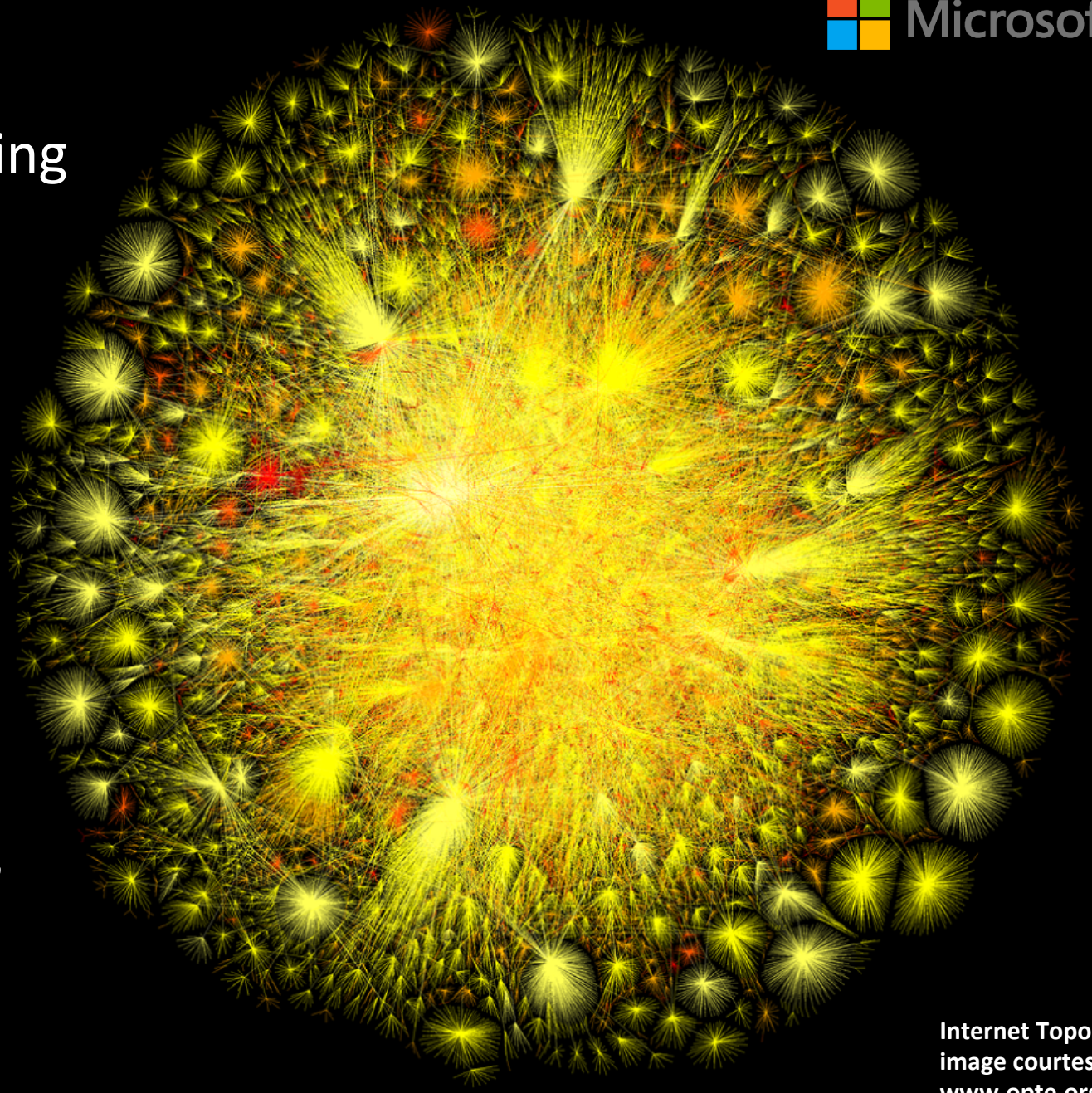
FastRoute:
A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs

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

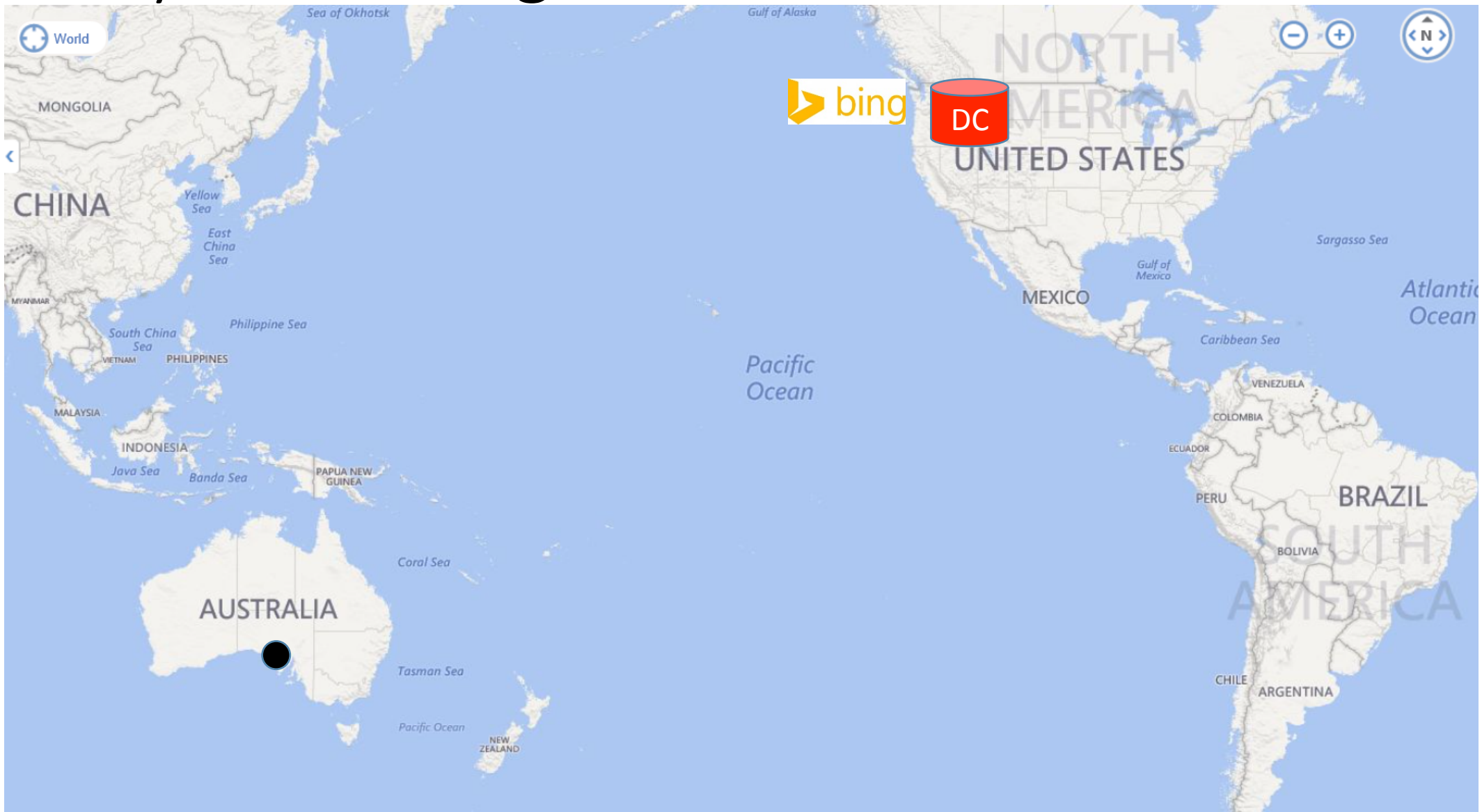


Internet Topology image courtesy of www.opte.org

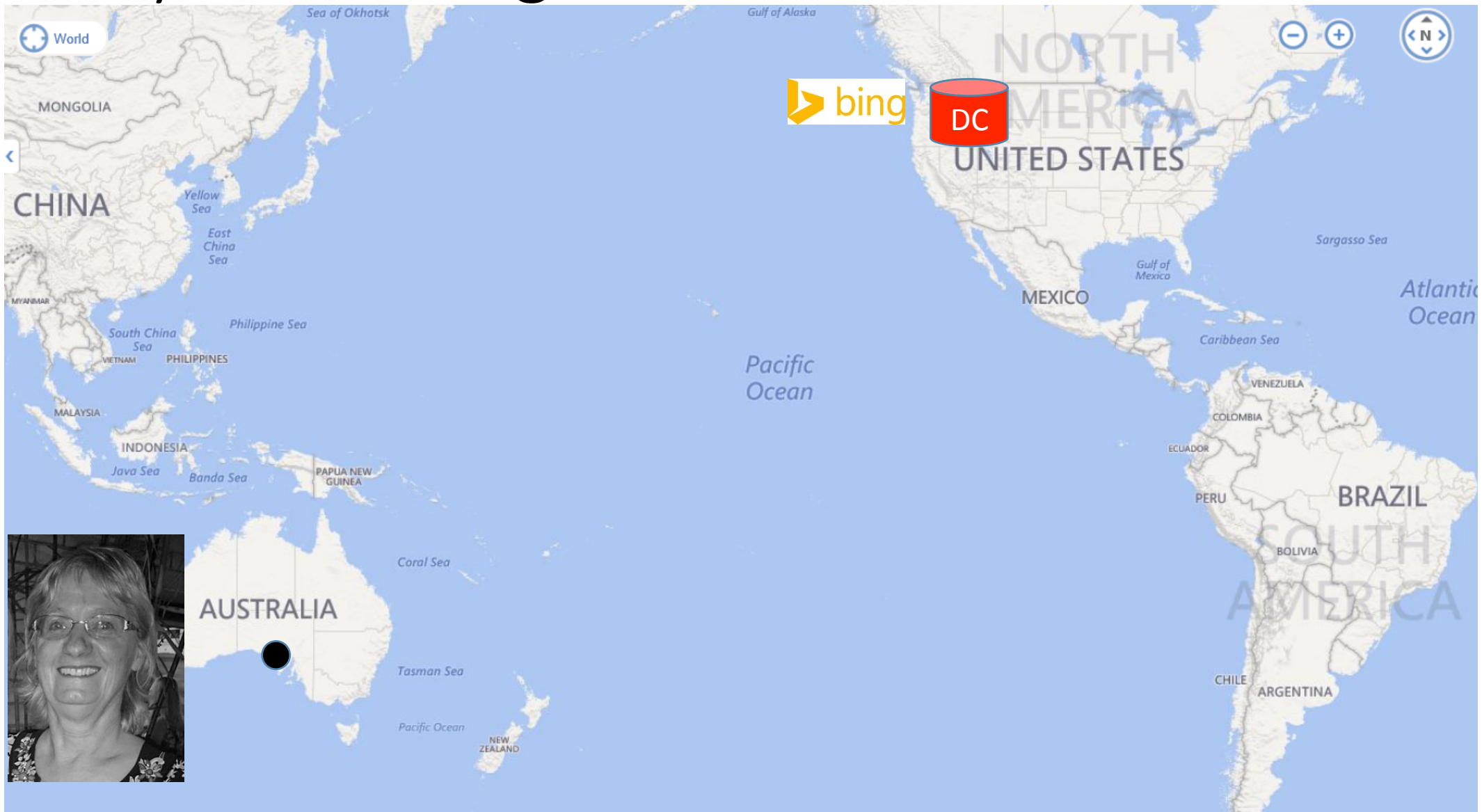
FastRoute Overview

- Microsoft's online services exist inside small set of datacenters distributed throughout the world.
- "Edge" nodes distributed throughout the Internet reduce network latency of such services.
- FastRoute is the fully distributed mechanism used to direct users to nearby edge.

Why use an Edge?

