BEYOND HARDWARE LOAD BALANCING

@HOFFNZ

WHERE WERE WE?

A very traditional model

TWITTER NETWORK

What do we have to attempt to address?

- Exponential traffic growth
- Regularly adding new services
- Challenges in scaling hardware to keep up
- Vendor hardware

WHERE DID WE COME FROM?

How did it look?

- Anycast between sites (this is still the case)
- Load balancer on a stick connected to router. Router policy routes traffic as needed into load balancer.
- Load balancers run health checks against servers
- Load balancers route traffic evenly to servers

WHAT'S THE PROBLEM?

Why did we want to move away from this?

- Cost of dedicated LB hardware + team to maintain it + ports to connect it
- Low flexibility - fixed options for health checking
- Complexity in routing layer to divert traffic into load balancers
- Harder to scale as traffic grows

What are the alternative options?

@HOFFNZ

1 - LB IN THE APPLICATION

Push the issue up to the app

- Load balancing is only an issue due to a requirement to hit “that single VIP” or “that single DNS name”
- What services actually have a hard requirement for this?
- Can we push load balancing up to the application? We do this already for many applications where we control client + server
- Unicast at the network layer

2 - ROUTER BASED LOAD BALANCING

Distribute the load balancing functionality

Where load balancing is actually required, can we redistribute it between the router and the server?

What would be involved in doing this?

How do we achieve equivalent functionality?

ROUTER BASED LOAD BALANCING

What does the load balancer actually do?

@HOFFNZ

WHAT DOES A VENDOR LOAD BALANCE ACTUALLY DO?

- Distribution of traffic over a server pool
- Graceful insertion/removal of hosts (don’t re-hash all the hosts every time one is removed/added)
- Health checks
- Advertise routes
- Map traffic to hosts that are not necessarily directly connected

WHAT DOES A VENDOR LOAD BALANCE ACTUALLY DO?

- Flexible load balancing methodologies
- Graceful TCP session migration (don’t interrupt existing TCP sessions when a box goes up/down)

WHERE CAN WE SEND THESE FUNCTIONS?

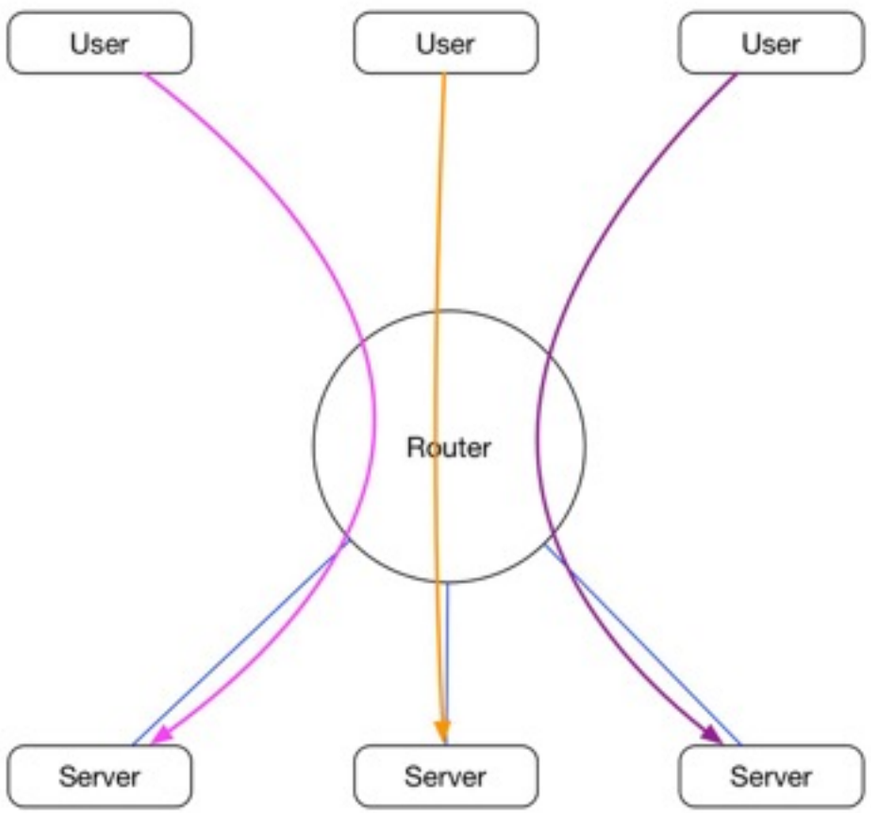
- ECMP —> Router
  - Can hash 64-128 wide (vendor dependent)
  - Can load balance wider if needed
    - Muti-layered ECMP (will result in ASIC resource burn)
    - DNS based load balancing over multiple VIPs for same service

