Predictable Very High Speed Networking for Big Science

Yatish Kumar
CTO Corsa Technology

Breaking Down The Title

Predictable

Traffic Engineered / Bandwidth Calendared or Mostly Vacant networks

Predictable networking returns $$ in storage

Very High Speed Networking

Efficient conversion of $$ into BW

Avoiding technology limitations

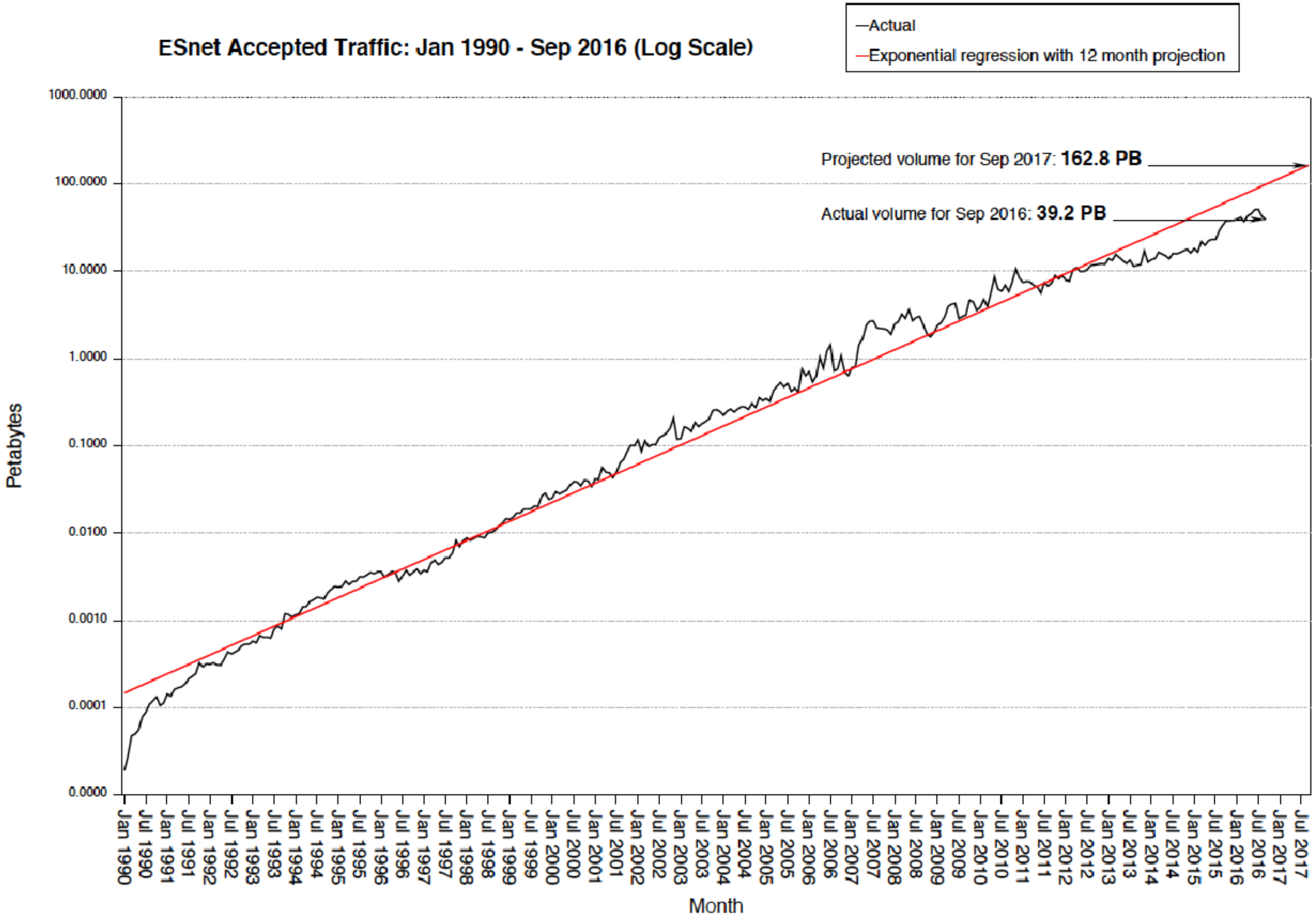
For Big Science

Moving large datasets and connecting instruments is a very special case of networking. No need to boil the ocean. Just a very big pond.

You Already Know Everything I Will Talk About

The purpose is to draw focus to what I have learned after countless hours of discussion with numerous NRE networks

Where is Data Intensive Science Headed ?

