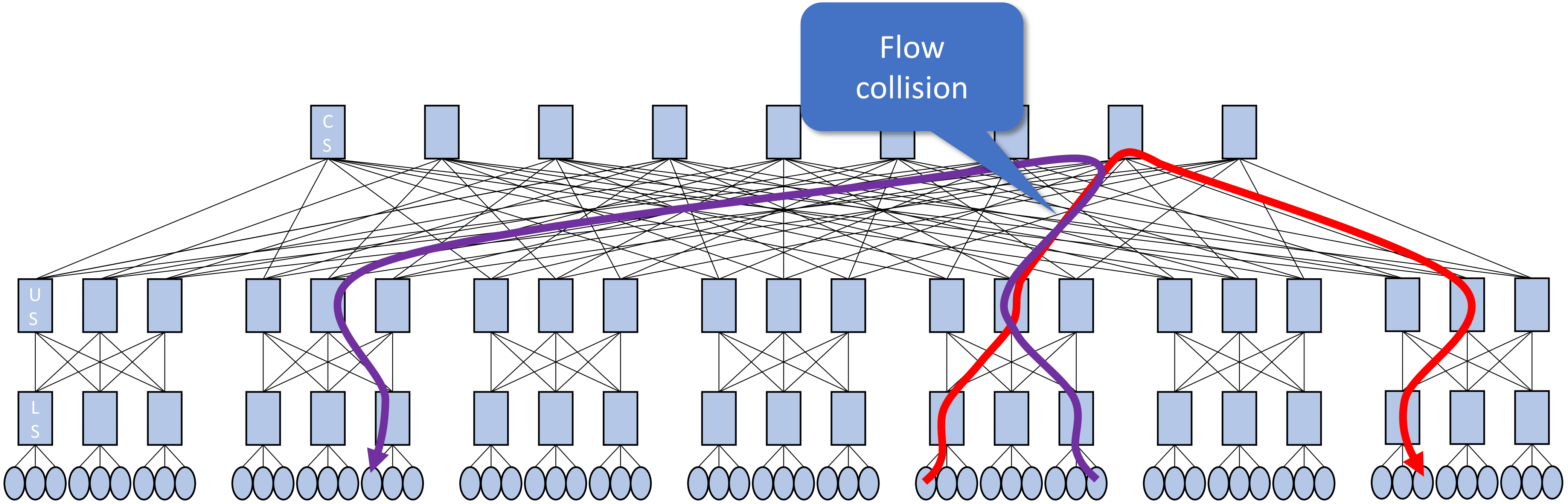
Load balancing strategies in AIML networks