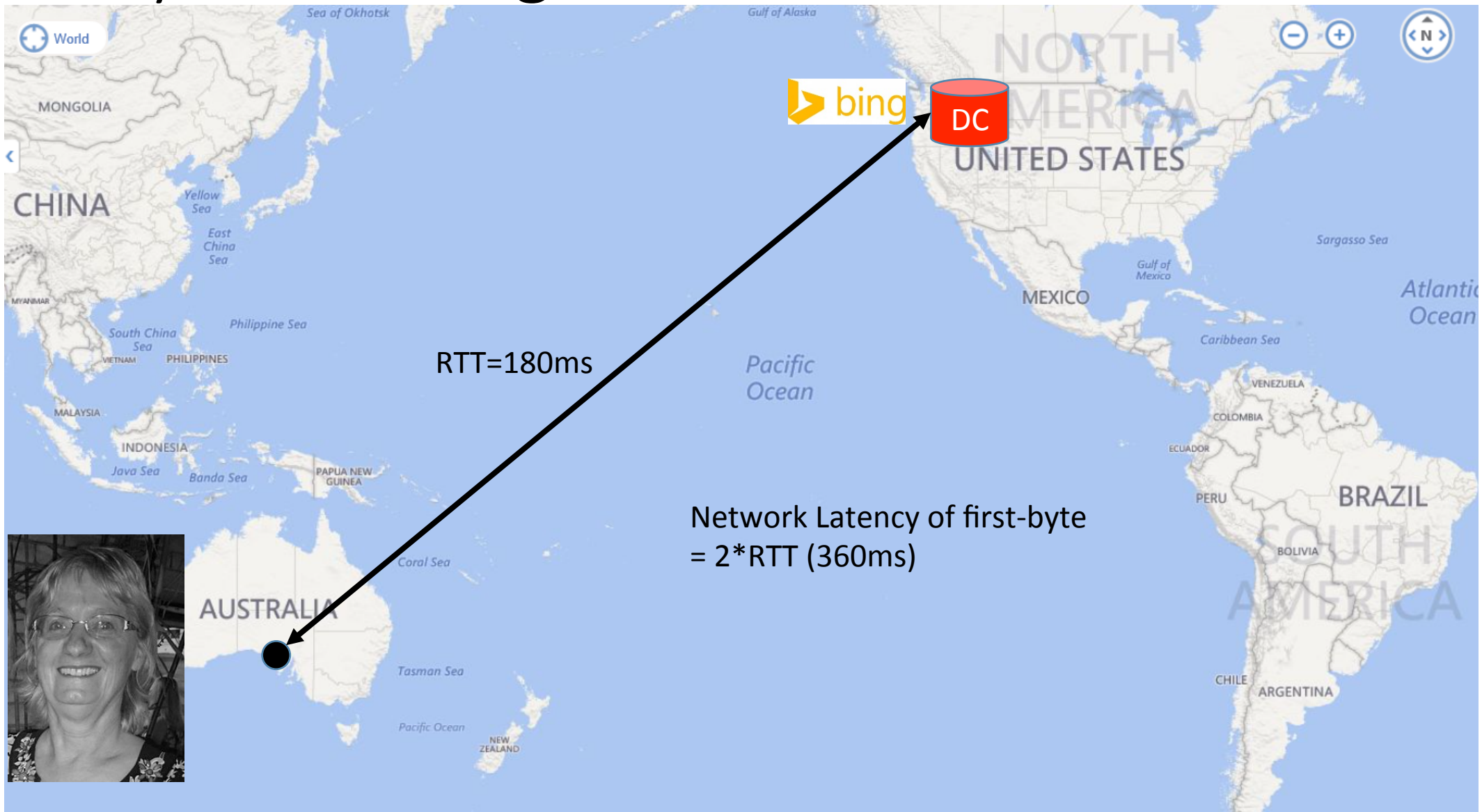


Why use an Edge?

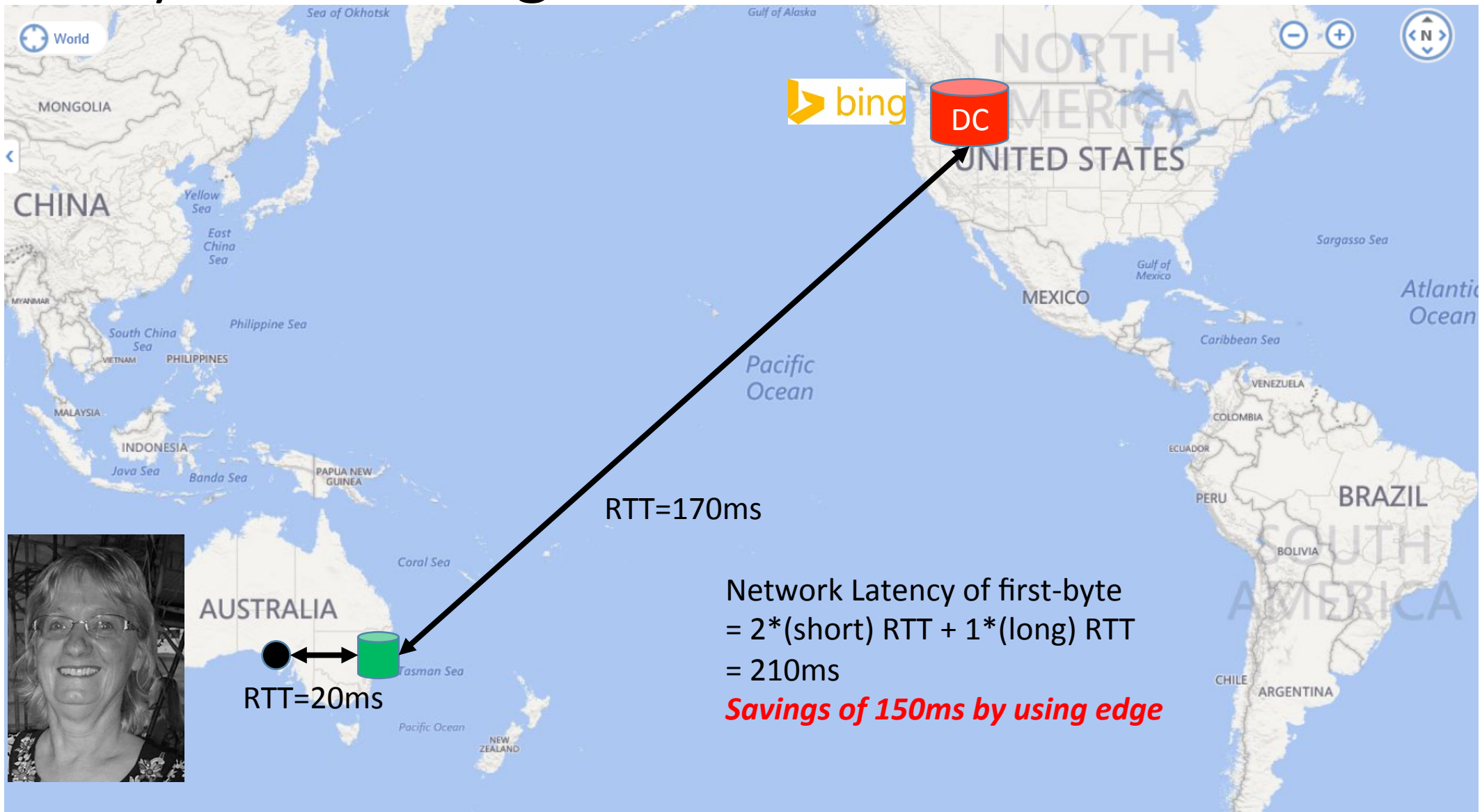


Mum

Why use an Edge?



Why use an Edge?



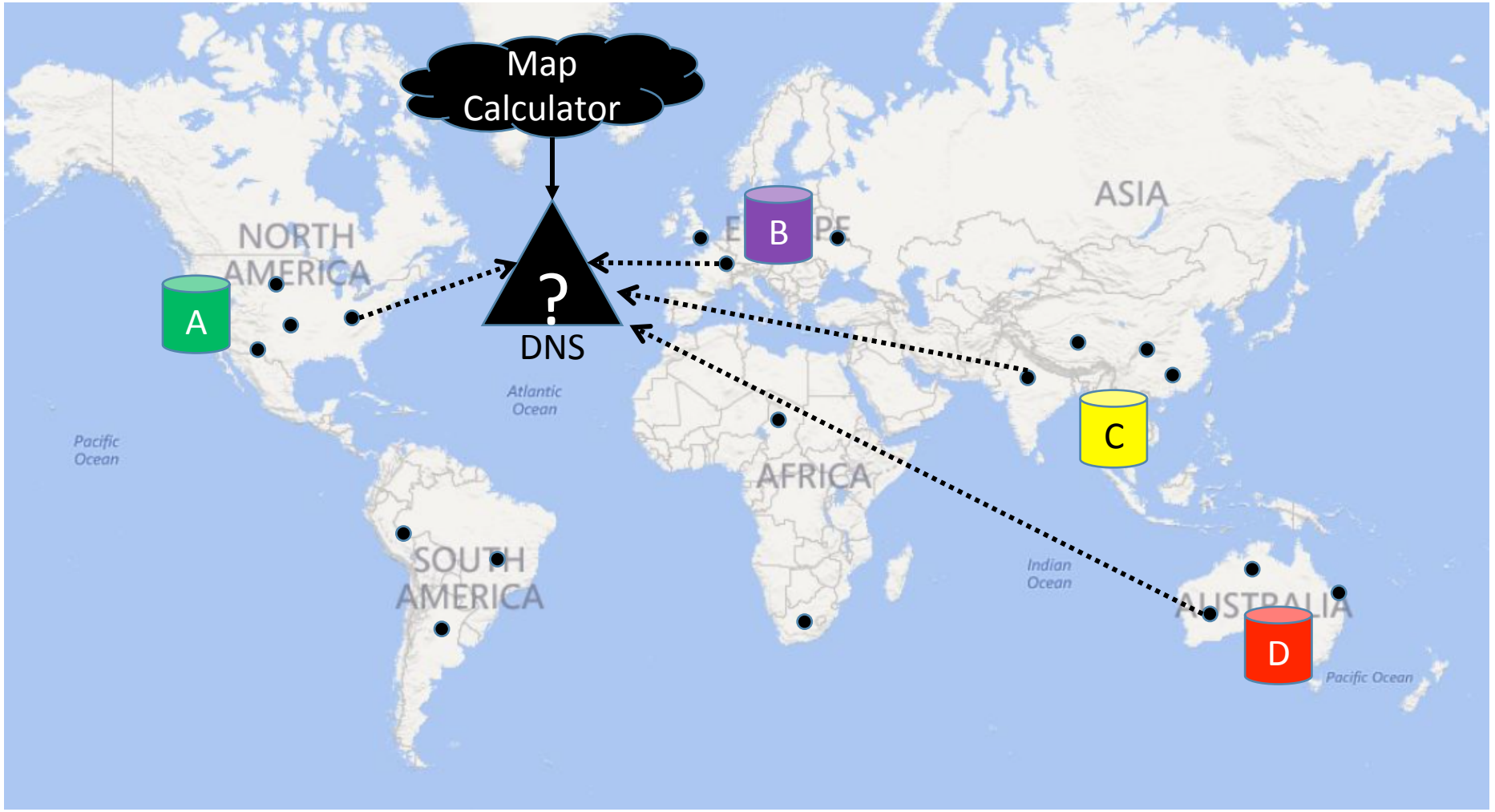
Which edge for which user?

- Which edge do I direct users to?
- How do I direct users to the right edge?

