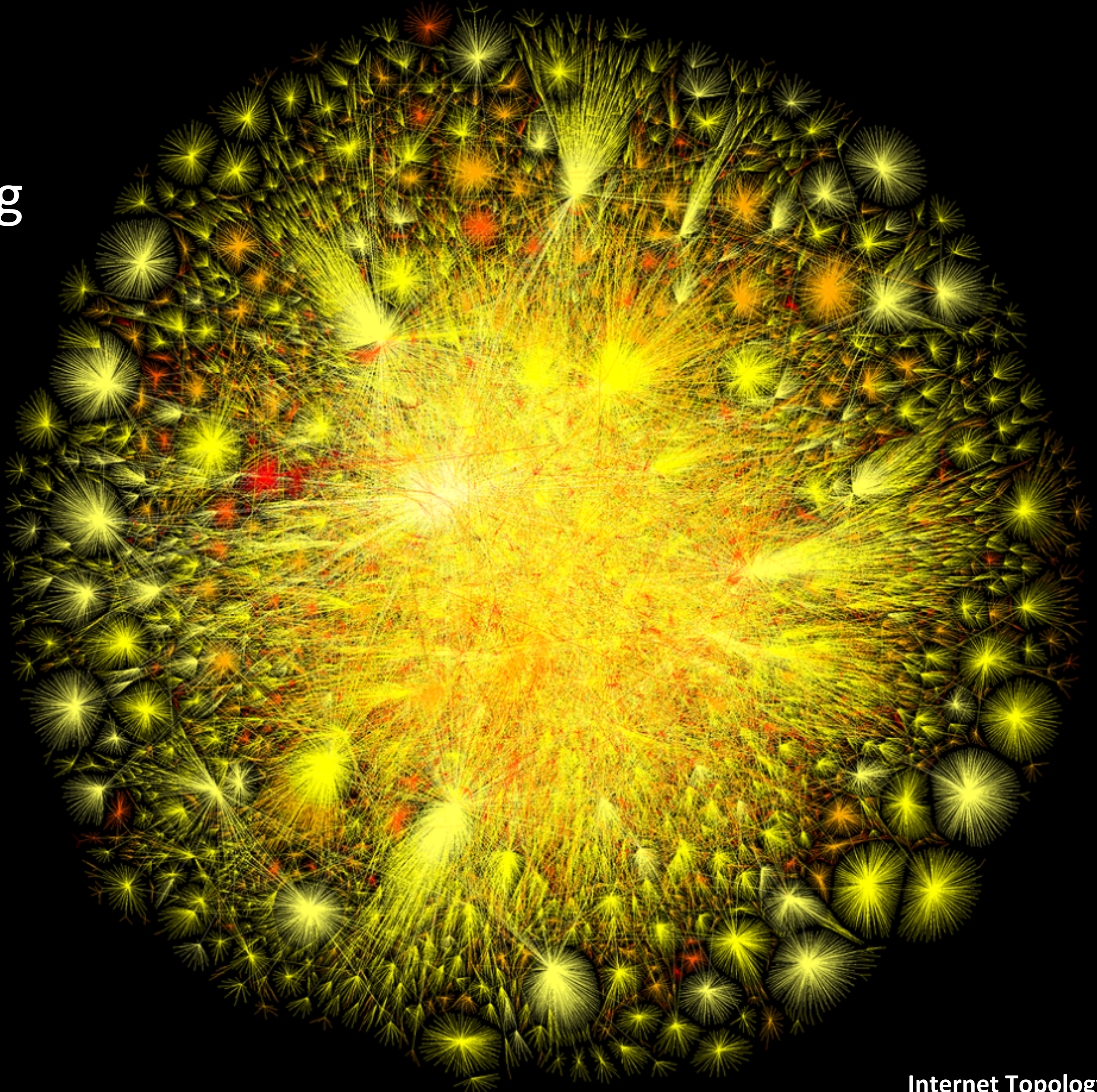
FastRoute:
A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs

Ashley Flavel, Pradeepkumar Mani, David A. Maltz, Nick Holt, Jie Liu, Yingying Chen, Oleg Surmachev

Microsoft



Internet Topology image courtesy of www.opte.org

FastRoute Overview

- Most online services exist inside small set of datacenters distributed throughout the world.

- “Edge” nodes distributed throughout the Internet can reduce network latency of such services.

- FastRoute is the fully distributed mechanism used to direct users to nearby edge.

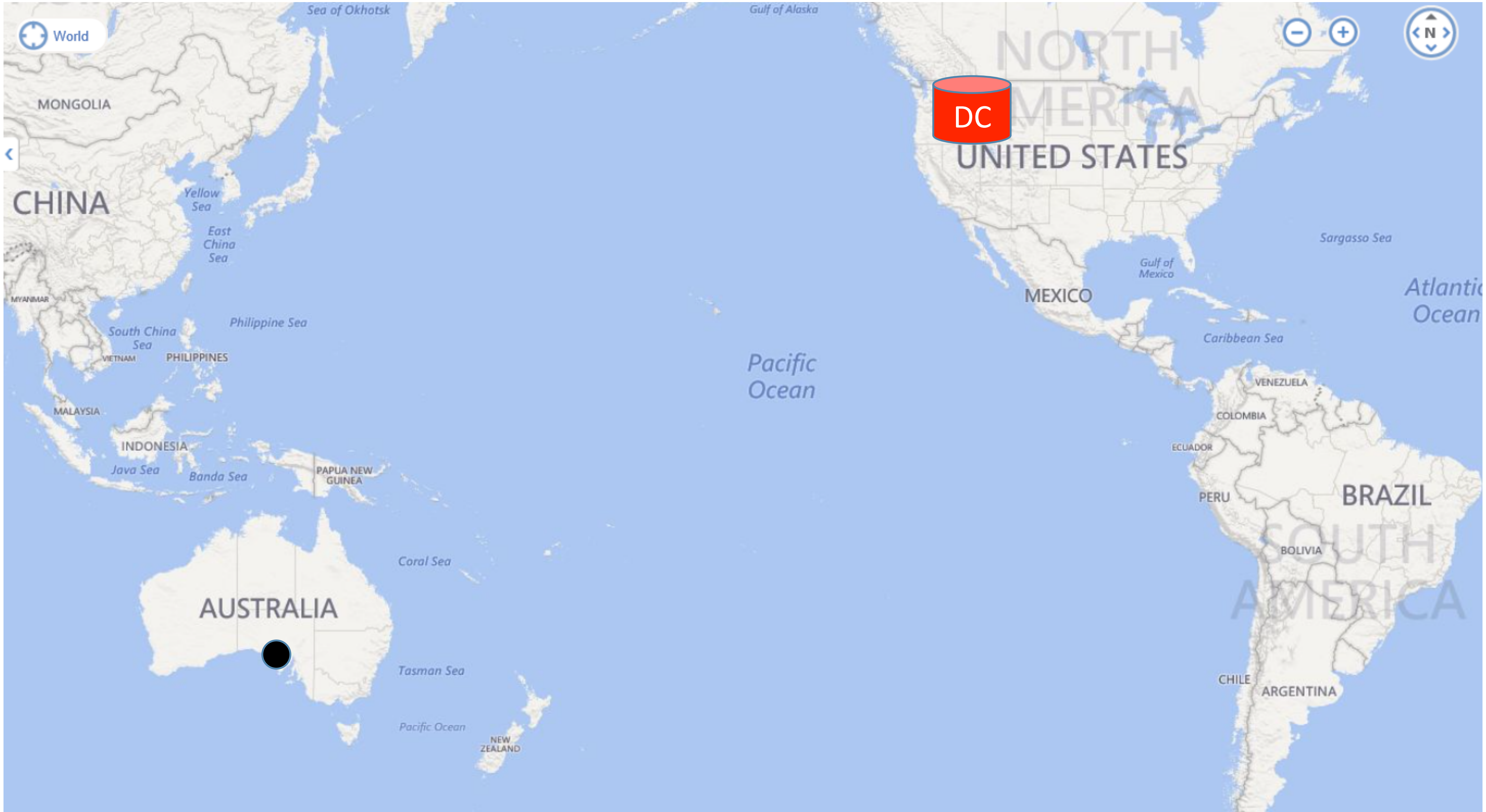
- Traffic routing in FastRoute Relies on Anycast

1. Why use an edge

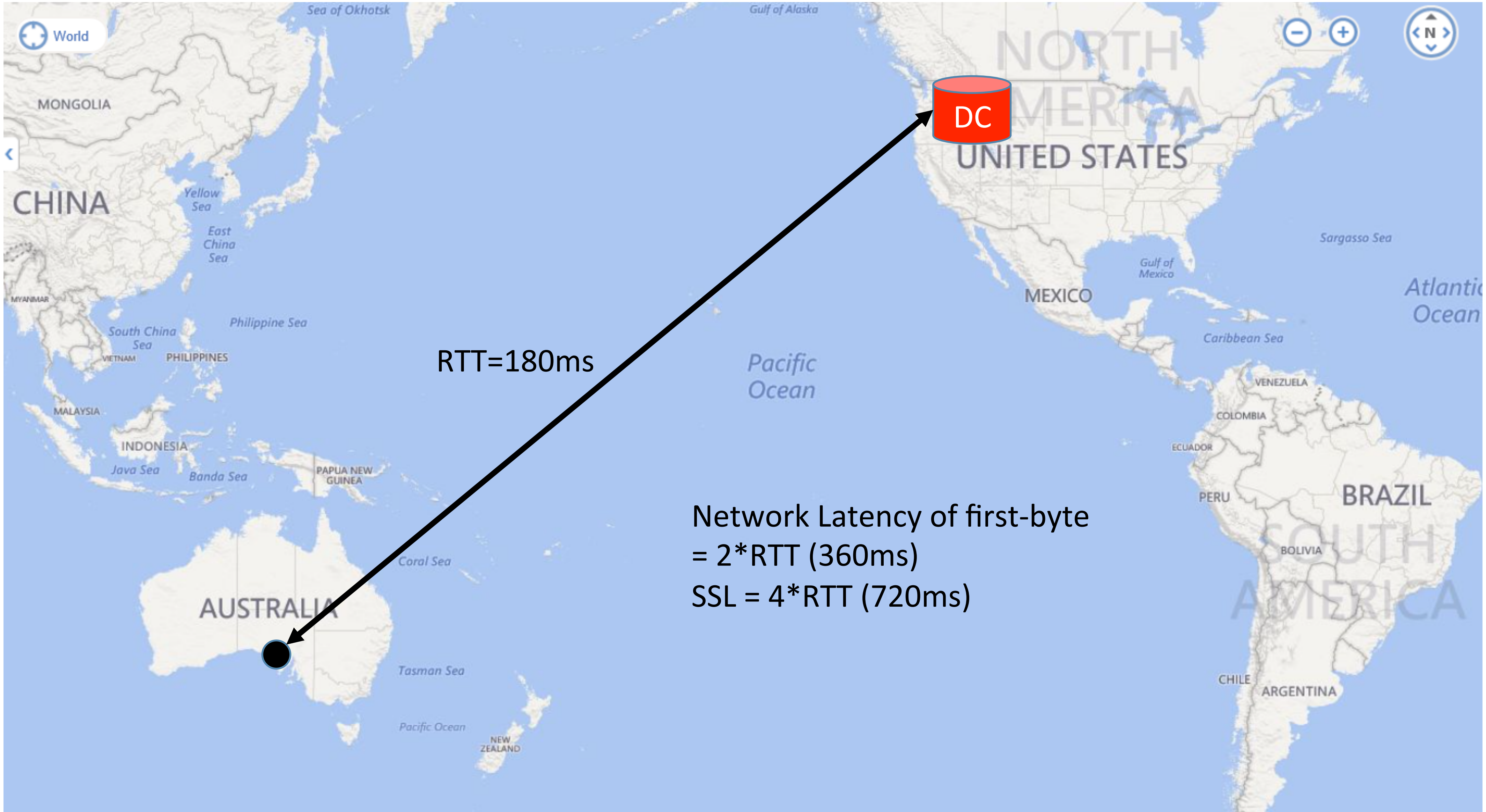
2. Choosing the “best” edge

3. Adding FastRoute for load management

Why use an Edge?



