Datacenter networking infrastructure

Part III



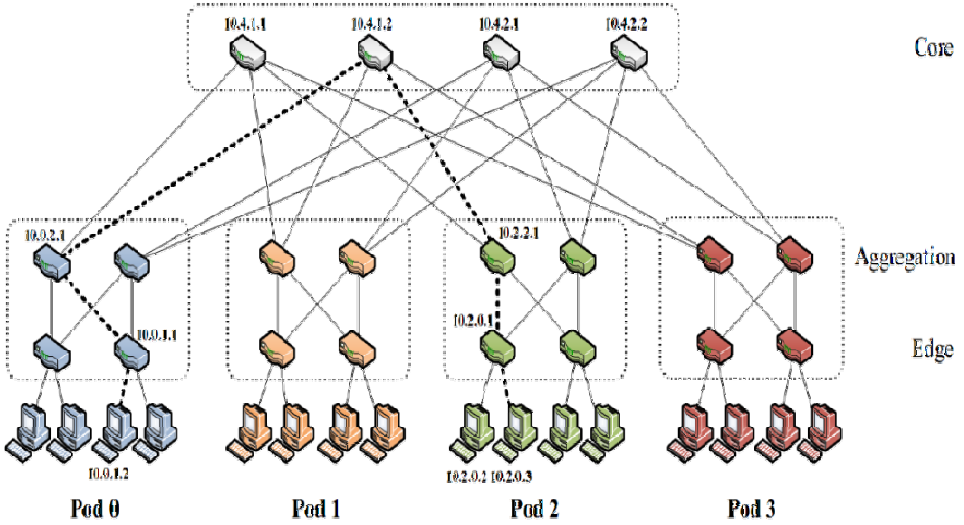
Lesson outline

- Datacenter network topology with commodity switches: Fat-Tree
  - Paper: Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat.
    A scalable, commodity data center network architecture.
    SIGCOMM Computer Communications Review, vol. 38, issue 4, pp. 63-74, August 2008
- Datacenter networks and alternate paths exploitation
  - TRILL protocol
    - Paper: Radia Perlman and Donald Eastlake. *Introduction to TRILL*.
      The Internet Protocol Journal, Volume 14, No. 3, pp. 2-20, September 2011
  - ECMP protocol

Fat-tree: a scalable commodity DC network architecture

- Topology derived from Clos networks
  - Network of small cheap switches
- 3-layers hierarchy
- $k^3/4$ hosts grouped in
  - k *pods* with $(k/2)^2$ hosts each
- Peculiar characteristics:
  - The number of links ($k/2$) from each switch to an upper layer switch equates the number of links ($k/2$) towards lower-layer switches
    → No oversubscription (1:1)
  - k-port switches at all layers