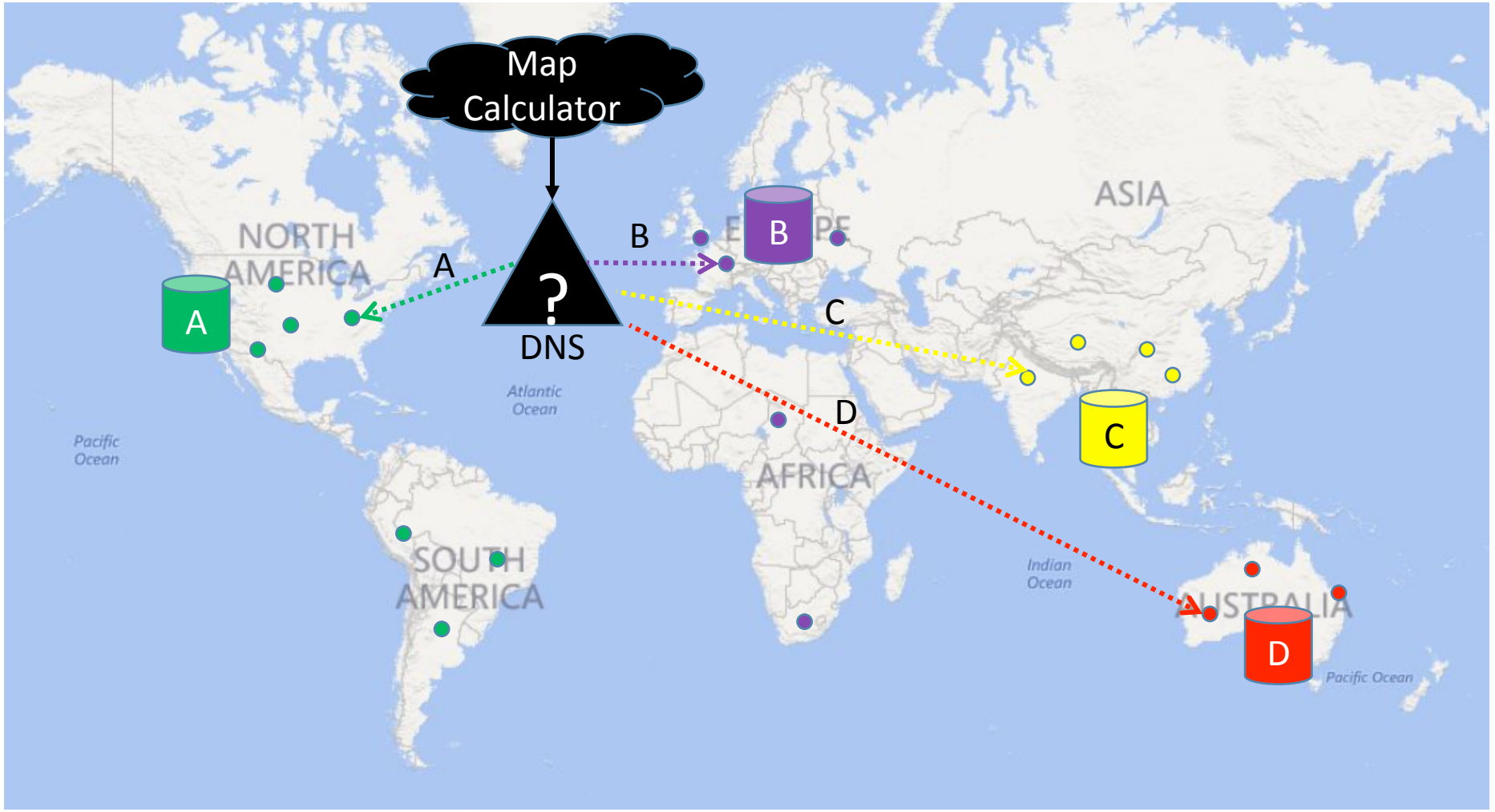
The “Map the Internet” Approach

The “Map the Internet” Approach

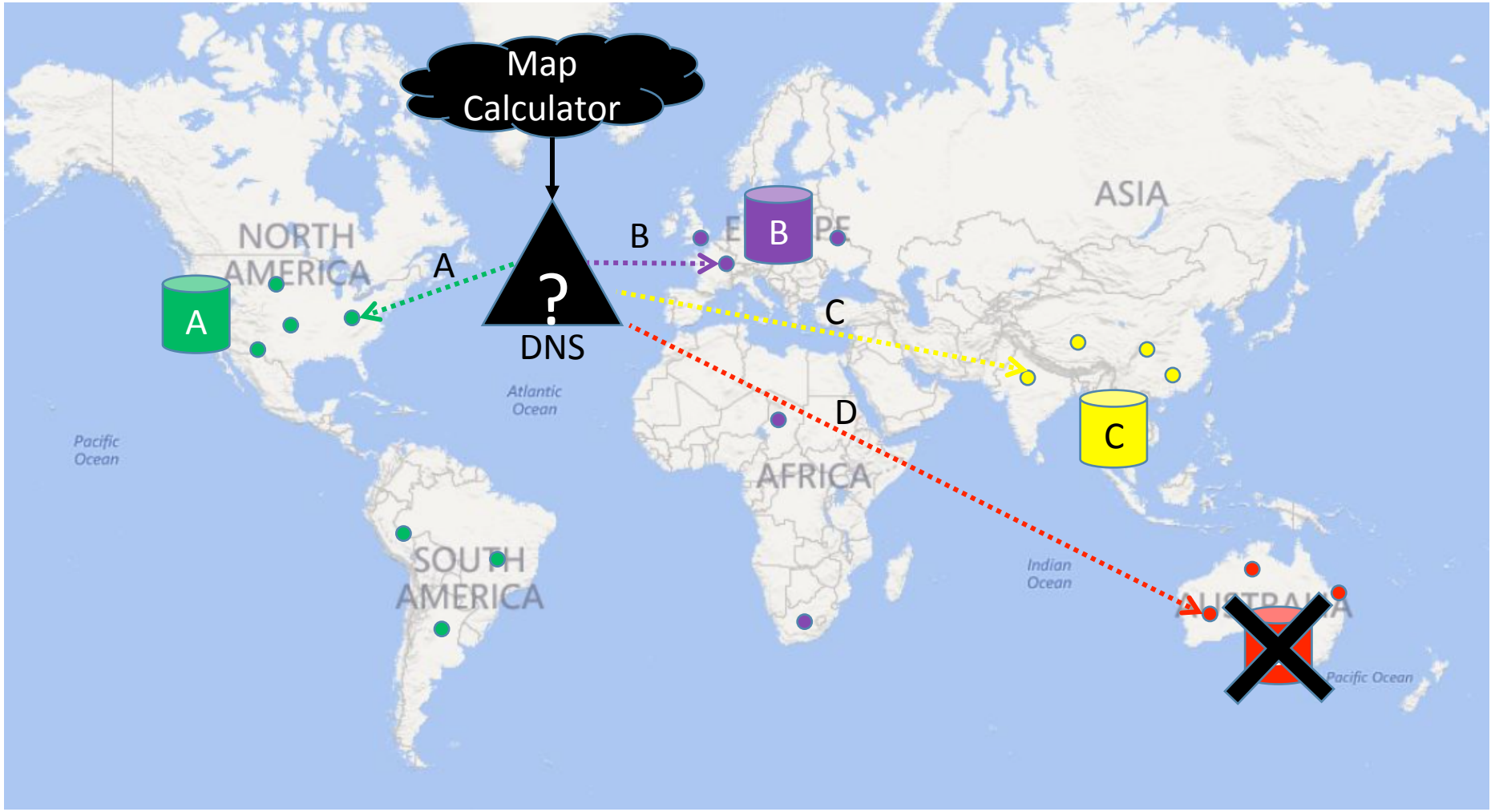
The “Map the Internet” Approach



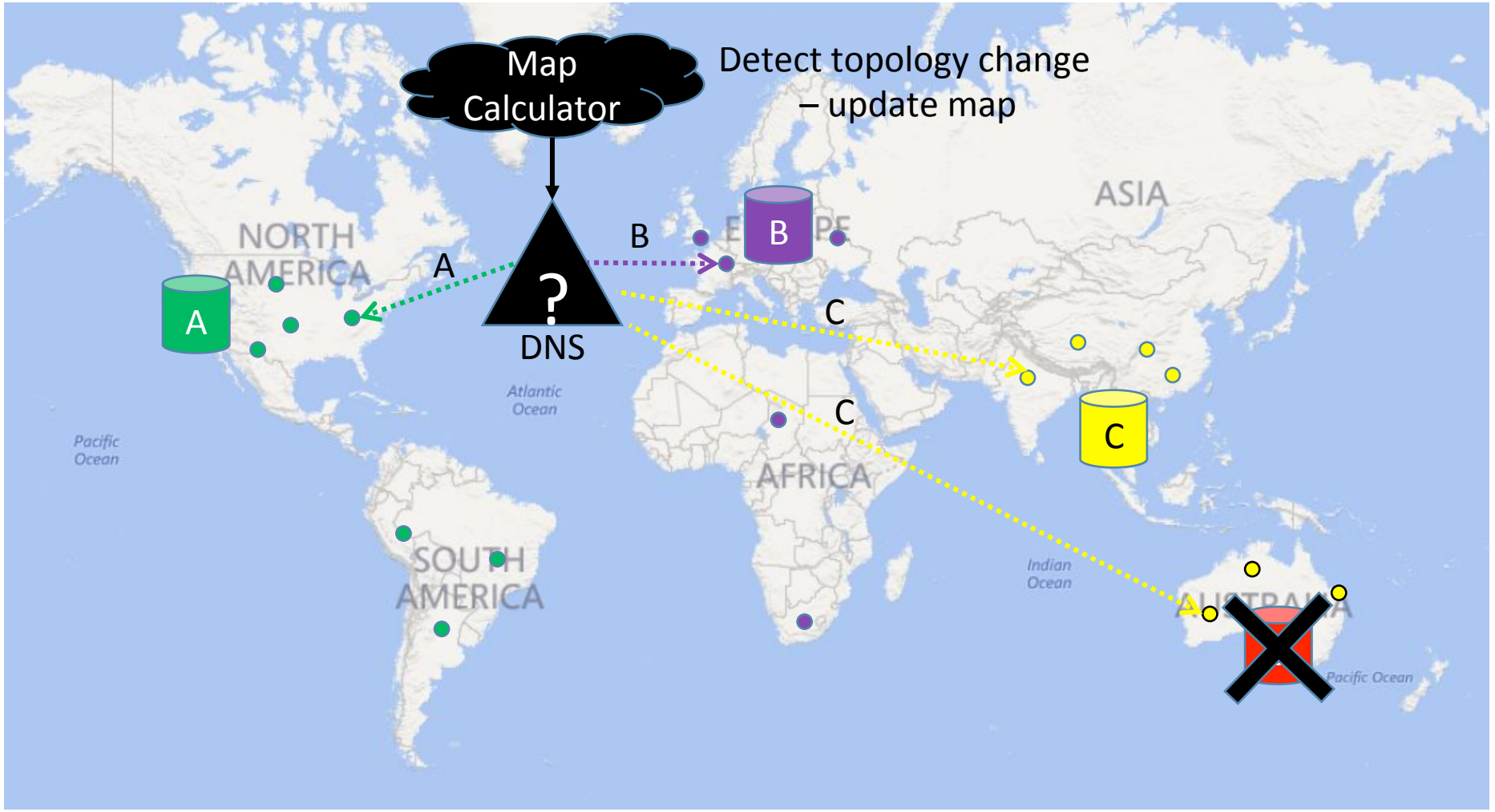
The “Map the Internet” Approach



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

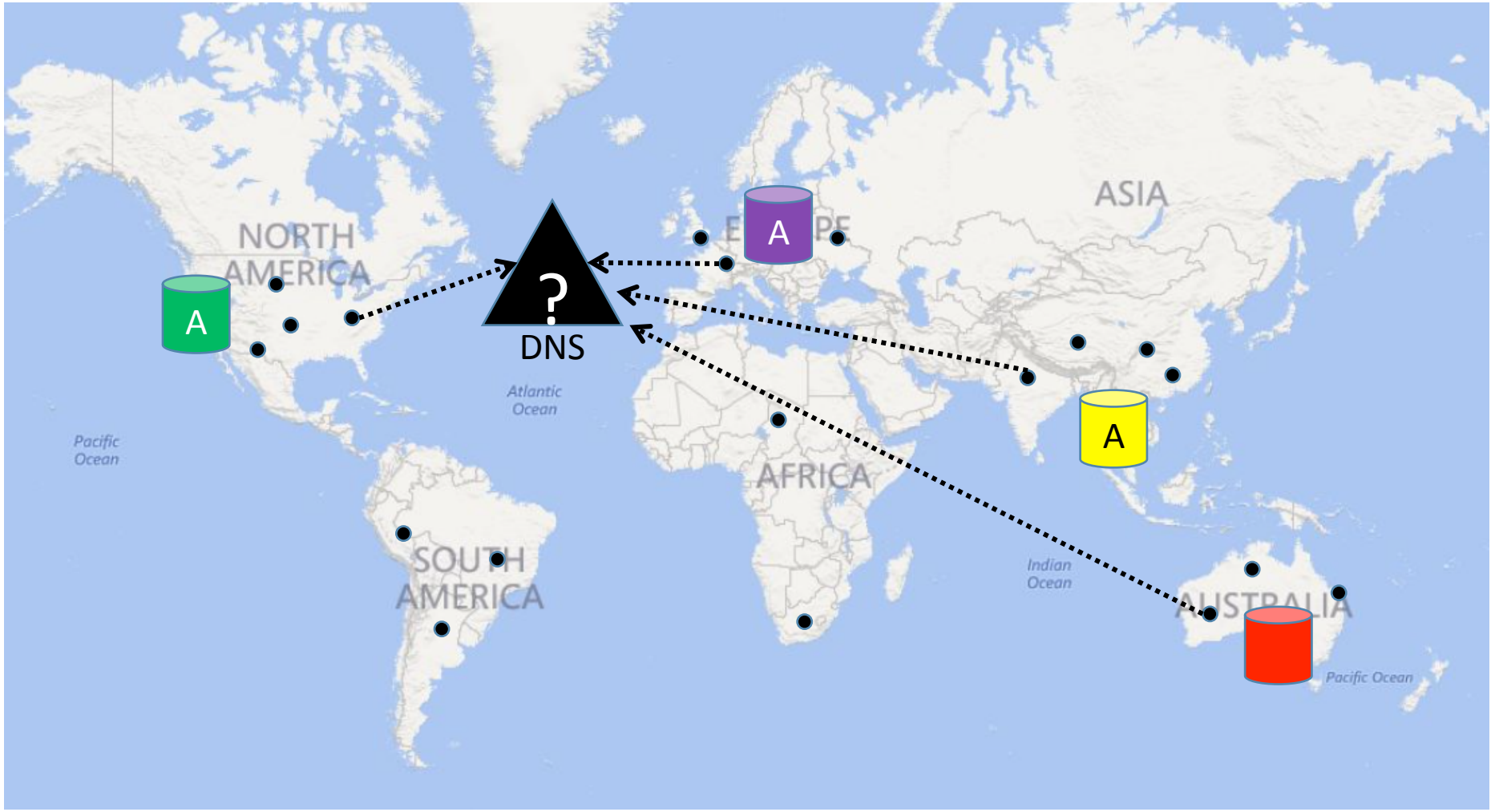
- Trade off
  - High operational cost and complexity (Large scale central global co-ordinator required)
  - DNS can be inaccurate for client proximity routing.

The “Map the Internet” Approach

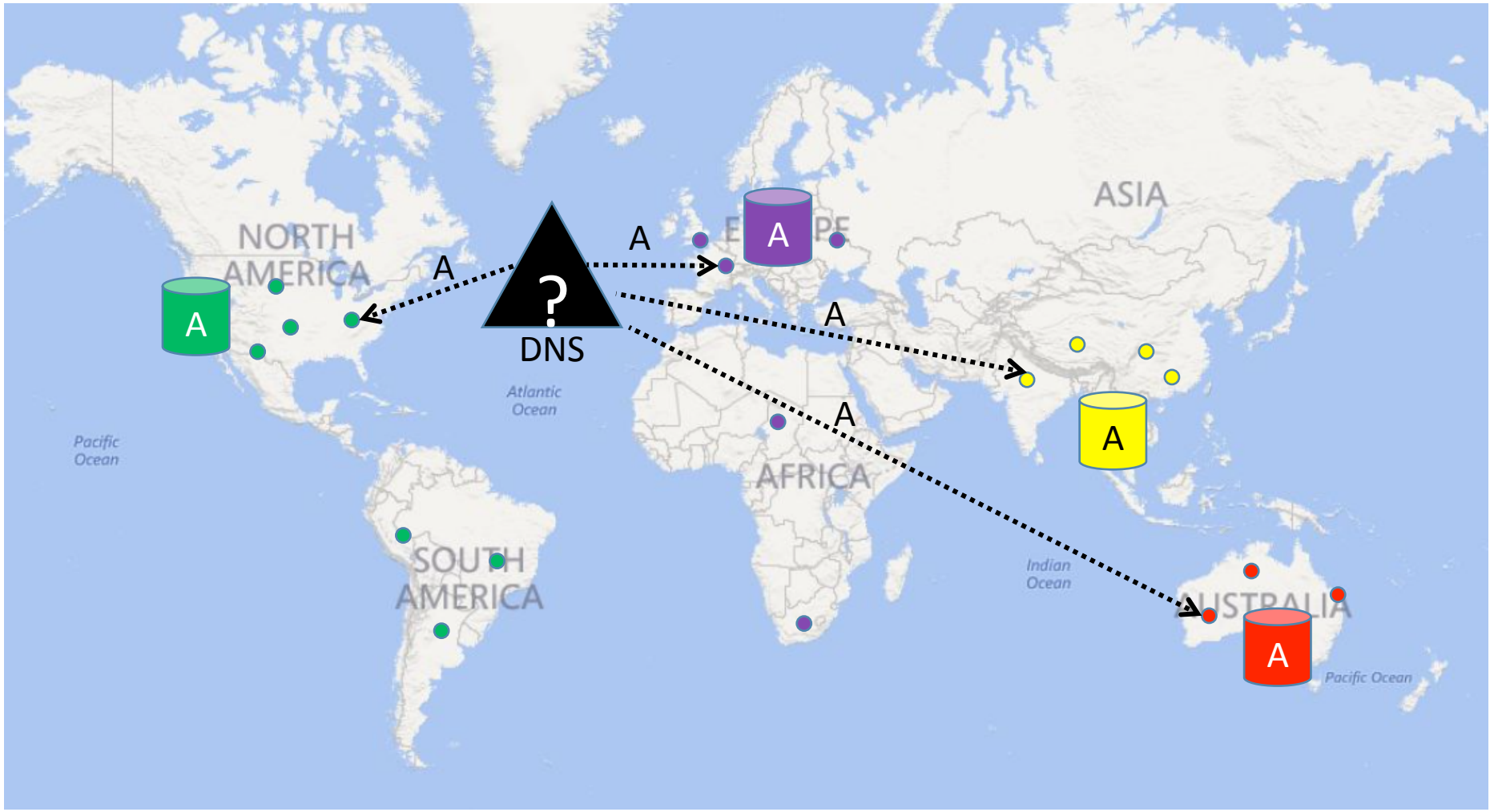
- Primary Benefit