Costin Raiciu

Broadcom and Politehnica of Bucharest

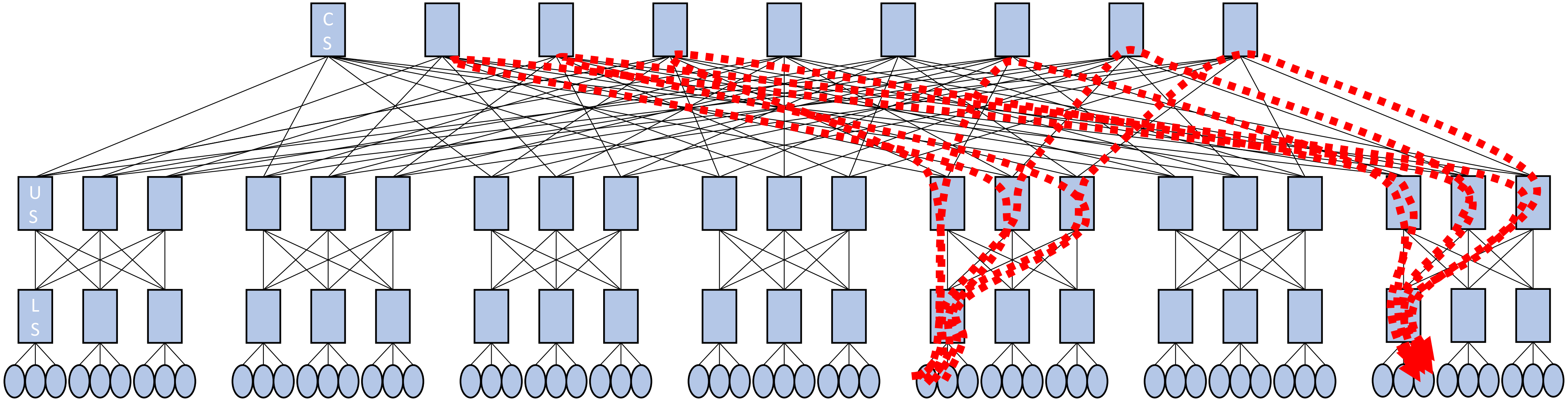
Flow collisions



Why not use MPTCP for AIML networks?

- Need to use many paths for the common case of
  - Symmetric highly loaded networks + short flows.
  - Best way to load balance short flows is to use many paths.
  - Load balancing works well for long flows, not so well for shorter flows.
- But with many paths, minimum MPTCP total window is #paths.
  - E.g. 256 paths means min 256 packet window. This equals BDP at 800Gbps.
  - Congestion collapse in incast.
- Path state for MPTCP is quite costly.
  - CWND, flight_size, sequence numbers, etc. (tens of bytes).