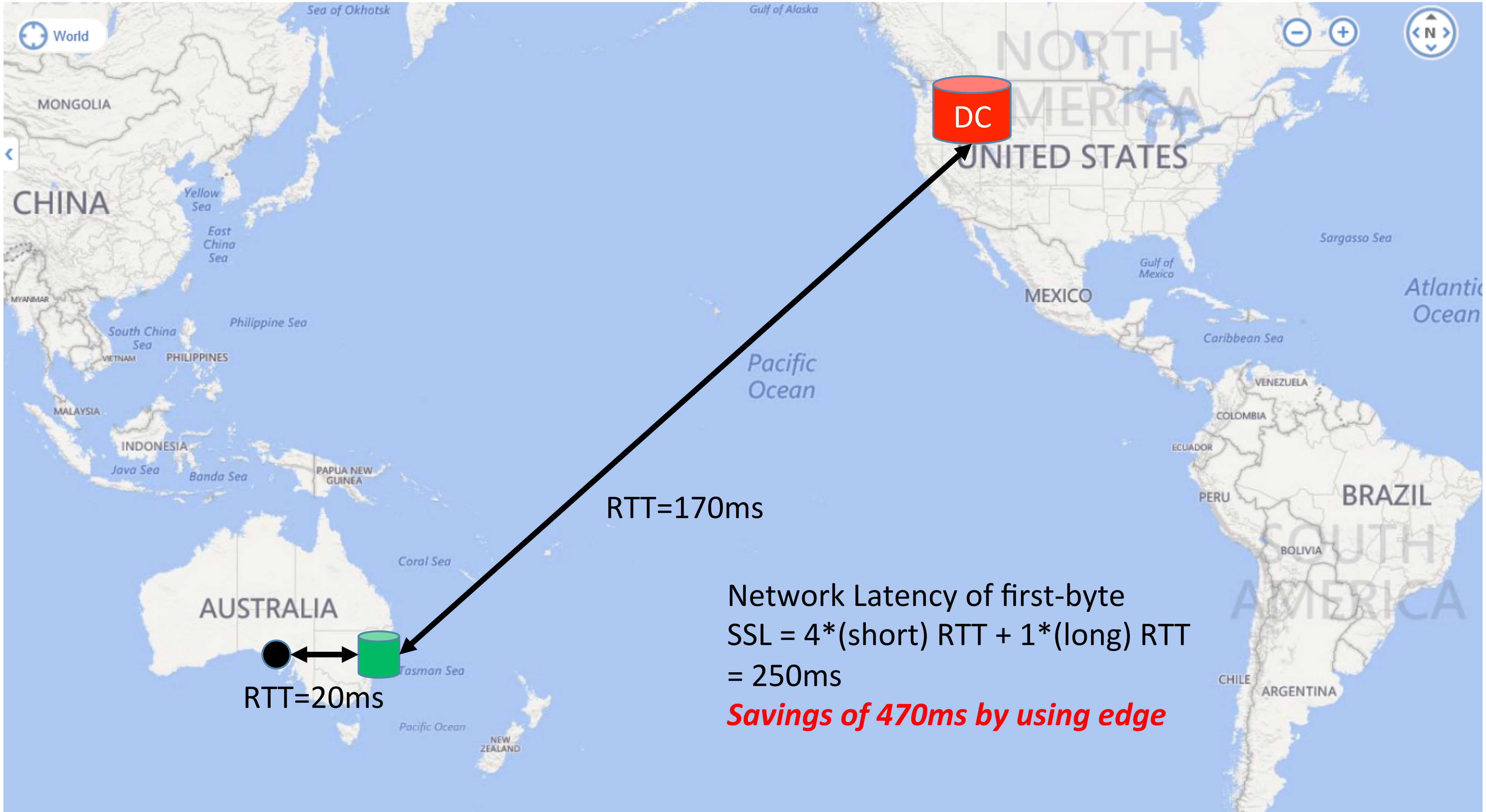
Why use an Edge?



RTT=180ms

Network Latency of first-byte
= 2*RTT (360ms)
SSL = 4*RTT (720ms)

Why use an Edge?



Network Latency of first-byte
SSL = 4*(short) RTT + 1*(long) RTT
= 250ms

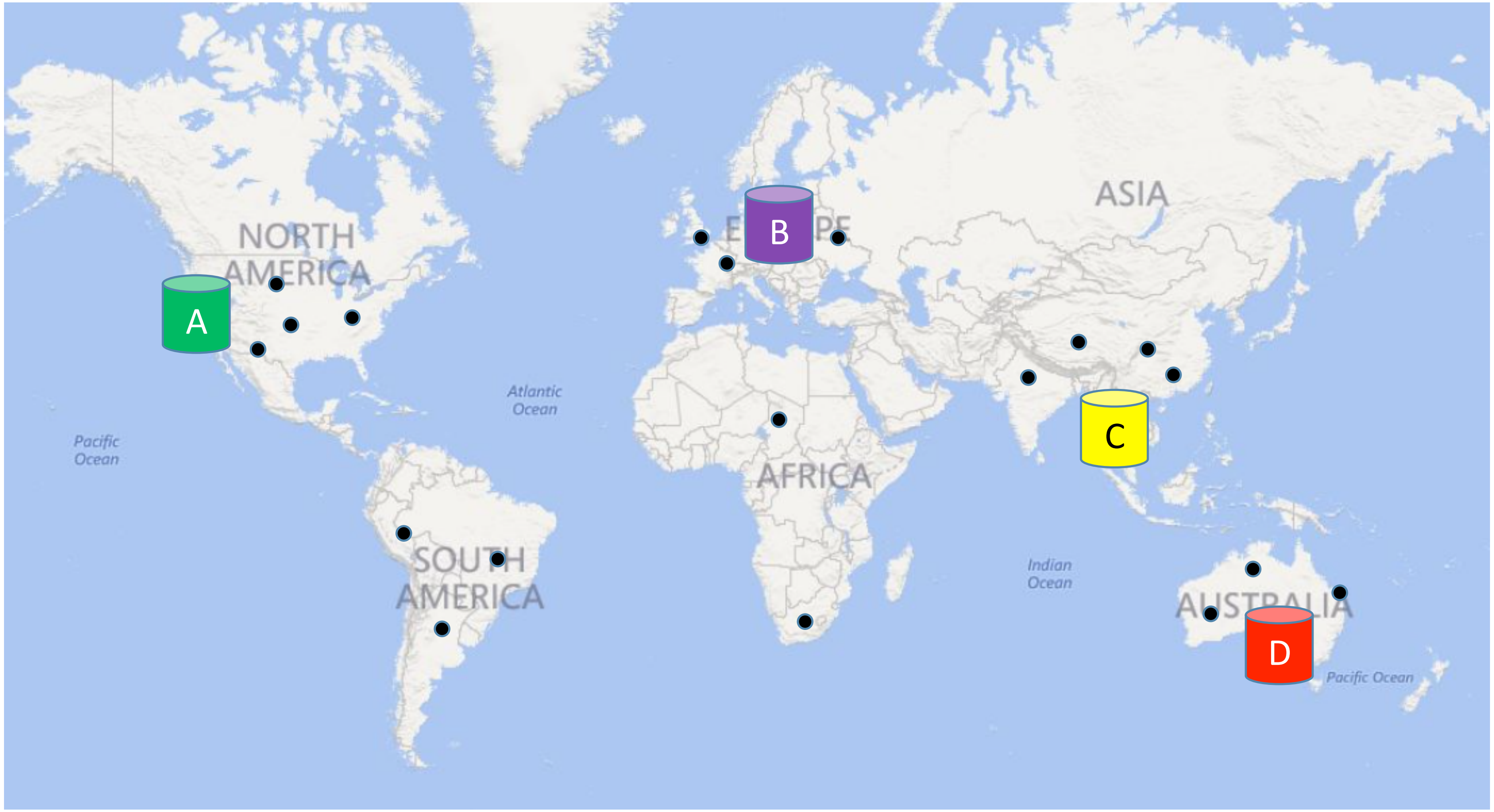
Savings of 470ms by using edge

Choosing the “best” edge?

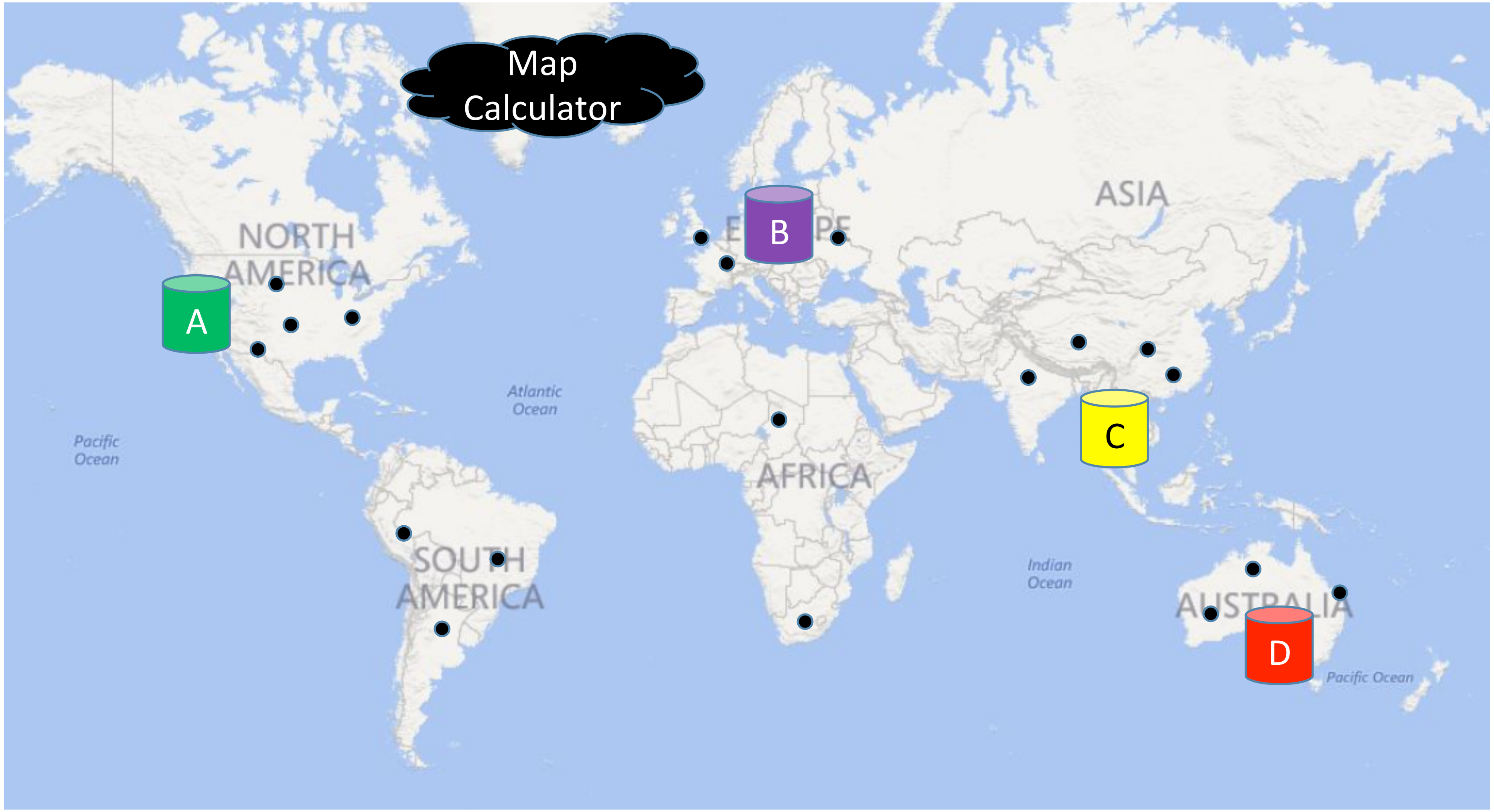
- How do I direct each user to the closest edge ?