  - Flexible Control: Can direct any DNS request to any node
- Trade off
  - High operational cost and complexity (Large scale central global co-ordinator required)
  - DNS can be inaccurate for client proximity routing.
- ***There is an alternative…***

The Anycast Approach

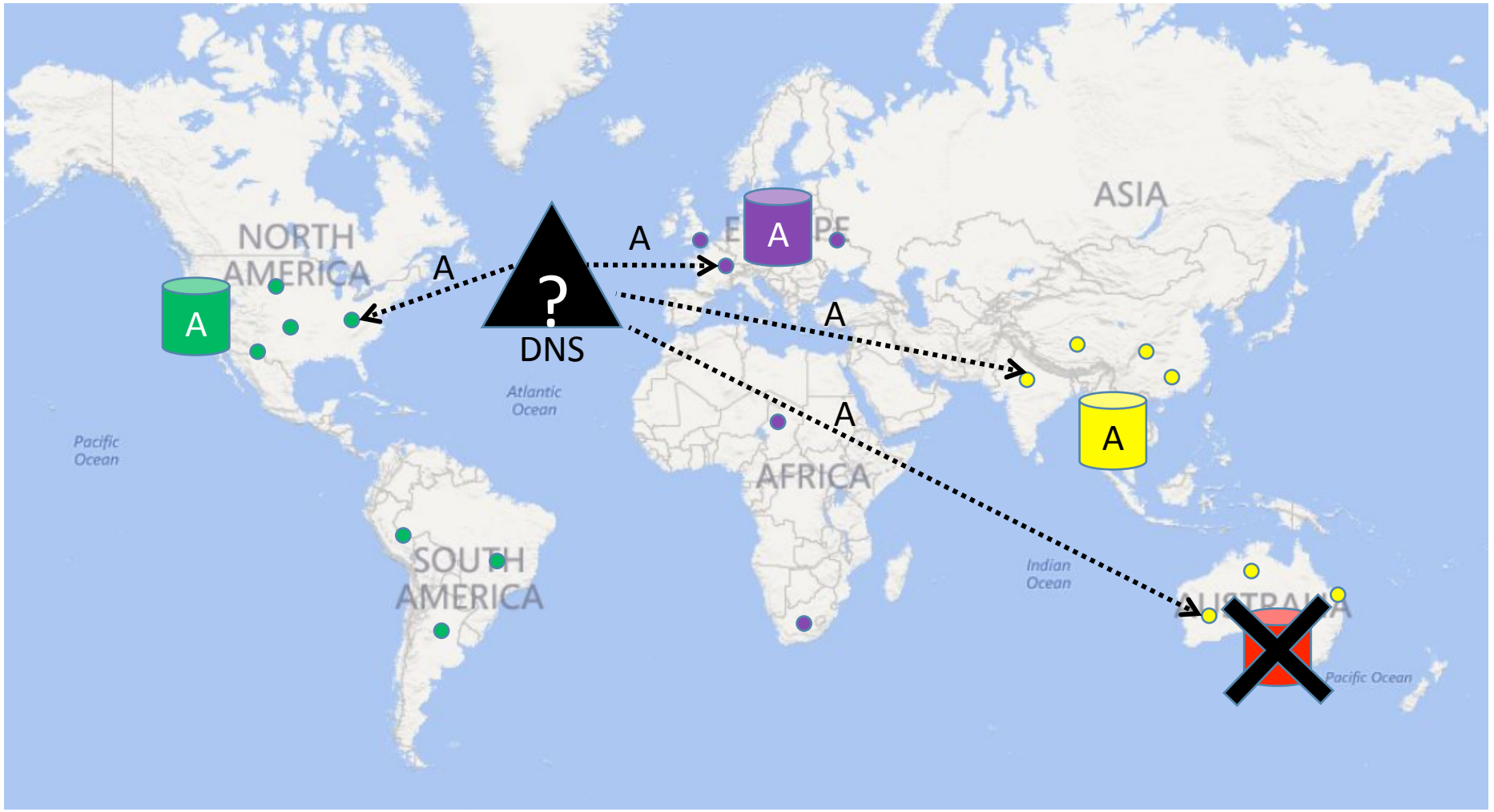
The Anycast Approach



The Anycast Approach



The Anycast Approach



Microsoft

The Anycast Approach

- Benefits
  - Simple
  - Avoids DNS-based proximity routing
- Trade off
  - Relinquish routing control to “The Internet”

Microsoft

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.

