
facebook
INFRASTRUCTURE

TTLd

Louis Plissonneau louisp@fb.com

Production Engineer (Network)

TTLd

Total TCP Loss detection (aka fancy acronym)

Pre-requisite:

- Completely own all your infrastructure

Own your datacenter

Own your racks

Own your hosts

Own your network

TTLd

Total TCP Loss detection (aka fancy acronym)

Goal:

- Surface End to End TCP retransmit throughout the network
- Use all production packets as probes

What if every packet was a probe?

- Precise **end to end performance** metric for TCP
- Probes are **following production traffic** (ECMP…)
- Measuring at any network device gives us E2E performance

Network Monitoring

Why “passive” monitoring is not enough?

- SNMP: Trusting network devices...

- Host TCP retransmit: Packet loss, everywhere...

Why “active” monitoring is not enough?

- **Injecting packets** in the network
  - **detect** service/customer impacting loss
  - **triangulate** loss to a device/interface

- Limitations
  - Potentially all injected packets could be dropped without production packets being lost or vice-versa
  - Number of probes is **many orders of magnitude** lower than number of production packets
    - It can miss low signal problems (bit flipping…)

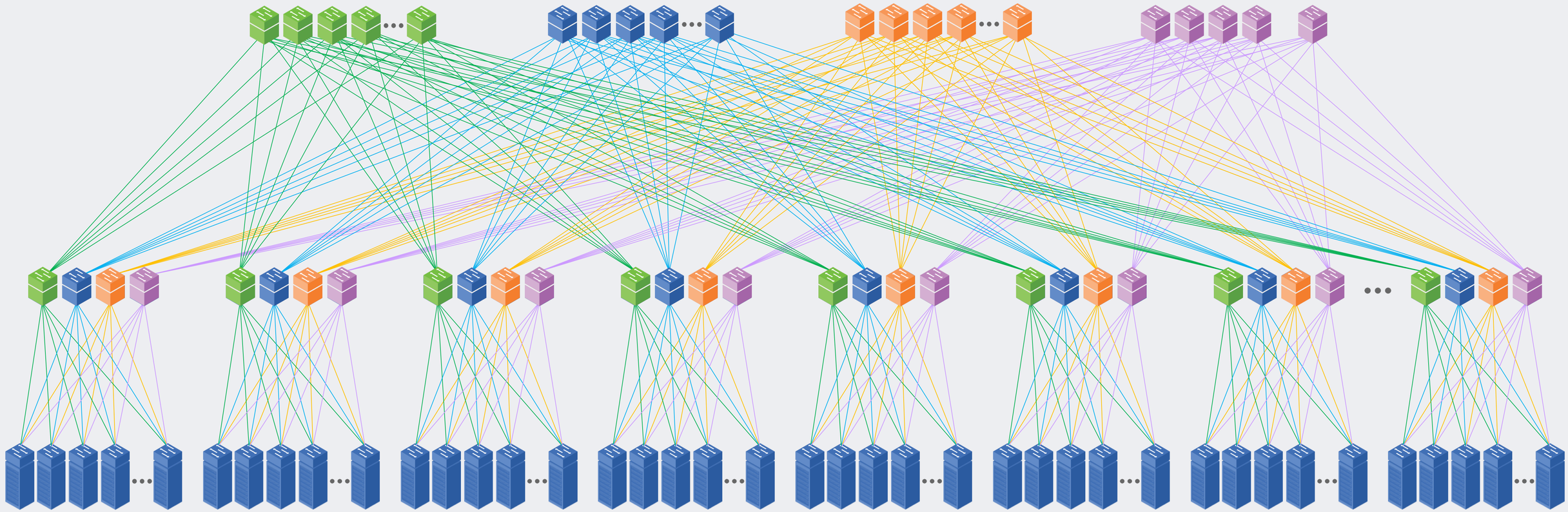
TTLd: a mixed approach

- **Every packet** in the network is a **probe**
- 1 bit in the packet header identifies it as retransmitted
- Use end host **marking** to be **precise**
- **Marked packets** are undistinguishable for network devices, so they **follow the same path**