- “Map the Internet”
- Anycast

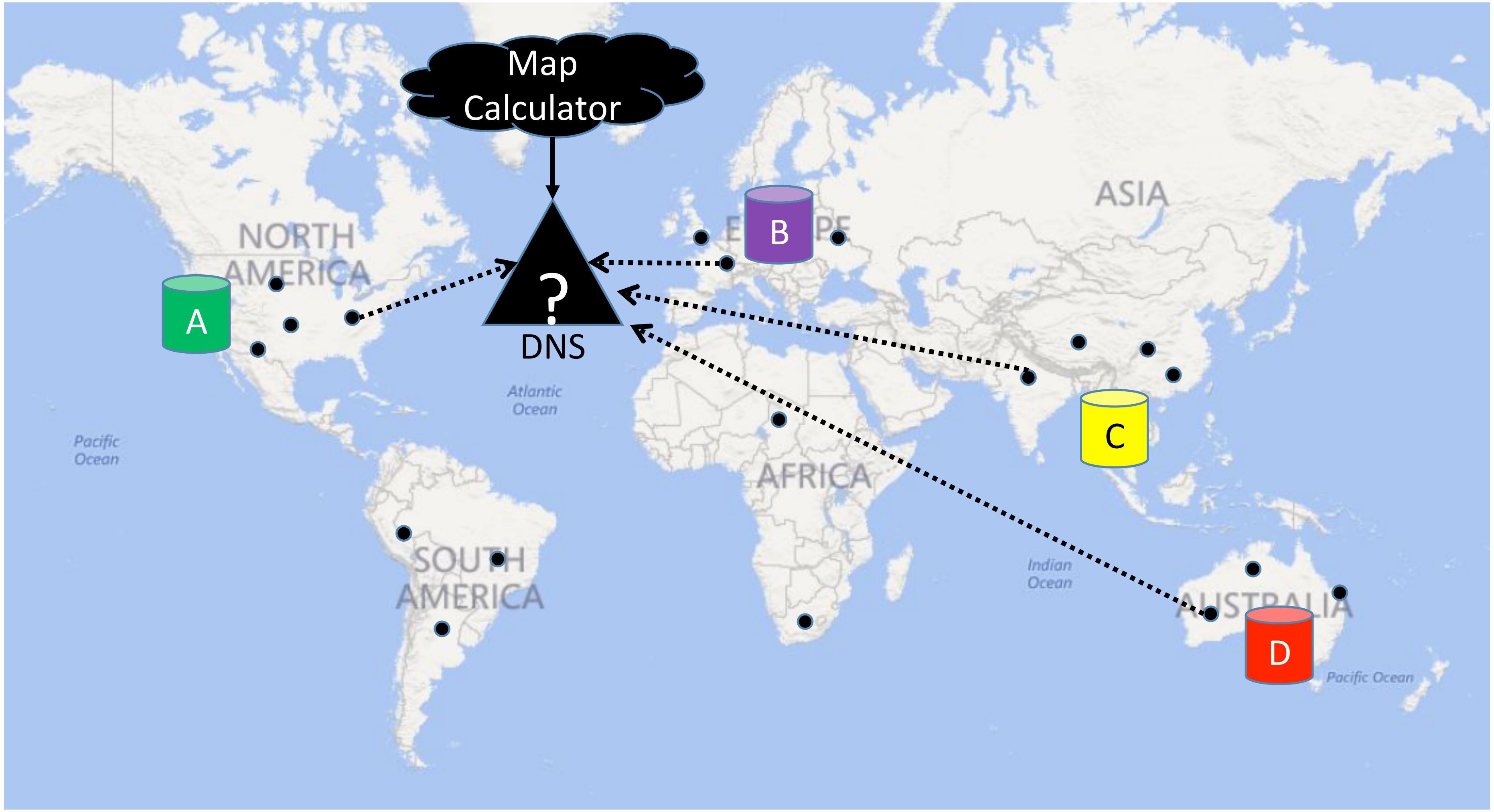
The “Map the Internet” Approach



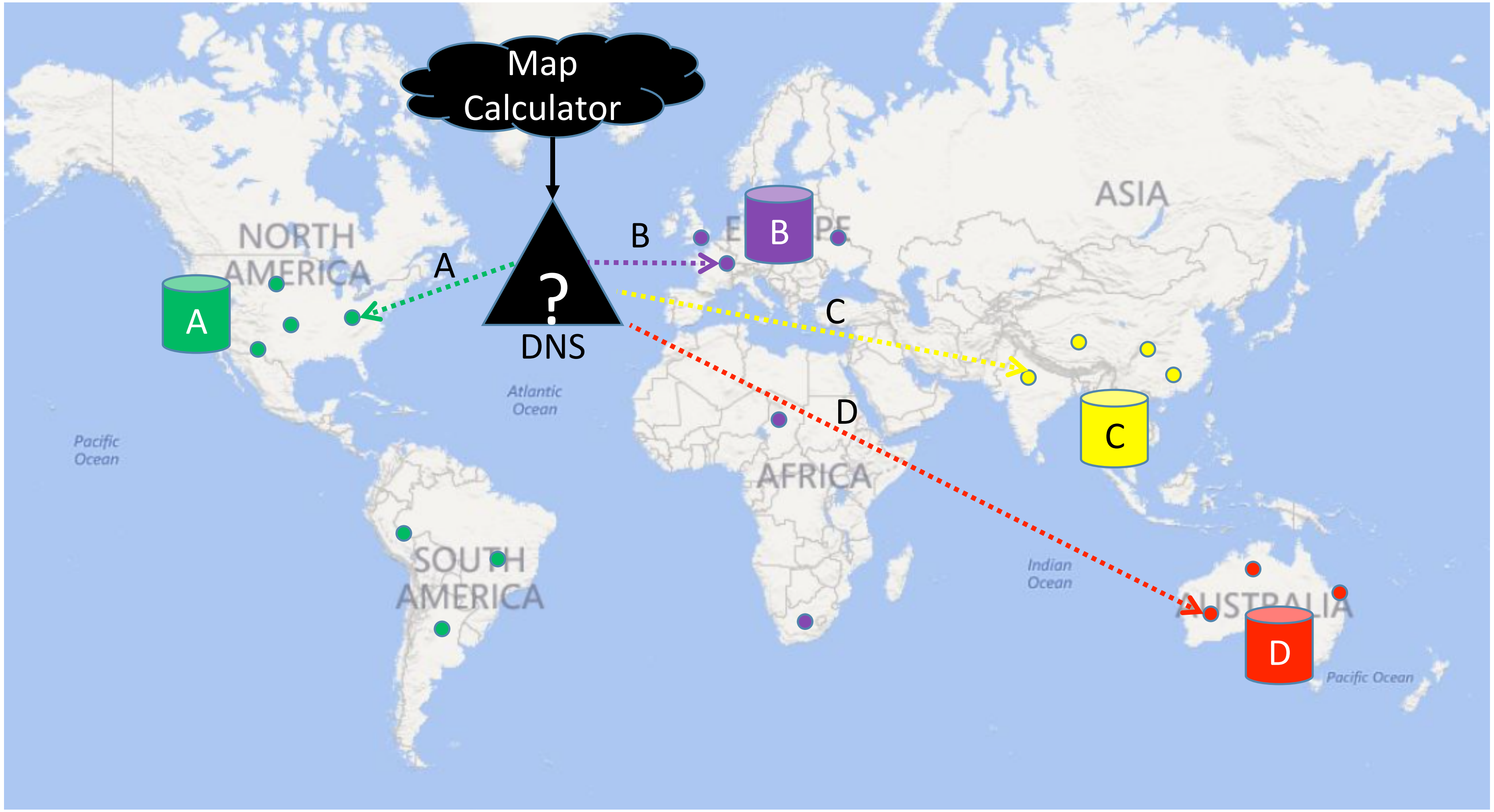
The “Map the Internet” Approach



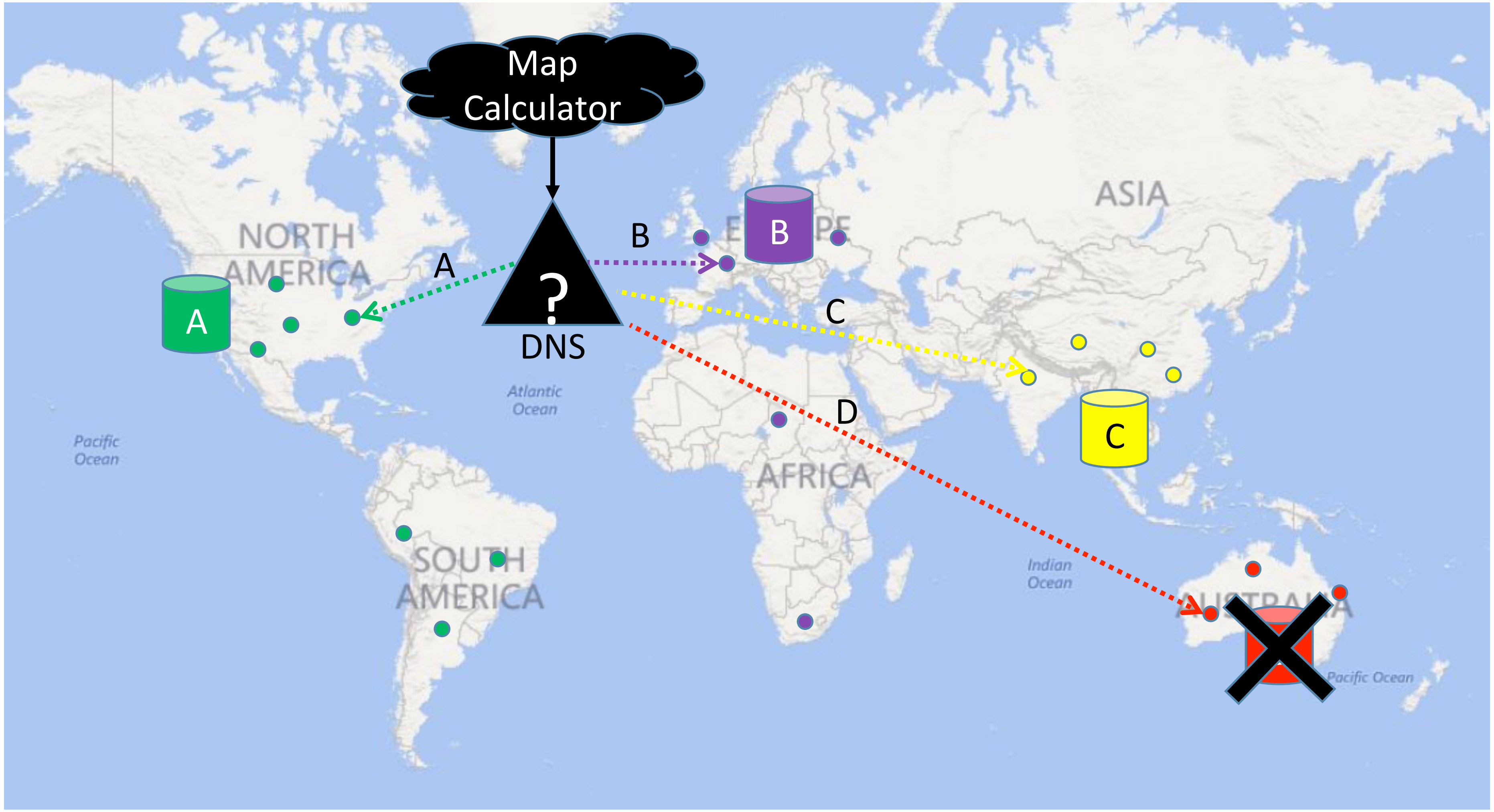
The “Map the Internet” Approach



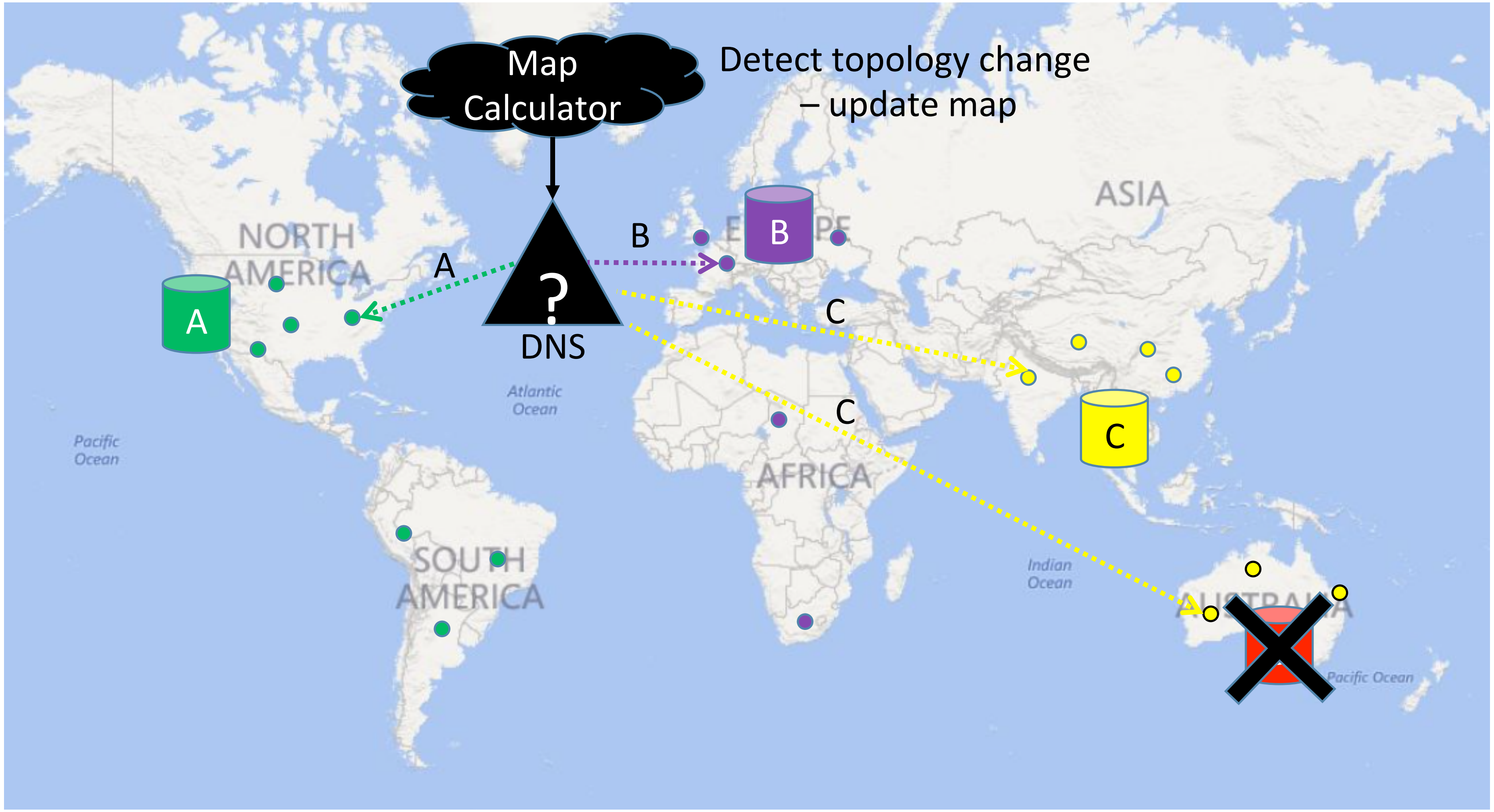
The “Map the Internet” Approach



The “Map the Internet” Approach



The “Map the Internet” Approach



The “Map the Internet” Approach

- Primary Benefit
  - Flexible Control: Can direct any DNS request to any node

- Trade off
  - High operational cost and complexity (Large scale central global co-ordinator required)
  - DNS can be inaccurate for client proximity routing
  - Availability requires very short TTLs