  - 256 * 20 = 5KB per connection!

Packet spraying in AIML networks

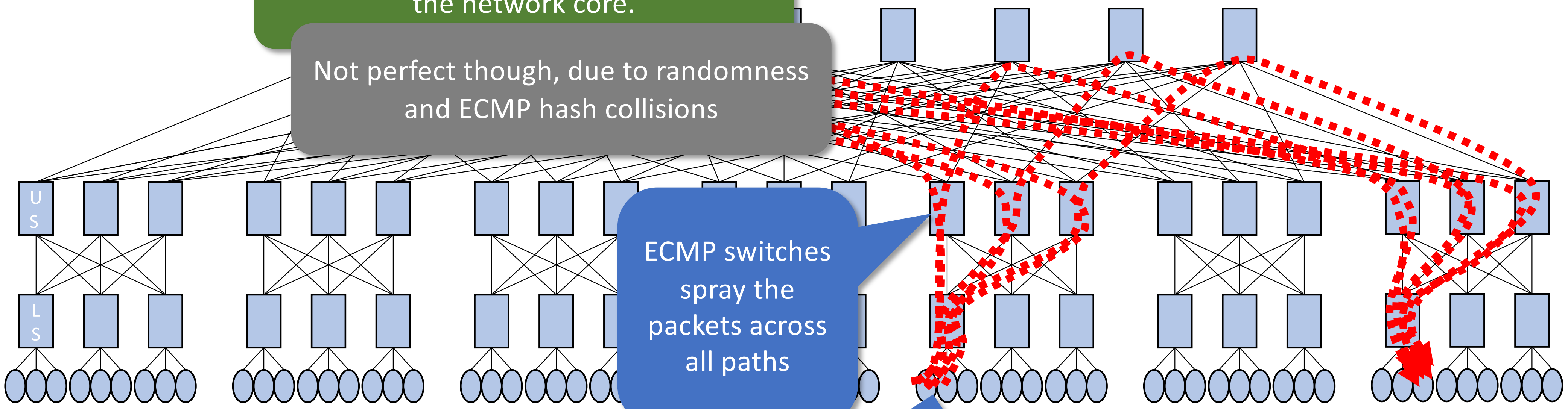


Congestion control with packet spraying

- Maintain a single congestion window that upper bounds flight size.
- Sender-driven congestion control (e.g. UET NSCC)
  - Targets sub-BDP standing queue at the bottleneck.
  - Use ECN and delay simultaneously.
    - Aggressive increase when queue ~ 0. Linear increase otherwise.
    - Multiplicative decrease when ECN mark & average delay above threshold.
  - Average delay across all paths.

- But how to load balance packets across paths?
  - Bad load balancing results in reducing CWND (across all paths).