WHERE CAN WE SEND THESE FUNCTIONS?

- ECMP —> Router
  - Consistent ECMP
  - Challenges with ICMP-too-large not being hashed correctly
    - Easily fixable by router vendors - ICMP-too-large response carries the L4 header from the original packet

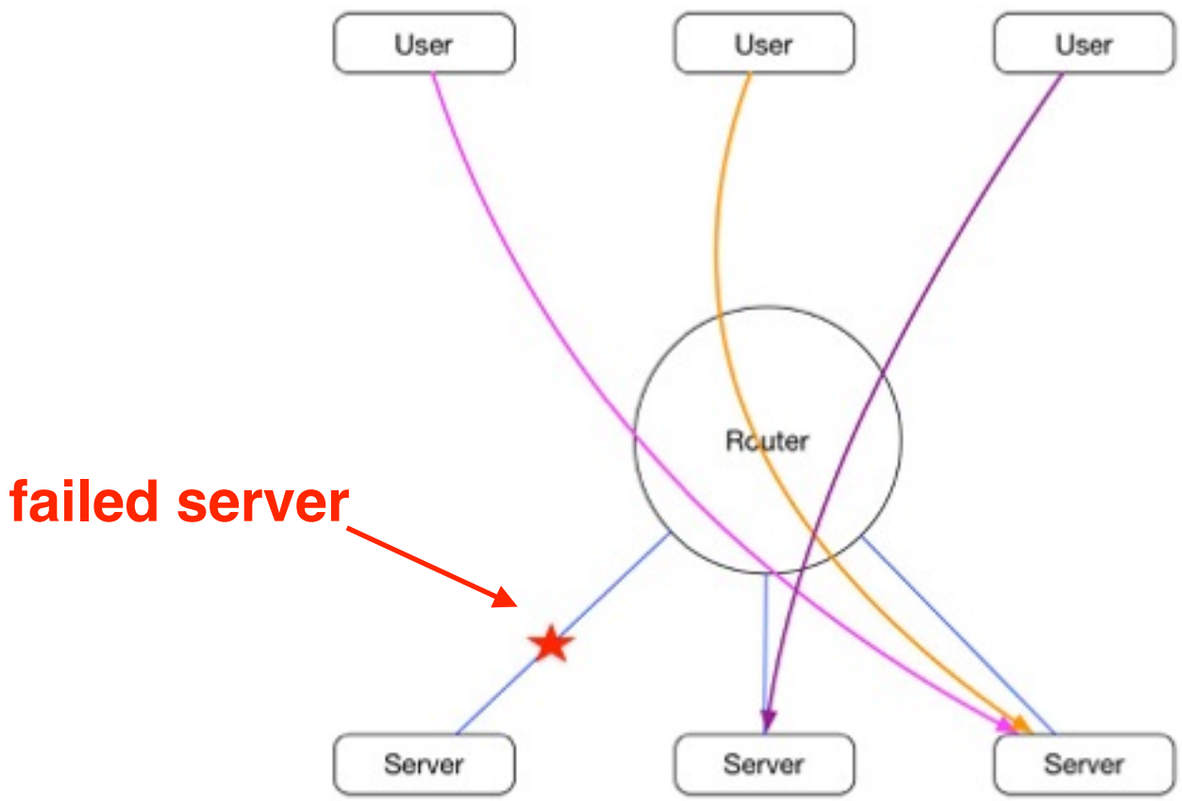
CONSISTENT ECMP - WHAT IS IT?