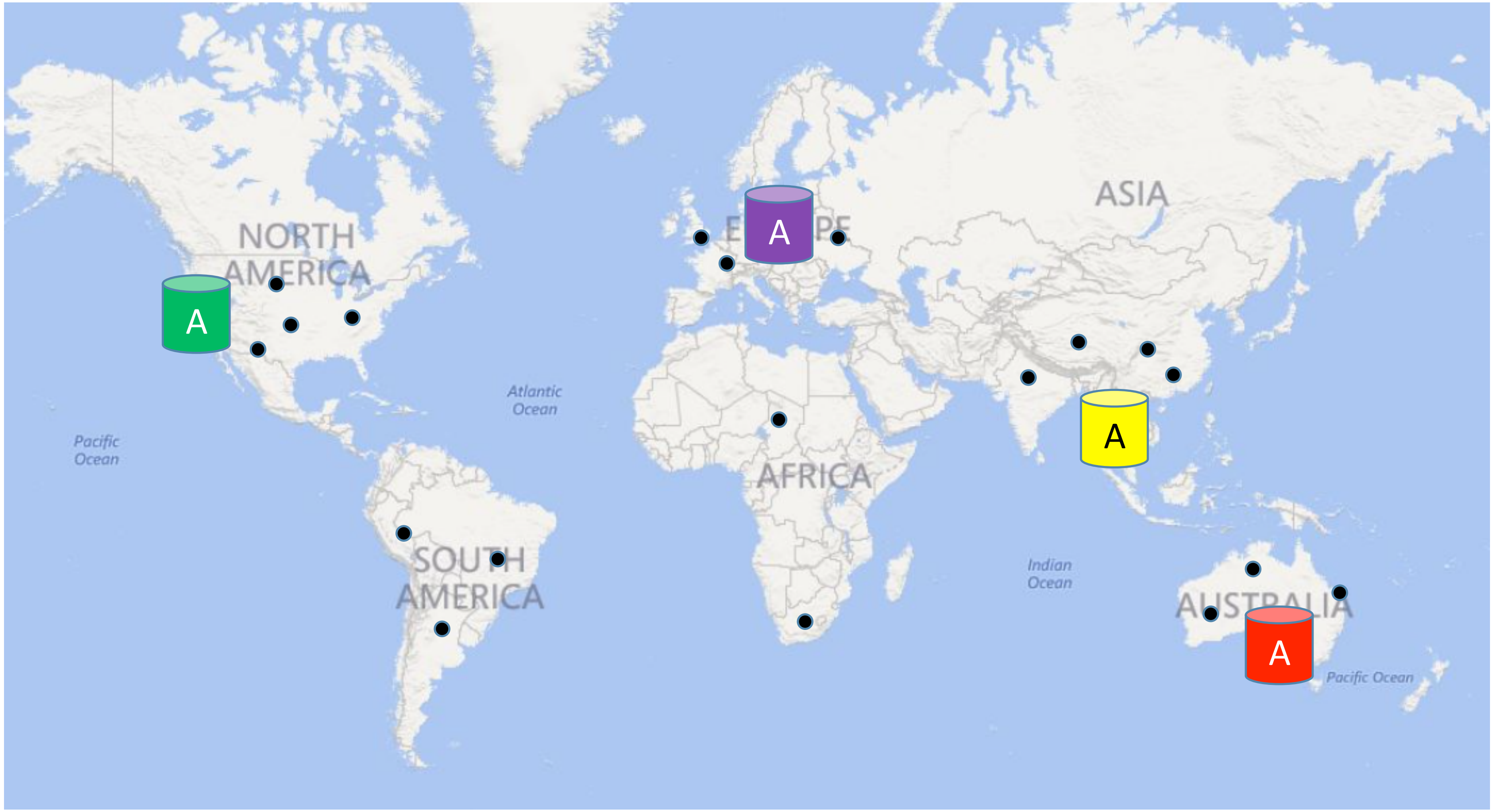
The “Map the Internet” Approach

- Primary Benefit
  - Flexible Control: Can direct any DNS request to any node

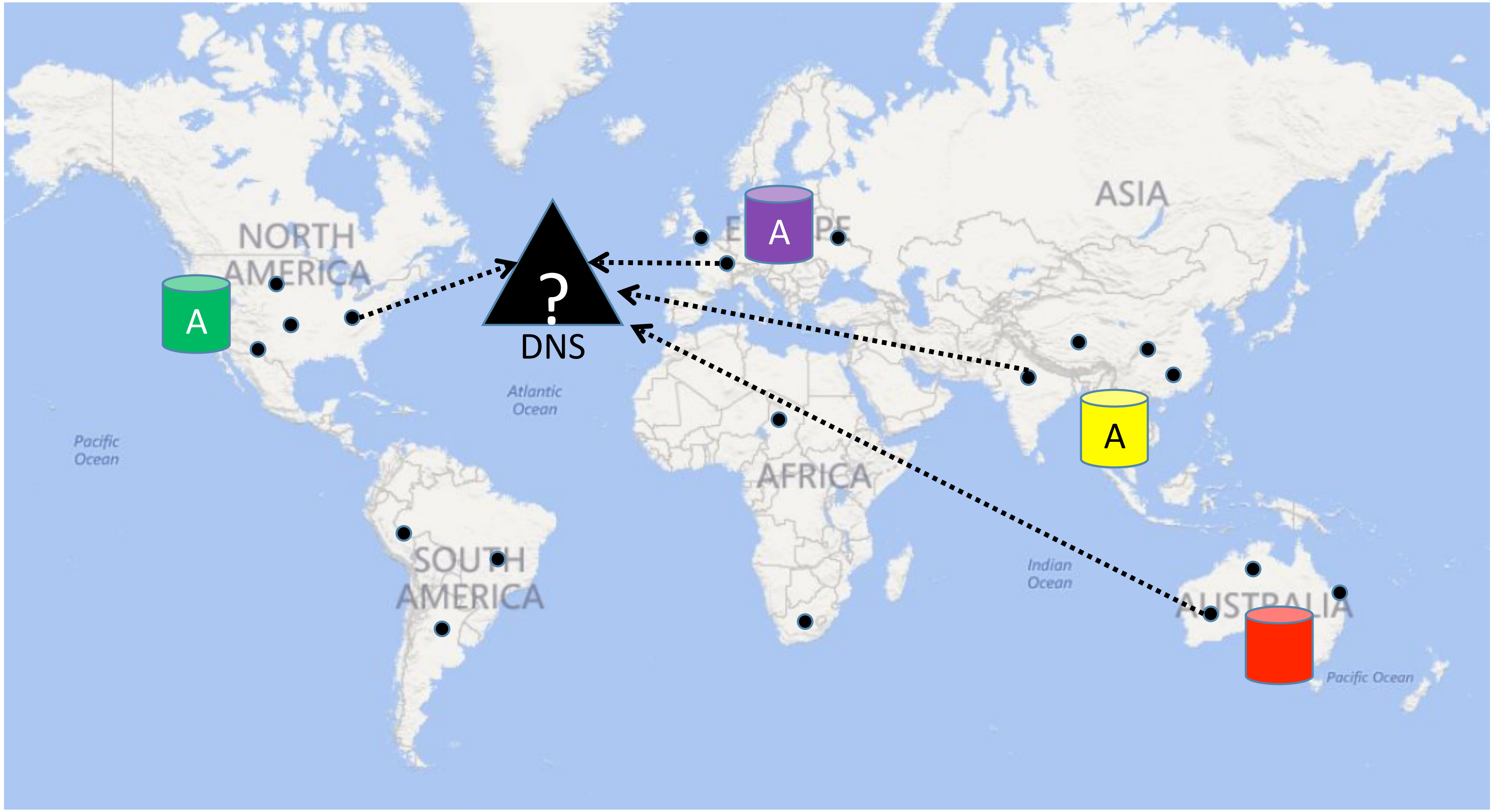
- Trade off
  - High operational cost and complexity (Large scale central global co-ordinator required)
  - DNS can be inaccurate for client proximity routing
  - Availability requires very short TTLs