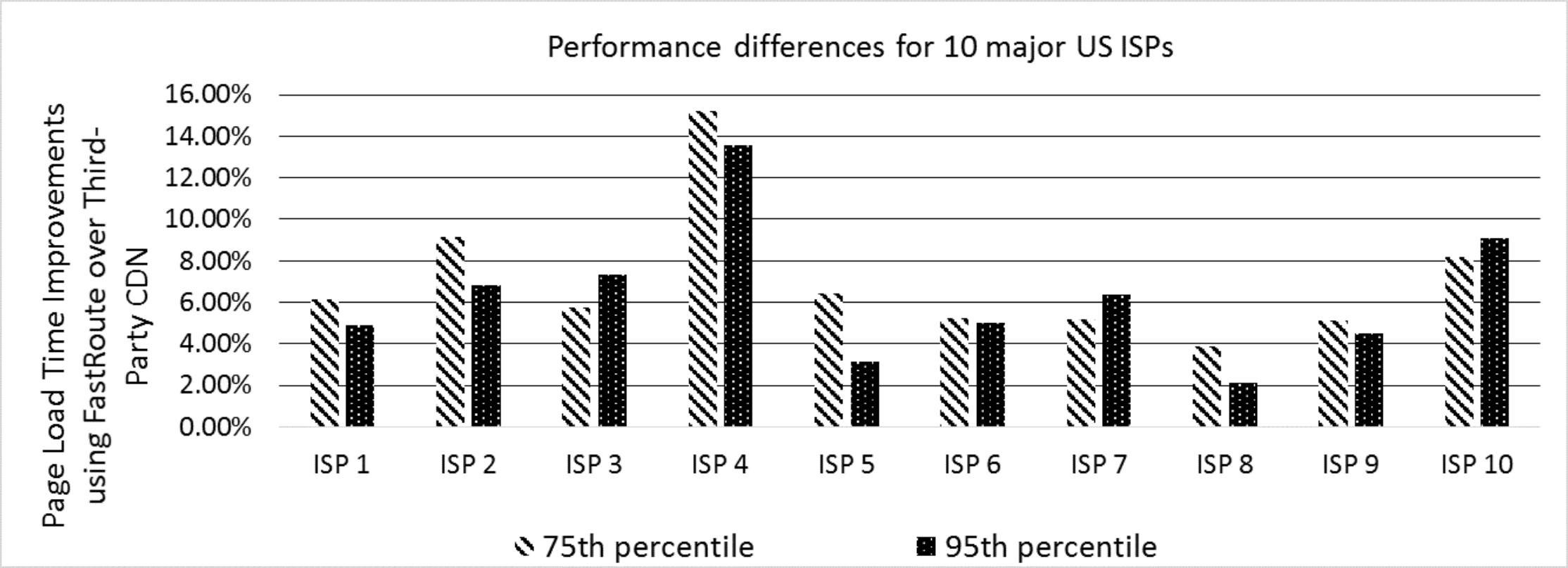
FastRoute

Questions:

1. Does Anycast provide sufficient performance?

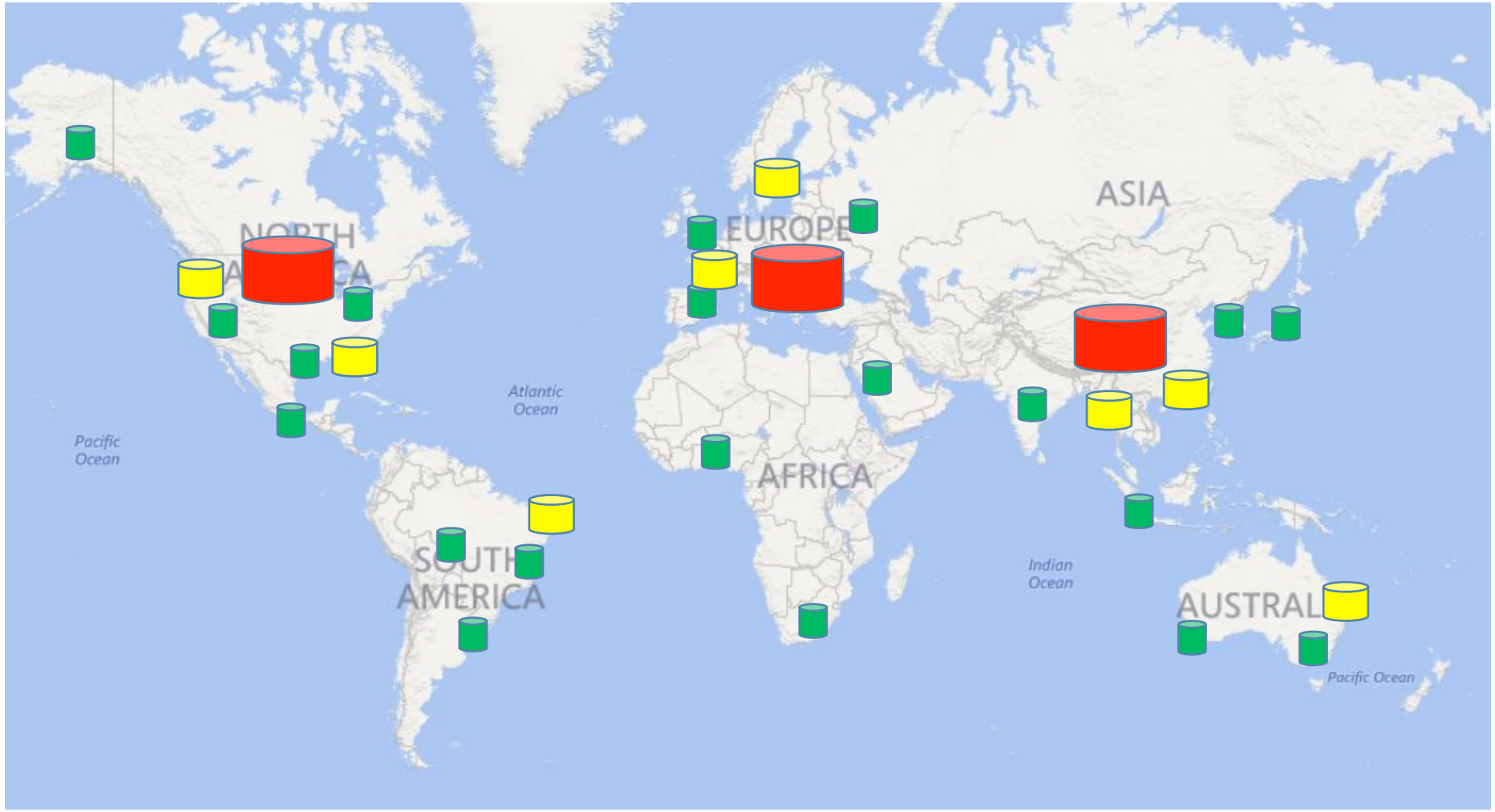
2. How can we manage the rare (but expected) individual overloaded node without sacrificing Anycast's simplicity?

Performance of Anycast?

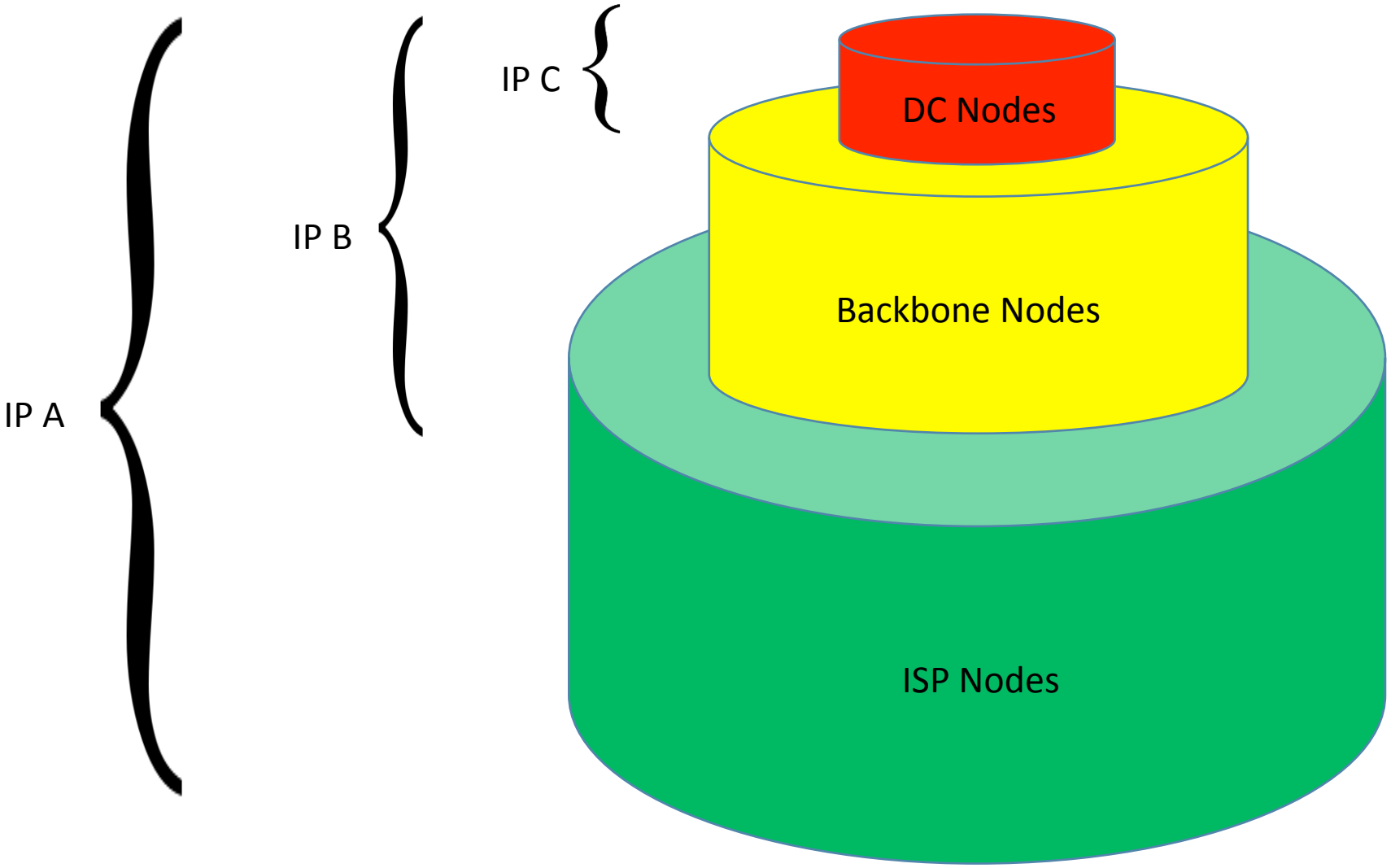


Note: Anything above 0% is good

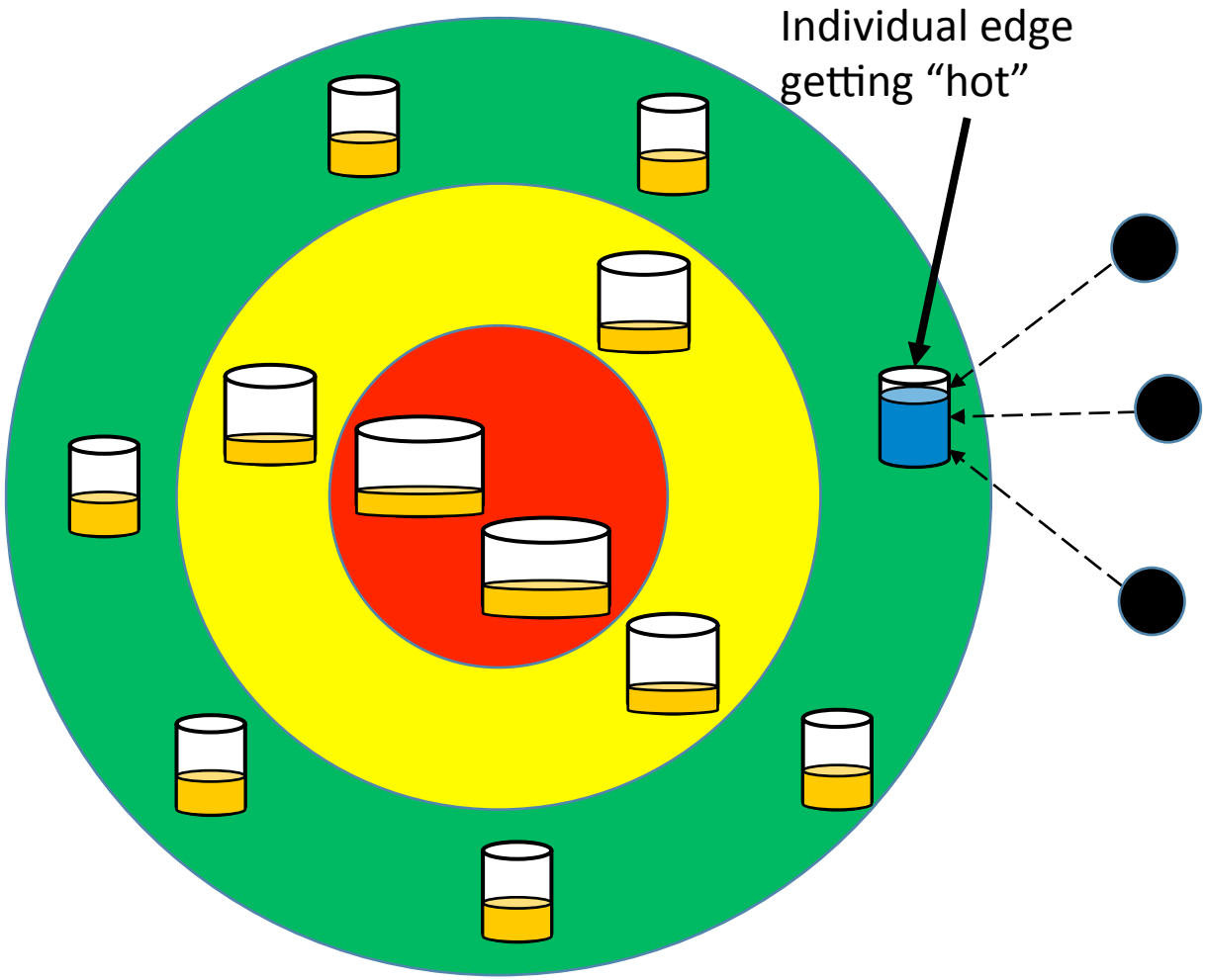
Not all Nodes are Equal!