- Normal operation



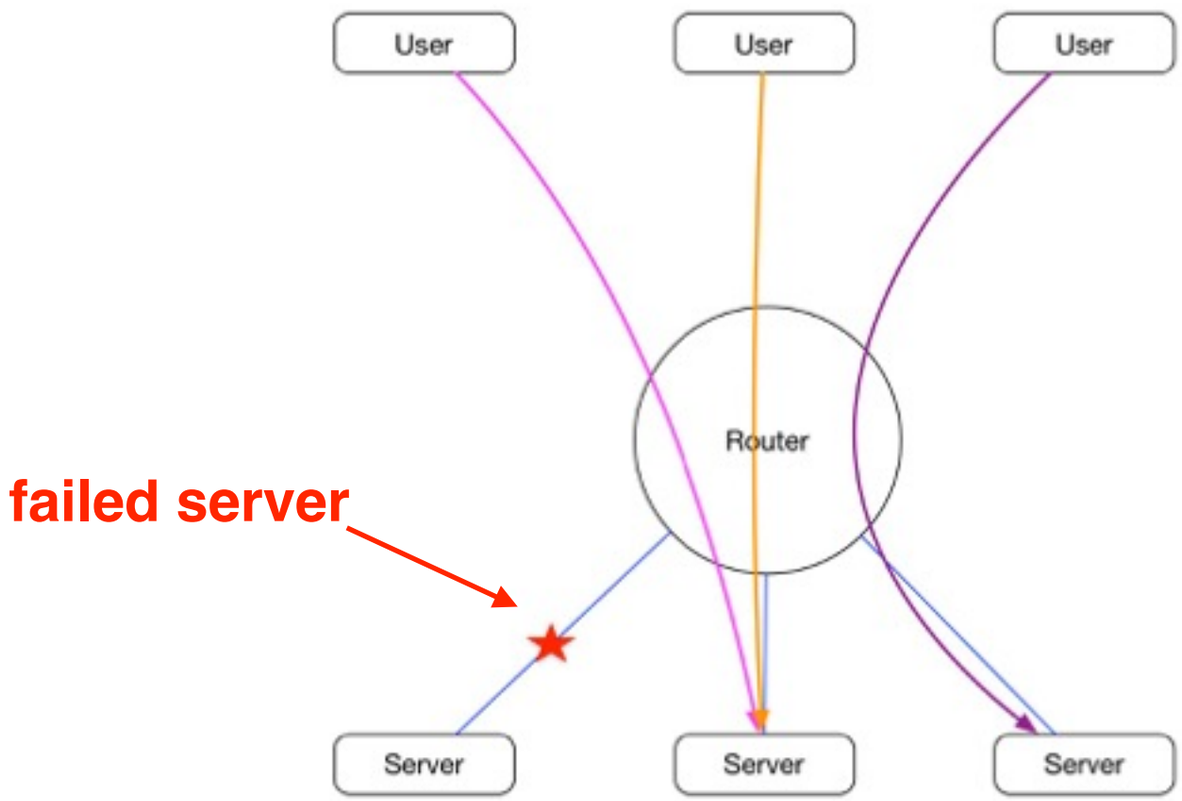
CONSISTENT ECMP - WHAT IS IT?

- Normal ECMP functionality - post failure all sessions re-hashed to different servers



CONSISTENT ECMP - WHAT IS IT?

- Consistent ECMP functionality - post failure only failed server sessions re-hashed



WHERE CAN WE SEND THESE FUNCTIONS?