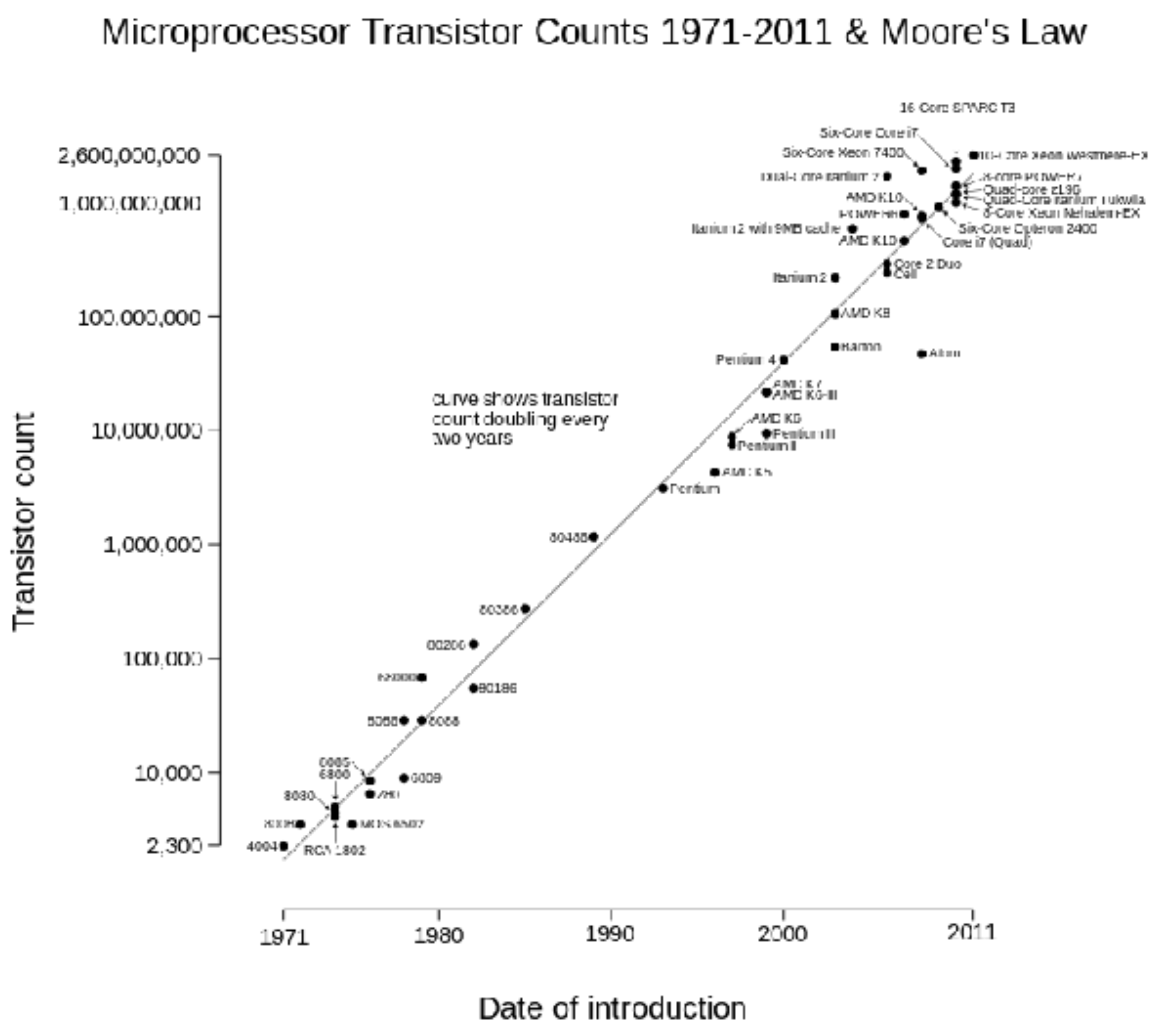


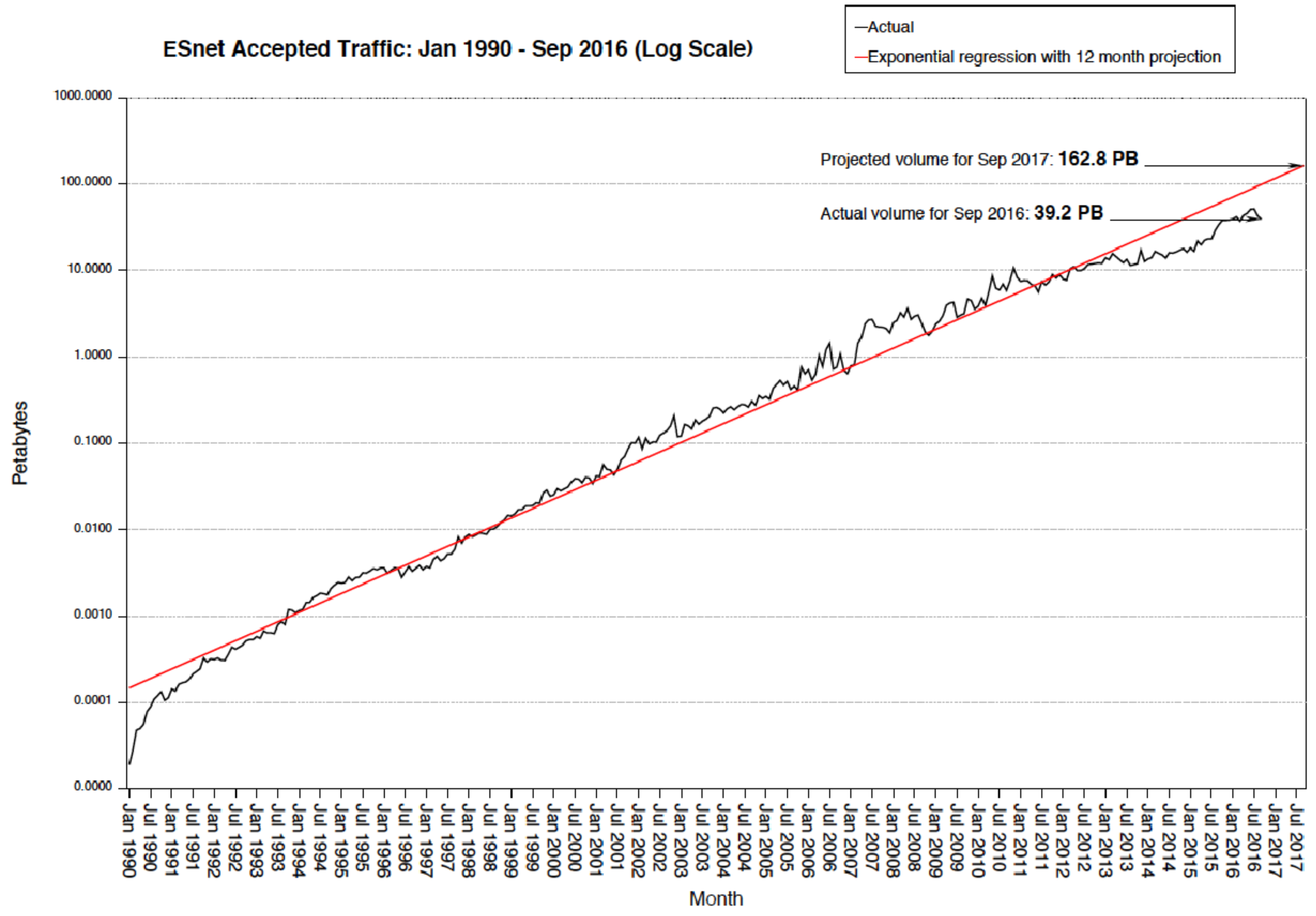
Demand is exponential

but

Budgets are linear or flat

Moore’s Law Ended at 28nm





The number of transistors in a dense integrated circuit doubles approximately every two years.

Exponential Performance Gain
Constant / Linear Cost per mm^2

Cause - Effect: 2013 - 2016 - Future

Networking solutions for the first two or three years of the transition have squeezed efficiency out of the eco system:

1. Finfets have provided 2x speed, but linear area gain
2. We have accumulated gains by skipping a nm generation.
    1. 28 nm ~~- 20 nm~~ - 16 nm - ~~10 nm~~ - 7 nm - ~~5/4/3 nm~~
3. We have commoditized HW so that silicon cost is a greater fraction of system cost. (Was around 10% now 25-50%)