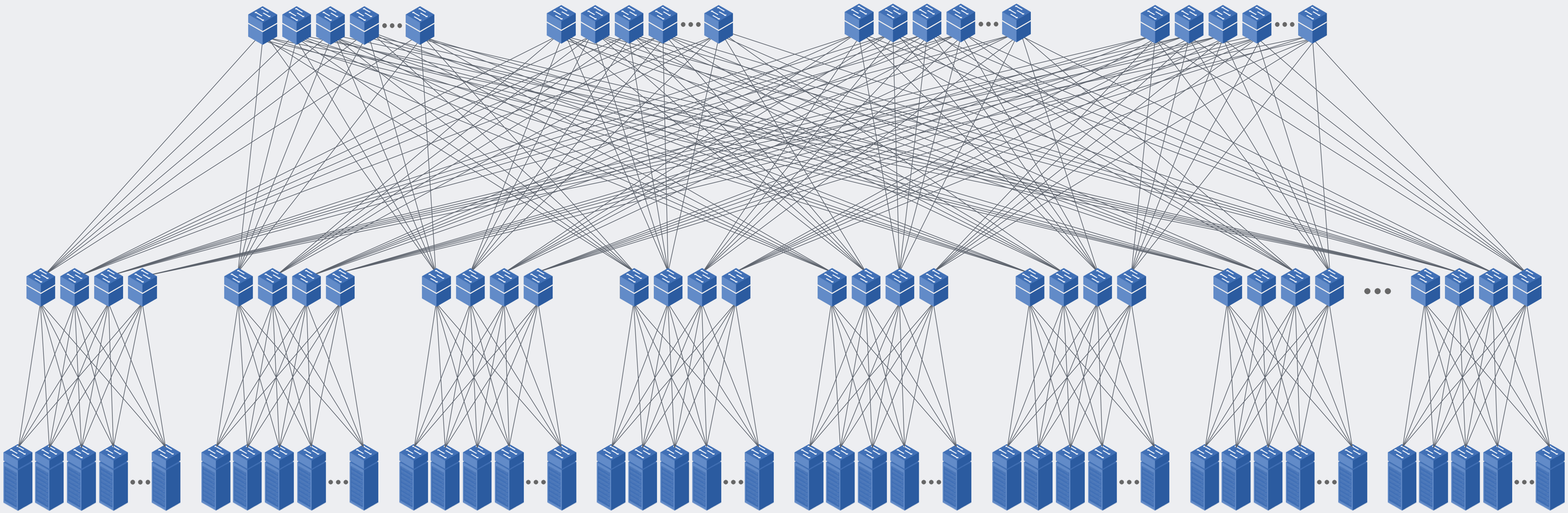
Where does loss come from?

- Network devices from a same “group” **balance traffic** hopefully **equally** according to ECMP hashing
- One device **exposing more retransmit** (in number or percent) than others may be dropping packets
- This also gives a view on **congestion** on devices not as per pure packets transmitted but with **E2E performance** view
- Neighboring device “groups” are seeing the impact of the bad device