- ***There is an alternative…***

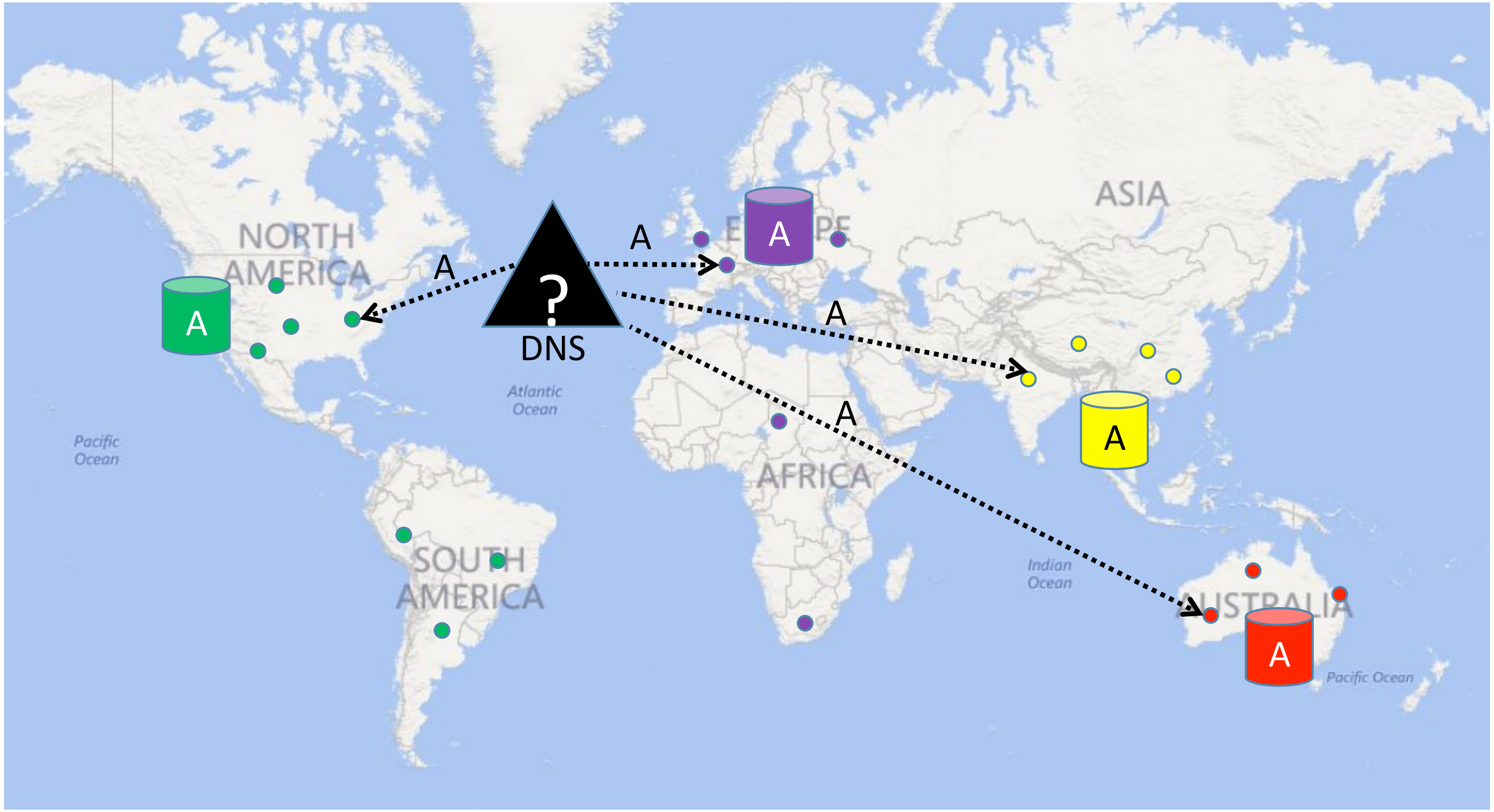
The Anycast Approach



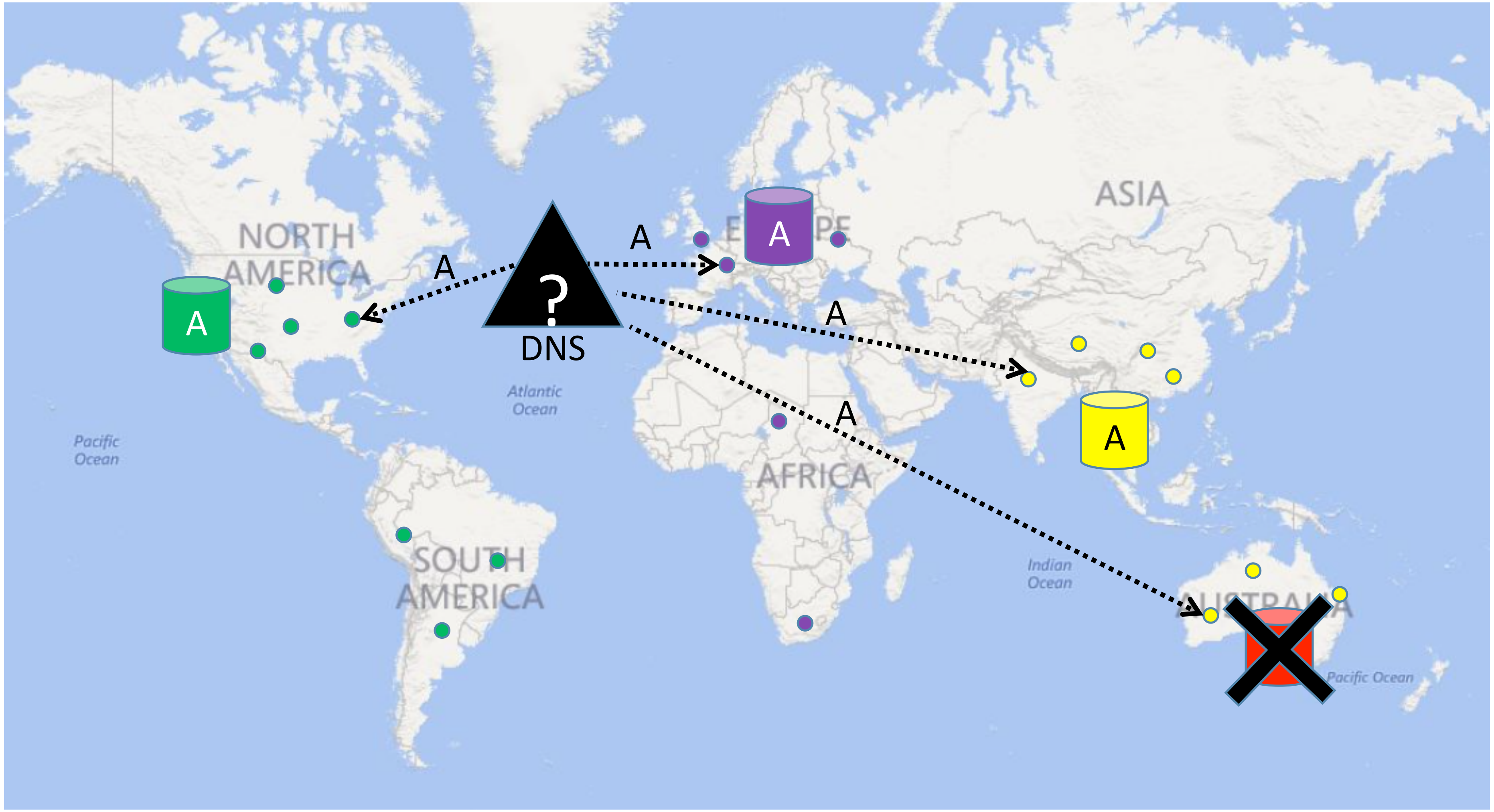
The Anycast Approach



The Anycast Approach



The Anycast Approach



The Anycast Approach

- Benefits
    - Simple, highly performant
    - Avoids DNS correlation issues
    - Fast reaction to changes (even with long TTL)

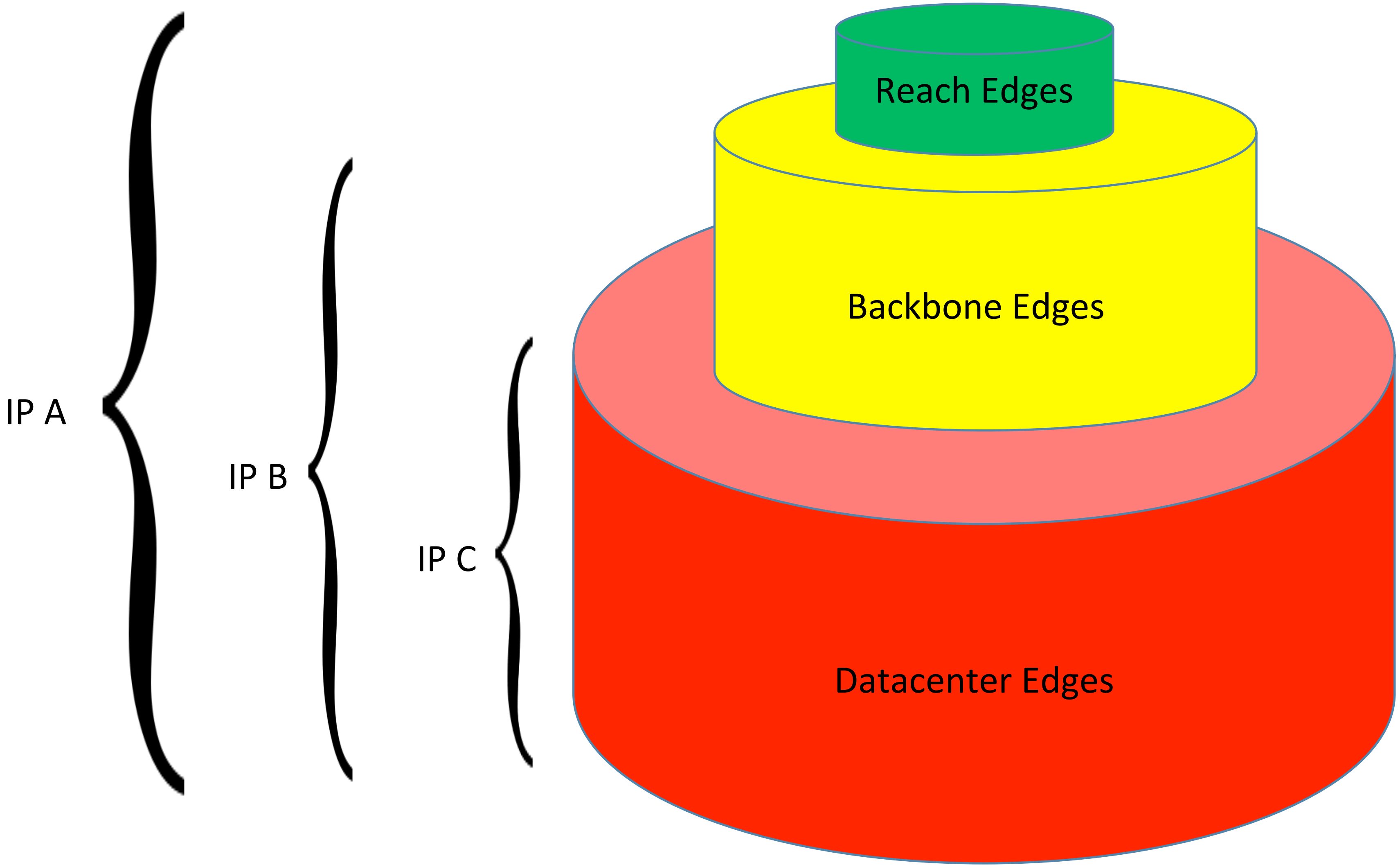
- Trade off
    - Relinquish routing control to “The Internet”
    - Have to size Edges based on organic traffic volume
    - Possibility of overload

FastRoute

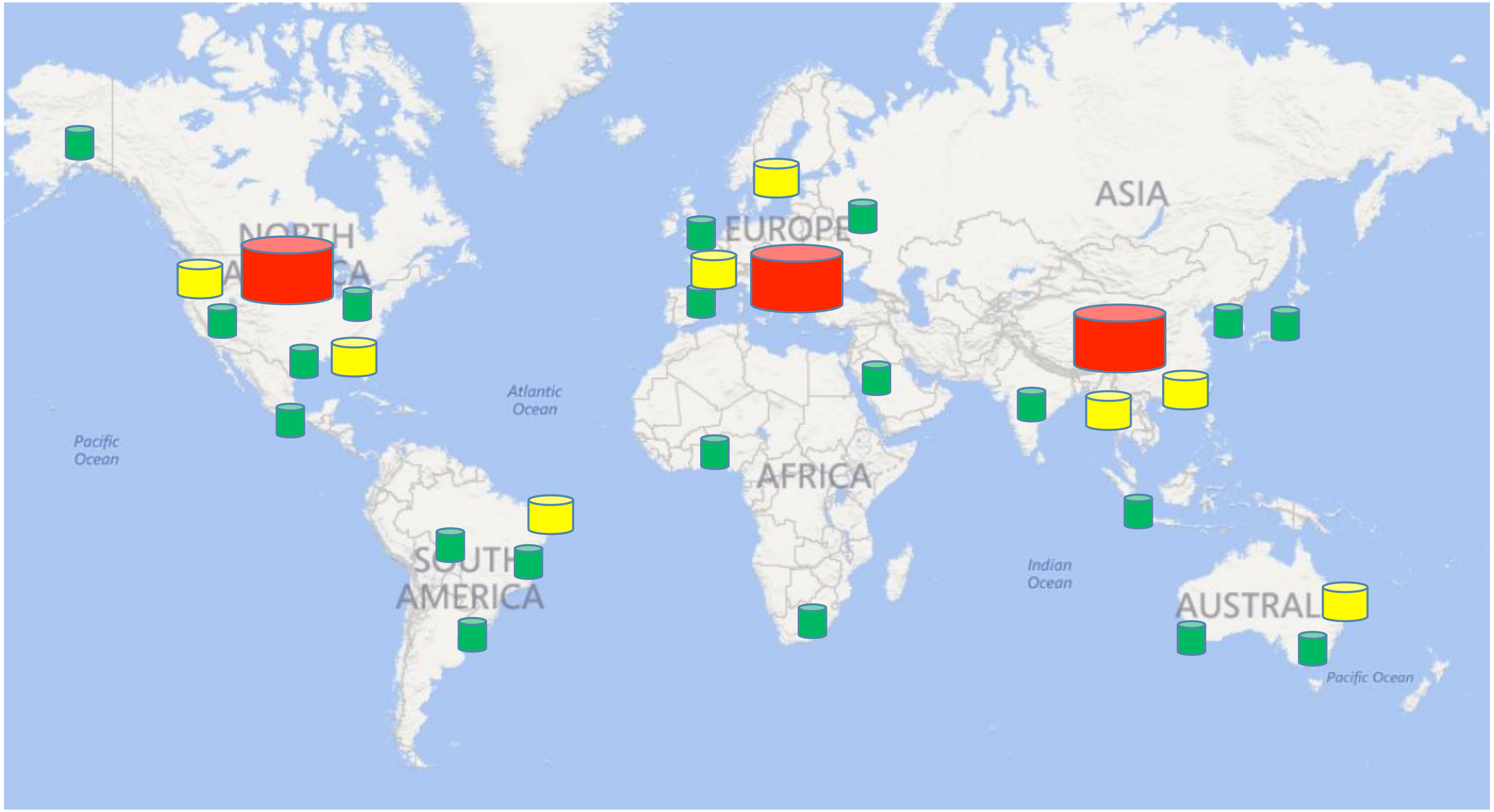
- Design Goals:
  - Simple (easy to operate)
  - Highly available (minimal downtime)
  - High Performance (better than existing solution)

- Desire:
  - A solution with the simplicity of Anycast, with ***just enough*** control to handle overloaded nodes.