4. We have improved contact and metal layers whilst running into barriers at the transistor layer

40nm 28nm | 16nm
Trident (640G) / Trident 2 (1.2T) / Tomahawk (3.2T) / Tomahawk 2 (6.4T)

Soon the options train will compress then what ?

7 nm will be the final node
We have 2 or 3 more years to react

The end of the FET transistor is not the end of the road

Carbon Nanotubes
Vertical nm Structures

http://semiengineering.com/to-7nm-and-beyond/

But first we have to make it past the next 5 to 10 years

Corner Stones for Growth (2017-2023)

Parallelism

Coherent DWDM + Super Channels + Many L2/L3 Chips

Simplification

Rethink our need for L2 and L3 protocols.

Rethink our need for overlay networks (More L2 and L3 protocols)

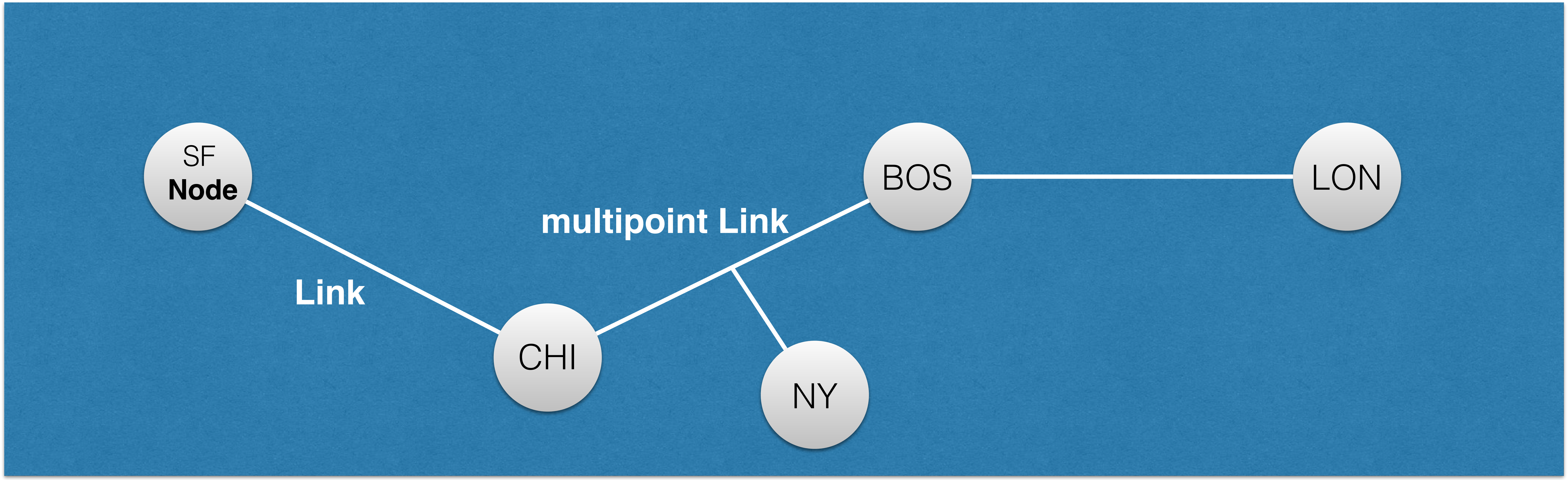
Rethink our need for running links at 100%