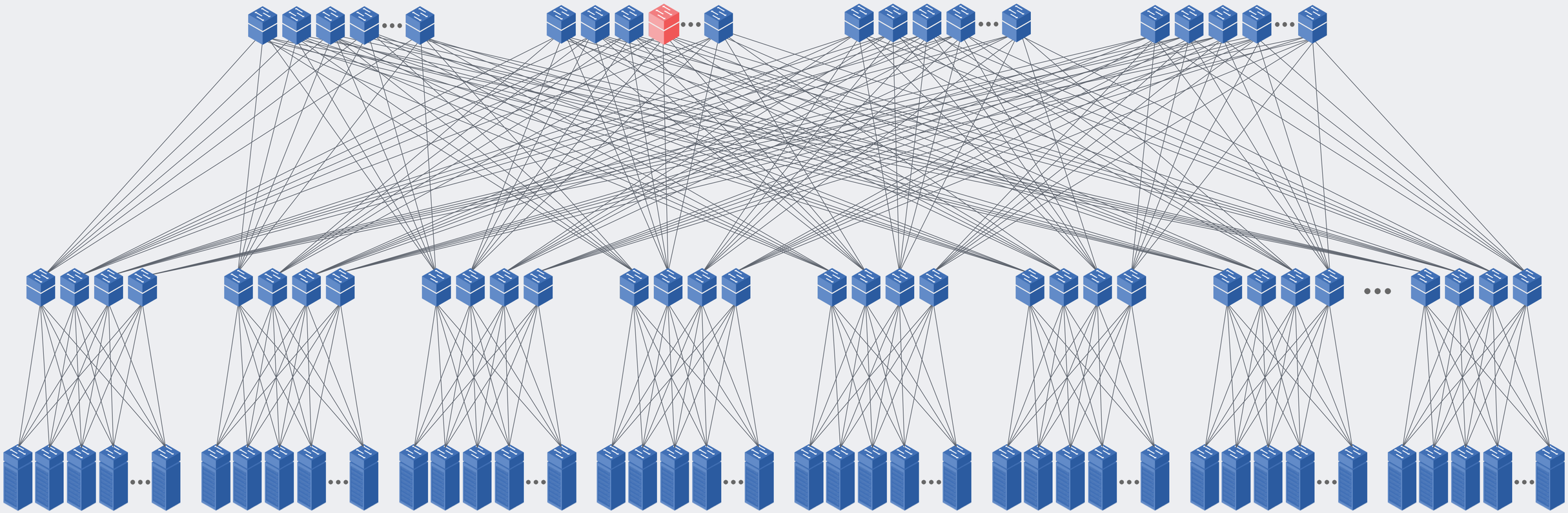
Typical Network Fabric Design



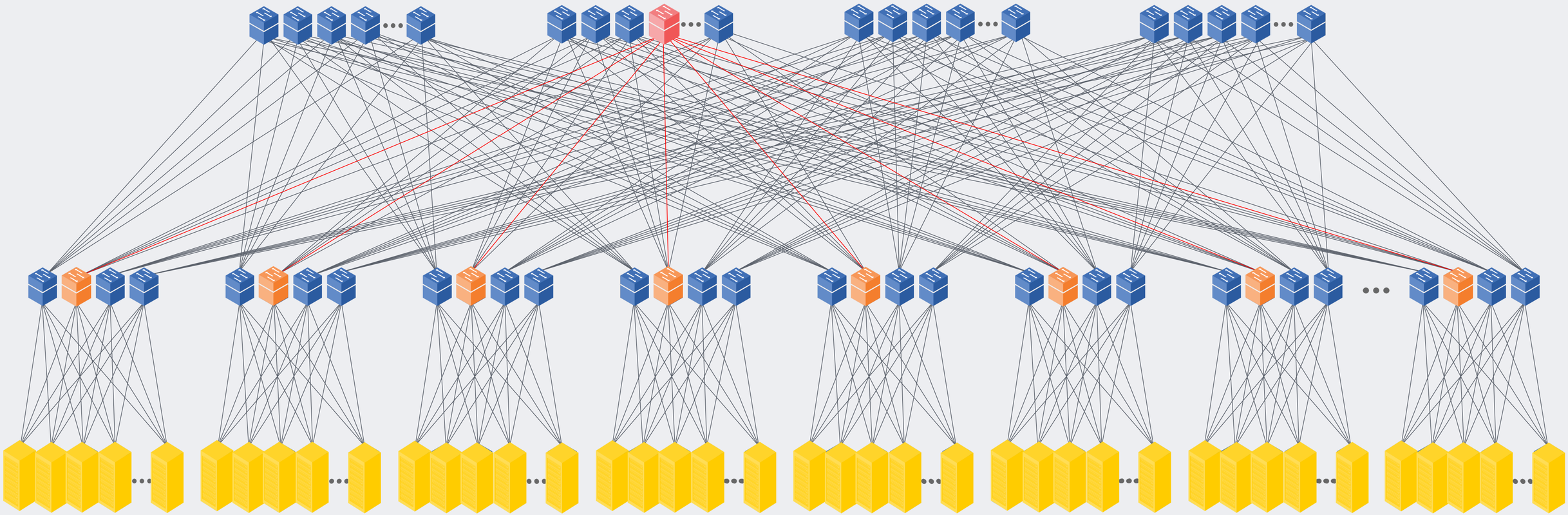
Typical Network Fabric Design



Typical Network Fabric Design



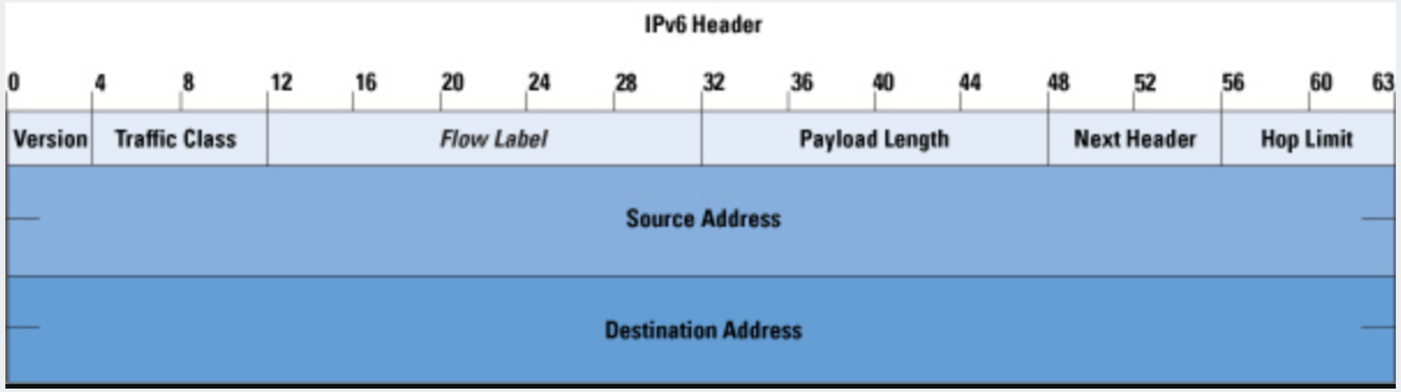
Typical Network Fabric Design



Technical details

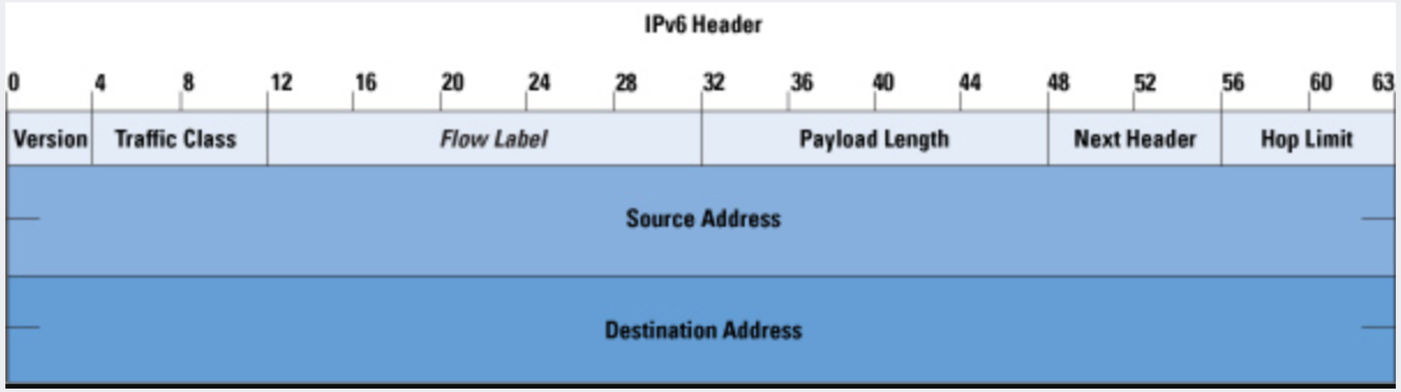
How to expose E2E performance?

- We need to find a **bit** in the **TCP header** that would be
  - easy to check
  - would appear the same to all network devices
  - would not change TCP behavior
  - would not change the packet flow if set or unset



How to expose E2E performance?