- ECMP —> Router
  - Consistent ECMP
  - Challenges with ICMP-too-large not being hashed correctly
    - Easily fixable by router vendors - ICMP-too-large response carries the L4 header from the original packet

WHERE CAN WE SEND THESE FUNCTIONS?

- Map traffic to hosts that are not necessarily directly connected
  - Can use GRE tunnels for inbound-traffic only (DSR like behavior) to achieve this
  - Outbound traffic still routed as normal traffic (outside of GRE tunnels) - allowing us to minimize resource burn on the router

WHERE CAN WE SEND THESE FUNCTIONS?

- Advertise routes to network & health-checks
  - Can advertise routes from server itself
    - Use BFD for fast-failover (be careful with timers on CPU implemented BFD)
  - We extended ExaBGP to perform health checking with knowledge of our application
    - ExaBGP health checking very modular

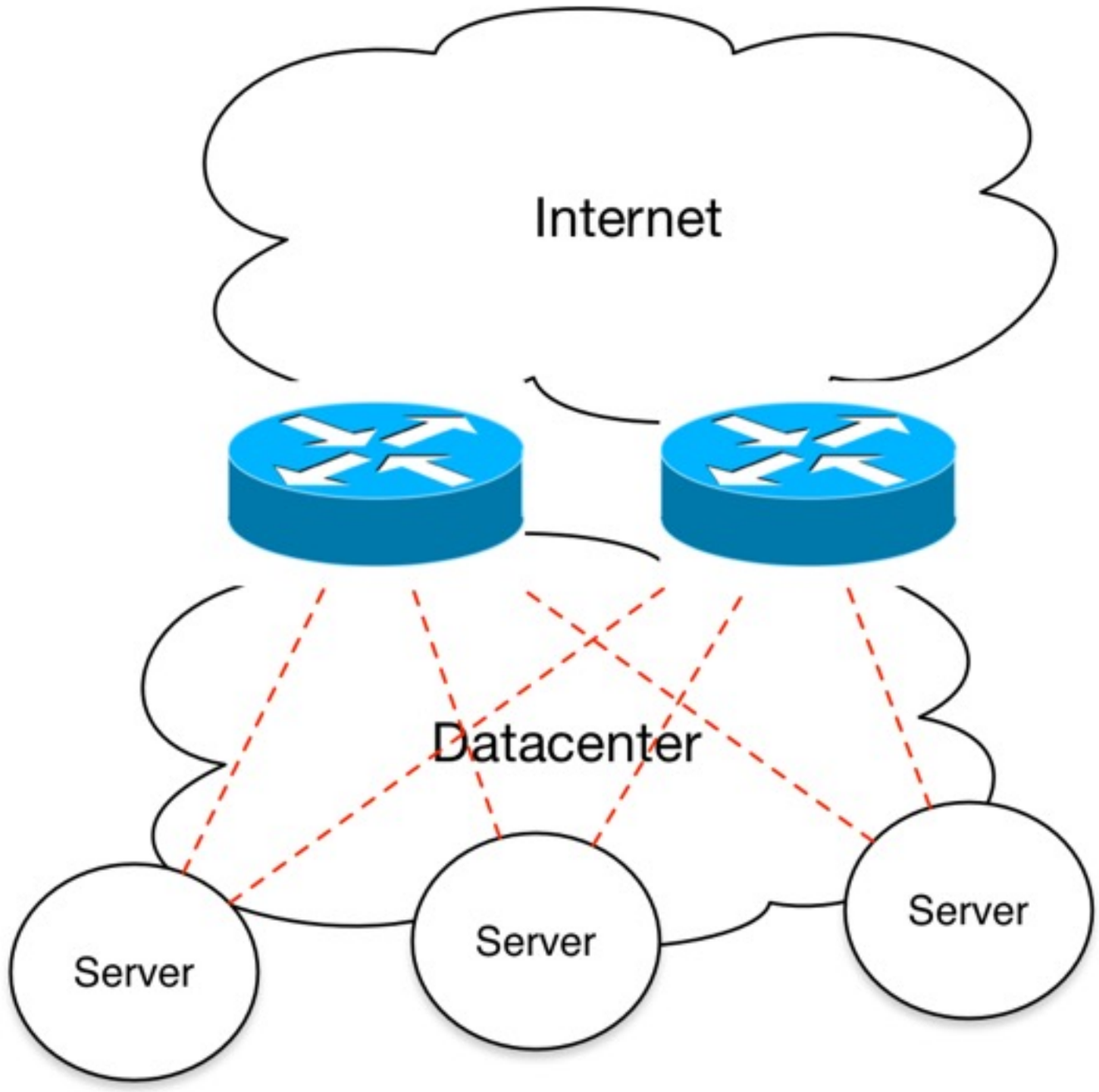
LIMITATIONS

- We can’t replicate the following functionality
  - Flexible load balancing methodology
  - Graceful TCP session migration (don’t interrupt existing TCP sessions when a box goes up/down)
- These aren’t required in our stack

IMPLEMENTATION

HOW DID WE DO IT?

@HOFFNZ


Internet
Datacenter
Server
Server
Server

HOW DID WE DO IT?

Connectivity

- Due to unequal server allocation between racks, ECMP to the rack would cause hotspots
- Not all legacy switch hardware can support large width ECMP or consistent hashing
- Can solve this by doing all ECMP (over GRE) on DC edge routers

HOW DID WE DO IT?

Connectivity

- GRE tunnels from the core routers to every server
- Allows us to ECMP between every server in a consistent manner
- Inbound traffic on GRE tunnels costs 2x additional ASIC passes through the router
- Need to distribute GRE tunnels over all ASICs