Rethink our need for Programmable Networks

Underlay Networks

Underlay Fabric

OSI Stack

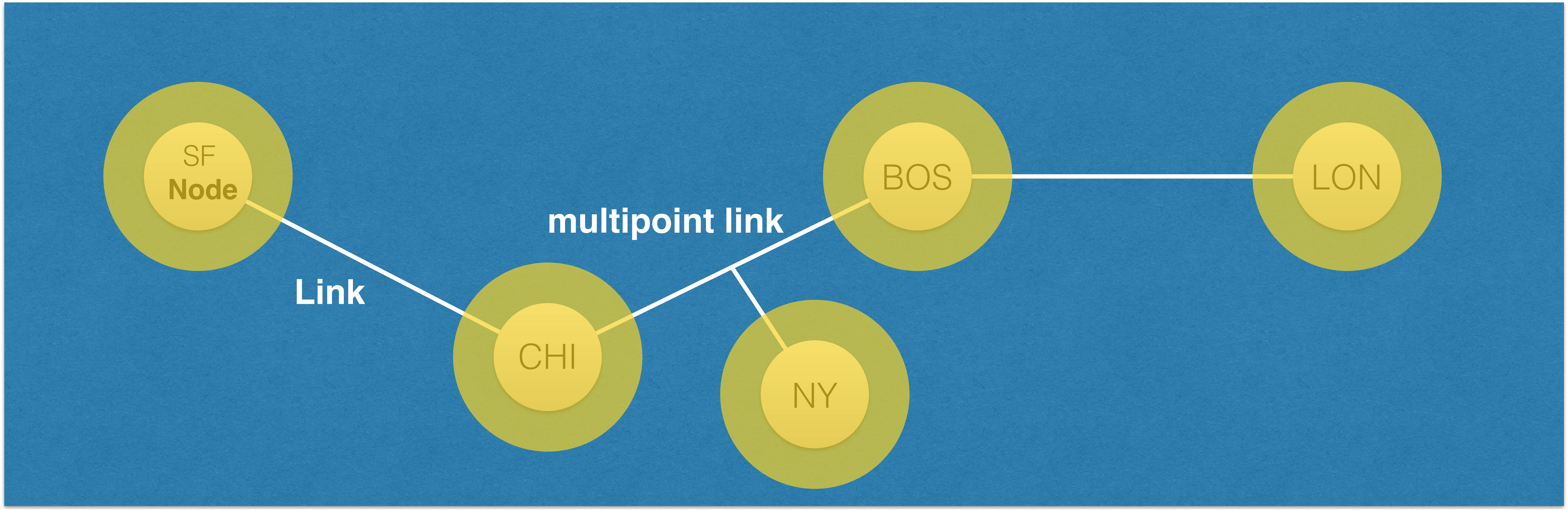


L3 Network Layer
L2 Link Layer
L1 Physical

1. The underlay network delivers packets between network elements
2. The underlay network deals with issues related to moving packets over physical distances
   - Traffic engineering, resilience to link failures are the primary objective