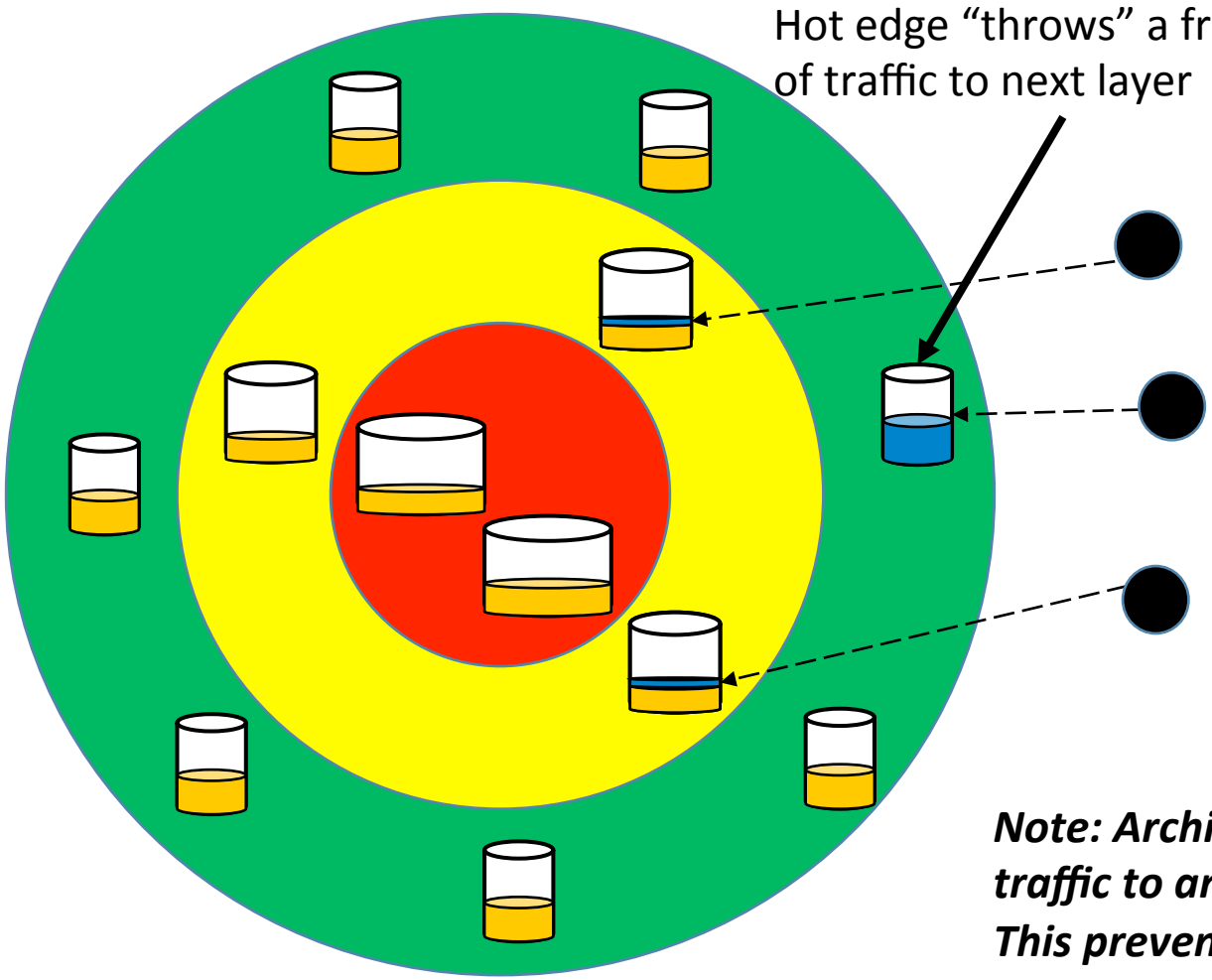
Utilizing Anycast “Layers”