HOW DID WE DO IT?

Connectivity

- BGP from the routers to the server inside the GRE tunnels
- BFD to ensure fast failure detection (loose timers to mitigate risk that we will overwhelm BFD keepalives in a DDoS)
- Consistent ECMP to ensure that we didn’t re-distribute TCP sessions every time a server came in/out of service

```
chassis {
    fpc 0 {
        pic 0 {
            tunnel-services {
                bandwidth 10g;
            }
        }
    }
}
interfaces {
    gr-0/0/0 {
        unit 0 {
          tunnel {
                source 2.2.2.2;
                destination 3.3.3.3;
            }
            family inet {
                address 1.1.1.160/32 {
                        destination 1.1.1.161;
                }
            }
        }
    }
}
routing-options {
    forwarding-table {
        export LOAD-BALANCING-POLICY;
    }
}
```

not a typo

```
protocols {
    bgp {
        group FRONT-END-SERVERS {
            import FRONT-END-SERVER-IMPORT;
            export REJECT-ALL;
            peer-as 65500;
            multipath;
            neighbor 1.1.1.161;
        }
    }
}
policy-options {
    policy-statement FRONT-END-SERVER-IMPORT {
        term VIP-IMPORT {
            from {
                route-filter 1.2.3.4/24 prefix-length-range /32-/32;
            }
            then {
                load-balance consistent-hash;
                accept;
            }
        }
        then reject;
    }
    policy-statement LOAD-BALANCING-POLICY {
        then {
            load-balance per-packet;
        }
    }
}
```

HOW DID WE DO IT?

Healthchecks & exabgp

- ExaBGP on the server
  - Run BFD on server also - again keep the timers loose - the servers don’t not have NIC level BFD abilities!
  - Health checks written into ExaBGP

WHAT WENT WRONG?