3-layers fat-tree network k=4

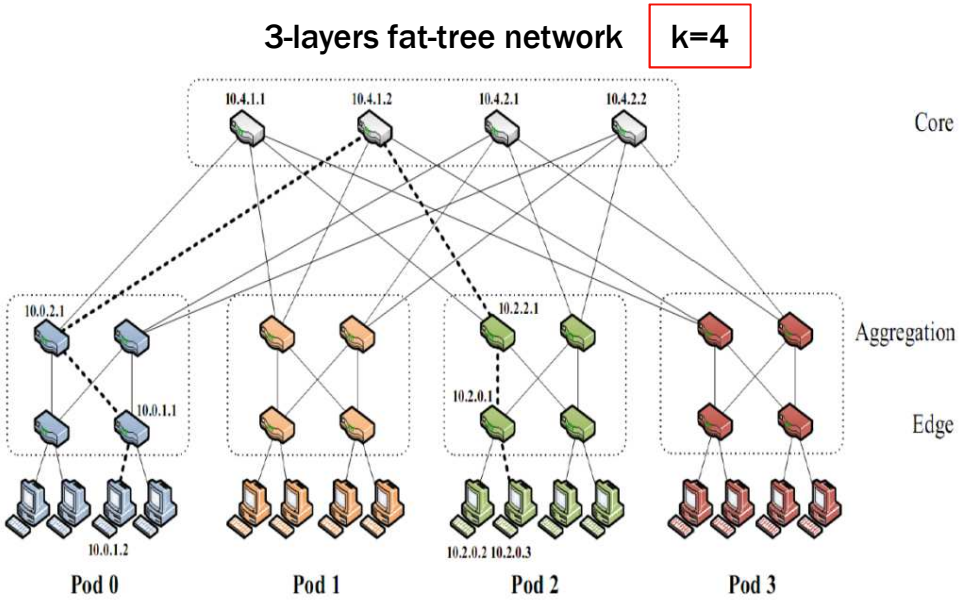


- Each edge switch connects k/2 hosts to k/2 aggregation switches
- Each aggregation switch connects k/2 edge switches to k/2 core switches
- $(k/2)^2$ core switches
- Resulting property: each layer of the hierarchy has the same aggregate bandwidth

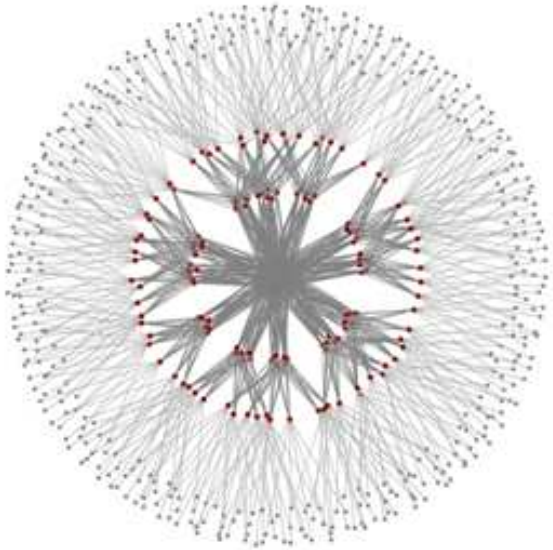
Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat.
A scalable, commodity data center network architecture.
SIGCOMM Computer Communications Review, 38, 4 (August 2008), pp. 63-74

Fat-tree (continues)

- $k^3/4$ hosts grouped in
  - k *pods* with $(k/2)^2$ hosts each
- Each edge switch connects k/2 hosts to k/2 aggregation switches
- Each aggregation switch connects k/2 edge switches to k/2 core switches
- $(5/4)\,k^2$ switches, of which $(k/2)^2$ core switches
- To obtain the required high capacity in the core layer a large number of links are aggregated



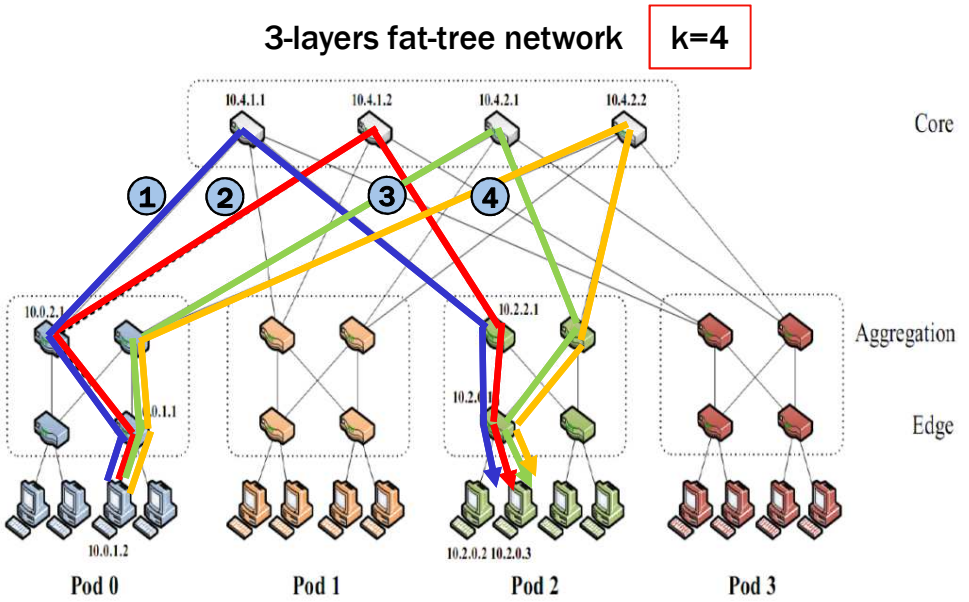
k	# hosts ($k^3/4$)	# core switches $(k/2)^2$	# switches $(5/4)\,k^2$
4	16	4	20
12	432	36	180
16	1.024	64	320
24	3.456	144	720
32	8.192	256	1.280
48	27.648	576	2.880
96	221.184	2.304	11.520