Underlay vs. Overlay Networks

Overlay Network



OSI Stack

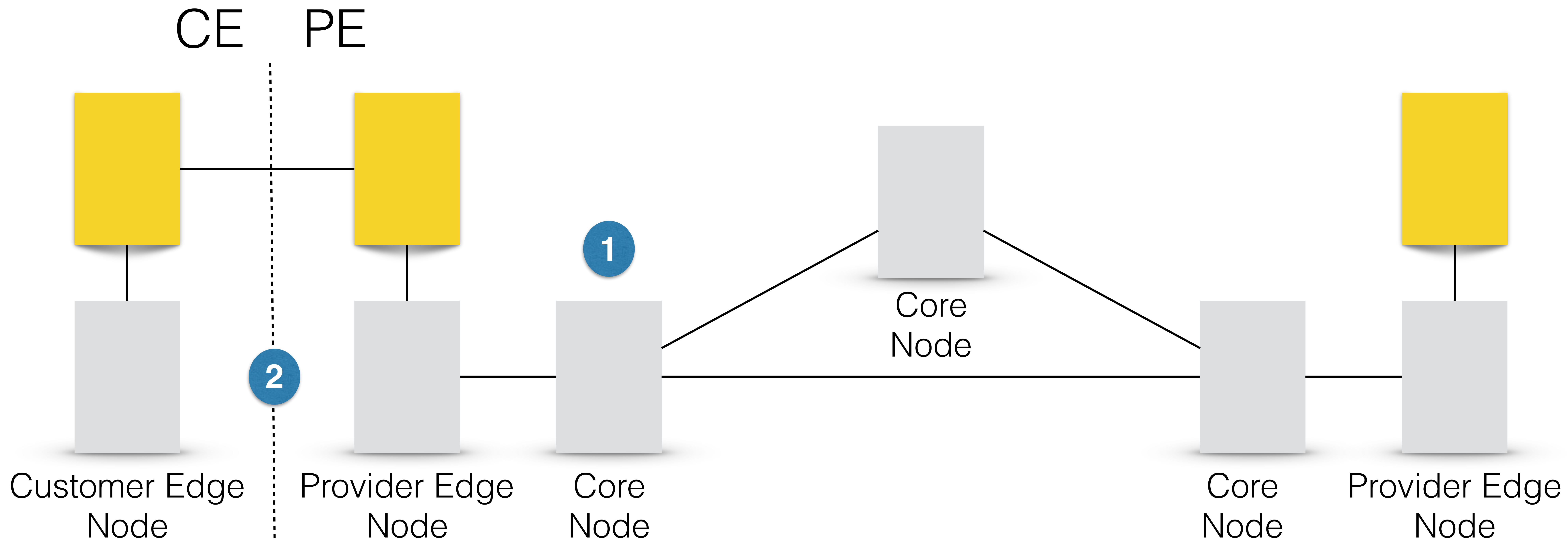
L3 Network Layer
L2 Link Layer

L3 Network Layer
L2 Link Layer
L1 Physical

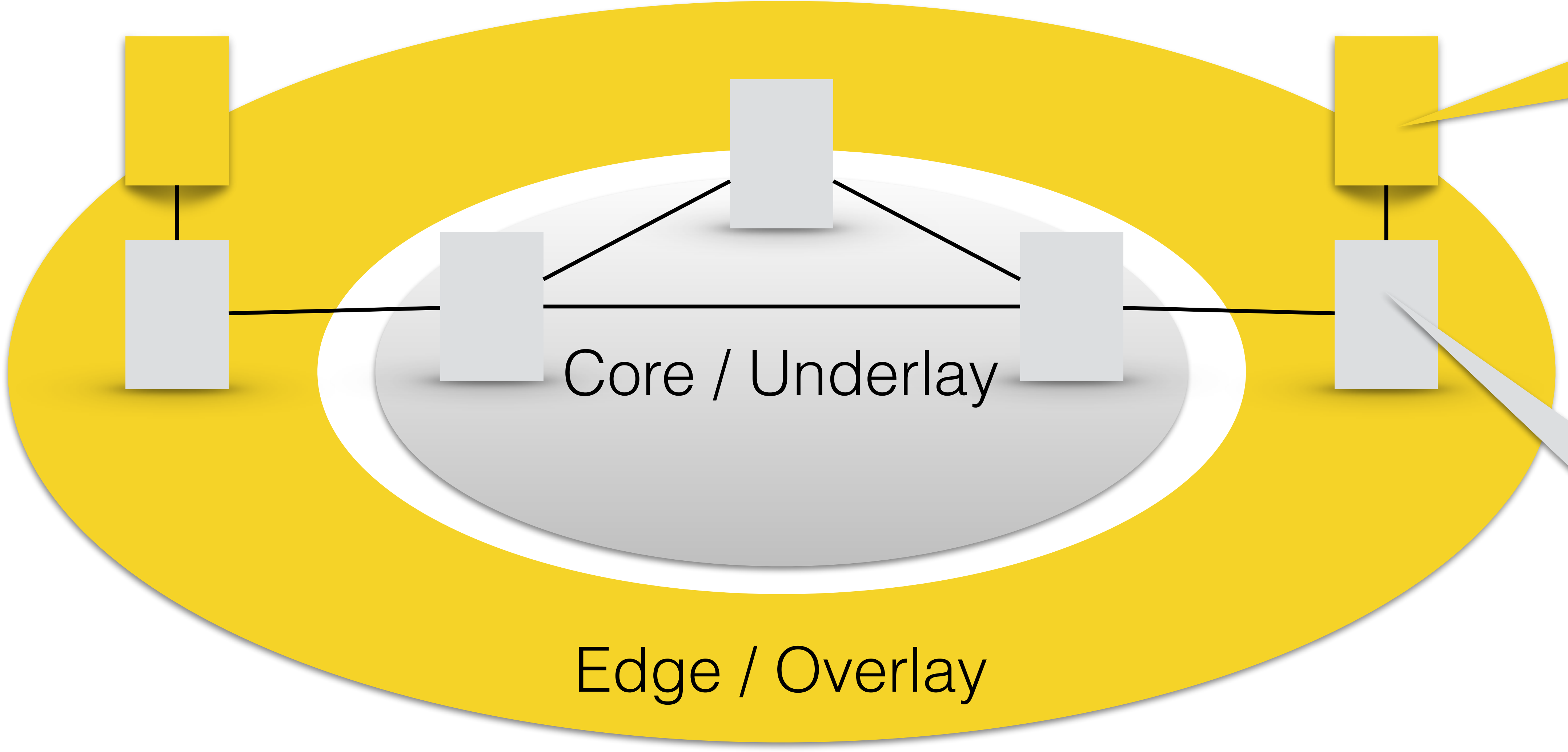
1. Overlay networks are built assuming underlay networks are performing their function
2. Primary objectives of overlay networks
   - Forwarding abstraction (L3VPN, L2VPN, Policy Based Routing etc..)
   - Multi-tenancy on the underlay (isolation)
   - Service level resilience (multiple peering, multiple underlays etc..)