CHALLENGES

@HOFFNZ

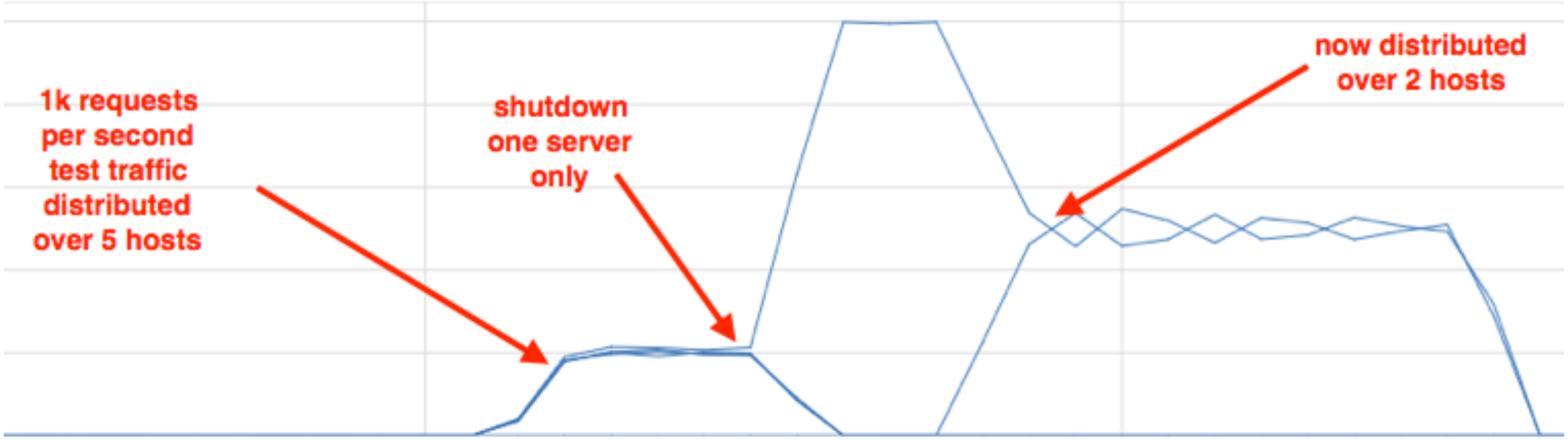
CHALLENGES

Hardening hosts against attacks

- Need to harden hosts to be able to receive packet floods.
- Some good information on how to do this can be found in a presentation by Jesper Dangaard Brouer, Red Hat - https://t.co/xaFM9m99Mn

