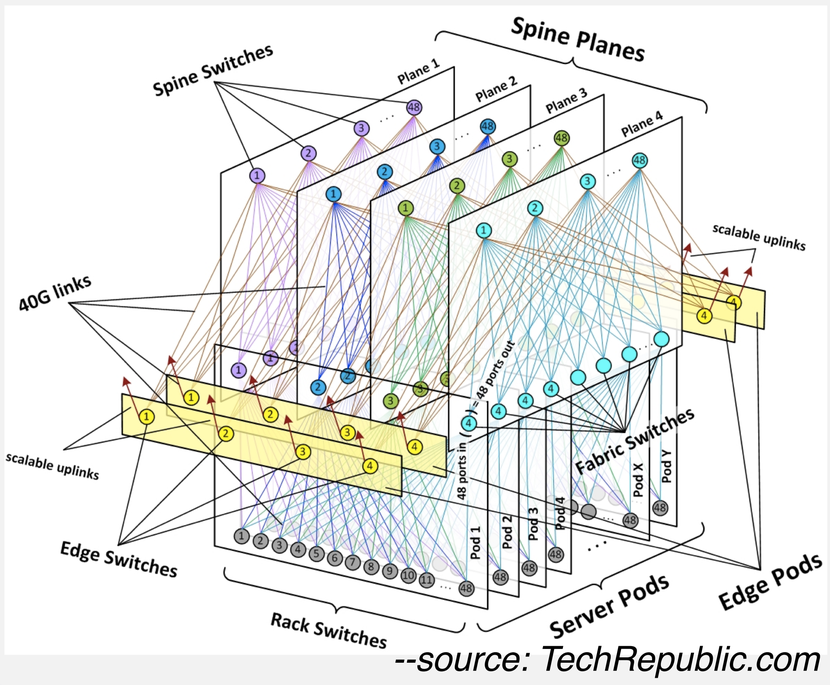
Simplifying Datacenter Network Debugging with PathDump

Praveen Tammana†, Rachit Agarwal‡, Myungjin Lee†
†School of Informatics - University of Edinburgh, ‡Computer Science - Cornell University

Open source available at: https://github.com/PathDump/



--source: TechRepublic.com

Datacenter networks are complex

- Increasingly larger scale
- Over 100,000 servers, over 10,000 switches
- Each server with 10 to 40 Gbps NIC
- High utilization: > 100 Tbps aggregate traffic

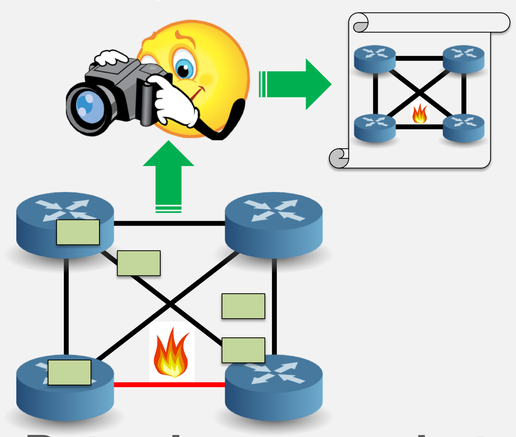
- Complexity due to the need for
  - ✓ High availability
  - ✓ High performance

Latency matters. Amazon found every 100ms of latency cost them 1% in sales. Google found an extra .5 seconds in search page generation time dropped traffic by 20%. A broker could lose $4 million in revenues per millisecond if their electronic trading platform is 5 milliseconds behind the competition.