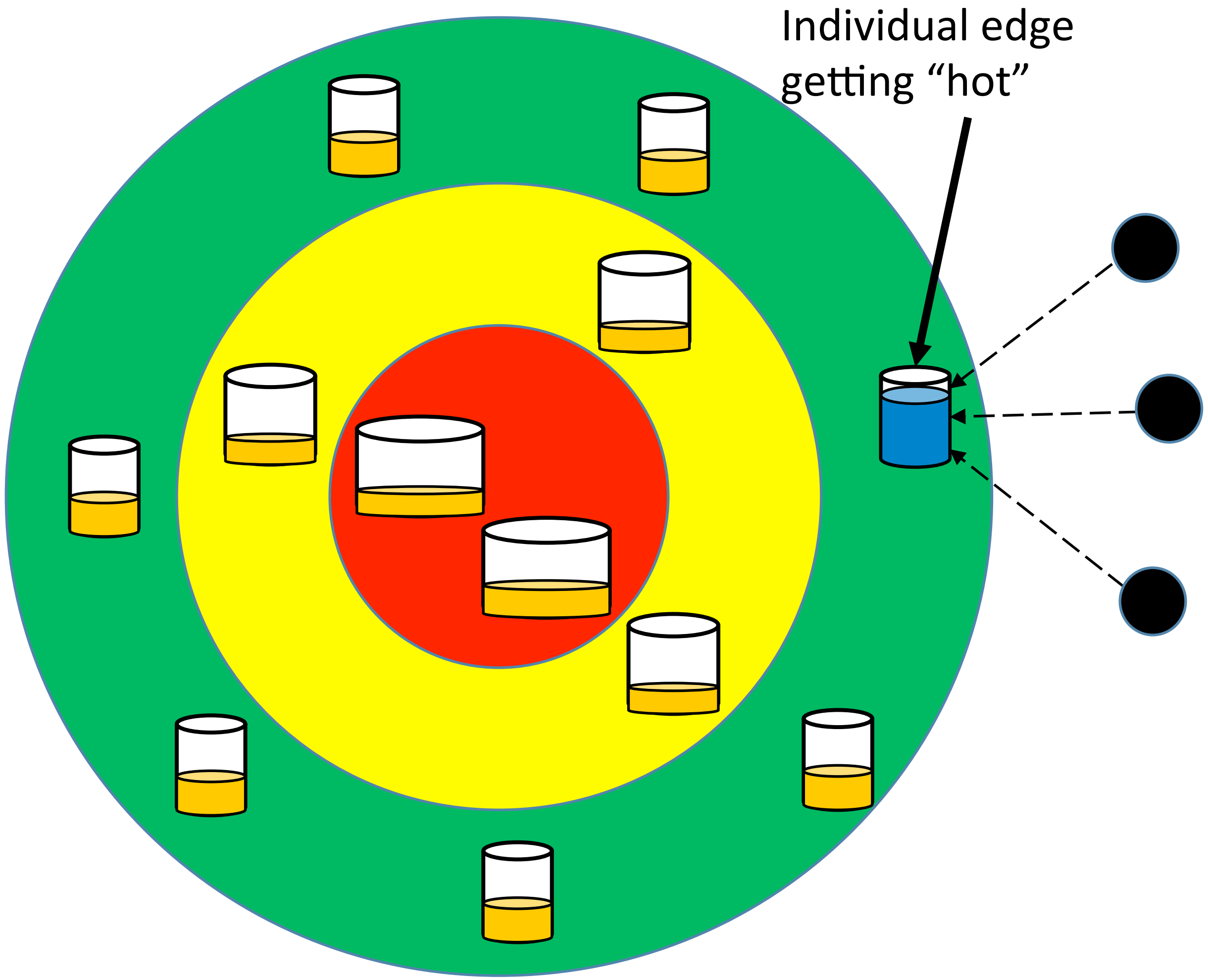
Utilizing Anycast “Layers”



Anycast “Layers”

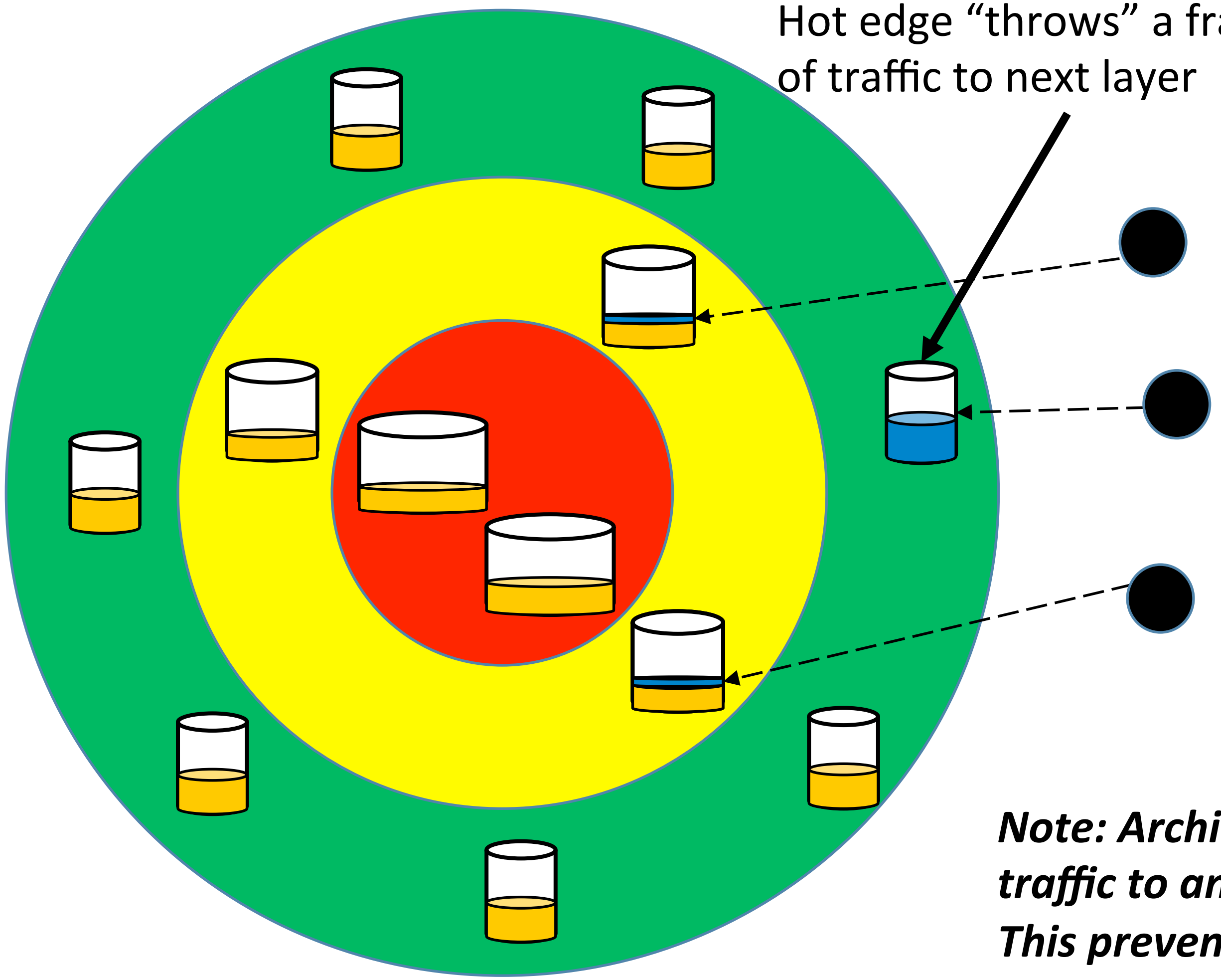


Load Management using Anycast Layers



Individual edge getting "hot"

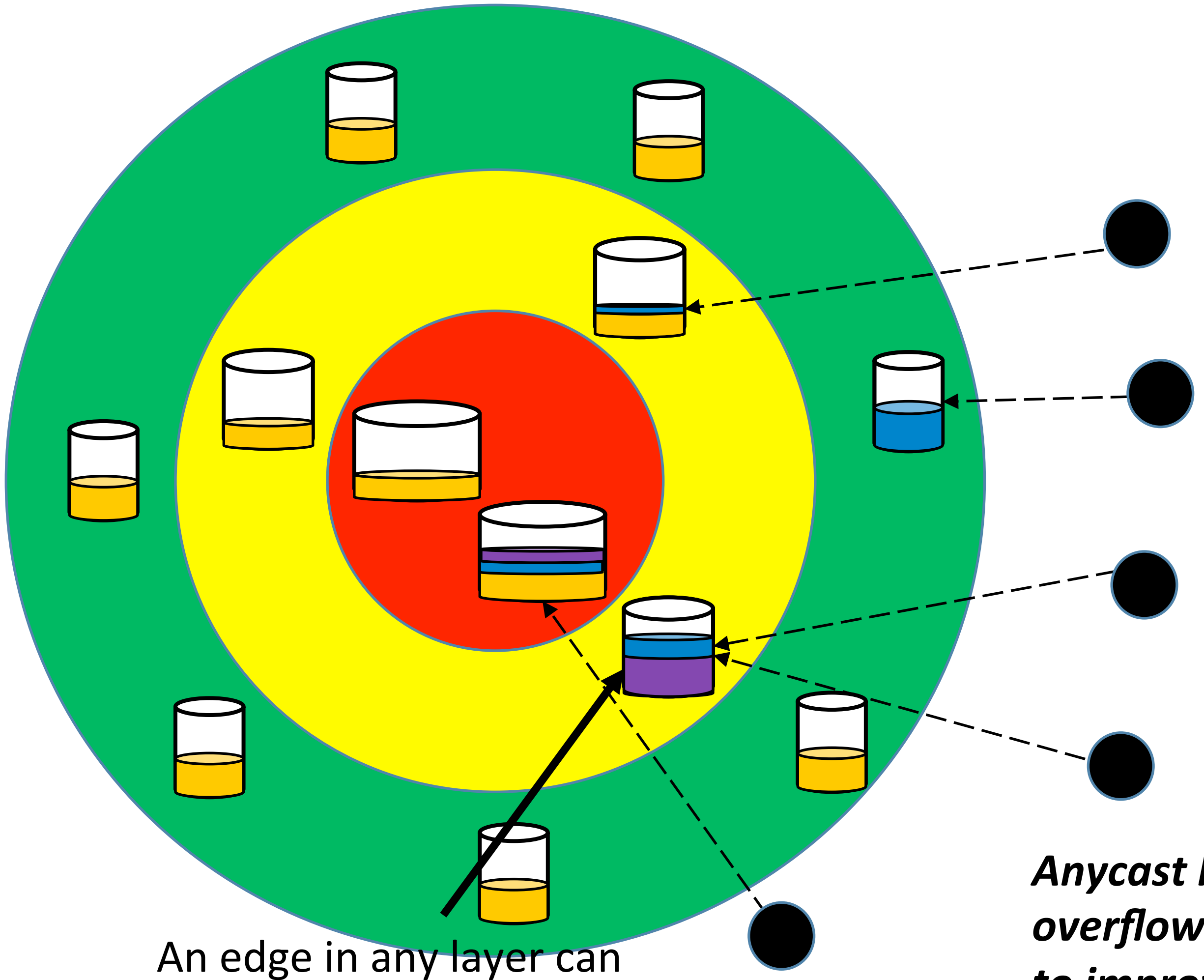
Load Management using Anycast Layers



Hot edge “throws” a fraction of traffic to next layer

Note: Architecture choice not to send traffic to another edge in same layer. This prevents oscillatory behavior.