- The winner is the Most Significant Bit (MSB) of TTL field (hop limit as we are all IPv6)



eBPF program

- Berkely Packet Filter has grown from filtering packets to tuning kernel handling of network events
  - Promoted by Brendan Gregg (Netflix) and Alexei Starovoitov (Facebook)

- Safe and efficient way to insert hooks in the kernel at runtime
  - ➢ Can be hooked in dedicated places (tc egress)
  - ➢ Can change the behaviour of kernel events (TCP retransmit)

E2E performance exposed

Marking MSB of TTL field (hop limit) via eBPF program

- Use of eBPF to mark retransmitted packets with TTL of 255
  - Marking from TCP stack retransmit hook
- **1 bit in the IP header** of any packet shows if it was **retransmitted**

E2E performance counting

Unsampled data through FBoss devices

- We have a **counter** and bump it through **ACL match** (MSB of TTL)
  - Best part is that as network devices have to decrement TTL of every packet, **checking this bit is practically free**!

- This is currently implemented on all Top Of Rack

Exposing E2E performance

Collect counters on FBoss

- **No sampling**
  - Precise view
  - Aggregation at source is possible via host retransmit dataset
  - Aggregation per destination rack now possible

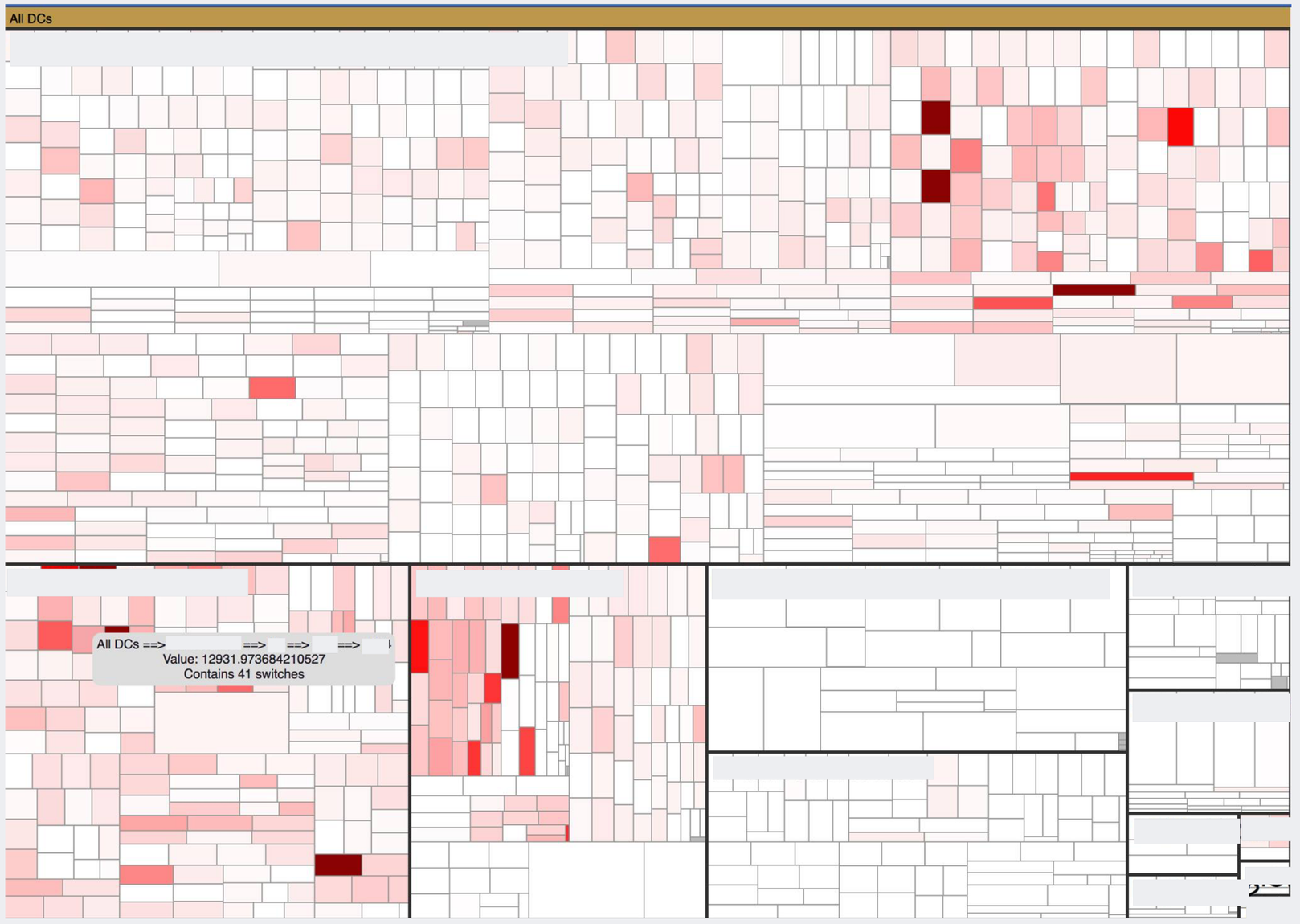
Exposing E2E performance

Collect counters on other providers' network devices

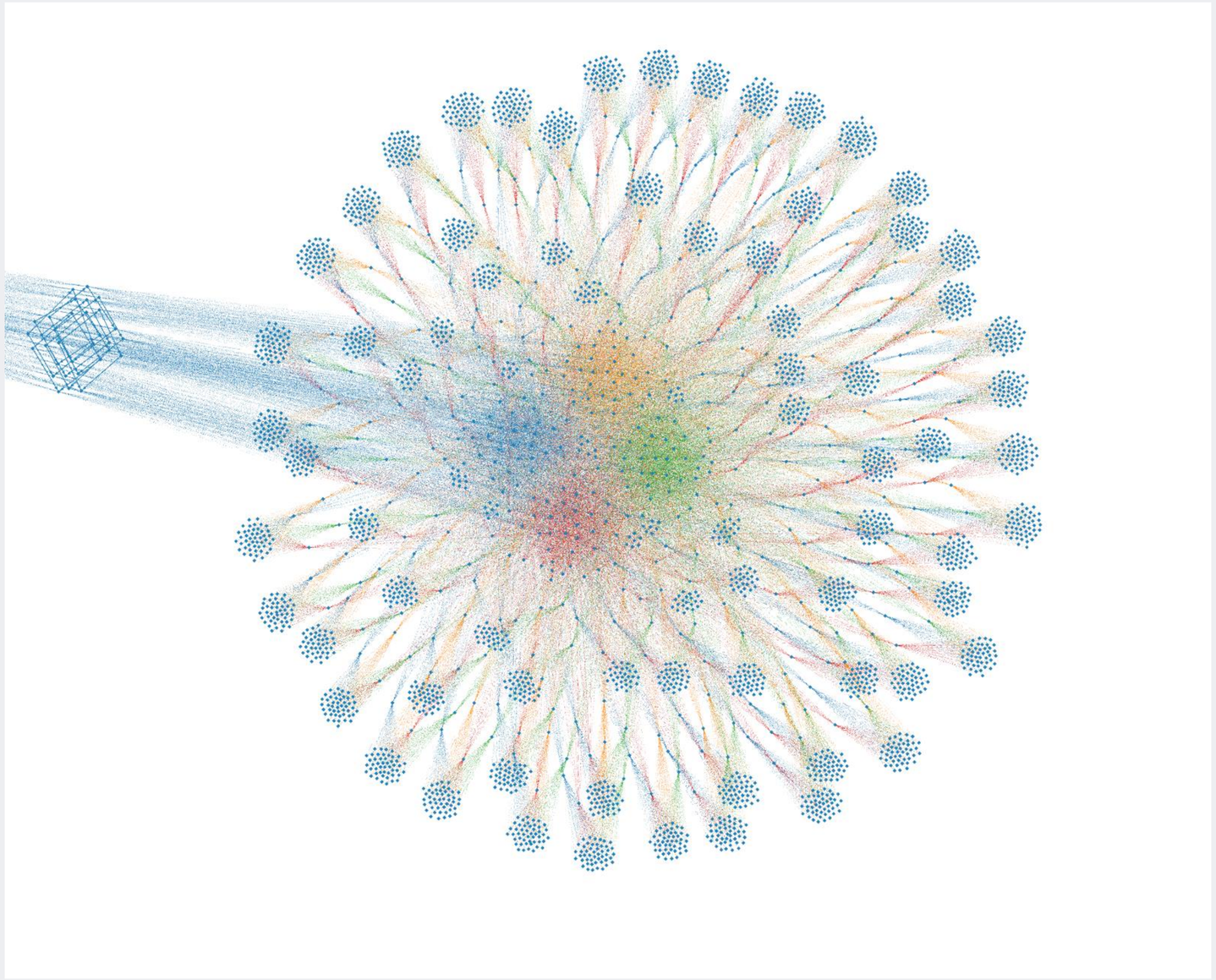
- Use collection framework (Facebook framework to collect counters from any proprietary network device) and ACL matching