Overlay Underlay Network

1. Core nodes do not run overlay protocols
2. Edge nodes do not hand off underlay connectivity between domains



Overlay Underlay OR Edge Core
Same Thing. Tomaato or Tomahto

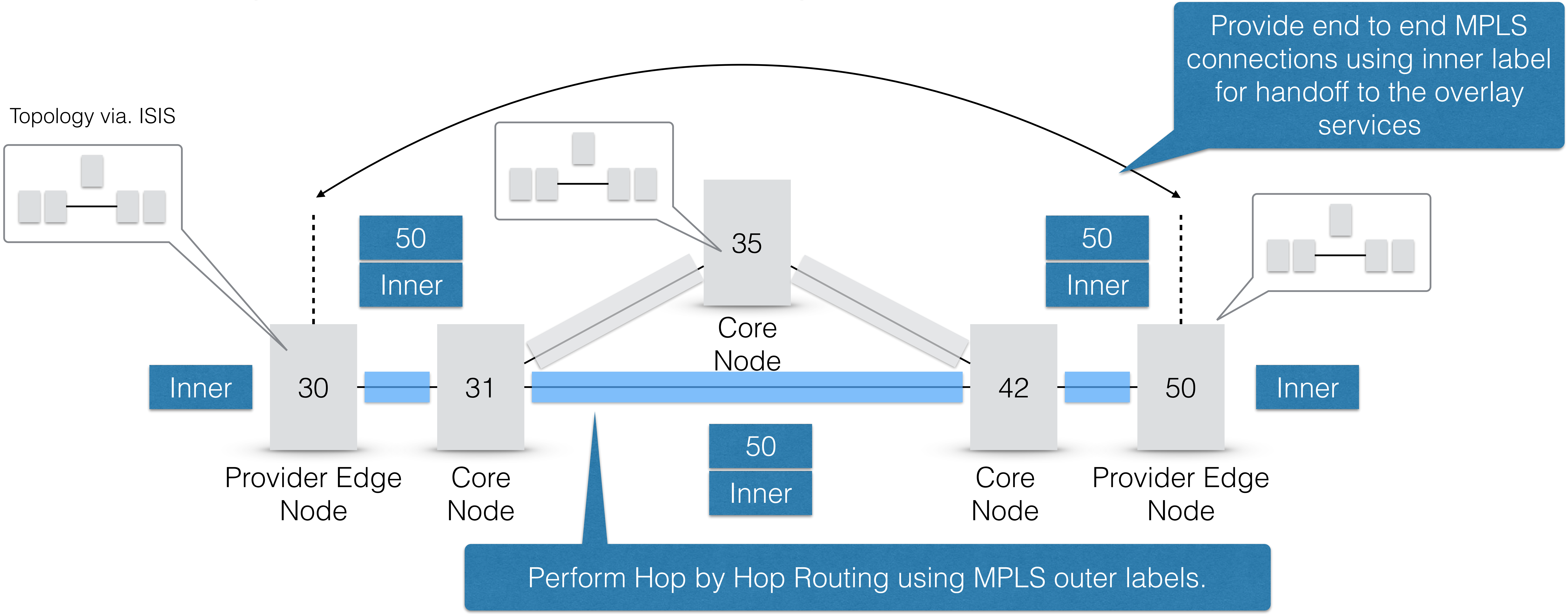


L3VPN - eBGP, ISIS, OSPF
L2VPN - VPLS
BoD - VLAN, MPLS pseudowire

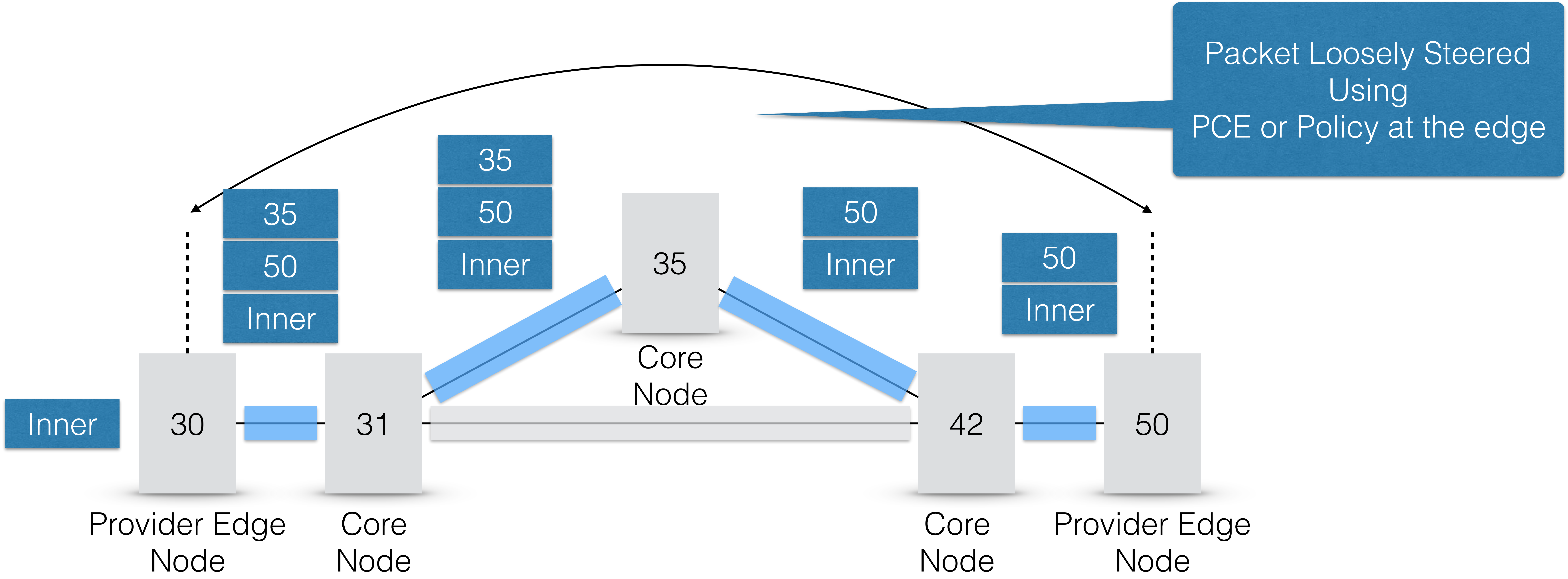
IP Networks - BGP + ISIS + Policy
E-VPN - MPLS edge + LSP Core
SPB - Ethernet edge + Ethernet Core

MPLS - LDP, RSVP, MPLS-TE
QinQ - Learning Tag substitution Flooding
MinM - ISIS / SPB
Routed - ISIS / L3 weights and policy
Seg Routing - PCE, ISIS

Segment routing in a nutshell



Segment routing in a nutshell



Nice Features of Segment Routing

1. Default path computation. However provides selective overrides.
2. Active / Active path selection.
    1. Compute distribution identifier at source.
    2. Use selective overrides to drive packets down both active paths.
3. No explosion of LSP paths.
    1. No network flow based states.