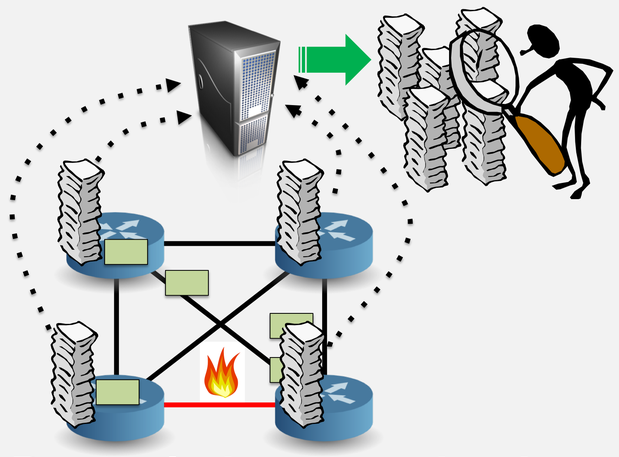
--source: The Gigaspaces blog

Network debuggers are even more complex

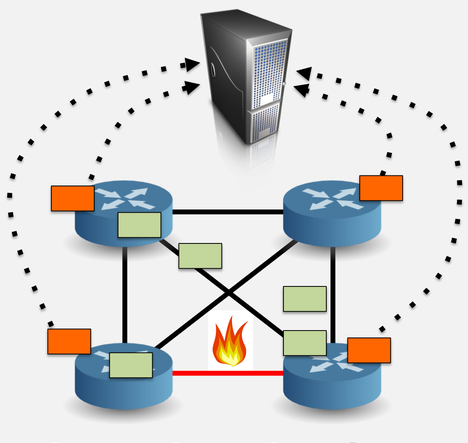
- Increasingly, network debugging functionality is pushed into the network due to improved switch programmability
- Existing tools employ complex in-network techniques such as data plane snapshot, per-switch per-packet logs, packet mirroring/sampling, and dynamic rule updates