- **No sampling**
  - Aggregated per ECMP hashing of production traffic
  - Exposes retransition on a specific device

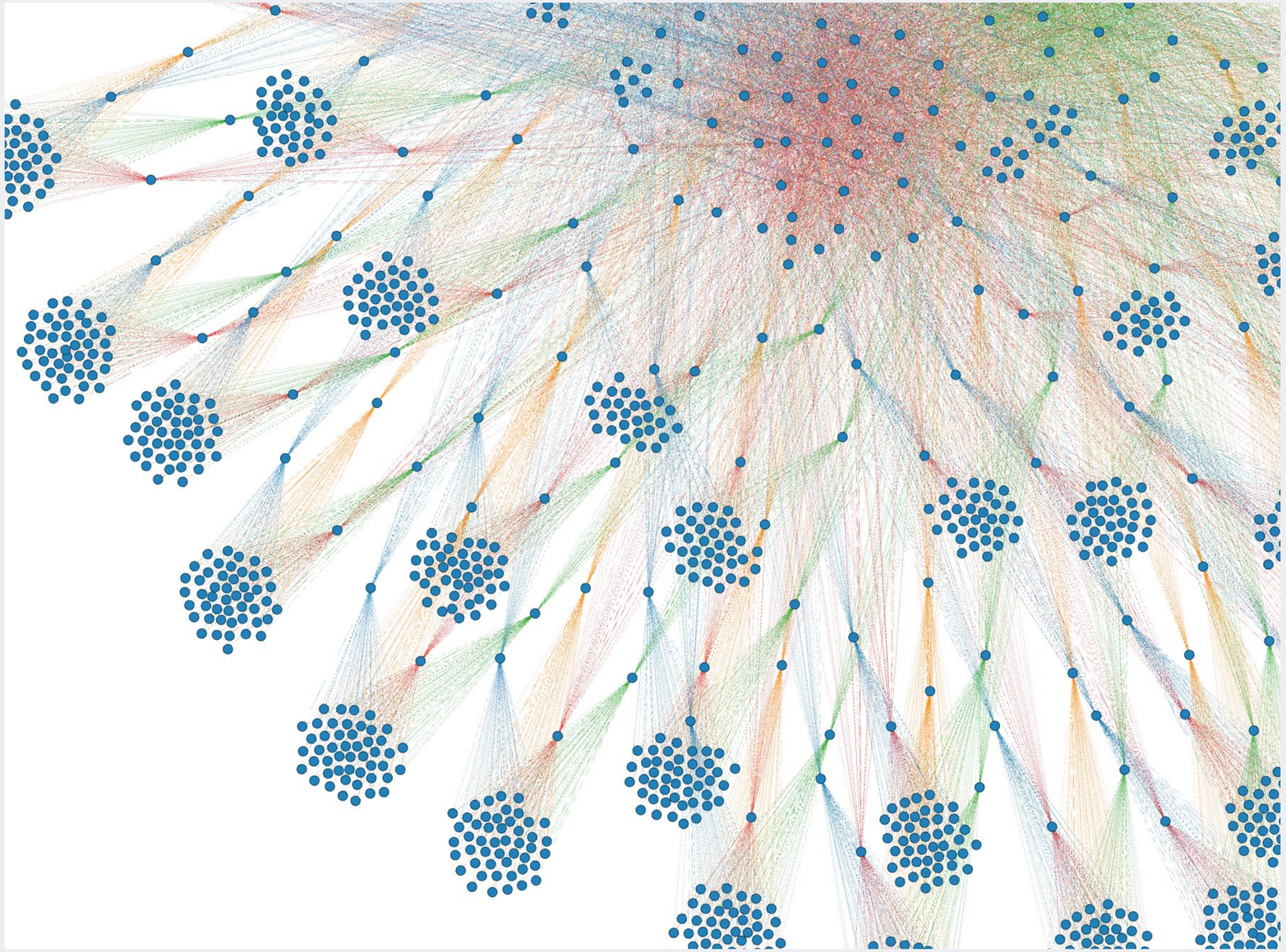
Visualization through internal tools