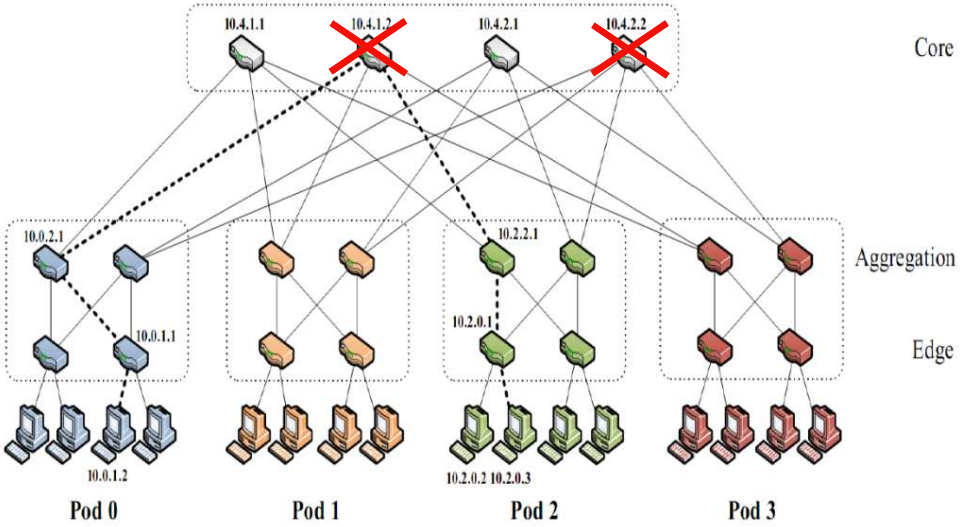


3-layers fat-tree
432 servers, 180 switches, k=12

Fat-tree (continues)

- Fat-tree network: redundancy
  - k different paths exist between any pair of hosts
  - Only one path exists between a given core switch and any possible host
  - Question:
    how is it possible to exploit the alternate paths ?
- The network topology in the picture has:
  - 16 access links (server-switch)
  - 16 edge-aggregation link
  - 16 aggregation-core links
- No bottlelnecks in the upper layers
- A limited amount of oversubscription may be introduced
  (for instance, by using only 2 core switches)





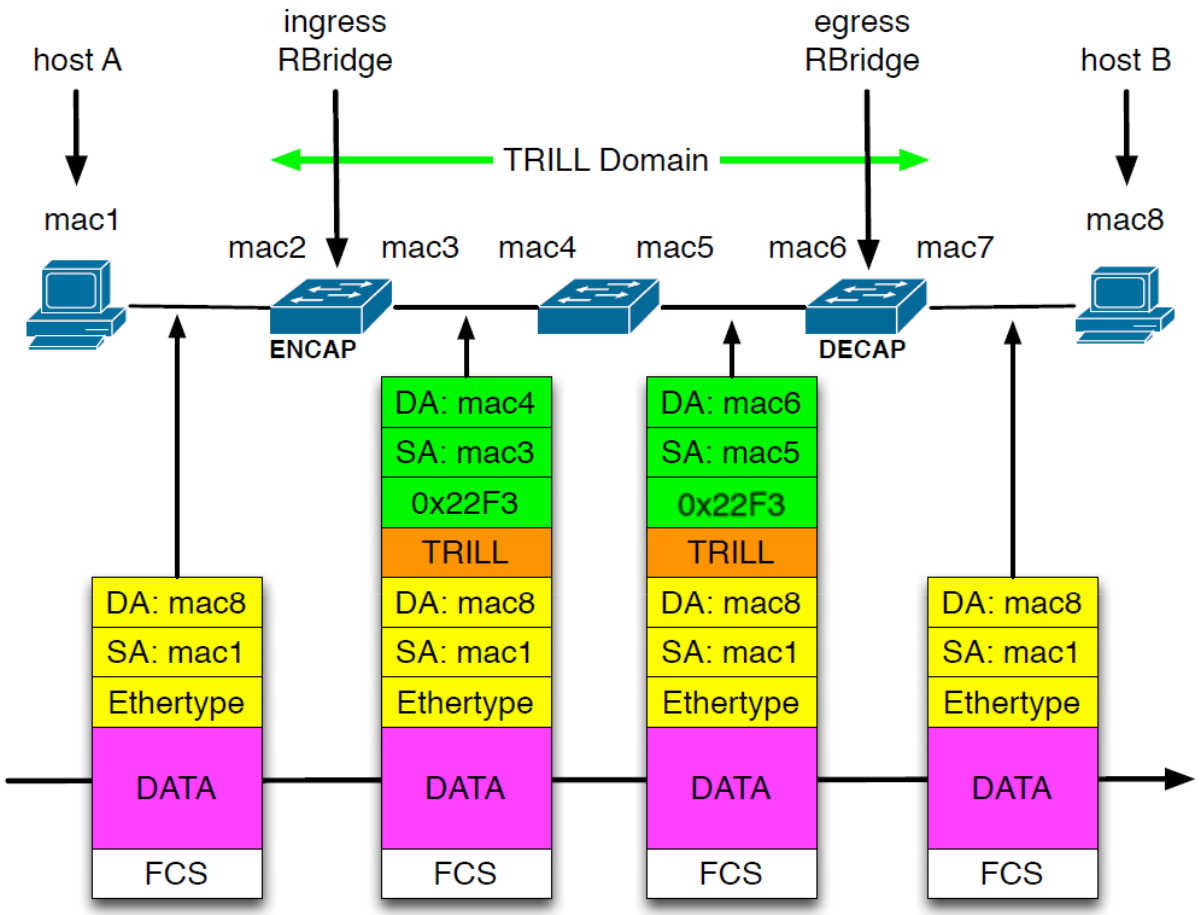
Datacenter networks and alternate paths exploitation