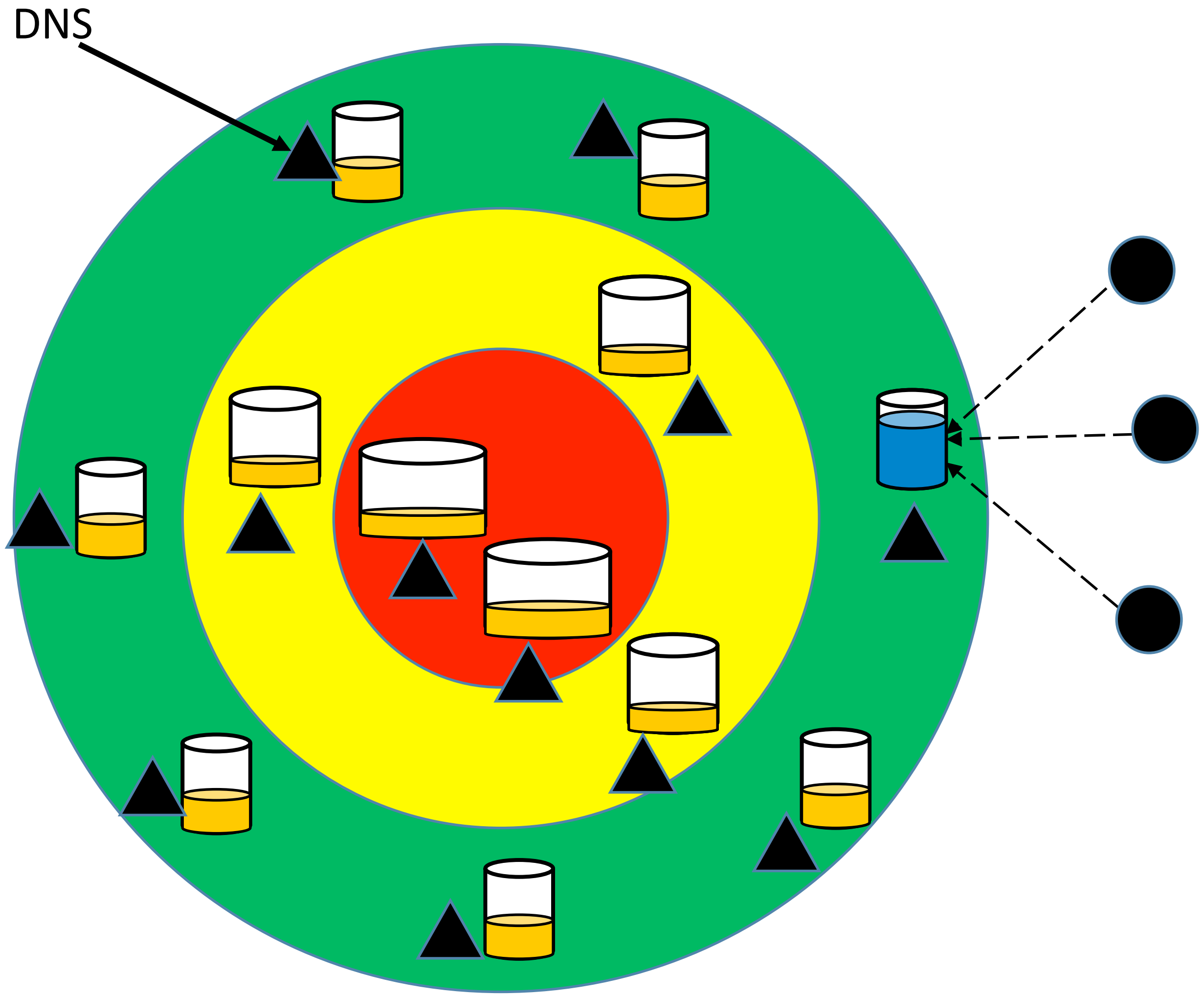
Load Management using Anycast Layers



An edge in any layer can "throw" to the next layer

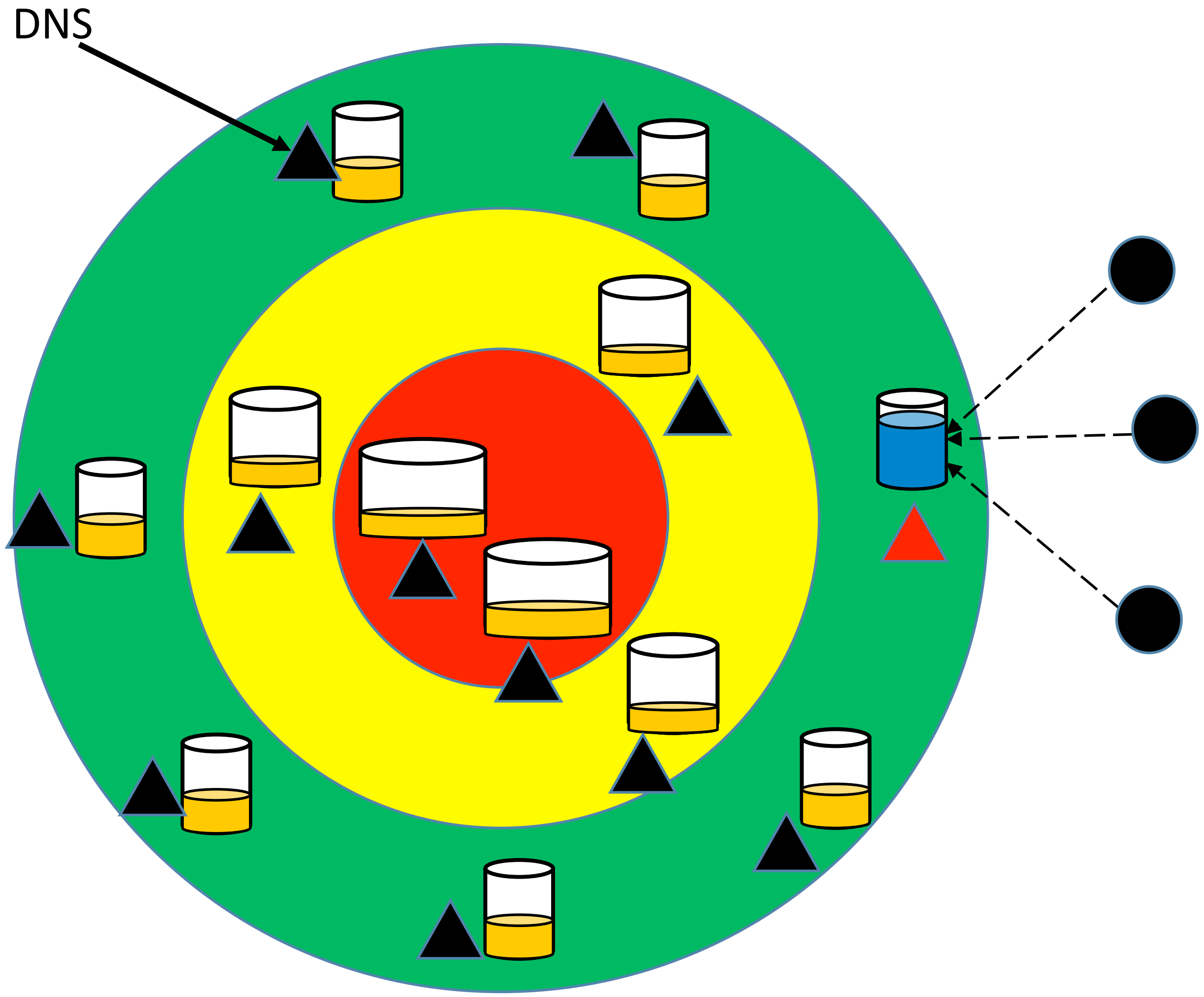
Anycast layer 0 is provisioned to absorb overflow. Further optimization can occur to improve absorption in this layer.

How to “throw” traffic to next layer?



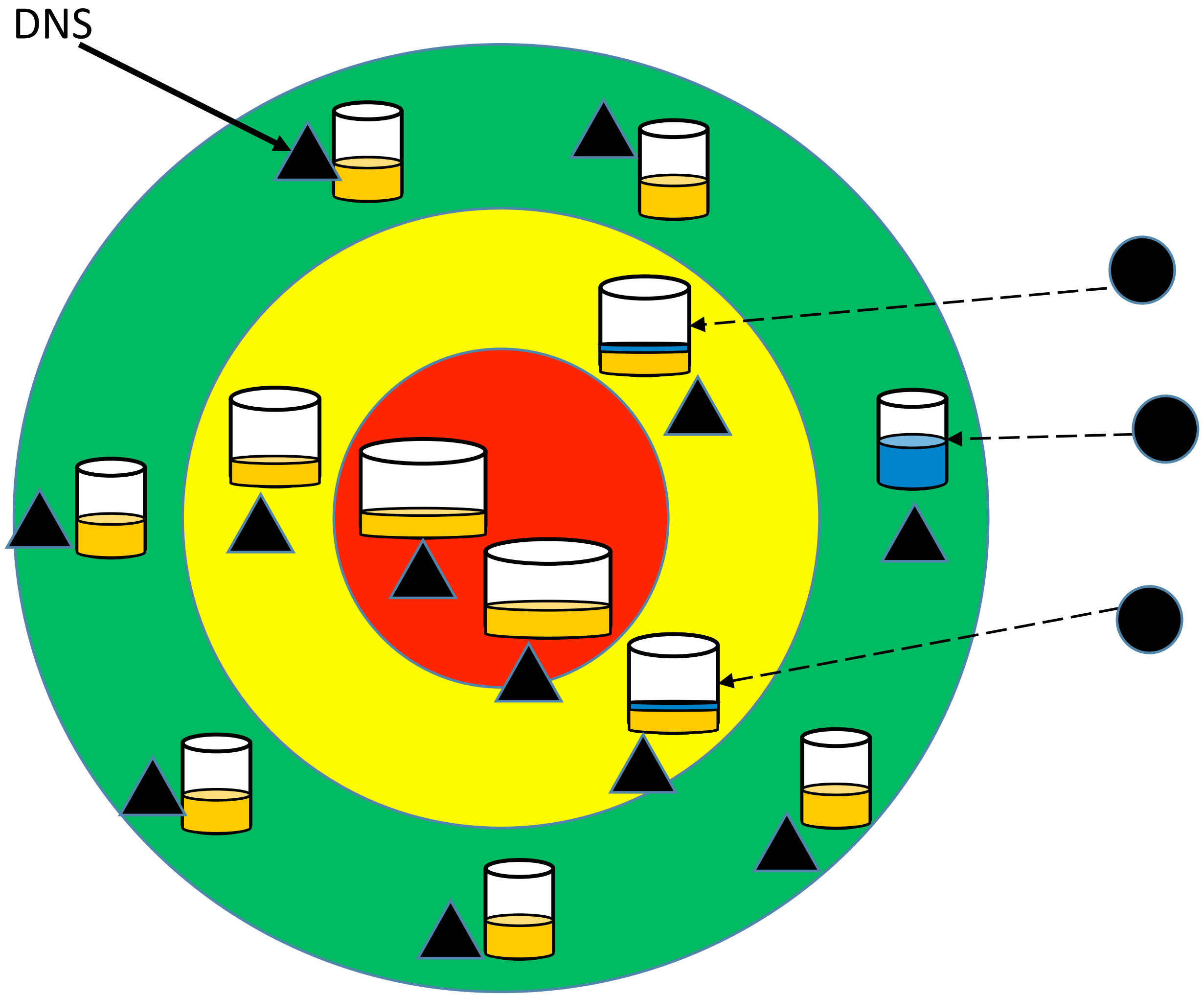
1. Co-locate DNS servers with HTTP proxies in every location

How to “throw” traffic to next layer?



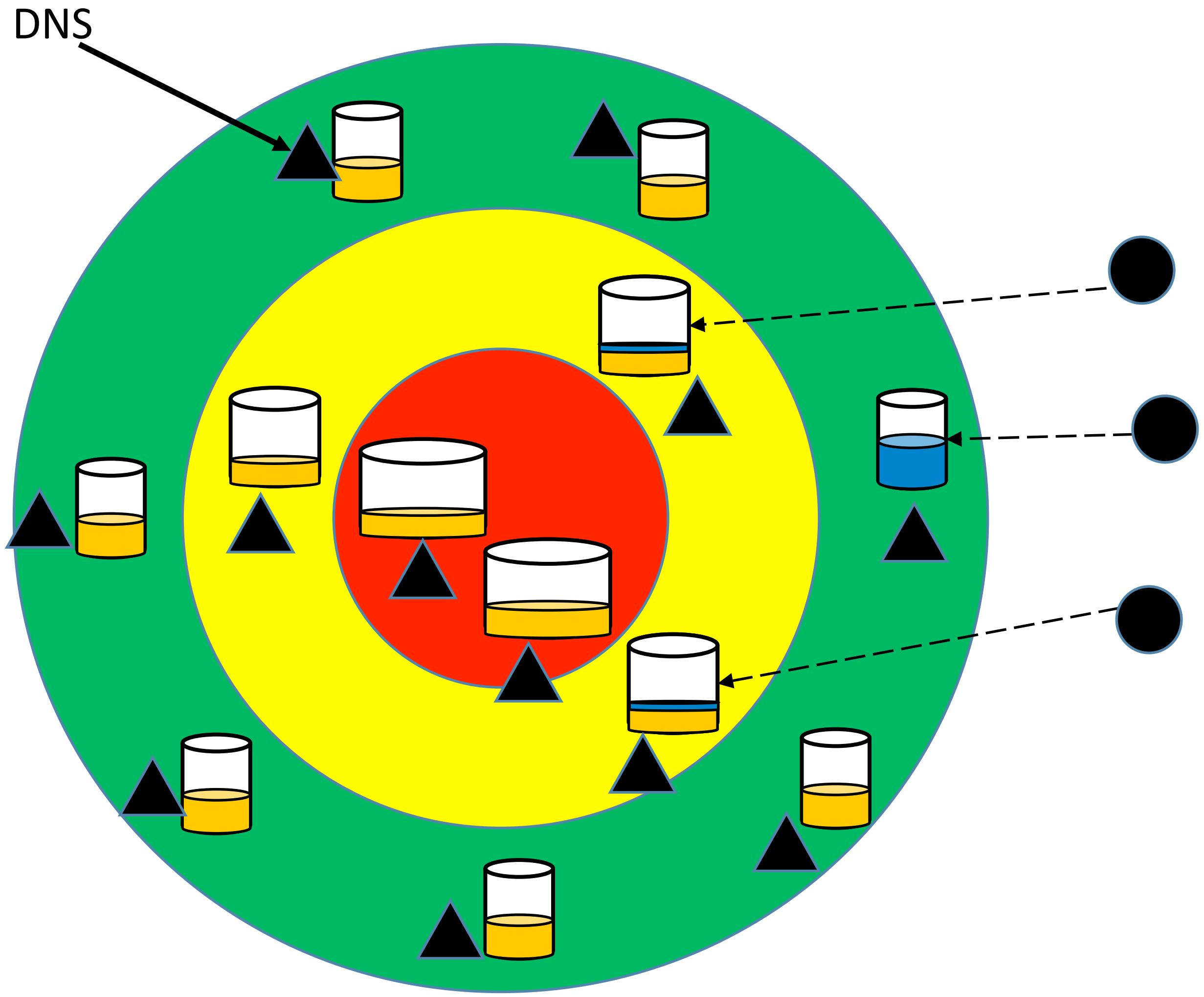
1. Co-locate DNS servers with HTTP proxies in every location
2. DNS monitors load in its own location

How to “throw” traffic to next layer?