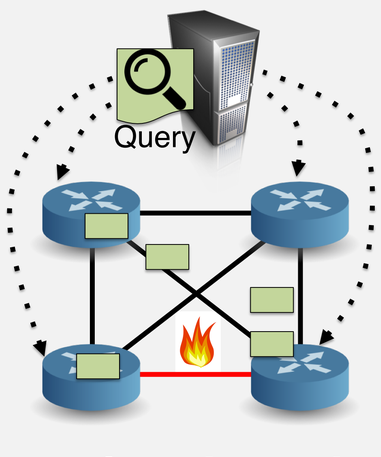
Data plane snapshot



Per-switch per-packet logs



Selective mirroring/sampling

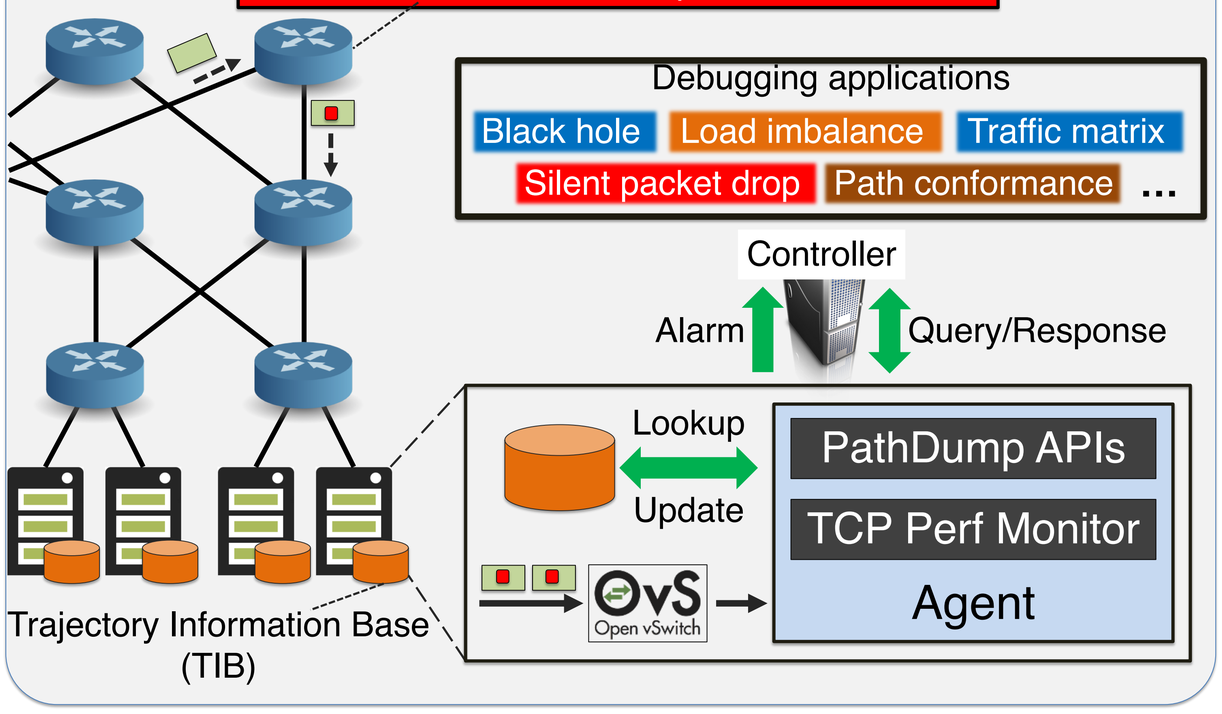


Dynamic rule updates

PathDump: A minimalistic network debugger

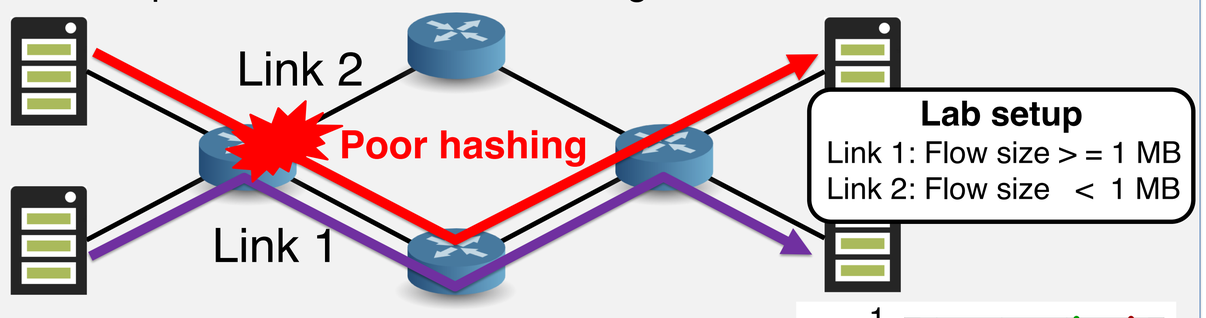
- Partition debugging functionality between switches and end-hosts
  - ✓ In-network tools focus on a smaller set of problems
  - ✓ Keeping networks and debugging tools as simple as possible

- Before forwarding a packet, check a condition
- If met, embeds its ID into packet header

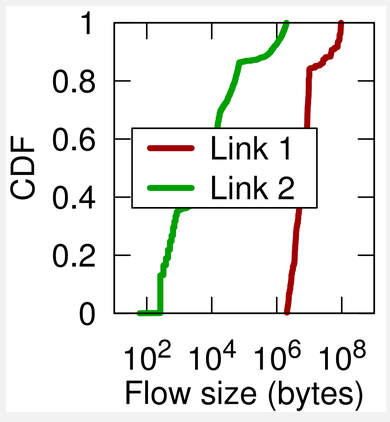


Example: Load imbalance diagnosis

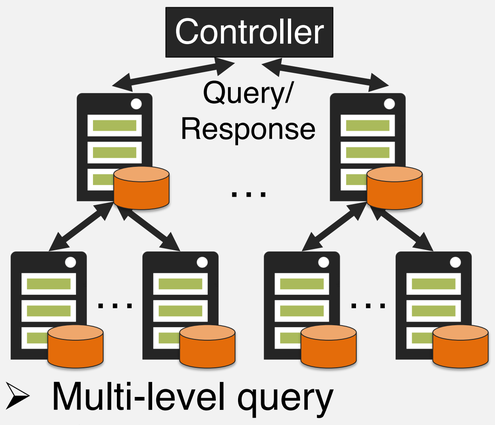
- Equal-Cost Multi-Path (ECMP) forwarding
  - ✓ Popular network load-balancing scheme in DCN



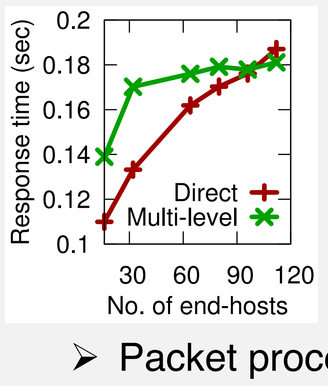

```
1: result = {}; binsize = 10000
2: linkIDs = (L1, L2); tRange = (t1, t2)
3: for lID in linkIDs:
4:     flows = getFlows (lID, tRange)
5:     for flow in flows:
6:         (bytes, pkts) = getCount (flow, tRange)
7:         result[lID][bytes/binsize] += 1
8: return result
```



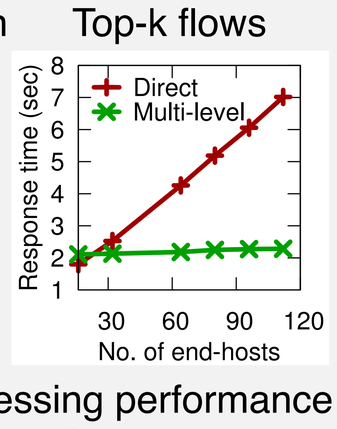
System performance evaluation



Flow size distribution



Top-k flows



- Multi-level query
  - ✓ To scale query performance
- Direct-query

- Packet processing performance
  - ✓ Supports **10 Gbps**
  - ✓ Thp loss < 4% to vanilla OVS

Conclusion and future work

- DCNs are complex; and their debuggers are even more complex
- Carefully partitions debugging functionality between network switches and edge devices

- Requires no complex operations from network switches
- Debugs a large class of network problems

- Future work: Real-time network debugging