- We have seen that DC networks are not single-root trees → alternate paths exist between any couple of hosts located in different racks
- If STP is used, the network topology is transformed into a loop-free tree by selectively disabling links → only a fraction of the network capacity is used
- To exploit alternate paths more complex solutions are needed
  - Routing is a typical L3 feature
- TRILL: IETF standard born for complex campus networks
  - A layer-2.5 solution that can be incrementally deployed
  - TRILL combines techniques from bridging and routing
- Cisco FabricPath: proprietary implementation of TRILL
- Brocade Virtual Cluster Switching: uses TRILL data plane but a proprietary control plane
- ECMP

TRILL: Transparent Interconnection of Lots of Links

- TRILL relies on special switches called R-Bridges
- R-Bridges run a link-state protocol:
  - learn the network topology through the exchange of *Link State Packets* (LSPs)
  - compute shortest path tree between them
- The link-state protocol used by TRILL is IS-IS
  - IS-IS was originally defined as an ISO/OSI standard (ISO/IEC 10589:2002) and later described in IETF RFC1142
  - IS-IS chosen because it runs directly over Layer 2, so it can be run without configuration
    - no IP addresses need to be assigned
- TRILL switches are identified by 6-byte IS-IS System ID and by 2-bytes nicknames
- TRILL is compatible with existing IP Routers: R-Bridges are transparent to IP routers
- R-Bridges encapsulate each packet they receive from hosts with a header bringing the ID of the next-hop R-Bridge in the shortest path to the destination
  - the R-bridge which is closest to the destination decapsulates the packet before delivering it to the destination
- TRILL data packets between R-Bridges have a **Local Link header** and a **TRILL header**
- For unicast packets:
  - Local Link header contains the addresses of the local source R-Bridge to the next hop R-Bridge
  - TRILL header specifies the first/ingress R-Bridge and the last/egress R-Bridge
- A 6-bits hop count is decreased at each R-Bridge