End-to-end packet spraying via ECMP



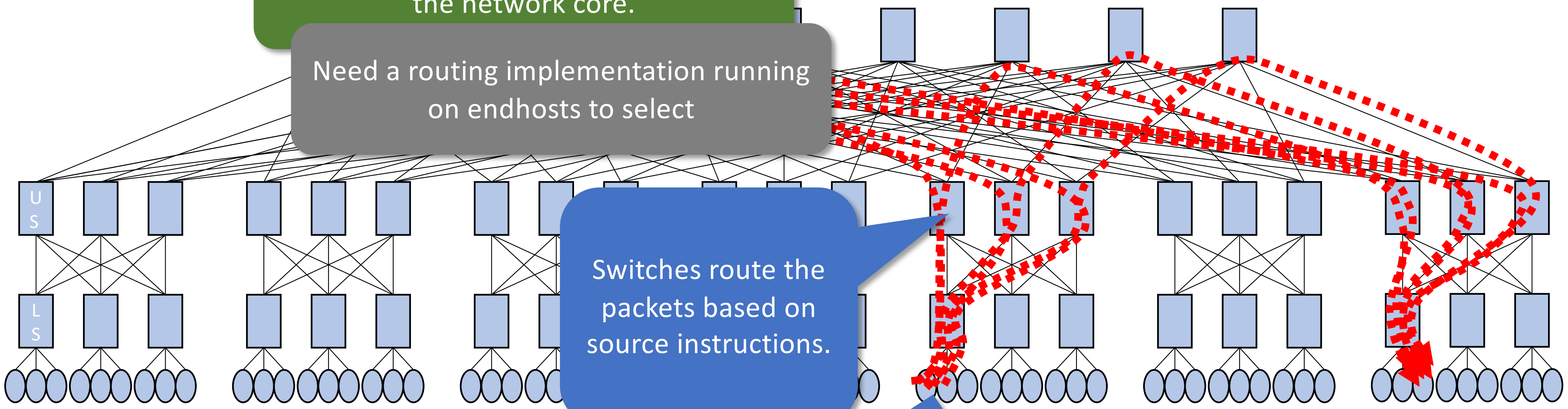
Good load balancing across all paths in the network core.

Not perfect though, due to randomness and ECMP hash collisions

ECMP switches spray the packets across all paths

Change entropy value (e.g. UDP src port) with **every** packet sent

End-to-end packet spraying with source routing.



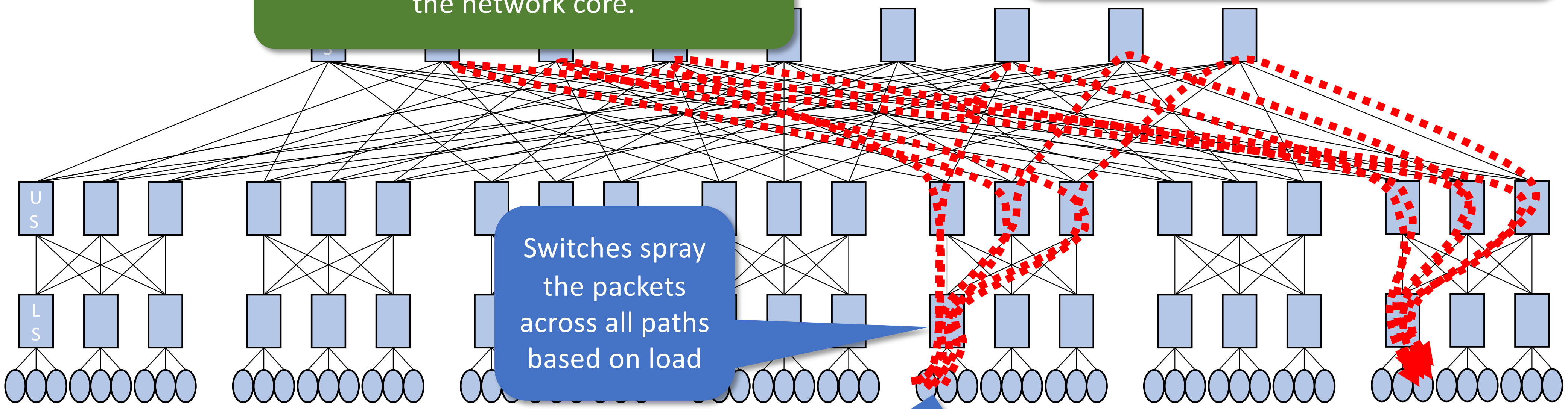
Good load balancing across all paths in the network core.