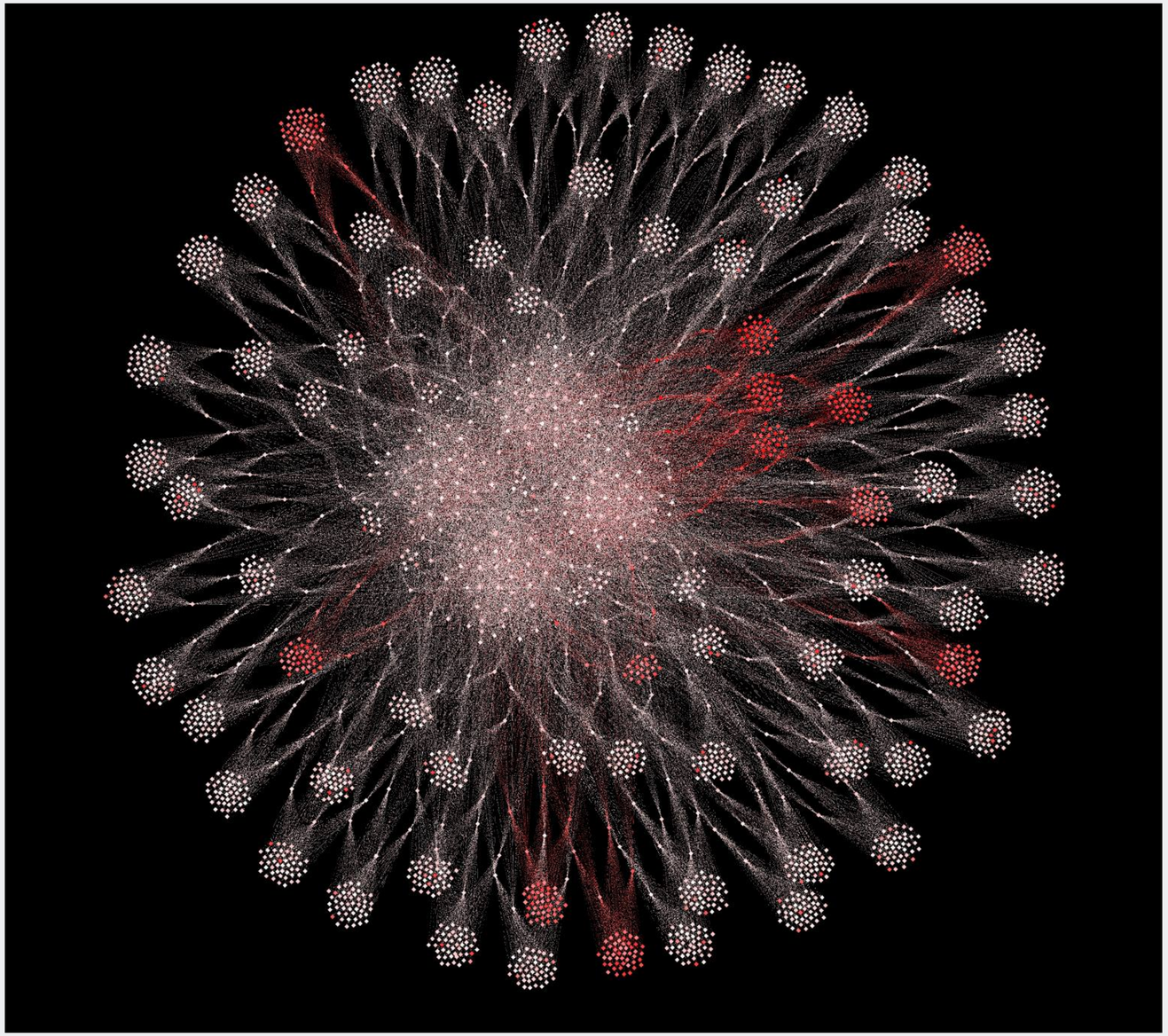
Visualization through internal tools



Visualization through internal tools



Visualization through internal tools



facebook