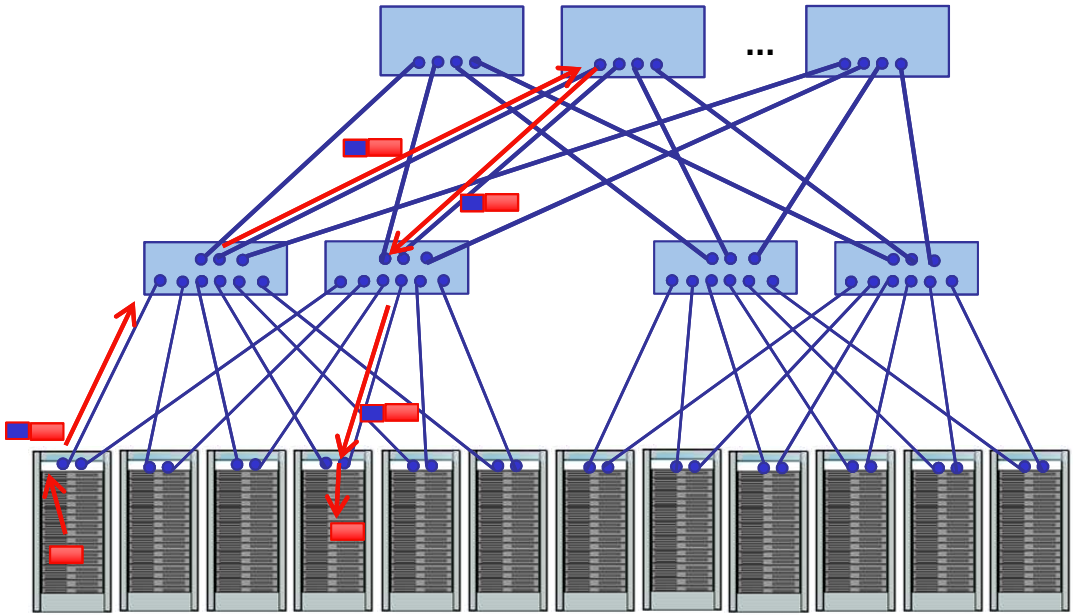
TRILL packet forwarding



- Figure by Ronald van der Pol, from “TRILL and IEEE 802.1aq Overview” (Apr.2012)

TRILL in the datacenter

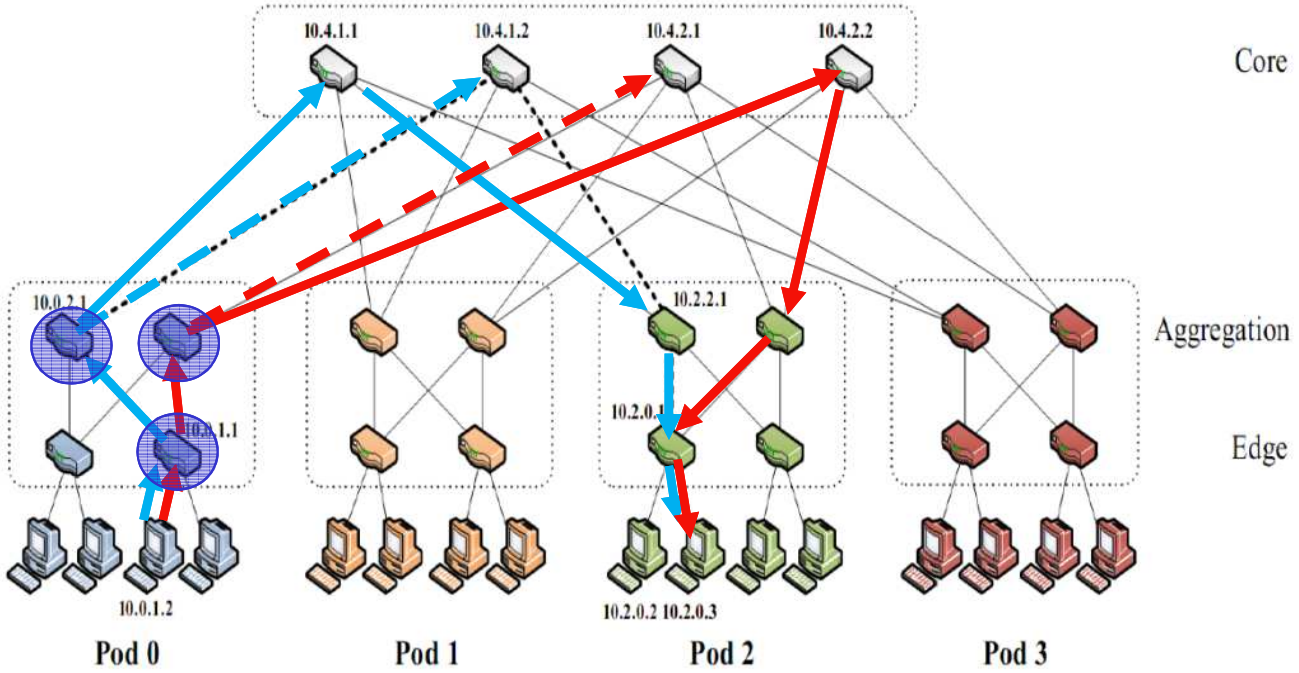
- TRILL implemented in the switches at all layers
- Hosts speak “pure IEEE 803.3” (Ethernet)
- In order to learn end-hosts’ identity (i.e. association to edge R-Bridge), a directory service is needed