Need a routing implementation running on endhosts to select

Switches route the packets based on source instructions.

Change path for **every** packet sent

Switch packet spraying



Good load balancing across all paths in the network core.

Not perfect though, due to no choice on the way down

Switches spray the packets across all paths based on load

Host has no control over multi-pathing.

Two basic approaches for spraying

- Host-based spraying –
    - ECMP + standard routing protocol (e.g. BGP).
    - Source routing – requires an SDN routing protocol to compute paths and deliver them to hosts.

- Switch spraying – adaptive routing / dynamic load balancing.

What are the pros and cons of each?

Host based spraying
What is the best way to spray packets?

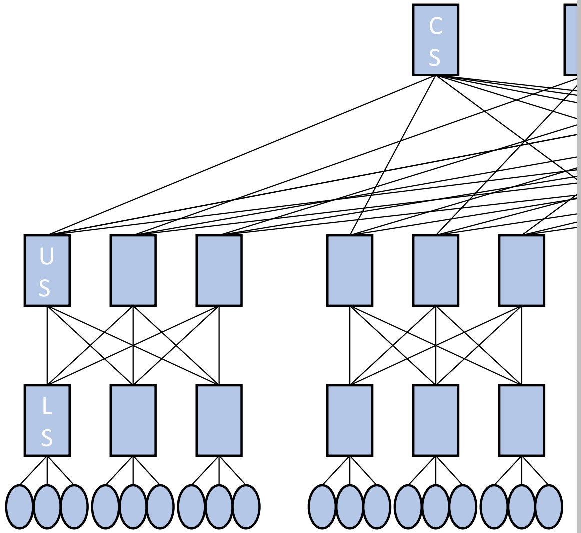
- **Simplest: oblivious load balancing**
  - Pick a random EV for each packet.
  - Works very well if network capacity is uniform.

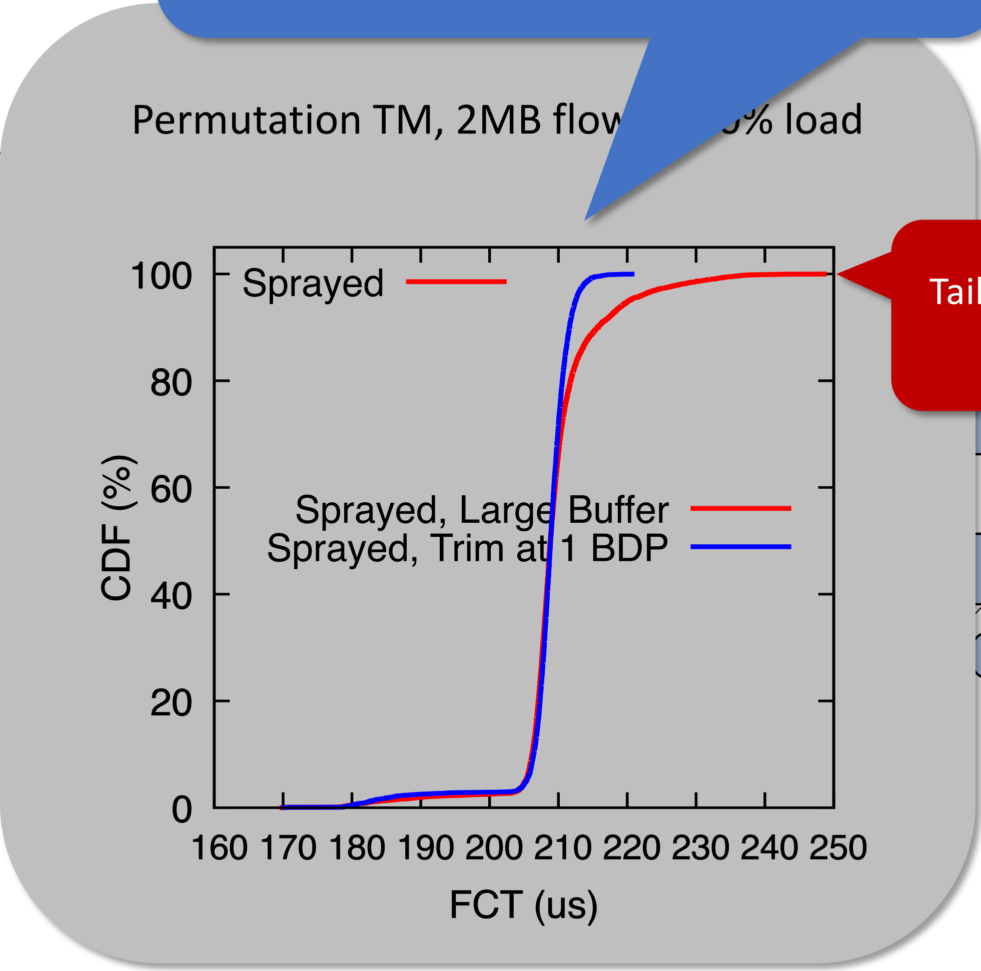
- Bitmap load balancing (e.g. UET bitmap algorithm)
  - Per EV state - one or a few bits.
  - When ACK indicates ECN mark, increment EV state.
  - When EV is next to be picked but non-zero state, decrement state, skip.