Load Management using Anycast Layers

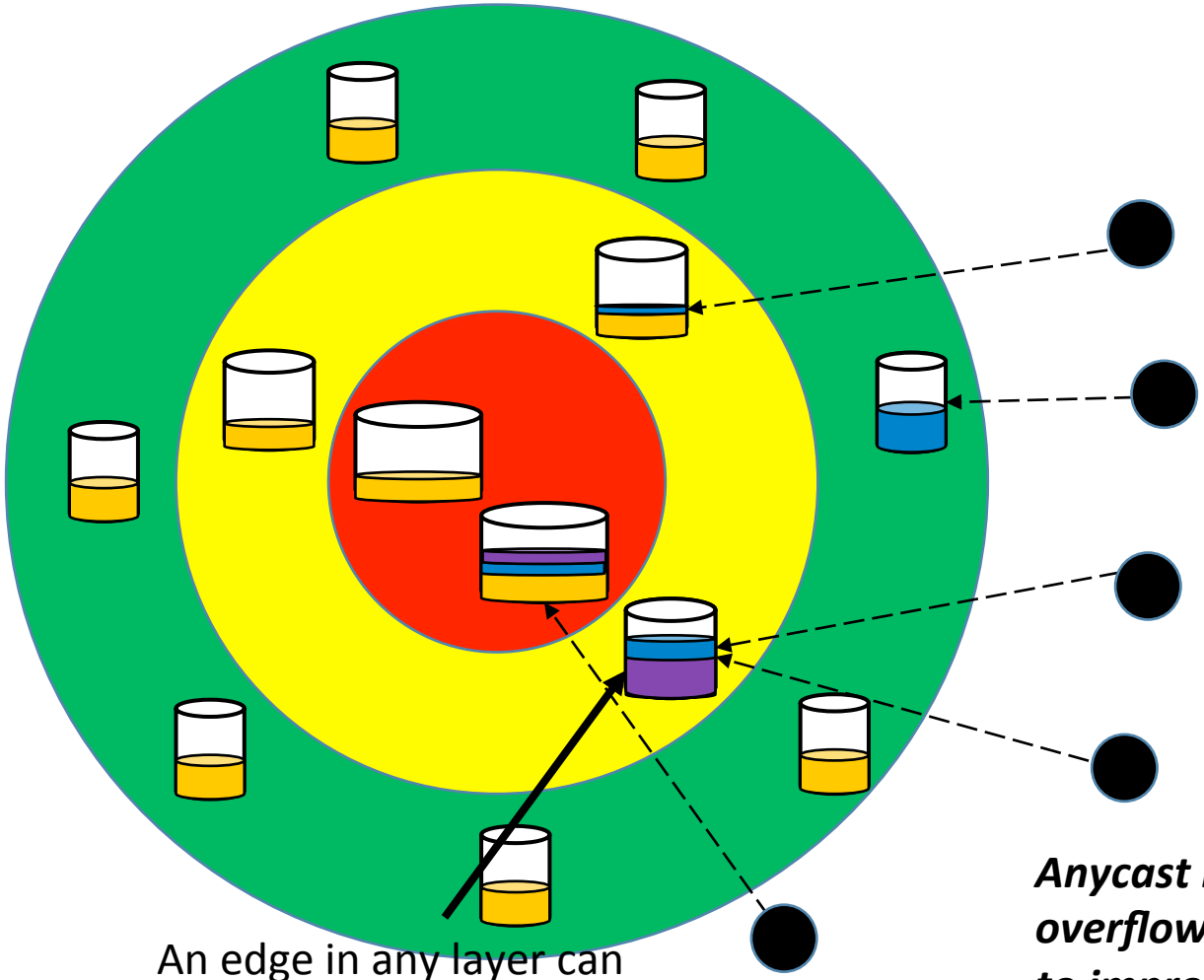


Load Management using Anycast Layers



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

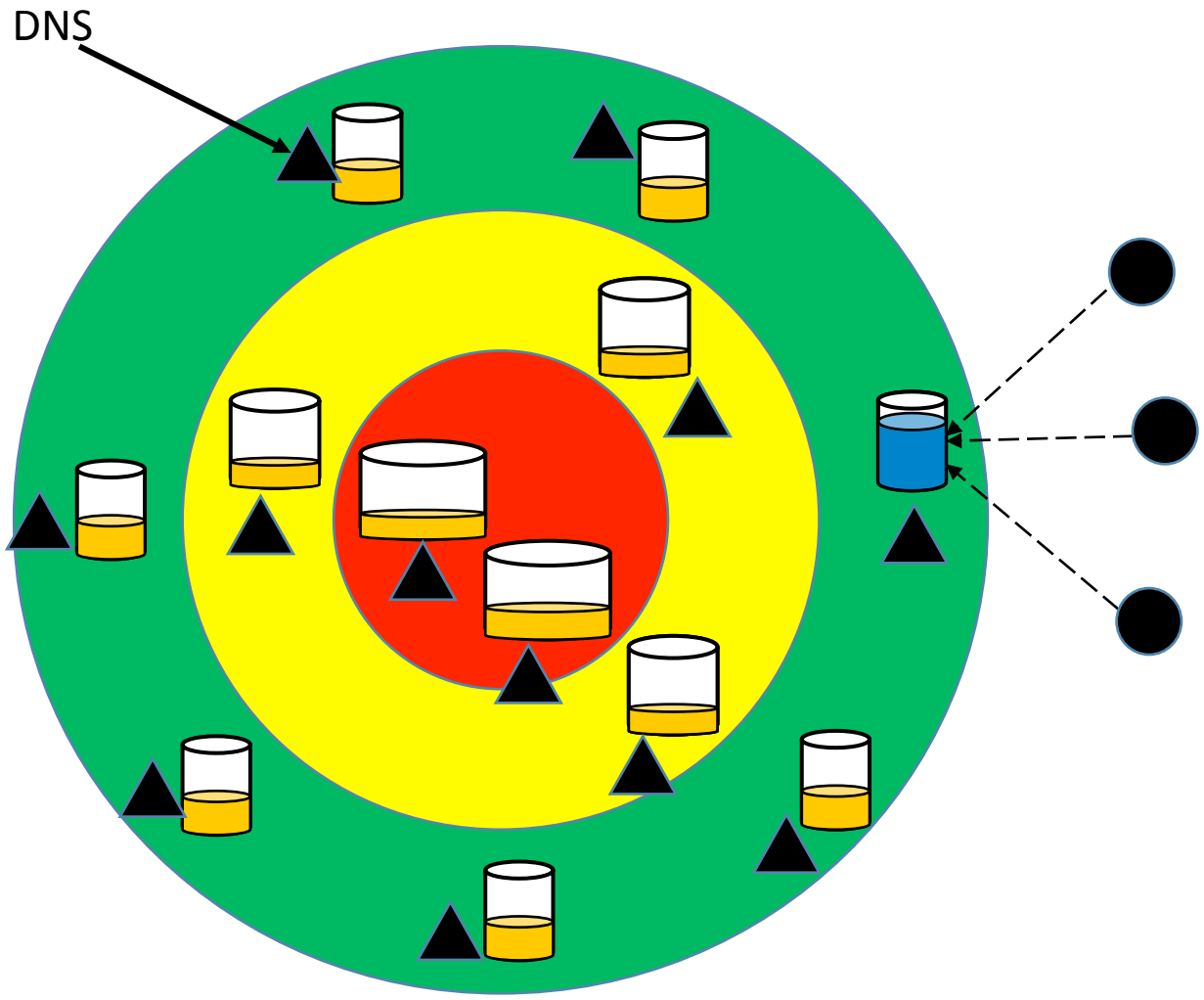
Load Management using Anycast Layers



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

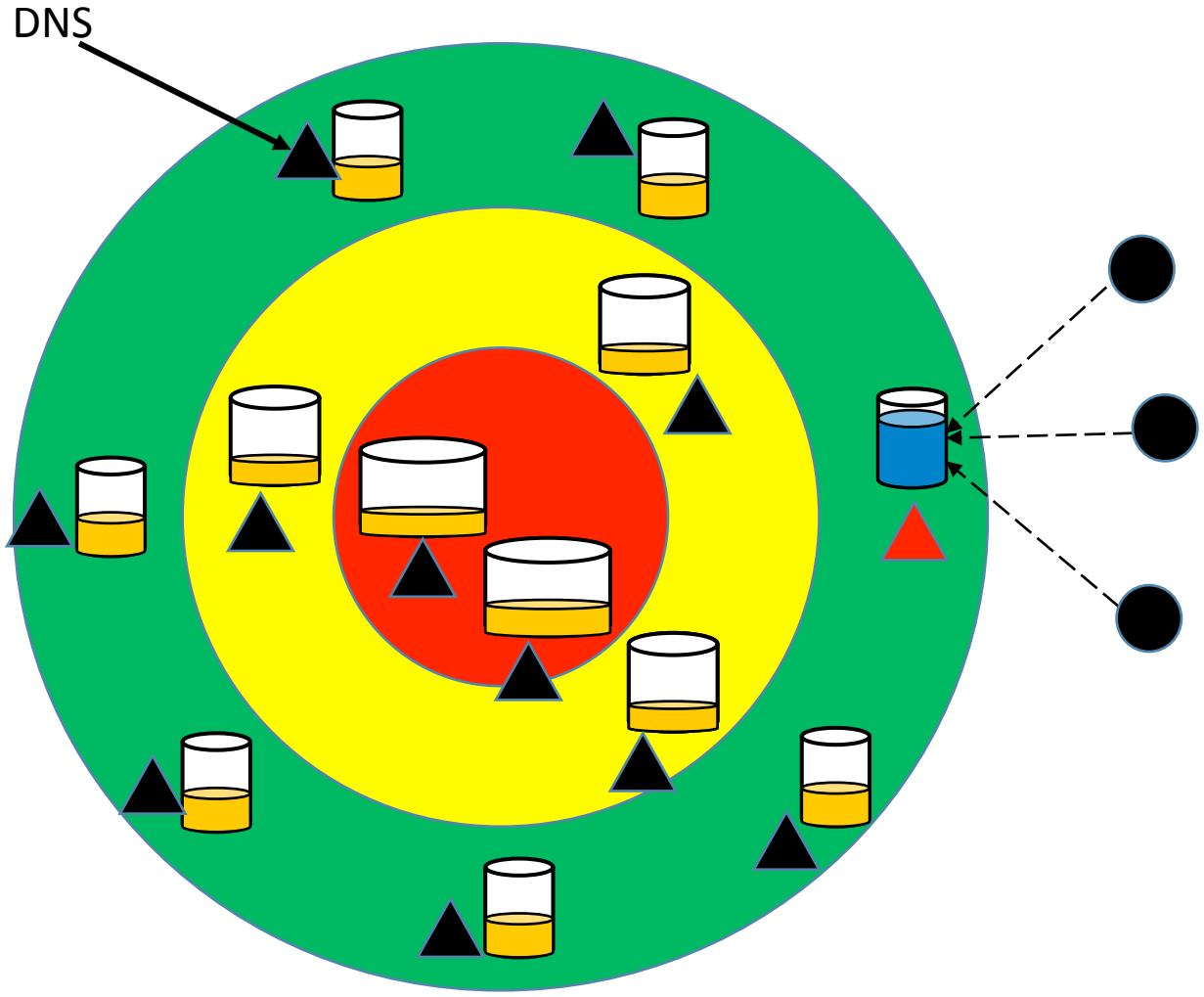
Anycast layer 0 is provisioned to absorb overflow. Further optimization can occur to improve absorption in this layer.

How to “throw” traffic to next layer?



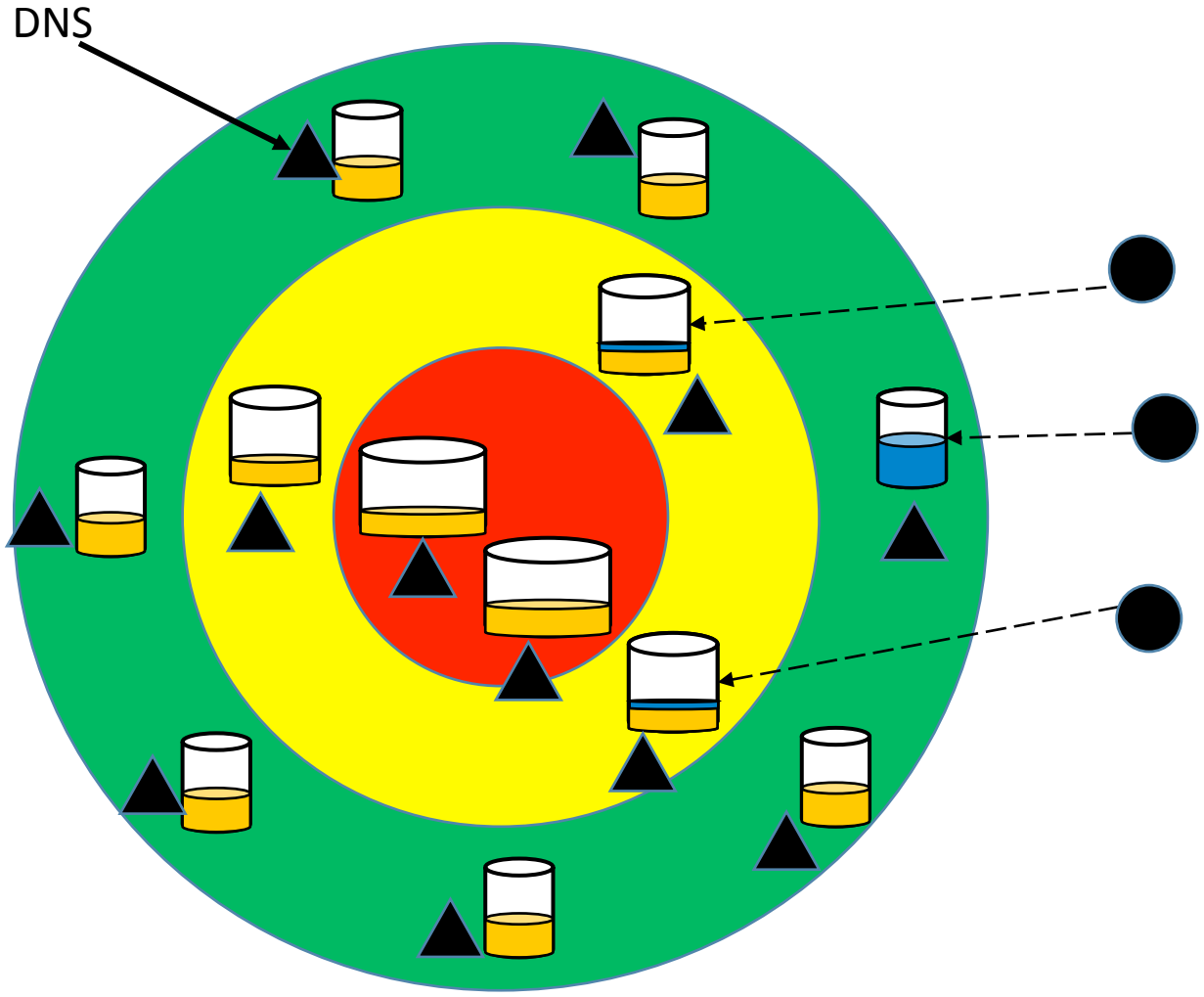
1. Co-locate DNS servers with HTTP proxies in every location

How to “throw” traffic to next layer?



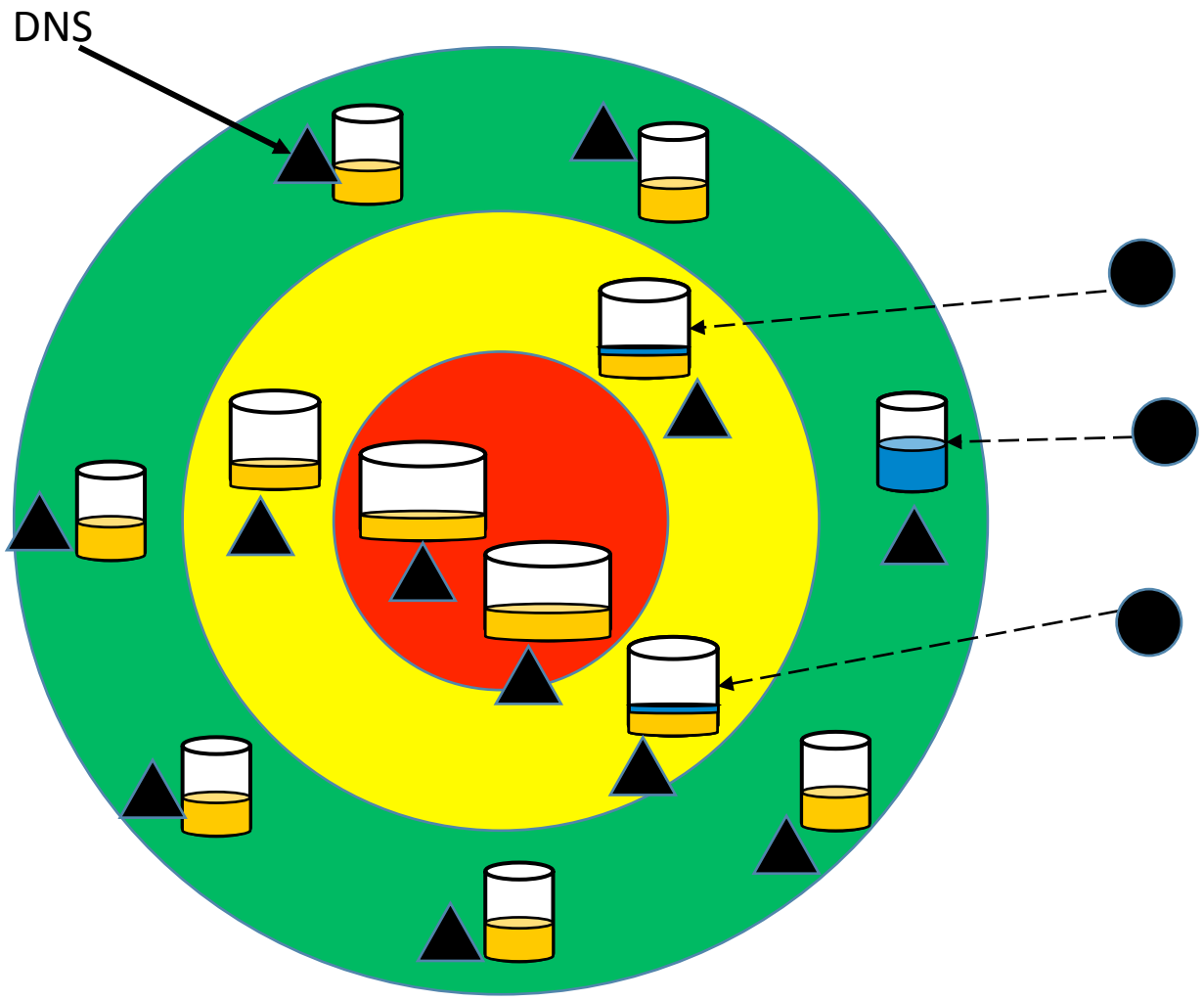
1. Co-locate DNS servers with HTTP proxies in every location
2. DNS monitors load in its own location