- Leaf switches encapsulate each packet they receive from hosts with a header bringing the ID of the next-hop R-Bridge in the shortest path to the destination
- The R-bridge which is closest to the destination decapsulates the packet before delivering it to the destination
- If multiple equal costs paths are presents towards a destination, an RBridge can distribute traffic over those multiple paths
  - For multi-path routing, ECMP is typically used
  - See ECMP in the next slides

Multi-path routing: ECMP

- In a datacenter with multiple possible paths between any (source, destination) couple ECMP allows to randomly spread traffic over alternative paths



At the first edge switch, traffic from 10.0.1.2 to 10.2.0.3 is randomly routed either on the left path or on the right path
Also aggregation switches may randomly choose one among two different paths

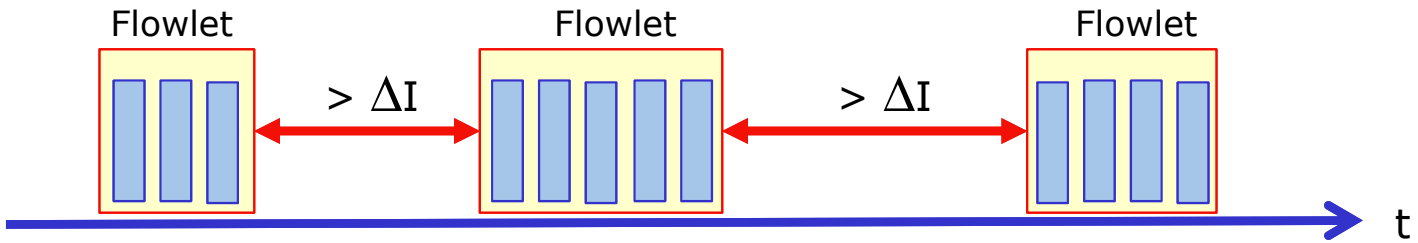
- If upper layer switches have internal queues filled up differently, packets may arrive mis-ordered to destination → TCP performance degrades
- To avoid this problem, packets of the same flow need to be routed on the same path

ECMP and flow hashing

- To avoid misordered delivery of packets belonging to the same flow, ECMP calculates the hash of the packet header to determine the output port at each switch
- In this manner, packets of the same flow, i.e. with same (source, destination), follow the same path and are not misordered
- Works well for a large number of small flows $\rightarrow$ traffic is evenly distributed
- If multiple long-lasting flows are mapped onto the same ports, this technique may lead to an unbalance of traffic flows
- This problem arises because, actually, the concept of flow above is too coarse
- To avoid this problem and achieve a more fine-grained balancing of traffic, randomization may occur at micro-flow or *flowlet* level

ECMP and flowlets

- A *flowlet* is a sequence of consecutive packets whose inter-arrival is smaller than the conservative estimate of latency difference between any two paths within the datacenter network
  - If two flowlets are routed along different paths, no misordered delivery may happen anyway



- Flowlet-based routing first proposed in FLARE in 2007
- Flowlet-to-path mapping is performed by using a hash table whose entries are

  (hash_key, last_seen_time, path_id)

- When a packet arrives, FLARE computes a hash of

  source IP, destination IP, source port, destination port

- and uses this as the key in the hash table

[FLARE] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. *Dynamic load balancing without packet reordering*. ACM *SIGCOMM Comput. Commun. Rev.* 37, 2, pp. 51-62, March 2007

ECMP issues: local decisions

- One issue with ECMP is that it only takes local decisions without any knowledge of further links status
- In this example topology, once the path has been pinned to the core switch, there's no further alternative to a given destination (i.e. only one path)
- If a link fails, ECMP can do nothing to prevent upstream switches to select the path that contains that link, even if an alternative path exists