    2. No distribution of IP or Mac tables from the edge to the core.
    3. Unless you want to.
4. Limited expansion of packet header.

Segment Routing is SDN for the Core WAN

Separation of the control plane and the data plane
Programmability of key behaviours:

- Topology and Forwarding Algorithm
- Active / Active multi-paths
- Centralized / Global traffic optimization
- Resilience and Recovery
- Hop by Hop Forwarding Decisions (True L3 Behaviour)

Match-Action is **not** the only SDN Paradigm

AWESOME… But where is this talk headed ?

The Arc Of Discussion

End of Moore's Law makes us want to grow BW by simplification AND parallelism

Simplification

- **Rethink our need for L2 and L3 protocols.**
- **Rethink our need for overlay networks (More L2 and L3 protocols)**
- Rethink our need for running links at < 100%
- **Rethink our need for Programmable Networks**

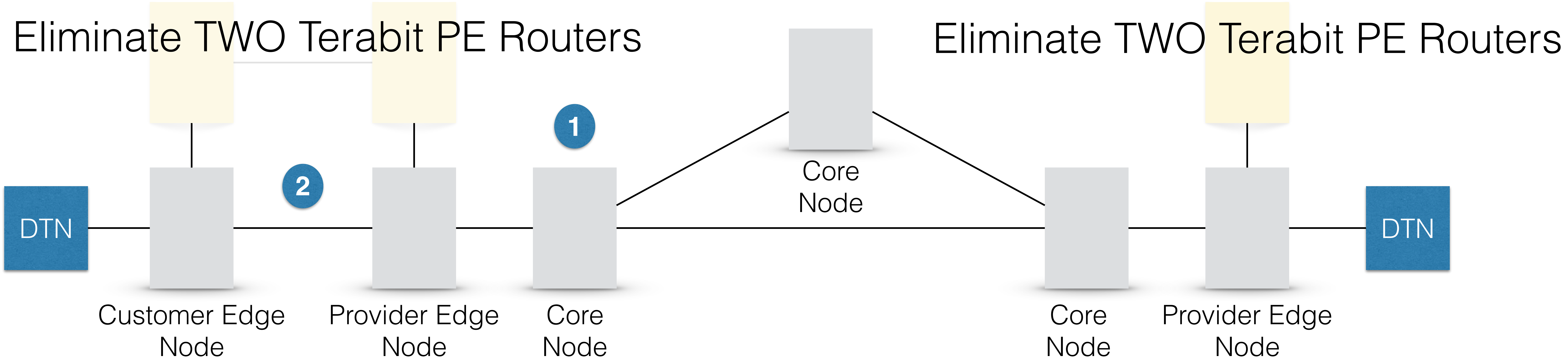
Segment routing is a promising tool for the purpose