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)

Microsoft

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)

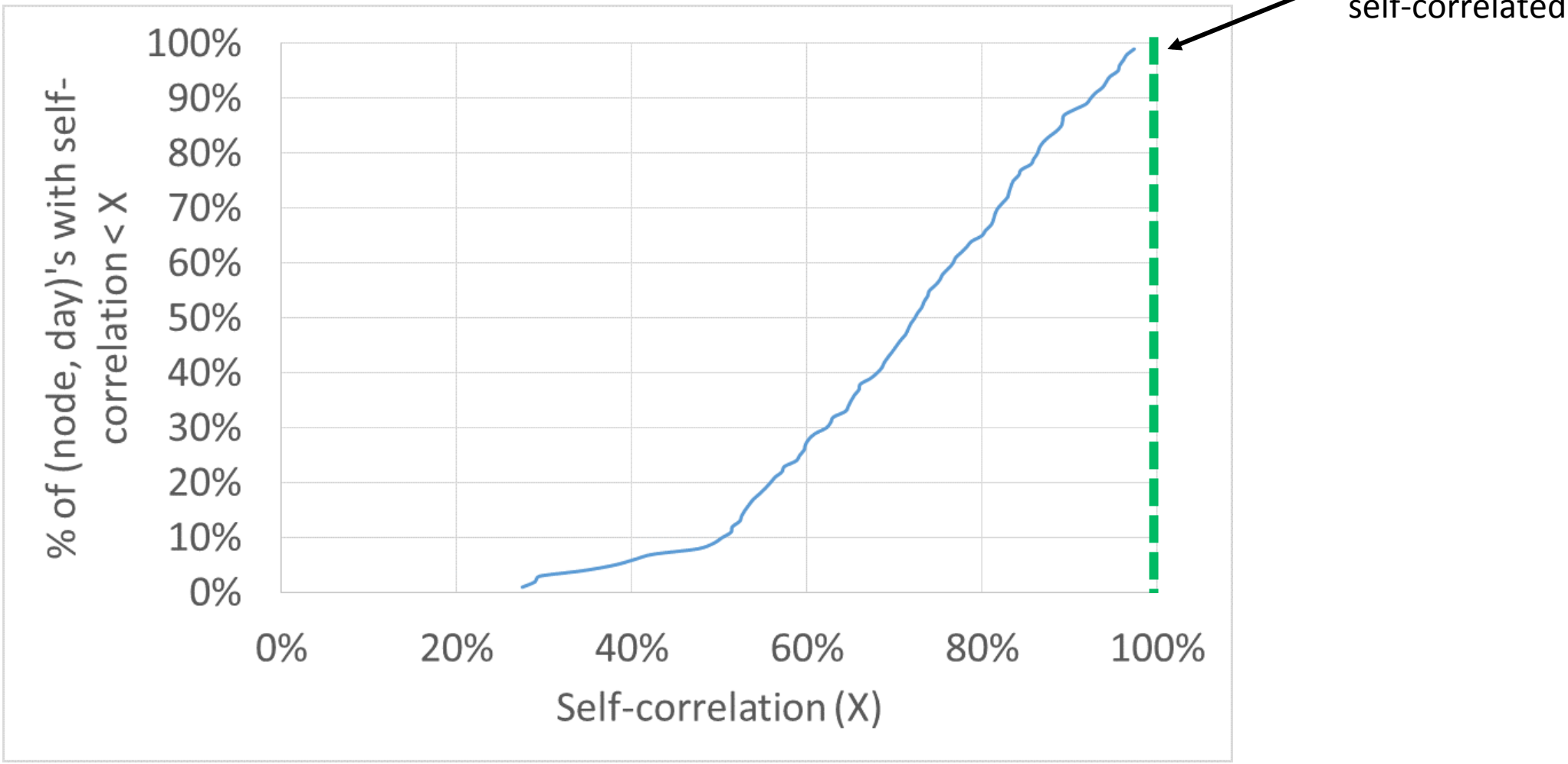
- This is not guaranteed for all requests.

Microsoft

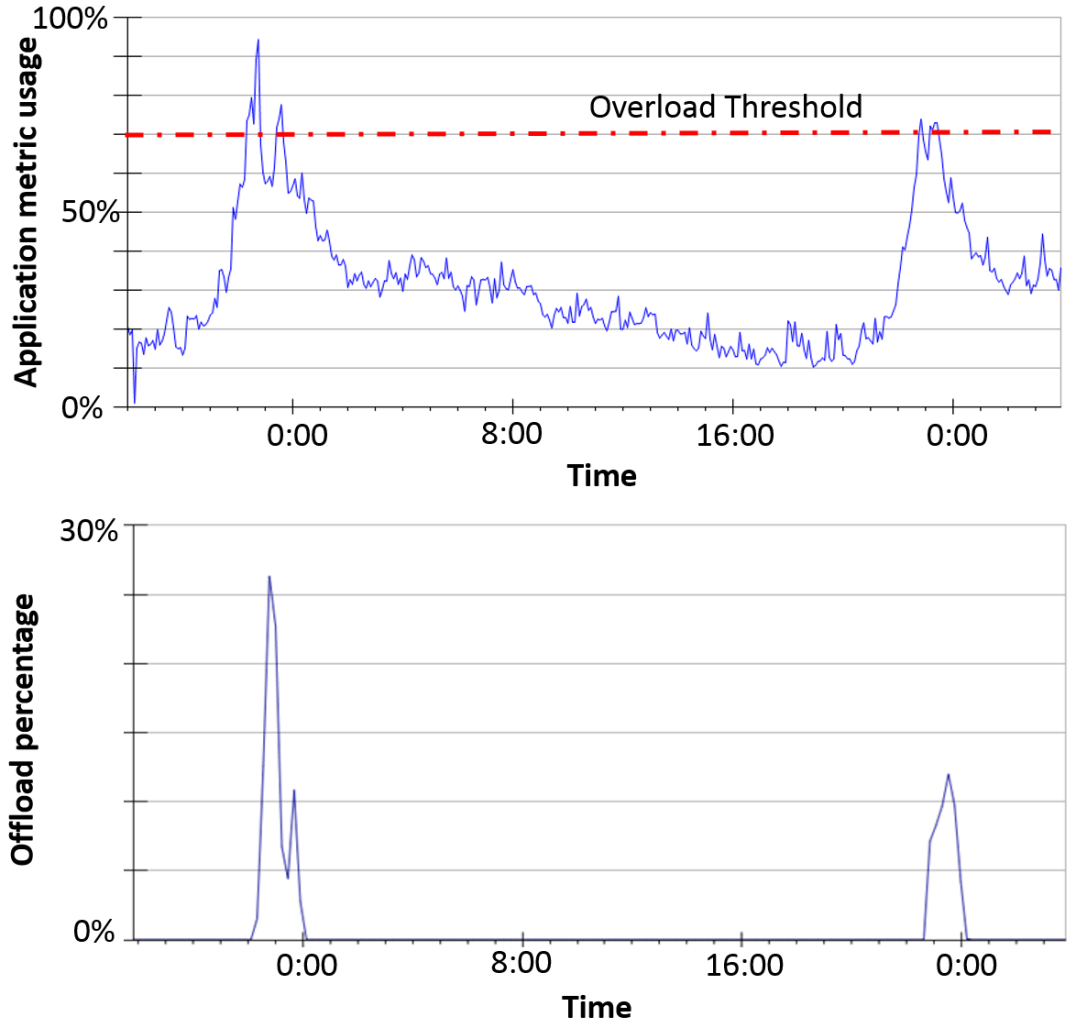
How to “throw” traffic to next layer?

- 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?

FastRoute Self-Correlation



DNS Load Management In Practice



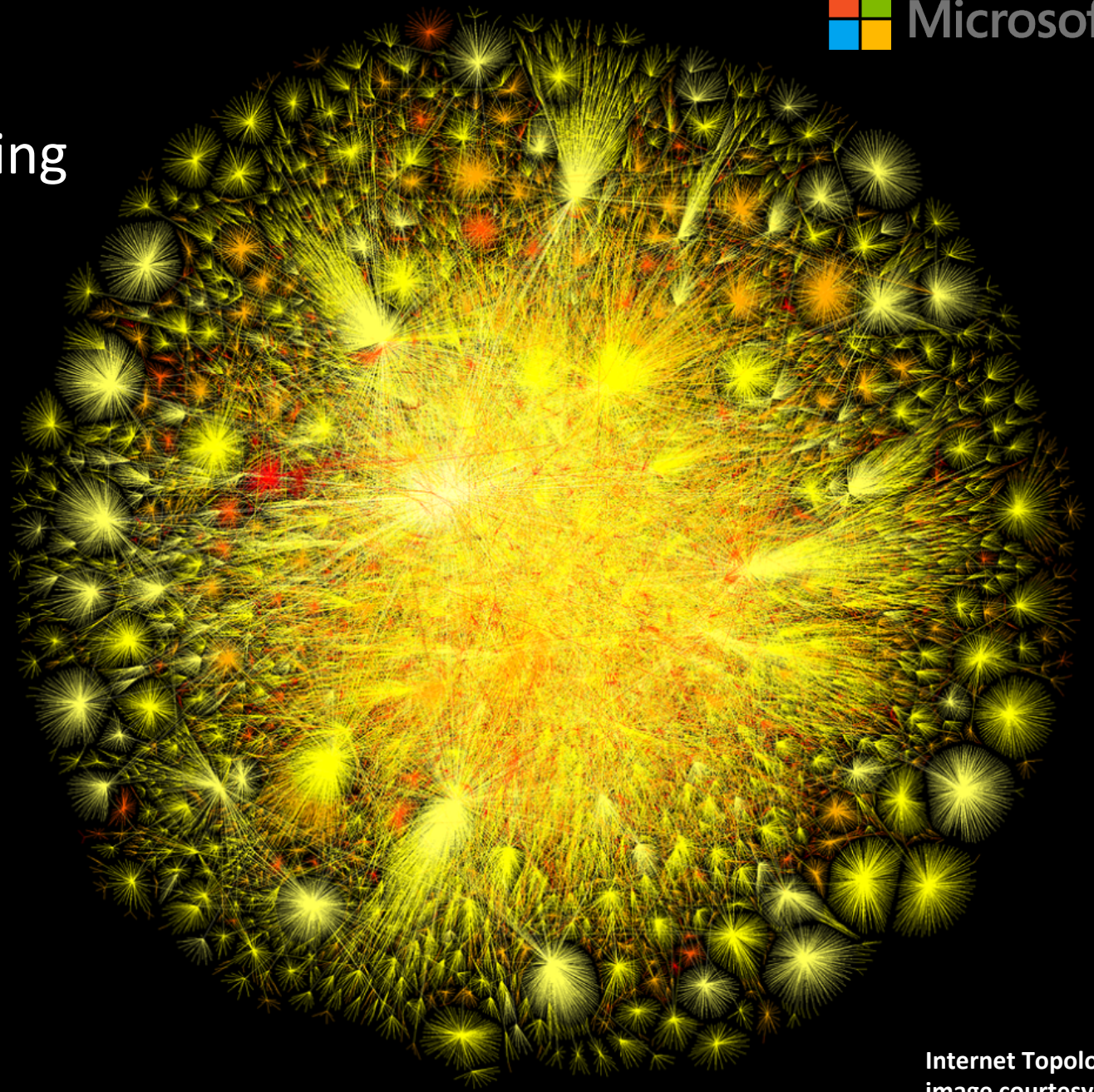
Microsoft

Basic 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.***

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



Questions?

Internet Topology image courtesy of www.opte.org