1. Co-locate DNS servers with HTTP proxies in every location
2. DNS monitors load in its own location
3. DNS probabilistically returns a CNAME (DNS redirection) to next layer

How to "throw" traffic to next layer?



1. Co-locate DNS servers with HTTP proxies in every location
2. DNS monitors load in its own location
3. DNS probabilistically returns a CNAME (DNS redirection) to next layer

Preserves the independence of each node (no real-time communication outside a node).

How to “throw” traffic to next layer?

- Major assumption
    - DNS request for a user lands in the same location as HTTP request (i.e. self-correlated)

How to “throw” traffic to next layer?

- Major assumption
  - DNS request for a user lands in the same location as HTTP request (i.e. self-correlated)

- This is not guaranteed for all requests.

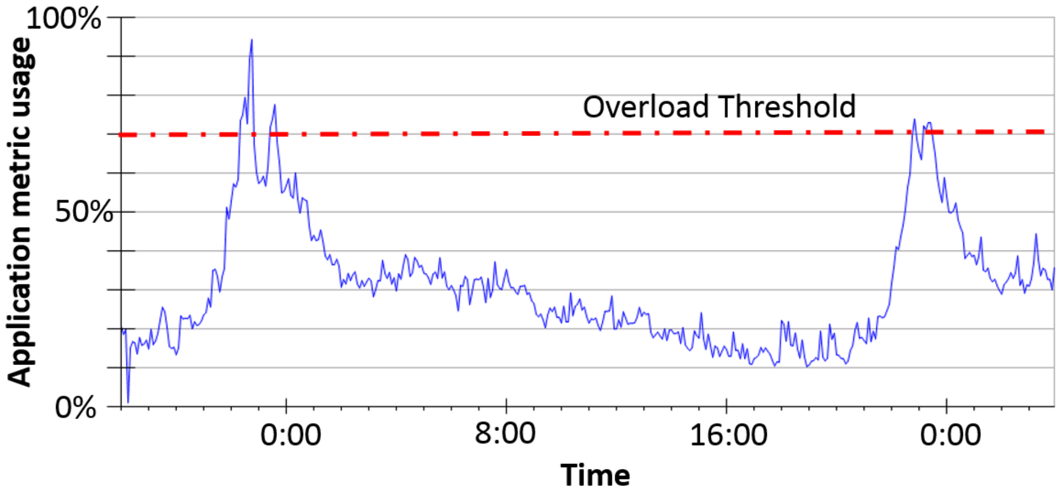
How to “throw” traffic to next layer?

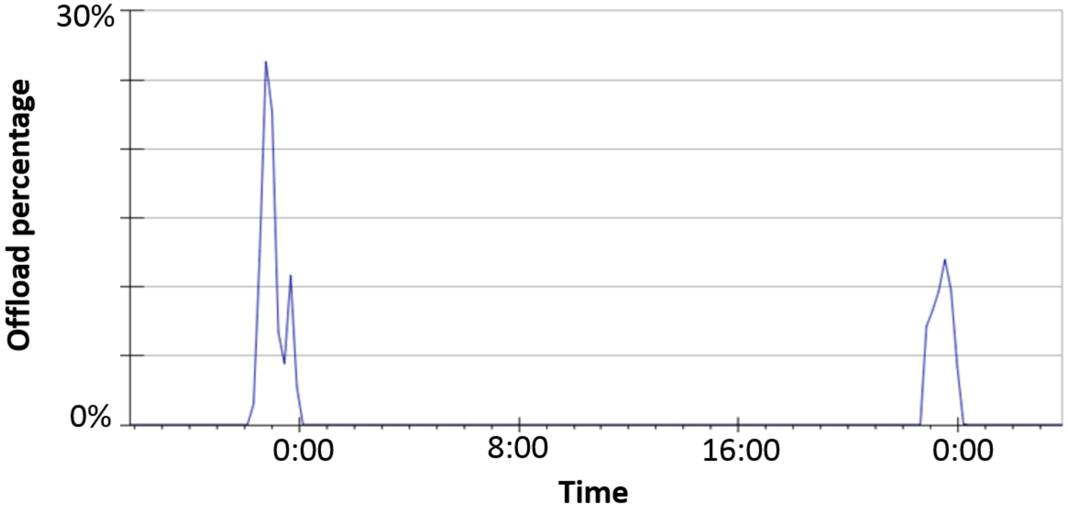
- Major assumption
  - DNS request for a user lands in the same location as HTTP request (i.e. self-correlated)

- This is not guaranteed for all requests.

- Is it good enough?
  - Yes – we see around 80% correlation
  - You only need to shed the percentage of overload

DNS Load Management In Practice





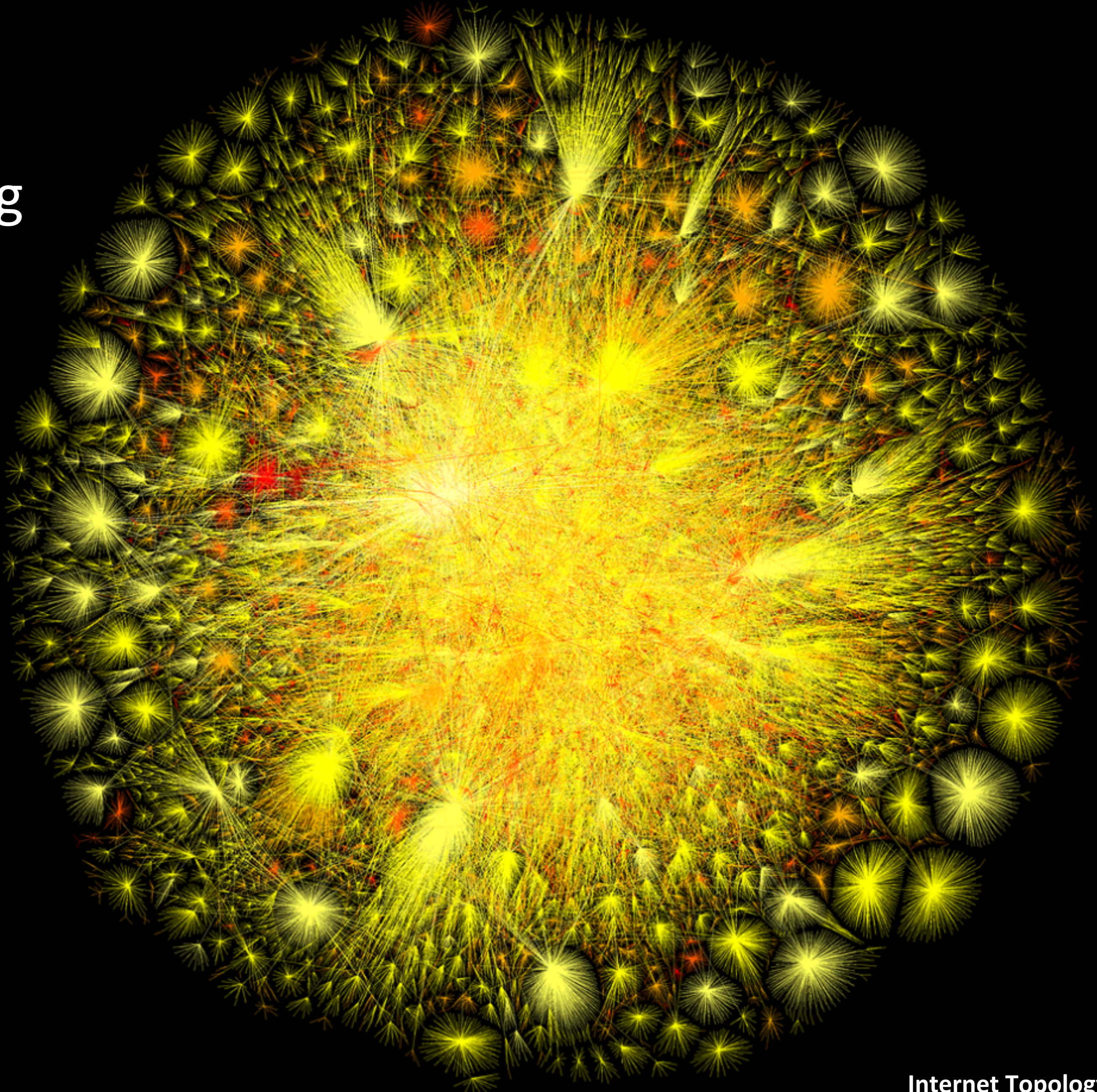
Architecture Summary

- Statically configure edges in multiple Anycast layers
- Each edge ***independently*** monitors its own load and decides whether to "throw" traffic to the next layer.
- Final layer is dimensioned sufficiently to handle all load
- *Edge nodes act independently without any knowledge outside the edge.*

- ***Maximum Anycast benefit requires collaboration to have traffic ingress proximal to eyeballs***

FastRoute:
A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs

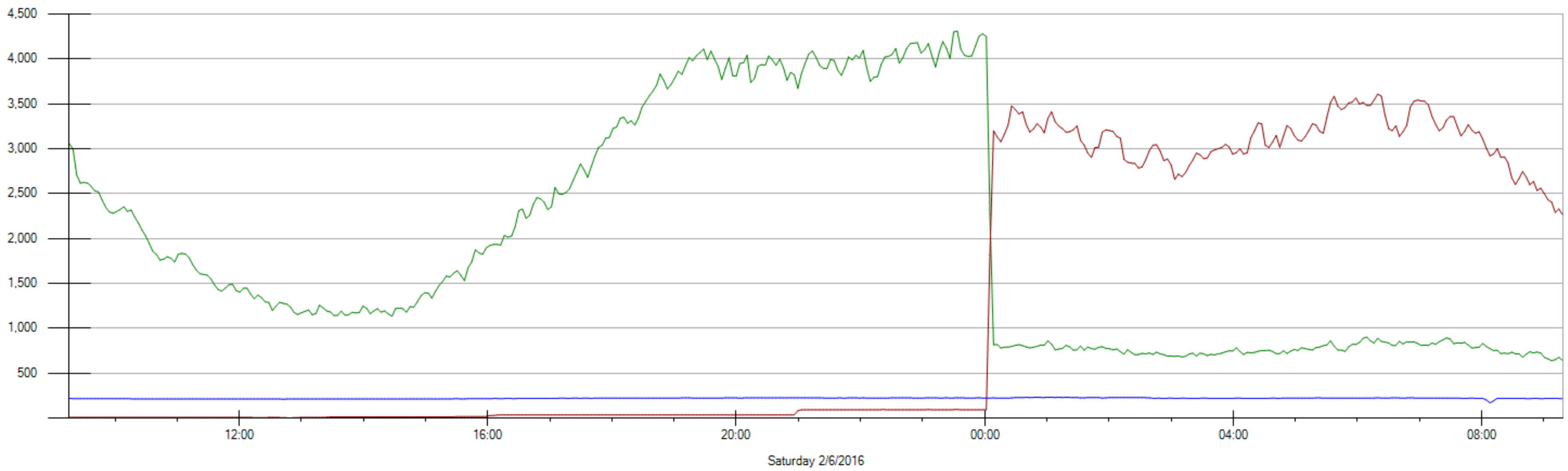
Questions?



Internet Topology
image courtesy of
www.opte.org

Traffic Shifts

- Traffic shifts immediately when nodes come online
- No BGP route shift seen externally to our network
- Operational simplicity – key for scale-out



Anycast Problems

Issues are all with inbound routing leading to Inconsistent entry point

- AMS-IX peering, LINX route server
- Alternating packets sent to each peering point
- No TCP connect possible
  - detected by tcpdump on 2 machines and seeing SYN and ACK land in different locations
- was customer firewall config issue
  - mitigation is to give out unicast, only some IPs in the /24 worked, changing source IP also worked

Anycast Problems (2)

- Route Flapping
- Not an issue in our case (global backbone, single ASN with consistent advertisements to peering)
- An edge withdrawing the route will only change internal routing which is full mesh and fast convergence – route as seen by the internet is stable
- Trajectory is good
  - Each new peer we pick up hears the routes directly
  - ISP based nodes pull local traffic – not leaked to transit

Long lived connections

- Currently serving
  - OS Updates
  - OS Images
  - Game Downloads
- Helped by modern download apps which retry ranges
- Important to RST when packets received on unestablished session (win default was “stealth mode”)

Performance Tuning

- DO also need investment in monitoring of inbound traffic patterns
- Interestingly the ingress point is revealed (cheap inbound tracert)

- Monitor and investigate P75 RTT changes (asn/city level)
- Monitor geo proximity of clients to ingress point

- Need collaboration with ISP community to deliver traffic to peering point closest to eyeballs
- Response will be served from same location
- Strong reduction in asymmetrical routing