Data Intensive Science is not General Purpose Networking

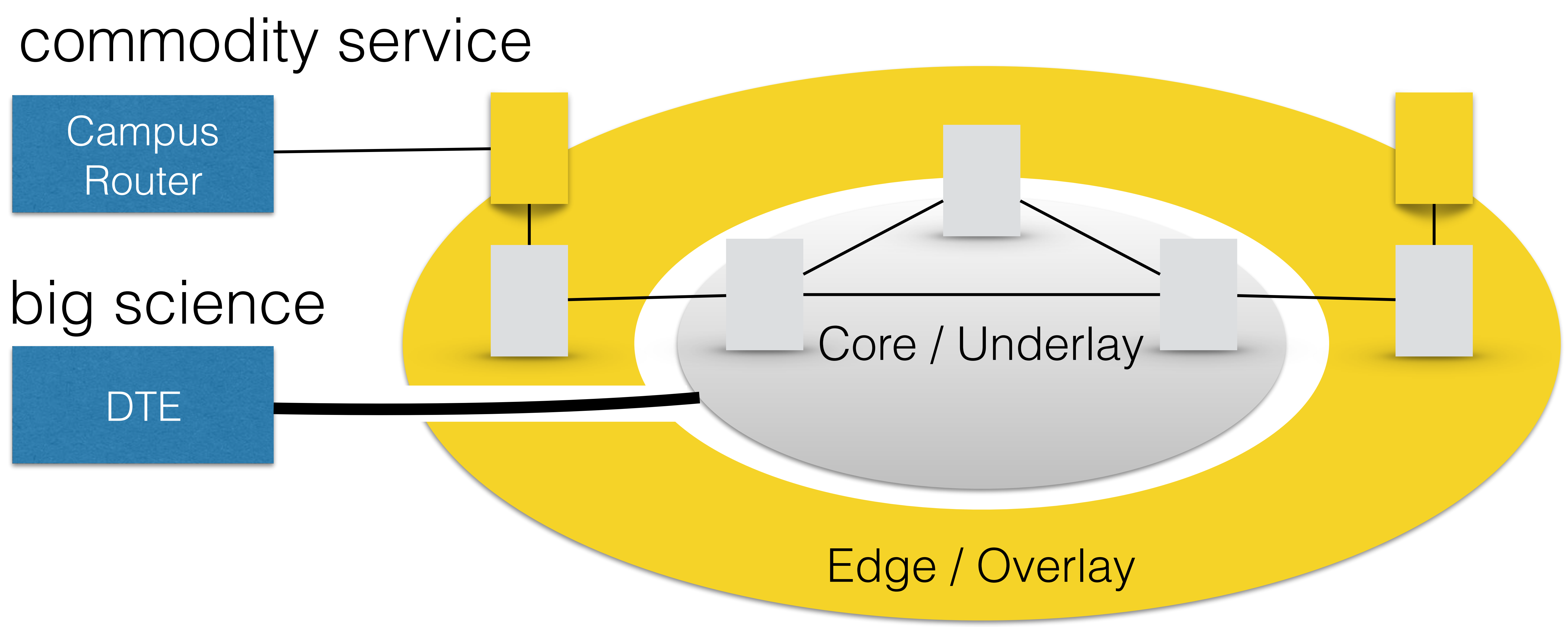
What can we gain from that fact ?

Big Science Can Possibly Violate The CE/PE Boundary

1. Core nodes do not run overlay protocols
2. ~~Edge nodes do not hand off underlay connectivity between domains~~



Many Many Challenges



How do we isolate commodity and big science in a unified core ?

How does a DTE determine a path through the core ?

How does an operator ensure trust with a campus ?

How do multiple science collaborations share a core ?

What management and stats interfaces need to exist ?

How does the NOC debug problems when they happen ?

There are solutions to these problems.
This is the community that can tackle the problem.

Nothing Ventured Nothing Gained.

Fun things to **Invent**

Edge based network path computation
Enhanced TCP / UDP transport assuming exclusive access to a link
Multi-Tenancy Bandwidth Calendar
Pseudo TDM Packet Protocols - (Personal Favourite)
Congestion management vs. Congestion Avoidance vs. Zero Congestion
QoS with Policing and Shaping rather than Weighted Fair Queuing
Minimal CE/PE Boundary

Grab a testbed and let’s see what we can break :)