- Recycled entropies (REPS):
  - Keep EV cache for which we got an ACK without ECN set.
  - Path selection: pick EV from cache if non-empty. Otherwise pick random EV.

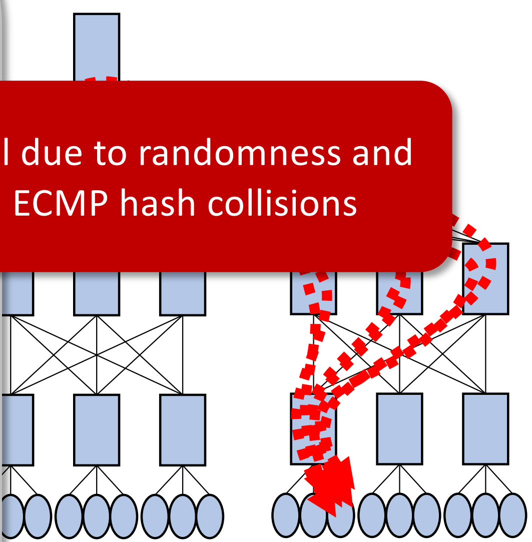
Oblivious packe

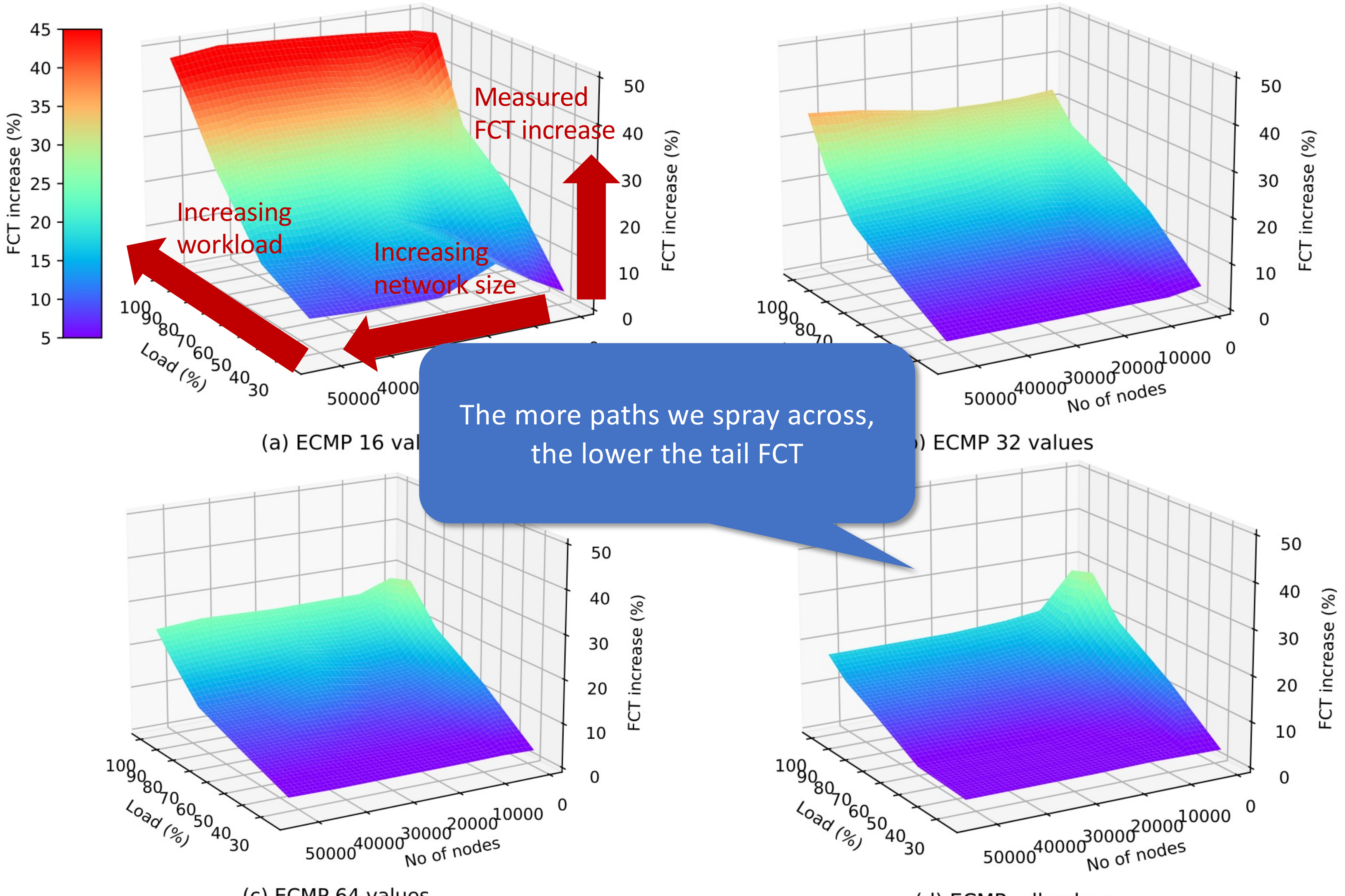
When sprayed load balancing is imperfect, queues can still build.
Trimming prevents queue building.
Packet gets trimmed, NACKed,
RTX on a different less loaded path.