CHALLENGES

Consistent ECMP didn’t initially work



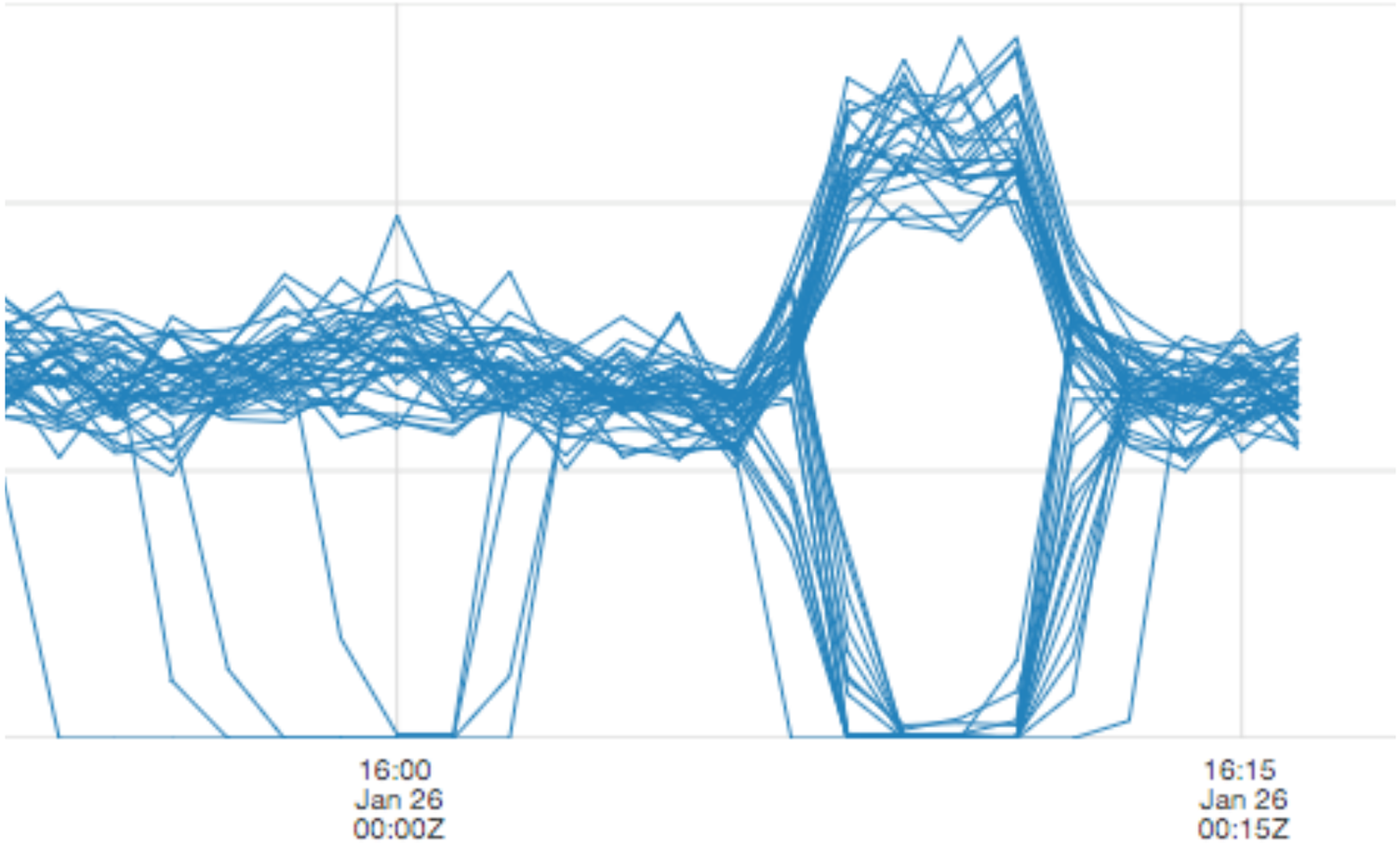
CHALLENGES

Router software support

- We had some struggles with our router platform correctly supporting consistent ECMP on GRE tunnels.
  - Single server bouncing would cause all TCP sessions to move to new host (reset session)
  - Worked with vendor to develop a fix for this.

CHALLENGES

Working Consistent ECMP



CHALLENGES

Re-hashing between routers

- Each router will maintain a unique hash table, some issues will cause TCP sessions to move between servers
  - Router failure
  - BGP path changes into the network
- This turned out to be an acceptably low amount of re-hashing events.

CHALLENGES

MTU discovery

- ICMP packet-too-large alerts contain the l3+l4 header in payload, but do not get hashed correctly
- Lowered the MSS to 1360
    - Fixes for 99.99% of clients
- TCP mtu-discovery

CHALLENGES

GRE tunnel overhead

- Need to ensure we are catering for the 24 byte GRE tunnel overhead
- Provided we are lowering our MTU to deal with client networks above this becomes a non-consideration
- If it is a consideration, can either reduce MSS, or increase MTU on DC segments

CHALLENGES

How do we deal with a flapping server?

- What happens when a server is flapping for some reason?
- Hardware LBs provided mechanisms to back off from this server until issues resolved. How do we move this to a more traditional router?
- Route damping achieves the necessary functionality (use carefully!)

CHALLENGES

Draining hosts or sites

- How do we drain hosts or sites?
  - Have always drained sites by ceasing to accept anycast routes from load balancers. Can implement same functionality facing servers.
  - Can cease to accept anycast routes from one server at a time to perform maintenance.

CHALLENGES

Wide-ranging vendor support?

- We don’t want to be locked into one vendor.
  - Confirmed that all but one of major backbone router vendors support this.
  - Performed testing to validate.

@HOFFNZ