Tail due to randomness and ECMP hash collisions





Measured FCT increase

Increasing workload

Increasing network size

(a) ECMP 16 val

) ECMP 32 values

(c) ECMP 64 values

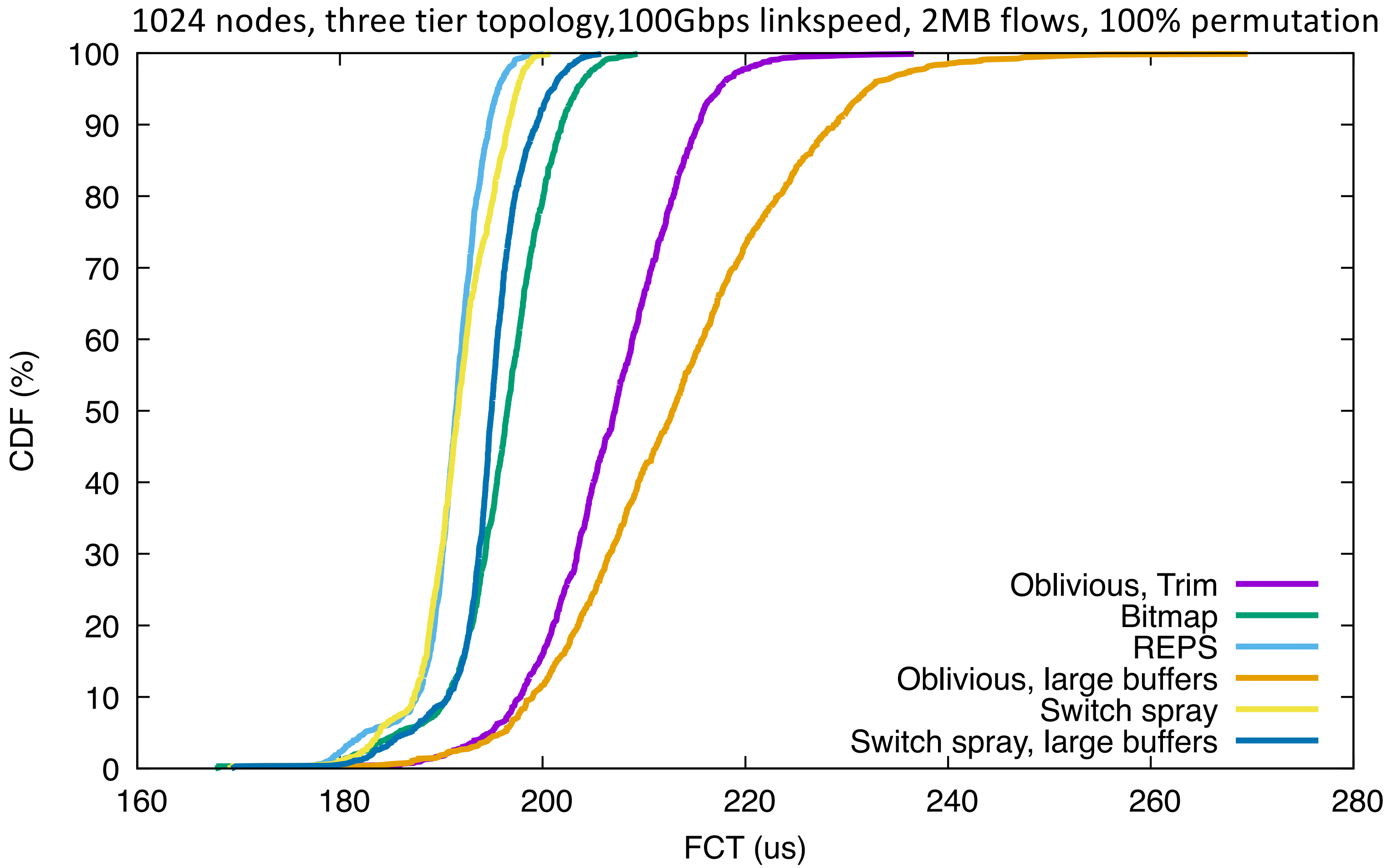
(d) ECMP, all values

The more paths we spray across, the lower the tail FCT

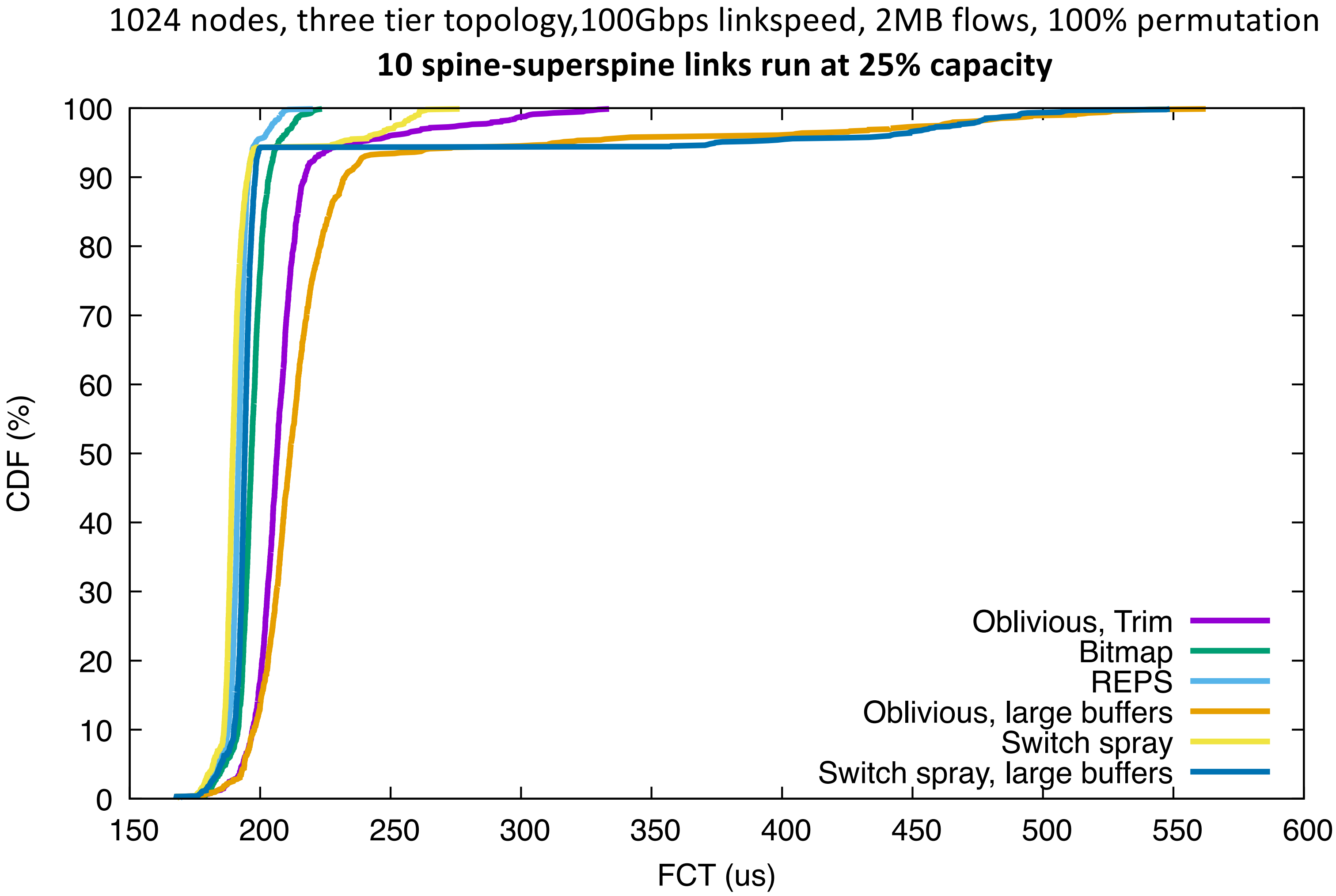
How does switch spray work?

- ECMP group => DLB group.
- During route lookup, routes in DLB group are consulted.
  - Contains all available paths towards destination.
  - Switch uses local information to decide which route (and associated egress port) to pick
  - Example metrics:
    - Queue length
    - Bandwidth utilization
    - PFC Port state.
    - Combination of the above possible.
- Works very well when path choice exists (e.g. going up the tree).
- Less well on the downward path / with asymmetries.
  - At the limit, behaves like oblivious endhost spraying.

Load balancing algorithms comparison



Load balancing with asymmetric capacity



Summary

- Packet spraying enables exploring many/all paths.
- Endhost or switch spraying possible.
- All schemes work almost perfect when network is perfectly symmetric.
    - Switch spraying
        - Lower average buffer utilization than endhost load balancing.
        - Works very well when path choice available (e.g. link bundles).
    - Endhost spraying:
        - State-based schemes can achieve similar FCT to switch spraying.
        - But lead to higher queue utilization.
        - Even oblivious works quite well.
- When network is asymmetric, poor load balancing leads to large FCT.
    - Switch spray on its own struggle – need additional mechanisms.
    - State-based endhost load balancing copes fairly well.